



Panduan Manajemen

Amazon Redshift



Amazon Redshift: Panduan Manajemen

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

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan antara para pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau mungkin tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

.....	xvi
Apa itu Amazon Redshift?	1
Apakah Anda pengguna Amazon Redshift pertama kali?	1
Ikhtisar fitur Amazon Redshift Tanpa Server	2
Ikhtisar cluster yang disediakan Amazon Redshift	5
Manajemen kluster	5
Akses dan keamanan kluster	6
Cluster pemantauan	8
Basis Data	9
Membandingkan Amazon Redshift Tanpa Server dengan gudang data yang disediakan Amazon Redshift	9
Menggunakan antarmuka manajemen Amazon Redshift untuk kluster yang disediakan	36
Bekerja dengan AWS SDKs	37
Menandatangani permintaan HTTP	38
Menyiapkan Amazon Redshift CLI	44
Amazon Redshift Tanpa Server	46
Apa itu Amazon Redshift Tanpa Server?	46
Konsol Tanpa Server Amazon Redshift	47
Pertimbangan saat menggunakan Amazon Redshift Serverless	51
Kapasitas komputasi untuk Amazon Redshift Serverless	55
Pertimbangan dan batasan untuk kapasitas titik akhir tanpa server	57
Penskalaan dan pengoptimalan berbasis AI	59
Penagihan untuk Amazon Redshift Tanpa Server	65
Penagihan untuk kapasitas komputasi	65
Penagihan untuk kapasitas komputasi sesuai permintaan	66
Penagihan untuk reservasi tanpa server	71
Penagihan untuk penyimpanan	77
Menggunakan uji coba gratis Amazon Redshift Serverless	78
Catatan penggunaan penagihan	78
Amazon Redshift Penagihan tanpa server dengan penyatuan koneksi	80
Optimalisasi biaya untuk Amazon Redshift Serverless dengan Zero-ETL	81
Menghubungkan ke Amazon Redshift Tanpa Server	82
Menghubungkan ke Amazon Redshift Tanpa Server	82
Menghubungkan ke Amazon Redshift Tanpa Server melalui driver JDBC	82

Menghubungkan ke Amazon Redshift Tanpa Server dengan API Data	84
Menghubungkan dengan SSL ke Amazon Redshift Tanpa Server	84
Menghubungkan ke Amazon Redshift Tanpa Server dari titik akhir VPC yang dikelola Amazon Redshift	87
Menghubungkan ke Amazon Redshift Tanpa Server dari titik akhir VPC antarmuka ()AWS PrivateLink	87
Menghubungkan ke Amazon Redshift Tanpa Server dari titik akhir VPC Redshift di akun lain	87
Sumber daya tambahan	92
Mendefinisikan peran database untuk pengguna federasi	93
Mendefinisikan peran database	94
Kasus penggunaan untuk mendefinisikan peran basis data yang akan diberikan kepada pengguna federasi	94
Sumber daya tambahan	96
Manajemen identitas dan akses di Amazon Redshift Tanpa Server	96
Memberikan izin	96
Memulai dengan kredensi IAM	98
Mengakses objek database dengan izin peran basis data	99
Memigrasi kluster yang disediakan ke Amazon Redshift Tanpa Server	101
Membuat snapshot dari kluster yang Anda sediakan	101
Menghubungkan ke Amazon Redshift Tanpa Server menggunakan driver	102
Menggunakan Amazon Redshift Serverless SDK	108
Kelompok kerja dan ruang nama	109
Kelompok kerja dan ruang nama menggunakan konsol	109
Grup kerja dan ruang nama menggunakan API Tanpa Server Amazon AWS Command Line Interface Redshift dan Amazon	110
Kelompok kerja	111
Namespace	117
Memantau kueri dan beban kerja	121
Menambahkan kebijakan pemantauan kueri	122
Memberikan izin pemantauan kueri untuk pengguna	124
Memberikan izin pemantauan kueri untuk peran	125
Menetapkan batas penggunaan	125
Menetapkan batas kueri	126
Mengatur antrian kueri	126
Memeriksa data ringkasan menggunakan dasbor	131

Pencatatan audit	132
Log peristiwa di CloudWatch	132
CloudWatch metrik	133
Snapshot dan titik pemulihan	142
AWS Backup integrasi	143
Membuat snapshot	144
Membuat snapshot akhir	145
Berbagi snapshot atau menghapus izin snapshot	145
Menjadwalkan snapshot	145
Memperbarui periode retensi snapshot	149
Menghapus snapshot	149
Memulihkan snapshot	149
Mengonversi titik pemulihan	151
Memulihkan titik pemulihan	151
Menyalin cadangan ke yang lain AWS Region	151
Memulihkan tabel	153
Berbagi data	154
Pertimbangan-pertimbangan	155
Memberikan akses untuk melihat datashares	155
Mendaftarkan ruang nama ke AWS Glue Data Catalog	156
Memberikan tag ke sumber daya	156
Cluster yang disediakan Amazon Redshift	159
Cluster dan node	159
Rincian tipe simpul	161
Menentukan jumlah node	164
Gunakan EC2 untuk membuat cluster Anda	165
Amazon Virtual Private Cloud (Amazon VPC)	165
Alarm ruang disk default	165
Status klaster	166
Pertimbangan untuk menggunakan cluster yang disediakan	169
Pertimbangan Wilayah dan Availability Zone	169
Pemeliharaan cluster	169
Operasi klaster	180
Membuat kluster	180
Membuat alarm ruang disk	184
Melihat cluster	184

Mengubah klaster	185
Mengubah ukuran cluster	186
Mengganti nama cluster	202
Memutakhirkan versi rilis cluster	204
Menjeda dan melanjutkan cluster	204
Melakukan boot ulang klaster	206
Merelokasi cluster	207
Menetapkan batas penggunaan	211
Mematikan dan menghapus cluster	213
Snapshot dan backup	214
Mendaftarkan cluster ke AWS Glue Data Catalog	239
Deployment multi-AZ	240
Menyiapkan penyebaran Multi-AZ	241
Menyiapkan Multi-AZ saat membuat cluster baru	244
Menyiapkan Multi-AZ untuk gudang data yang dipulihkan dari snapshot	246
Mengonversi gudang data AZ tunggal menjadi gudang data multi-AZ	247
Mengonversi gudang data Multi-AZ menjadi gudang data AZ tunggal	249
Mengubah ukuran gudang data Multi-AZ	250
Gagal dalam penerapan Multi-AZ	251
Melihat kueri dan beban untuk gudang data multi-AZ	253
Memantau kueri dalam penerapan Multi-AZ	254
Mengakhiri kueri untuk klaster	255
Memantau kinerja cluster	255
Data kinerja	256
Melihat data kinerja	272
Menganalisis eksekusi kueri	295
Membuat alarm	297
Mengakhiri kueri yang sedang berjalan	298
Metrik kinerja di konsol CloudWatch	299
Profiler kueri	300
Pemantauan Kueri dan Database	310
Kueri dan Pemantauan Database Berbasis Sys View	317
Trek	320
Beralih antar trek	321
Melacak dan memulihkan	321
Memperbarui grup kerja tanpa server	322

Mengelola versi	323
Menentukan versi workgroup atau cluster	323
Integrasi nol-ETL	324
Pertimbangan-pertimbangan	326
Pertimbangan saat menggunakan mode riwayat pada target	328
Pertimbangan ketika sumber integrasi nol-ETL adalah Aurora atau Amazon RDS	329
Pertimbangan ketika sumber integrasi nol-ETL adalah DynamoDB	330
Pertimbangan ketika sumber integrasi nol-ETL adalah aplikasi, seperti, Salesforce, SAP,, dan Zendesk ServiceNow	331
Mulai menggunakan integrasi nol-ETL	332
Membuat dan mengonfigurasi gudang data Amazon Redshift target	332
Nyalakan sensitivitas casing	333
Konfigurasi otorisasi di Amazon Redshift	335
Buat integrasi nol-ETL	340
Membuat database tujuan	352
Meminta data yang direplikasi	355
Menanyakan data yang direplikasi dengan tampilan terwujud	355
Menanyakan data yang direplikasi dari DynamoDB	357
Melihat integrasi nol-ETL	358
Modus sejarah	361
Berbagi data Anda	364
Memantau integrasi nol-ETL	365
Memantau integrasi nol-ETL dengan tampilan sistem Amazon Redshift	365
Memantau integrasi nol-ETL dengan Amazon EventBridge	365
Metrik untuk integrasi nol-ETL	365
Memodifikasi integrasi nol-ETL untuk DynamoDB	367
Hapus integrasi nol-ETL untuk DynamoDB	368
Wilayah yang Didukung	369
Aurora MySQL	370
Aurora PostgreSQL	371
Amazon DynamoDB	373
Amazon RDS for MySQL	375
Aplikasi perusahaan	377
Memecahkan masalah integrasi nol-ETL	379
Kueri database	396
Menghubungkan ke Amazon Redshift	396

Mengkueri database menggunakan editor kueri Amazon Redshift v2	397
Mengkonfigurasi Akun AWS	398
Membuka editor kueri v2	405
Menghubungkan ke database Amazon Redshift	411
Menjelajahi database Amazon Redshift	413
Membuat objek database	415
Melihat kueri dan riwayat tab	422
Berinteraksi dengan SQL generatif Amazon Q	423
Memuat data ke dalam database	436
Kueri penulisan	446
Notebook	452
Menanyakan AWS Glue Data Catalog	455
Meminta data lake	459
Datashares	461
Pertanyaan terjadwal	464
Memvisualisasikan hasil	475
Berkolaborasi dan berbagi sebagai sebuah tim	480
Menanyakan database menggunakan editor kueri v1	482
Pertimbangan-pertimbangan	483
Menghubungkan ke gudang data menggunakan alat klien SQL	483
Rekomendasi untuk menghubungkan dengan alat klien	484
Mengonfigurasi koneksi di Amazon Redshift	485
Mengkonfigurasi opsi keamanan untuk koneksi	677
Menghubungkan dari alat dan kode klien	686
Menggunakan profil otentikasi untuk terhubung ke Amazon Redshift	737
Memecahkan masalah koneksi di Amazon Redshift	740
Menggunakan API Data	749
Bekerja dengan API Data	749
Pertimbangan saat memanggil Data API	750
Memilih kredensi otentikasi basis data	752
Memetakan tipe data JDBC	753
Menjalankan pernyataan SQL dengan parameter	754
Menjalankan pernyataan SQL dengan token idempotensi	756
Menjalankan pernyataan SQL dengan penggunaan kembali sesi	757
Mengambil hasil	759
Mengizinkan akses	762

Propagasi identitas tepercaya	771
Memanggil API Data	774
Memecahkan masalah API Data	803
Menjadwalkan operasi API Data dengan Amazon EventBridge	804
Memantau API Data	808
Menggunakan AWS KMS dengan API Data	810
Menggunakan Amazon Sagemaker Unified Studio	813
Grup parameter	815
Nilai parameter default	816
Manajemen beban kerja	818
Sifat dinamis dan statis WLM	818
Properti untuk parameter konfigurasi WLM	818
Mengkonfigurasi parameter WLM menggunakan AWS CLI	826
Membuat grup parameter	834
Mengubah grup parameter	835
Membuat aturan pemantauan kueri	839
Menghapus grup parameter	840
Integrasi dengan AWS Mitra	842
Memuat data dengan AWS mitra	843
Simpul terpesan	845
Penawaran node cadangan	845
Membandingkan harga di antara penawaran node yang dicadangkan	846
Cara kerja node yang dicadangkan	848
Node cadangan dan penagihan terkonsolidasi	849
Contoh simpul cadangan	850
Contoh 1	850
Contoh 2	850
Contoh 3	850
Contoh 4	851
Contoh 5	851
Contoh 6	851
Membeli simpul terpesan	851
Keamanan	854
Perlindungan data	856
Enkripsi data	857
Tokenisasi data	876

Perutean lalu lintas internetwork	876
Manajemen identitas dan akses	877
Otentikasi dengan identitas	878
Pencegahan "confused deputy" lintas layanan	879
Kontrol akses	880
Gambaran umum manajemen akses	880
Menggunakan kebijakan berbasis identitas (kebijakan IAM)	887
Federasi penyedia identitas asli (iDP)	951
Menggunakan Peran Terkait Layanan	984
Menggunakan otentikasi IAM untuk menghasilkan kredensi pengguna basis data	993
Mengotorisasi akses ke layanan AWS	1049
Mengelola kata sandi admin	1086
Izin diperlukan untuk integrasi AWS Secrets Manager	1087
Rotasi rahasia kata sandi admin	1087
Pertimbangan menggunakan Amazon AWS Secrets Manager Redshift	1088
Mengambil ARN rahasia	1088
Membuat rahasia untuk kredensi koneksi database	1089
Pencatatan log dan pemantauan	1092
Pencatatan audit basis data	1093
Logging dengan CloudTrail	1110
Validasi kepatuhan	1122
Ketahanan	1123
Keamanan infrastruktur	1124
Isolasi jaringan	876
Grup keamanan	1126
Titik akhir VPC antarmuka	1126
Konfigurasi dan analisis kerentanan	1134
Menggunakan otentikasi Pusat Identitas	1135
Prasyarat	1135
Cara kerja otentikasi Pusat Identitas	1136
GetIdentityCenterAuthToken Operasi API	1136
GetIdentityCenterAuthToken untuk cluster yang disediakan	1136
GetIdentityCenterAuthToken untuk kelompok kerja tanpa server	1137
Sintaksis Respons	1138
Integrasi driver	1139
Penggunaan token langsung	1139

Contoh kode Java	1140
Persyaratan kebijakan IAM	1140
Izin API:	1141
Contoh kebijakan IAM	1141
Ketersediaan wilayah	1142
Pemecahan masalah	1143
Kesalahan API	1143
Kesalahan koneksi	1143
Tugas jaringan	1145
Nama domain khusus untuk koneksi klien	1145
Mendaftarkan nama domain	1146
Meminta sertifikat untuk nama domain	1147
Mengkonfigurasi domain kustom	1148
Menghubungkan ke klaster atau grup kerja yang telah disediakan	1150
Mengganti nama cluster yang memiliki domain kustom yang ditetapkan	1151
Menjelaskan asosiasi domain kustom	1151
Mengaitkan domain kustom dengan sertifikat yang berbeda	1152
Menghapus domain kustom	1153
Titik akhir VPC yang dikelola Redshift	1154
Pertimbangan-pertimbangan	1155
Memberikan akses ke VPC	1156
Membuat titik akhir VPC yang dikelola Redshift	1157
Sumber daya Redshift dalam VPC	1158
Membuat cluster atau workgroup di VPC	1161
Grup keamanan VPC	1162
Mengkonfigurasi pengaturan keamanan untuk klaster atau grup kerja	1163
Subnet untuk sumber daya Redshift	1167
Memblokir akses publik ke VPCs dan subnet	1170
Mengontrol lalu lintas jaringan dengan perutean VPC yang ditingkatkan	1171
Mengontrol lalu lintas database dengan titik akhir VPC	1173
Mengaktifkan perutean VPC yang disempurnakan	1174
Mengakses bucket Amazon S3 dengan Redshift Spectrum	1176
Peristiwa	1182
Langganan pemberitahuan acara cluster	1182
Membuat langganan pemberitahuan acara	1185
Pemberitahuan acara cluster yang disediakan	1185

Pemberitahuan acara Amazon Redshift Tanpa Server	1208
Pemberitahuan acara integrasi nol-ETL	1220
Kuota dan batas	1230
Kuota untuk objek Amazon Redshift	1230
Kuota untuk objek Amazon Redshift Tanpa Server	1237
Kuota untuk Amazon Redshift Data API	1241
Kuota untuk objek editor kueri v2	1244
Kuota dan batas untuk objek Amazon Redshift Spectrum	1245
Kendala penamaan	1246
Memberi tanda pada sumber daya	1250
Persyaratan penandaan	1251
Mengelola tag sumber daya	1251
AWS Backup integrasi	1253
Pertimbangan-pertimbangan	1254
Batasan	1255
Menggunakan AWS Backup dengan Amazon Redshift	1255
Versi cluster	1257
Tambalan 199	1258
Fitur baru	1258
Tambalan 198	1260
Fitur baru	1261
Tambalan 197	1262
Fitur baru	1263
Tambalan 196	1265
Fitur baru	1265
Tambalan 195	1265
Fitur baru	1266
Tambalan 194	1266
Fitur baru	1267
Tambalan 193	1267
Fitur baru	1268
Tambalan 192	1268
Fitur baru	1269
Tambalan 191	1269
Fitur baru	1269
Tambalan 190	1270

Fitur baru	1270
Tambalan 189	1271
Fitur baru	1271
Tambalan 188	1272
Fitur baru	1272
Tambalan 187	1273
Fitur baru	1274
Tambalan 186	1275
Fitur baru	1276
Tambalan 185	1277
Fitur baru	1278
Tambalan 184	1279
Fitur baru	1279
Tambalan 183	1280
Fitur baru	1281
Tambalan 182	1282
Fitur baru	1282
Tambalan 181	1284
Fitur baru	1284
Tambalan 180	1285
Fitur baru	1286
Tambalan 179	1287
Fitur baru	1288
Tambalan 178	1289
Fitur baru	1290
Tambalan 177	1292
Fitur baru	1293
Tambalan 176	1294
Fitur baru	1295
Tambalan 175	1296
Fitur baru	1296
Tambalan 174	1297
Fitur baru untuk versi ini	1297
Fitur baru untuk versi ini	1297
Fitur baru untuk versi ini	1297
Fitur baru untuk versi ini	1297

Fitur baru untuk versi ini	1297
Fitur baru untuk versi ini	1297
Fitur baru untuk versi ini	1297
Tambahan 173	1298
Fitur baru untuk versi ini	1298
Fitur baru untuk versi ini	1298
Fitur baru untuk versi ini	1298
Fitur baru untuk versi ini	1298
Fitur baru untuk versi ini	1298
Fitur baru untuk versi ini	1298
Fitur baru untuk versi ini	1298
Fitur baru untuk versi ini	1298
Fitur baru untuk versi ini	1298
Tambahan 172	1300
Fitur baru	1300
Tambahan 171	1301
Fitur baru	1301
Tambahan 170	1301
Fitur baru	1302
Tambahan 169	1302
Fitur baru	1302
Tambahan 168	1302
Fitur baru	1303
Perubahan perilaku	1304
Perubahan perilaku yang akan datang	1304
Scalar UDFs Python akan mencapai akhir dukungan setelah 30 Juni 2026	1304
Perubahan Perilaku Penyegaran Otomatis Tampilan Terwujud (MV) setelah 27 Februari 2026	1305
Amazon Redshift tidak akan mendukung fungsi yang mengakses informasi konsumen melalui datasharing setelah 16 Februari 2026	1305
Perubahan versi Minimum Transport Layer Security (TLS) efektif mulai 31 Januari 2026 ...	1305
Amazon Redshift tidak akan mendukung pembuatan UDFs Python skalar baru setelah 30 Oktober 2025	1306
Perubahan perilaku terbaru	1307

Amazon Redshift menggunakan Database Zona Waktu up-to-date IANA setelah 26 Agustus 2025	1307
Amazon Redshift Serverless RPU berubah efektif setelah 15 Agustus 2025	1308
Perubahan pencatatan audit basis data efektif setelah 10 Agustus 2025	1308
Perubahan Titik Akhir Cloud Pribadi Virtual untuk grup kerja tanpa server efektif setelah 27 Juni 2025	1309
Perubahan pemantauan kueri efektif setelah 2 Mei 2025	1309
Perubahan keamanan efektif setelah 10 Januari 2025	1310
Contoh kode	1313
Hal-hal mendasar	1314
Halo Amazon Redshift	1314
Pelajari dasar-dasarnya	1319
Tindakan	1385
Skenario	1443
Buat aplikasi web untuk melacak data Amazon Redshift	1444
Riwayat dokumen	1445

Amazon Redshift tidak akan lagi mendukung pembuatan Python UDFs baru mulai Patch 198. Python yang ada UDFs akan terus berfungsi hingga 30 Juni 2026. Untuk informasi lebih lanjut, lihat [posting blog](#).

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.

Apa itu Amazon Redshift?

Selamat datang di Panduan Manajemen Pergeseran Merah Amazon. Amazon Redshift adalah layanan gudang data dengan skala petabyte yang terkelola penuh di cloud. Amazon Redshift Serverless memungkinkan Anda mengakses dan menganalisis data tanpa semua konfigurasi gudang data yang disediakan. Sumber daya secara otomatis disediakan dan kapasitas gudang data diskalakan secara cerdas untuk memberikan kinerja yang cepat bahkan untuk beban kerja yang paling menuntut dan tidak dapat diprediksi. Anda tidak dikenakan biaya saat gudang data menganggur, jadi Anda hanya membayar untuk apa yang Anda gunakan. Anda dapat memuat data dan mulai melakukan kueri segera di editor kueri Amazon Redshift v2 atau di alat intelijen bisnis (BI) favorit Anda. Nikmati kinerja harga terbaik dan fitur SQL yang akrab di lingkungan easy-to-use administrasi nol.

Terlepas dari ukuran kumpulan data, Amazon Redshift menawarkan kinerja kueri cepat menggunakan alat berbasis SQL dan aplikasi intelijen bisnis yang sama yang Anda gunakan saat ini.

Apakah Anda pengguna Amazon Redshift pertama kali?

Jika Anda adalah pengguna pertama kali Amazon Redshift, kami sarankan Anda mulai dengan membaca bagian berikut:

- [Sorotan dan Harga Layanan](#) — Halaman detail produk ini menyediakan proposisi nilai Amazon Redshift, sorotan layanan, dan harga.
- [Memulai gudang data Amazon Redshift Tanpa Server — Topik ini memandu Anda melalui proses penyiapan gudang](#) data tanpa server, membuat sumber daya, dan menanyakan data sampel.
- [Panduan Pengembang Database Amazon Redshift](#) — Jika Anda seorang pengembang database, panduan ini menjelaskan cara mendesain, membuat, menanyakan, dan memelihara database yang membentuk gudang data Anda.

Jika Anda lebih suka mengelola sumber daya Amazon Redshift secara manual, Anda dapat membuat klaster yang disediakan untuk kebutuhan kueri data Anda. Untuk informasi selengkapnya, lihat [klaster Amazon Redshift](#).

Sebagai pengembang aplikasi, Anda dapat menggunakan library Amazon Redshift API atau AWS Software Development Kit (SDK) untuk mengelola cluster secara terprogram. Jika Anda menggunakan Amazon Redshift API, Anda harus mengautentikasi setiap permintaan HTTP atau

HTTPS ke API dengan menandatangani. Untuk informasi lebih lanjut tentang penandatanganan permintaan, buka [Menandatangani permintaan HTTP](#).

Untuk informasi tentang API, CLI, dan SDKs, buka tautan berikut:

- [Referensi API Tanpa Server Amazon Redshift](#)
- [Referensi API Amazon Redshift](#)
- [Referensi API API Data Amazon Redshift](#)
- [AWS CLI Referensi Perintah](#)
- Referensi SDK di [Alat untuk Amazon Web Services](#).

Ikhtisar fitur Amazon Redshift Tanpa Server

Sebagian besar fitur yang didukung oleh gudang data yang disediakan Amazon Redshift juga didukung oleh Amazon Redshift Serverless. Berikut ini adalah beberapa kemampuan utamanya.

Fitur	Deskripsi
Cuplikan	Anda dapat memulihkan snapshot Amazon Redshift Tanpa Server atau gudang data yang disediakan ke Amazon Redshift Serverless. Untuk informasi selengkapnya, lihat Snapshot dan titik pemulihan .
Poin pemulihan	Amazon Redshift Serverless secara otomatis menciptakan titik pemulihan setiap 30 menit. Poin pemulihan ini disimpan selama 24 jam. Anda dapat menggunakannya untuk memulihkan setelah menulis atau menghapus secara tidak sengaja. Saat Anda memulihkan dari titik pemulihan, semua data di database Amazon Redshift Tanpa Server Anda dikembalikan ke titik waktu sebelumnya. Anda juga dapat membuat snapshot dari titik pemulihan jika Anda perlu mempertahankan titik pemulihan untuk jangka waktu yang lebih lama. Untuk informasi selengkapnya, lihat Snapshot dan titik pemulihan .
Kapasitas RPU dasar	Anda dapat mengatur kapasitas dasar di Redshift Processing Units (RPU). Satu RPU menyediakan memori 16 GB. Pengaturan ini memberi Anda kemampuan untuk mengontrol keseimbangan antara sumber daya yang digunakan dan biaya untuk beban kerja Anda. Anda dapat meningkatkan nilai ini untuk menumbuhkan sumber daya yang tersedia dan meningkatkan kinerja kueri, atau menurunkan nilai untuk membatasi pengeluaran Anda. Defaultnya adalah 128 RPUs. Anda

Fitur	Deskripsi
	juga dapat menetapkan batas penggunaan, seperti RPU digunakan per hari, untuk mengontrol biaya. Untuk informasi selengkapnya, lihat Penagihan untuk Amazon Redshift Tanpa Server .
Batas penggunaan berbagi data	Anda dapat membatasi jumlah data yang ditransfer dari Wilayah produsen ke Wilayah konsumen menggunakan konsol atau API. Biaya transfer data ini berbeda AWS Region, dan diukur dalam terabyte. Untuk informasi selengkapnya tentang berbagi data, lihat Memulai berbagi data menggunakan konsol di Panduan Pengembang Database Amazon Redshift .
Fungsi yang ditentukan pengguna () UDFs	Anda dapat menjalankan fungsi yang ditentukan pengguna (UDFs) di Amazon Redshift Tanpa Server. Untuk informasi selengkapnya, lihat Membuat fungsi yang ditentukan pengguna di Panduan Pengembang Database Amazon Redshift .
Prosedur tersimpan	Anda dapat menjalankan prosedur tersimpan di Amazon Redshift Serverless. Untuk informasi selengkapnya, lihat Membuat prosedur tersimpan di Panduan Pengembang Database Amazon Redshift.
Pandangan terwujud	Anda dapat membuat tampilan terwujud di Amazon Redshift Tanpa Server. Untuk informasi selengkapnya, lihat Membuat tampilan terwujud di Panduan Pengembang Database Amazon Redshift .
Fungsi spasial	Anda dapat menjalankan fungsi spasial di Amazon Redshift Serverless. Untuk informasi selengkapnya, lihat Mengkueri data spasial di Panduan Pengembang Database Amazon Redshift.
Kueri federasi	Anda dapat menjalankan kueri untuk menggabungkan data dengan cluster Aurora DB dan database Amazon RDS dari Amazon Redshift Serverless. Untuk informasi selengkapnya, lihat Mengkueri data dengan kueri gabungan di Panduan Pengembang Database Amazon Redshift.
Kueri data lake	Anda dapat menjalankan kueri untuk menggabungkan data dari danau data Amazon S3 Anda dengan Amazon Redshift Tanpa Server. Untuk informasi selengkapnya, lihat Menanyakan data lake di Panduan Manajemen Pergeseran Merah Amazon.

Fitur	Deskripsi
HyperLogLog	Anda dapat menjalankan HyperLogLog fungsi di Amazon Redshift Serverless. Untuk informasi selengkapnya, lihat Menggunakan HyperLogLog sketsa di Panduan Pengembang Database Amazon Redshift.
Kueri data di seluruh basis data	Anda dapat melakukan kueri data di seluruh database dengan Amazon Redshift Serverless. Untuk informasi selengkapnya, lihat Mengkueri data di seluruh database di Panduan Pengembang Database Amazon Redshift.
Berbagi data	Anda dapat mengakses datashares di gudang data yang disediakan dengan Amazon Redshift Serverless. Untuk informasi selengkapnya, lihat Berbagi data di seluruh cluster di Panduan Pengembang Database Amazon Redshift.
Kueri data semi-terstruktur	Anda dapat menelan dan menyimpan data semi-terstruktur dengan tipe SUPER data dengan Amazon Redshift Serverless. Untuk informasi selengkapnya, lihat Menyerap dan menanyakan data semi-terstruktur di Panduan Pengembang Database Amazon Redshift.
Sumber daya penandaan	Anda dapat menggunakan API Tanpa Server Amazon Redshift AWS CLI atau Amazon Redshift untuk menandai sumber daya dengan metadata yang terkait dengan sumber daya. Untuk informasi selengkapnya, lihat Menandai sumber daya .
Pembelajaran mesin	Anda dapat menggunakan pembelajaran mesin Amazon Redshift dengan Amazon Redshift Serverless. Untuk informasi selengkapnya, lihat Menggunakan pembelajaran mesin di Panduan Pengembang Database Amazon Redshift.
Perintah dan fungsi SQL	Dengan beberapa pengecualian (seperti REBOOT_CLUSTER), Anda dapat menggunakan perintah dan fungsi Amazon Redshift SQL dengan Amazon Redshift Serverless. Untuk informasi selengkapnya, lihat referensi SQL di Panduan Pengembang Database Amazon Redshift.

Fitur	Deskripsi
Sumber daya CloudFormation	Dengan menggunakan CloudFormation templat, Anda dapat menerapkan dan memperbarui sumber daya Amazon Redshift Tanpa Server. Integrasi ini berarti Anda dapat menghabiskan lebih sedikit waktu untuk mengelola sumber daya dan fokus pada aplikasi Anda. Untuk informasi selengkapnya tentang CloudFormation sumber daya di Amazon Redshift Tanpa Server, lihat referensi jenis sumber daya Amazon Redshift Tanpa Server.
CloudTrail sumber daya	Amazon Redshift Serverless terintegrasi AWS CloudTrail untuk memberikan catatan tindakan yang diambil di Amazon Redshift Tanpa Server. CloudTrail menangkap semua panggilan API untuk Amazon Redshift Serverless sebagai peristiwa. Untuk informasi selengkapnya, lihat CloudTrail Amazon Redshift Tanpa Server.

Ikhtisar cluster yang disediakan Amazon Redshift

Layanan Amazon Redshift mengelola semua pekerjaan pengaturan, pengoperasian, dan penskalaan gudang data. Tugas-tugas ini termasuk penyediaan kapasitas, pemantauan dan pencadangan cluster, serta menerapkan tambalan dan peningkatan ke mesin Amazon Redshift.

Video berikut menunjukkan cara membuat klaster dan data kueri menggunakan editor kueri Amazon Redshift v2.

Manajemen klaster

Cluster Amazon Redshift adalah sekumpulan node, yang terdiri dari node pemimpin dan satu atau lebih node komputasi. Jenis dan jumlah node komputasi yang Anda butuhkan bergantung pada ukuran data Anda, jumlah kueri yang akan Anda jalankan, dan kinerja runtime kueri yang Anda butuhkan.

Membuat dan mengelola cluster

Bergantung pada kebutuhan pergudangan data Anda, Anda dapat memulai dengan cluster simpul tunggal yang kecil dan dengan mudah meningkatkan skala ke cluster multi-node yang lebih besar saat kebutuhan Anda berubah. Anda dapat menambah atau menghapus node komputasi ke cluster tanpa gangguan pada layanan. Untuk informasi selengkapnya, lihat [Cluster yang disediakan Amazon Redshift](#).

Reservasi node komputasi

Jika Anda berniat untuk menjaga klaster Anda berjalan selama satu tahun atau lebih, Anda dapat menghemat uang dengan memesan node komputasi untuk periode satu tahun atau tiga tahun. Reserving compute node menawarkan penghematan yang signifikan dibandingkan dengan tarif per jam yang Anda bayar saat Anda menyediakan node komputasi sesuai permintaan. Untuk informasi selengkapnya, lihat [Simpul terpesan](#).

Membuat snapshot cluster

Snapshot adalah point-in-time backup dari sebuah cluster. Ada dua jenis snapshot: otomatis dan manual. Amazon Redshift menyimpan snapshot ini secara internal di Amazon Simple Storage Service (Amazon S3) dengan menggunakan koneksi Secure Sockets Layer (SSL) terenkripsi. Jika Anda perlu memulihkan dari snapshot, Amazon Redshift membuat klaster baru dan mengimpor data dari snapshot yang Anda tentukan. Untuk informasi selengkapnya tentang snapshot, lihat [Cuplikan dan cadangan Amazon Redshift](#).

Akses dan keamanan klaster

Ada beberapa fitur yang terkait dengan akses cluster dan keamanan di Amazon Redshift. Fitur-fitur ini membantu Anda mengontrol akses ke cluster Anda, menentukan aturan konektivitas, dan mengenkripsi data dan koneksi. Fitur-fitur ini merupakan tambahan fitur yang terkait dengan akses database dan keamanan di Amazon Redshift. Untuk informasi selengkapnya tentang keamanan database, lihat [Mengelola Keamanan Database](#) di Panduan Pengembang Database Amazon Redshift.

AWS akun dan kredensi IAM

Secara default, cluster Amazon Redshift hanya dapat diakses oleh AWS akun yang membuat cluster. Cluster dikunci sehingga tidak ada orang lain yang memiliki akses. Dalam AWS akun Anda, Anda menggunakan layanan AWS Identity and Access Management (IAM) untuk membuat akun pengguna dan mengelola izin untuk akun tersebut untuk mengontrol operasi klaster. Untuk informasi selengkapnya, lihat [Keamanan di Amazon Redshift](#). Untuk informasi selengkapnya tentang mengelola identitas IAM, termasuk panduan dan praktik terbaik untuk peran IAM, lihat [Manajemen identitas dan akses di Amazon Redshift](#).

Grup keamanan

Secara default, cluster apa pun yang Anda buat ditutup untuk semua orang. Kredensi IAM hanya mengontrol akses ke sumber daya terkait Amazon Redshift API: konsol Amazon Redshift, antarmuka

baris perintah (CLI), API, dan SDK. Untuk mengaktifkan akses ke cluster dari alat klien SQL melalui JDBC atau ODBC, Anda menggunakan grup keamanan:

- Jika Anda menggunakan platform EC2-VPC untuk cluster Amazon Redshift Anda, Anda harus menggunakan grup keamanan VPC. Kami menyarankan Anda meluncurkan cluster Anda di platform EC2-VPC.

Anda tidak dapat memindahkan cluster ke VPC setelah diluncurkan dengan EC2-Classic. Namun, Anda dapat memulihkan snapshot EC2-Classic ke cluster EC2-VPC menggunakan konsol Amazon Redshift. Untuk informasi selengkapnya, lihat [Memulihkan cluster dari snapshot](#).

- Jika Anda menggunakan platform EC2-Classic untuk cluster Amazon Redshift Anda, Anda harus menggunakan grup keamanan Amazon Redshift.

Dalam kedua kasus tersebut, Anda menambahkan aturan ke grup keamanan untuk memberikan akses masuk eksplisit ke rentang CIDR/IP alamat tertentu atau ke grup keamanan Amazon Elastic Compute Cloud (Amazon EC2) jika klien SQL Anda berjalan pada instans Amazon EC2. Untuk informasi selengkapnya, lihat [Grup keamanan Amazon Redshift](#).

Selain aturan akses masuk, Anda membuat pengguna database untuk memberikan kredensi untuk mengautentikasi ke database dalam cluster itu sendiri. Untuk informasi lebih lanjut, lihat [Basis Data](#) di topik ini.

Enkripsi

Saat Anda menyediakan klaster, Anda dapat memilih untuk mengenkripsi klaster untuk keamanan tambahan. Saat Anda mengaktifkan enkripsi, Amazon Redshift menyimpan semua data dalam tabel yang dibuat pengguna dalam format terenkripsi. Anda dapat menggunakan AWS Key Management Service (AWS KMS) untuk mengelola kunci enkripsi Amazon Redshift Anda.

Enkripsi adalah properti abadi dari cluster. Satu-satunya cara untuk beralih dari cluster terenkripsi ke cluster yang tidak dienkripsi adalah dengan membongkar data dan memuatnya kembali ke cluster baru. Enkripsi berlaku untuk cluster dan cadangan apa pun. Saat Anda memulihkan cluster dari snapshot terenkripsi, cluster baru juga dienkripsi.

Untuk informasi selengkapnya tentang enkripsi, kunci, dan modul keamanan perangkat keras, lihat [Enkripsi basis data Amazon Redshift](#).

Koneksi SSL

Anda dapat menggunakan enkripsi Secure Sockets Layer (SSL) untuk mengenkripsi koneksi antara klien SQL dan cluster Anda. Untuk informasi selengkapnya, lihat [Mengkonfigurasi opsi keamanan untuk koneksi](#).

Cluster pemantauan

Ada beberapa fitur yang terkait dengan pemantauan di Amazon Redshift. Anda dapat menggunakan pencatatan audit database untuk menghasilkan log aktivitas, mengonfigurasi peristiwa, dan langganan notifikasi untuk melacak informasi yang diminati. Gunakan metrik di Amazon Redshift dan CloudWatch Amazon untuk mempelajari kesehatan dan kinerja cluster dan database Anda.

Pencatatan audit basis data

Anda dapat menggunakan fitur pencatatan audit database untuk melacak informasi tentang upaya otentikasi, koneksi, pemutusan, perubahan definisi pengguna database, dan kueri yang dijalankan dalam database. Informasi ini berguna untuk tujuan keamanan dan pemecahan masalah di Amazon Redshift. Log disimpan dalam ember Amazon S3. Untuk informasi selengkapnya, lihat [Pencatatan audit basis data](#).

Peristiwa dan notifikasi

Amazon Redshift melacak peristiwa dan menyimpan informasi tentangnya selama beberapa minggu di akun Anda. AWS Untuk setiap peristiwa, Amazon Redshift melaporkan informasi seperti tanggal peristiwa terjadi, deskripsi, sumber peristiwa (misalnya, cluster, grup parameter, atau snapshot), dan ID sumber. Anda dapat membuat langganan notifikasi acara Amazon Redshift yang menentukan satu set filter peristiwa. Saat terjadi peristiwa yang sesuai dengan kriteria filter, Amazon Redshift menggunakan Layanan Pemberitahuan Sederhana Amazon untuk memberi tahu Anda bahwa peristiwa tersebut telah terjadi. Untuk informasi selengkapnya tentang acara dan notifikasi, lihat [Acara Amazon Redshift](#).

Performa

Amazon Redshift menyediakan metrik kinerja dan data sehingga Anda dapat melacak kesehatan dan kinerja cluster dan database Anda. Amazon Redshift menggunakan CloudWatch metrik Amazon untuk memantau aspek fisik cluster, seperti pemanfaatan CPU, latensi, dan throughput. Amazon Redshift juga menyediakan kueri dan memuat data performa untuk membantu Anda memantau aktivitas database di klaster. Untuk informasi selengkapnya tentang metrik dan pemantauan kinerja, lihat [Memantau kinerja klaster Amazon Redshift](#).

Basis Data

Amazon Redshift membuat satu database saat Anda menyediakan klaster. Ini adalah database yang Anda gunakan untuk memuat data dan menjalankan kueri pada data Anda. Anda dapat membuat database tambahan sesuai kebutuhan dengan menjalankan perintah SQL. Untuk informasi selengkapnya tentang membuat database tambahan, buka [Langkah 1: Buat database](#) di Panduan Pengembang Database Amazon Redshift.

Saat Anda menyediakan klaster, Anda menentukan pengguna admin yang memiliki akses ke semua database yang dibuat di dalam klaster. Pengguna admin ini adalah pengguna super yang merupakan satu-satunya pengguna dengan akses ke database pada awalnya, meskipun pengguna ini dapat membuat pengguna super dan pengguna tambahan. Untuk informasi selengkapnya, buka [Superusers dan Users](#) di Amazon Redshift Database Developer Guide.

Amazon Redshift menggunakan grup parameter untuk menentukan perilaku semua database dalam klaster, seperti gaya presentasi tanggal dan presisi floating-point. Jika Anda tidak menentukan grup parameter saat menyediakan klaster, Amazon Redshift akan mengaitkan grup parameter default dengan klaster. Untuk informasi selengkapnya, lihat [Grup parameter Amazon Redshift](#).

Untuk informasi selengkapnya tentang database di Amazon Redshift, buka Panduan Pengembang Database [Amazon Redshift](#).

Membandingkan Amazon Redshift Tanpa Server dengan gudang data yang disediakan Amazon Redshift

Untuk Amazon Redshift Tanpa Server, beberapa konsep dan fitur berbeda dari fitur yang sesuai untuk gudang data yang disediakan Amazon Redshift. Misalnya, satu perbandingan yang kontras adalah bahwa Amazon Redshift Serverless tidak memiliki konsep cluster atau node. Tabel berikut menjelaskan fitur dan perilaku di Amazon Redshift Tanpa Server dan menjelaskan perbedaannya dari fitur yang setara di gudang data yang disediakan.

Fitur	Deskripsi	Nirserver	Disediakan
Workgroup dan Namespace	Untuk mengisolasi beban kerja dan	Namespace adalah kumpulan objek	Cluster yang disediakan adalah kumpulan node komputasi dan node pemimpin, yang Anda kelola secara langsung. Untuk informasi selengkapnya, lihat Cluster yang disediakan Amazon Redshift .

Fitur	Deskripsi	Nirserver	Disediakan
	mengelola sumber daya yang berbeda di Amazon Redshift Tanpa Server, Anda dapat membuat ruang nama dan grup kerja untuk mengelola penyimpanan dan menghitung sumber daya secara terpisah.	database dan pengguna. Workgroup adalah kumpulan sumber daya komputasi. Untuk informasi selengkapnya, lihat Amazon Redshift Tanpa Server untuk memahami desain Amazon Redshift Serverless.	

Fitur	Deskripsi	Nirserver	Disediakan
Jenis simpul	Saat Anda bekerja dengan Amazon Redshift Serverless, Anda tidak memilih jenis node atau menentukan jumlah node seperti yang Anda lakukan dengan cluster Amazon Redshift yang disediakan.	Amazon Redshift Serverless secara otomatis menyediakan dan mengelola kapasitas untuk Anda. Anda dapat secara opsional menentukan kapasitas gudang data dasar untuk memilih price/performance saldo yang tepat untuk beban kerja Anda. Anda juga dapat menentukan jam RPU maksimum untuk mengatur kontrol	Anda membangun klaster dengan tipe node yang memenuhi spesifikasi biaya dan kinerja Anda. Untuk informasi selengkapnya, lihat Cluster yang disediakan Amazon Redshift .

Fitur	Deskripsi	Nirserver	Disediakan
		biaya untuk memastikan bahwa biaya dapat diprediksi. Untuk informasi selengkapnya, lihat Kapasitas komputasi untuk Amazon Redshift Serverless .	

Fitur	Deskripsi	Nirserver	Disediakan
Manajemen beban kerja dan penskalaan konkurensi	Amazon Redshift dapat menskalakan untuk periode beban berat. Amazon Redshift Serverless juga dapat menskalakan untuk memenuhi periode beban tinggi yang terputus-putus.	Amazon Redshift Serverless secara otomatis mengelola sumber daya secara efisien dan menskalakan, berdasarkan beban kerja, dalam ambang batas kontrol biaya. Untuk informasi selengkapnya, lihat Penagihan untuk kapasitas komputasi .	Dengan gudang data yang disediakan, Anda mengaktifkan penskalaan konkurensi pada klaster Anda untuk menangani periode beban berat. Untuk informasi selengkapnya, lihat Penskalaan konkurensi .

Fitur	Deskripsi	Nirserver	Disediakan
Port	Nomor port yang Anda gunakan untuk menghubungkan.	Dengan Amazon Redshift Serverless, Anda dapat mengubah ke port lain dari rentang port 5431—5455 atau 8191—8215. Untuk informasi selengkapnya, lihat Menghubungkan ke Amazon Redshift Tanpa Server .	Dengan gudang data yang disediakan, Anda dapat memilih port apa pun untuk dihubungkan.

Fitur	Deskripsi	Nirserver	Disediakan
Mengubah ukuran	Menambah atau menghapus sumber daya komputasi untuk bekerja dengan baik untuk beban kerja.	Mengubah ukuran tidak berlaku di Amazon Redshift Tanpa Server. Namun Anda dapat mengubah kapasitas RPU gudang data dasar, berdasarkan an persyaratan harga dan kinerja Anda. Untuk informasi selengkapnya, lihat Kapasitas komputasi untuk Amazon Redshift Serverless .	Dengan kluster yang disediakan, Anda melakukan perubahan ukuran cluster untuk menambahkan node atau menghapus node. Untuk informasi selengkapnya, lihat Ikhtisar pengelolaan kluster di Amazon Redshift .

Fitur	Deskripsi	Nirserver	Disediakan
Menjeda dan melanjutkan	Anda dapat menjeda klaster yang disediakan saat Anda tidak memiliki beban kerja untuk dijalankan, untuk menghemat biaya.	Dengan Amazon Redshift Serverless, Anda hanya membayar saat kueri berjalan, jadi tidak perlu menjeda atau melanjutkan. Untuk informasi selengkapnya, lihat Penagihan untuk kapasitas komputasi .	Anda menjeda dan melanjutkan cluster secara manual, berdasarkan penilaian beban kerja Anda di berbagai waktu. Untuk informasi selengkapnya, lihat Ikhtisar pengelolaan klaster di Amazon Redshift .

Fitur	Deskripsi	Nirserver	Disediakan
Menanyakan data eksternal dengan kueri Spectrum	Anda dapat melakukan kueri data di bucket Amazon S3, dalam berbagai format, seperti JSON.	Penagihan bertambah saat menghitung sumber daya memproses beban kerja. Juga, penagihan bertambah ketika data Redshift Spectrum eksternal ditanyakan, seperti transaksi lainnya. Untuk informasi selengkapnya, lihat Penagihan untuk kapasitas komputasi .	Dengan gudang data yang disediakan, kapasitas Amazon Redshift Spectrum ada di server terpisah yang ditanyakan dari cluster Amazon Redshift. Untuk informasi selengkapnya, lihat Menanyakan data eksternal menggunakan Amazon Redshift Spectrum .

Fitur	Deskripsi	Nirserver	Disediakan
Tagihan sumber daya komputasi	Bagaimana penagihan bertambah untuk Amazon Redshift vs Amazon Redshift Tanpa Server.	Dengan Amazon Redshift Serverless, Anda membayar beban kerja yang Anda jalankan, dalam RPU-jam per detik, dengan biaya minimum 60 detik. Ini termasuk kueri yang mengakses data dalam format file terbuka di Amazon S3. Untuk informasi selengkapnya, lihat Penagihan untuk kapasitas komputasi .	Dengan klaster yang disediakan, penagihan terjadi per detik saat klaster tidak dijeda.

Fitur	Deskripsi	Nirserver	Disediakan
Jendela pemeliharaan	Cara kerja pemeliharaan server.	Dengan Amazon Redshift Serverless, tidak ada jendela pemeliharaan. Pembaruan ditangani dengan mulus. Untuk informasi lebih lanjut, lihat Apa itu Amazon Redshift Tanpa Server?	Dengan klaster yang disediakan, Anda menentukan jendela pemeliharaan saat penambalan terjadi. (Biasanya, Anda memilih waktu berulang saat penggunaan rendah.)

Fitur	Deskripsi	Nirserver	Disediakan
Enkripsi	Anda dapat mengaktifkan enkripsi basis data.	Amazon Redshift Serverless selalu dienkripsi dengan, dengan kunci terkelola atau AWS KMS dikelola pelanggan. AWS	Data dalam gudang data yang disediakan dapat dienkripsi dengan AWS KMS (dengan kunci yang AWS dikelola atau dikelola pelanggan), atau tidak dienkripsi. Lihat Enkripsi basis data Amazon Redshift .
Penagihan penyimpanan	Cara kerja penagihan untuk penyimpanan.	Untuk Amazon Redshift Tanpa Server. Tarif dihitung sesuai dengan GB per bulan. Lihat Penagihan untuk kapasitas komputasi .	Penyimpanan ditagih terpisah dari sumber daya komputasi untuk kluster yang disediakan dengan node. RA3

Fitur	Deskripsi	Nirserver	Disediakan
Manajemen pengguna	Bagaimana pengguna dikelola.	Untuk Amazon Redshift Serverless, pengguna adalah pengguna IAM atau Redshift. Untuk informasi selengkapnya, lihat Manajemen identitas dan akses di Amazon Redshift Tanpa Server. Untuk informasi selengkapnya tentang mengelola identitas IAM, termasuk praktik terbaik untuk peran IAM, lihat.	Untuk gudang data yang disediakan, pengguna adalah pengguna IAM atau Redshift. Untuk informasi selengkapnya, lihat Mengelola keamanan database di Panduan Pengembang Database Amazon Redshift. Untuk informasi selengkapnya tentang mengelola identitas IAM, termasuk praktik terbaik untuk peran IAM, lihat. Manajemen identitas dan akses di Amazon Redshift

Fitur	Deskripsi	Nirserver	Disediakan
		Manajemen identitas dan akses di Amazon Redshift	

Fitur	Deskripsi	Nirserver	Disediakan
Alat dan kompatibilitas JDBC dan ODBC	Bagaimana koneksi klien bekerja.	Amazon Redshift Serverless kompatibel dengan alat atau aplikasi klien yang sesuai dengan JDBC atau ODBC. Untuk informasi selengkapnya tentang driver, lihat Mengonfigurasi koneksi di Panduan Manajemen Amazon Redshift. Untuk informasi tentang menghubungkan ke cluster, lihat Menyambung ke gudang data Amazon Redshift menggunakan alat klien SQL .	

Fitur	Deskripsi	Nirserver	Disediakan
		lihat Menyambun g ke Redshift Tanpa Server.	
Persyarat an untuk kredensio nal saat masuk	Bagaimana kredensil ditangani.	Untuk Amazon Redshift Tanpa Server, Anda tidak perlu memasukka n kredensil di setiap instans. Untuk informasi selengkap nya, lihat Menghubun gkan ke Amazon Redshift Tanpa Server.	Akses ke Amazon Redshift memerlukan kredenal masuk dari pengguna yang terkait dengan peran IAM. Peran IAM memiliki izin khusus yang dilampirkan untuk gudang data yang disediakan. Setelah diautentikasi, pengguna dapat terhubung langsung ke database, ke konsol Redshift, dan ke editor kueri v2.

Fitur	Deskripsi	Nirserver	Disediakan
Data API	Anda dapat mengakses data dari layanan web dan aplikasi lainnya.	Amazon Redshift Serverless mendukung Amazon Redshift Data API. Dengan Amazon Redshift Serverless, Anda menggunakan <code>workgroup-name</code> parameter alih-alih <code>parameter.cluster-identity</code> Untuk informasi selengkapnya tentang memanggil Data API, lihat Menggunakan API Data Amazon Redshift .	Amazon Redshift yang disediakan mendukung Amazon Redshift Data API. Dengan cluster Amazon Redshift, Anda menggunakan parameter <code>cluster-identity</code> parameter <code>workgroup-name</code> Untuk informasi selengkapnya tentang memanggil Data API, lihat Menggunakan API Data Amazon Redshift .

Fitur	Deskripsi	Nirserver	Disediakan
Cuplikan	Memberikan point-in-time pemulihan.	Amazon Redshift Serverless mendukung snapshot dan titik pemulihan . Untuk informasi selengkapnya tentang snapshot dan titik pemulihan untuk namespace , lihat. Snapshot dan titik pemulihan	Cluster yang disediakan mendukung snapshot. Untuk informasi selengkapnya, lihat Mengelola snapshot menggunakan konsol .

Fitur	Deskripsi	Nirserver	Disediakan
Berbagi Data	Memberikan kemampuan untuk berbagi data antar database di akun yang sama atau di akun yang berbeda.	Amazon Redshift Serverless mendukung semua fitur berbagi data yang dilakukan oleh gudang data yang disediakan. Ini juga mendukung berbagi data antara Amazon Redshift Serverless dan gudang data, alat, atau aplikasi klien yang disediakan.	Cluster yang disediakan mendukung lintas basis data, lintas akun, Lintas wilayah, dan berbagi data. AWS Data Exchange Untuk informasi selengkapnya, lihat Berbagi data di seluruh cluster di Amazon Redshift .

Fitur	Deskripsi	Nirserver	Disediakan
Trek	Menyediakan jadwal untuk pembaruan perangkat lunak.	Amazon Redshift Serverless tidak memiliki konsep trek. Versi dan pembaruan ditangani oleh layanan. Untuk informasi selengkapnya tentang desain Amazon Redshift Serverless, lihat. Snapshot dan titik pemulihan	Cluster yang disediakan mendukung peralihan antara trek saat ini dan trailing.

Fitur	Deskripsi	Nirserver	Disediakan
Tabel dan tampilan sistem	Menyediakan cara untuk memantau sumber daya dan metadata sistem Anda.	Amazon Redshift Serverless mendukung tabel dan tampilan sistem baru. Untuk informasi selengkapnya tentang tabel sistem, lihat Memantau kueri dan beban kerja dengan Amazon Redshift Serverless . Untuk informasi tentang cara memigrasikan kueri Anda dari menggunakan tabel dan tampilan	Gudang data yang disediakan mendukung kumpulan tabel dan tampilan sistem yang ada untuk pemantauan dan tugas lain yang memerlukan metadata sistem.

Fitur	Deskripsi	Nirserver	Disediakan
		sistem yang disediakan lebih lama ke tampilan baru, lihat Memigrasi ke tampilan pemantauan SYS.	

Fitur	Deskripsi	Nirserver	Disediakan
Grup parameter	Ini adalah sekelompok parameter yang berlaku untuk semua database yang dibuat dalam sebuah cluster. Parameter ini mengkonfigurasi pengaturan database seperti batas waktu kueri dan gaya tanggal.	Amazon Redshift Serverless tidak memiliki konsep grup parameter.	Gudang data yang disediakan mendukung kelompok parameter. Untuk informasi selengkapnya tentang grup parameter untuk klaster yang disediakan, lihat Grup parameter Amazon Redshift

Fitur	Deskripsi	Nirserver	Disediakan
Pemantauan kueri	Menyediakan tampilan berbasis waktu dari kueri yang dijalankan.	Pemantauan kueri di Amazon Redshift Tanpa Server mengharuskan pengguna untuk terhubung ke database untuk menggunakan tabel sistem. Dengan demikian, pemantauan kueri dan tabel sistem sinkron. Kueri tabel sistem di Amazon Redshift Tanpa Server menggunakan pengguna database	Pemantauan kueri di kluster yang disediakan tidak menampilkan semua data dalam tabel sistem.

Fitur	Deskripsi	Nirserver	Disediakan
		yang dipetakan ke pengguna IAM untuk menggunakan pemantauan kueri. Untuk informasi selengkapnya tentang pemantauan kueri, lihat Memantau kueri dan beban kerja dengan Amazon Redshift Tanpa Server.	

Fitur	Deskripsi	Nirserver	Disediakan
Pencatatan audit	Memberikan informasi tentang koneksi dan aktivitas pengguna dalam database.	Dengan Amazon Redshift Serverless, CloudWatch adalah tujuan untuk log audit. Pengiriman log audit berbasis Amazon S3 tidak didukung untuk Amazon Redshift Tanpa Server. Untuk informasi selengkapnya, lihat Pencatatan audit untuk Amazon Redshift Tanpa Server .	Untuk kluster yang disediakan, pengiriman log audit berbasis Amazon S3 telah menjadi norma. Sekarang, pengiriman log audit ke CloudWatch diperluas untuk mencakup gudang data yang disediakan.

Fitur	Deskripsi	Nirserver	Disediakan
Pemberitahuan acara	Amazon EventBridge adalah layanan bus acara tanpa server yang dapat Anda gunakan untuk menghubungkan aplikasi Anda dengan data peristiwa dari berbagai sumber.	Amazon Redshift Serverless menggunakan EventBridge Amazon untuk mengelola pemberitahuan acara agar up-to-date. Anda tetap memperhatikan perubahan di gudang data Anda. Untuk informasi selengkapnya, lihat Pemberitahuan acara Amazon Redshift Tanpa Server dengan Amazon EventBridge .	Untuk kluster yang disediakan, Anda mengelola notifikasi peristiwa menggunakan konsol Amazon Redshift untuk membuat langganan acara. Untuk informasi selengkapnya, lihat Membuat langganan pemberitahuan acara .

Fitur	Deskripsi	Serverless	Disediakan
Kendala kursor	Amazon Redshift memberlakukan batasan pada ukuran semua set hasil kursor.	Amazon Redshift Serverless memiliki kursor maksimum total hasil set ukuran 150.000 MB.	Untuk kluster yang disediakan, ukuran set hasil total maksimum kursor bergantung pada jenis cluster. Untuk informasi selengkapnya, lihat Kendala kursor .

Menggunakan antarmuka manajemen Amazon Redshift untuk kluster yang disediakan

Note

Topik ini berfokus pada antarmuka manajemen Amazon Redshift untuk cluster yang disediakan. Ada antarmuka manajemen serupa untuk Amazon Redshift Serverless dan Amazon Redshift Data API.

Amazon Redshift mendukung beberapa antarmuka manajemen yang dapat Anda gunakan untuk membuat, mengelola, dan menghapus kluster Amazon Redshift: the AWS SDKs, the AWS Command Line Interface (AWS CLI), AWS CLI dan API manajemen Amazon Redshift.

Amazon Redshift API — Anda dapat memanggil API manajemen Amazon Redshift ini dengan mengirimkan permintaan. Permintaan adalah permintaan HTTP atau HTTPS yang menggunakan kata kerja HTTP GET atau POST dengan parameter bernama `Action`. Memanggil Amazon Redshift API adalah cara paling langsung untuk mengakses layanan Amazon Redshift. Namun, aplikasi Anda harus menangani detail tingkat rendah seperti penanganan kesalahan dan menghasilkan hash untuk menandatangani permintaan.

- Untuk informasi tentang membuat dan menandatangani permintaan Amazon Redshift API, lihat [Menandatangani permintaan HTTP](#)

- Untuk informasi tentang tindakan Amazon Redshift API dan tipe data untuk Amazon Redshift, lihat referensi Amazon [Redshift](#) API.

AWS SDKs— Anda dapat menggunakan AWS SDKs untuk melakukan operasi terkait kluster Amazon Redshift. Beberapa pustaka SDK membungkus API Amazon Redshift yang mendasarinya. Mereka mengintegrasikan fungsionalitas API ke dalam bahasa pemrograman tertentu dan menangani banyak detail tingkat rendah, seperti menghitung tanda tangan, menangani percobaan ulang permintaan, dan penanganan kesalahan. Memanggil fungsi pembungkus di pustaka SDK dapat sangat menyederhanakan proses penulisan aplikasi untuk mengelola cluster Amazon Redshift.

- Amazon Redshift didukung oleh AWS SDKs untuk Java, .NET, PHP, Python, Ruby, dan Node.js. Fungsi pembungkus untuk Amazon Redshift didokumentasikan dalam manual referensi untuk setiap SDK. Untuk daftar AWS SDKs dan tautan ke dokumentasinya, lihat [Alat untuk Amazon Web Services](#).
- Panduan ini memberikan contoh bekerja dengan Amazon Redshift menggunakan Java SDK. Untuk contoh kode AWS SDK yang lebih umum, lihat [Contoh kode untuk Amazon Redshift menggunakan AWS SDKs](#).

AWS CLICLI menyediakan seperangkat alat baris perintah yang dapat Anda gunakan untuk mengelola AWS layanan dari komputer Windows, Mac, dan Linux. AWS CLI Termasuk perintah berdasarkan tindakan Amazon Redshift API.

- Untuk informasi tentang menginstal dan menyiapkan Amazon Redshift CLI, lihat. [Menyiapkan Amazon Redshift CLI](#)
- [Untuk materi referensi tentang perintah Amazon Redshift CLI, lihat Amazon Redshift di Referensi.AWS CLI](#)

Menggunakan layanan ini dengan AWS SDK

AWS kit pengembangan perangkat lunak (SDKs) tersedia untuk banyak bahasa pemrograman populer. Setiap SDK menyediakan API, contoh kode, dan dokumentasi yang memudahkan developer untuk membangun aplikasi dalam bahasa pilihan mereka.

Dokumentasi SDK	Contoh kode
AWS SDK for C++	AWS SDK for C++ contoh kode

Dokumentasi SDK	Contoh kode
AWS CLI	AWS CLI contoh kode
AWS SDK for Go	AWS SDK for Go contoh kode
AWS SDK for Java	AWS SDK for Java contoh kode
AWS SDK for JavaScript	AWS SDK for JavaScript contoh kode
AWS SDK for Kotlin	AWS SDK for Kotlin contoh kode
AWS SDK for .NET	AWS SDK for .NET contoh kode
AWS SDK for PHP	AWS SDK for PHP contoh kode
AWS Tools for PowerShell	AWS Tools for PowerShell contoh kode
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) contoh kode
AWS SDK for Ruby	AWS SDK for Ruby contoh kode
AWS SDK for Rust	AWS SDK for Rust contoh kode
AWS SDK for SAP ABAP	AWS SDK for SAP ABAP contoh kode
AWS SDK for Swift	AWS SDK for Swift contoh kode

Ketersediaan contoh

Tidak dapat menemukan apa yang Anda butuhkan? Minta contoh kode menggunakan tautan Berikan umpan balik di bagian bawah halaman ini.

Menandatangani permintaan HTTP

Amazon Redshift mengharuskan setiap permintaan yang Anda kirim ke API manajemen diautentikasi dengan tanda tangan. Topik ini menjelaskan cara menandatangani permintaan Anda.

Jika Anda menggunakan salah satu Kit Pengembangan Perangkat AWS Lunak (SDKs) atau AWS Command Line Interface, penandatanganan permintaan ditangani secara otomatis, dan Anda dapat melewati bagian ini. Untuk informasi selengkapnya tentang penggunaan AWS SDKs, lihat [Menggunakan antarmuka manajemen Amazon Redshift untuk kluster yang disediakan](#). Untuk informasi selengkapnya tentang penggunaan Antarmuka Baris Perintah Amazon Redshift, buka referensi baris perintah [Amazon Redshift](#).

Untuk menandatangani permintaan, Anda menghitung tanda tangan digital dengan menggunakan fungsi hash kriptografi. Hash kriptografi adalah fungsi yang mengembalikan nilai hash unik yang didasarkan pada input. Input ke fungsi hash mencakup teks permintaan Anda dan kunci akses rahasia Anda yang bisa Anda dapatkan dari kredensi sementara. Fungsi hash mengembalikan nilai hash yang Anda sertakan dalam permintaan sebagai tanda tangan Anda. Tanda tangan adalah bagian header Authorization dari permintaan Anda.

Note

Pengguna membutuhkan akses terprogram jika mereka ingin berinteraksi dengan AWS luar. Konsol Manajemen AWS Cara untuk memberikan akses terprogram tergantung pada jenis pengguna yang mengakses AWS.

Untuk memberi pengguna akses programatis, pilih salah satu opsi berikut.

Pengguna mana yang membutuhkan akses programatis?	Untuk	Oleh
IAM	(Disarankan) Gunakan kredensial konsol sebagai kredensial sementara untuk menandatangani permintaan terprogram ke,, atau. AWS CLI AWS SDKs AWS APIs	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan. • Untuk itu AWS CLI, lihat Login untuk pengembangan AWS lokal di Panduan AWS Command Line Interface Pengguna. • Untuk AWS SDKs, lihat Login untuk pengembangan AWS lokal di

Pengguna mana yang membutuhkan akses programatis?	Untuk	Oleh
		Panduan Referensi Alat AWS SDKs dan Alat.
Identitas tenaga kerja (Pengguna yang dikelola di Pusat Identitas IAM)	Gunakan kredensi sementara untuk menandatangani permintaan terprogram keAWS CLI,, AWS SDKs atau. AWS APIs	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan. • UntukAWS CLI, lihat Mengkonfigurasi yang akan AWS CLI digunakan AWS IAM Identity Center dalam Panduan AWS Command Line Interface Pengguna. • Untuk AWSSDKs, alat, dan AWSAPIs, lihat Autentikasi Pusat Identitas IAM di Panduan Referensi Alat AWS SDKs dan Alat.
IAM	Gunakan kredensi sementara untuk menandatangani permintaan terprogram keAWS CLI,, AWS SDKs atau. AWS APIs	Mengikuti petunjuk dalam Menggunakan kredensi sementara dengan AWS sumber daya di Panduan Pengguna IAM.

Pengguna mana yang membutuhkan akses programatis?	Untuk	Oleh
IAM	(Tidak direkomendasikan) Gunakan kredensi jangka panjang untuk menandatangani permintaan terprogram ke AWS CLI, AWS SDKs atau AWS APIs	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan. • Untuk mengetahui AWS CLI, lihat Mengautentikasi menggunakan kredensi pengguna IAM di Panduan Pengguna AWS Command Line Interface • Untuk AWS SDKs dan alat, lihat Mengautentikasi menggunakan kredensi jangka panjang di Panduan Referensi Alat AWS SDKs dan Alat. • Untuk AWS APIs, lihat Mengelola kunci akses untuk pengguna IAM di Panduan Pengguna IAM.

Setelah Amazon Redshift menerima permintaan Anda, itu menghitung ulang tanda tangan dengan menggunakan fungsi hash yang sama dan input yang Anda gunakan untuk menandatangani permintaan. Jika tanda tangan yang dihasilkan cocok dengan tanda tangan dalam permintaan, Amazon Redshift memproses permintaan; jika tidak, permintaan akan ditolak.

Amazon Redshift mendukung otentikasi menggunakan versi [AWS tanda tangan](#) 4. Proses untuk menghitung tanda tangan terdiri dari tiga tugas. Tugas-tugas ini diilustrasikan dalam contoh berikut.

- [Tugas 1: Buat permintaan kanonik](#)

Atur ulang permintaan HTTP Anda ke dalam bentuk kanonik. Menggunakan bentuk kanonik diperlukan karena Amazon Redshift menggunakan bentuk kanonik yang sama untuk menghitung tanda tangan yang dibandingkan dengan yang Anda kirim.

- [Tugas 2: Buat string untuk ditandatangani](#)

Buat string yang akan Anda gunakan sebagai salah satu nilai input untuk fungsi hash kriptografi Anda. String, yang disebut string to sign, adalah rangkaian dari nama algoritme hash, tanggal permintaan, string cakupan kredensial, dan permintaan kanonikalisasi dari tugas sebelumnya. String lingkup kredensi itu sendiri adalah rangkaian informasi tanggal, wilayah, dan layanan.

- [Tugas 3: Hitung tanda tangan](#)

Hitung tanda tangan untuk permintaan Anda dengan menggunakan fungsi hash kriptografi yang menerima dua string input, string Anda untuk ditandatangani dan kunci turunan. Kunci turunan dihitung dengan memulai dengan kunci akses rahasia Anda dan menggunakan string cakupan kredensi untuk membuat serangkaian kode otentikasi pesan berbasis hash (HMAC-). SHA256

Contoh perhitungan tanda tangan

Contoh berikut memandu Anda melalui detail pembuatan tanda tangan untuk [CreateCluster](#) permintaan. Anda dapat menggunakan contoh ini sebagai referensi untuk memeriksa metode perhitungan tanda tangan Anda sendiri. Perhitungan referensi lainnya disertakan dalam [bagian Minta contoh tanda tangan](#) pada Panduan Pengguna IAM.

Anda dapat menggunakan permintaan GET atau POST untuk mengirim permintaan ke Amazon Redshift. Perbedaan antara keduanya adalah bahwa untuk permintaan GET parameter Anda dikirim sebagai parameter string kueri. Untuk permintaan POST mereka termasuk dalam badan permintaan. Contoh di bawah ini menunjukkan permintaan POST.

Contoh tersebut mengasumsikan sebagai berikut:

- Stempel waktu permintaan adalah `Fri, 07 Dec 2012 00:00:00 GMT`.
- Titik akhirnya adalah Wilayah AS Timur (Virginia Utara), `us-east-1`.

Sintaks permintaan umum adalah:

```
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
```

Bentuk kanonik permintaan yang dihitung untuk [Tugas 1: Membuat Permintaan Kanonik](#) adalah:

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
```

Baris terakhir dari permintaan kanonik adalah hash dari isi permintaan. Baris ketiga dalam permintaan kanonik kosong karena tidak ada parameter kueri untuk API ini.

String yang akan ditandatangani untuk [Tugas 2: Buat String untuk Ditandatangani](#) adalah:

```
AWS4-HMAC-SHA256
20121207T000000Z
20121207/us-east-1/redshift/aws4_request
06b6bef4f4f060a5558b60c627cc6c5b5b5a959b9902b5ac2187be80cbac0714
```

Baris pertama string to sign adalah algoritme, baris kedua adalah stempel waktu, baris ketiga adalah cakupan kredensial, dan baris terakhir adalah hash permintaan kanonik dari [Tugas 1: Membuat Permintaan Kanonik](#). Nama layanan untuk digunakan dalam cakupan kredensial adalah redshift.

Untuk [Tugas 3: Hitung Tanda Tangan](#), kunci turunan dapat direpresentasikan sebagai:

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

Kunci turunan dihitung sebagai rangkaian fungsi hash. Mulai dari pernyataan HMAC dalam rumus di atas, Anda menggabungkan frasa **AWS4** dengan kunci akses rahasia Anda dan menggunakan ini sebagai kunci untuk hash data "us-east-1". Hasil hash ini menjadi kunci untuk fungsi hash berikutnya.

Setelah Anda menghitung kunci turunan, Anda menggunakannya dalam fungsi hash yang menerima dua string input, string Anda untuk ditandatangani dan kunci turunan. Misalnya, jika Anda menggunakan kunci akses rahasia wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY dan string untuk menandatangani yang diberikan sebelumnya, maka tanda tangan yang dihitung adalah sebagai berikut:

```
9a6b557aa9f38dea83d9215d8f0eae54100877f3e0735d38498d7ae489117920
```

Langkah terakhir adalah membangun header Authorization. Untuk access key demonstrasi AKIAIOSFODNN7EXAMPLE, header (dengan jeda baris yang ditambahkan untuk keterbacaan) adalah:

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

Menyiapkan Amazon Redshift CLI

Bagian ini menjelaskan cara mengatur dan menjalankan alat baris AWS CLI perintah untuk digunakan dalam mengelola Amazon Redshift. [Alat baris perintah Amazon Redshift berjalan di AWS Command Line Interface \(AWS CLI\), yang pada gilirannya menggunakan Python \(thon.org/\). https://www.py](#) Ini AWS CLI dapat dijalankan pada sistem operasi apa pun yang mendukung Python.

Instalasi AWS Command Line Interface

Untuk mulai menggunakan alat baris perintah Amazon Redshift, Anda terlebih dahulu mengatur AWS CLI, dan kemudian Anda menambahkan file konfigurasi yang menentukan opsi Amazon Redshift CLI.

Jika Anda telah menginstal dan mengkonfigurasi AWS CLI untuk AWS layanan lain, Anda dapat melewati prosedur ini.

Untuk menginstal AWS Command Line Interface

1. Pergi ke [Instal atau perbarui ke versi terbaru dari AWS CLI](#), dan kemudian ikuti petunjuk untuk menginstal AWS CLI.

Untuk akses CLI, Anda memerlukan ID kunci akses dan kunci akses rahasia. Gunakan kredensial sementara alih-alih kunci akses jangka panjang jika memungkinkan. Kredensial sementara mencakup ID kunci akses, kunci akses rahasia, dan token keamanan yang menunjukkan kapan kredensialnya kedaluwarsa. Untuk informasi selengkapnya, lihat [Menggunakan kredensial sementara dengan AWS sumber daya](#) di Panduan Pengguna IAM.

2. Buat file yang berisi informasi konfigurasi seperti kunci akses, wilayah default, dan format output perintah. Kemudian atur variabel `AWS_CONFIG_FILE` lingkungan untuk referensi file itu. Untuk instruksi terperinci, buka [Mengkonfigurasi antarmuka baris AWS perintah](#) di Panduan AWS Command Line Interface Pengguna.
3. Jalankan perintah uji untuk mengonfirmasi bahwa AWS CLI antarmuka berfungsi. Misalnya, perintah berikut harus menampilkan informasi bantuan untuk AWS CLI:

```
aws help
```

Perintah berikut akan menampilkan informasi bantuan untuk Amazon Redshift:

```
aws redshift help
```

[Untuk materi referensi tentang perintah Amazon Redshift CLI, buka Amazon Redshift di Referensi.](#)
AWS CLI

Amazon Redshift Tanpa Server

Amazon Redshift Tanpa Server memudahkan Anda menjalankan dan menskalakan analitik tanpa harus menyediakan dan mengelola gudang data lokal. Dengan Amazon Redshift Serverless, analis data, pengembang, dan ilmuwan data sekarang dapat menggunakan Amazon Redshift untuk mendapatkan wawasan dari data dalam hitungan detik dengan memuat data ke dalam dan menanyakan catatan dari gudang data di cloud. Amazon Redshift secara otomatis menyediakan dan menskalakan kapasitas gudang data untuk menghadirkan kinerja yang cepat untuk beban kerja yang berat dan tidak dapat diprediksi. Anda hanya membayar untuk kapasitas yang Anda gunakan. Anda bisa mendapatkan keuntungan dari kesederhanaan ini tanpa mengubah analitik dan aplikasi intelijen bisnis yang ada.

[Untuk informasi tentang Amazon Redshift Tanpa Server, lihat Perjanjian Tingkat SLAs Layanan Amazon Redshift.](#)

Apa itu Amazon Redshift Tanpa Server?

Amazon Redshift Serverless secara otomatis menyediakan kapasitas gudang data dan secara cerdas menskalakan sumber daya yang mendasarinya. Amazon Redshift Serverless menyesuaikan kapasitas dalam hitungan detik untuk menghadirkan kinerja tinggi secara konsisten dan operasi yang disederhanakan bahkan untuk beban kerja yang paling menuntut dan mudah berubah sekalipun.

Dengan Amazon Redshift Serverless, Anda bisa mendapatkan keuntungan dari fitur-fitur berikut:

- Akses dan analisis data tanpa perlu menyiapkan, menyetel, dan mengelola klaster yang disediakan Amazon Redshift.
- Gunakan kapabilitas Amazon Redshift SQL yang unggul, kinerja terdepan di industri, dan integrasi data-lake untuk melakukan kueri secara mulus di seluruh gudang data, data lake, dan sumber data operasional.
- Memberikan kinerja tinggi secara konsisten dan operasi yang disederhanakan untuk beban kerja yang paling menuntut dan mudah berubah dengan penskalaan cerdas dan otomatis.
- Gunakan workgroup dan namespace untuk mengatur sumber daya komputasi dan data dengan kontrol biaya granular.
- Bayar hanya ketika gudang data sedang digunakan.

Dengan Amazon Redshift Serverless, Anda menggunakan antarmuka konsol untuk menjangkau gudang data tanpa server atau untuk membangun aplikasi. APIs Melalui gudang data, Anda dapat mengakses penyimpanan terkelola Amazon Redshift dan data lake Amazon S3 Anda.

Video ini menunjukkan kepada Anda bagaimana Amazon Redshift Serverless memudahkan menjalankan dan menskalakan analitik tanpa harus mengelola infrastruktur gudang data:

Konsol Tanpa Server Amazon Redshift

Untuk memulai menggunakan konsol Amazon Redshift Serverless, tonton video berikut: [Memulai Amazon Redshift](#) Tanpa Server.

Dasbor tanpa server

Pada halaman dasbor Tanpa Server, Anda dapat melihat ringkasan sumber daya dan grafik penggunaan Anda.

- Gambaran umum Namespace - Bagian ini menunjukkan jumlah snapshot dan datashares dalam namespace Anda.
- Workgroups - Bagian ini menampilkan semua workgroup dalam Amazon Redshift Serverless.
- Metrik kueri — Bagian ini menunjukkan aktivitas kueri selama satu jam terakhir.
- Kapasitas RPU yang digunakan - Bagian ini menunjukkan kapasitas yang digunakan selama satu jam terakhir.
- Uji coba gratis - Bagian ini menunjukkan kredit uji coba gratis yang tersisa di AWS akun Anda. Ini mencakup semua penggunaan sumber daya dan operasi Amazon Redshift Tanpa Server, termasuk snapshot, penyimpanan, workgroup, dan sebagainya, di bawah akun yang sama.
- Alarm - Bagian ini menunjukkan alarm yang Anda konfigurasi di Amazon Redshift Tanpa Server.

Pencadangan data

Pada tab Pencadangan data Anda dapat bekerja dengan yang berikut ini:

- Snapshots - Anda dapat membuat, menghapus, dan mengelola snapshot data Amazon Redshift Tanpa Server Anda. Periode retensi default adalah indefinitely, tetapi Anda dapat mengonfigurasi periode retensi menjadi nilai apa pun antara 1 dan 3653 hari. Anda dapat mengotorisasi Akun AWS untuk memulihkan ruang nama dari snapshot.

- Poin pemulihan - Menampilkan titik pemulihan yang dibuat secara otomatis sehingga Anda dapat pulih dari penulisan atau penghapusan yang tidak disengaja dalam 24 jam terakhir. Untuk memulihkan data, Anda dapat memulihkan titik pemulihan ke namespace yang tersedia. Anda dapat membuat snapshot dari titik pemulihan jika Anda ingin mempertahankan titik pemulihan untuk jangka waktu yang lebih lama. Periode retensi default adalah *indefinitely*, tetapi Anda dapat mengonfigurasi periode retensi menjadi nilai apa pun antara 1 dan 3653 hari.

Akses data

Pada tab Akses data Anda dapat bekerja dengan yang berikut:

- Pengaturan jaringan dan keamanan - Anda dapat melihat nilai terkait VPC, nilai AWS KMS enkripsi, dan nilai pencatatan audit. Anda hanya dapat memperbarui pencatatan audit.
- AWS KMS key— AWS KMS key Digunakan untuk mengenkripsi sumber daya di Amazon Redshift Serverless.
- Izin — Anda dapat mengelola peran IAM yang dapat diasumsikan oleh Amazon Redshift Serverless untuk menggunakan sumber daya atas nama Anda. Untuk informasi selengkapnya, lihat [Manajemen identitas dan akses di Amazon Redshift Tanpa Server](#).
- Titik akhir VPC yang dikelola Redshift — Anda dapat mengakses instans Amazon Redshift Tanpa Server dari VPC atau subnet lain. Untuk informasi selengkapnya, lihat [Menghubungkan ke Amazon Redshift Tanpa Server dari titik akhir VPC lainnya](#).

Batas

Pada tab Limits, Anda dapat bekerja dengan yang berikut:

- Kapasitas dasar dalam pengaturan unit pemrosesan Redshift (RPU) - Anda dapat mengatur kapasitas dasar yang digunakan untuk memproses beban kerja Anda. Untuk meningkatkan kinerja kueri, tingkatkan nilai RPU Anda.
- Batas penggunaan — Sumber daya komputasi maksimum yang dapat digunakan instans Amazon Redshift Tanpa Server dalam periode waktu sebelum tindakan dimulai. Anda membatasi jumlah sumber daya yang digunakan Amazon Redshift Serverless untuk menjalankan beban kerja Anda. Penggunaan diukur dalam jam Redshift Processing Unit (RPU). Jam RPU adalah jumlah yang RPU digunakan dalam satu jam. Anda menentukan tindakan yang akan terjadi ketika Anda mencapai batas yang Anda tetapkan, sebagai berikut:
 - Kirim peringatan.

- Log entri ke tabel sistem.
- Matikan kueri pengguna.

Anda dapat mengatur hingga empat batas.

- Batas kueri — Anda dapat menambahkan batas untuk memantau kinerja dan batas. Untuk informasi selengkapnya tentang batas pemantauan kueri, lihat aturan [pemantauan kueri WLM](#).

Untuk informasi selengkapnya, lihat [Kapasitas komputasi untuk Amazon Redshift Serverless](#).

Datashares

Pada tab Datashares Anda dapat bekerja dengan yang berikut:

- Datashares dibuat di pengaturan namespace saya — Anda dapat membuat datashare dan membagikannya dengan namespace lain dan. Akun AWS
- Datashares dari namespace lain dan Akun AWS — Anda dapat membuat database dari datashare dari namespace lain dan. Akun AWS

Untuk informasi selengkapnya tentang berbagi data, lihat [Berbagi data di Amazon Redshift Tanpa Server](#).

Pemantauan kueri dan basis data

Pada halaman Query dan pemantauan database, Anda dapat melihat grafik riwayat Query dan kinerja Database Anda.

Pada tab Query history, Anda melihat grafik berikut (Anda dapat memilih antara daftar Query dan Resource metrics):

- Query runtime — Grafik ini menunjukkan kueri mana yang berjalan dalam jangka waktu yang sama. Pilih bilah dalam grafik untuk melihat detail eksekusi kueri lainnya.
- Kueri dan beban - Bagian ini mencantumkan kueri dan beban berdasarkan Query ID.
- Kapasitas RPU yang digunakan - Grafik ini menunjukkan kapasitas keseluruhan dalam Unit RPUs Pemrosesan Redshift ().
- Koneksi database - Grafik ini menunjukkan jumlah koneksi database aktif.

Kinerja basis data

Pada tab Kinerja Database, Anda melihat grafik berikut:

- Kueri selesai per detik — Grafik ini menunjukkan jumlah rata-rata kueri yang diselesaikan per detik.
- Durasi kueri — Grafik ini menunjukkan jumlah waktu rata-rata untuk menyelesaikan kueri.
- Koneksi database - Grafik ini menunjukkan jumlah koneksi database aktif.
- Menjalankan kueri — Grafik ini menunjukkan jumlah total kueri yang berjalan pada waktu tertentu.
- Kueri antrian — Grafik ini menunjukkan jumlah total kueri yang diantrian pada waktu tertentu.
- Perincian waktu proses kueri — Grafik ini menunjukkan total waktu kueri yang dihabiskan untuk dijalankan berdasarkan jenis kueri.

Pemantauan sumber daya

Pada halaman Pemantauan sumber daya, Anda dapat melihat grafik sumber daya yang Anda konsumsi. Anda dapat memfilter data berdasarkan beberapa aspek.

- Filter metrik - Anda dapat menggunakan filter metrik untuk memilih filter untuk grup kerja tertentu, serta memilih rentang waktu dan interval waktu.
- Kapasitas RPU yang digunakan — Grafik ini menunjukkan kapasitas keseluruhan dalam unit RPU pemrosesan Redshift ().
- Penggunaan komputasi - Grafik ini menunjukkan penggunaan jam RPU berdasarkan periode untuk rentang waktu yang dipilih. Untuk rentang waktu kurang dari 6 jam, jam RPU ditampilkan dalam waktu yang tepat. Untuk rentang waktu 6 jam atau lebih, jam RPU ditampilkan sebagai rata-rata.
- Komputasi ekstra untuk pengoptimalan otomatis detik yang diisi daya — Grafik ini menunjukkan jumlah detik RPU yang dibebankan untuk pengoptimalan basis data otomatis untuk rentang waktu yang dipilih. Anda dikenakan biaya untuk pengoptimalan otomatis saat Amazon Redshift menggunakan sumber daya komputasi tambahan untuk menjalankannya. Untuk informasi selengkapnya, lihat [Mengalokasikan sumber daya komputasi tambahan untuk pengoptimalan basis data otomatis](#).

Datashares

Di halaman Datashares, Anda dapat mengelola datashares Di akun saya dan Dari akun lain. Untuk informasi selengkapnya tentang berbagi data, lihat [Berbagi data di Amazon Redshift Tanpa Server](#).

AWS Glue Data Catalog

Di AWS Glue Data Catalog, Anda dapat melihat status pendaftaran namespace Anda ke file. AWS Glue Data Catalog Tab ini hanya muncul setelah Anda memulai proses pendaftaran. Untuk informasi selengkapnya tentang mendaftarkan ruang nama ke AWS Glue Data Catalog, lihat kompatibilitas [Apache Iceberg untuk Amazon Redshift di Panduan Pengembang Database Amazon Redshift](#).

Pertimbangan saat menggunakan Amazon Redshift Serverless

Untuk daftar Wilayah AWS tempat Amazon Redshift Serverless tersedia, lihat titik akhir yang terdaftar untuk [Redshift](#) Serverless API di. Referensi Umum Amazon Web Services

Beberapa sumber daya yang digunakan oleh Amazon Redshift Serverless tunduk pada kuota. Untuk informasi selengkapnya, lihat [Kuota untuk objek Amazon Redshift Tanpa Server](#).

[Saat Anda MENDEKLARASIKAN kursor, spesifikasi ukuran set hasil untuk Amazon Redshift Serverless ditentukan dalam DECLARE](#). Amazon Redshift Serverless memiliki kursor maksimum total hasil set ukuran 150.000 MB.

Penambalan Online — Amazon Redshift Serverless menawarkan pembaruan perangkat lunak otomatis tanpa memerlukan jendela pemeliharaan tradisional. Ketika pembaruan baru tersedia, sistem menerapkannya dalam waktu 14 hari setelah rilis selama periode idle. Proses pembaruan biasanya memakan waktu hingga 15 menit. Jika tidak ada periode idle 15 menit yang terjadi dalam 14 hari, titik akhir Tanpa Server Anda mungkin mengalami ketidaktersediaan singkat. Selama waktu ini, koneksi aplikasi ke titik akhir mungkin gagal. Anda dapat memantau rilis patch Redshift di dokumentasi “Versi cluster untuk Amazon Redshift”. [Untuk informasi tentang Amazon Redshift Tanpa Server, lihat Perjanjian Tingkat SLAs Layanan Amazon Redshift](#).

Lacak - Saat Amazon Redshift merilis versi workgroup baru, workgroup Anda diperbarui secara otomatis. Anda dapat mengontrol apakah workgroup Anda diperbarui ke rilis terbaru atau ke rilis sebelumnya. Untuk informasi tentang trek, lihat [Trek untuk klaster yang disediakan Amazon Redshift dan grup kerja tanpa server](#).

Availability Zone IDs — Saat mengonfigurasi instans Amazon Redshift Tanpa Server, buka Pertimbangan tambahan, dan pastikan subnet yang IDs disediakan di Subnet berisi setidaknya dua Availability Zone yang didukung. IDs

- Untuk workgroup tanpa Enhanced VPC Routing (EVR), Anda memerlukan dua Availability Zones (). AZs
- Untuk kelompok kerja dengan EVR, Anda membutuhkan tiga. AZs

Untuk melihat subnet ke pemetaan ID Availability Zone, buka konsol VPC dan pilih Subnet untuk melihat daftar subnet dengan Availability Zone mereka. IDs IDs Verifikasi bahwa subnet Anda dipetakan ke ID Availability Zone yang didukung. Untuk membuat subnet, lihat [Membuat subnet di VPC Anda di Panduan Pengguna](#) Amazon VPC.

Dua subnet (tanpa EVR) — Anda harus memiliki setidaknya dua subnet, dan mereka harus menjangkau dua Availability Zone.

Tiga subnet (hanya dengan EVR) — Anda harus memiliki setidaknya tiga subnet saat Anda menggunakan EVR, dan mereka harus menjangkau tiga atau lebih Availability Zone.

Persyaratan alamat IP gratis — Saat menggunakan Redshift Serverless tanpa peningkatan perutean VPC (EVR) diaktifkan, Anda harus memiliki setidaknya tiga alamat IP gratis yang tersedia di setiap subnet. Ini adalah persyaratan berfungsinya layanan dengan baik.

Saat memperbarui RPU untuk penyebaran Redshift Tanpa Server, setidaknya tiga alamat IP gratis harus tersedia di setiap subnet untuk mengakomodasi persyaratan operasional layanan.

Untuk informasi selengkapnya tentang mengalokasikan alamat IP dan memahami pengalamatan IP di Amazon VPC, lihat [Pengalamatan IP untuk subnet Anda dan VPCs subnet di](#) Panduan Pengguna Amazon VPC.

Without EVR

Jika Anda tidak menggunakan perutean VPC yang disempurnakan, Anda harus memiliki setidaknya tiga alamat IP gratis untuk setiap subnet, terlepas dari ukuran RPU dasar (4 hingga 1024 RPU), atau penggunaan RPU dari kelompok kerja atau kelompok kerja Anda. Kebutuhan akan 3 alamat IP juga berlaku untuk kelompok kerja yang memiliki kemampuan penskalaan dan pengoptimalan berbasis AI diaktifkan.

With Enhanced VPC Routing (EVR)

Jika Anda menggunakan perutean VPC yang disempurnakan dengan Redshift Serverless, jumlah minimum alamat IP yang diperlukan saat membuat workgroup adalah sebagai berikut:

Unit Pemrosesan Pergeseran Merah () RPU	Diperlukan alamat IP gratis	Ukuran CIDR minimum
4	9	/27
8	9	/27

Unit Pemrosesan Pergeseran Merah () RPU	Diperlukan alamat IP gratis	Ukuran CIDR minimum
16	13	/27
32	13	/27
64	21	/27
128	37	/26
256	69	/25
512	133	/24
1024	261	/23

Dengan EVR, Anda juga memerlukan alamat IP gratis saat memperbarui workgroup Anda untuk menggunakan lebih banyak. RPU Jumlah alamat IP gratis yang diperlukan saat memperbarui subnet untuk workgroup adalah sebagai berikut:

Unit Pemrosesan Pergeseran Merah () RPU	Unit Pemrosesan Redshift yang Diperbarui () RPU	Diperlukan alamat IP gratis
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

Kapasitas RPU dasar maksimum 1024 hanya tersedia dalam hal berikut: Wilayah AWS

- Timur AS (N. Virginia)
- AS Timur (Ohio)
- AS Barat (Oregon)
- Europe (Ireland)
- Eropa (London)

Untuk informasi selengkapnya tentang mengalokasikan alamat IP, lihat [Pengalamatan IP di Panduan Pengguna Amazon VPC](#).

Ruang penyimpanan setelah migrasi — Saat memigrasikan cluster kecil yang disediakan Amazon Redshift ke Amazon Redshift Tanpa Server, Anda mungkin melihat peningkatan alokasi ruang penyimpanan setelah migrasi. Ini adalah hasil dari alokasi ruang penyimpanan yang dioptimalkan, menghasilkan ruang penyimpanan yang telah dialokasikan sebelumnya. Ruang ini digunakan selama periode waktu tertentu karena data tumbuh di Amazon Redshift Tanpa Server.

Datasharing antara Amazon Redshift Serverless dan Amazon Redshift yang disediakan cluster — Saat datasharing di mana Amazon Redshift Serverless adalah produsen dan cluster yang disediakan adalah konsumen, cluster yang disediakan harus memiliki versi cluster lebih lambat dari 1.0.38214. Jika Anda menggunakan versi cluster lebih awal dari ini, kesalahan terjadi saat Anda menjalankan kueri. Anda dapat melihat versi cluster di konsol Amazon Redshift di tab Maintenance. Anda juga bisa lari `SELECT version();`.

Waktu eksekusi kueri maksimum - Waktu eksekusi yang telah berlalu untuk kueri, dalam hitungan detik. Waktu eksekusi tidak termasuk waktu yang dihabiskan menunggu dalam antrian. Jika kueri melebihi waktu eksekusi yang ditetapkan, Amazon Redshift Serverless menghentikan kueri. Nilai yang valid adalah 0—86.399.

Migrasi untuk tabel dengan kunci pengurutan yang disisipkan — Saat memigrasikan klaster yang disediakan Amazon Redshift ke Amazon Redshift Tanpa Server, Redshift mengonversi tabel dengan tombol pengurutan interleaved dan `DISTSTYLE KEY` menjadi kunci pengurutan majemuk. `DISTSTYLE` tidak berubah. Untuk informasi selengkapnya tentang gaya distribusi, lihat [Bekerja dengan gaya distribusi data](#) di Panduan Pengembang Amazon Redshift. Untuk informasi selengkapnya tentang kunci sortir, lihat [Bekerja dengan kunci sortir](#).

Berbagi VPC - Anda dapat membuat grup kerja Amazon Redshift Tanpa Server di VPC bersama. Jika Anda melakukannya, kami menyarankan agar Anda tidak menghapus pembagian sumber daya karena dapat mengakibatkan workgroup menjadi tidak tersedia.

IPv6 Support:

Amazon Redshift Serverless mendukung konfigurasi workgroup Amazon Redshift Anda dengan konfigurasi IPv6 dan alamat (dual-stack) IPv4 atau IPv4 -only dalam Virtual Private Clouds (). AWS VPCs Anda dapat mengaktifkan IPv6 dukungan saat membuat grup kerja Amazon Redshift Serverless baru atau memodifikasi grup kerja yang ada untuk mendukung pengalamatan. IPv6 Dengan kemampuan ini, Anda dapat menerapkan gudang Amazon Redshift Tanpa Server di subnet VPC IPv6 yang diaktifkan dan mengonfigurasi pengaturan jaringan untuk mendukung kebutuhan ruang alamat aplikasi yang diperluas. Aplikasi Anda sekarang dapat berkomunikasi dengan gudang Amazon Redshift Tanpa Server menggunakan salah satu IPv4 atau IPv6 protokol, memastikan kompatibilitas dengan arsitektur jaringan yang ada dan masa depan.

Kapasitas komputasi untuk Amazon Redshift Serverless

Dengan Amazon Redshift Serverless, skala kapasitas komputasi secara otomatis naik dan turun agar sesuai dengan kebutuhan beban kerja Anda. Kapasitas komputasi mengacu pada daya pemrosesan dan memori yang dialokasikan ke beban kerja Amazon Redshift Tanpa Server Anda. Kasus penggunaan umum termasuk menangani periode lalu lintas puncak, menjalankan analisis kompleks, atau memproses volume data yang besar secara efisien. Persyaratan berikut memberikan detail tentang cara Amazon Redshift mengelola kapasitas komputasi.

RPUs

Amazon Redshift Serverless mengukur kapasitas gudang data di Redshift Processing Units (). RPUs RPUs adalah sumber daya yang digunakan untuk menangani beban kerja. Satu RPU menyediakan memori 16 GB.

Kapasitas dasar

Setelan ini menentukan kapasitas gudang data dasar yang digunakan Amazon Redshift untuk menyajikan kueri. Kapasitas dasar ditentukan dalam RPUs. Anda dapat mengatur kapasitas dasar di Redshift Processing Units ()RPUs. Menetapkan kapasitas dasar yang lebih tinggi memastikan peningkatan kinerja kueri, terutama untuk pekerjaan pemrosesan data yang membutuhkan banyak sumber daya. Kapasitas dasar default untuk Amazon Redshift Serverless adalah 128. RPUs Anda dapat menyesuaikan pengaturan kapasitas Basis dari 4 RPUs hingga 512 RPUs. Anda dapat

mengatur nilai ini ke 4 RPU, atau dalam satuan 8 pada atau di atas 8 RPU (8,16,24... 512). Anda dapat mengatur nilai ini menggunakan AWS konsol, operasi UpdateWorkgroup API, atau update-workgroup operasi di AWS CLI.

Dengan kapasitas dasar minimum 4 RPU, Anda memiliki fleksibilitas untuk menjalankan beban kerja yang lebih sederhana hingga lebih kompleks berdasarkan biaya gudang data dan persyaratan kapasitas Anda. Kapasitas RPU 4 basis ditargetkan ke gudang yang berisi data kurang dari 32TB, dan kapasitas RPU dasar RPU 8, 16, dan 24 ditargetkan untuk beban kerja yang membutuhkan kurang dari 128TB data. Jika kebutuhan data Anda lebih besar dari 128 TB, Anda harus menggunakan minimal 32 basis RPU. Selain itu, untuk beban kerja yang memiliki tabel dengan kolom angka besar dan konkurensi yang lebih tinggi, sebaiknya gunakan 32 basis atau lebih. RPU

Basis maksimum RPU yang tersedia, 1024, menambahkan tingkat sumber daya komputasi tertinggi ke beban kerja Anda. Ini memberikan lebih banyak fleksibilitas untuk mendukung beban kerja dengan kompleksitas besar dan mempercepat pemuatan dan kueri data.

Note

Kapasitas RPU dasar maksimum yang diperluas sebesar 1024 tersedia sebagai berikut. Wilayah AWS Di wilayah lain, kapasitas dasar maksimum adalah 512 RPU.

- Timur AS (N. Virginia)
- AS Timur (Ohio)
- AS Barat (Oregon)
- Europe (Ireland)
- Eropa (Frankfurt)

Anda dapat menambah atau mengurangi RPU unit 32 saat mengatur kapasitas dasar antara 512-1024.

Jika Anda mengelola beban kerja yang lebih besar dan lebih kompleks, pertimbangkan untuk meningkatkan ukuran gudang data Redshift Tanpa Server Anda. Gudang yang lebih besar memiliki akses ke sumber daya komputasi yang lebih banyak, memungkinkan mereka memproses kueri dengan lebih efisien.

Berikut adalah beberapa contoh di mana memiliki kapasitas dasar yang lebih tinggi bermanfaat:

- Anda memiliki kueri kompleks yang membutuhkan waktu lama untuk dijalankan
- Tabel Anda memiliki sejumlah besar kolom.
- Kueri Anda memiliki jumlah yang tinggi. JOINS
- Kueri Anda mengumpulkan atau memindai sejumlah besar data dari sumber eksternal, seperti data lake.

Untuk informasi lebih lanjut tentang kuota dan batas Amazon Redshift Tanpa Server, buka. [Kuota untuk objek Amazon Redshift Tanpa Server](#)

Pertimbangan dan batasan untuk kapasitas Amazon Redshift Tanpa Server

Berikut ini adalah pertimbangan dan batasan untuk kapasitas Amazon Redshift Serverless. Untuk pertimbangan Redshift Tanpa Server umum, lihat. [Pertimbangan saat menggunakan Amazon Redshift Serverless](#)

- Konfigurasi 4 basis RPU mendukung kapasitas penyimpanan terkelola hingga 32 TB. Jika Anda menggunakan lebih dari 32 TB penyimpanan terkelola, Anda tidak dapat mengatur RPU dasar menjadi kurang dari 8 RPU.
- Konfigurasi 8 atau 16 basis RPU mendukung kapasitas penyimpanan terkelola Redshift hingga 128 TB. Jika Anda menggunakan lebih dari 128 TB penyimpanan terkelola, Anda tidak dapat mengatur basis ke kurang dari 32 RPU.
- Mengedit kapasitas dasar grup kerja Anda mungkin membatalkan beberapa kueri yang berjalan di workgroup Anda.
- Skala Redshift Tanpa Server RPU untuk gudang data Anda menggunakan penambahan berikut:
 - 4 hingga 8 RPU: Peningkatan dalam langkah 4 RPU.
 - 8 hingga 512 RPU: Peningkatan dalam langkah 8 RPU.
 - 512 hingga 1024 RPU: Peningkatan dalam langkah 32 RPU
- Vacuum boost hanya didukung untuk 8 RPU dan Di Atas. Untuk 8 RPU dan kurang, gunakan perintah berikut sebagai gantinya:

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

Redshift Tanpa Server dengan kapasitas 4 Unit Pemrosesan Redshift () RPU

Redshift Tanpa Server dengan 4 RPU kapasitas dasar sangat ideal untuk beban kerja yang lebih kecil atau kurang menuntut. Titik masuk ini menawarkan solusi yang fleksibel dan hemat biaya. Konfigurasi entry-level ini mendukung gudang data hingga sumber daya berikut:

- Penyimpanan terkelola Redshift hingga 32 TB.
- Maksimal 100 kolom per tabel
- Memori 64 GB

Jika Anda perlu melampaui batasan ini, Anda harus meningkatkan kapasitas dasar Anda secara manual, daripada mengandalkan auto-scaling. Setelah Anda menskalakan gudang data Anda melebihi 4 RPU, gudang data Anda akan terus menggunakan lebih banyak RPU, dan Amazon Redshift tidak akan menskalakan gudang data Anda kembali ke 4. RPU

Note

Anda dapat membuat tabel dengan lebih dari 100 kolom saat menggunakan 4 basis RPU, namun, kami sarankan Anda membatasi tabel hingga 100 kolom. Melebihi batas ini dapat menyebabkan gudang data Anda kehabisan memori selama eksekusi kueri, yang menurunkan kinerja.

Anda dapat membuat gudang data yang menggunakan 4 RPU sebagai berikut: Wilayah AWS

- AS Timur (Ohio)
- AS Timur (Virginia Utara)
- AS Barat (California Utara)
- AS Barat (Oregon)
- Asia Pasifik (Mumbai)
- Asia Pasifik (Singapura)
- Asia Pasifik (Sydney)
- Asia Pasifik (Tokyo)
- Eropa (Irlandia)
- Eropa (Stockholm)

Penskalaan dan pengoptimalan berbasis AI

Fitur penskalaan dan pengoptimalan berbasis AI tersedia di semua AWS Wilayah di mana Amazon Redshift Serverless tersedia.

Amazon Redshift Serverless menawarkan fitur penskalaan dan pengoptimalan berbasis AI canggih untuk memenuhi beragam persyaratan beban kerja. Gudang data mungkin memiliki masalah penyediaan berikut:

- Gudang data dapat disediakan secara berlebihan untuk meningkatkan kinerja kueri intensif sumber daya
- Gudang data mungkin kurang disediakan untuk menghemat biaya.

Mencapai keseimbangan yang tepat antara kinerja dan biaya untuk beban kerja gudang data sangat menantang, terutama dengan kueri ad-hoc dan volume data yang terus bertambah. Saat menjalankan beban kerja campuran, yang terdiri dari kueri intensif sumber daya rendah dan tinggi, ada kebutuhan untuk penskalaan cerdas. Fitur penskalaan dan pengoptimalan berbasis AI secara otomatis menskalakan komputasi Tanpa Server atau RPU sebagai respons terhadap pertumbuhan data. Fitur ini juga membantu mempertahankan kinerja kueri dalam sasaran kinerja harga yang ditargetkan. Penskalaan dan pengoptimalan berbasis AI secara dinamis mengalokasikan sumber daya komputasi saat volume data meningkat, memastikan kueri terus memenuhi target kinerja. Penskalaan dan pengoptimalan berbasis AI memungkinkan layanan beradaptasi dengan mulus terhadap perubahan persyaratan beban kerja, tanpa perlu intervensi manual atau perencanaan kapasitas yang kompleks.

Amazon Redshift Serverless menyediakan solusi penskalaan yang lebih komprehensif dan responsif berdasarkan faktor-faktor seperti kompleksitas kueri dan volume data. Fitur ini memungkinkan untuk mengoptimalkan kinerja harga beban kerja sambil mempertahankan fleksibilitas untuk menangani berbagai beban kerja dan mengembangkan kumpulan data secara efisien. Amazon Redshift Serverless dapat secara otomatis melakukan pengoptimalan berbasis AI ke titik akhir Amazon Redshift Tanpa Server untuk memenuhi target performa harga yang ditentukan untuk grup kerja Tanpa Server Anda. Optimalisasi harga-kinerja otomatis ini sangat membantu jika Anda tidak tahu kapasitas dasar apa yang harus ditetapkan untuk beban kerja Anda, atau jika beberapa bagian dari beban kerja Anda mungkin mendapat manfaat dari lebih banyak sumber daya yang dialokasikan.

Contoh

Jika organisasi Anda biasanya menjalankan beban kerja yang hanya memerlukan 32 RPU tetapi tiba-tiba memperkenalkan kueri yang lebih kompleks, Anda mungkin tidak mengetahui kapasitas dasar yang sesuai. Menetapkan kapasitas dasar yang lebih tinggi menghasilkan kinerja yang lebih baik tetapi juga menimbulkan biaya yang lebih tinggi, sehingga biayanya mungkin tidak sesuai dengan harapan Anda. Menggunakan penskalaan berbasis AI dan pengoptimalan sumber daya, Amazon Redshift Serverless secara otomatis menyesuaikan RPU untuk memenuhi target kinerja harga Anda sambil menjaga biaya dioptimalkan untuk organisasi Anda. Optimalisasi otomatis ini berguna terlepas dari ukuran beban kerja. Pengoptimalan otomatis dapat membantu Anda memenuhi target kinerja harga organisasi Anda jika Anda memiliki sejumlah kueri kompleks.

 Note

Target harga-kinerja adalah pengaturan khusus kelompok kerja. Kelompok kerja yang berbeda dapat memiliki target harga-kinerja yang berbeda.

Agar biaya tetap dapat diprediksi, tetapkan batas kapasitas maksimum yang diizinkan untuk dialokasikan oleh Amazon Redshift Serverless ke beban kerja Anda.

Untuk mengonfigurasi target harga-kinerja, gunakan konsol. AWS Anda harus mengaktifkan target harga-kinerja secara eksplisit saat membuat grup kerja Tanpa Server. Anda juga dapat memodifikasi target harga-kinerja setelah Anda membuat grup kerja Tanpa Server. Ketika Anda mengaktifkan target harga-kinerja, itu diatur ke Balanced secara default.

Untuk mengedit target harga-kinerja untuk grup kerja Anda

1. Di konsol Amazon Redshift Tanpa Server, pilih konfigurasi Workgroup.
2. Pilih workgroup yang ingin Anda edit target harga-kinerja. Pilih tab Kinerja, lalu pilih Edit.
3. Pilih target Price-performance, dan sesuaikan slider ke pengaturan yang Anda inginkan.
4. Pilih Simpan perubahan.
5. Untuk memperbarui jumlah maksimum Amazon Redshift Serverless RPU yang dapat dialokasikan ke beban kerja Anda, pilih tab Batas pada bagian Konfigurasi Workgroup.

Anda dapat menggunakan slider target Price-performance untuk mengatur keseimbangan yang Anda inginkan antara biaya dan kinerja. Dengan menggerakkan slider, Anda dapat memilih salah satu opsi berikut:

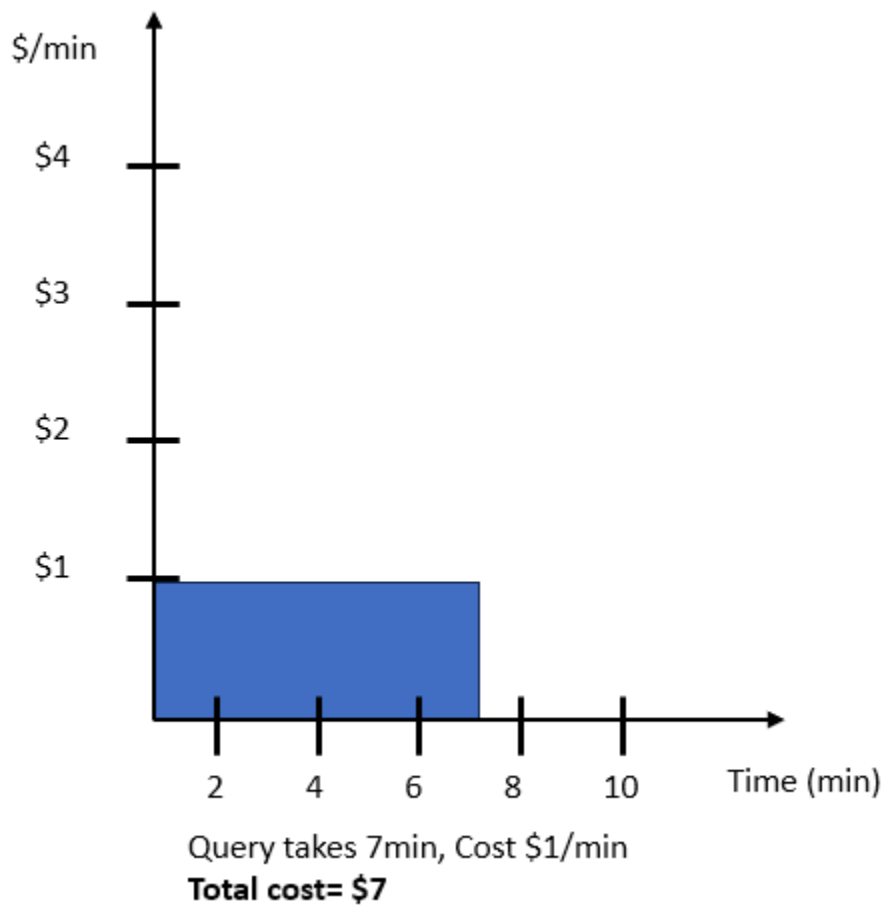
- **Mengoptimalkan biaya** — Pengaturan ini memprioritaskan penghematan biaya. Amazon Redshift Serverless mencoba untuk secara otomatis meningkatkan kapasitas komputasi saat melakukannya tidak menimbulkan biaya tambahan. Amazon Redshift Serverless juga mencoba mengurangi sumber daya komputasi dengan biaya lebih rendah, mungkin meningkatkan runtime kueri.
- **Seimbang** — Pengaturan ini menciptakan keseimbangan antara kinerja dan biaya. Amazon Redshift Tanpa Server menskalakan kinerja, dan dapat menghasilkan kenaikan atau penurunan biaya yang moderat. Ini adalah pengaturan yang disarankan untuk sebagian besar gudang data Amazon Redshift Tanpa Server.
- **Mengoptimalkan kinerja** - Pengaturan ini memprioritaskan kinerja. Amazon Redshift menskalakan secara agresif untuk kinerja tinggi, berpotensi menimbulkan biaya lebih tinggi.
- **Posisi menengah**: Anda juga dapat mengatur slider ke salah satu dari dua posisi perantara antara Seimbang dan Mengoptimalkan biaya atau Mengoptimalkan untuk kinerja. Gunakan pengaturan ini jika optimasi penuh untuk biaya atau kinerja terlalu ekstrim.

Pertimbangan saat memilih target harga-kinerja Anda

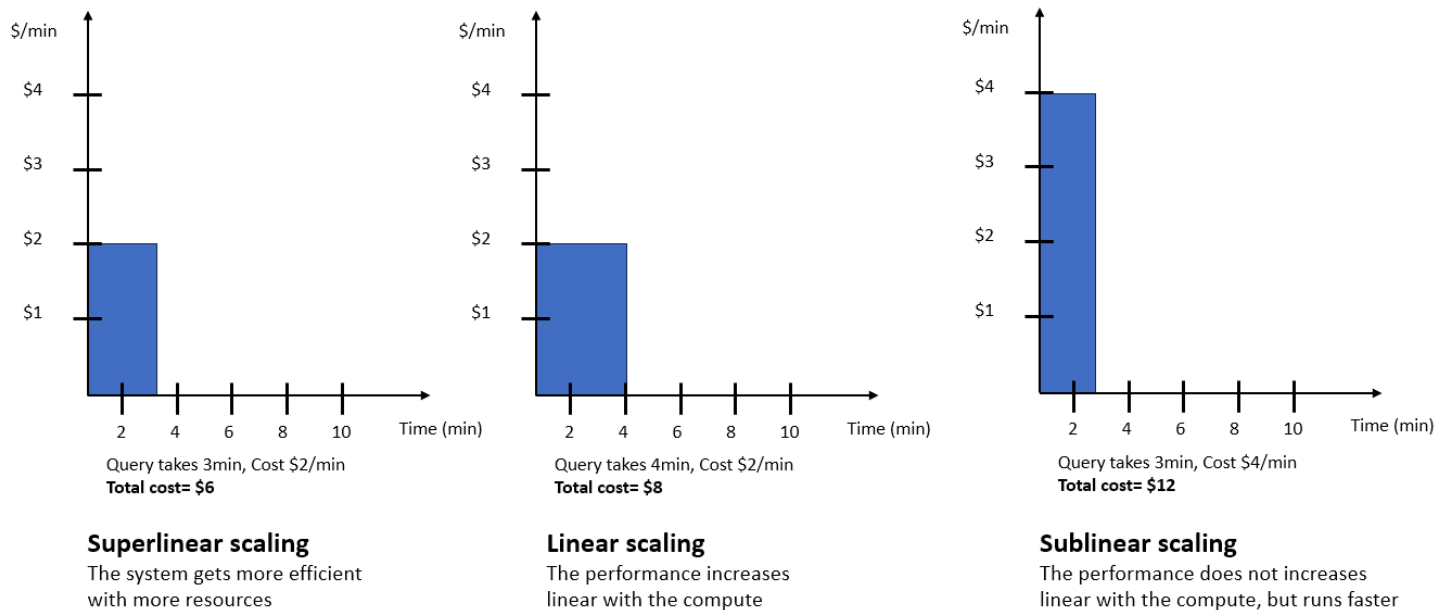
Anda dapat menggunakan penggeser harga-kinerja untuk memilih target harga-kinerja yang Anda inginkan untuk beban kerja Anda. Algoritma penskalaan dan pengoptimalan berbasis AI belajar dari waktu ke waktu dari riwayat beban kerja Anda, dan meningkatkan prediksi dan akurasi keputusan.

Contoh

Untuk contoh ini, asumsikan kueri yang membutuhkan waktu tujuh menit dan biaya \$7. Gambar berikut menunjukkan runtime kueri dan biaya tanpa penskalaan.



Kueri yang diberikan mungkin skala dalam beberapa cara yang berbeda, seperti yang ditunjukkan di bawah ini. Berdasarkan target harga-kinerja yang Anda pilih, penskalaan berbasis AI memprediksi bagaimana kueri memperdagangkan kinerja dan biaya, dan menskalakannya sesuai dengan itu. Memilih opsi slider yang berbeda menghasilkan hasil sebagai berikut:



- **Mengoptimalkan Biaya** — Dengan opsi Optimize for Cost, gudang data Anda menskalakan pilihan yang menurunkan biaya Anda. Dalam contoh sebelumnya, pendekatan penskalaan super linier menunjukkan perilaku ini. Penskalaan hanya akan terjadi jika dapat dilakukan dengan cara yang hemat biaya sesuai dengan prediksi model penskalaan. Jika model penskalaan memprediksi bahwa penskalaan yang dioptimalkan biaya tidak dimungkinkan untuk beban kerja yang diberikan, maka gudang data tidak akan diskalakan.
- **Seimbang** — Dengan opsi Balanced, sistem menskalakan sambil menyeimbangkan pertimbangan biaya dan kinerja, dengan potensi peningkatan biaya yang terbatas. Opsi Balanced melakukan penskalaan beban kerja superlinear, linier, dan mungkin sublinear.
- **Optimize for Performance** — Dengan opsi Optimizes for Performance, selain metode sebelumnya untuk meningkatkan kinerja, sistem juga menskalakan meskipun biayanya lebih tinggi, dan mungkin tidak sebanding dengan peningkatan runtime. Dengan Optimizes for Performance, sistem melakukan penskalaan superlinear, penskalaan linier, dan penskalaan sublinear jika memungkinkan. Semakin dekat posisi slider ke posisi Optimizes for Performance, semakin Amazon Redshift Serverless mengizinkan penskalaan sublinear.

Perhatikan hal berikut saat mengatur slider Price-Performance:

- Anda dapat mengubah pengaturan harga-kinerja kapan saja, tetapi penskalaan beban kerja tidak akan segera berubah. Penskalaan berubah seiring waktu saat sistem belajar tentang beban kerja saat ini. Kami menyarankan untuk memantau Grup Kerja Tanpa Server selama 1-3 hari untuk memverifikasi dampak pengaturan baru.

- Opsi slider harga-kinerja Kapasitas Max dan Max RPU-jam bekerja sama. Kapasitas maksimum dan jam RPU Maks adalah kontrol untuk membatasi maksimum RPU yang memungkinkan Amazon Redshift Serverless memungkinkan gudang data untuk diskalakan, dan jam RPU maksimum yang Amazon Redshift Serverless memungkinkan gudang data untuk dikonsumsi. Amazon Redshift Serverless selalu menghormati dan menerapkan pengaturan ini, terlepas dari pengaturan target harga-kinerja.

Memantau penskalaan otomatis sumber daya

Anda dapat memantau penskalaan RPU berbasis AI dengan cara berikut:

- Tinjau grafik kapasitas RPU yang digunakan di konsol Amazon Redshift.
- Pantau ComputeCapacity metrik di bawah AWS/Redshift-Serverless dan Workgroup di dalam CloudWatch.
- Kueri tampilan [SYS_QUERY_HISTORY](#). Berikan ID kueri atau teks kueri tertentu untuk mengidentifikasi periode waktu. Gunakan periode waktu ini untuk menanyakan tampilan [sistem SYS_SERVERLESS_USAGE](#) untuk menemukan nilainya. `compute_capacity` `compute_capacity` Bidang menunjukkan RPU skala selama runtime kueri.

Gunakan contoh berikut untuk query `SYS_QUERY_HISTORY` tampilan. Ganti nilai sampel dengan teks kueri Anda.

```
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
```

Jalankan kueri berikut untuk melihat bagaimana `compute_capacity` diskalakan selama periode dari `start_time` ke `end_time`. Ganti `start_time` dan `end_time` dalam query berikut dengan output dari query sebelumnya:

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

Untuk step-by-step petunjuk penggunaan fitur ini, lihat [Mengonfigurasi pemantauan, batasan, dan alarm di Amazon Redshift Tanpa Server](#) agar biaya tetap dapat diprediksi.

Pertimbangan saat menggunakan penskalaan dan pengoptimalan berbasis AI

Pertimbangkan hal berikut saat menggunakan penskalaan dan pengoptimalan berbasis AI:

- Untuk beban kerja yang ada di Amazon Redshift Serverless yang membutuhkan 32 hingga 512 Base RPU, sebaiknya gunakan penskalaan dan pengoptimalan berbasis AI Amazon Redshift Serverless untuk hasil yang optimal. Kami tidak merekomendasikan penggunaan fitur ini untuk kurang dari 32 Base RPU atau lebih dari 512 Base RPU beban kerja.
- Target harga-kinerja secara otomatis mengoptimalkan beban kerja, meskipun hasilnya dapat bervariasi. Sebaiknya gunakan fitur ini dari waktu ke waktu sehingga sistem dapat mempelajari pola spesifik Anda dengan menjalankan beban kerja yang representatif.
- Penskalaan dan pengoptimalan berbasis AI menggunakan waktu optimal untuk menerapkan pengoptimalan ke grup kerja Tanpa Server, bergantung pada beban kerja yang berjalan di instans Amazon Redshift Tanpa Server Anda.

Untuk mempelajari lebih lanjut tentang pengoptimalan berbasis AI dan penskalaan sumber daya, tonton video berikut.

Penagihan untuk Amazon Redshift Tanpa Server

Penagihan untuk kapasitas komputasi

Anda dapat membeli kapasitas untuk Amazon Redshift Serverless dengan dua cara:

- Anda dapat membeli kapasitas sesuai permintaan — Ketika Anda memilih kapasitas komputasi sesuai permintaan, Anda membayar sumber daya saat Anda pergi. Ini adalah pilihan terbaik jika Anda baru mulai menggunakan Amazon Redshift Tanpa Server atau jika Anda belum memiliki pemahaman yang baik tentang pola penggunaan tetap Anda. On-demand menawarkan fleksibilitas paling banyak. Untuk informasi selengkapnya, lihat [Penagihan untuk kapasitas komputasi sesuai permintaan](#).
- Anda dapat membeli reservasi — Reservasi memberikan diskon ketika Anda membeli sumber daya komputasi dalam jumlah yang telah ditetapkan untuk jangka waktu tertentu, misalnya selama satu tahun. Ini adalah ide yang baik ketika Anda tahu Anda akan menggunakan sejumlah kapasitas dengan mantap. Ini membantu untuk menghemat uang ketika Anda dapat memperkirakan

beberapa kebutuhan kapasitas Anda. Untuk informasi selengkapnya, lihat [Penagihan untuk reservasi tanpa server](#).

Anda dapat menggunakan reservasi dan sumber daya sesuai permintaan bersama-sama. Anda tidak perlu menggunakan satu atau yang lain.

Untuk informasi harga terperinci, lihat harga [Amazon Redshift](#).

Penagihan untuk kapasitas komputasi sesuai permintaan

Kapasitas dasar dan pengaruhnya terhadap penagihan

Saat kueri berjalan, Anda ditagih sesuai dengan kapasitas yang digunakan dalam durasi tertentu, dalam jam RPU per detik. Saat tidak ada kueri yang berjalan, Anda tidak ditagih untuk kapasitas komputasi. Anda juga dikenakan biaya untuk Redshift Managed Storage (RMS), berdasarkan jumlah data yang disimpan.

Saat Anda membuat workgroup, Anda memiliki opsi untuk mengatur kapasitas Basis untuk komputasi. Untuk memenuhi price/performance persyaratan beban kerja Anda di tingkat kelompok kerja, sesuaikan kapasitas dasar yang lebih tinggi atau lebih rendah untuk kelompok kerja yang ada. Pilih workgroup dari konfigurasi Workgroup dan pilih tab Limits untuk mengubah kapasitas dasar menggunakan konsol.

Seiring bertambahnya jumlah kueri, Amazon Redshift Tanpa Server menskalakan secara otomatis untuk memberikan kinerja yang konsisten.

Batas penggunaan jam RPU maksimum

Agar biaya dapat diprediksi untuk Amazon Redshift Serverless, Anda dapat mengatur jam RPU Maksimum yang digunakan per hari, per minggu, atau per bulan. Anda dapat mengaturnya menggunakan konsol atau dengan API. Ketika batas tercapai, Anda dapat menentukan bahwa entri log ditulis ke tabel sistem, atau Anda menerima peringatan, atau kueri pengguna dimatikan. Mengatur jam RPU maksimum membantu menjaga biaya Anda tetap terkendali. Pengaturan untuk jam RPU maksimum berlaku untuk grup kerja untuk kueri yang mengakses data di gudang data dan kueri yang mengakses data eksternal, seperti di tabel eksternal di Amazon S3.

Berikut ini adalah contohnya:

Asumsikan Anda menetapkan batas untuk 100 jam untuk setiap minggu. Untuk melakukan ini di konsol, Anda melakukan hal berikut:

1. Pilih grup kerja Anda, lalu pilih Kelola batas penggunaan di bawah tab Batas.
2. Tambahkan batas penggunaan, pilih frekuensi Mingguan, durasi 100 jam, dan pengaturan tindakan untuk Matikan kueri pengguna.

Dalam contoh ini, jika Anda mencapai batas 100 jam RPU selama seminggu, kueri dimatikan.

Menyetel jam RPU maksimum untuk grup kerja tidak membatasi kinerja atau sumber daya komputasi untuk grup kerja. Anda dapat menyesuaikan pengaturan kapan saja tanpa mempengaruhi pemrosesan kueri. Tujuan untuk menetapkan jam RPU maksimum adalah untuk membantu Anda memenuhi persyaratan harga dan kinerja Anda. [Untuk informasi selengkapnya tentang penagihan tanpa server, lihat harga Amazon Redshift.](#)

Cara lain untuk menjaga biaya Amazon Redshift Serverless dapat diprediksi adalah dengan menggunakan [Deteksi Anomali AWS Biaya](#) untuk mengurangi kemungkinan kejutan penagihan dan memberikan kontrol lebih besar.

 Note

[Kalkulator harga Amazon Redshift](#) sangat membantu untuk memperkirakan harga. Anda memasukkan sumber daya komputasi yang Anda butuhkan dan menyediakan pratinjau biaya.

Menetapkan kapasitas Maks untuk mengontrol biaya untuk sumber daya komputasi

Pengaturan kapasitas Max berfungsi sebagai plafon RPU yang dapat ditingkatkan oleh Amazon Redshift Serverless. Ini membantu mengontrol biaya Anda untuk sumber daya komputasi. Dengan cara yang mirip dengan bagaimana kapasitas dasar menetapkan jumlah minimum sumber daya komputasi yang tersedia, kapasitas Max menetapkan batas atas penggunaan RPU. Dengan begitu, ini membantu pengeluaran Anda mematuhi rencana Anda. Kapasitas maksimum berlaku khusus untuk setiap kelompok kerja dan membatasi penggunaan komputasi setiap saat.

Bagaimana kapasitas Max berbeda dari batas penggunaan jam RPU

Tujuan dari batas jam RPU maksimum dan pengaturan kapasitas Max adalah untuk mengontrol biaya. Tetapi mereka mencapai ini melalui cara yang berbeda. Poin-poin berikut menjelaskan perbedaannya:

- Kapasitas maksimum - Pengaturan ini menetapkan jumlah tertinggi yang digunakan Amazon Redshift Tanpa Server untuk tujuan penskalaan. RPU Ketika penskalaan komputasi otomatis diperlukan, memiliki nilai yang lebih tinggi untuk kapasitas Max dapat meningkatkan throughput kueri. Ketika batas kapasitas Max tercapai, kelompok kerja tidak meningkatkan sumber daya lebih jauh.
- Batas penggunaan jam RPU maksimum - Tidak seperti kapasitas Max, pengaturan ini tidak menetapkan batas kapasitas. Tapi itu melakukan tindakan lain untuk membantu Anda membatasi biaya. Ini termasuk menambahkan entri ke log, memberi tahu Anda, atau menghentikan kueri agar tidak berjalan, jika Anda mau.

Anda dapat menggunakan kapasitas Max secara eksklusif, atau Anda dapat memujinya dengan tindakan dari batas penggunaan jam RPU maksimum.

Kasus penggunaan kapasitas Max

Setiap workgroup dapat memiliki pengaturan kapasitas Max yang berbeda. Ini membantu Anda menegakkan persyaratan penganggaran. Untuk mengilustrasikan cara kerjanya, asumsikan hal berikut:

- Anda memiliki workgroup dengan kapasitas dasar diatur ke 256 RPU. Anda memiliki beban kerja tetap di lebih dari 256 RPU untuk sebagian besar bulan.
- Kapasitas maks diatur ke 512 RPU.

Asumsikan Anda memiliki penggunaan tinggi yang tidak terduga selama periode tiga hari untuk menghasilkan laporan statistik ad-hoc. Dalam hal ini, Anda memiliki kapasitas Max yang disetel untuk menghindari biaya komputasi di luar biaya 512 RPU. Ketika Anda melakukan ini, Anda dapat yakin bahwa kapasitas komputasi tidak akan melebihi batas atas ini.

Catatan penggunaan untuk kapasitas Max

Catatan ini dapat membantu Anda mengatur kapasitas Max dengan tepat:

- Setiap workgroup Amazon Redshift Tanpa Server dapat memiliki pengaturan kapasitas Maks yang berbeda.
- Jika Anda memiliki periode penggunaan sumber daya yang sangat tinggi dan kapasitas Max diatur ke tingkat RPU rendah, itu dapat menunda pemrosesan beban kerja dan menghasilkan pengalaman pengguna yang tidak optimal.

- Mengkonfigurasi pengaturan kapasitas Max tidak mengganggu kueri yang sedang berjalan, bahkan selama masa penggunaan RPU tinggi. Itu tidak berfungsi seperti batas penggunaan, yang dapat menghentikan kueri agar tidak berjalan. Ini hanya membatasi sumber daya komputasi yang tersedia untuk workgroup. Anda dapat melihat kapasitas yang digunakan selama periode waktu tertentu di dasbor Amazon Redshift Tanpa Server. Untuk informasi selengkapnya tentang melihat data ringkasan, lihat [Memeriksa data ringkasan Amazon Redshift Tanpa Server menggunakan dasbor](#).
- Pengaturan kapasitas Max teratas adalah 5632 RPU.

Cara mengatur kapasitas Max

Anda dapat mengatur kapasitas Max di konsol. Untuk workgroup yang ada, Anda dapat mengubah pengaturan di bawah konfigurasi Workgroup. Anda juga dapat menggunakan CLI untuk mengaturnya dengan menggunakan perintah seperti contoh berikut:

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

Ini menetapkan pengaturan kapasitas Max untuk workgroup dengan nama yang diberikan. Setelah mengaturnya, Anda dapat memeriksa nilai pada konsol untuk memverifikasinya. Anda juga dapat memeriksa nilai menggunakan CLI dengan menjalankan perintah. `get-workgroup`

Anda dapat mematikan pengaturan kapasitas Max dengan menyetelnya -1, seperti berikut ini:

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

Memantau penggunaan dan biaya Amazon Redshift Tanpa Server

Ada beberapa cara Anda dapat memperkirakan penggunaan dan penagihan untuk Amazon Redshift Serverless. Tampilan sistem dapat membantu karena metadata sistem, termasuk kueri dan data penggunaan, tepat waktu dan Anda tidak perlu melakukan pengaturan apa pun untuk menanyakannya. CloudWatch juga dapat berguna untuk memantau penggunaan instans Amazon Redshift Tanpa Server Anda, dan memiliki fitur tambahan untuk memberikan wawasan dan menetapkan tindakan.

Memvisualisasikan penggunaan dengan menanyakan tampilan sistem

Kueri tabel sistem `SYS_SERVERLESS_USAGE` untuk melacak penggunaan dan mendapatkan biaya untuk kueri:

```
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
```

Kueri ini memberikan biaya per hari yang dikeluarkan untuk Amazon Redshift Tanpa Server, berdasarkan penggunaan.

Catatan penggunaan untuk menentukan penggunaan dan biaya

- Anda membayar beban kerja yang Anda jalankan dalam RPU-jam per detik, dengan biaya minimum 60 detik.
- Catatan dari tabel sistem sys_serverless_usage menunjukkan biaya yang dikeluarkan dalam interval waktu 1 menit. Memahami kolom berikut ini penting:

Kolom charged_seconds:

- Menyediakan unit komputasi (RPU) detik yang diisi selama interval waktu. Hasilnya termasuk biaya minimum apa pun di Amazon Redshift Tanpa Server.
- Memiliki informasi tentang penggunaan sumber daya komputasi setelah transaksi selesai. Dengan demikian, nilai kolom ini mungkin 0 jika transaksi belum selesai.

Kolom compute_seconds:

- Menyediakan informasi penggunaan komputasi waktu nyata. Ini tidak termasuk biaya minimum apa pun di Amazon Redshift Serverless. Dengan demikian dapat berbeda sampai tingkat tertentu dari detik yang dibebankan yang ditagih selama interval.
- Menampilkan informasi penggunaan selama setiap transaksi (bahkan jika transaksi belum berakhir), dan karenanya data yang diberikan adalah real-time.
- Ada situasi di mana compute_seconds adalah 0 tetapi charged_seconds lebih besar dari 0, atau sebaliknya. Ini adalah perilaku normal yang dihasilkan dari cara data direkam dalam tampilan sistem. Untuk representasi detail penggunaan tanpa server yang lebih akurat, kami sarankan untuk menggabungkan data dalam SYS_SERVERLESS_USAGE.

Untuk informasi selengkapnya tentang memantau tabel dan tampilan, lihat [Memantau kueri dan beban kerja dengan Amazon Redshift Tanpa Server](#).

Memvisualisasikan penggunaan dengan CloudWatch

Anda dapat menggunakan metrik yang tersedia CloudWatch untuk melacak penggunaan. Metrik yang dihasilkan CloudWatch adalah `ComputeSeconds`, menunjukkan total detik RPU yang digunakan pada menit saat ini dan `ComputeCapacity`, menunjukkan total kapasitas komputasi untuk menit itu. Metrik penggunaan juga dapat ditemukan di konsol Redshift di dasbor Redshift Serverless. Untuk informasi selengkapnya CloudWatch, lihat [Apa itu Amazon CloudWatch?](#)

Penagihan untuk reservasi tanpa server

Amazon Redshift Serverless memungkinkan Anda menjalankan dan menskalakan analitik tanpa harus menyediakan dan mengelola klaster dengan model harga. `pay-as-you-go` Sekarang dengan reservasi tanpa server, Anda dapat lebih mengoptimalkan biaya komputasi Anda dan meningkatkan prediktabilitas biaya beban kerja yang ada dan baru di Redshift Serverless.

Amazon Redshift mengelola reservasi tanpa server di tingkat akun AWS pembayar, dan reservasi dapat dibagi di antara beberapa AWS akun, memungkinkan Anda mengurangi biaya komputasi hingga 24% untuk semua beban kerja Redshift Tanpa Server di akun Anda. AWS Amazon Redshift menagih reservasi tanpa server per jam dan pemesanan meter per detik, menawarkan model penagihan yang konsisten, 24 jam sehari, tujuh hari seminggu, sambil mempertahankan fleksibilitas yang ditawarkan oleh Redshift Serverless. Amazon Redshift membebaskan biaya penggunaan apa pun yang melebihi tingkat RPU yang ditentukan dengan tarif sesuai permintaan standar.

Note

Jika ingin membatasi penggunaan sesuai permintaan, Anda dapat menggunakan setelan kapasitas Maks untuk menetapkan batas penggunaan sumber daya untuk grup kerja Anda. Untuk informasi selengkapnya, lihat [Penagihan untuk Amazon Redshift Tanpa Server](#).

Manfaat reservasi tanpa server

Reservasi tanpa server adalah opsi harga diskon untuk Amazon Redshift Tanpa Server. Reservasi tanpa server memberi Anda opsi untuk berkomitmen pada sejumlah Unit Pemrosesan Redshift (RPU) tertentu selama satu tahun dengan diskon dari tarif sesuai permintaan (OD), tanpa pembayaran di muka. Anda dapat menerima diskon yang lebih besar dengan pembayaran di muka. Dengan reservasi tanpa server, Anda dapat mengoptimalkan biaya komputasi dan meningkatkan prediktabilitas biaya beban kerja yang ada dan baru di Tanpa Server.

Setiap reservasi tanpa server dibeli di tingkat AWS akun dan dapat dibagikan di antara beberapa grup kerja Amazon Redshift Tanpa Server di akun pembayar yang sama. Ini memberi Anda fleksibilitas dalam bagaimana diskon diterapkan. Beberapa kelompok kerja dengan pola beban kerja yang berbeda dapat berbagi reservasi.

Cara kerja reservasi tanpa server

Pemesanan RPU adalah proses sederhana yang hanya membutuhkan beberapa menit untuk menyelesaikannya. Ini termasuk menentukan tingkat RPU untuk cadangan dan jenis pembayaran. Amazon Redshift Serverless menggunakan alat AWS penagihan dan manajemen biaya standar yang membantu Anda menentukan tingkat reservasi yang Anda butuhkan dan memantau penggunaan Anda secara terus menerus. Reservasi tanpa server dikelola di tingkat akun AWS pembayar dan dapat dibagikan di bawah akun pembayar yang sama, dan memungkinkan Anda mengurangi biaya komputasi hingga 24% pada semua beban kerja Redshift Tanpa Server di akun Anda. AWS Reservasi tanpa server ditagih per jam dan diukur per detik, menawarkan model penagihan yang konsisten, 24 jam sehari, tujuh hari seminggu, sambil mempertahankan fleksibilitas yang ditawarkan oleh Redshift Serverless. Setiap penggunaan yang melebihi level RPU yang ditentukan dibebankan pada tarif sesuai permintaan Redshift Serverless standar.

Anda dapat membeli beberapa reservasi tanpa server dalam akun yang sama. AWS Ketika Anda membeli reservasi tanpa server tambahan, mereka saling berlapis-lapis. Misalnya, jika Anda membeli dua reservasi dan memilih 100 RPU untuk masing-masing, itu memberi Anda total 200 dengan tarif RPU diskon.

Note

Jika Anda ingin menetapkan batas untuk penggunaan sesuai permintaan, Anda dapat mengatur maksimum RPU di konsol Amazon Redshift Tanpa Server untuk grup kerja dengan memilih tab Batas, lalu memilih Kelola batas penggunaan.

Setelah Anda membeli reservasi tanpa server, reservasi langsung berlaku dan muncul di konsol Redshift di dasbor reservasi Tanpa Server.

Menganalisis penggunaan RPU (Redshift Processing Unit) Anda untuk menentukan tingkat reservasi yang Anda butuhkan

Reservasi Tanpa Server Redshift memungkinkan Anda mengunci biaya komputasi yang dapat diprediksi dan lebih rendah dengan berkomitmen pada sejumlah Unit Pemrosesan Redshift (RPU)

tertentu selama satu tahun, memberi Anda diskon dibandingkan harga sesuai permintaan. Diskon ini bisa mencapai 20 persen dengan opsi tanpa dimuka, atau hingga 24 persen saat Anda membayar di muka. Anda membeli Reservasi Tanpa Server Redshift di AWS tingkat akun pembayar, dan tabungan Anda secara otomatis berlaku untuk grup kerja Redshift Tanpa Server di akun AWS tertaut mana pun, sehingga Anda dapat mengelola anggaran secara terpusat sambil mendukung beberapa tim. Redshift Tanpa Server mengukur penggunaan pada granularitas per detik, dirata-ratakan setiap jam, dan kemudian ditagih setiap jam, memastikan Anda hanya membayar untuk kapasitas yang Anda gunakan. Reservasi Tanpa Server Redshift menggabungkan aplikasi fleksibel di seluruh akun dengan penghematan berbasis jangka waktu, memberi Anda harga analitik yang dapat diprediksi tanpa mengorbankan kelincahan Redshift Tanpa Server.

Menganalisis Penggunaan RPU untuk Reservasi

Anda dapat menentukan tingkat penggunaan RPU Anda dengan salah satu dari dua cara: Anda dapat menggunakan dasbor Redshift Tanpa Server untuk tampilan tujuh hari, atau menggunakan Cost Explorer untuk analisis jangka panjang. Prosedur berikut menunjukkan cara menganalisis penggunaan RPU Anda:

Metode 1: Dasbor Tanpa Server Redshift (tampilan 7 hari)

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Buka dasbor Tanpa Server.
3. Pilih workgroup Anda.
4. Lihat penggunaan kapasitas RPU untuk jangka waktu lama dari jam terakhir hingga satu minggu.

Metode 2: AWS Cost Explorer (Analisis jangka panjang)

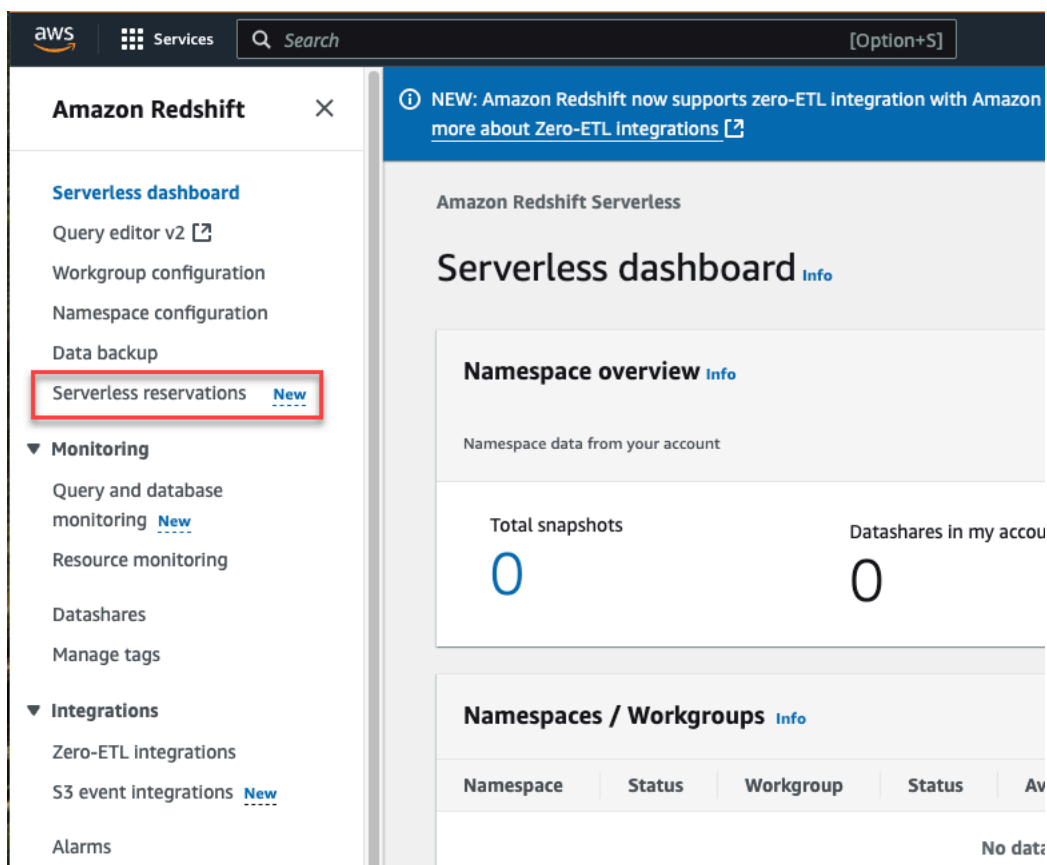
1. Masuk ke Konsol Manajemen AWS dan buka konsol Cost Explorer di <https://console.aws.amazon.com/costmanagement/>.
2. Atur Granularitas ke Per Jam
3. Kelompokkan berdasarkan jenis Penggunaan
4. Terapkan filter berikut:
 - Layanan: Redshift
 - Wilayah: Wilayah lokal Anda
 - Jenis penggunaan: Filter untuk Redshift: ServerlessUsage

5. Tinjau grafik Biaya dan penggunaan untuk penggunaan tanpa server per jam di wilayah pilihan Anda

Membeli reservasi tanpa server menggunakan konsol

Ketika Anda membeli reservasi, Anda memilih tingkat RPU yang akan didiskon. Sebelum memilih tingkat RPU Anda, ada baiknya mengetahui kapasitas dasar Anda dan kapasitas sesuai permintaan yang Anda gunakan dari waktu ke waktu. Bagian ini menunjukkan kepada Anda cara menentukan kapasitas Anda dan memesan reservasi tanpa server.

Untuk memulai, di konsol Redshift, pilih Tanpa Server, dan kemudian Reservasi tanpa server dari menu.



Konsol menampilkan deskripsi fitur dan daftar reservasi yang ada. Dari sini Anda dapat membeli reservasi, atau Anda dapat menggunakan laporan dan alat pemantauan yang tersedia untuk memeriksa penggunaan Anda saat ini. Ini membantu Anda menentukan tingkat RPU Anda dan berapa banyak RPUs yang sesuai untuk dipesan.

Untuk membeli reservasi, selesaikan langkah-langkah berikut:

1. Pilih Beli reservasi tanpa server.

Reservation overview [Info](#)
To view your reservations utilization and coverage, see the [Cost Explorer](#).

Total reservations
1 RPUs

Expiring (next 30 days)
0

Serverless reservations (1) [Info](#)

Any Status ▾ View recommendations Purchase Serverless reservations

2. Jalan-jalan muncul, yang memiliki serangkaian pilihan. Masukkan tingkat RPU reservasi Tanpa Server untuk memesan. Jika Anda tidak yakin seperti apa level ini, Anda dapat menggunakan alat yang dijelaskan lebih lanjut di bagian ini.

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

RPU

The value must range from 1 to any number.

3. Tetapkan jenis pembayaran. Anda dapat memilih untuk membayar di muka untuk reservasi Anda RPU, atau Anda dapat membayar setiap bulan. Jika Anda memilih untuk membayar di muka, Anda mendapatkan diskon yang lebih besar.

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](#).

4. Setelah selesai membuat pilihan, pilih Beli reservasi tanpa server lalu Konfirmasi.

Setelah Anda mengkonfirmasi reservasi, itu muncul dalam daftar reservasi.

Serverless reservations (1) [Info](#) View recommendations Purchase Serverless reservations

Any Status ▾ < 1 > ⚙

Reservation ID	Status ▾	Expiration date ▲	Reservation (RPU)	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

Catatan penggunaan

- Anda tidak dapat mengubah atau menghapus reservasi. Tetapi Anda dapat membuat reservasi tambahan untuk mendapatkan lebih banyak cakupan.
- Redshift Serverless menggunakan dicadangkan RPU untuk beban kerja sebelum menggunakan sesuai permintaan RPU, untuk memastikan penghematan biaya. Jika Anda melebihi jumlah RPU yang telah Anda pesan, Anda mulai mengenakan biaya untuk tambahan tersebut dengan tarif sesuai RPU permintaan Redshift Serverless.
- Kredit gratis untuk Amazon Redshift Tanpa Server tidak berlaku untuk reservasi tanpa server, hanya untuk ditagih sesuai permintaan. RPU

Contoh reservasi tanpa server

Dalam skenario ini, akun AWS pembayar/tertaut Anda memiliki dua grup kerja Amazon Redshift:

- Workgroup 1 memiliki penggunaan steady state, seperti untuk tim intelijen bisnis.
- Workgroup 2 memiliki beban kerja yang tidak dapat diprediksi dengan lonjakan penggunaan, seperti untuk operasi ETL.

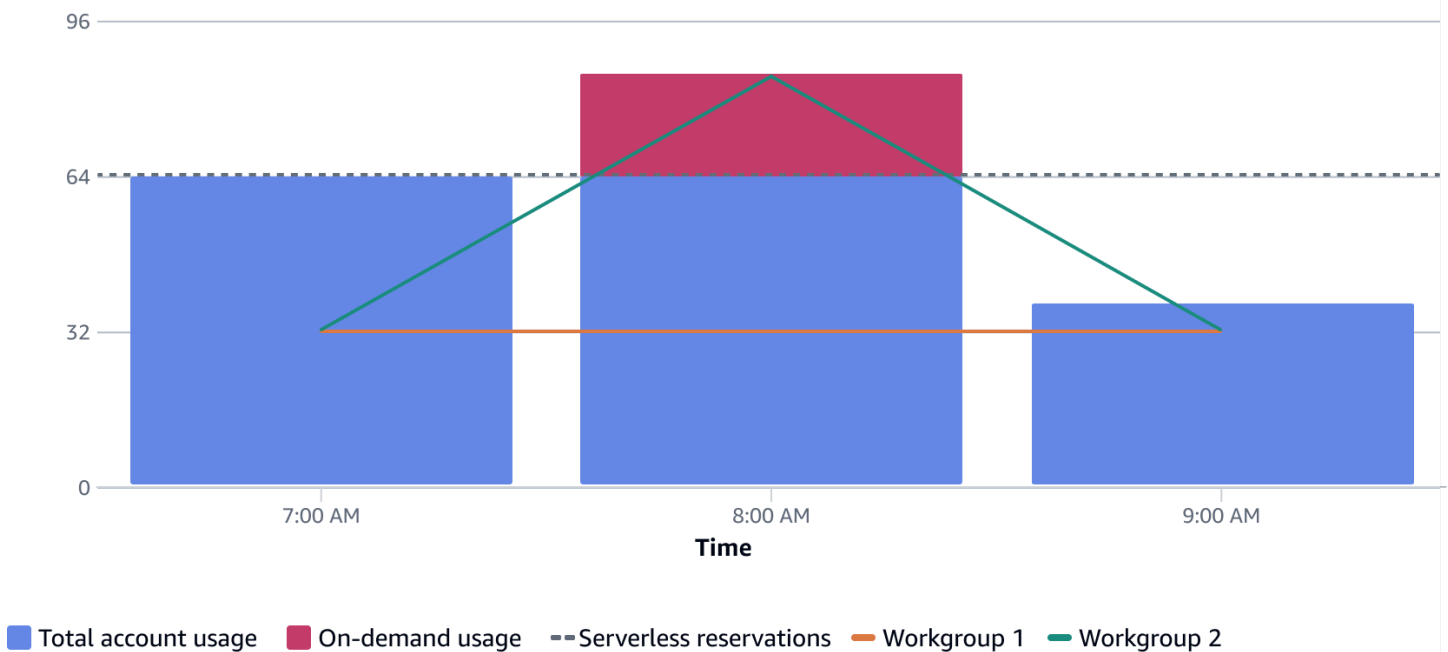
Anda ingin mengoptimalkan biaya untuk kelompok kerja ini, sehingga Anda membeli reservasi tanpa server satu tahun. Berdasarkan data historis, Anda menentukan bahwa kedua kelompok kerja mengkonsumsi 64 RPU pada kondisi mapan. Workgroup 2, bagaimanapun, kadang-kadang meningkat dari 32 RPU menjadi 48 RPU dan turun menjadi 24 RPU untuk waktu yang singkat. Anda menetapkan tingkat RPU reservasi Anda di 64 RPU untuk memulai, yang selaras dengan tren historis. Rincian penagihan per jam adalah sebagai berikut:

- Untuk jam pertama, mirip dengan tren penggunaan historis, kedua kelompok kerja menggunakan 32 RPU untuk total penggunaan akun 64 RPU. Untuk jam ini, semua RPU dikenakan biaya dengan tarif diskon reservasi tanpa server. Ini karena tingkat penggunaan 64 sama dengan RPU reservasi tanpa server 64 RPU.
- Untuk jam kedua, workgroup 1 terus menggunakan 32 RPU. Namun, workgroup 2 melonjak menjadi 48 RPU, dengan total penggunaan akun 80 RPU. Untuk jam ini, 64 RPU dikenakan tarif diskon reservasi tanpa server, dan 16 RPU dikenakan tarif sesuai permintaan Redshift Serverless.

- Untuk jam ketiga, workgroup 1 terus mengkonsumsi 32 RPU dan workgroup 2 berkurang menjadi 8 RPU. Pada jam ini, akun dikenakan tarif reservasi tanpa server 64 RPU, meskipun total akun adalah 40 RPU.

Lihat diagram berikut untuk evolusi penggunaan grup kerja, serta detail penagihan tarif reservasi sesuai permintaan dan tanpa server:

Usage (RPU)



Membeli reservasi tanpa server menggunakan AWS CLI atau Amazon Redshift API

Anda gunakan `create-reservation` untuk membuat reservasi RPU. Berikut ini menunjukkan perintah:

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

Anda mengatur `capacity` ke nomor yang ingin RPU Anda pesan.

Penagihan untuk penyimpanan

Kapasitas penyimpanan primer ditagih sebagai Redshift Managed Storage (RMS). Penyimpanan ditagih oleh GB/bulan. Penagihan penyimpanan terpisah dari penagihan untuk kapasitas komputasi.

Penyimpanan yang digunakan untuk snapshot pengguna ditagih dengan tarif penagihan cadangan standar, tergantung pada tingkat penggunaan Anda.

Biaya transfer data dan biaya pembelajaran mesin (ML) berlaku secara terpisah, sama seperti cluster yang disediakan. Replikasi snapshot dan berbagi data di seluruh AWS Wilayah ditagih dengan tarif transfer yang diuraikan pada halaman harga. Untuk informasi selengkapnya, lihat [harga Amazon Redshift](#).

Memvisualisasikan penggunaan tagihan dengan CloudWatch

MetrikSnapshotStorage, yang melacak penggunaan penyimpanan snapshot, dibuat dan dikirim ke CloudWatch. Untuk informasi selengkapnya CloudWatch, lihat [Apa itu Amazon CloudWatch?](#)

Menggunakan uji coba gratis Amazon Redshift Serverless

Amazon Redshift Serverless menawarkan uji coba gratis. Jika Anda berpartisipasi dalam uji coba gratis, Anda dapat melihat saldo kredit uji coba gratis di konsol Redshift, dan memeriksa penggunaan uji coba gratis di tampilan [sistem SYS_SERVERLESS_USAGE](#). Perhatikan bahwa detail penagihan untuk penggunaan uji coba gratis tidak muncul di konsol penagihan. Anda hanya dapat melihat penggunaan di konsol penagihan setelah uji coba gratis berakhir. [Untuk informasi lebih lanjut tentang uji coba gratis Amazon Redshift Serverless, lihat uji coba gratis Amazon Redshift Serverless.](#)

Catatan penggunaan penagihan

- Penggunaan rekaman - Permintaan atau transaksi hanya diukur dan dicatat setelah transaksi selesai, digulung kembali, atau dihentikan. Misalnya, jika transaksi berjalan selama dua hari, penggunaan RPU dicatat setelah selesai. Anda dapat memantau penggunaan yang sedang berlangsung secara real time dengan melakukan `querysys_serverless_usage`. Pencatatan transaksi dapat mencerminkan variasi penggunaan RPU dan biaya efek untuk jam tertentu dan untuk penggunaan sehari-hari.
- Menulis transaksi eksplisit - Ini penting sebagai praktik terbaik untuk mengakhiri transaksi. Jika Anda tidak mengakhiri atau mengembalikan transaksi terbuka, Amazon Redshift Serverless terus digunakan. RPU. Misalnya, jika Anda menulis eksplisit `BEGIN TRAN`, penting untuk memiliki korespondensi `COMMIT` dan `ROLLBACK` pernyataan.
- Kueri yang dibatalkan - Jika Anda menjalankan kueri dan membatalkannya sebelum selesai, Anda masih ditagih untuk waktu kueri dijalankan.
- Penskalaan - Instans Amazon Redshift Tanpa Server dapat memulai penskalaan untuk menangani periode pemuatan yang lebih tinggi, untuk mempertahankan kinerja yang konsisten. Penagihan

Amazon Redshift Tanpa Server Anda mencakup komputasi dasar dan kapasitas skala pada tingkat RPU yang sama.

- **Penskalaan** - Amazon Redshift Serverless meningkatkan skala dari kapasitas RPU dasarnya untuk menangani periode beban yang lebih tinggi. Dalam beberapa kasus, kapasitas RPU dapat tetap pada pengaturan yang lebih tinggi untuk periode setelah beban kueri turun. Kami menyarankan Anda mengatur jam RPU maksimum di konsol untuk menjaga terhadap biaya yang tidak terduga.
- **Tabel sistem** - Saat Anda menanyakan tabel sistem, waktu kueri ditagih.
- **Redshift Spectrum** - Jika Anda memiliki Amazon Redshift Serverless, dan Anda menjalankan kueri, tidak ada biaya terpisah untuk kueri data-lake. Untuk kueri pada data yang disimpan di Amazon S3, biayanya sama, berdasarkan waktu transaksi, seperti kueri pada data lokal.
- **Kueri federasi** - Kueri federasi dibebankan dalam hal penggunaan RPUs selama interval waktu tertentu, dengan cara yang sama seperti kueri pada gudang data atau data lake.
- **Penyimpanan** - Penyimpanan ditagih secara terpisah, berdasarkan GB/bulan.
- **Biaya minimum** - Biaya minimum adalah untuk 60 detik penggunaan sumber daya, diukur per detik.
- **Penagihan snapshot** - Penagihan snapshot tidak berubah. Ini dibebankan sesuai dengan penyimpanan, ditagih dengan tarif GB/bulan. Anda dapat mengembalikan gudang data Anda ke titik-titik tertentu dalam 24 jam terakhir dengan perincian 30 menit, gratis. Untuk informasi selengkapnya, lihat [harga Amazon Redshift](#).
- **Pengoptimalan otomatis berjalan menggunakan sumber daya komputasi tambahan** - Amazon Redshift Serverless biasanya menjalankan operasi pengoptimalan otomatis bersama kueri pengguna. Operasi ini dikenal sebagai otonom, dan Anda tidak dikenakan biaya untuk itu.

Jika Anda mengaktifkan pengalokasian sumber daya komputasi tambahan, Amazon Redshift akan menjalankan otonom bila diperlukan bahkan dalam periode aktivitas pengguna yang tinggi. Dalam kasus seperti itu, Anda dapat ditagih untuk waktu yang dihabiskan menjalankan otonom. Untuk informasi selengkapnya, lihat [Mengalokasikan sumber daya komputasi tambahan untuk pengoptimalan database otomatis](#) di Panduan Pengembang Database Amazon Redshift.

Praktik terbaik Amazon Redshift Tanpa Server untuk menjaga penagihan tetap dapat diprediksi

Berikut ini adalah praktik terbaik dan pengaturan bawaan yang membantu menjaga penagihan Anda tetap konsisten.

- Pastikan untuk mengakhiri setiap transaksi. Ketika Anda menggunakan BEGIN untuk memulai transaksi, penting untuk END itu juga.
- Gunakan penanganan kesalahan praktik terbaik untuk merespons kesalahan dengan anggun dan mengakhiri setiap transaksi. Meminimalkan transaksi terbuka membantu menghindari penggunaan RPU yang tidak perlu.
- Gunakan SESSION TIMEOUT untuk membantu mengakhiri transaksi terbuka dan sesi idle. Ini menyebabkan sesi apa pun tetap menganggur atau tidak aktif selama lebih dari 3600 detik (1 jam) hingga time out. Ini menyebabkan setiap transaksi tetap terbuka dan tidak aktif selama lebih dari 21600 detik (6 jam) hingga time out. Pengaturan batas waktu ini dapat diubah secara eksplisit untuk pengguna tertentu, seperti saat Anda ingin membuka sesi untuk kueri yang berjalan lama. Topik [CREATE USER](#) menunjukkan cara menyesuaikan SESSION TIMEOUT untuk pengguna.
- Dalam kebanyakan kasus, kami menyarankan Anda untuk tidak memperpanjang SESSION TIMEOUT nilainya, kecuali jika Anda memiliki kasus penggunaan yang mengharuskannya secara khusus. Jika sesi tetap menganggur, dengan transaksi terbuka, itu dapat mengakibatkan kasus di mana RPUs digunakan sampai sesi ditutup. Ini akan menghasilkan biaya yang tidak perlu.
- Amazon Redshift Serverless memiliki waktu maksimum 86.399 detik (24 jam) untuk kueri yang berjalan. Periode maksimum tidak aktif untuk transaksi terbuka adalah enam jam sebelum Amazon Redshift Serverless mengakhiri sesi yang terkait dengan transaksi. Untuk informasi selengkapnya, lihat [Kuota untuk objek Amazon Redshift Tanpa Server](#).

Amazon Redshift Penagihan tanpa server dengan penyatuan koneksi

Amazon Redshift Serverless memperlakukan semua kueri yang masuk sebagai aktivitas pengguna yang dapat ditagih, termasuk kueri pemeriksaan kesehatan ringan yang dikirim oleh kumpulan koneksi. Perilaku ini berlaku terlepas dari apakah kueri berasal dari aplikasi, JDBC/ODBC driver, atau kerangka kerja penyatuan koneksi. Setiap kueri pemeriksaan kesehatan memicu penggunaan komputasi, dan biaya dikeluarkan terlepas dari tujuan atau asal kueri. Akibatnya, mempertahankan kolam koneksi terbuka dapat menghasilkan biaya bahkan ketika tidak ada beban kerja pengguna yang sebenarnya berjalan.

Penggabungan koneksi mempertahankan kumpulan koneksi persisten antara aplikasi dan titik akhir Amazon Redshift Tanpa Server. Untuk memastikan koneksi ini tetap sehat dan tersedia, mekanisme penyatuan sering mengirim kueri ringan atau kosong (misalnya, `SELECT 1`) secara berkala. Kueri otomatis ini memverifikasi status koneksi.

Saat Anda menggunakan penyatuan koneksi, pertimbangkan praktik terbaik ini untuk meminimalkan biaya yang tidak diinginkan:

- Sesuaikan frekuensi pemeriksaan kesehatan dengan meninjau dan mengoptimalkan frekuensi pemeriksaan kesehatan atau kueri keep-alive dalam konfigurasi penyatuan koneksi Anda.
- Optimalkan pengaturan sistem idle dengan mengonfigurasi penyatuan koneksi untuk meminimalkan churn koneksi yang tidak perlu atau aktivitas kueri latar belakang selama waktu idle sistem.
- Terapkan penyatuan tingkat aplikasi atau manajemen siklus hidup koneksi yang ditingkatkan jika dapat mengurangi biaya overhead.
- Nonaktifkan kueri detak jantung atau validasi jika konfigurasi penyatuan koneksi Anda mengizinkannya. Periksa parameter string koneksi spesifik Anda atau file konfigurasi untuk menyesuaikan pengaturan ini.
- Menyesuaikan pengaturan keepalive TCP: Jika Anda tidak dapat menonaktifkan mekanisme detak jantung internal pengemudi, sesuaikan pengaturan keepalive Transmission Control Protocol (TCP) di sistem operasi atau tingkat aplikasi untuk mengatasi masalah batas waktu koneksi. Lihat dokumentasi sistem operasi, JDBC/ODBC driver, atau kumpulan koneksi Anda untuk detailnya.
- Optimalkan pengumpulan koneksi database: Konfigurasi kumpulan koneksi Anda (HiKaricp, Apache Database Connection Pool) untuk mengelola koneksi dan meminimalkan overhead koneksi. Fokus pada parameter seperti koneksi maksimum, batas waktu idle, dan kueri validasi (jika perlu). Pengoptimalan ini membantu menyelaraskan penggunaan komputasi Amazon Redshift Tanpa Server dengan permintaan beban kerja aktual, yang berpotensi mengurangi biaya.

Optimalisasi biaya untuk Amazon Redshift Serverless dengan Zero-ETL

Untuk mengoptimalkan biaya saat menjalankan integrasi nol-ETL di Amazon Redshift Tanpa Server, Anda dapat mengukur lingkungan dengan benar dan menyesuaikan pengaturan penyegaran berdasarkan kebutuhan beban kerja. Pertimbangkan untuk melakukan penyesuaian berikut:

- Gunakan kapasitas RPU dasar yang lebih rendah dari 8 RPU jika tersedia untuk beban kerja.
- Konfigurasi REFRESH_INTERVAL instance Redshift target Anda untuk menyeimbangkan kesegaran dengan biaya. Interval yang lebih pendek memastikan pembaruan mendekati waktu nyata tetapi menaikkan biaya komputasi. Interval yang lebih lama (5 menit atau lebih) mengurangi biaya untuk beban kerja di mana kesegaran langsung tidak penting, seperti pelaporan atau analisis historis. [Untuk mengedit target Redshift REFRESH_INTERVAL, lihat klausa interval refresh dalam deskripsi ALTER DATABASE.](#)

- Maksimalkan pemanfaatan lingkungan Amazon Redshift Tanpa Server Anda dengan menjalankan beban kerja analitik secara bersamaan saat data nol-ETL sedang dicerna. Ini memastikan bahwa kapasitas komputasi secara aktif melayani berbagai tujuan bisnis.

Menghubungkan ke Amazon Redshift Tanpa Server

Setelah menyiapkan instans Amazon Redshift Tanpa Server, Anda dapat menghubungkannya dengan berbagai metode, yang diuraikan di bawah ini. Jika Anda memiliki beberapa tim atau proyek dan ingin mengelola biaya secara terpisah, Anda dapat menggunakan terpisah Akun AWS.

Untuk daftar Wilayah AWS tempat Amazon Redshift Serverless tersedia, lihat titik akhir yang terdaftar untuk [Redshift](#) Serverless API di. Referensi Umum Amazon Web Services

Amazon Redshift Tanpa Server terhubung ke lingkungan tanpa server di lingkungan Anda saat ini. Akun AWS AWS Region Amazon Redshift Serverless berjalan dalam VPC dalam rentang port 5431-5455 dan 8191-8215. Defaultnya adalah 5439. Saat ini, Anda hanya dapat mengubah port dengan operasi API `UpdateWorkgroup` dan AWS CLI operasi `update-workgroup`.

Menghubungkan ke Amazon Redshift Tanpa Server

Anda dapat terhubung ke database (bernama `dev`) di Amazon Redshift Serverless dengan sintaks berikut.

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

Misalnya, string koneksi berikut menentukan Region `us-east-1`.

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

Menghubungkan ke Amazon Redshift Tanpa Server melalui driver JDBC

Anda dapat menggunakan salah satu metode berikut untuk terhubung ke Amazon Redshift Tanpa Server dengan klien SQL pilihan Anda menggunakan driver JDBC versi 2.x yang disediakan Amazon RedShift.

Untuk terhubung dengan kredensi login untuk otentikasi database menggunakan driver JDBC versi 2.x, gunakan sintaks berikut. Nomor port opsional; jika tidak disertakan, Amazon Redshift Serverless

default ke nomor port 5439. Anda dapat mengubah ke port lain dari rentang port 5431-5455 atau 8191-8215. Untuk mengubah port default untuk titik akhir tanpa server, gunakan API Amazon AWS CLI Redshift dan Amazon.

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

Misalnya, string koneksi berikut menentukan default workgroup, ID akun 123456789012, dan Region us-east-2.

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

Untuk terhubung dengan IAM menggunakan driver JDBC versi 2.x, gunakan sintaks berikut. Nomor port opsional; jika tidak disertakan, Amazon Redshift Serverless default ke nomor port 5439. Anda dapat mengubah ke port lain dari rentang port 5431-5455 atau 8191-8215. Untuk mengubah port default untuk titik akhir tanpa server, gunakan API Amazon AWS CLI Redshift dan Amazon.

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

Misalnya, string koneksi berikut menentukan default workgroup, ID akun 123456789012, dan Region us-east-2.

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

Untuk ODBC, gunakan sintaks berikut.

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

Jika Anda menggunakan versi driver JDBC sebelum 2.1.0.9 dan terhubung dengan IAM, Anda harus menggunakan sintaks berikut.

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

Misalnya, string koneksi berikut menentukan workgroup default dan AWS Region us-east-1.

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

Untuk informasi lebih lanjut tentang driver, lihat [Mengonfigurasi koneksi di Amazon Redshift](#).

Menemukan string koneksi JDBC dan ODBC Anda

Untuk terhubung ke workgroup Anda dengan alat klien SQL Anda, Anda harus memiliki string koneksi JDBC atau ODBC. Anda dapat menemukan string koneksi di konsol Amazon Redshift Tanpa Server, di halaman detail grup kerja.

Untuk menemukan string koneksi untuk workgroup

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Redshift Serverless.
3. Pada menu navigasi, pilih konfigurasi Workgroup, lalu pilih nama workgroup dari daftar untuk membuka detailnya.
4. String koneksi URL JDBC dan URL ODBC tersedia, bersama dengan detail tambahan, di bagian Informasi umum. Setiap string didasarkan pada AWS Wilayah tempat workgroup berjalan. Pilih ikon di sebelah string koneksi yang sesuai untuk menyalin string koneksi.

Menghubungkan ke Amazon Redshift Tanpa Server dengan API Data

Anda juga dapat menggunakan Amazon Redshift Data API untuk terhubung ke Amazon Redshift Tanpa Server. Gunakan workgroup-name parameter alih-alih cluster-identifier parameter dalam AWS CLI panggilan Anda.

Untuk informasi selengkapnya tentang API Data, lihat [Menggunakan API Data Amazon Redshift](#). Misalnya kode yang memanggil Data API dengan Python dan contoh lainnya, lihat [Memulai dengan Redshift Data API](#) dan lihat di folder dan di quick-start. use-cases GitHub

Menghubungkan dengan SSL ke Amazon Redshift Tanpa Server

Mengonfigurasi koneksi aman ke Amazon Redshift Tanpa Server

Untuk mendukung koneksi SSL, Redshift Serverless membuat dan menginstal sertifikat SSL ([ACM](#)) [AWS Certificate Manager yang dikeluarkan untuk](#) setiap workgroup. Sertifikat ACM dipercaya

publik oleh sebagian besar sistem operasi, browser web, dan klien. Anda mungkin perlu mengunduh bundel sertifikat jika klien atau aplikasi SQL Anda terhubung ke Redshift Tanpa Server menggunakan SSL dengan opsi koneksi disetel `sslmode ke,,` atau. `require verify-ca verify-full` Jika klien Anda membutuhkan sertifikat, Redshift Serverless menyediakan sertifikat bundel sebagai berikut:

- Unduh bundel dari <https://s3.amazonaws.com/redshift-downloads/amazon-trust-ca-bundle.crt>.
 - Nomor MD5 checksum yang diharapkan adalah 418dea9b6d5d5de7a8f1ac42e164cdcf.
 - Nomor checksum sha256 adalah 36dba8e4b8041cd14b9d60158893963301bcbb92e1c456847784de2acb5bd550.

Jangan gunakan bundel sertifikat sebelumnya yang berada di `https://s3.amazonaws.com/redshift-downloads/redshift-ca-bundle.crt`.

- Di AWS Region Tiongkok, unduh bundel <https://s3.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/amazon-trust-ca-bundle.crt>.
 - Nomor MD5 checksum yang diharapkan adalah 418dea9b6d5d5de7a8f1ac42e164cdcf.
 - Nomor checksum sha256 adalah 36dba8e4b8041cd14b9d60158893963301bcbb92e1c456847784de2acb5bd550.

Jangan gunakan bundel sertifikat sebelumnya yang berada di `https://s3.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/redshift-ca-bundle.crt` dan `https://s3.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/redshift-ssl-ca-cert.pem`

Important

Redshift Serverless telah mengubah cara sertifikat SSL dikelola. Anda mungkin perlu memperbarui sertifikat CA root kepercayaan Anda saat ini untuk terus terhubung ke grup kerja menggunakan SSL. Untuk informasi selengkapnya tentang sertifikat ACM untuk koneksi SSL, lihat. [Transisi ke sertifikat ACM untuk koneksi SSL](#)

Secara default, database workgroup menerima koneksi apakah menggunakan SSL atau tidak.

Untuk membuat workgroup baru yang hanya menerima koneksi SSL, gunakan `create-workgroup` perintah dan atur parameternya `require_ssl.true` Untuk menggunakan contoh berikut, ganti *yourNamespaceName* dengan nama namespace Anda dan ganti *yourWorkgroupName* dengan nama workgroup Anda.

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

Untuk memperbarui workgroup yang ada agar hanya menerima koneksi SSL, gunakan `update-workgroup` perintah dan atur `require_ssl` parameternya ke `true`. Perhatikan bahwa Redshift Serverless akan memulai ulang workgroup Anda saat Anda memperbarui parameter `require_ssl`. Untuk menggunakan contoh berikut, ganti *yourWorkgroupName* dengan nama workgroup Anda.

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

Amazon Redshift mendukung protokol perjanjian kunci Elliptic Curve Diffie—Hellman Ephemeral (ECDHE). Dengan ECDHE, klien dan server masing-masing memiliki elliptic curve public-private key pair yang digunakan untuk membangun rahasia bersama melalui saluran yang tidak aman. Anda tidak perlu mengonfigurasi apa pun di Amazon Redshift untuk mengaktifkan ECDHE. Jika Anda terhubung dari alat klien SQL yang menggunakan ECDHE untuk mengenkripsi komunikasi antara klien dan server, Amazon Redshift menggunakan daftar sandi yang disediakan untuk membuat koneksi yang sesuai. [Untuk informasi lebih lanjut, lihat Elliptic curve diffie—hellman di Wikipedia dan Ciphers di situs web OpenSSL.](#)

Mengonfigurasi koneksi SSL yang sesuai dengan FIPS ke Amazon Redshift Tanpa Server

Untuk membuat workgroup baru yang menggunakan koneksi SSL yang sesuai dengan FIPS, gunakan `create-workgroup` perintah dan atur parameternya `use_fips_ssl` `require_ssl` `true`. Untuk menggunakan contoh berikut, ganti *yourNamespaceName* dengan nama namespace Anda dan ganti *yourWorkgroupName* dengan nama workgroup Anda.

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

Untuk memperbarui workgroup yang ada agar menggunakan koneksi SSL yang sesuai dengan FIPS, gunakan `update-workgroup` perintah dan atur parameternya `use_fips_ssl` `require_ssl` `true`. Perhatikan bahwa Redshift Serverless akan memulai ulang workgroup Anda.

saat Anda memperbarui parameter. `use_fips_ssl` Untuk menggunakan contoh berikut, ganti *yourWorkgroupName* dengan nama workgroup Anda.

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

[Untuk informasi selengkapnya tentang mengonfigurasi Redshift Tanpa Server untuk menggunakan koneksi yang sesuai dengan FIPS, lihat `use\_fips\_ssl` di Panduan Pengembang Database Amazon Redshift.](#)

Menghubungkan ke Amazon Redshift Tanpa Server dari titik akhir VPC yang dikelola Amazon Redshift

Menghubungkan ke Amazon Redshift Tanpa Server dari titik akhir VPC lainnya

[Untuk informasi tentang menyiapkan atau mengonfigurasi titik akhir VPC terkelola untuk grup kerja Amazon Redshift Tanpa Server, lihat *Bekerja dengan titik akhir VPC yang dikelola RedShift*.](#)

Menghubungkan ke Amazon Redshift Tanpa Server dari titik akhir VPC antarmuka ()AWS PrivateLink

Untuk informasi tentang menghubungkan ke Amazon Redshift Serverless dari antarmuka VPC endpoint (), lihat *AWS PrivateLink* [Titik akhir VPC antarmuka](#)

Menghubungkan ke Amazon Redshift Tanpa Server dari titik akhir VPC Redshift di akun lain

Menghubungkan ke Amazon Redshift Tanpa Server dari titik akhir VPC silang

Amazon Redshift Serverless disediakan dalam VPC. Anda dapat memberikan akses ke VPC di akun lain untuk mengakses Amazon Redshift Tanpa Server di akun Anda. Ini mirip dengan koneksi dari titik akhir VPC yang dikelola, tetapi dalam hal ini koneksi berasal, misalnya, dari klien database di akun lain. Ada beberapa operasi yang dapat Anda lakukan:

- Pemilik database dapat memberikan akses ke VPC yang berisi Amazon Redshift Tanpa Server ke akun lain di Wilayah yang sama.

- Pemilik database dapat mencabut akses Amazon Redshift Tanpa Server.

Manfaat utama dari akses lintas akun adalah memungkinkan kolaborasi basis data yang lebih mudah. Pengguna tidak harus disediakan di akun yang berisi database untuk mengaksesnya, yang mengurangi langkah konfigurasi dan menghemat waktu.

Izin yang diperlukan untuk memberikan akses ke VPC di akun lain

Untuk memberikan akses atau mengubah akses yang diizinkan, pemberi memerlukan kebijakan izin yang ditetapkan dengan izin berikut:

- redshift-tanpa server: PutResourcePolicy
- redshift-tanpa server: GetResourcePolicy
- redshift-tanpa server: DeleteResourcePolicy
- EC2: CreateVpcEndpoint
- EC2: ModifyVpcEndpoint

Anda mungkin memerlukan izin lain yang ditentukan dalam kebijakan AWS AmazonRedshiftFullAccesssterkelola. Untuk informasi selengkapnya, lihat [Memberikan izin ke Amazon Redshift Tanpa Server](#).

Penerima hibah memerlukan kebijakan izin yang ditetapkan dengan izin berikut:

- redshift-tanpa server: ListWorkgroups
- redshift-tanpa server: CreateEndpointAccess
- redshift-tanpa server: UpdateEndpointAccess
- redshift-tanpa server: GetEndpointAccess
- redshift-tanpa server: ListEndpointAccess
- redshift-tanpa server: DeleteEndpointAccess

Sebagai praktik terbaik, sebaiknya lampirkan kebijakan izin ke peran IAM dan kemudian menetapkan ke pengguna dan grup sesuai kebutuhan. Untuk informasi selengkapnya, lihat [Identitas dan manajemen akses di Amazon Redshift](#).

Ini adalah contoh kebijakan sumber daya yang digunakan untuk mengonfigurasi akses lintas-VPC:

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"
          ]
        }
      }
    }
  ]
}
```

Prosedur yang mengikuti di bagian ini mengasumsikan bahwa pengguna yang melakukannya memiliki izin yang ditetapkan yang sesuai, misalnya, melalui peran IAM yang ditetapkan yang memiliki izin yang terdaftar. Prosedur juga mengasumsikan bahwa workgroup memiliki peran IAM yang dilampirkan dengan izin sumber daya yang sesuai.

Memberikan akses VPC ke akun lain, menggunakan konsol

Prosedur ini menunjukkan langkah-langkah untuk mengkonfigurasi akses database ketika Anda pemilik database, dan Anda ingin memberikan akses ke sana.

Memberikan akses dari akun pemilik

1. Di properti untuk workgroup Amazon Redshift Tanpa Server, pada tab Akses data, ada daftar yang disebut Akun yang diberikan. Ini menunjukkan akun dan VPCs diberikan akses ke workgroup. Temukan daftar dan pilih Berikan akses untuk menambahkan akun ke daftar.
2. Sebuah jendela muncul di mana Anda dapat menambahkan informasi penerima hibah. Masukkan ID AWS akun, yang merupakan ID 12 digit dari akun yang ingin Anda berikan akses.
3. Berikan akses ke semua VPCs untuk penerima hibah, atau khusus. VPCs Jika Anda memberikan akses hanya untuk spesifik VPCs, Anda dapat menambahkan IDs untuk ini dengan memasukkan masing-masing dan memilih Tambahkan VPC.
4. Simpan perubahan saat Anda selesai.

Saat Anda menyimpan perubahan, akun akan muncul di daftar akun yang Diberikan. Entri menunjukkan ID Akun dan daftar akses yang VPCs diberikan.

Pemilik database juga dapat mencabut akses ke akun. Pemilik dapat mencabut akses kapan saja.

Mencabut akses ke akun

1. Anda dapat mulai dari daftar akun yang diberikan. Pertama, pilih satu atau beberapa akun.
2. Pilih Cabut akses.

Setelah akses diberikan, administrator database untuk penerima hibah dapat memeriksa konsol untuk menentukan apakah mereka memiliki akses.

Menggunakan konsol untuk mengonfirmasi bahwa akses diberikan bagi Anda untuk mengakses akun lain

1. Di properti workgroup Amazon Redshift Tanpa Server, pada tab Akses data, ada daftar yang disebut Akun resmi. Ini menunjukkan akun yang dapat diakses dari workgroup ini. Penerima hibah tidak dapat menggunakan URL titik akhir grup kerja untuk mengakses grup kerja secara langsung. Untuk mengakses workgroup, Anda sebagai penerima hibah pergi ke bagian endpoint dan memilih buat endpoint.

2. Kemudian, sebagai penerima hibah, Anda memberikan nama titik akhir dan VPC untuk mengakses workgroup.
3. Setelah endpoint berhasil dibuat, itu muncul di bagian endpoint dan ada URL endpoint untuk itu. Anda dapat menggunakan URL endpoint ini untuk mengakses workgroup.

Memberikan akses ke akun lain, menggunakan perintah CLI

Akun yang memberikan akses harus terlebih dahulu memberikan akses ke akun lain untuk terhubung dengan menggunakan `put-resource-policy`. Pemilik database dapat memanggil `put-resource-policy` untuk mengotorisasi akun lain untuk membuat koneksi ke workgroup. Akun penerima hibah kemudian dapat digunakan `create-endpoint-authorization` untuk membuat koneksi ke grup kerja melalui yang diizinkan. VPCs

Berikut ini menunjukkan properti untuk `put-resource-policy`, yang dapat Anda panggil untuk memungkinkan akses ke akun dan VPC tertentu.

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

Setelah memanggil perintah, Anda dapat memanggil `get-resource-policy`, menentukan `resource-arn` untuk melihat akun mana dan VPCs diizinkan untuk mengakses sumber daya.

Panggilan berikut dapat dilakukan oleh penerima hibah. Ini menunjukkan informasi tentang akses yang diberikan. Secara khusus, ia mengembalikan daftar yang berisi akses yang VPCs diberikan.

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

Tujuannya adalah agar penerima hibah mendapatkan informasi dari akun pemberi tentang otorisasi titik akhir. `owner-account` Ini adalah akun berbagi. Ketika Anda menjalankan ini, ia mengembalikan `CrossAccountVpcs` untuk setiap workgroup, yang merupakan daftar yang diizinkan VPCs. Sebagai referensi, berikut ini menunjukkan semua properti yang tersedia untuk workgroup:

```
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

Sebagai pengingat, [relokasi cluster](#) bukanlah prasyarat untuk mengonfigurasi fitur jaringan Redshift tambahan. Anda juga tidak perlu mengaktifkannya untuk mengaktifkan yang berikut:

- Menghubungkan dari VPC lintas akun atau lintas wilayah ke Redshift — Anda dapat terhubung dari satu AWS virtual private cloud (VPC) ke yang lain yang berisi database Redshift, seperti yang dijelaskan di bagian ini.
- Menyiapkan nama domain khusus — Anda dapat membuat nama domain khusus, juga dikenal sebagai URL khusus, untuk klaster Amazon Redshift atau grup kerja Amazon Redshift Tanpa Server, untuk membuat nama titik akhir lebih mudah diingat dan sederhana. Untuk informasi selengkapnya, lihat [Menggunakan nama domain khusus untuk koneksi klien](#).

Sumber daya tambahan

Petunjuk untuk mengatur pengaturan lalu lintas jaringan Anda tersedia di [Aksesibilitas publik dengan konfigurasi grup keamanan default atau kustom](#). Ini termasuk kasus penggunaan di mana cluster dapat diakses publik.

Petunjuk untuk mengatur pengaturan lalu lintas jaringan Anda tersedia di [Aksesibilitas pribadi dengan konfigurasi grup keamanan default atau kustom](#). Ini termasuk kasus penggunaan di mana cluster tidak tersedia untuk internet.

Untuk informasi selengkapnya tentang koneksi aman ke Amazon Redshift Tanpa Server, termasuk pemberian izin, otorisasi akses ke layanan tambahan, dan membuat peran IAM, lihat. [Manajemen identitas dan akses di Amazon Redshift Tanpa Server](#)

Mendefinisikan peran database yang akan diberikan kepada pengguna federasi di Amazon Redshift Tanpa Server

Ketika Anda menjadi bagian dari sebuah organisasi, Anda memiliki kumpulan peran terkait. Misalnya, Anda memiliki peran untuk fungsi pekerjaan Anda, seperti programmer dan manajer. Peran Anda menentukan aplikasi dan data mana yang dapat Anda akses. Sebagian besar organisasi menggunakan penyedia identitas, seperti Microsoft Active Directory, untuk menetapkan peran kepada pengguna dan grup. Penggunaan peran untuk mengontrol akses sumber daya telah berkembang, karena organisasi tidak harus melakukan banyak manajemen pengguna individu.

Baru-baru ini, kontrol akses berbasis peran diperkenalkan di Amazon Redshift Serverless. Menggunakan peran database, Anda dapat mengamankan akses ke data dan objek, seperti skema atau tabel, misalnya. Atau Anda dapat menggunakan peran untuk menentukan sekumpulan izin yang ditinggikan, seperti untuk monitor sistem atau administrator database. Namun setelah Anda memberikan izin sumber daya ke peran database, ada langkah tambahan, yaitu menghubungkan peran pengguna dari organisasi ke peran database. Anda dapat menetapkan setiap pengguna ke peran database mereka saat masuk awal dengan menjalankan pernyataan SQL, tetapi itu banyak usaha. Cara yang lebih mudah adalah dengan menentukan peran database yang akan diberikan dan meneruskannya ke Amazon Redshift Tanpa Server. Ini memiliki keuntungan menyederhanakan proses masuk awal.

Anda dapat meneruskan peran ke Amazon Redshift Serverless menggunakan `GetCredentials`. Saat pengguna masuk untuk pertama kalinya ke database Amazon Redshift Tanpa Server, pengguna database terkait dibuat dan dipetakan ke peran database yang cocok. Topik ini merinci mekanisme untuk meneruskan peran ke Amazon Redshift Tanpa Server.

Melewati peran database memiliki beberapa kasus penggunaan utama:

- Ketika pengguna masuk melalui penyedia identitas pihak ketiga, biasanya dengan federasi yang dikonfigurasi, dan meneruskan peran melalui tag sesi.
- Ketika pengguna masuk melalui kredensial masuk IAM, dan peran mereka diteruskan melalui kunci tag dan nilai.

Untuk informasi selengkapnya tentang kontrol akses berbasis peran, lihat [Kontrol akses berbasis peran \(RBAC\)](#).

Mendefinisikan peran database

Sebelum Anda dapat meneruskan peran ke Amazon Redshift Serverless, Anda harus mengonfigurasi peran database dalam database Anda dan memberi mereka izin yang sesuai pada sumber daya database. Misalnya, dalam skenario sederhana, Anda dapat membuat peran database bernama penjualan dan memberikannya akses ke tabel kueri dengan data penjualan. Untuk informasi selengkapnya tentang cara membuat peran database dan memberikan izin, lihat [MEMBUAT PERAN](#) dan [PEMBERIAN](#).

Kasus penggunaan untuk mendefinisikan peran basis data yang akan diberikan kepada pengguna federasi

Bagian ini menguraikan beberapa kasus penggunaan di mana meneruskan peran database ke Amazon Redshift Tanpa Server dapat menyederhanakan akses ke sumber daya database.

Masuk menggunakan penyedia identitas

Kasus penggunaan pertama mengasumsikan bahwa organisasi Anda memiliki identitas pengguna dalam layanan manajemen identitas dan akses. Layanan ini dapat berbasis cloud, misalnya JumpCloud atau Okta, atau lokal, seperti Microsoft Active Directory. Tujuannya adalah untuk secara otomatis memetakan peran pengguna dari penyedia identitas ke peran database Anda ketika mereka masuk ke klien seperti Query editor V2, misalnya, atau dengan klien JDBC. Untuk mengatur ini, Anda harus menyelesaikan beberapa tugas konfigurasi. Sumber daya yang dimaksud meliputi:

1. Konfigurasi integrasi gabungan dengan penyedia identitas (iDP) Anda menggunakan hubungan kepercayaan. Ini adalah prasyarat. Saat Anda mengatur ini, penyedia identitas bertanggung jawab untuk mengautentikasi pengguna melalui pernyataan SAMP dan memberikan kredensi masuk. Untuk informasi selengkapnya, lihat [Mengintegrasikan penyedia solusi SAMP pihak ketiga](#) dengan AWS. Anda juga dapat menemukan informasi lebih lanjut di [akses Federate ke editor kueri Amazon Redshift V2 dengan Layanan Federasi Direktori Aktif \(AD FS\) atau akses masuk tunggal Federasi ke editor kueri Amazon Redshift v2](#) dengan Okta.
2. Pengguna harus memiliki izin kebijakan berikut:
 - `GetCredentials`— Memberikan kredensi untuk otorisasi sementara untuk masuk ke Amazon Redshift Tanpa Server.
 - `sts:AssumeRoleWithSAML` Menyediakan mekanisme untuk mengikat toko identitas perusahaan atau direktori ke akses berbasis peran AWS .
 - `sts:TagSession`— Izin untuk tindakan tag-session, pada prinsipal penyedia identitas.

Dalam hal ini, `AssumeRoleWithSAML` mengembalikan satu set kredensi keamanan untuk pengguna yang telah diautentikasi melalui respons SAMP yang diautentikasi. Operasi ini menyediakan mekanisme untuk mengikat penyimpanan identitas atau direktori ke AWS akses berbasis peran tanpa kredensi khusus pengguna. Untuk pengguna dengan izin `AssumeRoleWithSAML`, penyedia identitas bertanggung jawab untuk mengelola pernyataan SAMP yang digunakan untuk meneruskan informasi peran.

Sebagai praktik terbaik, sebaiknya lampirkan kebijakan izin ke peran IAM dan kemudian menetapkan ke pengguna dan grup sesuai kebutuhan. Untuk informasi selengkapnya, lihat [Identitas dan manajemen akses di Amazon Redshift](#).

3. Anda mengonfigurasi tag `RedshiftDbRoles` dengan nilai peran yang dipisahkan titik dua, dalam format `role1:role2`. Misalnya, `manager:engineer`. Ini dapat diambil dari implementasi tag sesi yang dikonfigurasi di penyedia identitas Anda. Permintaan otentikasi SALL melewati peran secara terprogram. Untuk informasi selengkapnya tentang meneruskan tag sesi, lihat [Melewati tag sesi AWS STS](#).

Dalam kasus di mana Anda meneruskan nama peran yang tidak ada dalam database, itu diabaikan.

Dalam kasus penggunaan ini, ketika pengguna masuk menggunakan identitas federasi, peran mereka diteruskan dalam permintaan otorisasi melalui kunci tag sesi dan nilai. Selanjutnya, setelah otorisasi, `GetCredentials` meneruskan peran ke database. Setelah koneksi berhasil, peran database dipetakan dan pengguna dapat melakukan tugas-tugas database yang sesuai dengan peran mereka. Bagian penting dari operasi adalah bahwa tag `RedshiftDbRoles` sesi diberi peran dalam permintaan otorisasi awal. Untuk informasi selengkapnya tentang meneruskan tag sesi, lihat [Melewati tag sesi menggunakan AssumeRoleWith SAMP](#).

Masuk menggunakan kredensi IAM

Dalam kasus penggunaan kedua, peran dapat diteruskan untuk pengguna dan mereka dapat mengakses aplikasi klien database melalui kredensi IAM.

1. Pengguna yang masuk dalam kasus ini harus diberi izin kebijakan untuk tindakan berikut:
 - `tag:GetResources`— Mengembalikan sumber daya yang ditandai terkait dengan tag tertentu.
 - `tag:GetTagKeys`— Mengembalikan kunci tag yang saat ini digunakan.

Sebagai praktik terbaik, sebaiknya lampirkan kebijakan izin ke peran IAM dan kemudian menentukannya ke pengguna dan grup sesuai kebutuhan. Untuk informasi selengkapnya, lihat [Identitas dan manajemen akses di Amazon Redshift](#).

2. Izinkan izin juga diperlukan untuk mengakses layanan database, seperti Amazon Redshift Tanpa Server.
3. Untuk kasus penggunaan ini, konfigurasi nilai tag untuk peran Anda AWS Identity and Access Management. Anda dapat memilih untuk mengedit tag dan membuat kunci tag yang dipanggil RedshiftDbRoles dengan string nilai tag yang menyertainya yang berisi peran. Misalnya, manajer:insinyur.

Ketika pengguna log in, peran mereka ditambahkan ke permintaan otorisasi dan diteruskan ke database. Hal ini dipetakan ke peran database yang ada.

Sumber daya tambahan

Seperti yang disebutkan dalam kasus penggunaan, Anda dapat mengonfigurasi hubungan kepercayaan antara IDP dan AWS. Untuk informasi selengkapnya, lihat [Mengonfigurasi IDP SAMP 2.0 Anda dengan mengandalkan kepercayaan pihak](#) dan menambahkan klaim.

Manajemen identitas dan akses di Amazon Redshift Tanpa Server

Akses ke Amazon Redshift memerlukan kredensial yang AWS dapat digunakan untuk mengautentikasi permintaan Anda. Kredensial tersebut harus memiliki izin untuk mengakses AWS sumber daya, seperti Amazon Redshift Tanpa Server.

Bagian berikut memberikan detail tentang bagaimana Anda dapat menggunakan AWS Identity and Access Management (IAM) dan Amazon Redshift untuk membantu mengamankan sumber daya Anda dengan mengontrol siapa yang dapat mengaksesnya. Untuk informasi selengkapnya, lihat [Manajemen identitas dan akses di Amazon Redshift](#).

Memberikan izin ke Amazon Redshift Tanpa Server

Untuk mengakses AWS layanan lain, Amazon Redshift Serverless memerlukan izin. Beberapa fitur Amazon Redshift memerlukan Amazon Redshift untuk mengakses layanan AWS lain atas nama Anda. Agar instans Amazon Redshift Tanpa Server Anda bertindak untuk Anda, berikan kredensial keamanan padanya. Metode yang lebih disukai untuk menyediakan kredensial keamanan adalah dengan menentukan peran AWS Identity and Access Management (IAM). Anda juga dapat membuat

peran IAM melalui konsol Amazon Redshift dan mengaturnya sebagai default. Untuk informasi selengkapnya, lihat [Membuat peran IAM sebagai default untuk Amazon Redshift](#).

Untuk mengakses AWS layanan lain, buat peran IAM dengan izin yang sesuai. Anda juga perlu mengaitkan peran tersebut dengan Amazon Redshift Serverless. Selain itu, tentukan Nama Sumber Daya Amazon (ARN) peran saat Anda menjalankan perintah Amazon Redshift atau tentukan kata kunci. default

Ketika mengubah hubungan kepercayaan untuk peran IAM dalam <https://console.aws.amazon.com/iam/>, pastikan bahwa itu berisi `redshift-serverless.amazonaws.com` dan `redshift.amazonaws.com` sebagai nama layanan utama. Untuk informasi tentang cara mengelola peran IAM untuk mengakses AWS layanan lain atas nama Anda, lihat [Mengotorisasi Amazon Redshift untuk AWS mengakses layanan atas nama Anda](#).

Membuat peran IAM sebagai default untuk Amazon Redshift

Saat Anda membuat peran IAM melalui konsol Amazon Redshift, Amazon Redshift secara terprogram membuat peran di dalam Anda. Akun AWS Amazon Redshift juga secara otomatis melampirkan kebijakan AWS terkelola yang ada padanya. Pendekatan ini berarti Anda dapat tetap berada di dalam konsol Amazon Redshift dan tidak perlu beralih ke konsol IAM untuk pembuatan peran.

Peran IAM yang Anda buat melalui konsol untuk klaster Anda memiliki kebijakan `AmazonRedshiftAllCommandsFullAccess` terkelola yang dilampirkan secara otomatis. Peran IAM ini memungkinkan Amazon Redshift untuk menyalin, membongkar, menanyakan, dan menganalisis data AWS untuk sumber daya di akun IAM Anda. Perintah terkait termasuk COPY, UNLOAD, CREATE EXTERNAL FUNCTION, CREATE EXTERNAL TABLE, CREATE EXTERNAL SCHEMA, CREATE MODEL, dan CREATE LIBRARY. Untuk informasi selengkapnya tentang cara membuat peran IAM sebagai default untuk Amazon Redshift, lihat. [Membuat peran IAM sebagai default untuk Amazon Redshift](#)

Untuk mulai membuat peran IAM sebagai default untuk Amazon Redshift, buka Konsol Manajemen AWS, pilih konsol Amazon Redshift, lalu pilih Redshift Serverless di menu. Dari dasbor Tanpa Server Anda dapat membuat grup kerja baru. Langkah-langkah pembuatan memandu Anda memilih peran IAM atau mengonfigurasi peran IAM baru.

Bila Anda memiliki workgroup Amazon Redshift Serverless yang sudah ada dan Anda ingin mengonfigurasi peran IAM untuk itu, buka. Konsol Manajemen AWS Pilih konsol Amazon Redshift, lalu pilih Redshift Serverless. Di konsol Amazon Redshift Tanpa Server, pilih konfigurasi Namespace untuk grup kerja yang ada. Di bawah Keamanan dan enkripsi, Anda dapat mengedit izin.

Menetapkan peran IAM ke namespace

Setiap peran IAM adalah AWS identitas dengan kebijakan izin yang menentukan tindakan apa yang dapat dilakukan setiap peran. AWS Peran ini dimaksudkan untuk diasumsikan oleh siapa saja yang membutuhkannya. Selain itu, setiap namespace adalah kumpulan objek, seperti tabel dan skema, dan pengguna. Saat menggunakan Amazon Redshift Tanpa Server, Anda dapat mengaitkan beberapa peran IAM dengan namespace Anda. Ini membuatnya lebih mudah untuk menyusun izin Anda dengan tepat untuk kumpulan objek database, sehingga peran dapat melakukan tindakan pada data internal dan eksternal. Misalnya, Anda dapat menjalankan COPY perintah dalam database Amazon Redshift untuk mengambil data dari Amazon S3 dan mengisi tabel Redshift.

Anda dapat mengaitkan beberapa peran ke namespace menggunakan konsol, seperti yang dijelaskan sebelumnya di bagian ini. Anda juga dapat menggunakan perintah `APICreateNamespace`, atau perintah CLI `create-namespace`. Dengan perintah API atau CLI, Anda dapat menetapkan peran IAM ke namespace dengan mengisi `IAMRoles` satu atau beberapa peran. Secara khusus, Anda menambahkan ARNs peran tertentu ke koleksi.

Mengelola namespace terkait peran IAM

Di bagian Konsol Manajemen AWS Anda dapat mengelola kebijakan izin untuk peran. AWS Identity and Access Management Anda dapat mengelola peran IAM untuk namespace, menggunakan pengaturan yang tersedia di bawah konfigurasi Namespace. Untuk informasi selengkapnya tentang ruang nama dan penggunaannya di Amazon Redshift Tanpa Server, lihat [Kelompok kerja dan ruang nama](#)

Memulai dengan kredensi IAM untuk Amazon Redshift

Saat Anda masuk ke konsol Amazon Redshift untuk pertama kalinya dan pertama kali mencoba Amazon Redshift Tanpa Server, kami sarankan Anda masuk sebagai pengguna dengan peran IAM terlampir yang memiliki kebijakan yang diperlukan. Setelah Anda mulai membuat instance Amazon Redshift Tanpa Server, Amazon Redshift merekam nama peran IAM yang Anda gunakan untuk masuk. Anda dapat menggunakan kredensial yang sama untuk masuk ke konsol Amazon Redshift dan konsol Amazon Redshift Tanpa Server.

Saat membuat instance Amazon Redshift Tanpa Server, Anda dapat membuat database. Gunakan editor kueri v2 untuk terhubung ke database dengan opsi kredensi sementara.

Untuk menambahkan nama pengguna admin baru dan kata sandi yang tetap ada untuk database, pilih Sesuaikan kredensi pengguna admin dan masukkan nama pengguna admin baru dan kata sandi pengguna admin.

Untuk mulai menggunakan Amazon Redshift Serverless dan membuat workgroup dan namespace di konsol untuk pertama kalinya, gunakan peran IAM dengan kebijakan izin yang dilampirkan. Pastikan bahwa peran ini memiliki izin administrator `arn:aws:iam::aws:policy/AdministratorAccess` atau izin Amazon Redshift lengkap yang `arn:aws:iam::aws:policy/AmazonRedshiftFullAccess` dilampirkan pada kebijakan IAM.

Skenario berikut menguraikan bagaimana kredensial IAM Anda digunakan oleh Amazon Redshift Tanpa Server saat Anda memulai konsol Amazon Redshift Tanpa Server:

- Jika Anda memilih Gunakan pengaturan default — Amazon Redshift Serverless menerjemahkan identitas IAM Anda saat ini ke superuser database. Anda dapat menggunakan identitas IAM yang sama dengan konsol Amazon Redshift Tanpa Server untuk melakukan tindakan pengguna super di database Anda di Amazon Redshift Tanpa Server.
- Jika Anda memilih Sesuaikan setelan tanpa menentukan nama pengguna Admin dan kata sandi Amazon Redshift Tanpa Server, kredensial IAM Anda saat ini akan digunakan sebagai kredensial pengguna admin default Anda.
- Jika Anda memilih Sesuaikan setelan dan tentukan nama pengguna dan kata sandi Admin Amazon Redshift Tanpa Server — Amazon Redshift Tanpa Server menerjemahkan identitas IAM Anda saat ini ke superuser database. Amazon Redshift Serverless juga membuat nama pengguna login jangka panjang dan pasangan kata sandi juga sebagai pengguna super. Anda dapat menggunakan identitas IAM Anda saat ini atau pasangan nama pengguna dan kata sandi yang dibuat untuk masuk ke database Anda sebagai pengguna super.

Mengakses objek database Amazon Redshift Tanpa Server dengan izin peran basis data

Prosedur ini menunjukkan cara memberikan izin untuk menanyakan tabel melalui peran [database Amazon Redshift](#). Peran ditetapkan melalui tag yang dilampirkan ke pengguna di IAM dan diteruskan ke Amazon Redshift saat mereka masuk. Ini adalah penjelasan dengan contoh konsep dalam [Mendefinisikan peran database untuk diberikan kepada pengguna federasi di Amazon Redshift Tanpa Server](#). Manfaat menyelesaikan langkah-langkah ini adalah Anda dapat mengaitkan pengguna dengan peran database dan menghindari pengaturan izin mereka untuk setiap objek database. Ini menyederhanakan mengelola kemampuan pengguna untuk query, memodifikasi, atau menambahkan data ke tabel dan untuk melakukan tindakan lain.

Prosedur ini mengasumsikan Anda telah menyiapkan database Amazon Redshift Tanpa Server dan Anda memiliki kemampuan untuk memberikan izin dalam database. Ini juga mengasumsikan

Anda memiliki izin untuk membuat pengguna IAM di AWS konsol, untuk membuat peran IAM, dan menetapkan izin kebijakan.

1. Buat pengguna IAM, menggunakan konsol IAM. Nanti, Anda akan terhubung ke database dengan pengguna ini.
2. Buat peran database Redshift, menggunakan editor kueri v2 atau klien SQL lainnya. Untuk informasi selengkapnya tentang membuat peran database, lihat [MEMBUAT PERAN](#).

```
CREATE ROLE urban_planning;
```

Kueri tampilan sistem [SVV_ROLES](#) untuk memeriksa apakah peran Anda dibuat. Ini juga mengembalikan peran sistem.

```
SELECT * from SVV_ROLES;
```

3. Berikan peran database yang Anda buat izin untuk memilih dari tabel. (Pengguna IAM yang Anda buat pada akhirnya akan masuk dan memilih catatan dari tabel melalui peran database.) Nama peran dan nama tabel dalam contoh kode berikut adalah sampel. Di sini, izin diberikan untuk memilih dari tabel bernama `cities`.

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

4. Gunakan AWS Identity and Access Management konsol untuk membuat peran IAM. Peran ini memberikan izin untuk menggunakan editor kueri v2. Buat peran IAM baru dan, untuk jenis entitas tepercaya, pilih AWS akun. Kemudian pilih Akun ini. Berikan peran izin kebijakan berikut:
 - AmazonRedshiftReadOnlyAccess
 - tag:GetResources
 - tag:GetTagKeys
 - Semua tindakan untuk `sqlworkbench`, termasuk `sqlworkbench:ListDatabases` dan `sqlworkbench:UpdateConnection`
5. Di konsol IAM, tambahkan tag dengan Kunci `RedshiftDbRoles` ke pengguna IAM yang Anda buat sebelumnya. Nilai tag harus sesuai dengan peran database yang Anda buat pada langkah pertama. Itu ada `urban_planning` di sampel.

Setelah Anda menyelesaikan langkah-langkah ini, tetapkan peran IAM ke pengguna yang Anda buat di konsol IAM. Saat pengguna masuk ke database dengan editor kueri v2, nama peran database

mereka di tag diteruskan ke Amazon Redshift dan dikaitkan dengannya. Dengan demikian, mereka dapat menanyakan tabel yang sesuai dengan peran database. Untuk mengilustrasikan, pengguna dalam sampel ini dapat melakukan query `cities` tabel melalui peran `urban_planning` database.

Memigrasi kluster yang disediakan ke Amazon Redshift Tanpa Server

Anda dapat memigrasikan kluster yang sudah ada ke Amazon Redshift Tanpa Server, memungkinkan penskalaan sumber daya komputasi sesuai permintaan dan otomatis. Memigrasikan kluster yang disediakan ke Amazon Redshift Serverless memungkinkan Anda mengoptimalkan biaya dengan hanya membayar sumber daya yang Anda gunakan dan secara otomatis menskalakan kapasitas berdasarkan tuntutan beban kerja. Kasus penggunaan umum untuk migrasi termasuk menjalankan kueri ad-hoc, pekerjaan pemrosesan data berkala, atau menangani beban kerja yang tidak dapat diprediksi tanpa penyediaan sumber daya yang berlebihan. Lakukan serangkaian tugas berikut untuk memigrasikan kluster Amazon Redshift yang telah disediakan ke opsi penerapan tanpa server.

Membuat snapshot dari kluster yang Anda sediakan

Note

Amazon Redshift secara otomatis mengonversi kunci interleaved menjadi kunci majemuk saat Anda memulihkan snapshot kluster yang disediakan ke namespace tanpa server.

Untuk mentransfer data dari cluster yang disediakan ke Amazon Redshift Serverless, buat snapshot kluster yang disediakan, lalu pulihkan snapshot di Amazon Redshift Serverless.

Note

Sebelum memigrasikan data ke grup kerja tanpa server, pastikan kebutuhan kluster yang disediakan kompatibel dengan jumlah RPU yang Anda pilih di Amazon Redshift Tanpa Server.

Untuk membuat snapshot dari kluster yang telah disediakan

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Clusters, Snapshots, dan kemudian pilih Create snapshot.
3. Masukkan properti definisi snapshot, lalu pilih Buat snapshot. Mungkin perlu beberapa waktu agar snapshot tersedia.

Untuk memulihkan snapshot klaster yang disediakan ke namespace tanpa server:

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Mulai di konsol cluster yang disediakan Amazon Redshift dan navigasikan ke halaman Cluster, Snapshots.
3. Pilih snapshot untuk digunakan.
4. Pilih Pulihkan snapshot, Pulihkan ke namespace tanpa server.
5. Pilih namespace untuk mengembalikan snapshot Anda.
6. Konfirmasikan bahwa Anda ingin memulihkan dari snapshot Anda. Tindakan ini menggantikan semua database di titik akhir tanpa server Anda dengan data dari klaster yang disediakan. Pilih Pulihkan.

[Untuk informasi selengkapnya tentang snapshot klaster yang disediakan, lihat snapshot Amazon Redshift.](#)

Menghubungkan ke Amazon Redshift Tanpa Server menggunakan driver

[Untuk terhubung ke Amazon Redshift Tanpa Server dengan klien SQL pilihan Anda, Anda dapat menggunakan driver JDBC yang disediakan Amazon Redshift versi 2.x.](#) Sebaiknya sambungkan ke Amazon Redshift menggunakan versi terbaru dari driver Amazon Redshift JDBC versi 2.x. Nomor port adalah opsional. Jika Anda tidak menyertakannya, Amazon Redshift Serverless default ke nomor port 5439. Anda dapat mengubah ke port lain dari rentang port 5431-5455 atau 8191-8215. Untuk mengubah port default untuk titik akhir tanpa server, gunakan API Amazon AWS CLI Redshift dan Amazon.

Untuk menemukan titik akhir yang tepat untuk digunakan untuk driver JDBC, ODBC, atau Python, lihat Konfigurasi Workgroup di Amazon Redshift Serverless. Anda juga dapat menggunakan operasi Amazon Redshift Serverless API GetWorkgroup atau AWS CLI operasi get-workgroups untuk mengembalikan informasi tentang workgroup Anda, lalu menyambung.

Menghubungkan menggunakan otentikasi berbasis kata sandi

Untuk membuat sambungan menggunakan driver Amazon Redshift JDBC versi 2.x dengan autentikasi berbasis kata sandi, gunakan sintaks berikut:

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

Untuk membuat sambungan menggunakan konektor Amazon Redshift Python dengan autentikasi berbasis kata sandi, gunakan sintaks berikut:

```
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
```

Untuk membuat sambungan menggunakan driver Amazon Redshift ODBC versi 2.x dengan autentikasi berbasis kata sandi, gunakan sintaks berikut:

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

Menghubungkan menggunakan IAM

Jika Anda lebih suka masuk dengan IAM, gunakan operasi API Tanpa [GetCredentials](#) Server Amazon Redshift.

Untuk menggunakan autentikasi IAM, tambahkan `iam:` ke URL Amazon Redshift JDBC `jdbc:redshift:` berikut, seperti yang ditunjukkan pada contoh berikut.

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

Titik akhir Amazon Redshift Tanpa Server ini tidak mendukung penyesuaian DBUser, dbGroup, atau pembuatan otomatis. Secara default, driver secara otomatis membuat pengguna database saat login. Kemudian menetapkan pengguna ke peran database Amazon Redshift berdasarkan tag yang ditentukan dalam IAM, atau berdasarkan grup yang ditentukan dalam penyedia identitas Anda (iDP).

Pastikan AWS identitas Anda memiliki kebijakan IAM yang benar untuk `redshift-serverless:GetCredentials` tindakan tersebut. Berikut ini adalah contoh kebijakan IAM yang memberikan izin yang benar ke AWS identitas untuk terhubung ke Amazon Redshift Tanpa Server. Untuk informasi selengkapnya tentang izin IAM, lihat [Menambahkan dan menghapus izin identitas IAM](#) di Panduan Pengguna IAM.

JSON

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

Untuk membuat koneksi menggunakan konektor Amazon Redshift Python dengan autentikasi berbasis IAM, gunakan `iam=true` dalam kode Anda, seperti yang ditunjukkan dalam sintaks berikut:

```
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
```

Untuk `IAM credentials`, Anda dapat menggunakan kredensial apa pun, termasuk yang berikut ini:

- AWS konfigurasi profil.
- Kredensi IAM (ID kunci akses, kunci akses rahasia, dan token sesi opsional).
- Federasi penyedia identitas.

Untuk membuat koneksi menggunakan driver Amazon Redshift ODBC versi 2.x dengan autentikasi berbasis IAM dan profil, gunakan sintaks berikut:

```
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;
```

Menghubungkan menggunakan IAM dengan API GetClusterCredentials

Note

Saat menghubungkan ke Amazon Redshift Tanpa Server, kami sarankan Anda menggunakan API. [GetCredentials](#) API ini menawarkan fungsionalitas kontrol akses berbasis peran (RBAC) yang komprehensif serta fitur baru lainnya yang tidak tersedia. GetClusterCredentials Kami mendukung GetClusterCredentials API untuk menyederhanakan transisi dari klaster yang disediakan ke grup kerja tanpa server, tetapi kami sangat menyarankan untuk melakukan migrasi ke penggunaan GetCredentials sesegera mungkin untuk kompatibilitas yang optimal.

Anda dapat membuat koneksi ke Amazon Redshift Serverless menggunakan API.

[GetClusterCredentials](#) Untuk menerapkan metode otentikasi ini, modifikasi klien atau aplikasi Anda dengan memasukkan parameter berikut:

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

Contoh berikut menunjukkan plugin BrowsersAML di ketiga driver. Ini merupakan salah satu dari beberapa pendekatan otentikasi yang tersedia. Contoh dapat dimodifikasi untuk menggunakan metode otentikasi alternatif atau plugin sesuai dengan kebutuhan spesifik Anda.

Izin kebijakan IAM untuk GetClusterCredentials

Berikut ini adalah contoh kebijakan IAM dengan izin yang diperlukan untuk digunakan GetClusterCredentials dengan Amazon Redshift Tanpa Server:

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}"
      ]
    }
  ]
}
```

Untuk membuat sambungan menggunakan driver Amazon Redshift JDBC versi 2.x dengan GetClusterCredentials, gunakan sintaks berikut:

```
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>"
```

Untuk membuat koneksi menggunakan konektor Amazon Redshift Python dengan `getClusterCredentials`, gunakan sintaks berikut:

```
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
```

Untuk membuat sambungan menggunakan driver Amazon Redshift ODBC versi 2.x dengan `getClusterCredentials`, gunakan sintaks berikut:

```
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>
```

Berikut ini adalah contoh konfigurasi ODBC DSN di Windows:

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

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/

Preferred Role:

Listen Port:

Timeout (sec):

Managed VPC URL:

OK Cancel Test

Menggunakan Amazon Redshift Serverless SDK

Jika Anda menulis skrip manajemen apa pun menggunakan Amazon Redshift SDK, Anda harus menggunakan Amazon Redshift Serverless SDK baru untuk mengelola Amazon Redshift Tanpa

Server dan sumber daya terkait. Untuk informasi selengkapnya tentang operasi API yang tersedia, lihat panduan Referensi API [Tanpa Server Amazon Redshift](#).

Kelompok kerja dan ruang nama

Untuk mengisolasi beban kerja dan mengelola sumber daya yang berbeda di Amazon Redshift Tanpa Server, Anda dapat membuat ruang nama dan grup kerja serta mengelola penyimpanan dan menghitung sumber daya secara terpisah.

Namespace adalah kumpulan objek database dan pengguna. Namespace terkait penyimpanan mengelompokkan skema, tabel, pengguna, atau AWS Key Management Service kunci untuk mengenkripsi data. Properti penyimpanan termasuk nama database dan kata sandi pengguna admin, izin, dan enkripsi dan keamanan. Sumber daya lain yang dikelompokkan di bawah ruang nama termasuk datashares, titik pemulihan, dan batas penggunaan. Anda dapat mengonfigurasi properti penyimpanan ini menggunakan konsol Amazon Redshift Serverless, the AWS Command Line Interface, atau Amazon Redshift Serverless untuk sumber daya tertentu. APIs

Workgroup adalah kumpulan sumber daya komputasi. Kelompok kerja terkait komputasi bersama-sama menghitung sumber daya seperti, grup subnet RPU's VPC, dan grup keamanan. Properti untuk workgroup termasuk pengaturan jaringan dan keamanan. Sumber daya lain yang dikelompokkan dalam kelompok kerja termasuk batas akses dan penggunaan. Anda dapat mengonfigurasi properti komputasi ini menggunakan konsol Amazon Redshift Serverless, the, AWS Command Line Interface atau Amazon Redshift Serverless. APIs

Anda dapat membuat satu atau lebih namespace dan workgroup. Setiap namespace hanya dapat memiliki satu workgroup yang terkait dengannya. Sebaliknya, setiap workgroup dapat dikaitkan dengan hanya satu namespace.

Kelompok kerja dan ruang nama menggunakan konsol

Menyiapkan Amazon Redshift Serverless melibatkan berjalan melalui beberapa langkah konfigurasi. Bila Anda mengikuti langkah-langkah untuk mengatur Amazon Redshift Serverless, Anda membuat namespace dan workgroup, dan mengaitkannya satu sama lain. Untuk memulai pengaturan konfigurasi Amazon Redshift Serverless menggunakan konsol Amazon Redshift Serverless, Anda dapat memilih Memulai Amazon Redshift Serverless untuk mengatur Amazon Redshift Serverless dan mulai berinteraksi dengannya. Anda dapat memilih lingkungan dengan pengaturan default, yang membuat penyiapan lebih cepat, atau secara eksplisit mengonfigurasi pengaturan sesuai kebutuhan organisasi Anda. Selama proses ini, Anda menentukan pengaturan untuk workgroup dan namespace Anda.

Setelah Anda mengatur lingkungan, [Properti Workgroup](#) dan [Properti namespace](#) membantu Anda membiasakan diri dengan pengaturan.

Grup kerja dan ruang nama menggunakan API Tanpa Server Amazon AWS Command Line Interface Redshift dan Amazon

Selain menggunakan AWS konsol, Anda juga dapat menggunakan API Tanpa Server Amazon Redshift AWS CLI atau Amazon Redshift untuk berinteraksi dengan grup kerja dan ruang nama. Tabel di bawah ini mencantumkan operasi API dan CLI yang dapat Anda gunakan untuk mengelola snapshot dan titik pemulihan.

Operasi API	Perintah CLI	Deskripsi
CreateNamespace	buat-namespace	Membuat namespace. Secara default, Amazon Redshift Serverless membuat ruang nama dengan AWS Key Management Service kunci default, tetapi Anda dapat menentukan kunci lain untuk mengenkripsi data Anda. Anda juga dapat membuat namespace dengan memulihkan snapshot. Lihat Bekerja dengan snapshot dan titik pemulihan untuk informasi selengkapnya.
UpdateNamespace	perbaruan-namespace	Memperbarui namespace.
GetNamespace	dapatkan-namespace	Mengambil informasi tentang namespace
ListNamespaces	daftar-ruang nama	Mengambil informasi tentang daftar namespace.
DeleteNamespace	hapus-namespace	Menghapus namespace.

Operasi API	Perintah CLI	Deskripsi
CreateWorkgroup	buat-workgroup	Membuat workgroup. Saat membuat workgroup, pastikan Anda memiliki namespace yang sudah ada yang dapat Anda kaitkan dengan workgroup. Saat membuat workgroup, Anda dapat menentukan sumber daya komputasi seperti subnet, grup keamanan, dan RPU.
UpdateWorkgroup	update-workgroup	Memperbarui grup kerja.
GetWorkgroup	dapatkan-kelompok kerja	Mengambil informasi tentang workgroup.
ListWorkgroups	daftar-kelompok kerja	Mengambil informasi tentang daftar kelompok kerja.
DeleteWorkgroup	hapus-workgroup	Menghapus workgroup.

Kelompok kerja

Dengan Amazon Redshift Tanpa Server, Anda dapat membuat dan mengelola grup kerja untuk mengisolasi dan mengontrol sumber daya komputasi untuk beban kerja atau pengguna yang berbeda. Workgroup memungkinkan Anda mengatur opsi konfigurasi seperti batas penskalaan memori dan konkurensi, dan memprioritaskan eksekusi kueri di seluruh beban kerja. Kelompok kerja terkait komputasi bersama-sama menghitung sumber daya seperti dan RPU grup subnet VPC.

Membuat workgroup dengan namespace

Selesaikan langkah-langkah berikut untuk membuat workgroup. Untuk informasi selengkapnya tentang konfigurasi workgroup, lihat [Properti Workgroup](#).

1. Pilih dasbor Tanpa Server. Kemudian pilih Buat workgroup.
2. Masukkan nama workgroup.

3. Pilih jenis alamat IP untuk workgroup. Pilihannya meliputi:
 - IPv4— Dengan opsi ini, AWS sumber daya Anda hanya berkomunikasi melalui protokol IPv4 pengalamatan.
 - Mode dual-stack — Dengan opsi ini, AWS sumber daya Anda dapat berkomunikasi melalui IPv4, IPv6, atau kedua protokol pengalamatan. Selain itu, Anda harus mengaitkan blok IPv6 CIDR dengan VPC dan subnet yang digunakan untuk workgroup Anda di Amazon VPC. Anda dapat menggunakan konsol VPC Amazon untuk membuat VPC Amazon atau memperbarui VPC Amazon yang ada untuk menggunakan pengalamatan. IPv6 Untuk informasi selengkapnya, lihat [IPv6dukungan untuk VPC Anda](#); di Panduan Pengguna Amazon VPC.
4. Pilih Virtual Private Cloud (VPC) untuk Amazon Redshift Tanpa Server. Ini menetapkan workgroup ke jaringan virtual tertentu di lingkungan Anda AWS . Saat menggunakan mode dual-stack, VPC Amazon yang Anda pilih harus mendukung pengalamatan. IPV6 Untuk informasi selengkapnya tentang VPC Amazon, lihat [Ikhtisar VPCs dan subnet](#).
5. Pilih satu atau beberapa grup keamanan VPC. Untuk informasi selengkapnya, lihat [Mengontrol lalu lintas ke sumber daya menggunakan grup keamanan](#).
6. Pilih apakah akan mengaktifkan sumber daya komputasi tambahan untuk pengoptimalan otomatis. Untuk informasi selengkapnya, lihat [Mengalokasikan sumber daya komputasi tambahan untuk pengoptimalan database otomatis](#) di Panduan Pengembang Database Amazon Redshift.
7. Di bawah Subnet, tentukan satu atau beberapa subnet untuk dikaitkan dengan database Anda. Subnet ini terdapat dalam VPC Amazon yang Anda pilih sebelumnya dan harus berada di tiga Availability Zone yang berbeda. Untuk informasi selengkapnya, lihat [Pertimbangan saat menggunakan Amazon Redshift](#) Tanpa Server.
8. Pilih kapasitas RPU dasar yang sesuai dengan kebutuhan Anda.

Pilih namespace

1. Pilih salah satu Buat namespace baru, dan masukkan nama namespace, atau Tambahkan ke namespace yang ada, dan pilih namespace dari daftar drop-down.
2. Untuk nama dan kata sandi Database, tentukan nama database pertama. Anda juga dapat menentukan admin selain admin konsol default Anda, dengan mengedit kredensi pengguna Admin.

3. Untuk Izin, Anda memilih peran IAM Associate untuk mengaitkan peran IAM tertentu dengan namespace dan workgroup Anda. Untuk informasi selengkapnya tentang mengaitkan peran IAM dengan Amazon Redshift, [lihat Identitas dan manajemen akses di Amazon Redshift](#).
4. Anda dapat menyesuaikan pengaturan enkripsi Anda dengan membuat kunci baru atau memilih kunci selain default. Untuk pencatatan Audit, pilih log yang akan diekspor. Setiap jenis log menentukan metadata yang berbeda. Pilih Lanjutkan untuk meninjau pilihan Anda.

Tinjau pilihan workgroup

1. Tinjau pengaturan Anda di bawah Tinjau dan buat. Ini menunjukkan pengaturan yang Anda pilih di langkah sebelumnya.
2. Pilih Simpan.

Setelah Anda membuat workgroup, itu ditambahkan ke daftar Workgroups.

Melihat properti untuk grup kerja

Di Amazon Redshift Tanpa Server, workgroup adalah kumpulan sumber daya komputasi yang tersedia untuk digunakan. Saat memilih Amazon Redshift Tanpa Server, di AWS konsol, Anda dapat memilih konfigurasi Workgroup dari menu navigasi untuk melihat daftar. Anda dapat menggunakan kotak Pencarian untuk menemukan grup kerja yang memenuhi kriteria pencarian Anda. Setiap entri workgroup memiliki beberapa properti yang ditampilkan:

- Workgroup - Nama workgroup. Anda dapat memilihnya untuk melihat dan mengedit properti workgroup.
- Status - Menunjukkan apakah workgroup tersedia.
- Namespace - Namespace yang terkait dengan workgroup. Setiap workgroup dikaitkan dengan satu namespace.
- Tanggal pembuatan - Tanggal (UTC) bahwa workgroup dibuat.
- Tag - Tag yang terkait dengan workgroup.

Selain itu, konfigurasi Workgroup memiliki daftar lain untuk grup kerja terkelola, yaitu grup kerja Amazon Redshift Tanpa Server yang dikelola oleh AWS Glue Untuk informasi selengkapnya tentang grup kerja terkelola, lihat [Kelompok kerja terkelola di Panduan](#) Pengembang Database Amazon Redshift.

Properti Workgroup

Anda dapat membuat daftar kelompok kerja dengan memilih konfigurasi Workgroup di menu sebelah kiri. Kemudian Anda dapat memilih workgroup dari daftar. Beberapa panel menunjukkan properti untuk workgroup. Anda juga dapat melakukan tindakan. Informasi umum menampilkan yang berikut:

- Workgroup - Nama workgroup.
- Namespace - Namespace yang terkait dengan workgroup. Anda dapat memilihnya untuk melihat propertinya. Workgroup dikaitkan dengan satu namespace.
- Tanggal dibuat - Saat workgroup dibuat.
- Status - Menunjukkan jika sumber daya workgroup tersedia. Jika tersedia, Anda dapat terhubung dengan klien ke instans Amazon Redshift Tanpa Server, untuk menanyakan data atau membuat sumber daya basis data, atau Anda dapat terhubung dengan editor kueri v2.
- Endpoint - URL.
- URL JDBC - URL untuk membangun koneksi klien JDBC. Anda dapat menggunakan URL ini untuk terhubung dengan driver JDBC untuk Amazon Redshift. Untuk informasi selengkapnya, lihat [Mengonfigurasi sambungan untuk driver JDBC versi 2.x untuk Amazon Redshift](#).
- URL ODBC - URL untuk membuat koneksi klien ODBC. Ini berisi properti, seperti database dan ID pengguna, dan nilai-nilainya.
- Versi Workgroup dan versi Patch - Amazon Redshift Serverless secara teratur merilis versi dan tambalan baru. Anda dapat menggunakan versi workgroup dan nomor versi Patch untuk melacak pembaruan perangkat lunak ke grup kerja Amazon Redshift Serverless Anda. Untuk informasi selengkapnya tentang perubahan dan fitur di tambalan tertentu, lihat [Versi klaster untuk Amazon Redshift](#).
- Workgroup ARN - Nama Sumber Daya Amazon untuk workgroup.
- Sumber daya komputasi tambahan untuk pengoptimalan otomatis - Apakah Amazon Redshift mengalokasikan sumber daya komputasi tambahan untuk melakukan pengoptimalan otomatis, bahkan selama periode penggunaan berat. Untuk informasi selengkapnya, lihat [Mengalokasikan sumber daya komputasi tambahan untuk pengoptimalan database otomatis](#) di Panduan Pengembang Database Amazon Redshift.

Tab Akses data berisi beberapa panel:

- Jaringan dan keamanan - Anda dapat melihat properti jaringan, seperti pengidentifikasi Virtual private cloud (VPC), daftar grup keamanan VPC, perutean VPC yang Ditingkatkan, jenis alamat

IP, dan pengaturan yang dapat diakses publik. Jika Anda memilih Edit, Anda dapat mengubah pengaturan ini. Selain itu, Anda dapat memilih Aktifkan perutean VPC yang disempurnakan, yang merutekan lalu lintas jaringan antara database tanpa server dan repositori data Anda melalui VPC, untuk meningkatkan privasi dan keamanan. Anda juga dapat memilih Aktifkan Akses Publik, yang membuat database dapat diakses publik dari luar VPC, memungkinkan instance dan perangkat terhubung.

Jenis alamat IP dapat diatur ke mode dual-stack untuk mendukung akses ke workgroup pada keduanya IPv4 secara IPv6 bersamaan. Untuk informasi selengkapnya tentang komunikasi lapisan jaringan Protokol Internet (IP), lihat [Protokol Internet](#) di Wikipedia.

- Titik akhir VPC terkelola Redshift - Anda dapat membuat titik akhir VPC terkelola untuk mengakses Amazon Redshift Tanpa Server dari VPC lain.

Tab Limits memiliki pengaturan untuk mengontrol kapasitas dan batas penggunaan untuk Amazon Redshift Serverless. Ini berisi panel berikut:

- Kapasitas dasar dalam unit pemrosesan Redshift (RPU) - Anda dapat mengatur kapasitas dasar sumber daya komputasi yang digunakan untuk memproses beban kerja Anda. Untuk informasi selengkapnya, lihat [Kapasitas komputasi untuk Amazon Redshift Serverless](#).
- Batas penggunaan - Anda dapat mengatur hingga empat batas untuk sumber daya komputasi maksimum yang dapat digunakan instans Amazon Redshift Tanpa Server dalam jangka waktu tertentu, dan memilih tindakan untuk Amazon Redshift Tanpa Server untuk dilakukan saat mencapai batas tersebut. Misalnya, Anda dapat mengatur workgroup Anda untuk memiliki dua batas, satu dari 500 jam RPU dan satu dari 900 jam RPU. Anda dapat meminta Amazon Redshift Serverless mengirimkan Anda peringatan ketika mencapai batas pertama 500 jam RPU, lalu matikan kueri pengguna ketika mencapai batas kedua 900 jam. Batasan ini membantu mengendalikan biaya Anda dan membuatnya lebih dapat diprediksi.
- Batas kueri - Anda dapat menetapkan batas pada kueri, seperti pengaturan batas waktu. Batasan ini membantu Anda mengoptimalkan biaya dan kinerja.

Tab Tag memiliki panel Tag, yang menampilkan tag apa pun yang Anda buat untuk workgroup Anda. Untuk informasi selengkapnya tentang menandai sumber daya, lihat [Menandai sumber daya di Amazon Redshift Tanpa Server](#).

Properti workgroup terkelola

Anda juga dapat memilih kelompok kerja yang dikelola oleh AWS Glue Data Catalog di bawah daftar Kelompok kerja terkelola.

Kelompok kerja terkelola memiliki properti yang berbeda dari kelompok kerja biasa. Untuk informasi selengkapnya tentang grup kerja terkelola, lihat [Kelompok kerja terkelola di Panduan](#) Pengembang Database Amazon Redshift.

Informasi umum menampilkan yang berikut:

- **Workgroup** - Nama workgroup yang dikelola.
- **Tanggal dibuat** - Tanggal (UTC) bahwa workgroup terkelola dibuat.
- **Katalog ARN** - Nama Sumber Daya Amazon (ARN) untuk grup kerja terkelola di AWS Glue Data Catalog
- **Status** - Menunjukkan jika sumber daya komputasi grup kerja terkelola tersedia. Jika sumber daya tersedia, Anda dapat menyambung ke katalog yang menggunakan grup kerja terkelola dengan klien SQL yang kompatibel dengan Apache Iceberg untuk melakukan kueri data atau membuat sumber daya basis data. Anda juga dapat terhubung ke katalog menggunakan editor kueri Amazon Redshift v2.

Pemantauan kueri dan database berisi grafik kinerja grup kerja terkelola, yang menunjukkan rata-rata waktu berlalu semua kueri dari grup kerja dari waktu ke waktu.

Tab Query history adalah daftar semua query dari workgroup yang dikelola. Rinciannya mencakup informasi seperti pengguna yang menjalankan kueri, mesin klien dari mana kueri berasal, dan ID dan status kueri. Tab Pengguna adalah daftar semua pengguna di workgroup. Tab Metrik kinerja menampilkan berbagai metrik seperti waktu kueri rata-rata, jumlah kueri yang diselesaikan, dan persentase kapasitas penyimpanan yang digunakan.

Menghapus workgroup

Anda dapat menghapus workgroup menggunakan konsol. Sebelum Anda melakukan ini, pastikan bahwa Anda memiliki data Anda dicadangkan dan snapshot di tempat. Sumber daya yang dihapus sebagai bagian dari grup kerja dalam banyak kasus tidak dapat diambil.

Selesaikan langkah-langkah berikut:

1. Pilih Amazon Redshift Tanpa Server, pilih konfigurasi Workgroup dan pilih Hapus instans Amazon Redshift Tanpa Server.
2. Dialog terbuka. Saat Anda memilih untuk menghapus grup kerja, semua batas penggunaan dihapus, semua titik akhir VPC dihapus, dan akses ke titik akhir VPC dihapus.

Ketik hapus dan pilih Hapus untuk mengonfirmasi.

Setelah Anda menyelesaikan langkah-langkahnya, status workgroup adalah Menghapus dan spanduk menunjukkan bahwa workgroup sedang dihapus. Saat proses penghapusan sedang berlangsung, beberapa fitur di bawah dasbor Tanpa Server dinonaktifkan. Tetapi Anda dapat mengonfigurasi cluster yang disediakan di dasbor kluster yang disediakan.

Setelah Anda menghapus workgroup, itu tidak muncul dengan namespace. Anda dapat memilih tombol Create workgroup untuk membuat yang baru.

Anda dapat menghapus workgroup yang ada dan mengaitkan workgroup baru dengan konfigurasi yang berbeda ke namespace yang sama. Saat membuat workgroup baru, pilih kapasitas dasar yang sesuai dengan ukuran data yang terkait dengan namespace.

Anda dapat mengaitkan workgroup dengan namespace yang dibuat dengan kunci yang dikelola pelanggan (CMK). Untuk informasi lebih lanjut tentang AWS KMS, lihat [AWS KMS konsep](#).

Namespace

Di Amazon Redshift Tanpa Server, namespace mendefinisikan wadah logis untuk objek database. Ini dapat menampung tabel, kelompok kerja, dan sumber daya database lainnya. Jika Anda belum membuat workgroup dan namespace, dan Anda sedang mencari petunjuk tentang cara memulai Amazon Redshift Serverless, lihat Menyiapkan Amazon Redshift Serverless untuk pertama [kalinya](#).

Properti namespace

Di Amazon Redshift Tanpa Server, namespace mendefinisikan wadah untuk objek database. Anda dapat memilih konfigurasi Namespace dari daftar navigasi, memilih namespace dari daftar, dan mengedit pengaturannya.

Informasi umum untuk namespace meliputi:

- Namespace - Nama.
- Namespace ID - Pengidentifikasi unik.

- ARN - Sebuah identifier unik yang digunakan untuk menentukan sumber daya di seluruh AWS. Ini berisi properti seperti wilayah dan layanan.
- Status - Status, seperti Tersedia.
- Tanggal dibuat - Tanggal (UTC) bahwa namespace dibuat.
- Penyimpanan yang digunakan - Ruang penyimpanan yang digunakan oleh namespace dan semua objeknya.
- Nama pengguna admin - Akun admin. Ini biasanya akun yang digunakan untuk membuat namespace.
- Nama database - Nama database yang terkandung oleh namespace.
- Jumlah tabel total - Hitungan tabel di semua skema.

Pengaturan dan properti tambahan untuk namespace ada di beberapa tab. Sumber daya yang dimaksud meliputi:

- Workgroup - Menunjukkan workgroup yang terkait dengan namespace.
- Data back up - Pada panel ini, Anda dapat mengkonfigurasi dan membuat snapshot, dan mengkonfigurasi titik pemulihan.
- Keamanan dan enkripsi - Anda dapat mengelola izin peran IAM dan melihat atau mengedit pengaturan keamanan dan enkripsi Anda. Ini termasuk status kunci enkripsi Anda, dan pengaturan untuk mengaktifkan pencatatan audit. Untuk informasi selengkapnya tentang pencatatan audit untuk Amazon Redshift Tanpa Server, lihat Pencatatan [audit untuk Amazon Redshift](#) Tanpa Server.
- Datashares - Menunjukkan datashares. Dengan berbagi data, Anda dapat memberikan akses ke data tanpa perlu menyalin atau memindahkannya. Untuk informasi selengkapnya tentang berbagi data, lihat [Berbagi data di Amazon Redshift](#) Tanpa Server.

Mencari namespace

Dari menu Amazon Redshift, Anda dapat memilih dari daftar Namespaces untuk melihat atau mengedit properti untuk namespace. Informasi di konsol mencakup nama namespace, nama admin, dan properti lainnya.

Pengaturan dan properti namespace ada di beberapa tab. Sumber daya yang dimaksud meliputi:

- Workgroup - Menunjukkan workgroup yang terkait dengan namespace.

- Data back up - Anda dapat mengkonfigurasi dan membuat snapshot, dan mengkonfigurasi titik pemulihan.
- Keamanan dan enkripsi - Anda dapat mengelola izin peran IAM dan melihat atau mengedit pengaturan keamanan dan enkripsi Anda. Ini termasuk status kunci enkripsi Anda dan pengaturan pencatatan audit Anda.
- Datashares - Menunjukkan datashares.

Mengedit keamanan dan enkripsi

Amazon Redshift Serverless diamankan dengan menggunakan enkripsi KMS. Anda dapat memperbarui pengaturan enkripsi melalui konsol:

1. Pilih konfigurasi Namespace dari menu utama di konsol, pilih namespace yang akan diedit, dan pilih Edit pada tab Keamanan dan enkripsi. Dialog muncul.
2. Anda dapat memilih Sesuaikan pengaturan enkripsi dan kemudian Pilih kunci yang dikelola AWS pelanggan untuk mengubah kunci yang digunakan untuk mengenkripsi sumber daya Anda.
3. Untuk pencatatan Audit, pilih log yang akan diekspor. Setiap jenis log menentukan metadata yang berbeda.
4. Untuk menyelesaikan pembaruan konfigurasi, pilih Simpan perubahan.

Mengubah AWS KMS kunci untuk namespace

Di Amazon Redshift, enkripsi melindungi data saat istirahat. Amazon Redshift Serverless menggunakan enkripsi AWS KMS kunci secara otomatis untuk mengenkripsi sumber daya dan snapshot Amazon Redshift Tanpa Server Anda. Sebagai praktik terbaik, sebagian besar organisasi meninjau jenis data yang mereka simpan dan memiliki rencana untuk memutar kunci enkripsi sesuai jadwal. Frekuensi untuk memutar kunci dapat bervariasi, tergantung pada kebijakan Anda untuk keamanan data. Amazon Redshift Tanpa Server mendukung perubahan AWS KMS kunci untuk namespace sehingga Anda dapat mematuhi kebijakan keamanan organisasi Anda.

Ketika Anda mengubah AWS KMS kunci, data tetap tidak berubah.

Mengubah AWS KMS kunci menggunakan konsol

Di Amazon Redshift, enkripsi melindungi data saat istirahat. Amazon Redshift Serverless menggunakan enkripsi AWS KMS kunci secara otomatis untuk mengenkripsi Amazon Redshift Serverless dan snapshot. Sebagai praktik terbaik, sebagian besar organisasi meninjau jenis data

yang mereka simpan dan memiliki rencana untuk memutar kunci enkripsi sesuai jadwal. Frekuensi untuk memutar kunci dapat bervariasi, tergantung pada kebijakan Anda untuk keamanan data. Amazon Redshift Tanpa Server mendukung perubahan AWS KMS kunci untuk namespace sehingga Anda dapat mematuhi kebijakan keamanan organisasi Anda.

Ketika Anda mengubah AWS KMS kunci, data tetap tidak berubah.

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih konfigurasi Namespace. Pilih namespace Anda dari daftar.
3. Dari tab Keamanan dan enkripsi, pilih Edit.
4. Pilih Sesuaikan pengaturan enkripsi dan kemudian pilih kunci untuk namespace. Anda dapat membuat kunci baru secara opsional.

Mengubah kunci AWS KMS enkripsi menggunakan AWS CLI

Gunakan `update-namespace` untuk mengubah AWS KMS kunci untuk namespace. Berikut ini menunjukkan sintaks untuk perintah:

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

Anda harus memiliki namespace yang dibuat atau perintah CLI menghasilkan kesalahan.

Waktu yang diperlukan untuk mengubah kunci tergantung pada jumlah data di Amazon Redshift Serverless. Ini biasanya memakan waktu lima belas menit per 8TB data yang disimpan.

Batasan

Anda tidak dapat mengubah dari KMS Key yang dikelola pelanggan menjadi AWS KMS kunci. Dalam hal ini, Anda harus membuat namespace baru.

Anda tidak dapat melakukan tindakan lain saat kuncinya sedang diubah.

Menghapus namespace

Jika Anda ingin menghapus namespace dengan workgroup terkait, Anda harus menghapus workgroup terlebih dahulu.

Di konsol Amazon Redshift Tanpa Server, selesaikan langkah-langkah berikut:

1. Pilih konfigurasi Namespace dari menu kiri dan kemudian pilih namespace yang ingin Anda hapus dari daftar.
2. Pilih Tindakan dan pilih Hapus namespace.
3. Kotak dialog terbuka. Anda dapat menyimpan data Anda dengan membuat snapshot manual sebelum menyelesaikan operasi penghapusan.

Ketik hapus dan pilih Hapus untuk mengonfirmasi.

Memantau kueri dan beban kerja dengan Amazon Redshift Serverless

Anda dapat memantau kueri dan beban kerja Tanpa Server Amazon Redshift dengan tampilan sistem yang disediakan.

Tampilan pemantauan adalah tampilan sistem di Amazon Redshift Tanpa Server yang digunakan untuk memantau penggunaan kueri dan beban kerja. Pandangan ini terletak di `pg_catalog` skema. Tampilan sistem yang tersedia telah dirancang untuk memberi Anda informasi yang diperlukan untuk memantau Amazon Redshift Tanpa Server, yang jauh lebih sederhana daripada yang diperlukan untuk cluster yang disediakan. Tampilan sistem SYS telah dirancang untuk bekerja dengan Amazon Redshift Serverless. Untuk menampilkan informasi yang disediakan oleh tampilan ini, jalankan pernyataan SQL `SELECT`.

Tampilan sistem didefinisikan untuk mendukung tujuan pemantauan berikut.

Pemantauan beban kerja

Anda dapat memantau aktivitas kueri Anda dari waktu ke waktu untuk:

- Memahami pola beban kerja, sehingga Anda tahu apa yang normal (baseline) dan apa yang ada dalam perjanjian tingkat layanan bisnis (). SLAs
- Identifikasi penyimpangan dengan cepat dari normal, yang mungkin merupakan masalah sementara atau sesuatu yang menjamin tindakan lebih lanjut.

Pemantauan pemuatan dan bongkar data

Pergerakan data masuk dan keluar dari Amazon Redshift Serverless adalah fungsi penting. Anda menggunakan `COPY` dan `UNLOAD` untuk memuat atau membongkar data, dan Anda harus

memantau kemajuan secara ketat dalam hal bytes/rows transfer dan file yang diselesaikan untuk melacak kepatuhan terhadap bisnis. SLAs Hal ini biasanya dilakukan dengan menjalankan query tabel sistem sering (yaitu, setiap menit) untuk melacak kemajuan dan meningkatkan peringatan untuk investigation/corrective tindakan jika penyimpangan signifikan terdeteksi.

Diagnostik kegagalan dan masalah

Ada kasus di mana Anda harus mengambil tindakan untuk kegagalan kueri atau runtime. Pengembang mengandalkan tabel sistem untuk mendiagnosis sendiri masalah dan menentukan solusi yang benar.

Penyempurnaan performa

Anda mungkin perlu menyetel kueri yang tidak memenuhi persyaratan SLA baik dari awal, atau telah menurun seiring waktu. Untuk menyetel, Anda harus memiliki detail runtime termasuk run plan, statistik, durasi, dan konsumsi sumber daya. Anda memerlukan data dasar untuk pertanyaan yang menyinggung untuk menentukan penyebab penyimpangan dan untuk memandu Anda cara meningkatkan kinerja.

Objek pengguna pemantauan acara

Anda perlu memantau tindakan dan aktivitas pada objek pengguna, seperti menyegarkan tampilan terwujud, menyedot debu, dan menganalisis. Ini termasuk peristiwa yang dikelola sistem seperti penyegaran otomatis untuk tampilan terwujud. Anda ingin memantau kapan suatu peristiwa berakhir jika itu dimulai pengguna, atau yang terakhir berhasil dijalankan jika sistem dimulai.

Pelacakan penggunaan untuk penagihan

Anda dapat memantau tren penggunaan Anda dari waktu ke waktu untuk:

- Menginformasikan perencanaan anggaran dan perkiraan ekspansi bisnis.
- Identifikasi peluang penghematan biaya potensial seperti menghapus data dingin.

Gunakan tampilan sistem SYS untuk memantau Amazon Redshift Tanpa Server;. Untuk informasi selengkapnya tentang tampilan pemantauan SYS, buka tampilan pemantauan [SYS di Panduan Pengembang Database Amazon Redshift](#).

Menambahkan kebijakan pemantauan kueri

Superuser dapat memberikan akses ke pengguna yang bukan pengguna super sehingga mereka dapat melakukan pemantauan kueri untuk semua pengguna. Pertama, Anda menambahkan

kebijakan untuk pengguna atau peran untuk menyediakan akses pemantauan kueri. Kemudian, Anda memberikan izin pemantauan kueri kepada pengguna atau peran.

Untuk menambahkan kebijakan pemantauan kueri

1. Pilih <https://console.aws.amazon.com/iam/>.
2. Di bagian Manajemen akses, pilih Kebijakan.
3. Pilih Buat Kebijakan.
4. Pilih JSON dan tempel definisi kebijakan berikut.

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. Pilih Tinjau kebijakan.
6. Untuk Nama, masukkan nama untuk kebijakan, seperti `query-monitoring`.
7. Pilih Buat kebijakan.

Setelah membuat kebijakan, Anda dapat memberikan izin yang sesuai.

Untuk memberikan akses dan menambahkan izin bagi pengguna, grup, atau peran Anda:

- Pengguna dan grup di AWS IAM Identity Center:

Buat rangkaian izin. Ikuti instruksi di [Buat rangkaian izin](#) dalam Panduan Pengguna AWS IAM Identity Center .

- Pengguna yang dikelola di IAM melalui penyedia identitas:

Buat peran untuk federasi identitas. Ikuti instruksi dalam [Buat peran untuk penyedia identitas pihak ketiga \(federasi\)](#) dalam Panduan Pengguna IAM.

- Pengguna IAM:

- Buat peran yang dapat diambil pengguna Anda. Ikuti instruksi dalam [Buat peran untuk pengguna IAM](#) dalam Panduan Pengguna IAM.
- (Tidak disarankan) Lampirkan kebijakan langsung ke pengguna atau tambahkan pengguna ke grup pengguna. Ikuti petunjuk dalam [Menambahkan izin ke pengguna \(konsol\)](#) dalam Panduan Pengguna IAM.

Memberikan izin pemantauan kueri untuk pengguna

Pengguna dengan `sys:monitor` izin dapat melihat semua kueri. Selain itu, pengguna dengan `sys:operator` izin dapat membatalkan kueri, menganalisis riwayat kueri, dan melakukan operasi vakum.

Untuk memberikan izin pemantauan kueri bagi pengguna

1. Masukkan perintah berikut untuk menyediakan akses monitor sistem, di mana nama pengguna adalah nama pengguna yang ingin Anda berikan aksesnya.

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

2. (Opsional) Masukkan perintah berikut untuk menyediakan akses operator sistem, di mana nama pengguna adalah nama pengguna yang ingin Anda berikan aksesnya.

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


Memberikan izin pemantauan kueri untuk peran

Pengguna dengan peran yang memiliki `sys:monitor` izin dapat melihat semua kueri. Selain itu, pengguna dengan peran yang memiliki `sys:operator` izin dapat membatalkan kueri, menganalisis riwayat kueri, dan melakukan operasi vakum.

Untuk memberikan izin pemantauan kueri untuk peran

1. Masukkan perintah berikut untuk menyediakan akses monitor sistem, di mana `role-name` adalah nama peran yang ingin Anda berikan aksesnya.

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

2. (Opsional) Masukkan perintah berikut untuk menyediakan akses operator sistem, di mana `role-name` adalah nama peran yang ingin Anda berikan aksesnya.

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

Menetapkan batas penggunaan, termasuk menetapkan batas RPU

Di bawah tab Batas untuk grup kerja, Anda dapat menambahkan satu atau beberapa batas penggunaan untuk mengontrol maksimum yang RPUs Anda gunakan dalam periode waktu tertentu, atau untuk menetapkan batas penggunaan berbagi data.

1. Pilih Kelola batas penggunaan. Bagian batas muncul di bagian bawah panel Compute use by period.
2. Tetapkan batas penggunaan dalam jumlah jam RPU.
3. Pilih Frekuensi, yaitu Harian, Mingguan, atau Bulanan. Ini menetapkan periode waktu untuk batas penggunaan. Memilih Harian dalam hal ini memberi Anda kontrol yang lebih rinci.
4. Tetapkan batas penggunaan, dalam jumlah jam.
5. Atur tindakan. Ini adalah sebagai berikut:
 - Masuk ke tabel sistem - Menambahkan catatan ke tampilan sistem [SYS_QUERY_HISTORY](#). Anda dapat menanyakan `usage_limit` kolom dalam tampilan ini untuk menentukan apakah kueri melebihi batas.

- Peringatan - Menggunakan Amazon SNS untuk mengatur langganan notifikasi dan mengirim pemberitahuan jika batas dilanggar. Anda dapat memilih topik Amazon SNS yang ada atau membuat yang baru.
- Matikan kueri pengguna - Menonaktifkan kueri untuk menghentikan penggunaan Amazon Redshift Tanpa Server. Ini juga mengirimkan pemberitahuan.

Dua tindakan pertama bersifat informasional, tetapi yang terakhir mematikan pemrosesan kueri.

6. Secara opsional, Anda dapat menetapkan batas penggunaan berbagi data Lintas Wilayah, yang membatasi jumlah data yang ditransfer dari Wilayah produsen ke Wilayah konsumen yang dapat ditanyakan konsumen. Untuk melakukan ini, pilih Tambahkan batas dan ikuti langkah-langkahnya.
7. Pilih Simpan perubahan di bagian bawah halaman untuk menyimpan batas.
8. Atur hingga 3 batas lagi seperlunya.

Untuk informasi konseptual selengkapnya tentang RPU dan penagihan, lihat [Penagihan Amazon Redshift Tanpa Server](#).

Menetapkan batas kueri

Di bawah tab Batas untuk grup kerja, Anda dapat menambahkan batas untuk memantau kinerja dan batas. Untuk informasi selengkapnya tentang batas pemantauan kueri, lihat aturan [pemantauan kueri WLM](#).

1. Pilih Kelola batas kueri. Pilih Tambahkan batas baru pada dialog Kelola batas kueri.
2. Pilih jenis batas yang ingin Anda tetapkan dan masukkan nilai untuk batas yang sesuai.
3. Pilih Simpan perubahan untuk menyimpan batas.

Ketika Anda mengubah batas kueri dan parameter konfigurasi, database Anda akan restart.

Mengatur antrian kueri

Amazon Redshift Serverless mendukung manajemen sumber daya kueri berbasis antrian. Anda dapat membuat antrean kueri khusus dengan aturan pemantauan yang disesuaikan untuk beban kerja yang berbeda-beda. Fitur ini memberikan kontrol granular atas penggunaan sumber daya.

Aturan pemantauan kueri (QMR) hanya berlaku di tingkat grup kerja Redshift Serverless, yang memengaruhi semua kueri yang dijalankan di workgroup ini secara seragam. Pendekatan berbasis antrian memungkinkan Anda membuat antrian dengan aturan pemantauan khusus. Anda dapat menetapkan antrian ini ke peran pengguna dan grup kueri tertentu. Setiap antrian beroperasi secara independen, dengan aturan yang hanya memengaruhi kueri dalam antrian tersebut.

Antrian memungkinkan Anda mengatur predikat berbasis metrik dan respons otomatis. Misalnya, Anda dapat mengonfigurasi aturan untuk secara otomatis membatalkan kueri yang melebihi batas waktu atau menggunakan terlalu banyak sumber daya.

Pertimbangan-pertimbangan

Pertimbangkan hal berikut saat menggunakan antrian tanpa server:

- Kunci konfigurasi Manajemen Beban Kerja (WLM) berikut yang digunakan di klaster yang disediakan Amazon Redshift tidak didukung dalam antrian Redshift Tanpa Server:,,,,,,.
`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`

Selain itu hop, tindakan `change_query_priority` tidak didukung di Tanpa Server.

- Tindakan hop (memindahkan kueri antar antrian) tidak didukung di Amazon Redshift Tanpa Server.
- Prioritas antrian hanya didukung untuk klaster yang disediakan Amazon Redshift.
- Amazon Redshift Serverless secara otomatis mengelola penskalaan dan alokasi sumber daya untuk kinerja optimal, sehingga Anda tidak perlu mengonfigurasi prioritas antrian secara manual.

Menyiapkan antrian kueri

Anda dapat membuat antrian di bawah tab Limits untuk grup kerja tanpa server menggunakan, Konsol Manajemen AWS, AWS CLI atau Redshift Serverless API.

Console

Ikuti langkah-langkah ini untuk membuat antrian untuk grup kerja tanpa server Anda.

1. Arahkan ke grup kerja Redshift Serverless Anda.
2. Pilih tab Limits.
3. Di bawah Query Queues, pilih Aktifkan Antrian.

 Important

Mengaktifkan antrian kueri adalah perubahan permanen. Anda tidak dapat kembali ke pemantauan tanpa antrian setelah diaktifkan.

4. Konfigurasi antrian Anda menggunakan parameter berikut:

Parameter tingkat antrian

- `name`- Pengidentifikasi antrian (wajib, unik, tidak kosong)
- `user_role`- Array peran pengguna (opsional)
- `query_group`- Array grup kueri (opsional)
- `query_group_wild_card`- 0 atau 1 untuk mengaktifkan pencocokan wildcard (opsional)
- `user_group_wild_card`- 0 atau 1 untuk mengaktifkan pencocokan wildcard (opsional)
- `rules`- Array aturan pemantauan (opsional)

Parameter tingkat aturan

- `rule_name`- Pengidentifikasi unik, maks 32 karakter (wajib)
- `predicate`- Array kondisi, 1-3 predikat (wajib)
- `action`- “batalkan” atau “log” (wajib)

Parameter tingkat predikat

- `metric_name`- Metrik untuk memantau (wajib)
- `operator`- “=”, “<”, or “>” (wajib)
- `value`- Ambang numerik (wajib)

Batas

- Antrian maks 8
- Max 25 aturan di semua antrian
- Maks 3 predikat per aturan
- Nama aturan harus unik secara global

Contoh Konfigurasi

Contoh tiga antrian: satu untuk kueri dasbor dengan batas waktu singkat, satu untuk kueri ETL dengan batas waktu yang lama dan antrian admin:

```
[
  {
    "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"]
  }
]

```

Dalam contoh ini:

- Kueri dasbor dibatalkan jika berjalan lebih dari 60 detik
- Kueri ETL dicatat jika dijalankan lebih dari satu jam
- Antrian admin tidak memiliki batasan sumber daya

CLI

Anda dapat mengelola antrian menggunakan `CreateWorkgroup` atau `UpdateWorkgroup` APIs dengan parameter `wlm_json_configuration` konfigurasi untuk menentukan antrian dalam format JSON.

```

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\"]}]\"}]'
```

Praktik terbaik

Ingatlah praktik terbaik berikut saat Anda menggunakan antrian tanpa server.

- Gunakan antrian terpisah untuk beban kerja dengan persyaratan batas yang berbeda (misalnya, ETL, pelaporan, atau analisis ad-hoc).
- Mulailah dengan ambang batas sederhana dan sesuaikan berdasarkan perilaku kueri dan pola penggunaan. Anda dapat memantau pola penggunaan kueri menggunakan tabel dan tampilan yang didokumentasikan dalam [tabel dan tampilan sistem untuk aturan pemantauan kueri](#).

Memeriksa data ringkasan Amazon Redshift Tanpa Server menggunakan dasbor

Dasbor Amazon Redshift Tanpa Server berisi kumpulan panel yang menampilkan at-a-glance metrik dan informasi tentang workgroup dan namespace Anda. Panel-panel ini meliputi:

- Ringkasan sumber daya - Menampilkan informasi tingkat tinggi tentang Amazon Redshift Tanpa Server, seperti penyimpanan yang digunakan dan metrik lainnya.
- Ringkasan kueri - Menampilkan informasi tentang kueri, termasuk kueri lengkap dan kueri yang sedang berjalan. Pilih Lihat detail untuk membuka layar yang memiliki filter tambahan.
- Kapasitas RPU yang digunakan - Menampilkan kapasitas keseluruhan yang digunakan selama periode waktu tertentu, seperti sepuluh jam sebelumnya, misalnya.
- Datashares - Menunjukkan jumlah datashares, yang digunakan untuk berbagi data antara, misalnya, akun. AWS Metrik menunjukkan datashares mana yang memerlukan otorisasi, dan informasi lainnya.
- Total penggunaan komputasi - Menunjukkan total jam RPU yang Anda konsumsi untuk grup kerja yang dipilih selama rentang waktu yang dipilih, hingga 7 hari terakhir.

Dari dasbor, Anda dapat dengan cepat menyelami metrik yang tersedia ini untuk memeriksa detail mengenai Amazon Redshift Tanpa Server, atau meninjau kueri, atau melacak item pekerjaan.

Pencatatan log audit untuk Amazon Redshift Nirserver

Anda dapat mengonfigurasi Amazon Redshift Tanpa Server untuk mengeksport data log koneksi, pengguna, dan aktivitas pengguna ke grup log di Amazon Logs. CloudWatch Dengan Amazon CloudWatch Logs, Anda dapat melakukan analisis real-time dari data log dan menggunakannya CloudWatch untuk membuat alarm dan melihat metrik. Anda dapat menggunakan CloudWatch Log untuk menyimpan catatan log Anda dalam penyimpanan yang tahan lama.

Anda dapat membuat CloudWatch alarm untuk melacak metrik menggunakan konsol Amazon Redshift. Untuk informasi selengkapnya tentang membuat alarm, lihat [Mengelola alarm](#).

Untuk mengeksport data log yang dihasilkan ke Amazon CloudWatch Logs, masing-masing log harus dipilih untuk diekspor di pengaturan konfigurasi Amazon Redshift Tanpa Server, di konsol. Anda dapat melakukan ini dengan memilih pengaturan konfigurasi Namespace, di bawah Keamanan dan enkripsi.

Log peristiwa di CloudWatch

Setelah memilih log Redshift mana yang akan diekspor, Anda dapat memantau peristiwa di Amazon CloudWatch Logs. Grup log baru secara otomatis dibuat untuk Amazon Redshift Serverless, yang `log_type` mewakili jenis log.

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

Saat Anda membuat workgroup dan namespace pertama Anda, default adalah nama namespace. Nama grup log bervariasi sesuai dengan apa yang Anda sebut namespace.

Misalnya, jika Anda mengeksport log koneksi, data log disimpan dalam grup log berikut.

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

Peristiwa log diekspor ke grup log menggunakan aliran log tanpa server. Perilaku tergantung pada kondisi berikut mana yang benar:

- Ada grup log dengan nama yang ditentukan. Redshift mengeksport data log menggunakan grup log yang ada. Untuk membuat grup log dengan periode penyimpanan log, filter metrik, dan akses pelanggan yang telah ditentukan sebelumnya, Anda dapat menggunakan konfigurasi otomatis, seperti yang disediakan oleh AWS CloudFormation

- Grup log dengan nama yang ditentukan tidak ada. Saat entri log yang cocok terdeteksi di log untuk instance, Amazon Redshift Serverless akan membuat grup log baru di Amazon Logs secara otomatis. CloudWatch Grup log menggunakan periode penyimpanan log default Never Expires. Untuk mengubah periode penyimpanan log, gunakan konsol Amazon CloudWatch Logs, API Amazon Logs AWS CLI, atau Amazon CloudWatch Logs. Untuk informasi selengkapnya tentang mengubah periode penyimpanan CloudWatch log di Log, lihat [Mengubah penyimpanan data log di Bekerja dengan grup log dan aliran log](#).

Untuk mencari informasi dalam peristiwa log, gunakan konsol Amazon CloudWatch Logs, API Amazon Logs AWS CLI, atau Amazon CloudWatch Logs. Untuk informasi selengkapnya tentang mencari dan memfilter data log, lihat [Menelusuri dan memfilter data log](#).

CloudWatch metrik

Amazon Redshift Metrik tanpa server dibagi menjadi metrik komputasi dan metrik data dan penyimpanan, masing-masing termasuk dalam kumpulan dimensi workgroup dan namespace. Untuk informasi selengkapnya tentang workgroup dan namespace, lihat [Workgroups](#) dan namespace.

CloudWatch metrik komputasi adalah sebagai berikut:

Nama metrik	Unit	Deskripsi	Set dimensi
QueriesCompletedPerSecond	Jumlah kueri	Jumlah kueri yang diselesaikan setiap detik.	{Database LatencyRange,, Kelompok Kerja}, {LatencyRange, Kelompok Kerja}
QueryDuration	Mikrodetik	Jumlah rata-rata waktu untuk menyelesaikan kueri.	{Database LatencyRange,, Kelompok Kerja}, {LatencyRange, Kelompok Kerja}
QueriesRunning	Jumlah kueri	Jumlah query yang berjalan pada suatu titik waktu.	{Database QueryType,, Kelompok Kerja},

Nama metrik	Unit	Deskripsi	Set dimensi
			{QueryType, Kelompok Kerja}
QueriesQueued	Jumlah kueri	Jumlah kueri dalam antrian pada suatu titik waktu.	{Database QueryType,, Kelompok Kerja}, {QueryType, Kelompok Kerja}
DatabaseConnections	Jumlah koneksi	Jumlah koneksi ke database pada suatu titik waktu.	{Database, Kelompok Kerja}, {Kelompok Kerja}
QueryRuntimeBreakdown	Milidetik	Total waktu kueri dijalankan, berdasarkan tahap kueri.	{Database, Panggung, Kelompok Kerja}, {Tahap, Kelompok Kerja}
ComputeCapacity	RPU	Jumlah rata-rata unit komputasi yang dialokasikan selama 30 menit terakhir, dibulatkan ke bilangan bulat terdekat.	{Kelompok Kerja}
ComputeSeconds	RPU-detik	Akumulasi detik unit komputasi yang digunakan dalam 30 menit terakhir.	{Kelompok Kerja}

Nama metrik	Unit	Deskripsi	Set dimensi
QueriesSucceeded	Jumlah kueri	Jumlah kueri yang berhasil dalam 5 menit terakhir.	{Database QueryType,, Kelompok Kerja}, {QueryType, Kelompok Kerja}
QueriesFailed	Jumlah kueri	Jumlah kueri yang gagal dalam 5 menit terakhir.	{Database QueryType,, Kelompok Kerja}, {QueryType, Kelompok Kerja}

Nama metrik	Unit	Deskripsi	Set dimensi
UsageLimitAvailable	RPU-jam atau TBs	Tergantung pada UsageType, UsageLimitAvailable mengembalikan yang berikut: - Jika UsageType adalah SERVERLESS_COMPUTE, UsageLimitAvailable mengembalikan jumlah RPU-jam yang tersisa yang dapat ditanyakan oleh workgroup dalam batas yang diberikan. - Jika UsageType adalah CROSS_REGION_DATASHARING, UsageLimitAvailable mengembal	{UsageLimitId, UsageType, Kelompok Kerja}

Nama metrik	Unit	Deskripsi	Set dimensi
		ikan jumlah yang tersisa dari yang dapat dipindai pelanggan dalam batas TBs yang diberikan.	

Nama metrik	Unit	Deskripsi	Set dimensi
UsageLimitConsumed	RPU-jam atau TBs	Tergantung pada UsageType, UsageLimitConsumed mengembalikan yang berikut: - Jika UsageType adalah SERVERLESS_COMPUTE, UsageLimitConsumed mengembalikan jumlah RPU-jam yang telah ditanyakan oleh workgroup dalam batas yang diberikan. - Jika UsageType adalah CROSS_REGION_DATASHARING, UsageLimitConsumed mengembalikan jumlah	{UsageLimitId, UsageType, Kelompok Kerja}

Nama metrik	Unit	Deskripsi	Set dimensi
		yang telah digunakan pelanggan untuk memindai dalam batas TBs yang diberikan.	
ExtraComputeForAutomaticOptimizationChargedSeconds	RPU-detik	Jumlah detik unit komputasi yang dibebankan untuk operasi pengoptimalan otomatis dalam 30 menit terakhir.	{Kelompok Kerja}

CloudWatch Metrik data dan penyimpanan adalah sebagai berikut:

Nama metrik	Unit	Deskripsi	Set dimensi
TotalTableCount	Jumlah tabel	Jumlah tabel pengguna yang ada pada suatu titik waktu. Total ini tidak termasuk tabel Amazon Redshift Spectrum.	{Database, Ruang Nama}
DataStorage	Megabyte	Jumlah megabyte yang digunakan	{Ruang nama}

Nama metrik	Unit	Deskripsi	Set dimensi
		, dalam disk atau ruang penyimpanan, untuk data Redshift.	

SnapshotStorageMetriknya adalah namespace- dan workgroup-agnostik.

CloudWatchSnapshotStoragemetrik adalah sebagai berikut:

Nama metrik	Unit	Deskripsi	Set dimensi
SnapshotStorage	Megabyte	Jumlah megabyte yang digunakan , dalam disk atau ruang penyimpanan, untuk Snapshots.	{}

Kumpulan dimensi adalah dimensi pengelompokan yang diterapkan pada metrik Anda. Anda dapat menggunakan grup dimensi ini untuk menentukan bagaimana statistik Anda diambil.

Tabel berikut merinci dimensi dan nilai dimensi untuk metrik tertentu:

Dimensi	Deskripsi dan nilai
DatabaseName	Nama basis data. Nilai khusus.
Latency	Kemungkinan nilainya adalah sebagai berikut: • Pendek - di bawah 10 detik • Sedang — antara 10 detik dan 10 menit • Panjang — lebih dari 10 menit

Dimensi	Deskripsi dan nilai
QueryType	Kemungkinan nilai adalah INSERT, DELETE, UPDATE, UNLOAD, LOAD, SELECT, CTAS, dan OTHER.
stage	Tahapan eksekusi untuk kueri. Kemungkinan nilainya adalah sebagai berikut: • QueryPlanning: Waktu yang dihabiskan mengurai dan mengoptimalkan pernyataan SQL. • QueryWaiting: Waktu yang dihabiskan menunggu di antrian WLM. • QueryExecutingRead: Waktu yang dihabiskan untuk mengeksekusi kueri baca. • QueryExecutingInsert: Waktu yang dihabiskan untuk mengeksekusi kueri sisipan. • QueryExecutingDelete: Waktu yang dihabiskan untuk mengeksekusi kueri hapus. • QueryExecutingUpdate: Waktu yang dihabiskan untuk mengeksekusi kueri pembaruan. • QueryExecutingCtas: Waktu yang dihabiskan untuk mengeksekusi membuat tabel sebagai kueri. • QueryExecutingUnload: Waktu yang dihabiskan untuk mengeksekusi kueri bongkar muat. • QueryExecutingCopy: Waktu yang dihabiskan untuk mengeksekusi kueri salinan. • QueryCommit: Waktu yang dihabiskan untuk berkomitmen.
Namespace	Nama namespace. Nilai khusus.

Dimensi	Deskripsi dan nilai
Workgroup	Nama workgroup. Nilai khusus.
UsageLimitId	Pengidentifikasi batas penggunaan.
UsageType	Fitur Amazon Redshift Tanpa Server dibatasi. Kemungkinan nilainya adalah sebagai berikut: • SERVERLESS_COMPUTE • CROSS_REGION_DATASHARING

Snapshot dan titik pemulihan

Cadangan di Amazon Redshift Serverless adalah point-in-time representasi objek dan data di namespace Anda. Ada dua jenis cadangan: snapshot yang dibuat secara manual dan titik pemulihan yang dibuat secara otomatis oleh Amazon Redshift Serverless untuk Anda.

Amazon Redshift Serverless secara otomatis membuat titik pemulihan setiap 30 menit atau setelah setiap 5 GB perubahan data per node, mana yang terjadi lebih dulu. Untuk dataset yang lebih besar (lebih dari 5 GB × jumlah node), interval minimal antara titik pemulihan adalah 15 menit. Semua titik pemulihan disimpan selama 24 jam.

Note

Anda tidak dapat membuat jadwal snapshot Anda sendiri untuk mengontrol kapan titik pemulihan dibuat.

Amazon Redshift Serverless membuat snapshot di Redshift Managed Storage (RMS). Untuk informasi selengkapnya, lihat [Kapasitas komputasi untuk Amazon Redshift Serverless](#).

Note

Tabel tanpa cadangan tidak didukung untuk klaster yang RA3 disediakan dan grup kerja Amazon Redshift Tanpa Server. Tabel yang ditandai sebagai no-backup di RA3 cluster atau grup kerja tanpa server diperlakukan sebagai tabel permanen yang akan selalu dicadangkan saat mengambil snapshot, dan selalu dipulihkan saat memulihkan dari snapshot. Untuk

menghindari biaya snapshot untuk tabel tanpa cadangan, potong sebelum mengambil snapshot.

Jika Anda menemukan bahwa Anda ingin mengambil data dalam snapshot atau titik pemulihan, Anda dapat memulihkan snapshot ke namespace tanpa server atau ke kluster yang disediakan. Ada tiga skenario di mana Anda dapat memulihkan snapshot:

- Kembalikan snapshot tanpa server ke namespace tanpa server.
- Kembalikan snapshot tanpa server ke cluster yang disediakan.
- Kembalikan snapshot cluster yang disediakan ke namespace tanpa server.

Saat mengembalikan snapshot tanpa server ke kluster yang disediakan, Anda harus memilih jenis node yang akan digunakan, seperti, dan jumlah node RA3, yang memungkinkan Anda mengontrol pengaturan di tingkat cluster atau node.

Untuk memulihkan snapshot kluster yang disediakan ke namespace tanpa server, mulai dari konsol yang disediakan Redshift, pilih snapshot yang akan dipulihkan, lalu pilih Pulihkan dari snapshot, Pulihkan ke namespace tanpa server. Amazon Redshift mengonversi tabel dengan kunci interleaved menjadi kunci pengurutan majemuk saat Anda memulihkan snapshot kluster yang disediakan ke namespace tanpa server. Untuk informasi selengkapnya tentang kunci sortir, lihat [Bekerja dengan kunci sortir](#).

Jika Anda ingin menambahkan konteks tambahan, Anda dapat menandai snapshot dan titik pemulihan dengan pasangan nilai kunci yang menyediakan metadata dan informasi ke snapshot dan titik pemulihan. Untuk informasi selengkapnya tentang menandai sumber daya, lihat Ikhtisar [sumber daya penandaan](#).

Terakhir, Anda juga dapat berbagi snapshot dengan AWS akun lain, yang memungkinkan mereka mengakses data dalam snapshot dan menjalankan kueri.

AWS Backup integrasi

Anda juga dapat membuat dan memulihkan snapshot menggunakan AWS Backup, layanan yang dikelola sepenuhnya yang membantu Anda memusatkan dan mengotomatiskan perlindungan data di seluruh AWS layanan, di cloud, dan di tempat. Untuk informasi selengkapnya, lihat [AWS Backup integrasi dengan Amazon Redshift](#). Untuk informasi tentang AWS Backup, lihat [Apa itu AWS Backup?](#) di Panduan AWS Backup Pengembang.

Membuat snapshot

Note

Tabel tanpa cadangan tidak didukung untuk klaster yang RA3 disediakan dan grup kerja Amazon Redshift Tanpa Server. Tabel yang ditandai sebagai no-backup di RA3 cluster atau grup kerja tanpa server diperlakukan sebagai tabel permanen yang akan selalu dicadangkan saat mengambil snapshot, dan selalu dipulihkan saat memulihkan dari snapshot. Untuk menghindari biaya snapshot untuk tabel tanpa cadangan, potong sebelum mengambil snapshot.

Untuk membuat snapshot, lakukan langkah-langkah dalam prosedur berikut.

Untuk membuat snapshot

1. Di konsol Amazon Redshift Tanpa Server, pilih Pencadangan data.
2. Pilih Buat snapshot.
3. Pilih namespace untuk membuat snapshot dari.
4. Masukkan pengenalan snapshot.
5. (Opsional) Pilih periode retensi. Jika Anda memilih Nilai kustom, pilih jumlah hari. Jumlah yang Anda pilih harus antara 1-3653 hari, inklusif. Defaultnya dipertahankan tanpa batas waktu.
6. Pilih Buat.

Untuk membuat snapshot dari konfigurasi namespace

1. Di konsol Amazon Redshift Tanpa Server, pilih konfigurasi Namespace.
2. Pilih namespace untuk membuat snapshot dari. Anda hanya dapat membuat snapshot dari namespace yang terkait dengan workgroup dan yang statusnya Tersedia.
3. Pilih tab Pencadangan data.
4. Pilih Buat snapshot.
5. Masukkan pengenalan snapshot.
6. (Opsional) Pilih periode retensi. Jika Anda memilih Nilai kustom, pilih jumlah hari. Jumlah yang Anda pilih harus antara 1-3653 hari, inklusif.
7. Pilih Buat.

Membuat snapshot akhir

Untuk membuat snapshot akhir dari semua data dalam namespace sebelum menghapus namespace, lakukan langkah-langkah dalam prosedur berikut.

Untuk membuat snapshot akhir

1. Di konsol Amazon Redshift Tanpa Server, pilih konfigurasi Namespace.
2. Pilih namespace yang akan dihapus.
3. Pilih Tindakan, Hapus.
4. Pilih Buat snapshot akhir.
5. Masukkan nama untuk snapshot.
6. Masukkan hapus.
7. Pilih Hapus.

Berbagi snapshot atau menghapus izin snapshot

Untuk berbagi snapshot dengan AWS akun lain atau menghapus akses akun ke snapshot, lakukan prosedur berikut.

Untuk berbagi atau menghapus akses ke snapshot

1. Di konsol Amazon Redshift Tanpa Server, pilih Pencadangan data.
2. Pilih snapshot untuk dibagikan.
3. Pilih Tindakan, Kelola akses.
4. Untuk berbagi snapshot dengan akun lain, masukkan Akun AWS ID. Untuk menghapus akses dari akun, pilih Hapus.
5. Pilih Simpan perubahan.

Menjadwalkan snapshot

Untuk mengontrol secara tepat kapan harus mengambil snapshot, Anda dapat membuat jadwal snapshot untuk namespaces tertentu. Saat menjadwalkan pembuatan snapshot, Anda dapat membuat acara satu kali atau menggunakan ekspresi cron Unix untuk membuat jadwal berulang. Ekspresi cron mendukung tiga bidang dan dipisahkan oleh spasi putih.

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

Bidang	Nilai-nilai	Wildcard
Menit	0–59	, - * /
Jam	0–23	, - * /
Day-of-month	1–31	, - * ? / L W
Bulan	1–12 atau JAN-DEC	, - * /
Day-of-week	1–7 atau SUN-SAT	, - * ? L #
Tahun	1970–2199	, - * /

Wildcard

- Wildcard , (koma) mencakup nilai tambahan. Di Day-of-week lapangan, MON, WED, FRI akan mencakup Senin, Rabu, dan Jumat. Nilai total dibatasi hingga 24 per bidang.
- Wildcard - (tanda hubung) menentukan rentang. Di Hour lapangan, 1-15 akan mencakup jam 1 hingga 15 dari hari yang ditentukan.
- Wildcard * (bintang) mencakup semua nilai di bidang. Di kolom Hours, * akan mencakup setiap jam.
- Wildcard / (garis miring) menentukan tambahan. Di Hours lapangan, Anda bisa masuk **1/10** untuk menentukan setiap jam ke-10, mulai dari jam pertama hari itu (misalnya, 01:00, 11:00, dan 21:00).
- Wildcard ? (tanda tanya) menentukan satu atau yang lain. Di **Day-of-month** lapangan Anda bisa masuk 7, dan jika Anda tidak peduli hari apa dalam minggu ketujuh, Anda bisa masuk? di Day-of-week lapangan.
- Wildcard L di kolom Day-of-month atau Day-of-week menentukan hari terakhir pada bulan atau minggu.
- Wildcard W di kolom Day-of-month menentukan hari kerja. Di kolom Day-of-month, 3W menentukan hari kerja yang paling dekat dengan pekan ketiga di bulan itu.
- Wildcard # di Day-of-week bidang menentukan contoh tertentu dari hari yang ditentukan dalam seminggu dalam sebulan. Sebagai contoh, 3#2 akan menjadi hari Selasa kedua setiap bulan: 3

mengacu pada hari Selasa karena itu adalah hari ketiga setiap minggu, dan 2 mengacu pada hari kedua dari jenis tersebut dalam bulan tersebut.

Note

Jika Anda menggunakan karakter '#', Anda hanya dapat menentukan satu ekspresi di day-of-week bidang. Misalnya, "3 #1 ,6 #3" tidak valid karena ditafsirkan sebagai dua ekspresi.

Batas

- Anda tidak dapat menentukan kolom Day-of-month dan Day-of-week dalam ekspresi cron yang sama. Jika Anda menentukan sebuah nilai di salah satu kolom, maka Anda harus menggunakan ? (tanda tanya) di kolom yang lain.
- Jadwal snapshot tidak mendukung frekuensi berikut:
 - Snapshot dijadwalkan lebih sering dari 1 per jam.
 - Snapshot dijadwalkan kurang dari 1 per hari (24 jam).

Jika Anda memiliki jadwal yang tumpang tindih yang menghasilkan penjadwalan snapshot dalam jendela 1 jam, kesalahan validasi akan terjadi.

Tabel berikut memiliki beberapa contoh string cron.

Menit	Jam	Hari dalam seminggu	Arti			
0	14-20/1	SELAM	Setiap jam antara jam 2 siang dan 8 malam pada hari Selasa.			
0	21	MON-FRI	Setiap malam jam 9 malam Senin-Jumat.			
30	0/6	SAT-MATAHARI	Setiap kenaikan 6 jam pada hari Sabtu dan Minggu dimulai			

Menit	Jam	Hari dalam seminggu	Arti			
			pada 30 menit setelah tengah malam (00:30) hari itu. Ini menghasilkan snapshot pada [00:30, 06:30, 12:30, dan 18:30] setiap hari.			
30	12/4	*	Setiap kenaikan 4 jam mulai pukul 12:30 setiap hari. Ini memutuskan untuk [12:30, 16:30, 20:30].			

Contoh berikut menunjukkan cara membuat jadwal yang berjalan dalam kenaikan 2 jam mulai dari 15:15 setiap hari.

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

Anda dapat menggunakan konsol Amazon Redshift Serverless, API, atau AWS CLI untuk membuat jadwal snapshot.

Untuk menjadwalkan snapshot

1. Di konsol Amazon Redshift Tanpa Server, pilih Pencadangan data.
2. Pilih jadwal Snapshot.
3. Pilih Buat jadwal.
4. Masukkan nama untuk jadwal snapshot.
5. Pilih namespace untuk membuat snapshot.
6. Masukkan ekspresi cron untuk jadwal atau gunakan pembuat jadwal untuk membuatnya.
7. (Opsional) Pilih periode retensi. Jika Anda memilih Nilai kustom, tentukan jumlah hari.
8. Pilih Buat jadwal.

Memperbarui periode retensi snapshot

Untuk memperbarui periode retensi snapshot, lakukan prosedur berikut.

Untuk memperbarui periode retensi snapshot

1. Di konsol Amazon Redshift Tanpa Server, pilih Pencadangan data.
2. Pilih snapshot untuk diperbarui.
3. Pilih Tindakan, Atur pengaturan snapshot manual.
4. Pilih periode retensi. Jika Anda memilih Nilai kustom, pilih jumlah hari.
5. Pilih Simpan perubahan.

Menghapus snapshot

Untuk menghapus snapshot, lakukan prosedur berikut.

Untuk menghapus snapshot

Note

Anda tidak dapat menghapus snapshot yang telah dibagikan dengan akun lain. Anda harus terlebih dahulu menghapus akses akun itu ke snapshot sebelum menghapus snapshot.

1. Di konsol Amazon Redshift Tanpa Server, pilih Pencadangan data.
2. Pilih snapshot untuk dihapus.
3. Pilih Tindakan, Hapus.
4. Pilih Hapus.

Memulihkan snapshot

Note

Tabel tanpa cadangan tidak didukung untuk klaster yang RA3 disediakan dan grup kerja Amazon Redshift Tanpa Server. Tabel yang ditandai sebagai no-backup di RA3 cluster atau

grup kerja tanpa server diperlakukan sebagai tabel permanen yang akan selalu dicadangkan saat mengambil snapshot, dan selalu dipulihkan saat memulihkan dari snapshot. Untuk menghindari biaya snapshot untuk tabel tanpa cadangan, potong sebelum mengambil snapshot.

Memulihkan snapshot ke namespace tanpa server menggantikan database saat ini dengan database dalam snapshot.

Memulihkan snapshot ke namespace tanpa server diselesaikan dalam dua fase. Fase pertama selesai dalam beberapa menit, mengembalikan data ke namespace Anda, dan membuatnya tersedia untuk kueri. Tahap kedua restorasi adalah di mana database Anda disetel, yang dapat menyebabkan masalah kinerja kecil. Fase kedua ini dapat berlangsung dari beberapa jam hingga beberapa hari, dan dalam beberapa kasus, beberapa minggu. Jumlah waktu tergantung pada ukuran data, tetapi kinerja semakin meningkat saat database disetel. Pada akhir fase ini, namespace tanpa server Anda sepenuhnya disetel, dan Anda dapat mengirimkan kueri tanpa masalah kinerja.

Untuk mengembalikan snapshot ke namespace tanpa server

1. Di konsol Amazon Redshift Tanpa Server, pilih Pencadangan data.
2. Pilih snapshot untuk dipulihkan. Anda hanya dapat memulihkan satu snapshot pada satu waktu.
3. Pilih Tindakan, Pulihkan ke namespace tanpa server.
4. Pilih namespace yang tersedia untuk dipulihkan. Anda hanya dapat mengembalikan ke ruang nama yang statusnya Tersedia.
5. Pilih Pulihkan.

Untuk memulihkan snapshot ke kluster yang disediakan

1. Di konsol Amazon Redshift Tanpa Server, pilih Pencadangan data.
2. Pilih snapshot untuk dipulihkan.
3. Pilih Tindakan, Pulihkan ke kluster yang disediakan.
4. Masukkan pengidentifikasi cluster.
5. Pilih tipe Node. Jumlah node tergantung pada jenis node.
6. Ikuti petunjuk pada halaman di halaman konsol untuk memasukkan properti untuk konfigurasi Cluster. Lihat [Membuat kluster](#) untuk informasi selengkapnya.

[Untuk informasi selengkapnya tentang snapshot pada kluster yang disediakan, lihat snapshot dan cadangan Amazon Redshift.](#)

Mengonversi titik pemulihan

Poin pemulihan di Amazon Redshift Tanpa Server dibuat kira-kira setiap 30 menit dan disimpan selama 24 jam. Untuk mengonversi titik pemulihan menjadi snapshot, lakukan langkah-langkah dalam prosedur berikut.

Untuk mengonversi titik pemulihan menjadi snapshot

1. Di konsol Amazon Redshift Tanpa Server, pilih Pencadangan data.
2. Di bawah Titik pemulihan, pilih waktu pembuatan titik pemulihan yang ingin Anda konversi ke snapshot.
3. Pilih Buat snapshot dari titik pemulihan.
4. Masukkan pengenalan Snapshot.
5. Pilih Buat.

Memulihkan titik pemulihan

Poin pemulihan di Amazon Redshift Tanpa Server dibuat kira-kira setiap 30 menit dan disimpan selama 24 jam. Untuk mengembalikan titik pemulihan ke namespace tanpa server, lakukan langkah-langkah dalam prosedur berikut

Untuk mengembalikan titik pemulihan ke namespace tanpa server

1. Di konsol Amazon Redshift Tanpa Server, pilih Pencadangan data.
2. Di bawah Poin pemulihan, pilih waktu pembuatan titik pemulihan yang ingin Anda pulihkan.
3. Pilih Pulihkan. Anda hanya dapat mengembalikan ke ruang nama yang statusnya Tersedia.
4. Masukkan restore di bidang input teks dan pilih Restore.

Menyalin cadangan ke yang lain AWS Region

Anda dapat mengonfigurasi Amazon Redshift Tanpa Server untuk secara otomatis menyalin snapshot dan titik pemulihan ke yang lain. AWS Region Saat Anda membuat snapshot di sumber AWS Region, itu disalin ke Wilayah tujuan. Anda dapat mengonfigurasi namespace Anda sehingga

hanya menyalin snapshot dan titik pemulihan ke satu tujuan AWS Region pada satu waktu. Untuk daftar Wilayah AWS tempat Amazon Redshift Serverless tersedia, lihat titik akhir yang terdaftar untuk [Redshift Serverless API](#) di. Referensi Umum Amazon Web Services

Saat mengonfigurasi penyalinan cadangan, Anda juga dapat menentukan periode retensi berapa lama Amazon Redshift Serverless harus menyimpan snapshot yang disalin. Anda tidak dapat mengubah periode retensi titik pemulihan, yang harus 1 hari. Periode retensi snapshot di Wilayah tujuan terpisah dari periode retensi snapshot di Wilayah sumber. Secara default, periode retensi adalah menyimpan snapshot tanpa batas waktu. Jika Anda memilih Nilai kustom pilih jumlah hari. Jumlah yang Anda pilih harus antara 1-3653 hari, inklusif.

Untuk mengubah Wilayah tujuan untuk menyalin snapshot, pertama nonaktifkan penyalinan cadangan, lalu tentukan Wilayah tujuan baru saat Anda mengaktifkan kembali penyalinan.

Setelah snapshot atau titik pemulihan disalin ke Wilayah tujuan, Anda dapat menggunakannya untuk memulihkan data ke Wilayah.

Secara default, data Anda dienkripsi dengan kunci yang AWS mengelola untuk Anda. Untuk menggunakan kunci yang berbeda, pilih kunci yang ingin Anda gunakan saat mengonfigurasi penyalinan cadangan di sumber AWS Region, dan Amazon Redshift Tanpa Server secara otomatis membuat hibah, yang memungkinkan enkripsi snapshot di tujuan. AWS Region

Untuk menyalin cadangan ke Wilayah lain, pastikan Anda memiliki izin IAM berikut:

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

Jika Anda menggunakan kunci KMS Anda sendiri untuk mengenkripsi cadangan Anda, Anda juga memerlukan izin berikut:

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

Untuk mengonfigurasi penyalinan snapshot atau titik pemulihan Anda ke yang lain AWS Region

1. Di konsol Amazon Redshift Tanpa Server, pilih namespace yang ingin Anda konfigurasi menyalin snapshot atau titik pemulihan.
2. Pilih Tindakan, Konfigurasi cadangan Lintas wilayah.

3. Pilih tujuan AWS Region untuk menyalin snapshot ke.
4. (Opsional) Pilih berapa lama untuk mempertahankan snapshot. Jika Anda memilih Nilai kustom pilih jumlah hari Jumlah yang Anda pilih harus antara 1-3653 hari, inklusif. Defaultnya adalah mempertahankan tanpa batas waktu.
5. (Opsional) Pilih AWS KMS kunci lain yang akan digunakan untuk mengenkripsi enkripsi di Wilayah tujuan.
6. Pilih Simpan konfigurasi.

Memulihkan tabel

Anda juga dapat memulihkan tabel tertentu dari snapshot atau titik pemulihan Saat melakukannya, Anda menentukan snapshot sumber atau titik pemulihan, database, skema, tabel, basis data target, skema, dan nama tabel baru. Tabel baru ini tidak dapat memiliki nama yang sama dengan tabel yang ada. Jika Anda ingin mengganti tabel yang ada dengan memulihkan tabel, Anda harus terlebih dahulu mengganti nama atau menjatuhkan tabel sebelum Anda mengembalikan tabel.

Note

Tabel tanpa cadangan tidak didukung untuk klaster yang RA3 disediakan dan grup kerja Amazon Redshift Tanpa Server. Tabel yang ditandai sebagai no-backup di RA3 cluster atau grup kerja tanpa server diperlakukan sebagai tabel permanen yang akan selalu dicadangkan saat mengambil snapshot, dan selalu dipulihkan saat memulihkan dari snapshot. Namun, pemulihan selektif tabel tanpa cadangan tidak didukung.

Tabel target dibuat menggunakan definisi kolom tabel sumber, atribut tabel, dan atribut kolom kecuali untuk kunci asing. Untuk mencegah konflik karena dependensi, tabel target tidak mewarisi kunci asing dari tabel sumber. Dependensi apa pun, seperti tampilan atau izin yang diberikan pada tabel sumber, tidak diterapkan ke tabel target.

Jika pemilik tabel sumber ada, maka pengguna tersebut adalah pemilik tabel yang dipulihkan, asalkan pengguna memiliki izin yang cukup untuk menjadi pemilik relasi dalam database dan skema yang ditentukan. Jika tidak, tabel yang dipulihkan dimiliki oleh pengguna admin yang dibuat saat cluster diluncurkan.

Tabel yang dipulihkan kembali ke keadaan semula pada saat cadangan diambil. Ini termasuk aturan visibilitas transaksi yang ditentukan oleh kepatuhan Amazon Redshift terhadap isolasi [serial](#),

yang berarti bahwa data akan segera terlihat dalam transaksi penerbangan yang dimulai setelah pencadangan.

Anda dapat menggunakan konsol Amazon Redshift Tanpa Server untuk memulihkan tabel dari snapshot.

Memulihkan tabel dari cadangan data memiliki batasan berikut:

- Anda hanya dapat mengembalikan satu tabel pada satu waktu.
- Dependensi apa pun, seperti tampilan atau izin yang diberikan pada tabel sumber, tidak diterapkan ke tabel target.
- Jika keamanan tingkat baris diaktifkan untuk tabel dipulihkan, Amazon Redshift Serverless mengembalikan tabel dengan keamanan tingkat baris diaktifkan.

Untuk memulihkan tabel menggunakan konsol Amazon Redshift Serverless

1. Di konsol Amazon Redshift Tanpa Server, pilih Pencadangan data.
2. Pilih snapshot atau titik pemulihan yang memiliki tabel untuk dipulihkan.
3. Pilih Tindakan, Kembalikan tabel dari snapshot atau Pulihkan tabel dari titik pemulihan.
4. Masukkan informasi tentang snapshot sumber atau titik pemulihan dan tabel target, lalu pilih Pulihkan tabel.

Berbagi data di Amazon Redshift Tanpa Server

Dengan berbagi data, Anda memiliki akses langsung ke data sehingga pengguna dapat melihat informasi yang paling up-to-date dan konsisten saat diperbarui di Amazon Redshift Tanpa Server.

Anda dapat membagikan data untuk tujuan pembacaan di berbagai instans Tanpa Server Amazon Redshift di dalam atau di seluruh Akun AWS

Anda dapat memulai berbagi data dengan menggunakan antarmuka SQL atau konsol Amazon Redshift. Untuk informasi selengkapnya, lihat [Berbagi data di Amazon Redshift di Panduan Pengembang](#) Database Amazon Redshift.

Dengan berbagi data, ruang nama Amazon Redshift Tanpa Server dan kluster yang disediakan dapat berbagi data langsung satu sama lain, baik di dalam seberang, atau di seberang. Akun AWS Akun AWS Wilayah AWS Untuk informasi selengkapnya, lihat [Wilayah tempat berbagi data tersedia](#).

Untuk mulai berbagi data dalam file Akun AWS, buka Konsol Manajemen AWS, lalu pilih konsol Amazon Redshift. Pilih konfigurasi Namespace dan kemudian Datashares.

Untuk memulai kueri data dalam datashare, buat database di namespace yang memiliki workgroup yang terkait dengannya. Dari datashare tertentu, pilih namespace yang memiliki workgroup yang terkait dengannya dan buat database untuk kueri data.

Pertimbangan-pertimbangan

Pertimbangkan hal berikut saat bekerja dengan berbagi data di Amazon Redshift Serverless:

- Amazon Redshift hanya mendukung klaster yang disediakan dari tipe instans ra3.16xlarge, ra3.4xlarge, dan ra3.xlplus, serta titik akhir tanpa server sebagai produsen atau konsumen berbagi data.
- Amazon Redshift Tanpa Server dienkripsi secara default.

Untuk daftar batasan pembagian data, termasuk objek database yang didukung, persyaratan enkripsi, dan persyaratan sort-key, lihat [Pertimbangan untuk berbagi data di Amazon Redshift di Panduan Pengembang Database Amazon Redshift](#).

Memberikan akses untuk melihat datashares

Superuser dapat memberikan akses ke pengguna yang bukan pengguna super sehingga mereka dapat melihat datashares yang dibuat oleh semua pengguna.

Untuk memberikan akses ke datashare bagi pengguna, gunakan perintah berikut untuk menyediakan akses datashare bagi pengguna, di mana `datashare_name` adalah nama datashare dan nama pengguna adalah nama pengguna yang ingin Anda berikan akses.

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

Untuk memberikan akses ke datashare untuk grup pengguna, pertama buat grup pengguna dengan pengguna. Untuk informasi tentang cara membuat grup pengguna, lihat [MEMBUAT GRUP](#). Kemudian, berikan akses datashare ke pengguna menggunakan perintah berikut, di mana `datashare_name` adalah nama datashare dan `user-group` adalah nama grup pengguna yang ingin Anda berikan akses.

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

Untuk informasi tentang cara menggunakan pernyataan GRANT, lihat [GRANT](#).

Mendaftarkan ruang nama ke AWS Glue Data Catalog

Anda dapat mendaftarkan seluruh ruang nama ke AWS Glue Data Catalog dan membuat katalog yang dikelola oleh AWS Glue. Anda dapat mengakses katalog ini dengan mesin SQL apa pun yang mendukung Apache Iceberg REST API. Untuk informasi selengkapnya tentang membuat katalog yang kompatibel dengan Apache Iceberg dari Amazon Redshift, lihat [kompatibilitas Apache Iceberg untuk Amazon Redshift di Panduan Pengembang Database Amazon Redshift](#).

Untuk mendaftarkan namespace tanpa server ke AWS Glue Data Catalog

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Redshift Serverless. Dasbor Tanpa Server muncul. Di bagian Namespaces/Workgroups adalah daftar namespace dan workgroup untuk akun Anda saat ini. AWS Region Jika Anda tidak memiliki ruang nama apa pun, pilih Buat grup kerja untuk membuat grup kerja dan namespace yang sesuai.
3. Pilih nama namespace yang ingin Anda daftarkan.
4. Dari Tindakan, pilih Daftar ke AWS Glue Data Catalog. Kotak Daftar ke AWS Glue Data Catalog pop-up muncul.
5. Masukkan ID AWS akun yang ingin Anda daftarkan namespace di bawah ID akun Tujuan. Ini adalah ID akun yang akan menyimpan katalog di AWS Glue Data Catalog.
6. Masukkan nama di bawah Daftar namespace sebagai. Ini akan menjadi nama namespace di Katalog Data.
7. Pilih Pendaftaran. Anda akan dibawa ke AWS Lake Formation konsol.
8. Ikuti proses pembuatan katalog di AWS Lake Formation. Untuk informasi tentang membuat katalog, lihat [Membawa data Amazon Redshift ke AWS Glue Data Catalog dalam Panduan AWS Lake Formation Pengembang](#).

Menandai sumber daya di Amazon Redshift Tanpa Server

Dalam AWS, tag adalah label yang ditentukan pengguna yang terdiri dari pasangan nilai kunci. Amazon Redshift Nirserver mendukung pemberian tag untuk menyediakan metadata tentang sumber daya secara sekilas.

Tag tidak diperlukan untuk sumber daya, tetapi mereka membantu menyediakan konteks.

Anda mungkin ingin menandai sumber daya dengan metadata dengan informasi yang terkait dengan sumber daya. Misalnya, Anda ingin melacak sumber daya mana yang termasuk dalam lingkungan pengujian dan lingkungan produksi. Anda dapat membuat lingkungan bernama kunci dan memberikan uji nilai atau produksi untuk mengidentifikasi sumber daya yang digunakan di setiap lingkungan. Jika Anda menggunakan penandaan di AWS layanan lain atau memiliki kategori standar untuk bisnis Anda, kami sarankan Anda membuat pasangan nilai kunci yang sama untuk sumber daya agar konsisten.

Jika Anda menghapus sumber daya, tag terkait akan dihapus. Anda dapat menggunakan konsol Amazon Redshift Serverless AWS CLI dan Amazon untuk menandai sumber daya tanpa server. Operasi API yang tersedia adalah `TagResource`, `UntagResource`, dan `ListTagsForResource`.

Setiap sumber daya memiliki satu set tag, yang merupakan kumpulan dari satu atau beberapa tag yang ditetapkan ke sumber daya. Setiap sumber daya dapat memiliki hingga 50 tag per set tag. Anda dapat menambahkan tag saat membuat sumber daya dan setelah sumber daya dibuat. Anda dapat menambahkan tag ke jenis sumber daya tanpa server berikut:

- Kelompok kerja
- Namespace
- Snapshot
- Poin pemulihan

Tag memiliki persyaratan sebagai berikut:

- Kunci tidak dapat diawali dengan `aws :`.
- Kunci harus unik per set tag.
- Kunci harus antara 1 dan 128 karakter yang diizinkan.
- Nilai harus antara 0 dan 256 karakter yang diizinkan.
- Nilai tidak harus unik per set tag.
- Karakter yang diizinkan untuk kunci dan nilai adalah huruf Unicode, digit, spasi putih, dan salah satu simbol berikut: `_`, `.`, `/`, `=`, `+`, `-`, `@`.
- Kunci dan nilai peka huruf besar dan kecil.

Untuk mengelola tag sumber daya Amazon Redshift Tanpa Server

1. Di konsol Amazon Redshift Tanpa Server, pilih Kelola Tag.
2. Masukkan jenis sumber daya yang akan dicari dan pilih Cari sumber daya. Pilih sumber daya yang ingin Anda kelola tag, lalu pilih Kelola tag.
3. Tentukan kunci dan nilai opsional yang ingin Anda tambahkan ke sumber daya. Saat memodifikasi tag, Anda dapat mengubah nilai tag, tetapi bukan kuncinya.
4. Setelah selesai menambahkan, menghapus, atau memodifikasi tag, pilih Simpan perubahan, lalu pilih Terapkan untuk menyimpan perubahan.

Cluster yang disediakan Amazon Redshift

Gudang data Amazon Redshift adalah kumpulan sumber daya komputasi yang disebut node, yang diatur ke dalam grup yang disebut cluster. Setiap klaster menjalankan mesin Amazon Redshift dan berisi satu atau lebih database.

Note

Saat ini, mesin Amazon Redshift versi 1.0 tersedia. Namun, saat mesin diperbarui, beberapa versi mesin Amazon Redshift mungkin tersedia untuk dipilih.

Cluster dan node di Amazon Redshift

Cluster Amazon Redshift terdiri dari node. Setiap cluster memiliki node pemimpin dan satu atau lebih node komputasi. Node pemimpin menerima kueri dari aplikasi klien, mem-parsing kueri, dan mengembangkan rencana eksekusi kueri. Node pemimpin kemudian mengoordinasikan eksekusi paralel dari rencana ini dengan node komputasi dan mengumpulkan hasil antara dari node ini. Kemudian akhirnya mengembalikan hasil kembali ke aplikasi klien.

Node komputasi menjalankan rencana eksekusi kueri dan mengirimkan data di antara mereka sendiri untuk melayani kueri ini. Hasil antara dikirim ke node pemimpin untuk agregasi sebelum dikirim kembali ke aplikasi klien. Untuk informasi selengkapnya tentang node pemimpin dan node komputasi, lihat [Arsitektur sistem gudang data di Panduan](#) Pengembang Database Amazon Redshift.

Note

Saat membuat klaster di Amazon Redshift console (<https://console.aws.amazon.com/redshiftv2/>), Anda bisa mendapatkan rekomendasi konfigurasi klaster berdasarkan ukuran data dan karakteristik kueri. Untuk menggunakan kalkulator ukuran ini, cari Bantu saya memilih di konsol di AWS Wilayah yang mendukung jenis RA3 simpul. Untuk informasi selengkapnya, lihat [Membuat kluster](#).

Saat Anda meluncurkan cluster, salah satu opsi yang Anda tentukan adalah tipe node. Tipe node menentukan CPU, RAM, kapasitas penyimpanan, dan tipe drive penyimpanan untuk setiap node.

Amazon Redshift menawarkan berbagai jenis node untuk mengakomodasi beban kerja Anda, dan sebaiknya pilih RA3 atau DC2 bergantung pada kinerja yang diperlukan, ukuran data, dan pertumbuhan data yang diharapkan.

RA3 node dengan penyimpanan terkelola memungkinkan Anda mengoptimalkan gudang data dengan menskalakan dan membayar komputasi dan penyimpanan terkelola secara independen. Dengan RA3, Anda memilih jumlah node berdasarkan persyaratan kinerja Anda dan hanya membayar untuk penyimpanan terkelola yang Anda gunakan. Ukuran RA3 cluster Anda berdasarkan jumlah data yang Anda proses setiap hari. Anda meluncurkan cluster yang menggunakan tipe RA3 node di virtual private cloud (VPC). Untuk informasi selengkapnya, lihat [Membuat klaster yang disediakan Redshift atau workgroup Amazon Redshift Serverless di VPC](#).

Penyimpanan terkelola Amazon Redshift menggunakan besar, kinerja tinggi SSDs di setiap RA3 node untuk penyimpanan lokal yang cepat dan Amazon S3 untuk penyimpanan tahan lama jangka panjang. Jika data dalam node tumbuh melebihi ukuran lokal besar SSDs, penyimpanan terkelola Amazon Redshift secara otomatis membongkar data tersebut ke Amazon S3. Anda membayar tarif rendah yang sama untuk penyimpanan terkelola Amazon Redshift terlepas dari apakah data berada dalam kinerja tinggi atau Amazon SSDs S3. Untuk beban kerja yang membutuhkan penyimpanan yang terus bertambah, penyimpanan terkelola memungkinkan Anda secara otomatis menskalakan kapasitas penyimpanan gudang data Anda terpisah dari node komputasi.

DC2 node memungkinkan Anda untuk memiliki gudang data komputasi intensif dengan penyimpanan SSD lokal disertakan. Anda memilih jumlah node yang Anda butuhkan berdasarkan ukuran data dan persyaratan kinerja. DC2 node menyimpan data Anda secara lokal untuk kinerja tinggi, dan seiring bertambahnya ukuran data, Anda dapat menambahkan lebih banyak node komputasi untuk meningkatkan kapasitas penyimpanan cluster. Untuk kumpulan data di bawah 1 TB (terkompresi), kami merekomendasikan jenis DC2 node untuk kinerja terbaik dengan harga terendah. Jika Anda mengharapkan data Anda tumbuh, sebaiknya gunakan RA3 node sehingga Anda dapat mengukur komputasi dan penyimpanan secara independen untuk mencapai peningkatan harga dan kinerja. Anda meluncurkan cluster yang menggunakan tipe DC2 node di virtual private cloud (VPC). Untuk informasi selengkapnya, lihat [Membuat klaster yang disediakan Redshift atau workgroup Amazon Redshift Serverless di VPC](#).

Jenis node tersedia dalam berbagai ukuran. Ukuran node dan jumlah node menentukan total penyimpanan untuk sebuah cluster. Untuk informasi selengkapnya, lihat [Rincian tipe simpul](#).

Beberapa jenis node memungkinkan satu node (single-node) atau dua atau lebih node (multi-node). Jumlah minimum node untuk cluster dari beberapa jenis node adalah dua node. Pada cluster simpul tunggal, node dibagikan untuk fungsionalitas pemimpin dan komputasi. Cluster simpul tunggal tidak

direkomendasikan untuk menjalankan beban kerja produksi. Pada cluster multi-node, node pemimpin terpisah dari node komputasi. Node pemimpin adalah tipe node yang sama dengan node komputasi. Anda hanya membayar untuk node komputasi.

Amazon Redshift menerapkan kuota ke sumber daya untuk setiap AWS akun di setiap Wilayah. AWS Kuota membatasi jumlah sumber daya yang dapat dibuat akun Anda untuk jenis sumber daya tertentu, seperti node atau snapshot, di dalam Wilayah. AWS Untuk informasi selengkapnya tentang kuota default yang berlaku untuk resource Amazon Redshift, lihat. [Kuota dan batas di Amazon Redshift](#)

Biaya cluster Anda tergantung pada AWS Region, tipe node, jumlah node, dan apakah node dicadangkan terlebih dahulu. Untuk informasi selengkapnya tentang biaya node, lihat halaman [harga Amazon Redshift](#).

Rincian tipe simpul

Tabel berikut merangkum spesifikasi node untuk setiap jenis dan ukuran node. Judul dalam tabel memiliki arti ini:

- vCPU adalah jumlah virtual CPUs untuk setiap node.
- RAM adalah jumlah memori dalam gibibytes (GiB) untuk setiap node.
- Irisan default per node adalah jumlah irisan di mana node komputasi dipartisi saat cluster dibuat atau diubah ukurannya dengan pengubahan ukuran klasik.

Jumlah irisan per node dapat berubah jika cluster diubah ukurannya menggunakan pengubahan ukuran elastis. Namun jumlah total irisan pada semua node komputasi di cluster tetap sama setelah mengubah ukuran elastis.

Saat Anda membuat cluster dengan operasi restore from snapshot, jumlah irisan cluster yang dihasilkan mungkin berubah dari cluster asli jika Anda mengubah jenis node.

- Penyimpanan adalah kapasitas dan jenis penyimpanan untuk setiap node.
- Rentang node adalah jumlah node minimum dan maksimum yang didukung Amazon Redshift untuk jenis dan ukuran node.

Note

Anda mungkin dibatasi untuk node yang lebih sedikit tergantung pada kuota yang diterapkan ke AWS akun Anda di AWS Wilayah yang dipilih. Untuk informasi selengkapnya

tentang kuota default yang berlaku untuk resource Amazon Redshift, lihat. [Kuota dan batas di Amazon Redshift](#)

- Kapasitas total adalah kapasitas penyimpanan total untuk cluster jika Anda menggunakan jumlah maksimum node yang ditentukan dalam rentang node.

Tabel berikut menjelaskan spesifikasi untuk RA3 node.

Jenis simpul	vCPU	RAM (GiB)	Irisan default per node	Batas penyimpanan terkelola per node ¹	Rentang node dengan create cluster	Total kapasitas penyimpanan terkelola ²	
ra3.large (simpul tunggal)	2	16	2	1 TB	1	1 TB ³	
ra3.large (multi-node)	2	16	2	8 TB	2-16	128 TB	
ra3.xlplus (simpul tunggal)	4	32	2	4 TB	1	4 TB ³	
ra3.xlplus (multi-node)	4	32	2	32 TB	2—32 ⁴	1024 TB ⁴	
ra3.4xlarge	12	96	4	128 TB	2—32 ⁵	8192 TB ⁵	
ra3.16xlarge	48	384	16	128 TB	2—128	16,384 TB	

¹ Batas penyimpanan untuk penyimpanan terkelola Amazon Redshift. Ini adalah batas yang sulit.

² Total batas penyimpanan terkelola adalah jumlah maksimum node dikalikan batas penyimpanan terkelola per node.

³ Untuk mengubah ukuran cluster simpul tunggal menjadi multi-node, hanya perubahan ukuran klasik yang didukung.

⁴ Anda dapat membuat cluster dengan tipe node `ra3.xplus` (multi-node) yang memiliki hingga 32 node. Untuk cluster multi-node, Anda dapat mengubah ukuran dengan mengubah ukuran elastis hingga maksimum 32 node.

⁵ Anda dapat membuat cluster dengan tipe node `ra3.4xlarge` dengan hingga 32 node. Anda dapat mengubah ukurannya dengan mengubah ukuran elastis hingga maksimum 64 node.

Tabel berikut menjelaskan spesifikasi untuk node komputasi padat.

Jenis simpul	vCPU	RAM (GiB)	Irisan default per node	Penyimpanan per node	Rentang simpul	Kapasitas total	
dc2.large	2	15	2	160 GB NVMe -SSD	1—32	5.12 TB	
dc2.8xlarge	32	244	16	2,56 TB NVMe -SSD	2—128	326 TB	

Note

Jenis node penyimpanan padat (DS2) tidak lagi tersedia.

Nama tipe node sebelumnya

Dalam rilis Amazon Redshift sebelumnya, jenis node tertentu memiliki nama yang berbeda. Anda dapat menggunakan nama sebelumnya di Amazon Redshift API dan AWS CLI. Namun, kami menyarankan Anda memperbarui skrip apa pun yang mereferensikan nama-nama tersebut untuk menggunakan nama saat ini. Nama saat ini dan sebelumnya adalah sebagai berikut.

Nama saat ini	Nama sebelumnya
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

Menentukan jumlah node

Karena Amazon Redshift mendistribusikan dan menjalankan kueri secara paralel di semua node komputasi klaster, Anda dapat meningkatkan kinerja kueri dengan menambahkan node ke cluster Anda. Ketika Anda menjalankan cluster dengan setidaknya dua node komputasi, data pada setiap node dicerminkan pada disk node lain untuk mengurangi risiko terjadinya kehilangan data.

Anda dapat memantau kinerja kueri di konsol Amazon Redshift dan dengan metrik Amazon CloudWatch . Anda juga dapat menambah atau menghapus node sesuai kebutuhan untuk mencapai keseimbangan antara harga dan kinerja untuk klaster Anda. Saat Anda meminta node tambahan, Amazon Redshift menangani semua detail penerapan, penyeimbangan beban, dan pemeliharaan data. Untuk informasi selengkapnya tentang kinerja klaster, lihat [Memantau kinerja klaster Amazon Redshift](#).

Node cadangan sesuai untuk beban kerja produksi kondisi mapan, dan menawarkan diskon signifikan atas node sesuai permintaan. Anda dapat membeli node cadangan setelah menjalankan eksperimen dan proof-of-concepts memvalidasi konfigurasi produksi Anda. Untuk informasi selengkapnya, lihat [Simpul terpesan](#).

Saat menjeda klaster, Anda menanggungkan penagihan sesuai permintaan selama klaster dijeda. Selama waktu jeda ini, Anda hanya membayar untuk penyimpanan cadangan. Ini membebaskan Anda dari perencanaan dan pembelian kapasitas gudang data di depan kebutuhan Anda, dan memungkinkan Anda mengelola lingkungan secara hemat biaya untuk tujuan pengembangan atau pengujian.

Untuk informasi tentang harga node sesuai permintaan dan cadangan, lihat harga [Amazon Redshift](#).

Gunakan EC2 untuk membuat cluster Anda

Cluster Amazon Redshift berjalan di EC2 instans Amazon yang dikonfigurasi untuk jenis dan ukuran node Amazon Redshift yang Anda pilih. Untuk informasi selengkapnya tentang platform jaringan ini, lihat [Platform yang Didukung](#) di Panduan EC2 Pengguna Amazon.

Note

Untuk mencegah masalah koneksi antara alat klien SQL dan database Amazon Redshift, sebaiknya lakukan salah satu dari dua hal. Anda dapat mengonfigurasi aturan masuk yang memungkinkan host untuk menegosiasikan ukuran paket. Atau, Anda dapat menonaktifkan bingkai TCP/IP jumbo dengan mengatur unit transmisi maksimum (MTU) ke 1500 pada antarmuka jaringan (NIC) instans Amazon EC2 Anda. Untuk informasi lebih lanjut tentang pendekatan ini, lihat [Kueri tampaknya hang dan terkadang gagal mencapai cluster](#).

Amazon Virtual Private Cloud (Amazon VPC)

Saat menggunakan Amazon VPC, kluster Anda berjalan di cloud pribadi virtual (VPC) yang secara logis diisolasi ke akun Anda. AWS Jika Anda menyediakan kluster Anda dengan Amazon VPC, Anda mengontrol akses ke kluster Anda dengan mengaitkan satu atau beberapa grup keamanan VPC dengan kluster. Untuk informasi selengkapnya, lihat [Grup Keamanan untuk VPC Anda](#) di Panduan Pengguna Amazon VPC.

Untuk membuat kluster di VPC, Anda harus terlebih dahulu membuat grup subnet kluster Amazon Redshift dengan memberikan informasi subnet VPC Anda, lalu berikan grup subnet saat meluncurkan cluster. Untuk informasi selengkapnya, lihat [Subnet untuk sumber daya Redshift](#).

Untuk informasi selengkapnya tentang Amazon Virtual Private Cloud (Amazon VPC), lihat halaman detail [produk Amazon VPC](#).

Alarm ruang disk default

Saat membuat kluster Amazon Redshift, Anda dapat mengonfigurasi CloudWatch alarm Amazon secara opsional untuk memantau persentase rata-rata ruang disk yang digunakan di semua node di cluster Anda. Kami akan menyebut alarm ini sebagai alarm ruang disk default.

Tujuan dari alarm ruang disk default adalah untuk membantu Anda memantau kapasitas penyimpanan cluster Anda. Anda dapat mengonfigurasi alarm ini berdasarkan kebutuhan gudang

data Anda. Misalnya, Anda dapat menggunakan peringatan sebagai indikator bahwa Anda mungkin perlu mengubah ukuran klaster Anda. Anda dapat mengubah ukuran ke jenis node yang berbeda atau menambahkan node, atau mungkin untuk membeli node cadangan untuk ekspansi masa depan.

Alarm ruang disk default dipicu ketika penggunaan disk mencapai atau melebihi persentase yang ditentukan untuk beberapa kali dan pada durasi tertentu. Secara default, alarm ini terpicu ketika persentase yang Anda tentukan tercapai, dan tetap pada atau di atas persentase itu selama lima menit atau lebih. Anda dapat mengedit nilai default setelah Anda meluncurkan cluster.

Saat CloudWatch alarm dipicu, Amazon Simple Notification Service (Amazon SNS) mengirimkan notifikasi ke penerima tertentu untuk memperingatkan mereka bahwa ambang batas persentase tercapai. Amazon SNS menggunakan topik untuk menentukan penerima dan pesan yang dikirim dalam pemberitahuan. Anda dapat menggunakan topik Amazon SNS yang ada; jika tidak, topik dibuat berdasarkan pengaturan yang Anda tentukan saat meluncurkan klaster. Anda dapat mengedit topik untuk alarm ini setelah Anda meluncurkan cluster. Untuk informasi selengkapnya tentang membuat topik Amazon SNS, lihat [Memulai Layanan Pemberitahuan Sederhana Amazon](#).

Setelah Anda meluncurkan cluster, Anda dapat melihat dan mengedit alarm dari jendela Status cluster di bawah CloudWatch Alarm. Namanya `percentage-disk-space-used-default-<string>`. Anda dapat membuka alarm untuk melihat topik Amazon SNS yang terkait dengannya dan mengedit pengaturan alarm. Jika Anda tidak memilih topik Amazon SNS yang ada untuk digunakan, topik yang dibuat untuk Anda diberi nama `<clustername>-default-alarm (<recipient>)`; misalnya, `(notify@example.com). examplecluster-default-alarms`

Untuk informasi selengkapnya tentang mengonfigurasi dan mengedit alarm ruang disk default, lihat [Membuat kluster](#) dan [Membuat alarm ruang disk](#).

Note

Jika Anda menghapus cluster Anda, alarm yang terkait dengan cluster tidak akan dihapus tetapi tidak akan memicu. Anda dapat menghapus alarm dari CloudWatch konsol jika Anda tidak lagi membutuhkannya.

Status klaster

Status cluster menampilkan status cluster saat ini. Tabel berikut memberikan deskripsi untuk setiap status cluster.

Status	Deskripsi
available	Cluster berjalan dan tersedia.
available, prep-for-resize	Cluster sedang dipersiapkan untuk mengubah ukuran elastis. Cluster berjalan dan tersedia untuk kueri baca dan tulis, tetapi operasi klaster, seperti membuat snapshot, tidak tersedia.
available, resize-cleanup	Operasi pengubahan ukuran elastis menyelesaikan transfer data ke node cluster baru. Cluster berjalan dan tersedia untuk kueri baca dan tulis, tetapi operasi klaster, seperti membuat snapshot, tidak tersedia.
cancelling- resize	Operasi pengubahan ukuran sedang dibatalkan.
creating	Amazon Redshift membuat cluster. Untuk informasi selengkapnya, lihat Membuat kluster .
deleting	Amazon Redshift menghapus cluster. Untuk informasi selengkapnya, lihat Mematikan dan menghapus cluster .
final-snapshot	Amazon Redshift mengambil snapshot terakhir dari cluster sebelum menghapusnya. Untuk informasi selengkapnya, lihat Mematikan dan menghapus cluster .
hardware- failure	Cluster mengalami kegagalan perangkat keras. Jika Anda memiliki cluster simpul tunggal, simpul tidak dapat diganti. Untuk memulihkan cluster Anda, pulihkan snapshot. Untuk informasi selengkapnya, lihat Cuplikan dan cadangan Amazon Redshift .
incompatible- hsm	Amazon Redshift tidak dapat terhubung ke modul keamanan perangkat keras (HSM). Periksa konfigurasi HSM antara cluster dan HSM. Untuk informasi selengkapnya, lihat Enkripsi menggunakan modul keamanan perangkat keras .
incompatible- network	Ada masalah dengan konfigurasi jaringan yang mendasarinya. Pastikan VPC tempat Anda meluncurkan cluster ada dan pengaturannya benar. Untuk informasi selengkapnya, lihat Sumber daya Redshift dalam VPC .

Status	Deskripsi
<code>incompatible-parameters</code>	Ada masalah dengan satu atau lebih nilai parameter dalam grup parameter terkait, dan nilai parameter atau nilai tidak dapat diterapkan. Ubah grup parameter dan perbarui nilai yang tidak valid. Untuk informasi selengkapnya, lihat Grup parameter Amazon Redshift .
<code>incompatible-restore</code>	Ada masalah saat memulihkan cluster dari snapshot. Coba pulihkan cluster lagi dengan snapshot yang berbeda. Untuk informasi selengkapnya, lihat Cuplikan dan cadangan Amazon Redshift .
<code>modifying</code>	Amazon Redshift menerapkan perubahan pada cluster. Untuk informasi selengkapnya, lihat Mengubah klaster .
<code>paused</code>	Cluster dijeda. Untuk informasi selengkapnya, lihat Menjeda dan melanjutkan cluster .
<code>rebooting</code>	Amazon Redshift me-reboot cluster. Untuk informasi selengkapnya, lihat Melakukan boot ulang klaster .
<code>renaming</code>	Amazon Redshift menerapkan nama baru ke cluster. Untuk informasi selengkapnya, lihat Mengganti nama cluster .
<code>resizing</code>	Amazon Redshift mengubah ukuran cluster. Untuk informasi selengkapnya, lihat Mengubah ukuran cluster .
<code>rotating-keys</code>	Amazon Redshift memutar kunci enkripsi untuk cluster. Untuk informasi selengkapnya, lihat Rotasi kunci enkripsi .
<code>storage-full</code>	Cluster telah mencapai kapasitas penyimpanannya. Ubah ukuran cluster untuk menambahkan node atau untuk memilih ukuran node yang berbeda. Untuk informasi selengkapnya, lihat Mengubah ukuran cluster .
<code>updating-hsm</code>	Amazon Redshift memperbarui konfigurasi HSM.

Pertimbangan untuk menggunakan kluster yang disediakan Amazon Redshift

Setelah kluster dibuat, Anda dapat menemukan informasi di bagian ini tentang wilayah tempat fitur tersedia, tugas pemeliharaan, jenis node, dan batas penggunaan.

Pertimbangan Wilayah dan Availability Zone

Amazon Redshift tersedia di beberapa AWS Wilayah. Secara default, Amazon Redshift menyediakan kluster Anda di Availability Zone (AZ) yang dipilih secara acak dalam AWS Wilayah yang Anda pilih. Semua node cluster disediakan di Availability Zone yang sama.

Anda dapat meminta Availability Zone tertentu secara opsional jika Amazon Redshift tersedia di zona tersebut. Misalnya, jika Anda sudah menjalankan instans Amazon EC2 di satu Availability Zone, Anda mungkin ingin membuat kluster Amazon Redshift di zona yang sama untuk mengurangi latensi. Di sisi lain, Anda mungkin ingin memilih Availability Zone lain untuk ketersediaan yang lebih tinggi. Amazon Redshift mungkin tidak tersedia di semua Availability Zone dalam suatu AWS Wilayah.

Untuk daftar AWS Wilayah yang didukung tempat Anda dapat menyediakan kluster Amazon Redshift, lihat titik akhir [Amazon Redshift di](#) bagian. Referensi Umum Amazon Web

Pemeliharaan cluster

Amazon Redshift secara berkala melakukan pemeliharaan untuk menerapkan peningkatan ke kluster Anda. Selama pembaruan ini, kluster Amazon Redshift Anda tidak tersedia untuk operasi normal. Anda memiliki beberapa cara untuk mengontrol cara kami mempertahankan kluster Anda. Misalnya, Anda dapat mengontrol kapan kami menerapkan pembaruan ke kluster Anda. Anda juga dapat memilih apakah kluster Anda menjalankan versi yang paling baru dirilis, atau versi yang dirilis sebelumnya ke versi yang paling baru dirilis. Terakhir, Anda memiliki opsi untuk menunda pembaruan pemeliharaan non-wajib untuk jangka waktu tertentu.

Jendela pemeliharaan

Amazon Redshift menetapkan jendela pemeliharaan 30 menit secara acak dari blok waktu 8 jam per AWS Wilayah, terjadi pada hari acak dalam seminggu (Senin hingga Minggu, inklusif).

Jendela pemeliharaan default

Daftar berikut menunjukkan blok waktu untuk setiap AWS Wilayah dari mana jendela pemeliharaan default ditetapkan:

- Wilayah AS Timur (Virginia N.): 03:00 - 11:00 UTC
- Wilayah AS Timur (Ohio): 03:00 - 11:00 UTC
- Wilayah AS Barat (California N.): 06:00 - 14:00 UTC
- Wilayah AS Barat (Oregon): 06:00 - 14:00 UTC
- Afrika (Cape Town) Wilayah: 20:00 - 04:00 UTC
- Wilayah Asia Pasifik (Hong Kong): 13:00 - 21:00 UTC
- Wilayah Asia Pasifik (Hyderabad): 16:30 — 00:30 UTC
- Wilayah Asia Pasifik (Jakarta): 15:00 - 23:00 UTC
- Wilayah Asia Pasifik (Malaysia): 14:00 - 22:00 UTC
- Wilayah Asia Pasifik (Melbourne): 12:00 - 20:00 UTC
- Wilayah Asia Pasifik (Mumbai): 16:30 — 00:30 UTC
- Wilayah Asia Pasifik (Selandia Baru): 10:00 - 18:00 UTC
- Wilayah Asia Pasifik (Osaka): 13:00 — 21:00 UTC
- Wilayah Asia Pasifik (Seoul): 13:00 - 21:00 UTC
- Wilayah Asia Pasifik (Singapura): 14:00 - 22:00 UTC
- Wilayah Asia Pasifik (Sydney): 12:00 - 20:00 UTC
- Wilayah Asia Pasifik (Taipei): 14:00 - 22:00 UTC
- Wilayah Asia Pasifik (Thailand): 15:00 - 23:00 UTC
- Wilayah Asia Pasifik (Tokyo): 13:00 — 21:00 UTC
- Wilayah Kanada (Tengah): 03:00 - 11:00 UTC
- Kanada Wilayah Barat (Calgary): 04:00 - 12:00 UTC
- Wilayah Tiongkok (Beijing): 13:00 - 21:00 UTC
- Wilayah Tiongkok (Ningxia): 13:00 - 21:00 UTC
- Wilayah Eropa (Frankfurt): 06:00 - 14:00 UTC
- Wilayah Eropa (Irlandia): 22:00 - 06:00 UTC
- Wilayah Eropa (London): 22:00 - 06:00 UTC
- Wilayah Eropa (Milan): 21:00 - 05:00 UTC
- Wilayah Eropa (Paris): 23:00 - 07:00 UTC

- Wilayah Eropa (Stockholm): 23:00 - 07:00 UTC
- Wilayah Eropa (Zurich): 20:00 - 04:00 UTC
- Wilayah Israel (Tel Aviv): 20:00 — 04:00 UTC
- Wilayah Meksiko (Tengah): 04:00 - 12:00 UTC
- Wilayah Eropa (Spanyol): 21:00 - 05:00 UTC
- Wilayah Timur Tengah (Bahrain): 13:00 - 21:00 UTC
- Wilayah Timur Tengah (UEA): 18:00 - 02:00 UTC
- Wilayah Amerika Selatan (São Paulo): 19:00 - 03:00 UTC

Jika acara pemeliharaan dijadwalkan untuk minggu tertentu, itu dimulai selama jendela pemeliharaan 30 menit yang ditetapkan. Saat Amazon Redshift melakukan pemeliharaan, Amazon Redshift menghentikan kueri atau operasi lain yang sedang berlangsung. Waktu henti yang dialami selama pemeliharaan terencana tidak dianggap sebagai waktu henti bulanan atau tidak tersedianya dalam Perjanjian Tingkat Layanan [Amazon Redshift](#). Sebagian besar pemeliharaan selesai selama jendela pemeliharaan 30 menit, tetapi beberapa tugas pemeliharaan mungkin terus berjalan setelah jendela ditutup. Jika tidak ada tugas pemeliharaan yang harus dilakukan selama jendela pemeliharaan terjadwal, kluster Anda akan terus beroperasi secara normal hingga jendela pemeliharaan terjadwal berikutnya.

Anda dapat mengubah jendela pemeliharaan terjadwal dengan memodifikasi cluster, baik secara terprogram atau dengan menggunakan konsol Amazon Redshift. Anda dapat menemukan jendela pemeliharaan dan mengatur hari dan waktu itu terjadi untuk cluster di bawah tab Maintenance.

Hal ini dimungkinkan untuk sebuah cluster untuk restart di luar jendela pemeliharaan. Ada beberapa alasan ini bisa terjadi. Satu lagi alasan umum adalah bahwa masalah telah terdeteksi dengan cluster dan operasi pemeliharaan sedang dilakukan untuk mengembalikannya ke keadaan sehat. Untuk informasi lebih lanjut, lihat artikel [Mengapa cluster Amazon Redshift saya reboot di luar jendela pemeliharaan?](#), yang memberikan rincian tentang mengapa ini mungkin terjadi.

Menunda pemeliharaan

Untuk menjadwalkan ulang jendela pemeliharaan kluster Anda, Anda dapat menunda pemeliharaan hingga 60 hari. Misalnya, jika jendela pemeliharaan kluster Anda diatur ke Rabu 08:30 — 09:00 UTC dan Anda perlu mengakses cluster Anda pada saat itu, Anda dapat menunda pemeliharaan ke periode waktu berikutnya.

Jika Anda menunda pemeliharaan, Amazon Redshift akan tetap menerapkan pembaruan perangkat keras atau pembaruan keamanan wajib lainnya ke cluster Anda. Cluster Anda tidak tersedia selama pembaruan ini.

Jika pembaruan perangkat keras atau pembaruan keamanan wajib lainnya dijadwalkan selama jendela pemeliharaan yang akan datang, Amazon Redshift mengirimkan Anda pemberitahuan terlebih dahulu di bawah kategori Tertunda. Untuk mempelajari lebih lanjut tentang Pemberitahuan acara Tertunda, lihat [Pemberitahuan acara klaster yang disediakan Amazon Redshift](#).

Anda juga dapat memilih untuk menerima pemberitahuan acara dari Amazon Simple Notification Service (Amazon SNS). Untuk informasi selengkapnya tentang berlangganan notifikasi acara dari Amazon SNS, lihat [Langganan pemberitahuan acara klaster Amazon Redshift](#)

Jika Anda menunda pemeliharaan klaster Anda, jendela pemeliharaan setelah periode penundaan tidak dapat ditangguhkan.

Note

Anda tidak dapat menunda pemeliharaan setelah dimulai.

Untuk informasi selengkapnya tentang pemeliharaan klaster, lihat dokumentasi berikut:

- [Jendela pemeliharaan](#)
- [Operasi klaster](#)
- [Mengubah klaster](#)

Memilih trek pemeliharaan cluster

Saat Amazon Redshift merilis versi cluster baru, klaster Anda diperbarui selama jendela pemeliharaannya. Anda dapat mengontrol apakah klaster Anda diperbarui ke rilis terbaru atau ke rilis sebelumnya.

Track mengontrol versi cluster mana yang diterapkan selama jendela pemeliharaan. Saat Amazon Redshift merilis versi cluster baru, versi tersebut ditetapkan ke trek saat ini, dan versi sebelumnya ditetapkan ke trailing track.

Untuk informasi tentang trek klaster, lihat [Trek untuk klaster yang disediakan Amazon Redshift dan grup kerja tanpa server](#).

Memahami bagaimana RA3 node memisahkan komputasi dan penyimpanan

Bagian ini merinci tugas yang tersedia untuk tipe RA3 node, menunjukkan penerapannya pada kumpulan kasus penggunaan dan merinci keunggulannya dibandingkan jenis node yang tersedia sebelumnya.

Keuntungan dan ketersediaan RA3 node

RA3 node memberikan keuntungan sebagai berikut:

- Mereka fleksibel untuk menumbuhkan kapasitas komputasi Anda tanpa meningkatkan biaya penyimpanan Anda. Dan mereka menskalakan penyimpanan Anda tanpa menyediakan kapasitas komputasi yang berlebihan.
- Mereka menggunakan kinerja tinggi SSDs untuk data panas Anda dan Amazon S3 untuk data dingin. Dengan demikian mereka memberikan kemudahan penggunaan, penyimpanan hemat biaya, dan kinerja kueri yang tinggi.
- Mereka menggunakan jaringan bandwidth tinggi yang dibangun di atas Sistem AWS Nitro untuk lebih mengurangi waktu yang dibutuhkan untuk data yang akan diturunkan dan diambil dari Amazon S3.

Pertimbangkan untuk memilih jenis RA3 node dalam kasus ini:

- Anda memerlukan fleksibilitas untuk menskalakan dan membayar komputasi terpisah dari penyimpanan.
- Anda menanyakan sebagian kecil dari total data Anda.
- Volume data Anda berkembang pesat atau diperkirakan akan tumbuh dengan cepat.
- Anda menginginkan fleksibilitas untuk mengukur cluster hanya berdasarkan kebutuhan kinerja Anda.

Untuk menggunakan tipe RA3 node, AWS Region Anda harus mendukung RA3. Untuk informasi selengkapnya, lihat [RA3 ketersediaan tipe simpul di AWS Wilayah](#).

Important

Anda dapat menggunakan tipe node `ra3.xlplus` hanya dengan versi cluster 1.0.21262 atau yang lebih baru. Anda dapat melihat versi cluster yang ada dengan konsol Amazon Redshift. Untuk informasi selengkapnya, lihat [Menentukan versi workgroup atau cluster](#).

Pastikan Anda menggunakan konsol Amazon Redshift baru saat bekerja dengan tipe RA3 node.

Selain itu, untuk menggunakan tipe RA3 node dengan operasi Amazon Redshift yang menggunakan track, nilai track pemeliharaan harus disetel ke versi cluster yang mendukung.

RA3 Untuk informasi selengkapnya tentang trek, lihat [Memilih trek pemeliharaan cluster](#).

Pertimbangkan hal berikut saat menggunakan tipe simpul RA3 simpul tunggal.

- Produsen dan konsumen Datasharing didukung.
- Untuk mengubah tipe node, hanya pengubahan ukuran klasik yang didukung. Mengubah tipe node dengan pengubahan ukuran elastis atau pemulihan snapshot tidak didukung. Skenario berikut didukung:
 - Pengubahan ukuran klasik dari 1-node dc2.xlarge menjadi 1-node ra3.xlplus, dan sebaliknya.
 - Pengubahan ukuran klasik dari 1-node dc2.xlarge menjadi multiple-node ra3.xlplus, dan sebaliknya.
 - Pengubahan ukuran klasik dari multiple-node dc2.xlarge menjadi 1-node ra3.xlplus, dan sebaliknya.

Bekerja dengan penyimpanan terkelola Amazon Redshift

Dengan penyimpanan terkelola Amazon Redshift, Anda dapat menyimpan dan memproses semua data di Amazon Redshift sambil mendapatkan lebih banyak fleksibilitas untuk menskalakan kapasitas komputasi dan penyimpanan secara terpisah. Anda terus menelan data dengan perintah COPY atau INSERT. Untuk mengoptimalkan kinerja dan mengelola penempatan data otomatis di seluruh tingkatan penyimpanan, Amazon Redshift memanfaatkan pengoptimalan seperti suhu blok data, usia blok data, dan pola beban kerja. Bila diperlukan, Amazon Redshift menskalakan penyimpanan secara otomatis ke Amazon S3 tanpa memerlukan tindakan manual apa pun.

Untuk informasi tentang biaya penyimpanan, lihat [harga Amazon Redshift](#).

Mengelola jenis RA3 simpul

Untuk memanfaatkan pemisahan komputasi dari penyimpanan, Anda dapat membuat atau meningkatkan klaster Anda dengan tipe node RA3. Untuk menggunakan tipe node RA3, buat cluster Anda di cloud pribadi virtual (EC2-VPC).

Untuk mengubah jumlah node cluster Amazon Redshift dengan tipe RA3 node, lakukan salah satu hal berikut:

- Tambahkan atau hapus node dengan operasi pengubahan ukuran elastis. Dalam beberapa situasi, menghapus node dari RA3 cluster tidak diperbolehkan dengan pengubahan ukuran elastis. Misalnya, ketika peningkatan jumlah node 2:1 menempatkan jumlah irisan per node pada 32. Untuk informasi selengkapnya, lihat [Mengubah ukuran cluster](#). Jika pengubahan ukuran elastis tidak tersedia, gunakan pengubahan ukuran klasik.
- Tambahkan atau hapus node dengan operasi pengubahan ukuran klasik. Pilih opsi ini saat Anda mengubah ukuran ke konfigurasi yang tidak tersedia melalui pengubahan ukuran elastis. Pengubahan ukuran elastis lebih cepat daripada pengubahan ukuran klasik. Untuk informasi selengkapnya, lihat [Mengubah ukuran cluster](#).

RA3 ketersediaan tipe simpul di AWS Wilayah

Jenis RA3 node hanya tersedia di AWS Wilayah berikut:

- Wilayah AS Timur (Virginia N.) (us-east-1)
- Wilayah AS Timur (Ohio) (us-east-2)
- Wilayah AS Barat (California N.) (us-west-1)
- Wilayah AS Barat (Oregon) (us-west-2)
- Wilayah Afrika (Cape Town) (af-south-1)
- Wilayah Asia Pasifik (Hong Kong) (ap-east-1)
- Wilayah Asia Pasifik (Hyderabad) (ap-south-2)
- Wilayah Asia Pasifik (Jakarta) (ap-southeast-3)
- Wilayah Asia Pasifik (Malaysia) (ap-tenggara 5)
- Wilayah Asia Pasifik (Melbourne) (ap-southeast-4)
- Wilayah Asia Pasifik (Mumbai) (ap-south-1)
- Wilayah Asia Pasifik (Selandia Baru) (ap-tenggara 6)
- Wilayah Asia Pasifik (Osaka) (ap-northeast-3)
- Wilayah Asia Pasifik (Seoul) (ap-northeast-2)
- Wilayah Asia Pasifik (Singapura) (ap-southeast-1)
- Wilayah Asia Pasifik (Sydney) (ap-southeast-2)

- Wilayah Asia Pasifik (Taipei) (ap-timur-2)
- Wilayah Asia Pasifik (Thailand) (ap-tenggara 7)
- Wilayah Asia Pasifik (Tokyo) (ap-northeast-1)
- Wilayah Kanada (Tengah) (ca-central-1)
- Wilayah Kanada Barat (Calgary) (ca-west-1)
- Wilayah Tiongkok (Beijing) (cn-utara-1)
- Wilayah Tiongkok (Ningxia) (cn-barat laut-1)
- Wilayah Eropa (Frankfurt) (eu-central-1)
- Wilayah Eropa (Zurich) (eu-central-2)
- Wilayah Eropa (Irlandia) (eu-west-1)
- Wilayah Eropa (London) (eu-west-2)
- Wilayah Eropa (Milan) (eu-south-1)
- Wilayah Eropa (Spanyol) (eu-south-2)
- Wilayah Eropa (Paris) (eu-west-3)
- Wilayah Eropa (Stockholm) (eu-north-1)
- Wilayah Israel (Tel Aviv) (il-central-1)
- Wilayah Meksiko (Tengah) (mx-central-1)
- Wilayah Timur Tengah (Bahrain) (me-south-1)
- Wilayah Timur Tengah (UEA) (me-central-1)
- Wilayah Amerika Selatan (São Paulo) (sa-east-1)
- AWS GovCloud (AS-Timur) (us-gov-east-1)
- AWS GovCloud (AS-Barat) (us-gov-west-1)

Memutakhirkan ke tipe RA3 node

Untuk memutakhirkan tipe node yang ada RA3, Anda memiliki opsi berikut untuk mengubah jenis node:

- Pulihkan dari snapshot — Amazon Redshift menggunakan snapshot terbaru dari cluster Anda dan mengembalikannya untuk membuat cluster baru. RA3 Segera setelah pembuatan cluster selesai (biasanya dalam beberapa menit), RA3 node siap untuk menjalankan beban kerja produksi penuh Anda. Karena komputasi terpisah dari penyimpanan, data panas dibawa ke cache lokal dengan

kecepatan cepat berkat bandwidth jaringan yang besar. Jika Anda memulihkan dari DC2 snapshot terbaru, RA3 menyimpan informasi blok panas dari DC2 beban kerja dan mengisi cache lokalnya dengan blok terpanas. Untuk informasi selengkapnya, lihat [Memulihkan cluster dari snapshot](#).

Untuk menjaga endpoint yang sama untuk aplikasi dan pengguna Anda, Anda dapat mengganti nama RA3 cluster baru dengan nama yang sama dengan cluster asli DC2. Untuk mengganti nama cluster, ubah cluster di konsol Amazon Redshift ModifyCluster atau operasi API. Untuk informasi selengkapnya, lihat [Mengganti nama cluster](#) atau [operasi ModifyCluster API](#) di Referensi Amazon Redshift API.

- Ubah ukuran elastis — mengubah ukuran cluster menggunakan pengubahan ukuran elastis. Saat Anda menggunakan pengubahan ukuran elastis untuk mengubah jenis node, Amazon Redshift secara otomatis membuat snapshot, membuat cluster baru, menghapus klaster lama, dan mengganti nama cluster baru. Operasi pengubahan ukuran elastis dapat dijalankan sesuai permintaan atau dapat dijadwalkan untuk berjalan di masa mendatang. Anda dapat dengan cepat memutakhirkan cluster tipe DC2 node yang ada RA3 dengan pengubahan ukuran elastis. Untuk informasi selengkapnya, lihat [Ubah ukuran elastis](#).

Tabel berikut menunjukkan rekomendasi saat memutakhirkan ke tipe RA3 node. (Rekomendasi ini juga berlaku untuk node yang dicadangkan.)

Rekomendasi dalam tabel ini adalah memulai jenis dan ukuran node cluster tetapi bergantung pada persyaratan komputasi beban kerja Anda. Untuk memperkirakan kebutuhan Anda dengan lebih baik, pertimbangkan untuk melakukan bukti konsep (POC) yang menggunakan [Test Drive](#) untuk menjalankan konfigurasi potensial. Menyediakan cluster untuk gudang data POC Anda alih-alih Redshift Serverless. Untuk informasi selengkapnya tentang melakukan pembuktian konsep, lihat [Melakukan bukti konsep \(POC\) untuk Amazon Redshift](#) di Panduan Pengembang Basis Data Amazon Redshift.

Jenis node yang ada	Jumlah node yang ada	Jenis node baru yang direkomendasikan	Upgrade tindakan	
dc2.8xlarge	2—15	ra3.4xlarge	Mulailah dengan 2 node ra3.4xlarge untuk setiap 1 node dc2.8xlarge 1.	

Jenis node yang ada	Jumlah node yang ada	Jenis node baru yang direkomen dasikan	Upgrade tindakan	
dc2.8xlarge	16—128	ra3.16xlarge	Mulailah dengan 1 node ra3.16xlarge untuk setiap 2 node dc2.8xlarge 1.	
dc2.large	1—4	ra3. besar	Mulailah dengan 1 node ra3.large untuk setiap 1 node dc2.large 1. Mulailah dengan 2 node ra3.large untuk setiap 2 node dc2.large 1. Mulailah dengan 3 node ra3.large untuk setiap 3 node dc2.large 1. Mulailah dengan 3 node ra3.large untuk setiap 4 node dc2.large 1.	
dc2.large	5—15	ra3.xlplus	Mulailah dengan 3 node ra3.xlplus untuk setiap 8 node dc2.large 1.	
dc2.large	16—32	ra3.4xlarge	Mulailah dengan 1 node ra3.4xlarge untuk setiap 8 node dc2.large 1, 2.	

¹ Node tambahan mungkin diperlukan tergantung pada persyaratan beban kerja. Menambahkan atau menghapus node berdasarkan persyaratan komputasi dari kinerja kueri yang Anda butuhkan.

² Cluster dengan tipe node dc2.large dibatasi hingga 32 node.

Jumlah minimum node untuk beberapa tipe RA3 node adalah 2 node. Pertimbangkan hal ini saat membuat RA3 cluster.

Fitur jaringan yang didukung oleh RA3 node

RA3 node mendukung kumpulan fitur jaringan yang tidak tersedia untuk jenis node lainnya. Bagian ini memberikan deskripsi singkat dari setiap fitur dan tautan ke dokumentasi tambahan:

- Titik akhir VPC klaster yang disediakan — Saat Anda membuat atau memulihkan klaster RA3 , Amazon Redshift menggunakan port dalam rentang 5431-5455 atau 8191-8215. Saat cluster disetel ke port di salah satu rentang ini, Amazon Redshift secara otomatis membuat titik akhir VPC di AWS akun Anda untuk cluster dan melampirkan alamat IP pribadi ke dalamnya. Jika Anda menyetel klaster agar dapat diakses publik, Redshift akan membuat alamat IP elastis di AWS akun Anda dan menempelkannya ke titik akhir VPC. Untuk informasi selengkapnya, lihat [Mengonfigurasi setelan komunikasi grup keamanan untuk klaster Amazon Redshift atau grup kerja Amazon Redshift Tanpa Server](#).
- RA3 Cluster subnet tunggal — Anda dapat membuat RA3 cluster dengan subnet tunggal, tetapi tidak dapat menggunakan fitur pemulihan bencana. Pengecualian terjadi jika Anda mengaktifkan relokasi klaster ketika subnet tidak memiliki beberapa Availability Zones (AZs).
- RA3 Kluster multi-subnet dan grup subnet — Anda dapat membuat RA3 klaster dengan beberapa subnet dengan membuat grup subnet saat Anda menyediakan cluster di virtual private cloud (VPC) Anda. Grup subnet cluster memungkinkan Anda menentukan satu set subnet di VPC Anda dan Amazon Redshift membuat cluster di salah satunya. Setelah membuat grup subnet, Anda dapat menghapus subnet yang sebelumnya Anda tambahkan, atau menambahkan lebih banyak. Untuk informasi selengkapnya, lihat [grup subnet klaster Amazon Redshift](#).
- Akses lintas-akun atau lintas titik akhir VPC — Anda dapat mengakses klaster yang disediakan atau grup kerja Tanpa Server Amazon Redshift dengan menyiapkan titik akhir VPC yang dikelola RedShift. Anda dapat mengaturnya sebagai koneksi pribadi antara VPC yang berisi cluster atau workgroup dan VPC tempat Anda menjalankan alat klien, misalnya. Dengan melakukan ini, Anda dapat mengakses gudang data tanpa menggunakan alamat IP publik dan tanpa merutekan lalu lintas melalui internet. Untuk informasi selengkapnya, lihat [Bekerja dengan titik akhir VPC yang dikelola RedShift](#).

- Relokasi cluster — Anda dapat memindahkan cluster ke Availability Zone (AZ) lain tanpa kehilangan data ketika ada gangguan layanan. Anda mengaktifkannya di konsol. Untuk informasi selengkapnya, lihat [Merelokasi cluster](#).
- Nama domain khusus - Anda dapat membuat nama domain khusus, juga dikenal sebagai URL khusus, untuk klaster Amazon Redshift Anda. Ini adalah catatan easy-to-read DNS yang merutekan koneksi SQL-client ke endpoint cluster Anda. Lihat informasi yang lebih lengkap di [Nama domain khusus untuk koneksi klien](#).

Operasi klaster

Setelah membuat klaster, Anda dapat melakukan operasi klaster untuk mengoptimalkan kinerja, mengontrol biaya, dan memastikan ketersediaan tinggi. Operasi klaster memungkinkan Anda mengubah ukuran, menjeda, melanjutkan, atau bahkan membuat ulang cluster saat kebutuhan pergudangan data Anda berkembang.

Kasus penggunaan umum termasuk penskalaan kapasitas komputasi untuk beban kerja puncak, menjeda cluster selama periode tidak aktif untuk mengurangi biaya, dan membuat ulang cluster dengan konfigurasi berbeda atau di Availability Zone yang berbeda untuk pemulihan bencana. Bagian berikut mencakup detail pelaksanaan berbagai operasi klaster untuk mengelola lingkungan Amazon Redshift Anda secara efektif.

Membuat kluster

Dengan Amazon Redshift, Anda dapat membuat klaster yang disediakan untuk meluncurkan gudang data baru. Cluster yang disediakan adalah kumpulan sumber daya komputasi yang disebut node, yang diatur ke dalam sistem tunggal, massively parallel processing (MPP).

Sebelum Anda membuat cluster, baca [Cluster yang disediakan Amazon Redshift](#) dan [Cluster dan node di Amazon Redshift](#).

Untuk membuat klaster DB

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster. Cluster untuk akun Anda di AWS Wilayah saat ini terdaftar. Subset properti dari setiap cluster ditampilkan dalam kolom dalam daftar.
3. Pilih Buat cluster untuk membuat cluster.
4. Ikuti petunjuk di halaman konsol untuk memasukkan properti untuk konfigurasi Cluster.

Langkah berikut menjelaskan konsol Amazon Redshift yang berjalan di konsol AWS Region yang mendukung tipe RA3 node. Untuk daftar jenis RA3 node pendukung Wilayah AWS tersebut, lihat [Ringkasan jenis RA3 node](#) di Panduan Manajemen Amazon Redshift.

Jika Anda tidak tahu seberapa besar ukuran cluster Anda, pilih Bantu saya pilih. Melakukan hal ini memulai kalkulator ukuran yang menanyakan pertanyaan tentang ukuran dan karakteristik kueri data yang Anda rencanakan untuk disimpan di gudang data Anda. Jika Anda mengetahui ukuran cluster Anda yang diperlukan (yaitu, jenis node dan jumlah node), pilih yang akan saya pilih. Kemudian pilih jenis Node dan jumlah Node untuk mengukur cluster Anda untuk bukti konsep.

 Note

Jika organisasi Anda memenuhi syarat dan klaster Anda sedang dibuat di AWS Region tempat Amazon Redshift Tanpa Server tidak tersedia, Anda mungkin dapat membuat klaster di bawah program uji coba gratis Amazon Redshift. Pilih salah satu Produksi atau Uji coba gratis untuk menjawab pertanyaan Untuk apa Anda berencana menggunakan cluster ini? Saat Anda memilih Uji coba gratis, Anda membuat konfigurasi dengan tipe simpul dc2.large. Untuk informasi selengkapnya tentang memilih uji coba gratis, lihat uji coba [gratis Amazon Redshift](#). [Untuk daftar Wilayah AWS tempat Amazon Redshift Serverless tersedia, lihat titik akhir yang terdaftar untuk Redshift Serverless API di Referensi Umum Amazon Web Services](#)

5. Di bagian konfigurasi Database, tentukan nilai untuk nama pengguna Admin. Untuk kata sandi Admin, Anda dapat memilih dari opsi berikut:
 - Buat kata sandi — Gunakan kata sandi yang dihasilkan oleh Amazon Redshift.
 - Tambahkan kata sandi admin secara manual — Gunakan kata sandi Anda sendiri.
 - Kelola kredensi admin di AWS Secrets Manager — Amazon Redshift AWS Secrets Manager menggunakan untuk membuat dan mengelola kata sandi admin Anda. Menggunakan AWS Secrets Manager untuk menghasilkan dan mengelola rahasia kata sandi Anda menimbulkan biaya. Untuk informasi tentang AWS Secrets Manager harga, lihat [AWS Secrets Manager Harga](#).
6. (Opsional) Ikuti petunjuk di halaman konsol untuk memasukkan properti untuk izin Cluster. Berikan izin klaster jika klaster Anda perlu mengakses AWS layanan lain untuk Anda, misalnya untuk memuat data dari Amazon S3.

7. Pilih Buat cluster untuk membuat cluster. Cluster mungkin membutuhkan waktu beberapa menit untuk siap digunakan.

Konfigurasi tambahan

Saat membuat cluster, Anda dapat menentukan properti tambahan untuk menyesuaikannya. Anda dapat menemukan detail lebih lanjut tentang beberapa properti ini dalam daftar berikut.

Jenis alamat IP

Pilih jenis alamat IP untuk cluster Anda. Anda dapat memilih agar sumber daya Anda berkomunikasi hanya melalui protokol IPv4 pengalamatan, atau memilih mode dual-stack, yang memungkinkan sumber daya Anda berkomunikasi melalui keduanya dan. IPv4 IPv6 Fitur ini hanya tersedia di Wilayah AWS GovCloud (AS-Timur) dan AWS GovCloud (AS-Barat). Untuk informasi selengkapnya tentang AWS Wilayah, lihat [Wilayah dan Zona Ketersediaan](#).

Cloud privat virtual (VPC)

Pilih VPC yang memiliki grup subnet cluster. Setelah cluster dibuat, grup subnet cluster tidak dapat diubah.

Grup parameter

Pilih grup parameter cluster untuk mengasosiasikan dengan cluster. Jika Anda tidak memilih salah satu, cluster menggunakan grup parameter default.

Enkripsi

Pilih apakah Anda ingin mengenkripsi semua data dalam cluster dan snapshot-nya. Jika Anda meninggalkan pengaturan default, None, enkripsi tidak diaktifkan. Jika Anda ingin mengaktifkan enkripsi, pilih apakah Anda ingin menggunakan AWS Key Management Service (AWS KMS) atau modul keamanan perangkat keras (HSM), lalu konfigurasi pengaturan terkait. Untuk informasi selengkapnya tentang enkripsi di Amazon Redshift, lihat [Enkripsi basis data Amazon Redshift](#)

- KMS

Pilih Use AWS Key Management Service (AWS KMS) jika Anda ingin mengaktifkan enkripsi dan gunakan AWS KMS untuk mengelola kunci enkripsi Anda. Juga, pilih kunci yang akan digunakan. Anda dapat memilih kunci default, kunci dari akun saat ini, atau kunci dari akun lain.

 Note

Jika Anda ingin menggunakan kunci dari AWS akun lain, masukkan Nama Sumber Daya Amazon (ARN) untuk kunci yang akan digunakan. Anda harus memiliki izin untuk menggunakan kunci. Untuk informasi selengkapnya tentang akses ke kunci AWS KMS, lihat [Mengontrol akses ke kunci Anda](#) di Panduan AWS Key Management Service Pengembang.

Untuk informasi selengkapnya tentang menggunakan kunci AWS KMS enkripsi di Amazon Redshift, lihat. [Enkripsi menggunakan AWS KMS](#)

- HSM

Pilih HSM jika Anda ingin mengaktifkan enkripsi dan menggunakan modul keamanan perangkat keras (HSM) untuk mengelola kunci enkripsi Anda.

Jika Anda memilih HSM, pilih nilai dari Koneksi HSM dan Sertifikat Klien HSM. Nilai-nilai ini diperlukan untuk Amazon Redshift dan HSM untuk membentuk koneksi tepercaya di mana kunci cluster dapat diteruskan. Koneksi HSM dan sertifikat klien harus diatur di Amazon Redshift sebelum Anda meluncurkan kluster. Untuk informasi selengkapnya tentang pengaturan koneksi HSM dan sertifikat klien, lihat [Enkripsi menggunakan modul keamanan perangkat keras](#).

Jalur pemeliharaan

Anda dapat memilih apakah versi cluster yang digunakan adalah trek Current, Trailing, atau terkadang Preview.

Pemantauan

Anda dapat memilih apakah akan membuat CloudWatch alarm.

Konfigurasi snapshot lintas wilayah

Anda dapat memilih apakah akan mengaktifkan snapshot lintas wilayah.

Periode retensi snapshot otomatis

Anda dapat memilih jumlah hari untuk menyimpan snapshot ini dalam waktu 35 hari. Jika jenis node adalah DC2, Anda dapat memilih nol (0) hari untuk tidak membuat snapshot otomatis.

Periode retensi snapshot manual

Anda dapat memilih jumlah hari atau Indefinitely untuk mempertahankan snapshot ini.

Sumber daya komputasi tambahan untuk pengoptimalan otomatis

Anda dapat memilih apakah akan mengalokasikan sumber daya komputasi tambahan untuk melakukan pengoptimalan otomatis, bahkan selama periode penggunaan berat. Untuk informasi selengkapnya, lihat [Mengalokasikan sumber daya komputasi tambahan untuk pengoptimalan database otomatis](#) di Panduan Pengembang Database Amazon Redshift.

Membuat alarm ruang disk

Anda dapat memantau penggunaan ruang disk dan menyetel alarm untuk diberi tahu saat ruang disk melebihi ambang batas yang ditentukan untuk cluster. Membuat alarm penggunaan ruang disk memungkinkan Anda untuk secara proaktif mengelola kapasitas penyimpanan dan mencegah masalah yang disebabkan oleh ruang disk yang tidak mencukupi, seperti kegagalan kueri atau kesalahan konsumsi data. Prosedur berikut memandu Anda melalui proses pembuatan alarm penggunaan ruang disk.

Untuk membuat alarm penggunaan ruang disk untuk cluster

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Alarm.
3. Untuk Tindakan, pilih Buat alarm. Halaman Create alarm muncul.
4. Ikuti instruksi di halaman.
5. Pilih Buat alarm.

Melihat cluster

Melihat klaster memungkinkan Anda memantau dan mengelola konfigurasi, status, dan metrik kinerja klaster Anda. Dengan melihat detail klaster, Anda dapat memperoleh wawasan tentang pemanfaatan sumber daya, waktu eksekusi kueri, dan kesehatan sistem. Prosedur berikut menunjukkan cara mengakses informasi cluster.

Untuk melihat cluster

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>

2. Pada menu navigasi, pilih Cluster. Cluster untuk akun Anda di AWS Wilayah saat ini terdaftar. Subset properti dari setiap cluster ditampilkan dalam kolom dalam daftar. Jika Anda tidak memiliki cluster apa pun, pilih Buat cluster untuk membuatnya.
3. Pilih nama cluster dalam daftar untuk melihat detail selengkapnya tentang cluster.

Mengubah klaster

Saat Anda memodifikasi klaster, perubahan pada opsi berikut akan segera diterapkan:

- Grup keamanan VPC
- Dapat diakses secara publik
- Kata sandi pengguna admin
- Koneksi HSM
- Sertifikat Klien HSM
- Detail pemeliharaan
- Preferensi snapshot

Perubahan pada opsi berikut berlaku hanya setelah cluster dimulai ulang:

- Pengidentifikasi cluster

Amazon Redshift memulai ulang cluster secara otomatis saat Anda mengubah pengenalan Cluster.

- Perutean VPC yang disempurnakan

Amazon Redshift memulai ulang cluster secara otomatis saat Anda mengubah perutean VPC yang Ditingkatkan.

- Kelompok parameter cluster
- Jenis alamat IP

Fitur ini hanya tersedia di Wilayah AWS GovCloud (AS-Timur) dan AWS GovCloud (AS-Barat).

Untuk informasi selengkapnya tentang AWS Wilayah, lihat [Wilayah dan Zona Ketersediaan](#).

Jika Anda mengurangi periode retensi snapshot otomatis, snapshot otomatis yang ada yang pengaturannya berada di luar periode retensi baru akan dihapus. Untuk informasi selengkapnya, lihat [Cuplikan dan cadangan Amazon Redshift](#).

Untuk informasi selengkapnya tentang properti cluster, lihat [Konfigurasi tambahan](#).

Untuk mengubah klaster

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster.
3. Pilih cluster yang akan dimodifikasi.
4. Pilih Edit. Halaman Edit cluster muncul.
5. Perbarui properti cluster. Beberapa properti yang dapat Anda modifikasi adalah:
 - Pengidentifikasi klaster
 - Retensi snapshot
 - Relokasi cluster

Untuk mengedit pengaturan untuk konfigurasi Jaringan dan keamanan, Pemeliharaan, dan Database, konsol menyediakan tautan ke tab detail klaster yang sesuai.

6. Pilih Simpan perubahan.

Mengubah ukuran cluster

Karena kapasitas dan kinerja pergudangan data Anda perlu berubah, Anda dapat mengubah ukuran klaster Anda untuk memanfaatkan opsi komputasi dan penyimpanan Amazon Redshift sebaik mungkin.

Saat Anda mengubah ukuran cluster, Anda menentukan sejumlah node atau tipe node yang berbeda dari konfigurasi cluster saat ini. Saat cluster sedang dalam proses mengubah ukuran, Anda tidak dapat menjalankan penulisan atau read/write kueri apa pun di cluster; Anda hanya dapat menjalankan kueri baca.

Untuk informasi lebih lanjut tentang mengubah ukuran cluster, termasuk berjalan melalui proses mengubah ukuran cluster menggunakan pendekatan yang berbeda, lihat. [Mengubah ukuran cluster](#)

Untuk mengubah ukuran cluster

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>

2. Pada menu navigasi, pilih Cluster.
3. Pilih cluster untuk mengubah ukuran.
4. Untuk Tindakan, pilih Ubah Ukuran. Halaman cluster Resize muncul.
5. Ikuti instruksi di halaman. Anda dapat mengubah ukuran cluster sekarang, sekali pada waktu tertentu, atau menambah dan mengurangi ukuran cluster Anda sesuai jadwal.
6. Tergantung pada pilihan Anda, pilih Ubah ukuran sekarang atau Jadwalkan Ubah ukuran.

Jika Anda memiliki node cadangan, Anda dapat meningkatkan ke node yang RA3 dicadangkan. Anda dapat melakukan ini ketika Anda menggunakan konsol untuk memulihkan dari snapshot atau untuk melakukan pengubahan ukuran elastis. Anda dapat menggunakan konsol untuk memandu Anda melalui proses ini. Untuk informasi selengkapnya tentang memutakhirkan ke RA3 node, lihat [Memutakhirkan ke tipe RA3 node](#).

Saat Anda melakukan operasi pengubahan ukuran untuk memutakhirkan dari tipe DC2 simpul.large ke tipe node RA3 .large, Amazon Redshift secara otomatis mengonversi kunci sortir interleaved menjadi kunci sortir majemuk. Konversi ini memungkinkan akses ke fitur penskalaan konkurensi, yang tidak mendukung kueri pada tabel dengan kunci pengurutan yang disisipkan. Meskipun konversi otomatis ini memastikan kompatibilitas dengan RA3 fitur, ini dapat memengaruhi pola kinerja kueri yang ada.

Jika Anda ingin mempertahankan kunci sortir interleaved setelah memutakhirkan ke RA3 node, Anda dapat membuat ulang tabel Anda dengan konfigurasi kunci pengurutan yang diinginkan setelah operasi pengubahan ukuran selesai. Namun, memilih opsi ini berarti Anda tidak akan dapat menggunakan penskalaan konkurensi untuk tabel ini.

Operasi pengubahan ukuran datang dalam dua jenis:

- Ubah ukuran elastis — Anda dapat menambahkan node ke atau menghapus node dari cluster Anda. Anda juga dapat mengubah jenis node, seperti dari DC2 node ke RA3 node. Pengubahan ukuran elastis biasanya selesai dengan cepat, rata-rata memakan waktu sepuluh menit. Untuk alasan ini, kami merekomendasikannya sebagai opsi pertama. Ketika Anda melakukan pengubahan ukuran elastis, itu mendistribusikan ulang irisan data, yang merupakan partisi yang dialokasikan memori dan ruang disk di setiap node. Pengubahan ukuran elastis sesuai saat Anda:
 - Tambahkan atau kurangi node di cluster yang ada, tetapi Anda tidak mengubah jenis node — Ini biasa disebut pengubahan ukuran di tempat. Saat Anda melakukan jenis pengubahan ukuran ini, beberapa kueri yang berjalan berhasil diselesaikan, tetapi yang lain dapat dihapus sebagai bagian dari operasi.

- Ubah tipe node untuk cluster — Saat Anda mengubah tipe node, snapshot dibuat dan data didistribusikan kembali dari cluster sumber ke cluster yang terdiri dari tipe node baru. Setelah selesai, kueri yang sedang berjalan dijatuhkan. Seperti perubahan ukuran di tempat, itu selesai dengan cepat.
- Perubahan ukuran klasik - Anda dapat mengubah jenis node, jumlah node, atau keduanya, dengan cara yang mirip dengan perubahan ukuran elastis. Perubahan ukuran klasik membutuhkan lebih banyak waktu untuk diselesaikan, tetapi ini dapat berguna dalam kasus di mana perubahan jumlah node atau tipe node yang akan dimigrasi tidak termasuk dalam batas untuk mengubah ukuran elastis. Ini dapat diterapkan, misalnya, ketika perubahan jumlah node sangat besar.

Topik

- [Ubah ukuran elastis](#)
- [Ubah ukuran klasik](#)

Ubah ukuran elastis

Operasi perubahan ukuran elastis, ketika Anda menambah atau menghapus node dari jenis yang sama, memiliki tahapan berikut:

1. Perubahan ukuran elastis membutuhkan snapshot cluster. Tabel tanpa cadangan hanya didukung untuk DC2 node. Untuk semua jenis cluster lainnya, tabel tanpa cadangan disertakan dalam snapshot. Untuk informasi selengkapnya, lihat [Tidak termasuk tabel dari snapshot](#). Jika klaster Anda tidak memiliki snapshot terbaru, karena Anda menonaktifkan snapshot otomatis, operasi pencadangan dapat memakan waktu lebih lama. (Untuk meminimalkan waktu sebelum operasi perubahan ukuran dimulai, kami sarankan Anda mengaktifkan snapshot otomatis atau membuat snapshot manual sebelum memulai perubahan ukuran.) Saat Anda memulai perubahan ukuran elastis dan operasi snapshot sedang berlangsung, perubahan ukuran dapat gagal jika operasi snapshot tidak selesai dalam beberapa menit. Untuk informasi selengkapnya, lihat [Cuplikan dan cadangan Amazon Redshift](#).
2. Operasi memigrasikan metadata cluster. Cluster tidak tersedia selama beberapa menit. Mayoritas kueri dijeda sementara dan koneksi tetap terbuka. Ada kemungkinan, bagaimanapun, untuk beberapa kueri yang akan dihapus. Tahap ini singkat.
3. Koneksi sesi dipulihkan dan kueri dilanjutkan.

4. Pengubahan ukuran elastis mendistribusikan kembali data ke irisan simpul, di latar belakang. Cluster tersedia untuk operasi baca dan tulis, tetapi beberapa kueri dapat memakan waktu lebih lama untuk dijalankan.
5. Setelah operasi selesai, Amazon Redshift mengirimkan pemberitahuan acara.

Saat Anda menggunakan pengubahan ukuran elastis untuk mengubah tipe node, ia bekerja sama dengan saat Anda menambahkan atau mengurangi node dari tipe yang sama. Pertama, snapshot dibuat. Kluster target baru disediakan dengan data terbaru dari snapshot, dan data ditransfer ke cluster baru di latar belakang. Selama periode ini, data hanya dibaca. Saat pengubahan ukuran hampir selesai, Amazon Redshift memperbarui titik akhir untuk menunjuk ke cluster baru dan semua koneksi ke cluster sumber dihapus.

Tidak mungkin pengubahan ukuran elastis akan gagal. Namun, dalam kasus kegagalan, rollback terjadi secara otomatis di sebagian besar kasus tanpa memerlukan intervensi manual.

Jika Anda memiliki node cadangan, misalnya node DC2 cadangan, Anda dapat memutakhirkan ke node RA3 cadangan saat Anda melakukan pengubahan ukuran. Anda dapat melakukan ini ketika Anda melakukan pengubahan ukuran elastis atau menggunakan konsol untuk memulihkan dari snapshot. Konsol memandu Anda melalui proses ini. Untuk informasi selengkapnya tentang memutakhirkan ke RA3 node, lihat [Memutakhirkan ke tipe RA3 node](#).

Pengubahan ukuran elastis tidak mengurutkan tabel atau merebut kembali ruang disk, jadi ini bukan pengganti operasi vakum. Untuk informasi lebih lanjut, lihat tabel [Menyedot debu](#).

Ubah ukuran elastis memiliki kendala berikut:

- Kluster pengubahan ukuran dan berbagi data yang elastis - Saat menambahkan atau mengurangi node pada klaster yang merupakan produsen untuk berbagi data, Anda tidak dapat menghubungkannya dari konsumen saat Amazon Redshift memigrasikan metadata klaster. Demikian pula, jika Anda melakukan pengubahan ukuran elastis dan memilih tipe node baru, berbagi data tidak tersedia saat koneksi dijatuhkan dan ditransfer ke cluster target baru. Dalam kedua jenis pengubahan ukuran elastis, produsen tidak tersedia selama beberapa menit.
- Transfer data dari snapshot bersama - Untuk menjalankan pengubahan ukuran elastis pada klaster yang mentransfer data dari snapshot bersama, setidaknya satu cadangan harus tersedia untuk klaster. Anda dapat melihat cadangan di daftar snapshot konsol Amazon Redshift, perintah `describe-cluster-snapshots` CLI, atau operasi API. `DescribeClusterSnapshots`

- Pembatasan platform - Pengubahan ukuran elastis hanya tersedia untuk cluster yang menggunakan platform EC2-VPC. Untuk informasi selengkapnya, lihat [Gunakan EC2 untuk membuat cluster Anda](#).
- Pertimbangan penyimpanan - Pastikan konfigurasi node baru Anda memiliki penyimpanan yang cukup untuk data yang ada. Anda mungkin harus menambahkan node tambahan atau mengubah konfigurasi.
- Ukuran cluster sumber vs target - Jumlah node dan tipe node yang dapat diubah ukurannya dengan pengubahan ukuran elastis ditentukan oleh jumlah node di cluster sumber dan tipe node yang dipilih untuk cluster yang diubah ukurannya. Untuk menentukan kemungkinan konfigurasi yang tersedia, Anda dapat menggunakan konsol. Atau Anda dapat menggunakan `describe-node-configuration-options` AWS CLI perintah dengan `action-type resize-cluster` opsi. Untuk informasi selengkapnya tentang mengubah ukuran menggunakan konsol Amazon Redshift, lihat. [Mengubah ukuran cluster](#)

Contoh perintah CLI berikut menjelaskan opsi konfigurasi yang tersedia. Dalam contoh ini, cluster bernama `mycluster` adalah cluster `dc2.large` 8-node.

```
aws redshift describe-node-configuration-options --cluster-identifier mycluster --  
region eu-west-1 --action-type resize-cluster
```

Perintah ini mengembalikan daftar opsi dengan jenis node yang direkomendasikan, jumlah node, dan pemanfaatan disk untuk setiap opsi. Konfigurasi yang dikembalikan dapat bervariasi berdasarkan cluster input tertentu. Anda dapat memilih salah satu konfigurasi yang dikembalikan saat Anda menentukan opsi perintah `resize-cluster` CLI.

- Langit-langit pada node tambahan - Ubah ukuran elastis memiliki batas pada node yang dapat Anda tambahkan ke cluster. Misalnya, cluster `dc2` mendukung pengubahan ukuran elastis hingga menggandakan jumlah node. Untuk mengilustrasikan, Anda dapat menambahkan node ke cluster `dc2.xlarge` 4-node untuk menjadikannya cluster lima simpul, atau menambahkan lebih banyak node hingga Anda mencapai delapan.

Note

Batas pertumbuhan dan pengurangan didasarkan pada jenis node asli dan jumlah node di cluster asli atau pengubahan ukuran klasik terakhirnya. Jika pengubahan ukuran elastis akan melebihi batas pertumbuhan atau pengurangan, gunakan pengubahan ukuran klasik.

Dengan beberapa tipe node ra3, Anda dapat meningkatkan jumlah node hingga empat kali jumlah yang ada. Secara khusus, misalkan cluster Anda terdiri dari node ra3.4xlarge atau ra3.16xlarge. Anda kemudian dapat menggunakan pengubahan ukuran elastis untuk meningkatkan jumlah node dalam cluster 8-node menjadi 32. Atau Anda dapat memilih nilai di bawah batas. (Perlu diingat bahwa kemampuan untuk menumbuhkan cluster sebesar 4x tergantung pada ukuran cluster sumber.) Jika cluster Anda memiliki node ra3.xlplus, batasnya dua kali lipat.

Semua tipe node ra3 mendukung penurunan jumlah node menjadi seperempat dari jumlah yang ada. Misalnya, Anda dapat mengurangi ukuran cluster dengan node ra3.4xlarge dari 12 node menjadi 3, atau ke angka di atas minimum.

Tabel berikut mencantumkan batas pertumbuhan dan pengurangan untuk setiap jenis node yang mendukung pengubahan ukuran elastis.

Jenis simpul asli	Batas pertumbuhan	Batas pengurangan
ra3.16xlarge	4x (dari 4 hingga 16 node, misalnya)	Untuk seperempat dari jumlah (dari 16 hingga 4 node, misalnya)
ra3.4xlarge	4x	Untuk seperempat dari jumlah
ra3.xlplus	2x (dari 4 hingga 8 node, misalnya)	Untuk seperempat dari jumlah
ra3. besar	2x	Setengah dari jumlah
dc2.8xlarge	2x	Untuk setengah dari jumlah (dari 16 hingga 8 node, misalnya)
dc2.large	2x	Setengah dari jumlah

Note

Memilih jenis node lama saat Anda mengubah ukuran RA3 cluster — Jika Anda mencoba mengubah ukuran dari cluster dengan RA3 node ke tipe node lain, seperti DC2, pesan

peringatan validasi muncul di konsol, dan operasi pengubahan ukuran tidak akan selesai. Ini terjadi karena mengubah ukuran ke tipe simpul lama tidak didukung. Ini untuk mencegah pelanggan mengubah ukuran ke jenis node yang tidak digunakan lagi atau segera tidak digunakan lagi. Ini berlaku untuk pengubahan ukuran elastis dan pengubahan ukuran klasik.

Ubah ukuran klasik

Pengubahan ukuran klasik menangani kasus penggunaan di mana perubahan ukuran cluster atau tipe node tidak didukung oleh pengubahan ukuran elastis. Saat Anda melakukan pengubahan ukuran klasik, Amazon Redshift membuat kluster target dan memigrasikan data dan metadata Anda ke kluster sumber.

Ubah ukuran klasik untuk RA3 dapat memberikan ketersediaan yang lebih baik

Pengubahan ukuran klasik telah ditingkatkan ketika tipe node target adalah RA3. Hal ini dilakukan dengan menggunakan backup dan restore operasi antara sumber dan target cluster. Ketika pengubahan ukuran dimulai, cluster sumber dimulai ulang dan tidak tersedia selama beberapa menit. Setelah itu, cluster tersedia untuk operasi baca dan tulis sementara pengubahan ukuran berlanjut di latar belakang.

Memeriksa cluster Anda

Untuk memastikan Anda memiliki kinerja dan hasil terbaik saat Anda melakukan pengubahan ukuran klasik ke RA3 klaster, lengkapi daftar periksa ini. Ketika Anda tidak mengikuti daftar periksa, Anda mungkin tidak mendapatkan beberapa manfaat dari mengubah ukuran klasik dengan RA3 node, seperti kemampuan untuk melakukan operasi baca dan tulis.

1. Ukuran data harus di bawah 2 petabyte. (Satu petabyte sama dengan 1.000 terabyte.) Untuk memvalidasi ukuran data Anda, buat snapshot dan periksa ukurannya. Anda juga dapat menjalankan kueri berikut untuk memeriksa ukurannya:

```
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_infoTabel hanya terlihat oleh pengguna super.

2. Sebelum Anda memulai pengubahan ukuran klasik, pastikan Anda memiliki snapshot manual yang tidak lebih dari 10 jam. Jika tidak, ambil snapshot.
3. Snapshot yang digunakan untuk melakukan pengubahan ukuran klasik tidak dapat digunakan untuk pemulihan tabel atau tujuan lain.
4. Cluster harus dalam VPC.

Operasi penyortiran dan distribusi yang dihasilkan dari pengubahan ukuran klasik ke RA3

Selama pengubahan ukuran klasik ke RA3, tabel dengan distribusi KEY yang dimigrasikan sebagai distribusi EVEN dikonversi kembali ke gaya distribusi aslinya. Durasi ini tergantung pada ukuran data dan seberapa sibuk cluster Anda. Beban kerja kueri diberikan prioritas yang lebih tinggi untuk menjalankan migrasi data. Untuk informasi selengkapnya, lihat [Gaya distribusi](#). Baik membaca dan menulis ke database bekerja selama proses migrasi ini, tetapi dapat memakan waktu lebih lama untuk menyelesaikan kueri. Namun, penskalaan konkurensi dapat meningkatkan kinerja selama waktu ini dengan menambahkan sumber daya untuk beban kerja kueri. [Anda dapat melihat kemajuan migrasi data dengan melihat hasil dari tampilan SYS_RESTORE_STATE dan SYS_RESTORE_LOG](#). Informasi lebih lanjut tentang pemantauan berikut.

Setelah cluster diubah ukurannya sepenuhnya, perilaku pengurutan berikut terjadi:

- Jika pengubahan ukuran menghasilkan cluster yang memiliki lebih banyak irisan, tabel distribusi KEY menjadi sebagian tidak disortir, tetapi tabel EVEN tetap diurutkan. Selain itu, informasi tentang berapa banyak data yang diurutkan mungkin tidak mutakhir, langsung mengikuti pengubahan ukuran. Setelah pemulihan kunci, vakum otomatis menyortir tabel dari waktu ke waktu.
- Jika pengubahan ukuran menghasilkan cluster yang memiliki irisan lebih sedikit, distribusi KEY dan tabel distribusi EVEN menjadi sebagian tidak disortir. Vakum otomatis menyortir meja dari waktu ke waktu.

Untuk informasi lebih lanjut tentang vakum meja otomatis, lihat [Menyedot debu](#) tabel. Untuk informasi selengkapnya tentang irisan di node komputasi, lihat [Arsitektur sistem gudang data](#).

Langkah mengubah ukuran klasik saat cluster target RA3

Pengubahan ukuran klasik terdiri dari langkah-langkah berikut, ketika jenis cluster target RA3 dan Anda telah memenuhi prasyarat yang dirinci di bagian sebelumnya.

1. Migrasi dimulai dari cluster sumber ke cluster target. Saat kluster target baru disediakan, Amazon Redshift mengirimkan pemberitahuan peristiwa bahwa pengubahan ukuran telah dimulai. Ini

- memulai ulang cluster Anda yang ada, yang menutup semua koneksi. Jika kluster Anda yang ada adalah cluster produsen berbagi data, koneksi dengan cluster konsumen juga ditutup. Restart membutuhkan waktu beberapa menit.
2. Setelah restart, database tersedia untuk dibaca dan ditulis. Selain itu, berbagi data dilanjutkan, yang membutuhkan beberapa menit tambahan.
 3. Data dimigrasikan ke cluster target. Ketika tipe node target adalah RA3, membaca dan menulis tersedia selama migrasi data.
 4. Saat proses pengubahan ukuran hampir selesai, Amazon Redshift memperbarui titik akhir ke cluster target, dan semua koneksi ke cluster sumber akan dihapus. Cluster target menjadi produsen untuk berbagi data.
 5. Pengubahan ukuran selesai. Amazon Redshift mengirimkan pemberitahuan acara.

Anda dapat melihat kemajuan pengubahan ukuran di konsol Amazon Redshift. Waktu yang dibutuhkan untuk mengubah ukuran cluster tergantung pada jumlah data.

Note

Memilih jenis node lama saat Anda mengubah ukuran RA3 cluster — Jika Anda mencoba mengubah ukuran dari cluster dengan RA3 node ke tipe node lain, seperti DC2, pesan peringatan validasi muncul di konsol, dan operasi pengubahan ukuran tidak akan selesai. Ini terjadi karena mengubah ukuran ke tipe simpul lama tidak didukung. Ini untuk mencegah pelanggan mengubah ukuran ke jenis node yang tidak digunakan lagi atau segera tidak digunakan lagi. Ini berlaku untuk pengubahan ukuran elastis dan pengubahan ukuran klasik.

Memantau pengubahan ukuran klasik saat kluster target RA3

[Untuk memantau pengubahan ukuran klasik dari kluster yang disediakan yang sedang berlangsung, termasuk distribusi KEY, gunakan SYS_RESTORE_STATE.](#) Ini menunjukkan persentase selesai untuk tabel yang dikonversi. Anda harus menjadi pengguna super untuk mengakses data.

Jatuhkan tabel yang tidak Anda butuhkan saat Anda melakukan pengubahan ukuran klasik. Ketika Anda melakukan ini, tabel yang ada dapat didistribusikan lebih cepat.

Langkah mengubah ukuran klasik saat kluster target tidak RA3

Pengubahan ukuran klasik terdiri dari berikut ini, ketika tipe node target adalah apa pun selain RA3, seperti DC2, misalnya.

1. Migrasi dimulai dari cluster sumber ke cluster target. Saat kluster target baru disediakan, Amazon Redshift mengirimkan pemberitahuan peristiwa bahwa pengubahan ukuran telah dimulai. Ini memulai ulang cluster Anda yang ada, yang menutup semua koneksi. Jika kluster Anda yang ada adalah cluster produsen berbagi data, koneksi dengan cluster konsumen juga ditutup. Restart membutuhkan waktu beberapa menit.

Perhatikan bahwa setiap relasi database, seperti tabel atau tampilan terwujud, dibuat dengan tidak BACKUP NO dipertahankan selama pengubahan ukuran klasik. Untuk informasi selengkapnya, lihat [MEMBUAT TAMPILAN TERWUJUD](#).

2. Setelah restart, database tersedia hanya sebagai baca. Berbagi data dilanjutkan, yang membutuhkan beberapa menit tambahan.
3. Data dimigrasikan ke cluster target. Database tetap dibaca saja.
4. Saat proses pengubahan ukuran hampir selesai, Amazon Redshift memperbarui titik akhir ke cluster target, dan semua koneksi ke cluster sumber akan dihapus. Cluster target menjadi produsen untuk berbagi data.
5. Pengubahan ukuran selesai. Amazon Redshift mengirimkan pemberitahuan acara.

Anda dapat melihat kemajuan pengubahan ukuran di konsol Amazon Redshift. Waktu yang dibutuhkan untuk mengubah ukuran cluster tergantung pada jumlah data.

Note

Diperlukan waktu sehari-hari atau mungkin berminggu-minggu untuk mengubah ukuran cluster dengan sejumlah besar data ketika cluster target tidak RA3, atau tidak memenuhi prasyarat untuk cluster RA3 target yang dirinci di bagian sebelumnya.

Perhatikan juga bahwa kapasitas penyimpanan yang digunakan untuk cluster dapat naik setelah mengubah ukuran klasik. Ini adalah perilaku sistem normal ketika cluster memiliki irisan data tambahan yang dihasilkan dari pengubahan ukuran klasik. Penggunaan kapasitas tambahan ini dapat terjadi bahkan ketika jumlah node di cluster tetap sama.

Ubah ukuran elastis vs ukuran klasik

Tabel berikut membandingkan perilaku antara dua jenis pengubahan ukuran.

Perilaku	Ubah ukuran elastis	Ubah ukuran klasik	Koment				
Retensi data sistem	Pengubahan ukuran elastis mempertahankan data log sistem.	Pengubahan ukuran klasik tidak menyimpan tabel dan data sistem.	Jika Anda mengaktifkan pencatatan audit di kluster sumber, Anda dapat terus mengakses log di Amazon S3 atau di CloudWatch, setelah mengubah ukuran. Anda dapat menyimpan atau menghapus log ini seperti yang ditentukan				

Perilaku	Ubah ukuran elastis	Ubah ukuran klasik	Koment				
			n oleh kebijakan data Anda.				
Mengubah jenis node	Ubah ukuran elastis, ketika tipe node tidak berubah: Ubah ukuran di tempat, dan sebagian besar kueri ditahan. Ubah ukuran elastis, dengan tipe node baru dipilih: Sebuah cluster baru dibuat. Kueri dijatuhkan saat proses pengubahan ukuran selesai.	Classic Resize: Sebuah cluster baru dibuat. Kueri dijatuhkan selama proses pengubahan ukuran.					

Perilaku	Ubah ukuran elastis	Ubah ukuran klasik	Koment				
Sesi dan retensi kueri	Pengubahan ukuran elastis mempertahankan sesi dan kueri saat tipe node sama di cluster sumber dan target. Jika Anda memilih jenis node baru, kueri akan dihapus.	Pengubahan ukuran klasik tidak mempertahankan sesi dan kueri. Pertanyaan dijatuhkan.	Saat kueri dihapus, Anda dapat mengharapkan beberapa penurunan kinerja. Yang terbaik adalah melakukan operasi pengubahan ukuran selama periode penggunaan ringan.				

Perilaku	Ubah ukuran elastis	Ubah ukuran klasik	Koment				
Membatalkan operasi perubahan ukuran	Anda tidak dapat membatalkan perubahan ukuran elastis.	Anda dapat membatalkan operasi perubahan ukuran klasik sebelum selesai dengan memilih Batalkan perubahan ukuran dari detail cluster di konsol Amazon Redshift.	Jumlah waktu yang diperlukan untuk membatalkan perubahan ukuran tergantung pada tahap operasi perubahan ukuran saat Anda membatalkan. Saat Anda melakukan ini, klaster tidak tersedia sampai operasi pembatala				

Perilaku	Ubah ukuran elastis	Ubah ukuran klasik	Koment				
			n selesai. Jika operasi pengubaha n ukuran berada di tahap akhir, Anda tidak dapat membatal an. Untuk mengubah ukuran klasik ke RA3 klaster, Anda tidak dapat membatal an.				

Menjadwalkan perubahan ukuran

Anda dapat menjadwalkan operasi perubahan ukuran untuk klaster Anda untuk ditingkatkan guna mengantisipasi penggunaan yang tinggi atau untuk mengurangi penghematan biaya. Penjadwalan

berfungsi untuk mengubah ukuran elastis dan mengubah ukuran klasik. Anda dapat mengatur jadwal di konsol Amazon Redshift. Untuk informasi selengkapnya, lihat [Mengubah ukuran cluster](#), di bagian Mengelola kluster menggunakan konsol. Anda juga dapat menggunakan AWS CLI atau operasi Amazon Redshift API untuk menjadwalkan pengubahan ukuran. Untuk informasi selengkapnya, lihat [create-scheduled-action](#) di Referensi AWS CLI Perintah atau [CreateScheduledAction](#) di Referensi API Amazon Redshift.

Snapshot, pulihkan, dan ubah ukuran

[Pengubahan ukuran elastis](#) adalah metode tercepat untuk mengubah ukuran cluster Amazon Redshift. Jika pengubahan ukuran elastis bukan pilihan untuk Anda dan Anda memerlukan akses tulis hampir konstan ke kluster Anda, gunakan operasi snapshot dan pulihkan dengan pengubahan ukuran klasik seperti yang dijelaskan di bagian berikut. Pendekatan ini mensyaratkan bahwa setiap data yang ditulis ke cluster sumber setelah snapshot diambil harus disalin secara manual ke cluster target setelah sakelar. Bergantung pada berapa lama salinannya, Anda mungkin perlu mengulangnya beberapa kali sampai Anda memiliki data yang sama di kedua cluster. Kemudian Anda dapat beralih ke cluster target. Proses ini mungkin berdampak negatif pada kueri yang ada sampai kumpulan data lengkap tersedia di kluster target. Namun, ini meminimalkan jumlah waktu yang tidak dapat Anda tulis ke database.

Pendekatan snapshot, restore, dan classic resize menggunakan proses berikut:

1. Ambil snapshot dari cluster Anda yang ada. Cluster yang ada adalah cluster sumber.
2. Perhatikan waktu snapshot diambil. Melakukan hal ini berarti Anda nantinya dapat mengidentifikasi titik ketika Anda perlu menjalankan kembali proses ekstrak, bertransaksi, memuat (ETL) untuk memuat data pasca-snapshot apa pun ke dalam basis data target.
3. Kembalikan snapshot ke cluster baru. Cluster baru ini adalah cluster target. Verifikasi bahwa data sampel ada di cluster target.
4. Ubah ukuran cluster target. Pilih jenis node baru, jumlah node, dan pengaturan lain untuk cluster target.
5. Tinjau beban dari proses ETL Anda yang terjadi setelah Anda mengambil snapshot dari cluster sumber. Pastikan untuk memuat ulang data yang sama dalam urutan yang sama ke dalam cluster target. Jika Anda memiliki pemuatan data yang sedang berlangsung, ulangi proses ini beberapa kali hingga data sama di cluster sumber dan target.
6. Hentikan semua kueri yang berjalan di cluster sumber. [Untuk melakukan ini, Anda dapat me-reboot cluster, atau Anda dapat masuk sebagai superuser dan menggunakan perintah](#)

[PG_CANCEL_BACKEND](#) dan [PG_TERMINATE_BACKEND](#). Mem-boot ulang cluster adalah cara termudah untuk memastikan bahwa cluster tidak tersedia.

7. Ganti nama cluster sumber. Misalnya, ganti nama dari `examplecluster` ke `examplecluster-source`.
8. Ganti nama cluster target untuk menggunakan nama cluster sumber sebelum mengganti nama. Misalnya, ganti nama cluster target dari sebelumnya menjadi. `examplecluster` Mulai saat ini, aplikasi apa pun yang menggunakan titik akhir yang berisi `examplecluster` terhubung ke cluster target.
9. Hapus kluster sumber setelah Anda beralih ke kluster target, dan verifikasi bahwa semua proses berfungsi seperti yang diharapkan.

Atau, Anda dapat mengganti nama cluster sumber dan target sebelum memuat ulang data ke kluster target. Pendekatan ini berfungsi jika Anda tidak mengharuskan sistem dan laporan dependen apa pun segera diperbarui dengan yang ada di cluster target. Dalam hal ini, langkah 6 bergerak ke akhir proses yang dijelaskan sebelumnya.

Proses rename hanya diperlukan jika Anda ingin aplikasi terus menggunakan endpoint yang sama untuk terhubung ke cluster. Jika Anda tidak memerlukan ini, Anda dapat memperbarui aplikasi apa pun yang terhubung ke cluster untuk menggunakan titik akhir kluster target tanpa mengganti nama cluster.

Ada beberapa manfaat untuk menggunakan kembali nama cluster. Pertama, Anda tidak perlu memperbarui string koneksi aplikasi karena titik akhir tidak berubah, meskipun kluster yang mendasarinya berubah. Kedua, item terkait seperti CloudWatch alarm Amazon dan notifikasi Amazon Simple Notification Service (Amazon SNS) terkait dengan nama cluster. Ikatan ini berarti Anda dapat terus menggunakan alarm dan notifikasi yang sama dengan yang Anda atur untuk cluster. Penggunaan berkelanjutan ini terutama menjadi perhatian di lingkungan produksi di mana Anda menginginkan fleksibilitas untuk mengubah ukuran cluster tanpa mengonfigurasi ulang item terkait, seperti alarm dan notifikasi.

Mengganti nama cluster

Anda dapat mengganti nama cluster jika Anda ingin cluster menggunakan nama yang berbeda. Karena titik akhir kluster Anda menyertakan nama cluster (juga disebut sebagai pengidentifikasi cluster), titik akhir berubah untuk menggunakan nama baru setelah penggantian nama selesai. Misalnya, jika Anda memiliki kluster bernama `examplecluster` dan mengganti namanya

menjadikan `newcluster`, titik akhir berubah untuk menggunakan pengenal. `newcluster` Setiap aplikasi yang terhubung ke cluster harus diperbarui dengan endpoint baru.

Anda dapat mengganti nama cluster jika Anda ingin mengubah cluster yang terhubung dengan aplikasi Anda tanpa harus mengubah titik akhir dalam aplikasi tersebut. Dalam hal ini, Anda harus terlebih dahulu mengganti nama cluster asli dan kemudian mengubah cluster kedua untuk menggunakan kembali nama cluster asli sebelum mengganti nama. Melakukan hal ini diperlukan karena pengidentifikasi klaster harus unik di dalam akun dan wilayah Anda, sehingga cluster asli dan cluster kedua tidak dapat memiliki nama yang sama. Anda dapat melakukan ini jika Anda memulihkan cluster dari snapshot dan tidak ingin mengubah properti koneksi dari aplikasi dependen apa pun.

 Note

Jika Anda menghapus cluster asli, Anda bertanggung jawab untuk menghapus snapshot cluster yang tidak diinginkan.

Saat Anda mengganti nama klaster, status klaster berubah `renaming` hingga proses selesai. Nama DNS lama yang digunakan oleh cluster segera dihapus, meskipun bisa tetap di-cache selama beberapa menit. Nama DNS baru untuk cluster yang diganti namanya menjadi efektif dalam waktu sekitar 10 menit. Cluster yang diganti namanya tidak tersedia sampai nama baru menjadi efektif. Cluster akan di-boot ulang dan koneksi yang ada ke cluster akan dihapus. Setelah ini selesai, titik akhir akan berubah untuk menggunakan nama baru. Untuk alasan ini, Anda harus menghentikan kueri agar tidak berjalan sebelum memulai penggantian nama dan memulai ulang setelah penggantian nama selesai.

Snapshot cluster dipertahankan, dan semua snapshot yang terkait dengan cluster tetap terkait dengan cluster tersebut setelah diganti namanya. Misalnya, Anda memiliki cluster yang melayani database produksi Anda dan cluster memiliki beberapa snapshot. Jika Anda mengganti nama cluster dan kemudian menggantinya di lingkungan produksi dengan snapshot, cluster yang Anda ganti nama masih memiliki snapshot yang ada yang terkait dengannya.

CloudWatch Alarm Amazon dan notifikasi acara Amazon Simple Notification Service (Amazon SNS) dikaitkan dengan nama cluster. Jika Anda mengganti nama cluster, Anda harus memperbarui ini sesuai dengan itu. Anda dapat memperbarui CloudWatch alarm di CloudWatch konsol, dan Anda dapat memperbarui notifikasi acara Amazon SNS di konsol Amazon Redshift di panel Acara. Data pemuatan dan kueri untuk cluster terus menampilkan data dari sebelum penggantian nama dan

setelah penggantian nama. Namun, data kinerja diatur ulang setelah proses penggantian nama selesai.

Untuk informasi selengkapnya, lihat [Mengubah klaster](#).

Memutakhirkan versi rilis cluster

Anda dapat memutakhirkan versi pemeliharaan rilis klaster yang memiliki nilai Status Rilis rilis Baru yang tersedia. Saat Anda memutakhirkan versi pemeliharaan, Anda dapat memilih untuk segera memutakhirkan atau meningkatkan di jendela pemeliharaan berikutnya.

Important

Jika Anda segera memutakhirkan, klaster Anda offline hingga pemutakhiran selesai.

Untuk memutakhirkan cluster ke versi rilis baru

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster.
3. Pilih cluster untuk meng-upgrade.
4. Untuk Tindakan, pilih Upgrade versi cluster. Halaman versi klaster Upgrade muncul.
5. Ikuti instruksi di halaman.
6. Pilih Upgrade versi cluster.

Menjeda dan melanjutkan cluster

Jika Anda memiliki cluster yang hanya perlu tersedia pada waktu tertentu, Anda dapat menjeda cluster dan kemudian melanjutkan. Sementara cluster dijeda, penagihan sesuai permintaan ditangguhkan. Hanya penyimpanan cluster yang dikenakan biaya. Untuk informasi selengkapnya tentang harga, lihat halaman [harga Amazon Redshift](#).

Saat Anda menjeda klaster, Amazon Redshift membuat snapshot, mulai menghentikan kueri, dan menempatkan cluster dalam status jeda. Jika Anda menghapus klaster yang dijeda tanpa meminta snapshot akhir, maka Anda tidak dapat memulihkan cluster. Anda tidak dapat membatalkan atau memutar kembali jeda atau melanjutkan operasi setelah dimulai.

Anda dapat menjeda dan melanjutkan cluster di konsol Amazon Redshift, dengan, atau dengan AWS CLI operasi Amazon Redshift API.

Anda dapat menjadwalkan tindakan untuk menjeda dan melanjutkan cluster. Saat Anda menggunakan konsol Amazon Redshift baru untuk membuat jadwal berulang untuk menjeda dan melanjutkan, maka dua tindakan terjadwal dibuat untuk rentang tanggal yang Anda pilih. Nama tindakan terjadwal diberi akhiran dengan `-pause` dan `-resume`. Panjang total nama harus sesuai dengan ukuran maksimum nama tindakan terjadwal.

Anda tidak dapat menjeda jenis cluster berikut:

- Cluster EC2-klasik.
- Cluster yang tidak aktif, misalnya, cluster yang saat ini sedang dimodifikasi.
- Cluster modul keamanan perangkat keras (HSM).
- Cluster yang memiliki snapshot otomatis dimatikan.

Saat memutuskan untuk menjeda klaster, pertimbangkan hal berikut:

- Koneksi atau kueri ke klaster tidak tersedia.
- Anda tidak dapat melihat informasi pemantauan kueri klaster yang dijeda di konsol Amazon Redshift.
- Anda tidak dapat memodifikasi cluster yang dijeda. Tindakan terjadwal apa pun di cluster tidak dilakukan. Ini termasuk membuat snapshot, mengubah ukuran cluster, dan operasi pemeliharaan cluster.
- Metrik perangkat keras tidak dibuat. Perbarui CloudWatch alarm Anda jika alarm disetel pada metrik yang hilang.
- Anda tidak dapat menyalin snapshot otomatis terbaru dari cluster yang dijeda ke snapshot manual.
- Sementara cluster berhenti, itu tidak dapat dilanjutkan sampai operasi jeda selesai.
- Saat Anda menjeda klaster, penagihan ditangguhkan. Namun, operasi jeda biasanya selesai dalam waktu 15 menit, tergantung pada ukuran cluster.
- Log audit diarsipkan dan tidak dipulihkan pada resume.
- Setelah klaster dijeda, jejak dan log mungkin tidak tersedia untuk memecahkan masalah yang terjadi sebelum jeda.
- Jika Anda mengelola kredensi admin Anda menggunakan AWS Secrets Manager dan menjeda klaster Anda, rahasia klaster Anda tidak akan dihapus dan Anda akan terus ditagih untuk rahasia

tersebut. Untuk informasi selengkapnya tentang mengelola kata sandi admin Redshift Anda AWS Secrets Manager, lihat. [Mengelola kata sandi admin Amazon Redshift menggunakan AWS Secrets Manager](#)

- Tabel tanpa cadangan pada cluster dipulihkan pada resume untuk jenis RA3 contoh. Mereka tidak dipulihkan pada resume DC2 misalnya jenis. Untuk informasi selengkapnya tentang tabel tanpa cadangan, lihat [Tidak termasuk tabel dari snapshot](#).

Saat Anda melanjutkan cluster, pertimbangkan hal berikut:

- Versi cluster dari cluster yang dilanjutkan diperbarui ke versi pemeliharaan berdasarkan jendela pemeliharaan cluster.
- Jika Anda menghapus subnet yang terkait dengan cluster yang dijeda, Anda mungkin memiliki jaringan yang tidak kompatibel. Dalam hal ini, pulihkan cluster Anda dari snapshot terbaru.
- Jika Anda menghapus alamat IP Elastis saat cluster dijeda, maka alamat IP Elastis baru diminta.
- Jika Amazon Redshift tidak dapat melanjutkan cluster dengan elastic network interface sebelumnya, Amazon Redshift mencoba mengalokasikan yang baru.
- Saat Anda melanjutkan cluster, alamat IP node Anda mungkin berubah. Anda mungkin perlu memperbarui pengaturan VPC Anda untuk mendukung alamat IP baru ini untuk fitur seperti COPY from Secure Shell (SSH) atau COPY dari Amazon EMR.
- Jika Anda mencoba melanjutkan cluster yang tidak dijeda, operasi resume mengembalikan kesalahan. Jika operasi resume merupakan bagian dari tindakan terjadwal, ubah atau hapus tindakan terjadwal untuk mencegah kesalahan di masa depan.
- Bergantung pada ukuran cluster, dibutuhkan beberapa menit untuk melanjutkan cluster sebelum kueri dapat diproses. Selain itu, kinerja kueri dapat terpengaruh untuk beberapa periode waktu saat cluster sedang dihidrasi ulang setelah resume selesai.

Melakukan boot ulang klaster

Mem-boot ulang cluster adalah operasi cluster yang memulai ulang cluster dengan konfigurasi yang sama seperti sebelum reboot. Anda dapat me-reboot cluster untuk menerapkan pembaruan pemeliharaan yang tertunda, mengatur ulang perubahan konfigurasi, memulihkan dari masalah tertentu, atau memecahkan masalah klaster. Mem-boot ulang klaster dapat membantu memastikan kinerja, keamanan, dan stabilitas lingkungan Amazon Redshift yang optimal. Prosedur berikut memberikan langkah-langkah terperinci untuk me-reboot cluster Amazon Redshift.

Saat Anda me-reboot cluster, status cluster disetel ke `rebooting` dan acara cluster dibuat saat reboot selesai. Setiap modifikasi cluster yang tertunda diterapkan pada reboot ini.

Untuk me-reboot sebuah cluster

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster.
3. Pilih cluster untuk reboot.
4. Untuk Tindakan, pilih Reboot cluster. Halaman cluster Reboot muncul.
5. Pilih Reboot cluster.

Merelokasi cluster

Dengan menggunakan relokasi di Amazon Redshift, Anda mengizinkan Amazon Redshift memindahkan klaster ke Availability Zone (AZ) lain tanpa kehilangan data atau perubahan pada aplikasi Anda. Dengan relokasi, Anda dapat melanjutkan operasi ketika ada gangguan layanan pada cluster Anda dengan dampak minimal.

Saat relokasi cluster diaktifkan, Amazon Redshift mungkin memilih untuk memindahkan cluster dalam beberapa situasi. Secara khusus, ini terjadi ketika masalah di Availability Zone saat ini mencegah operasi cluster yang optimal atau untuk meningkatkan ketersediaan layanan. Anda juga dapat menjalankan fungsi relokasi jika kendala sumber daya di Availability Zone tertentu mengganggu operasi klaster. Contohnya adalah kemampuan untuk melanjutkan atau mengubah ukuran cluster. Amazon Redshift menawarkan fitur relokasi tanpa biaya tambahan.

Saat klaster Amazon Redshift dipindahkan ke Availability Zone baru, cluster baru memiliki titik akhir yang sama dengan cluster asli. Aplikasi Anda dapat terhubung kembali ke titik akhir dan melanjutkan operasi tanpa modifikasi atau kehilangan data. Namun, relokasi mungkin tidak selalu dimungkinkan karena potensi kendala sumber daya di Availability Zone tertentu.

Relokasi klaster Amazon Redshift hanya didukung untuk jenis RA3 instans. RA3 tipe instance menggunakan Redshift Managed Storage (RMS) sebagai lapisan penyimpanan yang tahan lama. Salinan terbaru dari data klaster selalu tersedia di Availability Zone lainnya di suatu AWS Wilayah. Dengan kata lain, Anda dapat memindahkan cluster Amazon Redshift ke Availability Zone lain tanpa kehilangan data.

Saat Anda mengaktifkan relokasi untuk klaster, Amazon Redshift memigrasikan klaster Anda untuk berada di belakang proxy. Melakukan hal ini membantu mengimplementasikan akses independen lokasi ke sumber daya komputasi klaster. Migrasi menyebabkan cluster di-boot ulang. Saat klaster dipindahkan ke Availability Zone lain, pemadaman terjadi saat klaster baru dibawa kembali online di Availability Zone yang baru. Namun, Anda tidak perlu membuat perubahan apa pun pada aplikasi Anda karena titik akhir cluster tetap tidak berubah bahkan setelah cluster dipindahkan ke Availability Zone yang baru.

Relokasi klaster diaktifkan secara default pada cluster yang baru dibuat atau dipulihkan RA3 yang grup subnetnya mencakup beberapa Availability Zone. Amazon Redshift menetapkan 5439 sebagai port default saat membuat cluster yang disediakan. Anda dapat mengubah ke port lain dari rentang port 5431-5455 atau 8191-8215. (Jangan mengubah ke port di luar rentang. Ini menghasilkan kesalahan.) Untuk mengubah port default untuk klaster yang disediakan, gunakan konsol Amazon Redshift AWS CLI, atau Amazon Redshift API. Untuk mengubah port default untuk grup kerja tanpa server, gunakan API Tanpa Server Amazon Redshift AWS CLI atau Amazon Redshift.

Jika Anda mengaktifkan relokasi dan saat ini Anda menggunakan alamat IP node pemimpin untuk mengakses klaster Anda atau Perutean VPC yang Ditingkatkan, pastikan untuk mengubah akses itu. Sebagai gantinya, gunakan alamat IP yang terkait dengan titik akhir virtual private cloud (VPC) cluster. Untuk menemukan alamat IP cluster ini, temukan dan gunakan titik akhir VPC di bagian Jaringan dan keamanan pada halaman detail cluster. Untuk mendapatkan detail selengkapnya tentang titik akhir VPC, masuk ke konsol VPC Amazon.

Anda juga dapat menggunakan perintah AWS Command Line Interface (AWS CLI) `describe-vpc-endpoints` untuk mendapatkan elastic network interface yang terkait dengan endpoint. Anda dapat menggunakan `describe-network-interfaces` perintah untuk mendapatkan alamat IP terkait. Untuk informasi selengkapnya tentang perintah Amazon Redshift, lihat [AWS CLI Perintah yang tersedia di Referensi AWS CLI Perintah](#).

Batasan

Saat menggunakan relokasi Amazon Redshift, perhatikan batasan berikut:

- Relokasi klaster mungkin tidak dapat dilakukan di semua skenario karena potensi keterbatasan sumber daya di Availability Zone tertentu. Jika ini terjadi, Amazon Redshift tidak mengubah cluster asli.
- Relokasi tidak didukung pada keluarga produk DC2 contoh.
- Anda tidak dapat melakukan relokasi di seluruh AWS Wilayah.

- Relokasi Amazon Redshift default ke nomor port 5439. Anda juga dapat mengubah ke port lain dalam rentang 5431-5455 atau 8191-8215.

Mengelola relokasi menggunakan konsol

Anda dapat mengelola pengaturan untuk relokasi cluster menggunakan konsol Amazon Redshift.

Mematikan relokasi saat membuat cluster baru

Gunakan prosedur berikut untuk mematikan relokasi saat membuat cluster baru.

Untuk mematikan relokasi untuk klaster baru

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster.
3. Pilih Buat cluster untuk membuat cluster baru. Untuk informasi selengkapnya tentang cara membuat klaster, lihat [Memulai gudang data yang disediakan Amazon Redshift di Panduan Memulai](#) Pergeseran Merah Amazon.
4. Di bagian Backup, untuk relokasi Cluster, pilih Dinonaktifkan. Relokasi diaktifkan secara default.
5. Pilih Buat klaster.

Memodifikasi relokasi untuk cluster yang ada

Gunakan prosedur berikut untuk mengubah pengaturan relokasi untuk cluster yang ada.

Untuk mengubah pengaturan relokasi untuk klaster yang ada

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster. Cluster untuk akun Anda di AWS Wilayah saat ini terdaftar. Subset properti dari setiap cluster ditampilkan dalam kolom dalam daftar.
3. Pilih nama cluster yang ingin Anda ubah dari daftar. Halaman detail cluster muncul.
4. Pilih tab Maintenance, lalu di bagian Backup details pilih Edit.
5. Di bawah Backup, pilih Dinonaktifkan. Relokasi diaktifkan secara default.
6. Pilih Ubah klaster.

Merelokasi cluster

Gunakan prosedur berikut untuk memindahkan cluster secara manual ke Availability Zone lain. Ini sangat berguna ketika Anda ingin menguji pengaturan jaringan Anda di Availability Zone sekunder atau ketika Anda mengalami kendala sumber daya di Availability Zone saat ini.

Untuk memindahkan klaster ke Availability Zone lain

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster. Cluster untuk akun Anda di AWS Wilayah saat ini terdaftar. Subset properti dari setiap cluster ditampilkan dalam kolom dalam daftar.
3. Pilih nama cluster yang ingin Anda pindahkan dari daftar. Halaman detail cluster muncul.
4. Untuk Tindakan, pilih Relokasi. Halaman klaster Relokasi muncul.
5. (Opsional) Pilih Availability Zone. Jika Anda tidak memilih Availability Zone, Amazon Redshift memilihnya untuk Anda.

Amazon Redshift memulai relokasi dan menampilkan cluster sebagai relokasi. Setelah relokasi selesai, status cluster berubah menjadi tersedia.

Mengelola relokasi menggunakan Amazon Redshift CLI

Anda dapat mengelola pengaturan untuk relokasi cluster menggunakan AWS Command Line Interface (CLI).

Dengan AWS CLI, perintah contoh berikut membuat cluster Amazon Redshift **mycluster** bernama yang telah diaktifkan relokasi.

```
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
```

Jika cluster Anda saat ini menggunakan port yang berbeda, Anda harus memodifikasinya untuk digunakan dari rentang port 5431-5455 atau 8191-8215 sebelum memodifikasinya untuk mengaktifkan relokasi. Defaultnya adalah 5439. Contoh perintah berikut memodifikasi port jika cluster Anda tidak menggunakan salah satu dari rentang yang diberikan.

```
aws redshift modify-cluster --cluster-identifier mycluster --port 5439
```

Contoh perintah berikut mencakup availability-zone-relocation parameter pada cluster Amazon Redshift.

```
aws redshift modify-cluster --cluster-identifier mycluster --availability-zone-relocation
```

Contoh perintah berikut mematikan availability-zone-relocation parameter pada cluster Amazon Redshift.

```
aws redshift modify-cluster --cluster-identifier mycluster --no-availability-zone-relocation
```

Contoh perintah berikut memanggil relokasi di cluster Amazon Redshift.

```
aws redshift modify-cluster --cluster-identifier mycluster --availability-zone us-east-1b
```

Menetapkan batas penggunaan pada klaster

Anda dapat menambahkan hingga empat batas penggunaan untuk mengontrol penggunaan untuk masing-masing hal berikut:

- Penskalaan konkurensi
- Pengoptimalan otomatis berjalan menggunakan sumber daya komputasi tambahan
- Penggunaan Redshift Spectrum
- Berbagi data Lintas Wilayah

Menetapkan batas penggunaan untuk klaster yang disediakan

Berikut ini adalah prosedur untuk menetapkan batas penggunaan pada klaster yang disediakan:

Untuk menetapkan batas penggunaan untuk klaster

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Arahkan ke klaster yang disediakan yang ingin Anda tetapkan batasnya.

3. Dari halaman detail klaster, pilih Kelola batas penggunaan dari menu tarik-turun Tindakan. Anda juga dapat memilih tab Maintenance untuk cluster, lalu gulir ke bawah dan pilih Buat batas penggunaan.
 4. Pilih Tambahkan batas untuk batas penggunaan yang ingin Anda tetapkan. Anda dapat menambahkan hingga 4 batas untuk fitur tertentu.
 5. Tetapkan jangka waktu untuk batas penggunaan, yaitu Harian, Mingguan, atau Bulanan.
 6. Tetapkan batas Penggunaan.
 - Untuk penskalaan konkurensi dan pengoptimalan otomatis yang dijalankan menggunakan batas sumber daya komputasi tambahan, batas penggunaan adalah jumlah waktu yang dihabiskan Amazon Redshift menggunakan fitur dalam periode waktu tertentu. Dalam hal ini, batas penggunaan diatur dalam jam dan menit.
 - Untuk Redshift Spectrum, batas penggunaan adalah jumlah data yang dipindai dari Amazon S3. Dalam hal ini, batas penggunaan diatur dalam terabyte (TB).
 - Untuk berbagi data lintas wilayah, batas penggunaan adalah jumlah data yang ditransfer dari Wilayah produsen ke Wilayah konsumen yang dapat ditanyakan konsumen. Dalam hal ini, batas penggunaan diatur dalam terabyte (TB).
 7. Atur Action for Amazon Redshift untuk diambil saat klaster Anda mencapai batas. Ini adalah sebagai berikut:
 - Masuk ke tabel sistem - Menambahkan catatan ke tampilan sistem [SYS_QUERY_HISTORY](#). Anda dapat menanyakan kolom usage_limit dalam tampilan ini untuk menentukan apakah kueri melebihi batas.
 - Peringatan - Menggunakan Amazon SNS untuk mengatur langganan notifikasi dan mengirim notifikasi jika batas dilanggar. Anda dapat memilih topik Amazon SNS yang ada, membuat topik baru, atau melanjutkan tanpa topik.
 - Nonaktifkan fitur - Menonaktifkan fitur. Anda juga dapat memilih untuk menggunakan Amazon SNS untuk mengirim pemberitahuan. Pengguna dapat terus menggunakan cluster untuk tugas-tugas lain.
- Dua tindakan pertama bersifat informasional, tetapi yang terakhir mematikan penggunaan fitur.
8. Pilih Simpan perubahan di bagian bawah halaman untuk menyimpan batas. Jika Anda menetapkan lebih dari satu batas sekaligus, Simpan perubahan akan menyimpan semuanya sekaligus.

Mematikan dan menghapus cluster

Anda dapat mematikan klaster Anda jika Anda ingin menghentikannya agar tidak berjalan dan menimbulkan biaya. Saat Anda memamatikannya, Anda dapat membuat snapshot akhir secara opsional. Jika Anda membuat snapshot akhir, Amazon Redshift akan membuat snapshot manual cluster Anda sebelum memamatikannya. Jika Anda berencana untuk menyediakan cluster baru dengan data dan konfigurasi yang sama dengan yang Anda hapus, Anda memerlukan snapshot manual. Dengan menggunakan snapshot manual, Anda dapat memulihkan snapshot nanti dan melanjutkan menggunakan cluster.

Jika Anda tidak lagi membutuhkan cluster dan datanya, Anda dapat memamatikannya tanpa membuat snapshot akhir. Dalam hal ini, cluster dan data dihapus secara permanen.

Terlepas dari apakah Anda mematikan cluster Anda dengan snapshot manual terakhir, semua snapshot otomatis yang terkait dengan cluster akan dihapus setelah cluster dimatikan. Setiap snapshot manual yang terkait dengan cluster dipertahankan. Setiap snapshot manual yang disimpan, termasuk snapshot akhir opsional, dikenakan biaya pada tingkat penyimpanan Amazon Simple Storage Service jika Anda tidak memiliki cluster lain yang berjalan saat Anda mematikan klaster, atau jika Anda melebihi penyimpanan gratis yang tersedia yang disediakan untuk klaster Amazon Redshift yang sedang berjalan. Untuk informasi selengkapnya tentang biaya penyimpanan snapshot, lihat halaman [harga Amazon Redshift](#).

Menghapus cluster juga menghapus rahasia terkait AWS Secrets Manager .

Untuk menghapus klaster

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster.
3. Pilih cluster yang akan dihapus.
4. Untuk Tindakan, pilih Hapus. Halaman Delete cluster muncul.
5. Pilih Hapus klaster.

Note

Saat Anda menghapus klaster dan memilih untuk membuat snapshot akhir, Amazon Redshift akan menghentikan permintaan penghapusan jika operasi pemulihan sedang berlangsung di

cluster. Jika ini terjadi, Anda dapat menghapus cluster tanpa snapshot akhir, atau Anda dapat menghapusnya dengan snapshot terakhir setelah pemulihan selesai.

Cuplikan dan cadangan Amazon Redshift

Snapshot adalah point-in-time backup dari sebuah cluster. Ada dua jenis snapshot: otomatis dan manual. Amazon Redshift menyimpan snapshot ini secara internal di Amazon S3 dengan menggunakan koneksi Secure Sockets Layer (SSL) terenkripsi.

Amazon Redshift secara otomatis mengambil snapshot tambahan yang melacak perubahan pada cluster sejak snapshot otomatis sebelumnya. Snapshot otomatis menyimpan semua data yang diperlukan untuk memulihkan cluster dari snapshot. Anda dapat membuat jadwal snapshot untuk mengontrol kapan snapshot otomatis diambil, atau Anda dapat mengambil snapshot manual kapan saja.

Saat memulihkan dari snapshot, Amazon Redshift membuat cluster baru dan membuat klaster baru tersedia sebelum semua data dimuat, sehingga Anda dapat segera mulai menanyakan klaster baru. Cluster mengalirkan data sesuai permintaan dari snapshot sebagai respons terhadap kueri aktif, lalu memuat data yang tersisa di latar belakang.

Saat meluncurkan cluster, Anda dapat mengatur periode retensi untuk snapshot otomatis dan manual. Anda dapat mengubah periode retensi default untuk snapshot otomatis dan manual dengan memodifikasi cluster. Anda dapat mengubah periode retensi untuk snapshot manual saat membuat snapshot atau dengan memodifikasi snapshot.

Anda dapat memantau kemajuan snapshot dengan melihat detail snapshot di Konsol Manajemen AWS, atau dengan memanggil [describe-cluster-snapshots](#) CLI atau tindakan API. [DescribeClusterSnapshots](#) Untuk snapshot yang sedang berlangsung, ini menampilkan informasi seperti ukuran snapshot tambahan, kecepatan transfer, waktu yang telah berlalu, dan perkiraan waktu yang tersisa.

Untuk memastikan bahwa cadangan Anda selalu tersedia untuk klaster Anda, Amazon Redshift menyimpan snapshot dalam bucket Amazon S3 yang dikelola secara internal yang dikelola oleh Amazon Redshift. Untuk mengelola biaya penyimpanan, evaluasi berapa hari Anda perlu menyimpan snapshot otomatis dan mengonfigurasi periode retensi yang sesuai. Hapus snapshot manual yang tidak lagi Anda butuhkan. Untuk informasi selengkapnya tentang biaya penyimpanan cadangan, lihat halaman [harga Amazon Redshift](#).

Anda juga dapat membuat dan memulihkan snapshot menggunakan AWS Backup, layanan yang dikelola sepenuhnya yang membantu Anda memusatkan dan mengotomatiskan perlindungan data di seluruh AWS layanan, di cloud, dan di tempat. Untuk informasi selengkapnya, lihat [AWS Backup integrasi dengan Amazon Redshift](#). Untuk informasi tentang AWS Backup, lihat [Apa itu AWS Backup?](#) di Panduan AWS Backup Pengembang.

Bekerja dengan snapshot dan backup di Amazon Redshift Serverless

Amazon Redshift Tanpa Server, seperti kluster yang disediakan, memungkinkan Anda mengambil cadangan sebagai point-in-time representasi objek dan data di namespace. Ada dua jenis cadangan di Amazon Redshift Serverless: snapshot yang dibuat secara manual dan titik pemulihan yang dibuat Amazon Redshift Serverless secara otomatis. Anda dapat menemukan informasi lebih lanjut tentang bekerja dengan snapshot untuk Amazon Redshift Tanpa Server [di](#) Snapshots dan titik pemulihan.

Anda juga dapat memulihkan snapshot dari klaster yang disediakan ke namespace tanpa server. Untuk informasi selengkapnya, lihat [Memulihkan namespace tanpa server dari snapshot](#).

Snapshot otomatis

Saat snapshot otomatis diaktifkan untuk kluster, Amazon Redshift secara berkala mengambil snapshot dari cluster tersebut. Secara default Amazon Redshift mengambil snapshot setiap delapan jam atau mengikuti setiap 5 GB per node perubahan data, atau mana yang lebih dulu. Jika data Anda lebih besar dari 5 GB* jumlah node, jumlah waktu terpendek di antara pembuatan snapshot otomatis adalah 15 menit. Atau, Anda dapat membuat jadwal snapshot untuk mengontrol saat snapshot otomatis diambil. Jika Anda menggunakan jadwal khusus, jumlah waktu minimum antara snapshot otomatis adalah satu jam. Snapshot otomatis diaktifkan secara default saat Anda membuat cluster.

Snapshot otomatis dihapus pada akhir periode retensi. Periode retensi default adalah satu hari, tetapi Anda dapat memodifikasinya dengan menggunakan konsol Amazon Redshift atau secara terprogram dengan menggunakan Amazon Redshift API atau CLI.

Untuk menonaktifkan snapshot otomatis, atur periode retensi ke nol. Jika Anda menonaktifkan snapshot otomatis, Amazon Redshift berhenti mengambil snapshot dan menghapus snapshot otomatis yang ada untuk cluster. Anda tidak dapat menonaktifkan snapshot otomatis untuk tipe RA3 node. Anda dapat mengatur periode retensi otomatis tipe RA3 node dari 1—35 hari.

Hanya Amazon Redshift yang dapat menghapus snapshot otomatis; Anda tidak dapat menghapusnya secara manual. Amazon Redshift menghapus snapshot otomatis di akhir periode retensi snapshot, saat Anda menonaktifkan snapshot otomatis untuk klaster, atau saat Anda

menghapus klaster. Amazon Redshift mempertahankan snapshot otomatis terbaru hingga Anda menonaktifkan snapshot otomatis atau menghapus cluster.

Jika Anda ingin menyimpan snapshot otomatis untuk jangka waktu yang lebih lama, Anda dapat membuatnya sebagai snapshot manual. Snapshot otomatis dipertahankan hingga akhir periode retensi, tetapi snapshot manual yang sesuai dipertahankan hingga Anda menghapusnya secara manual atau hingga akhir periode retensi.

Jadwal snapshot otomatis

Untuk mengontrol secara tepat kapan snapshot diambil, Anda dapat membuat jadwal snapshot dan melampirkannya ke satu atau lebih cluster. Saat Anda mengubah jadwal snapshot, jadwal diubah untuk semua cluster terkait. Jika klaster tidak memiliki jadwal snapshot terlampir, cluster menggunakan jadwal snapshot otomatis default.

Jadwal snapshot adalah seperangkat aturan jadwal. Anda dapat menentukan aturan jadwal sederhana berdasarkan interval tertentu, seperti setiap 8 jam atau setiap 12 jam. Anda juga dapat menambahkan aturan untuk mengambil foto pada hari-hari tertentu dalam seminggu, pada waktu tertentu, atau selama periode tertentu. Aturan juga dapat didefinisikan menggunakan ekspresi cron mirip Unix.

Format jadwal snapshot

Di konsol Amazon Redshift, Anda dapat membuat jadwal snapshot. Kemudian, Anda dapat melampirkan jadwal ke cluster untuk memicu pembuatan snapshot sistem. Jadwal dapat dilampirkan ke beberapa cluster, dan Anda dapat membuat beberapa definisi cron dalam jadwal untuk memicu snapshot.

Anda dapat menentukan jadwal untuk snapshot Anda menggunakan sintaks cron. [Definisi jadwal ini menggunakan sintaks cron mirip Unix yang dimodifikasi](#). Anda menentukan waktu di [Waktu universal terkoordinasi \(UTC\)](#). Anda dapat membuat jadwal dengan frekuensi maksimum satu jam dan presisi minimum satu menit.

Ekspresi cron yang dimodifikasi Amazon Redshift memiliki 3 bidang wajib, yang dipisahkan oleh spasi putih.

Sintaksis

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

Bidang	Nilai-nilai	Wildcard
Menit	0–59	, - * /
Jam	0–23	, - * /
Day-of-month	1–31	, - * ? / L W
Bulan	1–12 atau JAN-DEC	, - * /
Day-of-week	1–7 atau SUN-SAT	, - * ? L #
Tahun	1970–2199	, - * /

Wildcard

- Wildcard , (koma) mencakup nilai tambahan. Di Day-of-week lapangan, MON, WED, FRI akan mencakup Senin, Rabu, dan Jumat. Nilai total dibatasi hingga 24 per bidang.
- Wildcard - (tanda hubung) menentukan rentang. Di Hour lapangan, 1-15 akan mencakup jam 1 hingga 15 dari hari yang ditentukan.
- Wildcard * (bintang) mencakup semua nilai di bidang. Di kolom Hours, * akan mencakup setiap jam.
- Wildcard / (garis miring) menentukan tambahan. Di Hours lapangan, Anda bisa masuk **1/10** untuk menentukan setiap jam ke-10, mulai dari jam pertama hari itu (misalnya, 01:00, 11:00, dan 21:00).
- Wildcard ? (tanda tanya) menentukan satu atau yang lain. Di **Day-of-month** lapangan Anda bisa masuk 7, dan jika Anda tidak peduli hari apa dalam minggu ketujuh, Anda bisa masuk? di Day-of-week lapangan.
- Wildcard L di kolom Day-of-month atau Day-of-week menentukan hari terakhir pada bulan atau minggu.
- Wildcard W di kolom Day-of-month menentukan hari kerja. Di kolom Day-of-month, 3W menentukan hari kerja yang paling dekat dengan pekan ketiga di bulan itu.
- Wildcard # di Day-of-week bidang menentukan contoh tertentu dari hari yang ditentukan dalam seminggu dalam sebulan. Sebagai contoh, 3#2 akan menjadi hari Selasa kedua setiap bulan: 3 mengacu pada hari Selasa karena itu adalah hari ketiga setiap minggu, dan 2 mengacu pada hari kedua dari jenis tersebut dalam bulan tersebut.

Note

Jika Anda menggunakan karakter '#', Anda hanya dapat menentukan satu ekspresi di day-of-week bidang. Misalnya, "3 #1 ,6 #3" tidak valid karena ditafsirkan sebagai dua ekspresi.

Batas

- Anda tidak dapat menentukan kolom Day-of-month dan Day-of-week dalam ekspresi cron yang sama. Jika Anda menentukan sebuah nilai di salah satu kolom, maka Anda harus menggunakan ? (tanda tanya) di kolom yang lain.
- Jadwal snapshot tidak mendukung frekuensi berikut:
 - Snapshot dijadwalkan lebih sering dari 1 per jam.
 - Snapshot dijadwalkan kurang dari 1 per hari (24 jam).

Jika Anda memiliki jadwal yang tumpang tindih yang menghasilkan penjadwalan snapshot dalam jendela 1 jam, kesalahan validasi akan terjadi.

Anda dapat membuat jadwal, Anda dapat menggunakan contoh cron berikut.

Menit	Jam	Hari dalam seminggu	Arti			
0	14-20/1	SEL	Setiap jam antara jam 2 siang dan 8 malam pada hari Selasa.			
0	21	MON-FRI	Setiap malam jam 9 malam Senin-Jumat.			
30	0/6	SAT-MATAHARI	Setiap kenaikan 6 jam pada hari Sabtu dan Minggu dimulai pada 30 menit setelah tengah malam (00:30)			

Menit	Jam	Hari dalam seminggu	Arti			
			hari itu. Ini menghasilkan snapshot pada [00:30, 06:30, 12:30, dan 18:30] setiap hari.			
30	12/4	*	Setiap kenaikan 4 jam mulai pukul 12:30 setiap hari. Ini memutuskan untuk [12:30, 16:30, 20:30].			

Misalnya berjalan sesuai jadwal dengan kenaikan setiap 2 jam mulai pukul 15:15 setiap hari. Ini memutuskan untuk [15:15, 17:15, 19:15, 21:15, 23:15], tentukan:

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

Anda dapat membuat beberapa definisi jadwal cron dalam sebagai jadwal. Misalnya AWS CLI perintah berikut berisi dua jadwal cron dalam satu jadwal.

```
create-snapshot-schedule --schedule-identifier "my-test" --schedule-definition "cron(0 17 SAT,SUN)" "cron(0 9,17 MON-FRI)"
```

Snapshot manual

Anda dapat mengambil snapshot manual kapan saja. Secara default, snapshot manual dipertahankan tanpa batas waktu, bahkan setelah Anda menghapus cluster Anda. Anda dapat menentukan periode retensi saat membuat snapshot manual, atau Anda dapat mengubah periode retensi dengan memodifikasi snapshot. Untuk informasi selengkapnya tentang mengubah periode retensi, lihat [Memodifikasi periode retensi snapshot manual](#).

Jika snapshot dihapus, Anda tidak dapat memulai operasi baru yang mereferensikan snapshot itu. Namun, jika operasi pemulihan sedang berlangsung, operasi pemulihan itu akan berjalan hingga selesai.

Amazon Redshift memiliki kuota yang membatasi jumlah snapshot manual yang dapat Anda buat; kuota ini per akun per Wilayah. AWS AWS Kuota default tercantum di [Kuota dan batas di Amazon Redshift](#).

Penyimpanan snapshot

Karena snapshot menimbulkan biaya penyimpanan, penting bagi Anda untuk menghapusnya saat Anda tidak lagi membutuhkannya. Amazon Redshift menghapus snapshot otomatis dan manual di akhir periode retensi snapshot masing-masing. Anda juga dapat menghapus snapshot manual menggunakan Konsol Manajemen AWS atau dengan perintah [batch-delete-cluster-snapshots](#) CLI.

Anda dapat mengubah periode retensi untuk snapshot manual dengan memodifikasi pengaturan snapshot manual.

Anda bisa mendapatkan informasi tentang berapa banyak penyimpanan yang dikonsumsi snapshot Anda menggunakan Amazon Redshift Console atau menggunakan [perintah CLI deskripsi-penyimpanan](#).

Tidak termasuk tabel dari snapshot

Secara default, semua tabel permanen yang ditentukan pengguna disertakan dalam snapshot. Jika tabel, seperti tabel pementasan, tidak perlu dicadangkan, Anda dapat secara signifikan mengurangi waktu yang diperlukan untuk membuat snapshot dan memulihkan dari snapshot. Anda juga mengurangi ruang penyimpanan di Amazon S3 dengan menggunakan tabel tanpa cadangan. Untuk membuat tabel tanpa cadangan, sertakan parameter `BACKUP NO` saat Anda membuat tabel. Untuk informasi selengkapnya, lihat [MEMBUAT TABEL](#) dan [MEMBUAT TABEL AS](#) di Panduan Pengembang Database Amazon Redshift.

Note

Tabel tanpa cadangan tidak didukung untuk kluster yang RA3 disediakan dan grup kerja Amazon Redshift Tanpa Server. Tabel yang ditandai sebagai no-backup di RA3 cluster atau grup kerja tanpa server diperlakukan sebagai tabel permanen yang akan selalu dicadangkan saat mengambil snapshot, dan selalu dipulihkan saat memulihkan dari snapshot. Untuk

menghindari biaya snapshot untuk tabel tanpa cadangan, potong sebelum mengambil snapshot.

Membuat snapshot manual

Anda dapat membuat snapshot manual cluster dari daftar snapshot sebagai berikut. Atau, Anda dapat mengambil snapshot cluster di panel konfigurasi cluster. Untuk informasi selengkapnya, lihat [Cuplikan dan cadangan Amazon Redshift](#).

Note

Tabel tanpa cadangan tidak didukung untuk klaster yang RA3 disediakan dan grup kerja Amazon Redshift Tanpa Server. Tabel yang ditandai sebagai no-backup di RA3 cluster atau grup kerja tanpa server diperlakukan sebagai tabel permanen yang akan selalu dicadangkan saat mengambil snapshot, dan selalu dipulihkan saat memulihkan dari snapshot. Untuk menghindari biaya snapshot untuk tabel tanpa cadangan, potong sebelum mengambil snapshot.

Untuk membuat snapshot manual

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Clusters, Snapshots, lalu pilih Create snapshot. Halaman snapshot untuk membuat snapshot manual ditampilkan.
3. Masukkan properti definisi snapshot, lalu pilih Buat snapshot. Mungkin perlu beberapa waktu agar snapshot tersedia.

Membuat jadwal snapshot

Amazon Redshift mengambil snapshot otomatis dan bertahap dari data Anda secara berkala dan menyimpannya ke Amazon S3. Selain itu, Anda dapat mengambil snapshot manual dari data Anda kapan pun Anda mau.

Semua tugas snapshot di konsol Amazon Redshift dimulai dari daftar snapshot. Anda dapat memfilter daftar dengan menggunakan rentang waktu, jenis snapshot, dan cluster yang terkait dengan snapshot. Selain itu, Anda dapat mengurutkan daftar berdasarkan tanggal, ukuran, dan jenis

snapshot. Bergantung pada jenis snapshot yang Anda pilih, Anda mungkin memiliki opsi berbeda yang tersedia untuk bekerja dengan snapshot.

Untuk mengontrol secara tepat kapan snapshot diambil, Anda dapat membuat jadwal snapshot dan melampirkannya ke satu atau lebih cluster. Anda dapat melampirkan jadwal saat membuat cluster atau dengan memodifikasi cluster. Untuk informasi selengkapnya, lihat [Jadwal snapshot otomatis](#).

Untuk membuat jadwal snapshot

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster, Snapshots, lalu pilih tab Jadwal Snapshot. Jadwal snapshot ditampilkan.
3. Pilih Tambahkan jadwal untuk menampilkan halaman untuk menambahkan jadwal.
4. Masukkan properti definisi jadwal, lalu pilih Tambahkan jadwal.
5. Pada halaman yang muncul, Anda dapat melampirkan cluster ke jadwal snapshot baru Anda, lalu pilih OK.

Berbagi snapshot

Anda dapat membagikan snapshot manual yang ada dengan akun AWS pelanggan lain dengan mengotorisasi akses ke snapshot. Anda dapat mengotorisasi hingga 20 untuk setiap snapshot dan 100 untuk setiap AWS Key Management Service (AWS KMS) kunci. Artinya, jika Anda memiliki 10 snapshot yang dienkrpsi dengan satu kunci KMS, maka Anda dapat mengotorisasi 10 AWS akun untuk memulihkan setiap snapshot, atau kombinasi lain yang menambahkan hingga 100 akun dan tidak melebihi 20 akun untuk setiap snapshot. Seseorang yang masuk sebagai pengguna di salah satu akun resmi kemudian dapat menjelaskan snapshot atau mengembalikannya untuk membuat cluster Amazon Redshift baru di bawah akun mereka. Misalnya, jika Anda menggunakan akun AWS pelanggan terpisah untuk produksi dan pengujian, pengguna dapat masuk menggunakan akun produksi dan berbagi snapshot dengan pengguna di akun pengujian. Seseorang yang masuk sebagai pengguna akun pengujian kemudian dapat memulihkan snapshot untuk membuat cluster baru yang dimiliki oleh akun pengujian untuk pengujian atau pekerjaan diagnostik.

Snapshot manual dimiliki secara permanen oleh akun AWS pelanggan tempat ia dibuat. Hanya pengguna di akun yang memiliki snapshot yang dapat mengotorisasi akun lain untuk mengakses snapshot, atau untuk mencabut otorisasi. Pengguna di akun resmi hanya dapat mendeskripsikan atau memulihkan snapshot apa pun yang telah dibagikan dengan mereka; mereka tidak dapat

menyalin atau menghapus snapshot yang telah dibagikan dengan mereka. Otorisasi tetap berlaku sampai pemilik snapshot mencabutnya. Jika otorisasi dicabut, pengguna yang sebelumnya berwenang kehilangan visibilitas snapshot dan tidak dapat meluncurkan tindakan baru apa pun yang mereferensikan snapshot. Jika akun sedang dalam proses memulihkan snapshot saat akses dicabut, pemulihan berjalan hingga selesai. Anda tidak dapat menghapus snapshot saat memiliki otorisasi aktif; Anda harus terlebih dahulu mencabut semua otorisasi.

AWS Akun pelanggan selalu diizinkan untuk mengakses snapshot yang dimiliki oleh akun. Upaya untuk mengotorisasi atau mencabut akses ke akun pemilik akan menerima kesalahan. Anda tidak dapat memulihkan atau menggambarkan snapshot yang dimiliki oleh akun AWS pelanggan yang tidak aktif.

Setelah Anda memiliki akses resmi ke akun AWS pelanggan, tidak ada pengguna di akun tersebut yang dapat melakukan tindakan apa pun pada snapshot kecuali mereka berperan dengan kebijakan yang memungkinkan mereka melakukannya.

- Pengguna di akun pemilik snapshot dapat mengotorisasi dan mencabut akses ke snapshot hanya jika mereka berperan dengan kebijakan IAM yang memungkinkan mereka melakukan tindakan tersebut dengan spesifikasi sumber daya yang menyertakan snapshot. Misalnya, kebijakan berikut memungkinkan pengguna atau peran dalam AWS akun 012345678912 untuk mengotorisasi akun lain untuk mengakses snapshot bernama: my-snapshot20130829

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "redshift:AuthorizeSnapshotAccess",
        "redshift:RevokeSnapshotAccess"
      ],
      "Resource": [
        "arn:aws:redshift:us-east-1:012345678912:snapshot:*/my-snapshot20130829"
      ]
    }
  ]
}
```

- Pengguna di AWS akun yang telah dibagikan snapshot tidak dapat melakukan tindakan pada snapshot itu kecuali mereka memiliki izin yang mengizinkan tindakan tersebut. Anda dapat melakukan ini dengan menetapkan kebijakan untuk peran dan mengambil peran.
- Untuk membuat daftar atau mendeskripsikan snapshot, mereka harus memiliki kebijakan IAM yang memungkinkan tindakan tersebut. `DescribeClusterSnapshots` Kode berikut menunjukkan contoh:

JSON

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

- Untuk memulihkan snapshot, pengguna harus mengambil peran dengan kebijakan IAM yang memungkinkan `RestoreFromClusterSnapshot` tindakan dan memiliki elemen sumber daya yang mencakup cluster yang mereka coba buat dan snapshot. Misalnya, jika pengguna dalam akun 012345678912 telah berbagi snapshot my-snapshot20130829 dengan akun219876543210, untuk membuat klaster dengan memulihkan snapshot, pengguna dalam akun 219876543210 harus mengambil peran dengan kebijakan seperti berikut:

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"
]
}
]
}
```

- Setelah akses ke snapshot dicabut dari AWS akun, tidak ada pengguna di akun itu yang dapat mengakses snapshot tersebut. Hal ini terjadi bahkan jika akun tersebut memiliki kebijakan IAM yang memungkinkan tindakan pada sumber snapshot yang dibagikan sebelumnya.

Berbagi snapshot cluster menggunakan konsol

Di konsol, Anda dapat mengotorisasi pengguna lain untuk mengakses snapshot manual yang Anda miliki, dan nanti Anda dapat mencabut akses itu ketika tidak lagi diperlukan.

Untuk berbagi snapshot dengan akun lain

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster, Snapshots, lalu pilih snapshot manual untuk dibagikan.
3. Untuk Tindakan, pilih Pengaturan snapshot manual untuk menampilkan properti snapshot manual.
4. Masukkan akun atau akun untuk dibagikan di bagian Kelola akses, lalu pilih Simpan.

Pertimbangan keamanan untuk berbagi snapshot terenkripsi

Saat Anda memberikan akses ke snapshot terenkripsi, Redshift mengharuskan kunci terkelola pelanggan AWS KMS yang digunakan untuk membuat snapshot dibagikan dengan akun atau akun yang melakukan pemulihan. Jika kunci tidak dibagikan, mencoba memulihkan snapshot menghasilkan kesalahan yang ditolak akses. Akun penerima tidak memerlukan izin tambahan untuk memulihkan snapshot bersama. Saat Anda mengotorisasi akses snapshot dan membagikan kunci, akses otorisasi identitas harus memiliki `kms:DescribeKey` izin pada kunci yang digunakan untuk mengenkripsi snapshot. Izin ini dijelaskan secara lebih rinci dalam [AWS KMS izin](#). Untuk informasi selengkapnya, lihat [DescribeKey](#) di dokumentasi referensi Amazon Redshift API.

Kebijakan kunci yang dikelola pelanggan dapat diperbarui secara terprogram atau di konsol. AWS Key Management Service

 Note

Jika Anda menggunakan kunci KMS default, Anda tidak perlu mengambil tindakan atau mengubah apa pun untuk berbagi snapshot. AWS KMS

Mengizinkan akses ke kunci AWS KMS untuk snapshot terenkripsi

Untuk membagikan kunci terkelola pelanggan AWS KMS untuk snapshot terenkripsi, perbarui kebijakan kunci dengan melakukan langkah-langkah berikut:

1. Perbarui kebijakan kunci KMS dengan Nama Sumber Daya Amazon (ARN) akun AWS yang Anda bagikan `Principal` seperti dalam kebijakan kunci KMS.
2. Izinkan `kms:Decrypt` tindakan.

Dalam contoh kebijakan kunci berikut, pengguna 111122223333 adalah pemilik kunci KMS, dan pengguna 444455556666 adalah akun tempat kunci dibagikan. Kebijakan kunci ini memberikan akses AWS akun ke contoh kunci KMS dengan menyertakan ARN untuk identitas akun AWS root bagi 444455556666 pengguna `Principal` sebagai kebijakan, dan dengan mengizinkan `kms:Decrypt` tindakan.

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": "*"
    }
}
```

Setelah akses diberikan ke kunci KMS yang dikelola pelanggan, akun yang mengembalikan snapshot terenkripsi harus membuat peran AWS Identity and Access Management (IAM), atau pengguna, jika belum memilikinya. Selain itu, AWS akun tersebut juga harus melampirkan kebijakan IAM ke peran IAM atau pengguna yang memungkinkan mereka memulihkan snapshot database terenkripsi, menggunakan kunci KMS Anda.

Untuk informasi selengkapnya tentang memberikan akses ke AWS KMS kunci, lihat [Mengizinkan pengguna di akun lain menggunakan kunci KMS](#), di panduan AWS Key Management Service pengembang.

Untuk ikhtisar kebijakan utama, lihat [Cara Amazon Redshift menggunakan AWS KMS](#).

Menyalin snapshot otomatis

Snapshot otomatis dihapus secara otomatis saat periode retensi berakhir, saat Anda menonaktifkan snapshot otomatis, atau saat Anda menghapus kluster. Jika Anda ingin menyimpan snapshot otomatis, Anda dapat menyalinnya ke snapshot manual.

Untuk menyalin snapshot otomatis

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster, Snapshots, lalu pilih snapshot yang akan disalin.
3. Untuk Tindakan, pilih Salin snapshot otomatis untuk menyalin snapshot.
4. Perbarui properti snapshot baru, lalu pilih Salin.

Menyalin snapshot ke Wilayah lain AWS

Anda dapat mengonfigurasi Amazon Redshift untuk secara otomatis menyalin snapshot (otomatis atau manual) untuk cluster ke Wilayah lain. AWS Ketika snapshot dibuat di AWS Region utama

cluster, itu disalin ke Region sekunder AWS . Kedua AWS Wilayah tersebut masing-masing dikenal sebagai AWS Wilayah sumber dan AWS Wilayah tujuan. Jika Anda menyimpan salinan snapshot Anda di AWS Wilayah lain, Anda dapat memulihkan klaster Anda dari data terbaru jika ada yang memengaruhi AWS Wilayah utama. Anda dapat mengonfigurasi klaster Anda untuk menyalin snapshot ke hanya satu AWS Wilayah tujuan pada satu waktu. Untuk daftar Wilayah Amazon Redshift, lihat [Wilayah dan titik akhir](#) di. Referensi Umum Amazon Web Services

Saat Anda mengaktifkan Amazon Redshift untuk secara otomatis menyalin snapshot ke AWS Wilayah lain, Anda menentukan AWS Wilayah tujuan untuk menyalin snapshot. Untuk snapshot otomatis, Anda juga dapat menentukan periode retensi untuk menyimpannya di AWS Wilayah tujuan. Setelah snapshot otomatis disalin ke AWS Wilayah tujuan dan mencapai periode waktu penyimpanan di sana, snapshot tersebut dihapus dari Wilayah tujuan AWS . Melakukan hal ini membuat penggunaan snapshot Anda tetap rendah. Untuk menyimpan snapshot otomatis untuk waktu yang lebih singkat atau lebih lama di AWS Wilayah tujuan, ubah periode retensi ini.

Periode penyimpanan yang Anda tetapkan untuk snapshot otomatis yang disalin ke AWS Wilayah tujuan terpisah dari periode penyimpanan untuk snapshot otomatis di Wilayah sumber. AWS Periode retensi default untuk snapshot yang disalin adalah tujuh hari. Periode tujuh hari itu hanya berlaku untuk snapshot otomatis. Di AWS Wilayah sumber dan tujuan, snapshot manual dihapus pada akhir periode retensi snapshot atau saat Anda menghapusnya secara manual.

Anda dapat menonaktifkan salinan snapshot otomatis untuk cluster kapan saja. Saat Anda menonaktifkan fitur ini, snapshot tidak lagi disalin dari Wilayah sumber ke AWS Wilayah tujuan AWS . Setiap snapshot otomatis yang disalin ke AWS Wilayah tujuan akan dihapus saat mencapai batas periode retensi, kecuali Anda membuat salinan snapshot manual dari mereka. Cuplikan manual ini, dan snapshot manual apa pun yang disalin dari AWS Wilayah tujuan, disimpan di AWS Wilayah tujuan hingga Anda menghapusnya secara manual.

Untuk mengubah AWS Wilayah tujuan tempat Anda menyalin snapshot, pertama nonaktifkan fitur salin otomatis. Kemudian aktifkan kembali, tentukan Wilayah tujuan AWS baru.

Setelah snapshot disalin ke AWS Wilayah tujuan, itu menjadi aktif dan tersedia untuk tujuan restorasi.

Untuk menyalin snapshot untuk kluster yang AWS KMS dienkripsi —ke AWS Wilayah lain, buat hibah untuk Amazon Redshift untuk menggunakan kunci terkelola pelanggan di Wilayah tujuan. AWS Kemudian pilih hibah itu saat Anda mengaktifkan penyalinan snapshot di Wilayah sumber AWS . Untuk informasi selengkapnya tentang mengonfigurasi hibah salinan snapshot, lihat. [Menyalin AWS KMS—snapshot terenkripsi ke yang lain AWS Region](#)

Memulihkan cluster dari snapshot

Snapshot berisi data dari database apa pun yang berjalan di cluster Anda. Ini juga berisi informasi tentang cluster Anda, termasuk jumlah node, tipe node, dan nama pengguna admin. Jika Anda memulihkan klaster dari snapshot, Amazon Redshift menggunakan informasi klaster untuk membuat cluster baru. Kemudian mengembalikan semua database dari data snapshot.

Note

Tabel tanpa cadangan tidak didukung untuk klaster yang RA3 disediakan dan grup kerja Amazon Redshift Tanpa Server. Tabel yang ditandai sebagai no-backup di RA3 cluster atau grup kerja tanpa server diperlakukan sebagai tabel permanen yang akan selalu dicadangkan saat mengambil snapshot, dan selalu dipulihkan saat memulihkan dari snapshot.

Untuk cluster baru yang dibuat dari snapshot asli, Anda dapat memilih konfigurasi, seperti jenis node dan jumlah node. Cluster dipulihkan di AWS Wilayah yang sama dan Availability Zone acak yang dipilih sistem, kecuali Anda menentukan Availability Zone lain dalam permintaan Anda. Saat memulihkan klaster dari snapshot, Anda dapat memilih trek pemeliharaan yang kompatibel untuk klaster baru secara opsional.

Note

Saat Anda mengembalikan snapshot ke cluster dengan konfigurasi yang berbeda, snapshot harus diambil pada cluster dengan versi cluster 1.0.10013, atau yang lebih baru.

Saat pemulihan sedang berlangsung, peristiwa biasanya dipancarkan dalam urutan berikut:

1. `RESTORE_STARTED - REDSHIFT-EVENT-2008` dikirim ketika proses pemulihan dimulai.
2. `RESTORE_SUCCEEDED — REDSHIFT-EVENT-3003` dikirim ketika cluster baru telah dibuat.

Cluster tersedia untuk kueri.

3. `DATA_TRANSFER_COMPLETED — REDSHIFT-EVENT-3537` dikirim saat transfer data selesai.

 Note

RA3 cluster hanya memancarkan peristiwa `RESTORE_STARTED` dan `RESTORE_SUCCEEDED`. Tidak ada transfer data eksplisit yang harus dilakukan setelah `RESTORE` berhasil karena tipe RA3 node menyimpan data di penyimpanan terkelola Amazon Redshift. Dengan RA3 node, data terus ditransfer antara RA3 node dan penyimpanan terkelola Amazon Redshift sebagai bagian dari pemrosesan kueri normal. RA3 node menyimpan data panas secara lokal dan menyimpan blok yang lebih jarang ditanyakan di penyimpanan terkelola Amazon Redshift secara otomatis.

Anda dapat memantau kemajuan pemulihan dengan memanggil operasi [DescribeClusters](#) API, atau melihat detail klaster di Konsol Manajemen AWS. Untuk pemulihan yang sedang berlangsung, ini menampilkan informasi seperti ukuran data snapshot, kecepatan transfer, waktu yang telah berlalu, dan perkiraan waktu yang tersisa. Untuk deskripsi metrik ini, lihat [RestoreStatus](#).

Anda tidak dapat menggunakan snapshot untuk mengembalikan klaster aktif ke status sebelumnya.

 Note

Saat Anda mengembalikan snapshot ke cluster baru, grup keamanan default dan grup parameter digunakan kecuali Anda menentukan nilai yang berbeda.

Anda mungkin ingin mengembalikan snapshot ke cluster dengan konfigurasi berbeda karena alasan berikut:

- Ketika sebuah cluster terdiri dari tipe node yang lebih kecil dan Anda ingin mengkonsolidasikannya menjadi tipe node yang lebih besar dengan node yang lebih sedikit.
- Ketika Anda telah memantau beban kerja Anda dan menentukan kebutuhan untuk pindah ke tipe node dengan lebih banyak CPU dan penyimpanan.
- Bila Anda ingin mengukur kinerja beban kerja pengujian dengan tipe node yang berbeda.

Restore memiliki kendala berikut:

- Konfigurasi node baru harus memiliki penyimpanan yang cukup untuk data yang ada. Bahkan ketika Anda menambahkan node, konfigurasi baru Anda mungkin tidak memiliki penyimpanan yang cukup karena cara data didistribusikan kembali.

- Operasi pemulihan memeriksa apakah snapshot dibuat pada versi cluster yang kompatibel dengan versi cluster dari cluster baru. Jika cluster baru memiliki tingkat versi yang terlalu dini, maka operasi pemulihan gagal dan melaporkan lebih banyak informasi dalam pesan kesalahan.
- Konfigurasi yang mungkin (jumlah node dan tipe node) yang dapat Anda pulihkan ditentukan oleh jumlah node di cluster asli dan tipe node target dari cluster baru. Untuk menentukan kemungkinan konfigurasi yang tersedia, Anda dapat menggunakan konsol Amazon Redshift atau perintah `describe-node-configuration-options` AWS CLI dengan `action-type restore-cluster`. Untuk informasi selengkapnya tentang pemulihan menggunakan konsol Amazon Redshift, lihat. [Memulihkan cluster dari snapshot](#)

Langkah-langkah berikut mengambil sebuah cluster dengan banyak node dan mengkonsolidasikannya menjadi tipe node yang lebih besar dengan jumlah node yang lebih kecil menggunakan AWS CLI. Untuk contoh ini, kita mulai dengan cluster sumber 24 node. Dalam hal ini, misalkan kita sudah membuat snapshot dari cluster ini dan ingin mengembalikannya ke tipe node yang lebih besar.

1. Jalankan perintah berikut untuk mendapatkan rincian cluster 24-node kami.

```
aws redshift describe-clusters --region eu-west-1 --cluster-identifier
mycluster-123456789012
```

2. Jalankan perintah berikut untuk mendapatkan detail snapshot.

```
aws redshift describe-cluster-snapshots --region eu-west-1 --snapshot-identifier
mycluster-snapshot
```

3. Jalankan perintah berikut untuk menjelaskan opsi yang tersedia untuk snapshot ini.

```
aws redshift describe-node-configuration-options --snapshot-identifier mycluster-
snapshot --region eu-west-1 --action-type restore-cluster
```

Perintah ini mengembalikan daftar opsi dengan jenis node yang direkomendasikan, jumlah node, dan pemanfaatan disk untuk setiap opsi. Untuk contoh ini, perintah sebelumnya mencantumkan konfigurasi node yang mungkin berikut. Kami memilih untuk mengembalikan ke cluster tiga simpul.

```
{
  "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. Jalankan perintah berikut untuk mengembalikan snapshot ke dalam konfigurasi cluster yang kita pilih. Setelah cluster ini dipulihkan, kami memiliki konten yang sama dengan cluster sumber, tetapi data telah dikonsolidasikan menjadi tiga dc2.8xlarge node.

```

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

```

Jika Anda memiliki node cadangan, misalnya node yang DC2 dicadangkan, Anda dapat meningkatkan ke node yang RA3 dicadangkan. Anda dapat melakukan ini saat memulihkan dari

snapshot atau melakukan perubahan ukuran elastis. Anda dapat menggunakan konsol untuk memandu Anda melalui proses ini. Untuk informasi selengkapnya tentang memutakhirkan ke RA3 node, lihat [Memutakhirkan ke tipe RA3 node](#).

Untuk memulihkan cluster dari snapshot di konsol

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster, Snapshots, lalu pilih snapshot untuk dipulihkan.
3. Pilih Pulihkan dari snapshot untuk melihat konfigurasi Cluster dan nilai detail Cluster dari cluster baru yang akan dibuat menggunakan informasi snapshot.
4. Perbarui properti cluster baru, lalu pilih Pulihkan cluster dari snapshot.

Setelah memulihkan snapshot cluster Anda, gudang data yang dipulihkan dienkripsi dengan kunci AWS KMS kustom yang sama yang digunakan pada saat snapshot diambil. Jika snapshot tidak memiliki kunci KMS khusus, logika enkripsi cadangan Amazon Redshift bergantung pada faktor-faktor berikut:

- Jenis gudang data Amazon Redshift tempat Anda memulihkan snapshot.
- Jenis enkripsi cluster pada saat snapshot diambil.

Untuk mempelajari cara gudang data Anda dienkripsi setelah Anda memulihkannya dari snapshot klaster, lihat tabel berikut:

Jenis tujuan	Jenis enkripsi snapshot	Jenis enkripsi tujuan
Cluster yang disediakan	Dienkripsi dengan Kunci yang dikelola AWS	Dienkripsi dengan Kunci yang dikelola AWS
Cluster yang disediakan	Dienkripsi dengan Kunci milik AWS	Dienkripsi dengan Kunci milik AWS
Namespace tanpa server	Dienkripsi dengan Kunci yang dikelola AWS	Dienkripsi dengan Kunci milik AWS
Namespace tanpa server	Dienkripsi dengan Kunci milik AWS	Dienkripsi dengan Kunci milik AWS

Jika AWS Secrets Manager mengelola kata sandi admin klaster Anda pada saat snapshot diambil, Anda harus terus menggunakan AWS Secrets Manager untuk mengelola kata sandi admin. Anda dapat memilih untuk tidak menggunakan rahasia setelah memulihkan klaster dengan memperbarui kredensi admin cluster di halaman detail cluster.

Jika Anda memiliki node cadangan, Anda dapat memutakhirkan ke node yang RA3 dicadangkan. Anda dapat melakukan ini saat memulihkan dari snapshot atau melakukan perubahan ukuran elastis. Anda dapat menggunakan konsol untuk memandu Anda melalui proses ini. Untuk informasi selengkapnya tentang memutakhirkan ke RA3 node, lihat [Memutakhirkan ke tipe RA3 node](#).

Memulihkan tabel dari snapshot

Anda dapat memulihkan satu tabel dari snapshot alih-alih memulihkan seluruh cluster. Saat memulihkan satu tabel dari snapshot, Anda menentukan snapshot sumber, database, skema, dan nama tabel, serta database target, skema, dan nama tabel baru untuk tabel yang dipulihkan.

Note

Tabel tanpa cadangan tidak didukung untuk klaster yang RA3 disediakan dan grup kerja Amazon Redshift Tanpa Server. Tabel yang ditandai sebagai no-backup di RA3 cluster atau grup kerja tanpa server diperlakukan sebagai tabel permanen yang akan selalu dicadangkan saat mengambil snapshot, dan selalu dipulihkan saat memulihkan dari snapshot. Namun, pemulihan selektif tabel tanpa cadangan tidak didukung.

Nama tabel baru tidak bisa menjadi nama tabel yang ada. Untuk mengganti tabel yang ada dengan tabel yang dipulihkan dari snapshot, ganti nama atau jatuhkan tabel yang ada sebelum Anda mengembalikan tabel dari snapshot.

Tabel target dibuat menggunakan definisi kolom tabel sumber, atribut tabel, dan atribut kolom kecuali untuk kunci asing. Untuk mencegah konflik karena dependensi, tabel target tidak mewarisi kunci asing dari tabel sumber. Dependensi apa pun, seperti tampilan atau izin yang diberikan pada tabel sumber, tidak diterapkan ke tabel target.

Jika pemilik tabel sumber ada, maka pengguna database tersebut adalah pemilik tabel dipulihkan, asalkan pengguna memiliki izin yang cukup untuk menjadi pemilik relasi dalam database dan skema yang ditentukan. Jika tidak, tabel yang dipulihkan dimiliki oleh pengguna admin yang dibuat saat cluster diluncurkan.

Tabel yang dipulihkan kembali ke keadaan semula pada saat cadangan diambil. Ini termasuk aturan visibilitas transaksi yang ditentukan oleh kepatuhan Amazon Redshift terhadap isolasi [serial](#), yang berarti bahwa data akan segera terlihat dalam transaksi penerbangan yang dimulai setelah pencadangan.

Memulihkan tabel dari snapshot memiliki batasan berikut:

- Anda dapat memulihkan tabel hanya ke cluster yang sedang berjalan aktif dan dari snapshot yang diambil dari cluster itu.
- Anda dapat mengembalikan hanya satu tabel pada satu waktu.
- Anda tidak dapat memulihkan tabel dari snapshot cluster yang diambil sebelum cluster diubah ukurannya. Pengecualian adalah Anda dapat mengembalikan tabel setelah mengubah ukuran elastis jika jenis simpul tidak berubah.
- Dependensi apa pun, seperti tampilan atau izin yang diberikan pada tabel sumber, tidak diterapkan ke tabel target.
- Jika keamanan tingkat baris dihidupkan untuk tabel dipulihkan, Amazon Redshift mengembalikan tabel dengan keamanan tingkat baris diaktifkan.

Untuk mengembalikan tabel dari snapshot

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Clusters, lalu pilih cluster yang ingin Anda gunakan untuk mengembalikan tabel.
3. Untuk Tindakan, pilih Pulihkan tabel untuk menampilkan halaman Pulihkan tabel.
4. Masukkan informasi tentang snapshot, tabel sumber, dan tabel target yang akan digunakan, lalu pilih Pulihkan tabel.

Example Contoh: Memulihkan tabel dari snapshot menggunakan AWS CLI

Contoh berikut menggunakan `restore-table-from-cluster-snapshot` AWS CLI perintah untuk mengembalikan `my-source-table` tabel dari `sample-database` skema di `my-snapshot-id` Anda dapat menggunakan AWS CLI perintah `describe-table-restore-status` untuk meninjau status operasi pemulihan Anda. Contoh mengembalikan snapshot ke `mycluster-example` cluster dengan nama tabel baru. `my-new-table`

```
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
```

Memulihkan namespace tanpa server dari snapshot

Memulihkan namespace tanpa server dari snapshot menggantikan semua database namespace dengan database di snapshot. Untuk informasi selengkapnya tentang snapshot tanpa server, lihat [Snapshot](#) dan titik pemulihan. Amazon Redshift secara otomatis mengonversi tabel dengan kunci interleaved menjadi kunci majemuk saat Anda memulihkan snapshot klaster yang disediakan ke namespace Amazon Redshift Tanpa Server. Untuk informasi selengkapnya tentang kunci sortir, lihat [Bekerja dengan kunci sortir](#).

Untuk memulihkan snapshot dari klaster yang disediakan ke namespace tanpa server Anda.

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Clusters, Snapshots, lalu pilih snapshot yang akan digunakan.
3. Pilih Pulihkan dari snapshot, Pulihkan ke namespace tanpa server.
4. Pilih namespace yang ingin Anda pulihkan.
5. Konfirmasikan bahwa Anda ingin memulihkan dari snapshot Anda. Pilih pulihkan. Tindakan ini menggantikan semua database di namespace tanpa server dengan data dari klaster yang Anda sediakan.

Mengonfigurasi salinan snapshot lintas wilayah untuk klaster yang tidak terenkripsi

Anda dapat mengonfigurasi Amazon Redshift untuk menyalin snapshot untuk cluster ke Wilayah lain. AWS Untuk mengonfigurasi salinan snapshot lintas wilayah, Anda perlu mengaktifkan fitur salin ini untuk setiap cluster dan mengonfigurasi tempat menyalin snapshot dan berapa lama untuk menyimpan snapshot otomatis atau manual yang disalin di Wilayah tujuan. AWS Saat salinan lintas wilayah diaktifkan untuk klaster, semua snapshot manual dan otomatis baru disalin ke Wilayah yang ditentukan. AWS Nama snapshot yang disalin diawali dengan. **copy:**

Untuk mengonfigurasi snapshot Lintas wilayah

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Clusters, lalu pilih cluster yang ingin Anda pindahkan snapshot.
3. Untuk Tindakan, pilih Konfigurasi snapshot lintas wilayah.

Kotak dialog Configure Cross-region muncul.

4. Untuk Salin snapshot, pilih Ya.
5. Di AWS Wilayah Tujuan, pilih AWS Wilayah untuk menyalin snapshot.
6. Di Periode retensi snapshot otomatis (hari), pilih jumlah hari yang Anda inginkan agar snapshot otomatis disimpan di AWS Wilayah tujuan sebelum dihapus.
7. Di Periode retensi snapshot manual, pilih nilai yang mewakili jumlah hari yang Anda inginkan snapshot manual disimpan di AWS Wilayah tujuan sebelum dihapus. Jika Anda memilih Nilai kustom, periode retensi harus antara 1 hingga 3653 hari.
8. Pilih Simpan.

Mengonfigurasi salinan snapshot lintas wilayah untuk kluster —terenkripsi AWS KMS

Saat meluncurkan kluster Amazon Redshift, Anda dapat mengonfigurasi hibah salinan snapshot untuk kunci root di akun Anda di tujuan. AWS Region Jika Anda tidak mengonfigurasi hibah, snapshot di wilayah tujuan dienkripsi dengan kunci bawaan AWS yang dimiliki. Dengan melakukan ini, Anda mengaktifkan Amazon Redshift untuk melakukan operasi enkripsi di Wilayah tujuan AWS .

Prosedur berikut menjelaskan proses mengaktifkan salinan snapshot lintas wilayah untuk cluster - enkripsi. AWS KMS Untuk informasi selengkapnya tentang enkripsi di Amazon Redshift dan hibah salinan snapshot, lihat. [Menyalin AWS KMS—snapshot terenkripsi ke yang lain AWS Region](#)

Untuk mengonfigurasi snapshot Lintas wilayah untuk kluster —terenkripsi AWS KMS

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Clusters, lalu pilih cluster yang ingin Anda pindahkan snapshot.
3. Untuk Tindakan, pilih Konfigurasi snapshot lintas wilayah.

Kotak dialog Configure Cross-region muncul.

4. Untuk Salin snapshot, pilih Ya.

5. Di AWS Wilayah Tujuan, pilih AWS Wilayah untuk menyalin snapshot.
6. Di Periode retensi snapshot otomatis (hari), pilih jumlah hari yang Anda inginkan agar snapshot otomatis disimpan di AWS Wilayah tujuan sebelum dihapus.
7. Di Periode retensi snapshot manual, pilih nilai yang mewakili jumlah hari yang Anda inginkan snapshot manual disimpan di AWS Wilayah tujuan sebelum dihapus. Jika Anda memilih Nilai kustom, periode retensi harus antara 1 hingga 3653 hari.
8. Pilih Simpan.

Memodifikasi periode retensi snapshot manual

Anda dapat mengubah periode retensi untuk snapshot manual dengan memodifikasi pengaturan snapshot.

Untuk mengubah periode retensi snapshot manual

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster, Snapshots, lalu pilih snapshot manual untuk diubah.
3. Untuk Tindakan, pilih Pengaturan snapshot manual untuk menampilkan properti snapshot manual.
4. Masukkan properti yang direvisi dari definisi snapshot, lalu pilih Simpan.

Memodifikasi periode retensi untuk salinan snapshot lintas wilayah

Setelah mengonfigurasi salinan snapshot lintas wilayah, Anda mungkin ingin mengubah pengaturan. Anda dapat dengan mudah mengubah periode retensi dengan memilih jumlah hari baru dan menyimpan perubahan.

Warning

Anda tidak dapat mengubah AWS Wilayah tujuan setelah salinan snapshot lintas wilayah dikonfigurasi.

Jika Anda ingin menyalin snapshot ke AWS Wilayah lain, nonaktifkan salinan snapshot lintas wilayah terlebih dahulu. Kemudian aktifkan kembali dengan AWS Wilayah tujuan baru dan periode retensi. Setiap snapshot otomatis yang disalin akan dihapus setelah Anda menonaktifkan salinan snapshot lintas wilayah. Dengan demikian, Anda harus menentukan

apakah ada yang ingin Anda simpan dan menyalinnya ke snapshot manual sebelum menonaktifkan salinan snapshot lintas wilayah.

Untuk memodifikasi snapshot Lintas wilayah

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Clusters, lalu pilih cluster yang ingin Anda ubah snapshot.
3. Untuk Tindakan, pilih Konfigurasi snapshot lintas wilayah untuk menampilkan properti snapshot.
4. Masukkan properti yang direvisi dari definisi snapshot, lalu pilih Simpan.

Menghapus snapshot manual

Anda dapat menghapus snapshot manual dengan memilih satu atau lebih snapshot dalam daftar snapshot.

Untuk menghapus snapshot manual

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster, Snapshots, lalu pilih snapshot yang akan dihapus.
3. Untuk Tindakan, pilih Hapus snapshot untuk menghapus snapshot.
4. Konfirmasikan penghapusan snapshot yang terdaftar, lalu pilih Hapus.

Mendaftarkan cluster ke AWS Glue Data Catalog

Anda dapat mendaftarkan seluruh cluster ke AWS Glue Data Catalog dan membuat katalog yang dikelola oleh AWS Glue. Anda dapat mengakses katalog ini dengan mesin SQL apa pun yang mendukung Apache Iceberg REST API. Untuk informasi selengkapnya tentang membuat katalog yang kompatibel dengan Apache Iceberg dari Amazon Redshift, lihat [kompatibilitas Apache Iceberg untuk Amazon Redshift di Panduan Pengembang Database Amazon Redshift](#).

Untuk mendaftarkan cluster ke AWS Glue Data Catalog

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>

2. Pada menu navigasi, pilih Cluster. Cluster untuk akun Anda saat ini AWS Region terdaftar. Subset properti dari setiap cluster ditampilkan dalam kolom dalam daftar. Jika Anda tidak memiliki cluster apa pun, pilih Buat cluster untuk membuatnya.
3. Pilih nama cluster yang ingin Anda daftarkan.
4. Dari Tindakan, pilih Daftar ke AWS Glue Data Catalog. Kotak Daftar ke AWS Glue Data Catalog pop-up muncul.
5. Masukkan ID AWS akun yang ingin Anda daftarkan klaster di bawah ID akun Tujuan. Ini adalah ID akun yang akan menyimpan katalog di AWS Glue Data Catalog.
6. Masukkan nama di bawah Daftar namespace sebagai. Ini akan menjadi nama cluster di Katalog Data.
7. Pilih Pendaftaran. Anda akan dibawa ke AWS Lake Formation konsol.
8. Ikuti proses pembuatan katalog di AWS Lake Formation. Untuk informasi tentang membuat katalog, lihat [Membawa data Amazon Redshift ke AWS Glue Data Catalog dalam Panduan AWS Lake Formation Pengembang](#).

Deployment multi-AZ

Amazon Redshift mendukung beberapa penerapan Availability Zones (Multi-AZ) untuk kluster yang disediakan. RA3 Dengan menggunakan penerapan Multi-AZ, gudang data Amazon Redshift Anda dapat terus beroperasi dalam skenario kegagalan saat peristiwa tak terduga terjadi di Availability Zone. Penerapan multi-AZ menyebarkan sumber daya komputasi di dua Availability Zones (AZs) dan sumber daya komputasi ini dapat diakses melalui satu titik akhir. Jika terjadi kegagalan seluruh Availability Zone, sumber daya komputasi yang tersisa di Availability Zone kedua tersedia untuk melanjutkan pemrosesan beban kerja. Amazon Redshift mengenakan tarif komputasi per jam yang sama RA3 saat menjalankan gudang data multi-AZ. Biaya penyimpanan tetap sama seperti yang dibagikan di semua Availability Zone di dalam dan AWS Region.

Saat ini, Amazon Redshift mendukung nol Recovery Point Objective (RPO) yang memungkinkan data menjadi terkini dan up-to-date jika terjadi kegagalan. Dengan penerapan Multi-AZ, Amazon Redshift semakin meningkatkan kemampuan pemulihan yang ada dan mengurangi Recovery Time Objective (RTO). Hal ini dimungkinkan karena penerapan Multi-AZ dapat pulih lebih cepat dari kegagalan atau bencana sehingga meningkatkan Perjanjian Tingkat Layanan Amazon Redshift (SLA) menjadi 99,99% dibandingkan dengan 99,9% dengan gudang data AZ tunggal.

Menyiapkan penyebaran Multi-AZ

Untuk mengatur penyebaran Multi-AZ, pilih opsi Multi-AZ dan tentukan jumlah node komputasi yang akan disediakan di setiap Availability Zone. Amazon Redshift secara otomatis menyebarkan sumber daya komputasi yang sama di dua Availability Zone dan semua sumber daya komputasi selalu tersedia untuk pemrosesan baca dan tulis selama operasi normal. Hal ini memungkinkan penyebaran multi-AZ untuk bertindak sebagai gudang data tunggal dengan titik akhir tunggal, menghilangkan kebutuhan untuk perubahan aplikasi ketika bencana terjadi. Meskipun penerapan Multi-AZ memproses kueri individual menggunakan sumber daya komputasi yang hanya berada di satu Availability Zone, ia dapat secara otomatis mendistribusikan pemrosesan beberapa kueri simultan ke kedua Availability Zone untuk meningkatkan throughput keseluruhan untuk beban kerja konkurensi yang tinggi.

Anda juga dapat mengonversi gudang data Single-AZ yang ada menjadi gudang data multi-AZ atau sebaliknya. Semuanya tetap sama kecuali sumber daya komputasi tambahan disediakan di Availability Zone kedua. Saat bermigrasi ke Multi-AZ dari cluster single-AZ yang ada, Anda mungkin diminta untuk menggandakan jumlah node cluster yang dibutuhkan, untuk memfasilitasi performa kueri tunggal yang dipertahankan. Sebagian besar beban kerja mengamati peningkatan throughput pemrosesan kueri secara keseluruhan dengan gudang data multi-AZ karena ada dua kali jumlah sumber daya komputasi yang tersedia.

Jika terjadi kegagalan di Availability Zone, Amazon Redshift terus beroperasi dengan menggunakan sumber daya di Availability Zone yang tersisa secara otomatis. Namun, koneksi pengguna mungkin terputus dan harus dibangun kembali. Selain itu, kueri yang berjalan di Availability Zone yang gagal dapat gagal dan harus dicoba ulang. Namun, Anda dapat menyambung kembali ke klaster dan menjadwalkan ulang kueri dengan segera, dan Amazon Redshift akan memproses kueri di Availability Zone yang tersisa. Kueri yang dikeluarkan pada atau setelah kegagalan terjadi mungkin mengalami penundaan runtime saat gudang data multi-AZ pulih.

Note

Untuk mencapai kinerja yang lebih baik dan ketersediaan yang lebih tinggi, kami sarankan Anda menggunakan ISOLASI SNAPSHOT dengan cluster multi-AZ Anda. Untuk informasi selengkapnya, lihat [MEMBUAT DATABASE](#).

Batasan

Gudang data multi-AZ memiliki kemampuan fungsional yang sama dengan gudang data Single-AZ, kecuali untuk batasan berikut yang berlaku untuk gudang data multi-AZ:

- Anda tidak dapat membuat gudang data Multi-AZ yang tidak terenkripsi. Pastikan untuk menambahkan enkripsi saat membuat gudang data multi-AZ baru, mengubah gudang data Single-AZ menjadi gudang data multi-AZ, atau mengubah gudang data Single-AZ menjadi gudang data multi-AZ.
- Anda tidak dapat membuat penyebaran multi-AZ node tunggal untuk salah satu jenis RA3 instans. Pilih 2 atau lebih node per Availability Zone saat membuat penerapan Multi-AZ.
- Amazon Redshift tidak mendukung konfigurasi subnet yang dapat mendukung kurang dari tiga Availability Zone. Dengan kata lain, grup subnet yang dikonfigurasi membutuhkan tiga atau lebih subnet.
- Anda tidak dapat memindahkan penerapan Multi-AZ ke Availability Zone lain. Relokasi akan ditentukan dan dilakukan secara otomatis oleh Amazon Redshift saat menggunakan penerapan Multi-AZ.
- Anda tidak dapat menjeda atau melanjutkan penerapan Multi-AZ.
- Anda tidak dapat menjalankan penerapan Multi-AZ di luar rentang port yang didukung 5431 hingga 5455 dan 8191 hingga 8215.
- Anda tidak dapat menggunakan tampilan STL, SVCS, SVL, SVV, STV dengan penerapan multi-AZ karena hanya mendukung tampilan pemantauan sistem (tampilan SYS_*). Ubah kueri pemantauan Anda untuk menggunakan tampilan pemantauan sistem (tampilan SYS_*).
- Anda tidak dapat melampirkan alamat IP Elastis ke cluster yang ada dengan Multi-AZ diaktifkan.
- Anda tidak dapat mengonversi cluster dengan alamat IP Elastis terlampir dari Single-AZ ke Multi-AZ.
- Penyebaran Multi-AZ Amazon Redshift tersedia dalam hal berikut: Wilayah AWS
 - AS Timur (Ohio) (us-east-2)
 - US East (N. Virginia) (us-east-1)
 - US West (Oregon) (us-west-2)
 - Africa (Cape Town) (af-south-1)
 - Asia Pacific (Hong Kong) (ap-east-1)
 - Asia Pasifik (Taipei) (ap-timur-2)
 - Asia Pasifik (Hyderabad) (ap-south-2)

- Asia Pasifik (Jakarta) (ap-southeast-3)
- Asia Pasifik (Malaysia) (ap-tenggara 5)
- Asia Pasifik (Melbourne) (ap-southeast-4)
- Asia Pasifik (Mumbai) (ap-south-1)
- Asia Pasifik (Osaka) (ap-northeast-3)
- Asia Pasifik (Seoul) (ap-northeast-2)
- Asia Pasifik (Singapura) (ap-southeast-1)
- Asia Pasifik (Sydney) (ap-southeast-2)
- Asia Pasifik (Selandia Baru) (ap-tenggara 6)
- Asia Pasifik (Thailand) (ap-tenggara 7)
- Asia Pasifik (Tokyo) (ap-northeast-1)
- Kanada (Pusat) (ca-central-1)
- Tiongkok (Beijing) (cn-utara-1)
- Tiongkok (Ningxia) (cn-barat laut-1)
- Eropa (Frankfurt) (eu-central-1)
- Eropa (Irlandia) (eu-west-1)
- Eropa (London) (eu-west-2)
- Eropa (Milan) (eu-south-1)
- Eropa (Paris) (eu-west-3)
- Eropa (Spanyol) (eu-south-2)
- Eropa (Stockholm) (eu-north-1)
- Eropa (Zurich) (eu-central-2)
- Israel (Tel Aviv) (il-central-1)
- Meksiko (Tengah) (mx-central-1)
- Middle East (Bahrain) (me-south-1)
- Timur Tengah (UEA) (me-central-1)
- Amerika Selatan (Sao Paulo) (sa-east-1)
- AWS GovCloud (AS-Timur) (us-gov-east-1)
- **AWS GovCloud (AS-Barat) (us-gov-west-1)**

- Gudang data multi-AZ yang dapat diakses publik mendukung 1 grup keamanan VPC lebih sedikit daripada gudang data multi-AZ tunggal dan multi-AZ yang dapat diakses secara pribadi.

Menyiapkan Multi-AZ saat membuat cluster baru

Amazon Redshift Multi-AZ mendukung dua Availability Zone sekaligus. Amazon Redshift secara otomatis memilih Availability Zones berdasarkan konfigurasi grup subnet yang dipilih. Anda dapat mengonversi gudang data Availability Zone tunggal yang ada menjadi gudang data Multi-AZ atau memulihkan dari snapshot untuk mengonfigurasinya menjadi gudang data Multi-AZ.

Menggunakan konsol Amazon Redshift, Anda dapat dengan mudah membuat penerapan Multi-AZ baru. Untuk membuat penyebaran Multi-AZ baru menggunakan konsol Amazon Redshift, pilih opsi Multi-AZ saat membuat gudang data. Tentukan jumlah node komputasi yang diperlukan dalam satu Availability Zone, dan Amazon Redshift akan menerapkan jumlah node tersebut di masing-masing dari dua Availability Zone. Semua node akan digunakan untuk membaca dan menulis pemrosesan beban kerja selama operasi normal. Anda juga dapat menggunakan AWS CLI `create-cluster` perintah untuk membuat gudang data Multi-AZ baru menggunakan `multi-az` parameter.

Anda dapat mengonversi gudang data Single-AZ yang ada menjadi gudang data multi-AZ, Anda dapat menggunakan konsol Amazon Redshift atau perintah menggunakan AWS CLI `modify-cluster` parameter. `multi-az` Atau, Anda dapat memulihkan dari snapshot untuk mengonfigurasi gudang data Single-AZ ke gudang data multi-AZ baik menggunakan konsol Amazon Redshift atau perintah menggunakan parameter. AWS CLI `restore-from-cluster-snapshot` `multi-az`

Penerapan multi-AZ hanya mendukung tipe RA3 node yang menggunakan Amazon Redshift Managed Storage (RMS). Amazon Redshift menyimpan data dalam RMS, yang menggunakan Amazon S3 dan dapat diakses di semua Availability Zone di AWS Region, tanpa harus mereplikasi data di tingkat Amazon Redshift.

Anda dapat mengatur penerapan Multi-AZ saat membuat klaster baru baik menggunakan konsol Amazon Redshift atau. AWS Command Line Interface

Menggunakan konsol

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Dasbor cluster yang disediakan, dan pilih Cluster. Cluster untuk akun Anda saat ini AWS Region terdaftar. Subset properti dari setiap cluster ditampilkan dalam kolom dalam daftar.

3. Pilih tombolnya Buat cluster untuk membuka halaman buat cluster.
4. Masukkan properti untuk cluster Anda. Untuk informasi umum tentang membuat cluster, lihat [Membuat kluster](#).
5. Pilih salah satu jenis RA3 node dari daftar drop-down tipe Node. Opsi konfigurasi AZ menjadi tersedia hanya ketika Anda memilih jenis RA3 node.
6. Di bawah konfigurasi AZ, pilih Multi-AZ.
7. Di bawah Jumlah node per AZ, masukkan setidaknya dua node untuk klaster Anda.
8. Anda memiliki opsi untuk memuat data sampel atau membawa data Anda sendiri:
 - Dalam Data sampel, pilih Muat data sampel untuk memuat kumpulan data sampel ke dalam klaster Amazon Redshift Anda. Amazon Redshift memuat kumpulan data sampel Tikit ke dalam database dev default dan skema publik. Amazon Redshift secara otomatis memuat kumpulan data sampel ke dalam klaster Amazon Redshift Anda. Anda dapat mulai menggunakan editor kueri v2 untuk menanyakan data.
 - Untuk membawa data Anda sendiri ke cluster Amazon Redshift, ikuti langkah-langkah dalam [Membawa data Anda sendiri ke Amazon Redshift](#).
9. Gulir ke bawah ke Konfigurasi tambahan, perluas Jaringan dan keamanan, dan pastikan Anda menerima grup subnet Cluster default atau memilih yang lain. Jika Anda memilih grup subnet cluster lain, pastikan ada 3 Availability Zones di grup subnet yang Anda pilih.
10. Di bawah Konfigurasi tambahan, perluas konfigurasi Database.
11. Untuk menggunakan kunci kustom alih-alih AWS KMS kunci yang AWS dimiliki default, klik Sesuaikan pengaturan enkripsi di bawah Enkripsi basis data.
12. Di bawah Pilih tombol KMS, Anda dapat memilih AWS Key Management Service kunci atau memasukkan ARN. Atau, Anda dapat mengklik Buat AWS Key Management Service kunci di AWS Key Management Service konsol. Untuk informasi selengkapnya tentang membuat kunci KMS, lihat [Membuat Kunci](#) di Panduan AWS Key Management Service Pengembang.
13. Klik Buat cluster. Ketika pembuatan cluster berhasil, Anda dapat melihat detail di halaman detail cluster. Anda dapat menggunakan klien SQL Anda untuk memuat dan menanyakan data.

Menggunakan AWS Command Line Interface

Untuk mengatur Multi-AZ saat membuat cluster menggunakan AWS Command Line Interface

- Dari AWS CLI gunakan `create-cluster` perintah dan `multi-az` parameter sebagai berikut.

```
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
```

Menyiapkan Multi-AZ untuk gudang data yang dipulihkan dari snapshot

Untuk membuat cluster Multi-AZ baru dengan memulihkannya dari snapshot, selesaikan prosedur berikut.

Menggunakan konsol

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Clusters, Snapshots, lalu pilih snapshot yang akan digunakan.
3. Pilih Pulihkan snapshot, Pulihkan ke klaster yang disediakan.
4. Masukkan properti untuk cluster Anda. Untuk informasi umum tentang membuat cluster, lihat [Membuat kluster](#).
5. Pilih salah satu jenis RA3 node dari daftar drop-down tipe Node. Opsi konfigurasi AZ menjadi tersedia hanya ketika Anda memilih jenis RA3 node.
6. Di bawah konfigurasi AZ, pilih Multi-AZ.
7. Di bawah Jumlah node per AZ, masukkan setidaknya dua node untuk klaster Anda.
8. Anda memiliki opsi untuk memuat data sampel atau membawa data Anda sendiri:
 - Dalam Data sampel, pilih Muat data sampel untuk memuat kumpulan data sampel ke dalam klaster Amazon Redshift Anda. Amazon Redshift memuat kumpulan data sampel Tickit ke dalam database dev default dan skema publik. Amazon Redshift secara otomatis

memuat kumpulan data sampel ke dalam kluster Amazon Redshift Anda. Anda dapat mulai menggunakan editor kueri v2 untuk menanyakan data.

- Untuk membawa data Anda sendiri ke cluster Amazon Redshift, ikuti langkah-langkah di [Muat data dari Amazon S3 ke Amazon Redshift](#).
9. Gulir ke bawah ke Konfigurasi tambahan, perluas Jaringan dan keamanan, dan pastikan Anda menerima grup subnet Cluster default atau memilih yang lain. Jika Anda memilih grup subnet cluster lain, pastikan ada 3 Availability Zones di grup subnet yang Anda pilih.
 10. Di bawah Konfigurasi tambahan, perluas konfigurasi Database.
 11. Di bawah enkripsi Database, untuk menggunakan kunci KMS khusus selain kunci bawaan yang AWS dimiliki, klik Sesuaikan pengaturan enkripsi. Opsi ini tidak dipilih secara default.
 12. Di bawah Pilih tombol KMS, Anda dapat memilih AWS Key Management Service kunci atau memasukkan ARN. Atau, Anda dapat mengklik Buat AWS Key Management Service kunci di AWS Key Management Service konsol. Untuk informasi selengkapnya tentang membuat kunci KMS, lihat [Membuat Kunci](#) di Panduan AWS Key Management Service Pengembang.
 13. Klik Pulihkan cluster dari snapshot. Ketika restorasi cluster berhasil, Anda dapat melihat detailnya di halaman detail cluster.

Menggunakan AWS Command Line Interface

- Dari AWS CLI, gunakan `restore-from-cluster-snapshot` perintah sebagai berikut.

```
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/
```

Mengonversi gudang data AZ tunggal menjadi gudang data multi-AZ

Dengan mengonversi gudang data AZ tunggal menjadi gudang data multi-AZ, gudang data Anda akan sangat tersedia dengan jaminan SLA 99,99%. Kinerja kueri individu akan tetap sama bahkan dengan gudang data multi-AZ. Untuk beban kerja konkurensi yang lebih tinggi, Anda akan melihat peningkatan dalam throughput keseluruhan karena Amazon Redshift dapat mengeksekusi permintaan menggunakan sumber daya komputasi di dua Availability Zone.

 Note

Amazon Redshift tidak akan memungkinkan Anda untuk membagi sumber daya komputasi yang ada saat mengonversi dari Single-AZ ke Multi-AZ, atau sebaliknya. Operasi ini tidak didukung untuk mempertahankan kinerja kueri individual yang konsisten.

Menggunakan konsol

Untuk mengonversi cluster Single-AZ ke gudang data multi-AZ menggunakan konsol

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Dasbor cluster yang disediakan, dan pilih Cluster. Cluster untuk akun Anda saat ini AWS Region terdaftar. Subset properti dari setiap cluster ditampilkan dalam kolom dalam daftar.
3. Pilih cluster yang ingin Anda konversi ke penyebaran multi-AZ. Halaman detail cluster muncul.
4. Untuk Tindakan, pilih Aktifkan Multi-AZ. Ringkasan modifikasi muncul. Klik Aktifkan Multi-AZ.
5. Ketika ada kesalahan, lakukan salah satu hal berikut, lalu klik Aktifkan Multi-AZ.
 - Enkripsi cluster - Pilih Properti untuk mengedit pengaturan enkripsi di bagian konfigurasi Database di bawah tab Properties pada halaman detail cluster.
 - Grup subnet - Pilih grup Subnet untuk mengedit pengaturan grup subnet cluster dengan mengklik tautan grup subnet. Jika Anda memilih grup subnet cluster lain, pastikan ada 3 Availability Zones di grup subnet yang Anda pilih.
 - Pengaturan port - Pilih Properti untuk mengedit pengaturan port di bagian konfigurasi Database di bawah tab Properties pada halaman detail cluster.
6. Anda dapat menggunakan klien SQL Anda untuk memuat dan menanyakan data.

Menggunakan AWS Command Line Interface

- Dari AWS CLI, gunakan `modify-cluster` perintah dan `multi-az` parameter sebagai berikut.

```
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
```

Anda tidak dapat menggunakan tampilan STL, SVCS, SVL, SVV, atau STV dengan penerapan multi-AZ karena hanya mendukung tampilan pemantauan sistem (tampilan SYS_*). Ubah kueri pemantauan Anda untuk menggunakan tampilan pemantauan sistem (tampilan SYS_*).

Mengonversi gudang data Multi-AZ menjadi gudang data AZ tunggal

Dengan mengonversi gudang data Multi-AZ menjadi gudang data AZ tunggal, gudang data Anda tidak akan mendapatkan jaminan SLA 99,99% yang ditawarkan Multi-AZ. Kinerja kueri individual akan tetap sama tetapi throughput keseluruhan akan terpengaruh karena sumber daya komputasi di Availability Zone kedua tidak akan tersedia. Anda memiliki opsi untuk mengaktifkan penskalaan konkurensi untuk menskalakan throughput secara otomatis untuk kinerja yang konsisten bahkan dengan AZ tunggal.

Note

Amazon Redshift tidak akan memungkinkan Anda untuk membagi sumber daya komputasi yang ada saat mengonversi dari Single-AZ ke Multi-AZ, atau sebaliknya. Operasi ini tidak didukung untuk mempertahankan kinerja kueri individual yang konsisten.

Menggunakan konsol

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Dasbor cluster yang disediakan, dan pilih Cluster. Cluster untuk akun Anda saat ini AWS Region terdaftar. Subset properti dari setiap cluster ditampilkan dalam kolom dalam daftar.
3. Pilih cluster yang ingin Anda konversi ke penyebaran multi-AZ. Halaman detail cluster muncul.
4. Untuk Tindakan, pilih Nonaktifkan Multi-AZ. Ringkasan modifikasi muncul. Klik Nonaktifkan Multi-AZ.

Menggunakan AWS Command Line Interface

- Dari AWS CLI, gunakan `modify-cluster` perintah dan `no-multi-az` parameter sebagai berikut.

```
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
```

Setelah gudang data Anda dikonversi ke Single-AZ, itu akan kehilangan jaminan SLA 99,99. Secara keseluruhan throughput juga akan terpengaruh. Saat perubahan disimpan, Anda dapat melihat detailnya di halaman detail cluster.

Mengubah ukuran gudang data Multi-AZ

Anda dapat mengubah ukuran gudang data Multi-AZ dan menentukan sejumlah node atau tipe node yang berbeda dari konfigurasi gudang data saat ini.

Menggunakan konsol

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Dasbor cluster yang disediakan, dan pilih Cluster. Cluster untuk akun Anda saat ini AWS Region terdaftar. Subset properti dari setiap cluster ditampilkan dalam kolom dalam daftar.
3. Pilih cluster yang ingin Anda ubah ukuran gudang data Multi-AZ. Halaman detail cluster muncul.
4. Untuk Tindakan, pilih Ubah Ukuran. Halaman cluster Resize muncul.
5. Ikuti instruksi di halaman. Anda dapat mengubah ukuran cluster sekarang, sekali pada waktu tertentu, atau menambah dan mengurangi ukuran cluster Anda sesuai jadwal.
6. Di bawah Konfigurasi baru, pilih salah satu jenis RA3 node dari daftar drop-down tipe Node.
7. Klik Ubah ukuran cluster.

Menggunakan AWS Command Line Interface

Untuk mengubah ukuran gudang data Multi-AZ menggunakan AWS Command Line Interface

- Dari AWS CLI, gunakan `resize-cluster` perintah untuk mengubah jumlah node untuk Availability Zone tunggal sebagai berikut.

```
aws redshift resize-cluster \  
  --cluster-identifier test-maz-11  
  --cluster-type multi-node  
  --node-type ra3.4xlarge  
  --number-of-nodes 6
```

Gagal dalam penerapan Multi-AZ

Gudang data Multi-AZ Anda adalah sumber daya komputasi koleksi yang digunakan secara bersamaan di dua Availability Zone. Sumber daya komputasi yang digunakan di Availability Zone primer disebut sebagai komputasi primer dan yang ada di Availability Zone sekunder disebut sebagai komputasi sekunder. Gudang data multi-AZ dapat pulih secara otomatis tanpa campur tangan pengguna selama peristiwa yang tidak mungkin terjadi seperti Availability Zone atau kegagalan infrastruktur. Proses pemulihan melibatkan kegagalan dari komputasi primer ke komputasi sekunder dan menunjuk sumber daya komputasi sekunder sebagai primer. Selain itu, sumber daya komputasi sekunder baru disediakan di Availability Zone ketiga. Proses pemulihan otomatis diukur dalam hal RTO dan RPO.

- Tujuan waktu pemulihan (RTO) — Waktu yang dibutuhkan sistem untuk kembali ke kondisi kerja setelah bencana. Dengan kata lain, RTO mengukur waktu henti.
- Tujuan titik pemulihan (RPO) — Jumlah data yang dapat hilang (diukur dalam waktu). Untuk gudang data Amazon Redshift Multi-AZ, RPO biasanya nol karena semua data disimpan di Amazon Redshift Managed Storage (RMS), didukung oleh Amazon Simple Storage Service, yang sangat tahan lama dan tersedia secara default.

Note

Kinerja kinerja kueri individu tidak akan berubah setelah failover terjadi. Hasil keseluruhan gudang data Anda akan berkurang untuk waktu yang singkat karena tidak tersedianya sumber daya komputasi di salah satu Availability Zone. Namun, Amazon Redshift akan

secara otomatis memperoleh kapasitas di Availability Zone lain untuk memastikan kapasitas pemrosesan gudang data yang sama dipulihkan.

Selain proses pemulihan otomatis, Anda juga dapat memicu proses ini secara manual untuk gudang data Anda menggunakan opsi komputasi utama Failover. Anda dapat menggunakan pendekatan ini untuk menguji bagaimana Multi-AZ akan membantu aplikasi Anda untuk ketersediaan tinggi yang lebih tinggi dan kontinuitas yang lebih baik.

Menggunakan konsol

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Lakukan salah satu tindakan berikut:
 - Pada menu navigasi, pilih Cluster. Di bawah Cluster, pilih cluster. Halaman detail cluster muncul.
 - Dari dasbor cluster, pilih cluster.
3. Dari Actions, pilih Failover primary compute.
4. Saat diminta, klik Konfirmasi.

Menggunakan AWS Command Line Interface

- Dari AWS CLI, gunakan `failover-primary-compute` perintah sebagai berikut.

```
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
```

Setelah operasi di atas dikonfirmasi, Amazon Redshift akan melakukan langkah yang sama seperti pemulihan otomatis dari Availability Zone atau kegagalan infrastruktur. Proses ini akan menyebabkan node komputasi di Availability Zone primer menjadi tidak tersedia dan sumber daya komputasi di Availability Zone sekunder akan ditetapkan sebagai komputasi utama. Ketika pemulihan cluster berhasil diselesaikan, penyebaran Multi-AZ menjadi tersedia. Gudang data Multi-AZ Anda juga akan

secara otomatis menyediakan komputasi sekunder baru di Availability Zone ketiga lainnya segera setelah tersedia.

Selama proses ini, status cluster di konsol ditampilkan sebagai modifikasi sepanjang waktu, karena cluster secara otomatis pulih dan mengkonfigurasi ulang kembali ke pengaturan penyebaran multi-AZ. Cluster dapat menerima koneksi baru dengan segera. Koneksi yang ada dan kueri dalam pesawat mungkin dibatalkan. Anda dapat mencobanya kembali segera.

Melihat kueri dan beban untuk gudang data multi-AZ

Anda dapat melihat informasi tentang kueri yang berjalan dalam 7 hari terakhir terlepas dari jenis, ukuran, dan status (jeda atau lanjutkan) klaster Anda.

Informasi yang ditampilkan pada halaman kueri dan pemuatan diisi dengan informasi dari tabel sistem Amazon Redshift (tampilan SYS_*). Informasi ini memungkinkan Anda menampilkan informasi tambahan tentang kueri dan penawaran penyimpanan selama 7 hari. Diagnostik kueri menjadi lebih cepat, memungkinkan Anda memfilter data berdasarkan database, nama pengguna, atau jenis pernyataan SQL. Untuk melihat filter dan informasi tambahan ini tentang semua kueri yang dijalankan, perhatikan prasyarat berikut:

- Anda harus terhubung ke database dengan memilih Connect to database.
- Pengguna database Anda harus memiliki peran dan izin sys:operator atau sys:monitor untuk melakukan pemantauan kueri. Untuk informasi tentang peran sistem, lihat peran yang [ditentukan sistem Amazon Redshift di Panduan Pengembang](#) Database Amazon Redshift.

Anda akan melihat filter tambahan dan informasi kueri ini setelah Anda terhubung ke database.

Untuk menampilkan data kinerja kueri dari Kueri dan beban

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Kueri dan beban untuk menampilkan daftar kueri untuk akun Anda.
3. Anda mungkin harus terhubung ke database untuk melihat filter tambahan. Jika diperlukan, klik Connect to database dan ikuti petunjuk untuk menyambung ke database.

Secara default, daftar menampilkan kueri untuk semua cluster Anda selama 24 jam terakhir. Anda dapat mengubah cakupan tanggal yang ditampilkan di konsol.

Untuk menampilkan data kinerja kueri dari pemantauan Query

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster. Di bawah Cluster, pilih cluster.
3. Pilih Pemantauan kueri.
4. Bergantung pada konfigurasi atau versi klaster Anda, Anda mungkin harus terhubung ke database untuk melihat filter tambahan. Jika diperlukan, klik Connect to database dan ikuti petunjuk untuk menyambung ke database.

Memantau kueri dalam penerapan Multi-AZ

Penerapan multi-AZ menggunakan sumber daya komputasi yang digunakan di kedua Availability Zone dan dapat terus beroperasi jika sumber daya di Availability Zone tertentu tidak tersedia. Semua sumber daya komputasi akan digunakan setiap saat. Hal ini memungkinkan operasi penuh di dua Availability Zone secara aktif-aktif untuk operasi baca dan tulis.

Anda dapat menanyakan tampilan SYS_ dalam skema pg_catalog untuk memantau runtime kueri dalam penerapan multi-AZ. Tampilan SYS_ menampilkan aktivitas runtime kueri atau statistik dari cluster primer dan sekunder. Untuk daftar tampilan pemantauan, lihat [Memantau tampilan](#).

Ikuti langkah-langkah berikut untuk memantau runtime kueri untuk setiap Availability Zone dalam penerapan Multi-AZ:

1. Arahkan ke konsol Amazon Redshift dan sambungkan ke database dalam penerapan Multi-AZ Anda dan jalankan kueri melalui editor kueri.
2. Jalankan kueri sampel apa pun pada penerapan Amazon Redshift Multi-AZ.
3. Untuk penerapan Multi-AZ, Anda dapat mengidentifikasi kueri dan Availability Zone di mana ia dijalankan dengan menggunakan kolom compute_type dalam tabel SYS_QUERY_HISTORY. primary adalah singkatan dari query yang dijalankan pada cluster primer dalam penyebaran multi-AZ, dan singkatan sekunder untuk query yang dijalankan pada cluster sekunder dalam penyebaran multi-AZ.

Query berikut menggunakan kolom compute_type untuk memantau query.

```
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;
```

Mengakhiri kueri untuk klaster

Prosedur ini berlaku untuk cluster multi-AZ dan Single-AZ.

Untuk mengakhiri kueri

Anda juga dapat menggunakan halaman Kueri untuk mengakhiri kueri yang sedang berlangsung.

Pengguna database Anda harus memiliki peran dan izin sys:operator untuk mengakhiri kueri yang sedang berjalan. Untuk informasi tentang peran sistem, lihat peran yang [ditentukan sistem Amazon Redshift di Panduan Pengembang](#) Database Amazon Redshift.

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Kueri dan beban untuk menampilkan daftar kueri untuk akun Anda.
3. Pilih kueri yang sedang berjalan yang ingin Anda akhiri dalam daftar, lalu pilih Hentikan kueri.

Memantau kinerja klaster Amazon Redshift

Amazon Redshift menyediakan metrik kinerja dan data sehingga Anda dapat melacak kesehatan dan kinerja cluster dan database Anda. Di bagian ini, kami membahas jenis data yang dapat Anda gunakan di Amazon Redshift, khususnya di konsol Amazon Redshift.

Data kinerja yang dapat Anda gunakan di konsol Amazon Redshift terbagi dalam dua kategori:

- CloudWatch Metrik Amazon — CloudWatch Metrik Amazon membantu Anda memantau aspek fisik klaster Anda, seperti pemanfaatan CPU, latensi, dan throughput. Data metrik ditampilkan langsung di konsol Amazon Redshift. Anda juga dapat melihatnya di CloudWatch konsol. Atau, Anda dapat menggunakannya dengan cara lain Anda bekerja dengan metrik, seperti dengan AWS CLI atau salah satu metrik. AWS SDKs
- Kueri/Muat data kinerja - Data kinerja membantu Anda memantau aktivitas dan kinerja database. Data ini digabungkan di konsol Amazon Redshift untuk membantu Anda dengan mudah mengkorelasikan apa yang Anda lihat CloudWatch dalam metrik dengan kueri database tertentu dan peristiwa pemuatan. Anda juga dapat membuat kueri kinerja kustom Anda sendiri dan

menjalankannya langsung di database. Data kinerja kueri dan pemuatan hanya ditampilkan di konsol Amazon Redshift. Itu tidak dipublikasikan sebagai CloudWatch metrik.

Data kinerja diintegrasikan ke dalam konsol Amazon Redshift, menghasilkan pengalaman yang lebih kaya dengan cara-cara berikut:

- Data kinerja yang terkait dengan klaster ditampilkan secara kontekstual saat Anda melihat klaster, di mana Anda mungkin memerlukannya untuk membuat keputusan tentang klaster seperti mengubah ukuran.
- Beberapa metrik kinerja ditampilkan dalam unit yang diskalakan lebih tepat di konsol Amazon Redshift dibandingkan dengan CloudWatch. Misalnya `WriteThroughput`, ditampilkan dalam GB/s (dibandingkan dengan bytes/s in CloudWatch), yang merupakan unit yang lebih relevan untuk ruang penyimpanan tipikal node.
- Anda dapat dengan mudah menampilkan data kinerja untuk node cluster bersama-sama pada grafik yang sama. Dengan cara ini, Anda dapat dengan mudah memantau kinerja semua node dari sebuah cluster. Anda juga dapat melihat data kinerja untuk setiap node.

Amazon Redshift menyediakan data kinerja (baik CloudWatch metrik maupun kueri dan data muat) tanpa biaya tambahan. Data kinerja direkam setiap menit. Anda dapat mengakses nilai historis data kinerja di konsol Amazon Redshift. [Untuk informasi mendetail tentang penggunaan CloudWatch untuk mengakses data kinerja Amazon Redshift yang diekspos sebagai CloudWatch metrik, lihat Apa itu? CloudWatch](#) di Panduan CloudWatch Pengguna Amazon.

Data kinerja di Amazon Redshift

Menggunakan CloudWatch metrik untuk Amazon Redshift, Anda bisa mendapatkan informasi tentang kesehatan dan kinerja klaster Anda dan melihat informasi di tingkat node. Saat bekerja dengan metrik ini, perlu diingat bahwa setiap metrik memiliki satu atau lebih dimensi yang terkait dengannya. Dimensi ini memberi tahu Anda metrik apa yang berlaku, yaitu ruang lingkup metrik. Amazon Redshift memiliki dua dimensi berikut:

- Metrik yang memiliki `NodeID` dimensi adalah metrik yang menyediakan data kinerja untuk node cluster. Kumpulan metrik ini mencakup node pemimpin dan komputasi. Contoh metrik ini meliputi `CPUUtilization`, `ReadIOPS`, `WriteIOPS`.
- Metrik yang hanya memiliki `ClusterIdentifier` dimensi adalah metrik yang menyediakan data kinerja untuk cluster. Contoh metrik ini termasuk `HealthStatus` dan `MaintenanceMode`.

Note

Dalam beberapa kasus metrik, metrik khusus cluster mewakili agregasi perilaku simpul. Dalam kasus ini, berhati-hatilah dalam interpretasi nilai metrik karena perilaku node pemimpin digabungkan dengan node komputasi.

Untuk informasi umum tentang CloudWatch metrik dan dimensi, lihat [CloudWatch konsep](#) di Panduan CloudWatch Pengguna Amazon.

Untuk deskripsi lebih lanjut tentang CloudWatch metrik untuk Amazon Redshift, lihat bagian berikut.

Topik

- [Metrik Pergeseran Merah Amazon](#)
- [Dimensi untuk metrik Amazon Redshift](#)
- [Kueri Amazon Redshift dan memuat data kinerja](#)

Metrik Pergeseran Merah Amazon

Namespace `AWS/Redshift` mencakup metrik berikut. Kecuali dinyatakan lain, metrik dikumpulkan pada interval 1 menit.

Metrik	Deskripsi
<code>CommitQueueLength</code>	Jumlah transaksi yang menunggu untuk dilakukan pada titik waktu tertentu. Unit: Hitungan Dimensi: <code>ClusterIdentifier</code>
<code>ConcurrencyScalingActiveClusters</code>	Jumlah cluster penskalaan konkurensi yang secara aktif memproses kueri pada waktu tertentu. Unit: Hitungan Dimensi: <code>ClusterIdentifier</code>

Metrik	Deskripsi
ConcurrencyScaling Seconds	Jumlah detik yang digunakan oleh cluster penskalaan konkurensi yang memiliki aktivitas pemrosesan kueri aktif. Unit: Hitungan Dimensi: <code>ClusterIdentifier</code>
CPUUtilization	Persentase penggunaan CPU. Untuk cluster, metrik ini mewakili agregasi dari semua node (pemimpin dan komputasi) nilai pemanfaatan CPU. Unit: Persen Dimensions: <code>ClusterIdentifier</code> , <code>NodeID</code> Dimensi: <code>ClusterIdentifier</code>
DatabaseConnections	Jumlah koneksi database ke sebuah cluster. Unit: Hitungan Dimensi: <code>ClusterIdentifier</code>

Metrik	Deskripsi
HealthStatus	Menunjukkan kesehatan cluster. Setiap menit cluster terhubung ke database dan melakukan query sederhana. Jika mampu melakukan operasi ini dengan sukses, cluster dianggap sehat. Kalau tidak, cluster tidak sehat. Status yang tidak sehat dapat terjadi ketika database cluster berada di bawah beban yang sangat berat atau jika ada masalah konfigurasi dengan database di cluster.  Note Di Amazon CloudWatch, metrik ini dilaporkan sebagai 1 atau 0 sedangkan di konsol Amazon Redshift, metrik ini ditampilkan dengan kata-kata HEALTHY atau UNHEALTHY untuk kenyamanan. Saat metrik ini ditampilkan di konsol Amazon Redshift, rata-rata pengambilan sampel diabaikan dan hanya HEALTHY atau ditampilkan. UNHEALTHY Di Amazon CloudWatch, nilai yang berbeda dari 1 dan 0 mungkin terjadi karena masalah pengambilan sampel. Setiap nilai di bawah 1 HealthStatus untuk dilaporkan sebagai 0 (UNHEALTHY). Unit: Hitung (1/0) (HEALTHY/UNHEALTHY di konsol Amazon Redshift) Dimensi: ClusterIdentifier

Metrik	Deskripsi
MaintenanceMode	Menunjukkan apakah cluster dalam mode pemeliharaan.  Note Di Amazon CloudWatch, metrik ini dilaporkan sebagai 1 atau 0 sedangkan di konsol Amazon Redshift, metrik ini ditampilkan dengan kata-kata ON atau OFF untuk kenyamanan. Saat metrik ini ditampilkan di konsol Amazon Redshift, rata-rata pengambilan sampel diabaikan dan hanya ON atau ditampilkan. OFF Di Amazon CloudWatch, nilai yang berbeda dari 1 dan 0 mungkin terjadi karena masalah pengambilan sampel. Setiap nilai yang lebih besar dari 0 MaintenanceMode untuk dilaporkan sebagai 1 (ON). Unit: Hitung (1/0) (ON/OFF di konsol Amazon Redshift). Dimensi: ClusterIdentifier
MaxConfiguredConcurrencyScalingClusters	Jumlah maksimum cluster penskalaan konkurensi yang dikonfigurasi dari grup parameter. Untuk informasi selengkapnya, lihat Grup parameter Amazon Redshift. Unit: Count (Jumlah) Dimensi: ClusterIdentifier
NetworkReceiveThroughput	Tingkat di mana node atau cluster menerima data. Unit: Bytes/Second (MB/s di konsol Amazon Redshift) Dimensions: ClusterIdentifier , NodeID Dimensi: ClusterIdentifier

Metrik	Deskripsi
NetworkTransmitThroughput	Tingkat di mana node atau cluster menulis data. Unit: Bytes/Second (MB/s di konsol Amazon Redshift) Dimensions: ClusterIdentifier , NodeID Dimensi: ClusterIdentifier
PercentageDiskSpaceUsed	Persentase ruang disk yang digunakan. Unit: Persen Dimensi: ClusterIdentifier Dimensions: ClusterIdentifier , NodeID
QueriesCompletedPerSecond	Jumlah rata-rata kueri yang diselesaikan per detik. Dilaporkan dalam interval 5 menit. Metrik ini tidak didukung pada kluster simpul tunggal. Unit: Jumlah/Detik Dimensions: ClusterIdentifier , latency Dimensions: ClusterIdentifier , wlmid
QueryDuration	Jumlah rata-rata waktu untuk menyelesaikan kueri. Dilaporkan dalam interval 5 menit. Metrik ini tidak didukung pada kluster simpul tunggal. Unit: Mikrodetik Dimensi:ClusterIdentifier ,NodeID,latency Dimensions: ClusterIdentifier , latency Dimensi:ClusterIdentifier ,NodeID,wlmid

Metrik	Deskripsi
QueryRuntimeBreakdown	Total waktu kueri yang dihabiskan untuk dijalankan berdasarkan tahap kueri. Dilaporkan dalam interval 5 menit. Unit: Milidetik Dimensi: ClusterIdentifier, NodeId, panggung Dimensi: ClusterIdentifier, panggung
ReadIOPS	Jumlah rata-rata operasi pembacaan disk per detik. Unit: Jumlah/Detik Dimensions: ClusterIdentifier , NodeID Dimensi: ClusterIdentifier
ReadLatency	Jumlah rata-rata waktu yang dibutuhkan untuk operasi I/O membaca disk. Unit: detik Dimensions: ClusterIdentifier , NodeID Dimensi: ClusterIdentifier
ReadThroughput	Jumlah byte rata-rata yang dibaca dari disk per detik. Unit: Byte (Gb/s di konsol Amazon Redshift) Dimensions: ClusterIdentifier , NodeID Dimensi: ClusterIdentifier
RedshiftManagedStorageTotalCapacity	Total kapasitas penyimpanan terkelola. Unit: Megabyte Dimensi: ClusterIdentifier

Metrik	Deskripsi
TotalTableCount	Jumlah tabel pengguna terbuka pada titik waktu tertentu. Total ini tidak termasuk tabel Amazon Redshift Spectrum. Unit: Hitungan Dimensi: ClusterIdentifier
WLMQueueLength	Jumlah kueri yang menunggu untuk memasuki antrian manajemen beban kerja (WLM). Unit: Hitungan Dimensions: ClusterIdentifier , service class Dimensions: ClusterIdentifier , QueueName
WLMQueueWaitTime	Total waktu kueri yang dihabiskan menunggu dalam antrian manajemen beban kerja (WLM). Dilaporkan dalam interval 5 menit. Satuan: Milidetik. Dimensions: ClusterIdentifier , QueryPriority Dimensions: ClusterIdentifier , wlmid Dimensions: ClusterIdentifier , QueueName
WLMQueriesCompletedPerSecond	Jumlah rata-rata kueri yang diselesaikan per detik untuk antrian manajemen beban kerja (WLM). Dilaporkan dalam interval 5 menit. Metrik ini tidak didukung pada kluster simpul tunggal. Unit: Jumlah/Detik Dimensions: ClusterIdentifier , wlmid Dimensions: ClusterIdentifier , QueueName

Metrik	Deskripsi
WLMQueryDuration	Rata-rata lamanya waktu untuk menyelesaikan kueri untuk antrian manajemen beban kerja (WLM). Dilaporkan dalam interval 5 menit. Metrik ini tidak didukung pada kluster simpul tunggal. Unit: Mikrodetik Dimensions: <code>ClusterIdentifier</code> , <code>wlmid</code> Dimensions: <code>ClusterIdentifier</code> , <code>QueueName</code>
WLMRunningQueries	Jumlah kueri yang berjalan dari cluster utama dan cluster penskalaan konkurensi per antrian WLM. Unit: Hitungan Dimensions: <code>ClusterIdentifier</code> , <code>wlmid</code> Dimensions: <code>ClusterIdentifier</code> , <code>QueueName</code>
WriteIOPS	Jumlah rata-rata operasi tulis per detik. Unit: Jumlah/Detik Dimensions: <code>ClusterIdentifier</code> , <code>NodeID</code> Dimensi: <code>ClusterIdentifier</code>
WriteLatency	Jumlah rata-rata waktu yang dibutuhkan untuk operasi I/O tulis disk. Unit: detik Dimensions: <code>ClusterIdentifier</code> , <code>NodeID</code> Dimensi: <code>ClusterIdentifier</code>

Metrik	Deskripsi
WriteThroughput	Jumlah byte rata-rata yang ditulis dari disk per detik. Unit: Byte (Gb/s di konsol Amazon Redshift) Dimensions: <code>ClusterIdentifier</code> , <code>NodeID</code> Dimensi: <code>ClusterIdentifier</code>
SchemaQuota	Kuota yang dikonfigurasi untuk skema. Unit: Megabyte Dimensi: <code>ClusterIdentifier</code> , <code>Database</code>, <code>Schema</code> Berkala/Dorong: <code>Periodic</code> Frekuensi: <code>5 minutes</code> Kriteria berhenti: Skema turun atau kuota dihapus
NumExceededSchemaQuotas	Jumlah skema dengan kuota terlampaui. Unit: Hitungan Dimensi: <code>ClusterIdentifier</code> Berkala/Dorong: <code>Periodic</code> Frekuensi: <code>5 minutes</code> Kriteria berhenti: N/A

Metrik	Deskripsi
StorageUsed	Disk atau ruang penyimpanan yang digunakan oleh skema. Unit: Megabyte Dimensi:ClusterIdentifier ,Database, Schema Berkala/Dorong: Periodic Frekuensi: 5 minutes Kriteria berhenti: Skema turun atau kuota dihapus
PercentageQuotaUsed	Persentase disk atau ruang penyimpanan yang digunakan relatif terhadap kuota skema yang dikonfigurasi. Unit: Persen Dimensi:ClusterIdentifier ,Database, Schema Berkala/Dorong: Periodic Frekuensi: 5 minutes Kriteria berhenti: Skema turun atau kuota dihapus

Metrik	Deskripsi
UsageLimitAvailable	Tergantung pada FeatureType, UsageLimitAvailable mengembalikan yang berikut: • Jika yaCONCURRENCY_SCALING , UsageLimitAvailable mengembalikan jumlah total waktu yang dapat digunakan dengan penskalaan konkurensi dalam kenaikan 1 menit. FeatureType Waktu ini tidak dihitung sebagai waktu yang tersedia CONCURRENCY_SCALING_AUTO_TASK dan sebaliknya. • Jika yaCONCURRENCY_SCALING_AUTO_TASK , UsageLimitAvailable mengembalikan jumlah total waktu yang dapat digunakan oleh penskalaan konkurensi untuk tugas pengoptimalan otomatis. FeatureType Waktu ini tidak dihitung sebagai waktu yang tersedia untuk CONCURRENCY_SCALING dan sebaliknya. • Jika yaEXTRA_COMPUTE_FOR_AUTOMATIC_OPTIMIZATION , UsageLimitAvailable mengembalikan jumlah total waktu yang tersedia untuk digunakan untuk tugas pengoptimalan otomatis. FeatureType • Jika FeatureType yaCROSS_REGION_DATASHARING , UsageLimitAvailable mengembalikan jumlah total data yang dapat dipindai dalam peningkatan 1-TB. • Jika FeatureType yaSPECTRUM, UsageLimitAvailable mengembalikan jumlah total data yang dapat dipindai dalam peningkatan 1-TB. Unit: Menit atau TBs Dimensi:ClusterIdentifier ,FeatureType , UsageLimitId

Metrik	Deskripsi
UsageLimitConsumed	Tergantung pada FeatureType, UsageLimitConsumed mengembalikan yang berikut: • Jika yaCONCURRENCY_SCALING , UsageLimitConsumed mengembalikan jumlah total waktu yang digunakan oleh penskalaan konkurensi dalam kenaikan 1 menit. FeatureType Waktu ini tidak dihitung sebagai waktu yang digunakan untuk CONCURRENCY_SCALING_AUTO_TASK dan sebaliknya. • Jika yaCONCURRENCY_SCALING_AUTO_TASK , UsageLimitConsumed mengembalikan jumlah total waktu yang digunakan oleh penskalaan konkurensi untuk tugas pengoptimalan otomatis. FeatureType Waktu ini tidak dihitung sebagai waktu yang digunakan untuk CONCURRENCY_SCALING dan sebaliknya. • Jika yaEXTRA_COMPUTE_FOR_AUTOMATIC_OPTIMIZATION , UsageLimitConsumed mengembalikan jumlah total waktu yang digunakan untuk tugas pengoptimalan otomatis. FeatureType • Jika yaCROSS_REGION_DATASHARING , UsageLimitConsumed mengembalikan jumlah total data yang dipindai dalam peningkatan 1-TB. FeatureType • Jika yaSPECTRUM, UsageLimitConsumed mengembalikan jumlah total data yang dipindai dalam peningkatan 1-TB. FeatureType Unit: Menit atau TBs Dimensi:ClusterIdentifier ,FeatureType , UsageLimitId

Dimensi untuk metrik Amazon Redshift

Data Amazon Redshift dapat difilter sepanjang salah satu dimensi dalam tabel berikut.

Dimensi	Deskripsi
latency	Kemungkinan nilainya adalah sebagai berikut: • pendek - di bawah 10 detik • sedang — antara 10 detik dan 10 menit • lama — lebih dari 10 menit
NodeID	Filter meminta data yang spesifik untuk node cluster. NodeID adalah “Leader”, “Shared”, atau “Compute-N” di mana N adalah 0, 1,... untuk jumlah node dalam cluster. “Shared” berarti bahwa cluster hanya memiliki satu node, yaitu node pemimpin dan node komputasi digabungkan. Metrik dilaporkan untuk node pemimpin dan node komputasi hanya untuk CPUUtilization, NetworkTransmitThroughput, dan ReadIOPS. Metrik lain yang menggunakan NodeId dimensi dilaporkan hanya untuk node komputasi.
ClusterIdentifier	Filter meminta data yang khusus untuk cluster. Metrik yang khusus untuk cluster meliputi HealthStatus, MaintenanceMode, dan DatabaseConnections. Metrik umum untuk dimensi ini (misalnya, ReadIOPS) yang juga merupakan metrik node mewakili agregat data metrik node. Berhati-hatilah dalam menafsirkan metrik ini karena mereka menggabungkan perilaku pemimpin dan node komputasi.
service class	Identifier untuk kelas WLM layanan.
stage	Tahapan eksekusi untuk kueri. Kemungkinan nilainya adalah sebagai berikut: • QueryPlanning: Waktu yang dihabiskan mengurai dan mengoptimalkan pernyataan SQL. • QueryWaiting: Waktu yang dihabiskan menunggu di antrian WLM. • QueryExecutingRead: Waktu yang dihabiskan untuk mengeksekusi kueri baca.

Dimensi	Deskripsi
	• QueryExecutingInsert: Waktu yang dihabiskan untuk mengeksekusi kueri sisipan. • QueryExecutingDelete: Waktu yang dihabiskan untuk mengeksekusi kueri hapus. • QueryExecutingUpdate: Waktu yang dihabiskan untuk mengeksekusi kueri pembaruan. • QueryExecutingCtas: Waktu yang dihabiskan untuk mengeksekusi membuat tabel sebagai kueri. • QueryExecutingUnload: Waktu yang dihabiskan untuk mengeksekusi kueri bongkar muat. • QueryExecutingCopy: Waktu yang dihabiskan untuk mengeksekusi kueri salinan. • QueryCommit: Waktu yang dihabiskan untuk berkomitmen.
wlmid	Pengenal untuk antrian manajemen beban kerja.
QueryPriority	Prioritas kueri. Peluang nilai adalah CRITICAL, HIGHEST, HIGH, NORMAL, LOW, dan LOWEST.
QueueName	Nama antrian manajemen beban kerja.
FeatureType	Fitur yang dibatasi oleh batas penggunaan. Nilai yang mungkin adalah CONCURRENCY_SCALING , CROSS_REGION_DATAS HARING , dan SPECTRUM.
UsageLimitId	Pengidentifikasi untuk batas penggunaan.

Kueri Amazon Redshift dan memuat data kinerja

Selain CloudWatch metrik, Amazon Redshift menyediakan kueri dan memuat data kinerja. Data kinerja kueri dan beban dapat digunakan untuk membantu Anda memahami hubungan antara kinerja database dan metrik klaster. Misalnya, jika Anda melihat bahwa CPU cluster melonjak, Anda dapat menemukan lonjakan pada grafik CPU cluster dan melihat kueri yang sedang berjalan pada saat itu. Sebaliknya, jika Anda meninjau kueri tertentu, data metrik (seperti CPU) ditampilkan dalam konteks sehingga Anda dapat memahami dampak kueri pada metrik klaster.

Data kinerja kueri dan pemuatan tidak dipublikasikan sebagai CloudWatch metrik dan hanya dapat dilihat di konsol Amazon Redshift. Data kinerja kueri dan pemuatan dihasilkan dari kueri dengan tabel sistem database Anda (untuk informasi selengkapnya, lihat [Referensi tabel sistem](#) di Panduan Pengembang Amazon Redshift). Anda juga dapat membuat kueri kinerja basis data kustom Anda sendiri, tetapi kami sarankan memulai dengan kueri dan memuat data kinerja yang disajikan di konsol. Untuk informasi selengkapnya tentang mengukur dan memantau performa database Anda sendiri, lihat [Mengelola performa](#) di Panduan Pengembang Amazon Redshift.

Tabel berikut menjelaskan berbagai aspek kueri dan memuat data yang dapat Anda akses di konsol Amazon Redshift.

Kueri/Muat data	Deskripsi
Ringkasan kueri	Daftar kueri dalam periode waktu tertentu. Daftar dapat diurutkan berdasarkan nilai seperti ID kueri, runtime kueri, dan status. Lihat data ini di tab Pemantauan kueri pada halaman detail klaster.
Detail kueri	Memberikan detail tentang kueri tertentu termasuk: • Properti kueri seperti ID kueri, tipe, klaster tempat kueri dijalankan, dan runtime. • Detail seperti status kueri dan jumlah kesalahan. • Pernyataan SQL yang dijalankan. • Rencana penjelasan jika tersedia. • Data kinerja cluster selama eksekusi kueri (untuk informasi selengkapnya, lihat Melihat data riwayat kueri).
Muat ringkasan	Daftar semua beban dalam periode waktu tertentu. Daftar dapat diurutkan berdasarkan nilai seperti ID kueri, runtime kueri, dan status. Lihat data ini di tab Pemantauan kueri pada halaman detail klaster.
Memuat detail	Memberikan rincian tentang operasi beban tertentu termasuk: • Memuat properti seperti ID kueri, tipe, klaster tempat kueri dijalankan, dan runtime. • Detail seperti status beban dan jumlah kesalahan. • Pernyataan SQL yang dijalankan.

Kueri/Muat data	Deskripsi
	- Daftar file yang dimuat. - Data kinerja cluster selama operasi beban (untuk informasi lebih lanjut, lihat Melihat data riwayat kueri).

Melihat data kinerja

Di bagian ini, Anda dapat menemukan cara melihat data kinerja di konsol Amazon Redshift, yang mencakup informasi tentang kinerja klaster dan kueri. Selain itu, Anda dapat membuat alarm pada metrik cluster langsung dari konsol Amazon Redshift.

Saat melihat data performa di konsol Amazon Redshift, Anda melihatnya berdasarkan klaster. Grafik data kinerja untuk klaster dirancang untuk memberi Anda akses ke data untuk menjawab pertanyaan kinerja Anda yang paling umum. Untuk beberapa data kinerja (lihat [Data kinerja di Amazon Redshift](#)), Anda juga dapat menggunakan CloudWatch untuk menyesuaikan grafik metrik lebih lanjut. Misalnya, Anda dapat memilih waktu yang lebih lama atau menggabungkan metrik di seluruh cluster. Untuk informasi selengkapnya tentang bekerja dengan CloudWatch konsol, lihat [Metrik kinerja di konsol CloudWatch](#).

Tonton video berikut untuk mempelajari cara memantau, mengisolasi, dan mengoptimalkan kueri menggunakan fitur pemantauan kueri di konsol Amazon Redshift: [Pemantauan Kueri dengan Amazon Redshift](#).

Topik

- [Melihat data kinerja cluster](#)
- [Melihat data riwayat kueri](#)
- [Melihat data kinerja database](#)
- [Melihat data konkurensi beban kerja dan penskalaan konkurensi](#)
- [Melihat data optimasi otomatis](#)
- [Melihat kueri dan beban](#)
- [Melihat dan menganalisis detail kueri](#)
- [Melihat kinerja cluster saat kueri berjalan](#)
- [Melihat metrik cluster selama operasi pemuatan](#)
- [Melihat bagan rincian beban kerja cluster](#)

Melihat data kinerja cluster

Dengan menggunakan metrik klaster di Amazon Redshift, Anda dapat melakukan tugas kinerja umum berikut:

- Tentukan apakah metrik klaster tidak normal pada rentang waktu tertentu dan, jika demikian, identifikasi kueri yang bertanggung jawab atas hit kinerja.
- Periksa apakah kueri historis atau saat ini memengaruhi kinerja klaster. Jika Anda mengidentifikasi kueri bermasalah, Anda dapat melihat detailnya termasuk kinerja klaster selama eksekusi kueri. Anda dapat menggunakan informasi ini dalam mendiagnosis mengapa kueri lambat dan apa yang dapat dilakukan untuk meningkatkannya.

Untuk melihat data kinerja

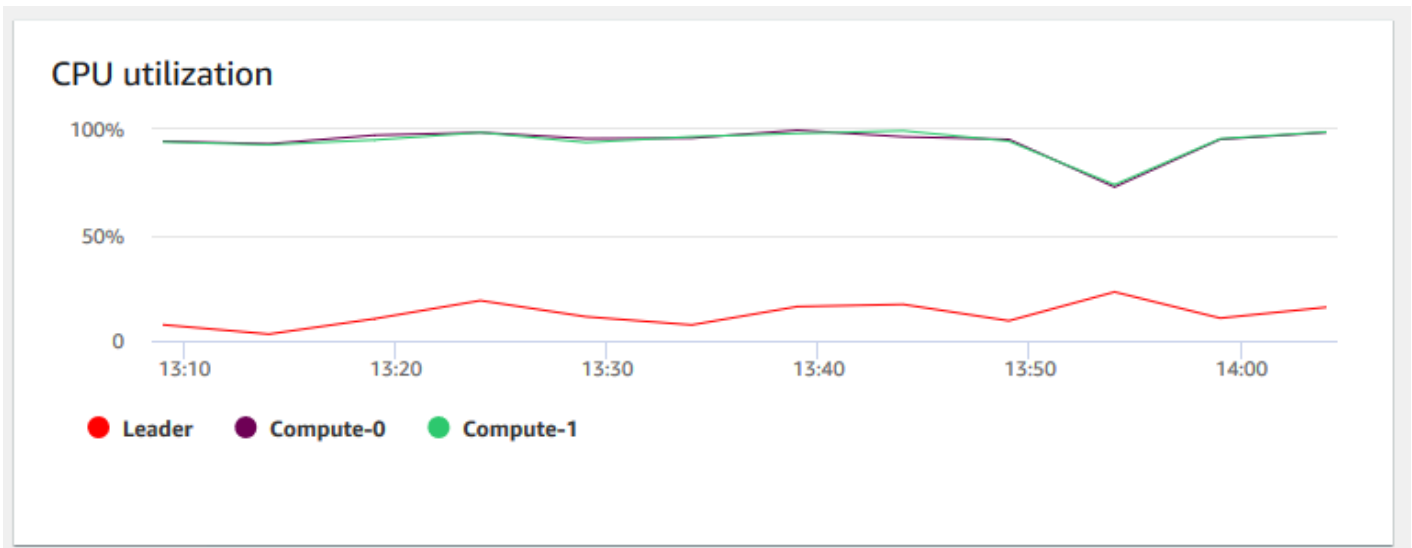
1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster, lalu pilih nama cluster dari daftar untuk membuka detailnya. Rincian cluster ditampilkan, yang dapat mencakup kinerja Cluster, pemantauan Kueri, Database, Datashares, Jadwal, Pemeliharaan, dan tab Properti.
3. Pilih tab Kinerja klaster untuk informasi kinerja termasuk yang berikut ini:
 - Pemanfaatan CPU
 - Persentase ruang disk yang digunakan
 - Koneksi database
 - Status kondisi
 - Durasi kueri
 - Throughput kueri
 - Aktivitas penskalaan konkurensi

Banyak lagi metrik yang tersedia. Untuk melihat metrik yang tersedia dan memilih yang ditampilkan, pilih ikon Preferensi.

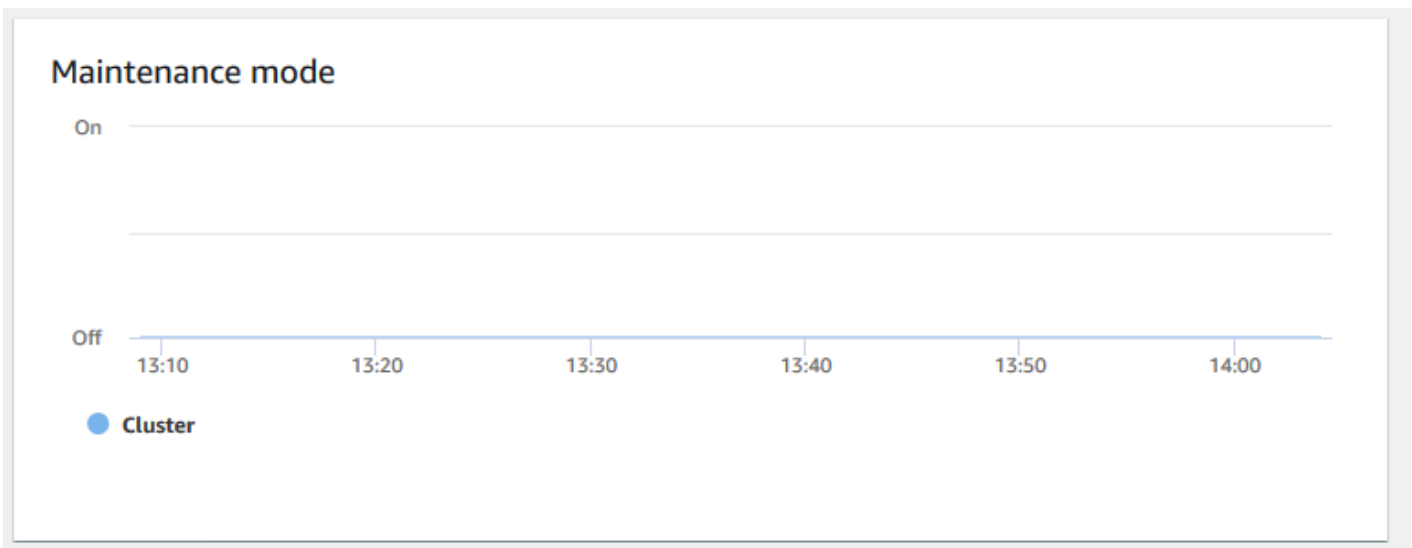
Grafik kinerja cluster

Contoh berikut menunjukkan beberapa grafik yang ditampilkan di konsol Amazon Redshift baru.

- **Pemanfaatan CPU** — Menunjukkan persentase pemanfaatan CPU untuk semua node (pemimpin dan komputasi). Untuk menemukan waktu ketika penggunaan kluster terendah sebelum menjadwalkan migrasi cluster atau operasi yang memakan sumber daya lainnya, pantau bagan ini untuk melihat pemanfaatan CPU per individu atau semua node.

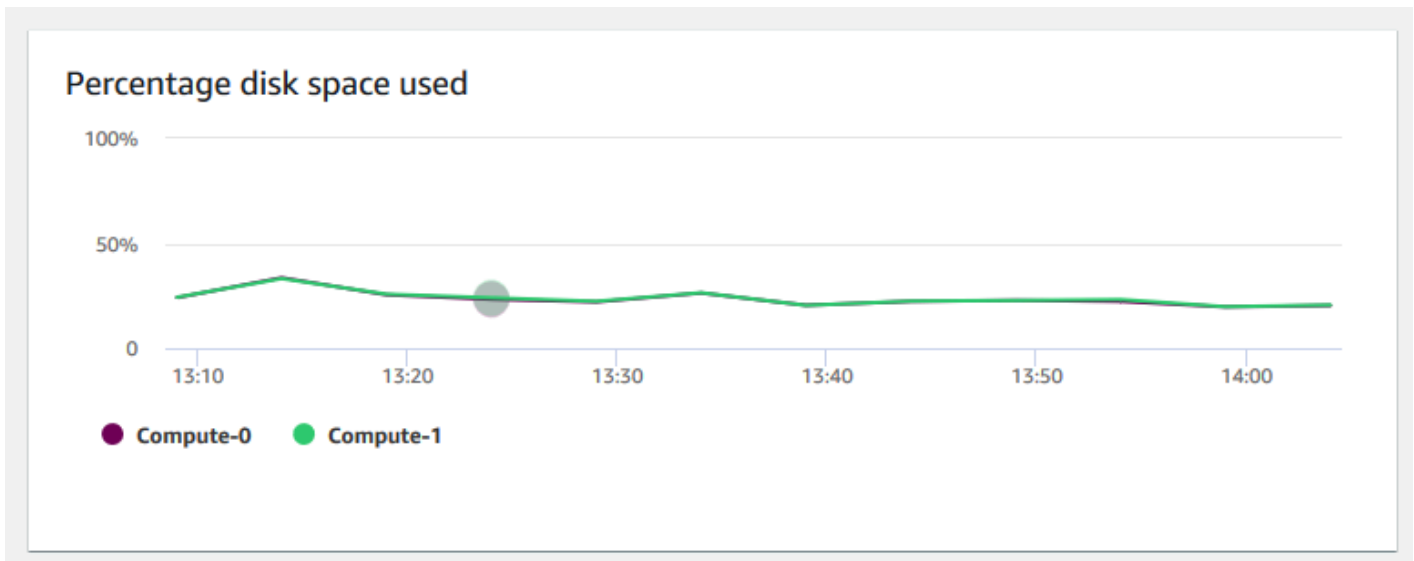


- **Mode pemeliharaan** - Menunjukkan apakah cluster berada dalam mode pemeliharaan pada waktu yang dipilih dengan menggunakan On dan Off indikator. Anda dapat melihat waktu ketika cluster sedang menjalani pemeliharaan. Anda kemudian dapat mengkorelasikan waktu ini dengan operasi yang dilakukan ke cluster untuk memperkirakan waktu henti di masa depan untuk peristiwa berulang.

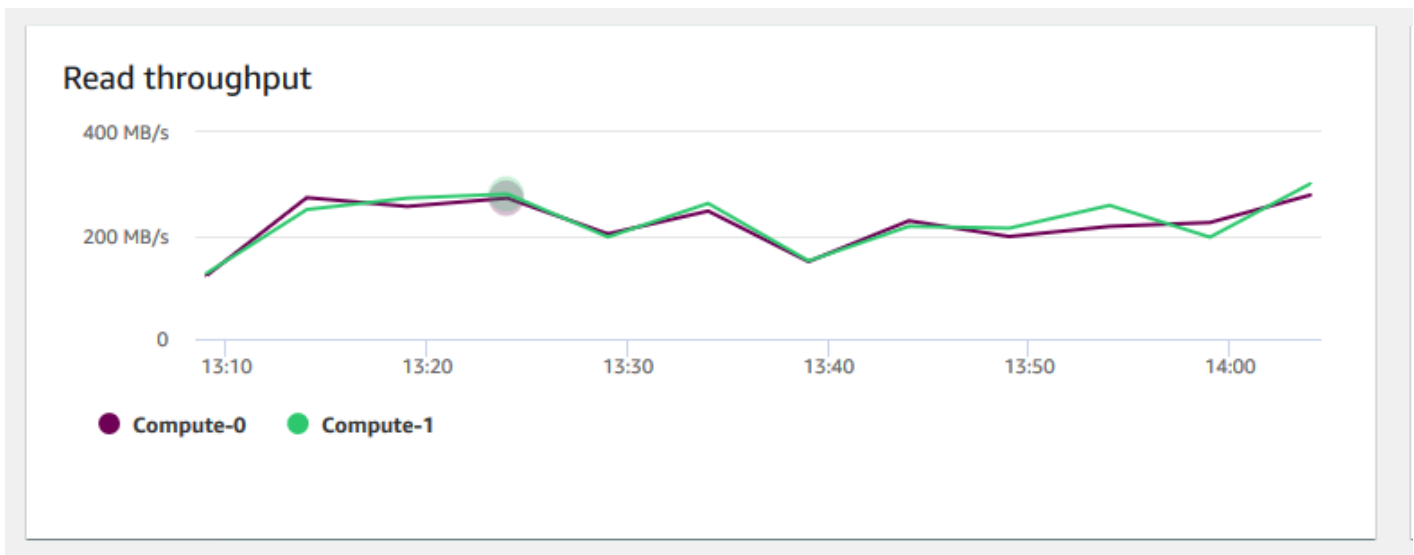


- **Persentase ruang disk yang digunakan** - Menunjukkan persentase penggunaan ruang disk per setiap node komputasi, dan bukan untuk cluster secara keseluruhan. Anda dapat menjelajahi bagan ini untuk memantau pemanfaatan disk. Operasi pemeliharaan seperti VACUUM dan COPY

menggunakan ruang penyimpanan sementara menengah untuk operasi penyortiran mereka, sehingga lonjakan penggunaan disk diharapkan.

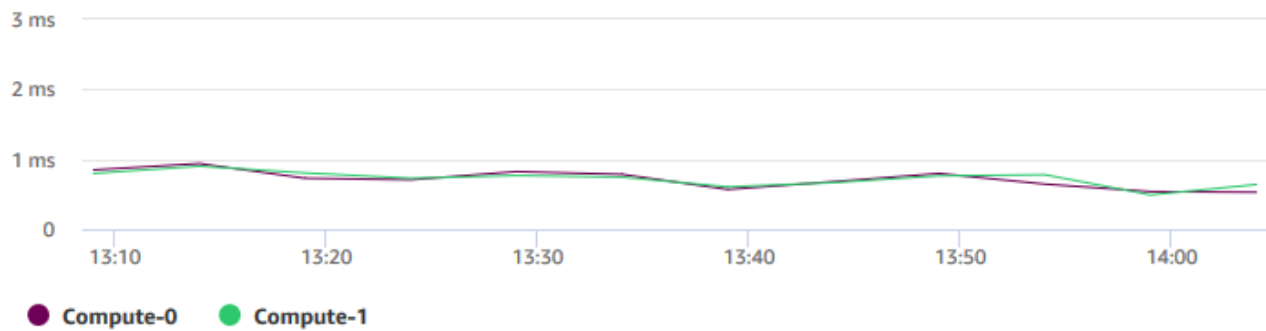


- **Baca throughput** — Menunjukkan jumlah rata-rata megabyte yang dibaca dari disk per detik. Anda dapat mengevaluasi bagan ini untuk memantau aspek fisik cluster yang sesuai. Throughput ini tidak termasuk lalu lintas jaringan antara instance di cluster dan volumenya.



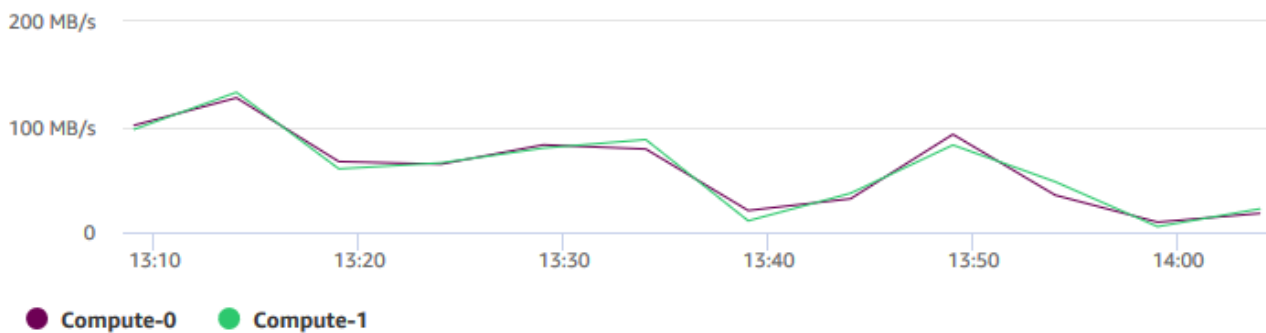
- **Baca latensi** - Menunjukkan jumlah rata-rata waktu yang dibutuhkan untuk I/O operasi pembacaan disk per milidetik. Anda dapat melihat waktu respons untuk mengembalikan data. Ketika latensi tinggi, itu berarti bahwa pengirim menghabiskan lebih banyak waktu idle (tidak mengirim paket baru), yang mengurangi seberapa cepat throughput tumbuh.

Read latency

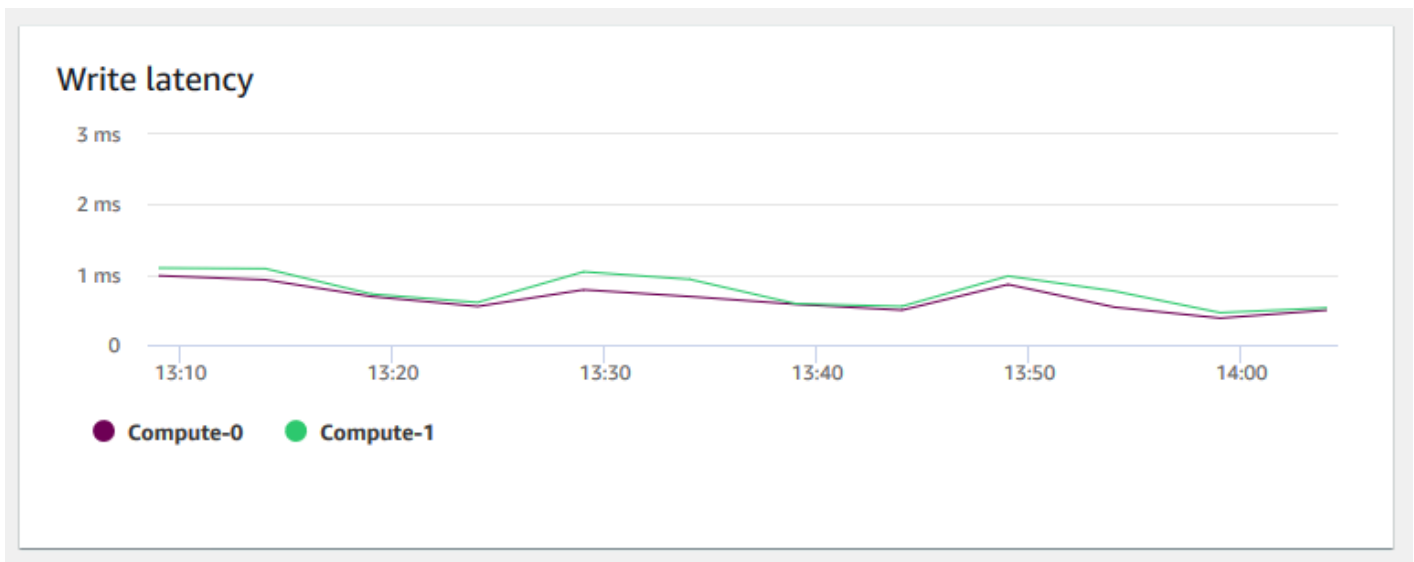


- **Tulis throughput** — Menunjukkan jumlah rata-rata megabyte yang ditulis ke disk per detik. Anda dapat mengevaluasi metrik ini untuk memantau aspek fisik cluster yang sesuai. Throughput ini tidak termasuk lalu lintas jaringan antara instance di cluster dan volumenya.

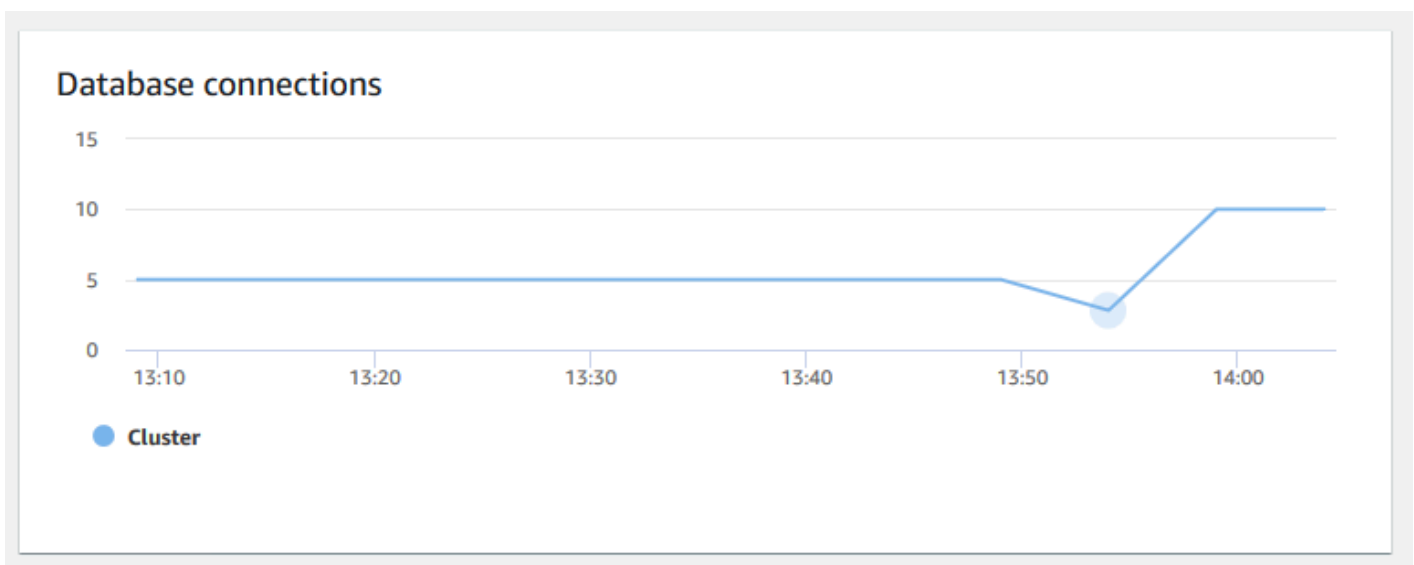
Write throughput



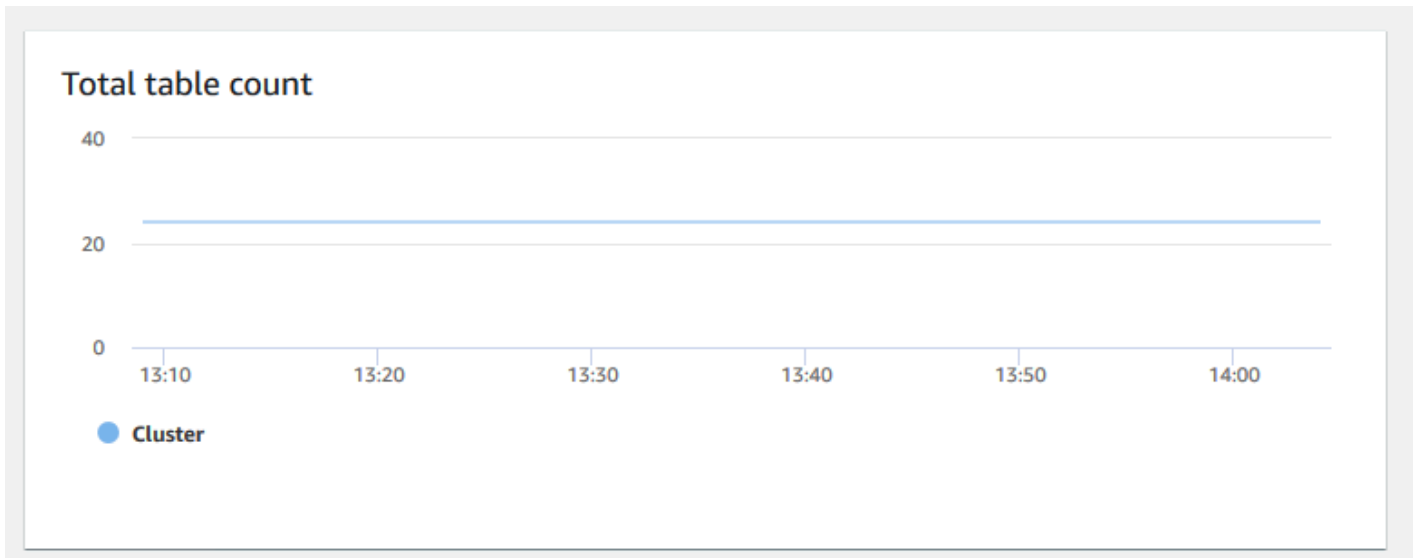
- **Latensi tulis** - Menunjukkan jumlah waktu rata-rata dalam milidetik yang diambil untuk operasi penulisan I/O disk. Anda dapat mengevaluasi waktu pengembalian pengakuan tulis. Ketika latensi tinggi, itu berarti bahwa pengirim menghabiskan lebih banyak waktu idle (tidak mengirim paket baru), yang mengurangi seberapa cepat throughput tumbuh.



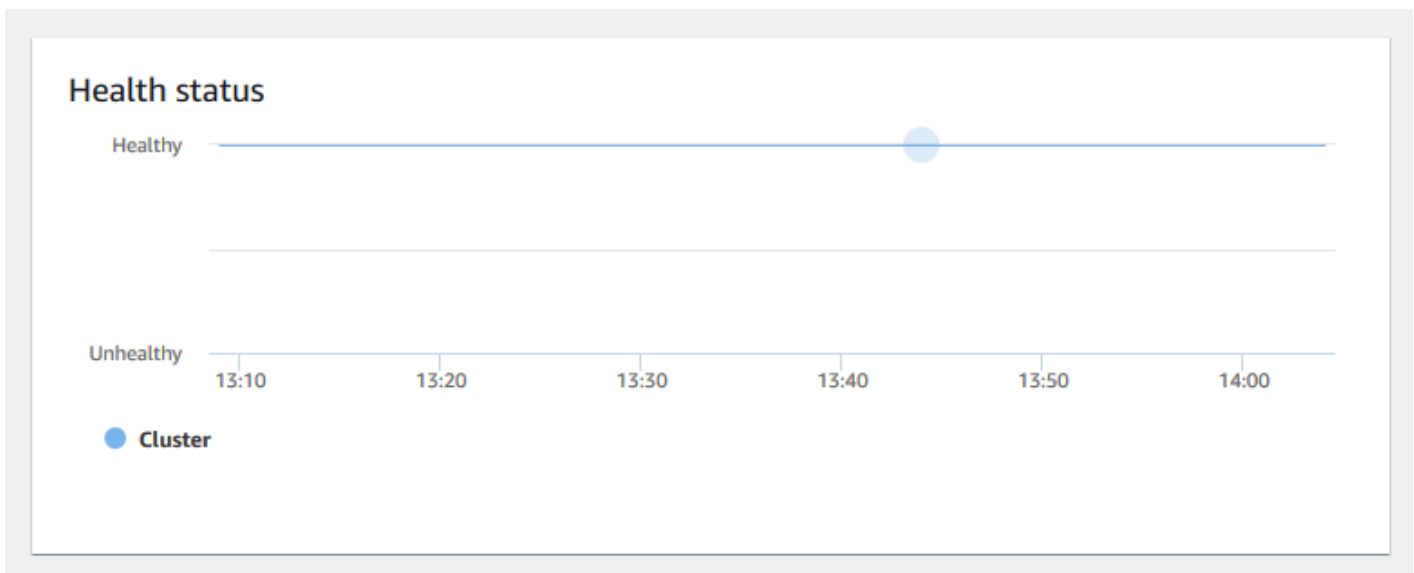
- Koneksi database - Menunjukkan jumlah koneksi database ke sebuah cluster. Anda dapat menggunakan bagan ini untuk melihat berapa banyak koneksi yang dibuat ke database dan menemukan waktu ketika penggunaan cluster terendah.



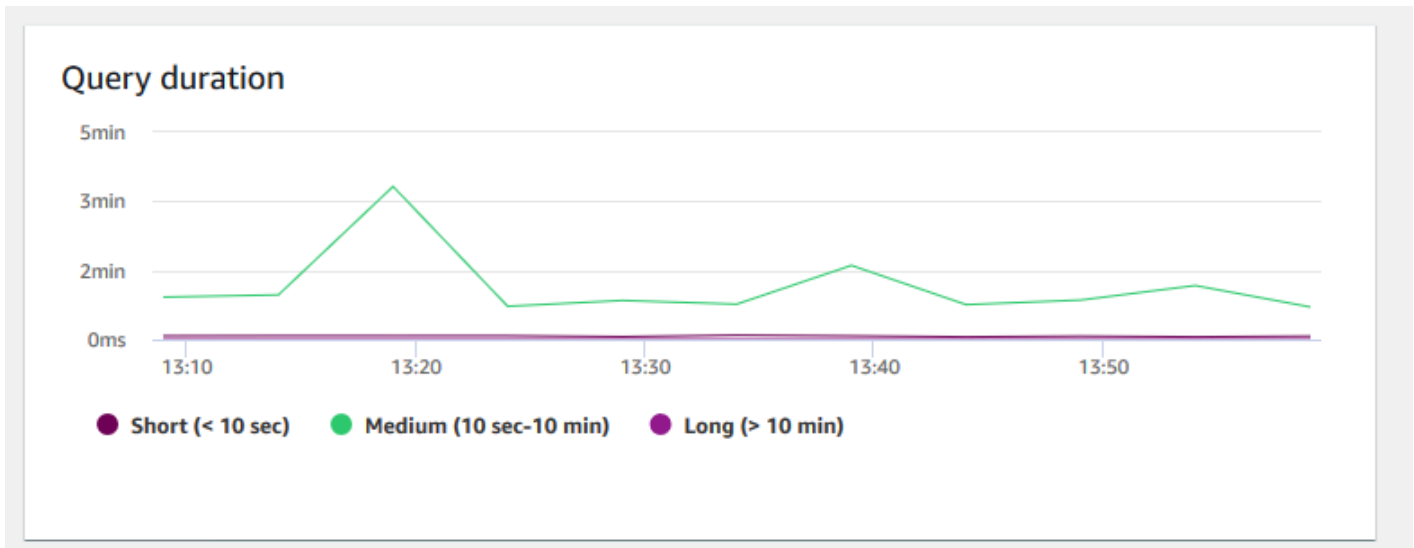
- Jumlah tabel total - Menunjukkan jumlah tabel pengguna yang terbuka pada titik waktu tertentu dalam sebuah cluster. Anda dapat memantau kinerja cluster saat jumlah tabel terbuka tinggi.



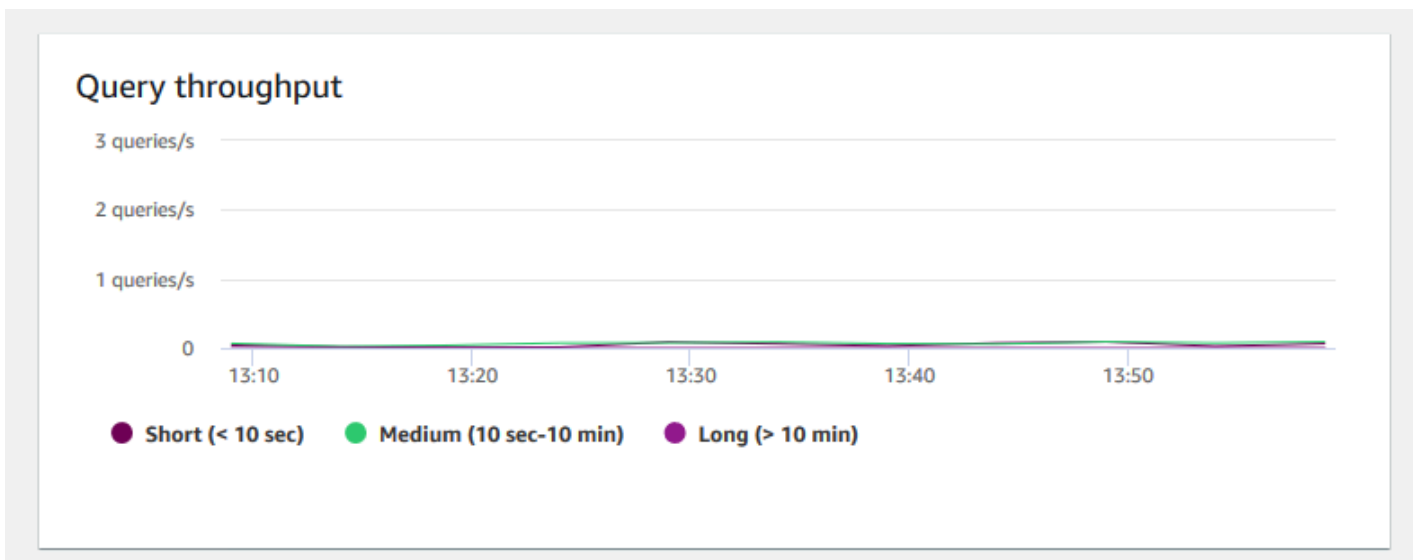
- **Status Kesehatan** — Menunjukkan kesehatan cluster sebagai **Healthy** atau **Unhealthy**. Jika cluster dapat terhubung ke database dan melakukan query sederhana berhasil, cluster dianggap sehat. Kalau tidak, cluster tidak sehat. Status yang tidak sehat dapat terjadi ketika database cluster berada di bawah beban yang sangat berat atau jika ada masalah konfigurasi dengan database di cluster.



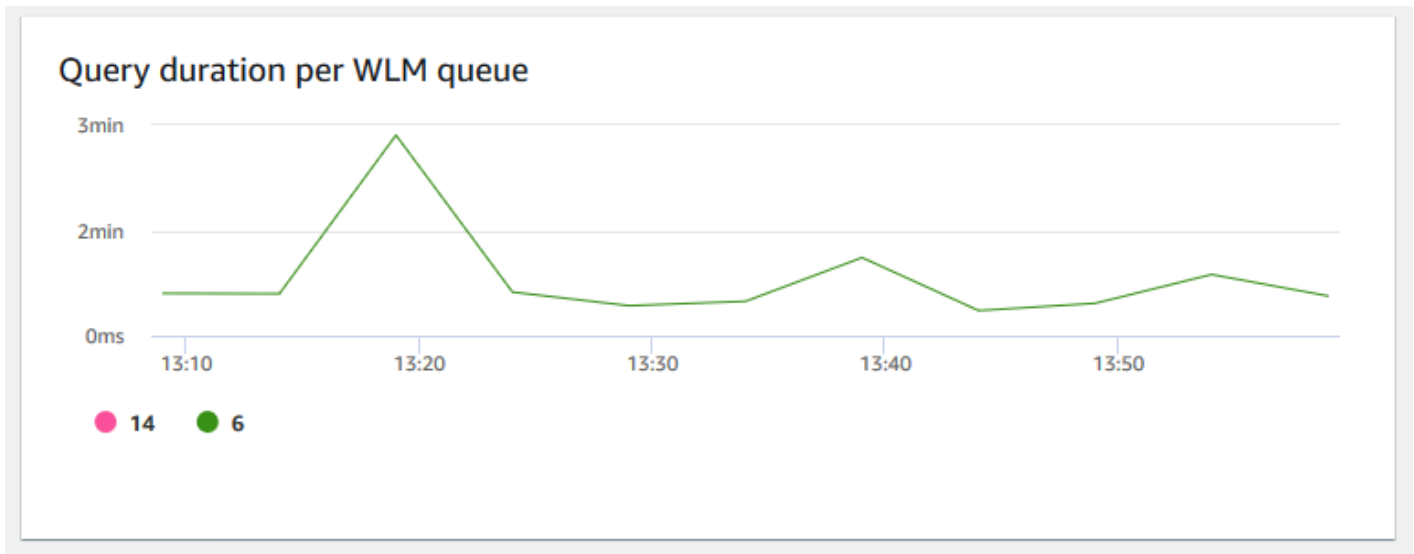
- **Durasi kueri** - Menunjukkan jumlah waktu rata-rata untuk menyelesaikan kueri dalam mikrodetik. Anda dapat membandingkan data pada bagan ini untuk mengukur I/O kinerja dalam cluster dan menyetel kueri yang paling memakan waktu jika perlu.



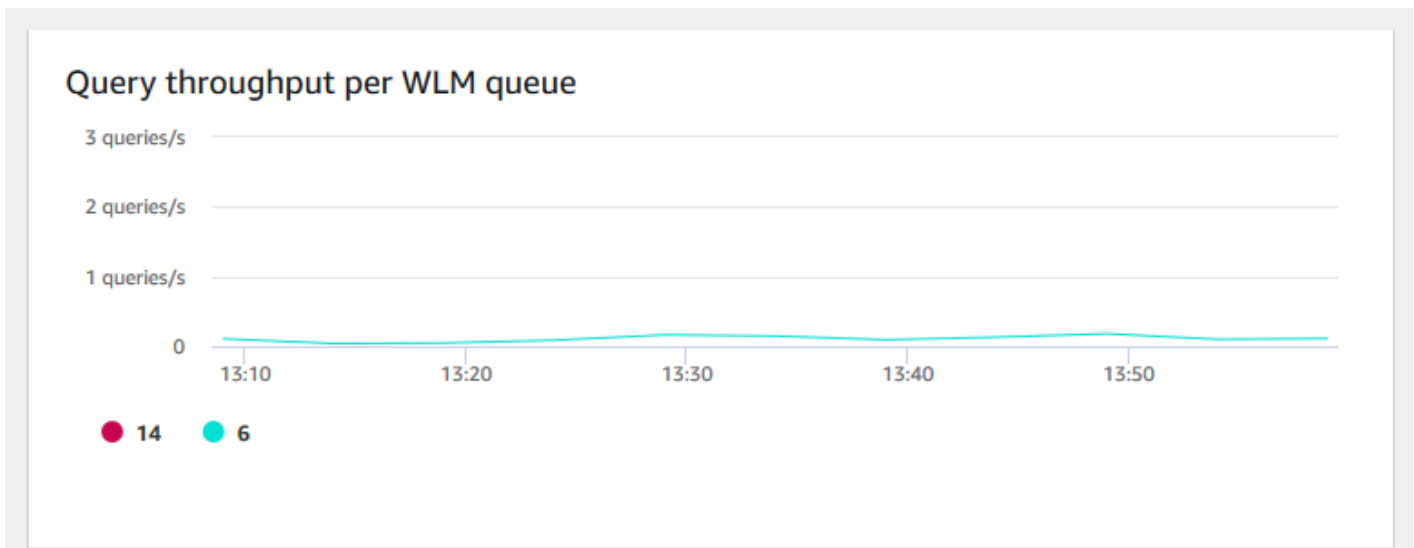
- Hasil kueri - Menunjukkan jumlah rata-rata kueri yang diselesaikan per detik. Anda dapat menganalisis data pada bagan ini untuk mengukur kinerja database dan mengkarakterisasi kemampuan sistem untuk mendukung beban kerja multipengguna secara seimbang.



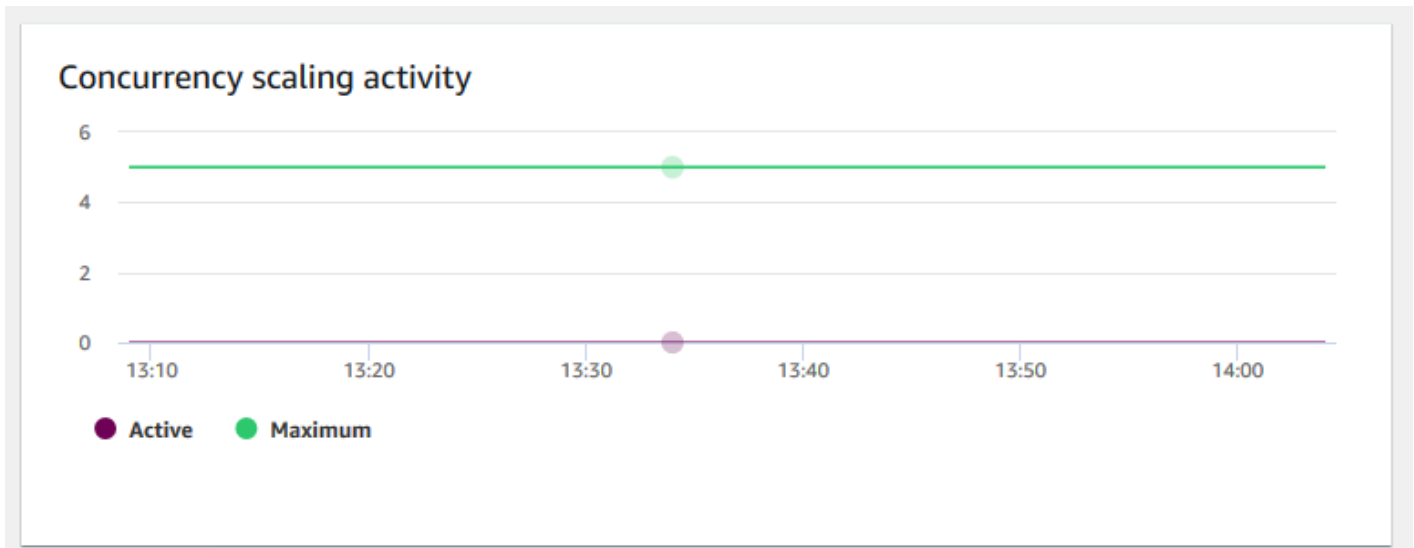
- Durasi kueri per antrian WLM - Menunjukkan jumlah waktu rata-rata untuk menyelesaikan kueri dalam mikrodetik. Anda dapat membandingkan data pada bagan ini untuk mengukur I/O kinerja per antrian WLM dan menyetel kueri yang paling memakan waktu jika perlu.



- Throughput kueri per antrian WLM - Menunjukkan jumlah rata-rata kueri yang diselesaikan per detik. Anda dapat menganalisis data pada bagan ini untuk mengukur kinerja database per antrian WLM.



- Aktivitas penskalaan konkurensi — Menunjukkan jumlah cluster penskalaan konkurensi aktif. Saat penskalaan konkurensi diaktifkan, Amazon Redshift secara otomatis menambahkan kapasitas kluster tambahan saat Anda membutuhkannya untuk memproses peningkatan kueri baca bersamaan.



Melihat data riwayat kueri

Anda dapat menggunakan metrik riwayat kueri di Amazon Redshift untuk melakukan hal berikut:

- Mengisolasi dan mendiagnosis masalah kinerja kueri.
- Bandingkan metrik runtime kueri dan metrik kinerja cluster pada timeline yang sama untuk melihat bagaimana keduanya mungkin terkait. Melakukannya membantu mengidentifikasi kueri yang berkinerja buruk, mencari kueri bottleneck, dan menentukan apakah Anda perlu mengubah ukuran klaster untuk beban kerja Anda.
- Telusuri detail kueri tertentu dengan memilihnya di timeline. Ketika Query ID dan properti lainnya ditampilkan berturut-turut di bawah grafik, maka Anda dapat memilih kueri untuk melihat detail kueri. Detail termasuk, misalnya, pernyataan SQL query, rincian eksekusi, dan rencana kueri. Untuk informasi selengkapnya, lihat [Melihat dan menganalisis detail kueri](#).
- Tentukan apakah pekerjaan pemuatan Anda berhasil diselesaikan dan penuh perjanjian tingkat layanan Anda (SLAs).

Untuk menampilkan data riwayat kueri

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster, lalu pilih nama cluster dari daftar untuk membuka detailnya. Rincian cluster ditampilkan, yang dapat mencakup kinerja Cluster, pemantauan Kueri, Database, Datashares, Jadwal, Pemeliharaan, dan tab Properti.

3. Pilih tab Pemantauan kueri untuk metrik tentang kueri Anda.
4. Di bagian Pemantauan kueri, pilih tab Riwayat kueri.

Menggunakan kontrol pada jendela, Anda dapat beralih antara daftar Query dan metrik Cluster.

Saat Anda memilih daftar Kueri, tab menyertakan grafik berikut:

- Query runtime — Aktivitas query pada timeline. Gunakan grafik ini untuk melihat kueri mana yang berjalan dalam jangka waktu yang sama. Pilih kueri untuk melihat detail eksekusi kueri lainnya. Sumbu x menunjukkan periode yang dipilih. Anda dapat memfilter kueri grafik dengan menjalankan, menyelesaikan, memuat, dan sebagainya. Setiap bar mewakili query, dan panjang bar mewakili runtime dari awal bar sampai akhir. Kueri dapat mencakup pernyataan manipulasi data SQL (seperti SELECT, INSERT, DELETE) dan load (seperti COPY). Secara default, 100 kueri berjalan terpanjang teratas ditampilkan untuk periode waktu yang dipilih.
- Kueri dan beban — Daftar kueri dan beban yang berjalan di cluster. Jendela menyertakan opsi untuk Menghentikan kueri jika kueri sedang berjalan.

Saat Anda memilih metrik Cluster, tab menyertakan grafik berikut:

- Query runtime — Aktivitas query pada timeline. Gunakan grafik ini untuk melihat kueri mana yang berjalan dalam jangka waktu yang sama. Pilih kueri untuk melihat detail eksekusi kueri lainnya.
- Pemanfaatan CPU — Pemanfaatan CPU cluster oleh node pemimpin dan rata-rata node komputasi.
- Kapasitas penyimpanan yang digunakan — Persentase kapasitas penyimpanan yang digunakan.
- Koneksi database aktif — Jumlah koneksi database aktif ke cluster.

Pertimbangkan hal berikut saat bekerja dengan grafik riwayat kueri:

- Pilih bilah yang mewakili kueri tertentu pada bagan runtime Kueri untuk melihat detail tentang kueri tersebut. Anda juga dapat, memilih ID kueri pada Kueri dan memuat daftar untuk melihat detailnya.
- Anda dapat menggesek untuk memilih bagian dari bagan runtime Query untuk memperbesar untuk menampilkan periode waktu tertentu.
- Pada bagan runtime Query, agar semua data dipertimbangkan oleh filter yang Anda pilih, teruskan halaman melalui semua halaman yang tercantum pada daftar Kueri dan muat.

- Anda dapat mengubah kolom mana dan jumlah baris yang ditampilkan pada daftar Kueri dan beban menggunakan jendela preferensi yang ditampilkan oleh ikon roda gigi pengaturan.
- Daftar Kueri dan beban juga dapat ditampilkan dengan menavigasi dari ikon Kueri navigator kiri, Kueri, dan beban. Untuk informasi selengkapnya, lihat [Melihat kueri dan beban](#).

Grafik riwayat kueri

Contoh berikut menunjukkan grafik yang ditampilkan di konsol Amazon Redshift baru.

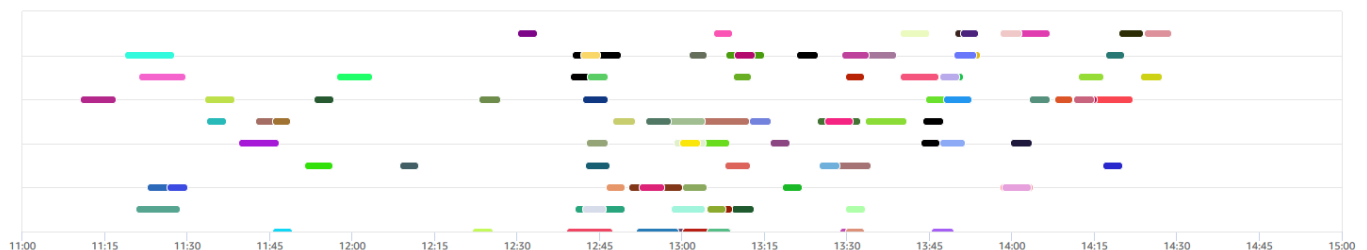
Note

Grafik konsol Amazon Redshift hanya berisi data untuk 100.000 kueri terbaru.

• Waktu proses kueri

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.



• Kueri dan beban

Queries and loads(100)

Filter queries

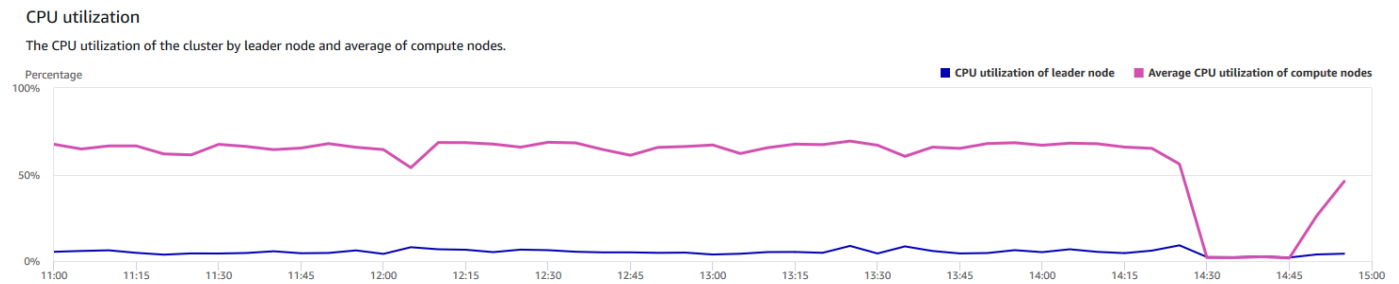
1

2

Terminate query

☐	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

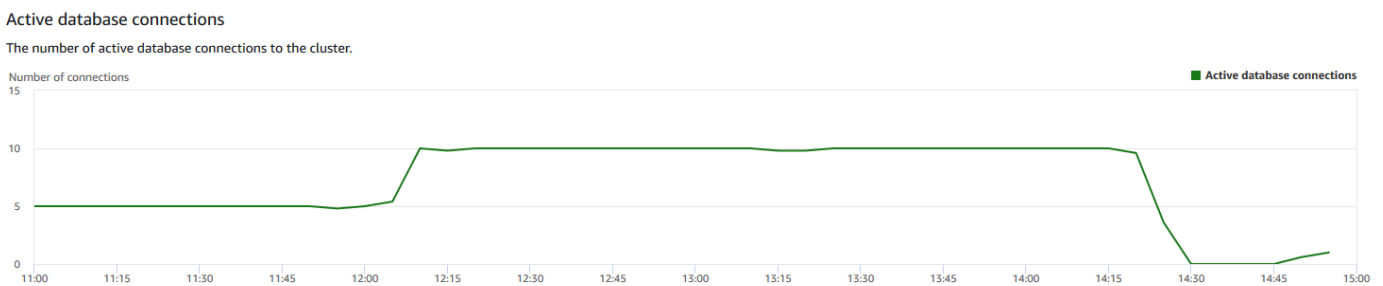
• Pemanfaatan CPU



- Kapasitas penyimpanan yang digunakan



- Koneksi basis data aktif



Melihat data kinerja database

Anda dapat menggunakan metrik kinerja database di Amazon Redshift untuk melakukan hal berikut:

- Analisis waktu yang dihabiskan oleh kueri dengan memproses tahapan. Anda dapat mencari tren yang tidak biasa dalam jumlah waktu yang dihabiskan dalam satu panggung.
- Analisis jumlah kueri, durasi, dan throughput kueri berdasarkan rentang durasi (pendek, sedang, panjang).
- Cari tren tentang waktu tunggu kueri berdasarkan prioritas kueri (Terendah, Rendah, Normal, Tinggi, Tertinggi, Kritis).
- Cari tren dalam durasi kueri, throughput, atau waktu tunggu berdasarkan antrian WLM.

Untuk menampilkan data kinerja database

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster, lalu pilih nama cluster dari daftar untuk membuka detailnya. Detail cluster ditampilkan, termasuk kinerja Cluster, pemantauan Query, Database, Datashares, Jadwal, Pemeliharaan, dan tab Properti.
3. Pilih tab Pemantauan kueri untuk metrik tentang kueri Anda.
4. Di bagian Pemantauan kueri, pilih tab Kinerja basis data.

Menggunakan kontrol di jendela, Anda dapat beralih antara metrik Cluster dan metrik antrian WLM.

Saat Anda memilih metrik Cluster, tab menyertakan grafik berikut:

- Kerusakan eksekusi beban kerja - Waktu yang digunakan dalam tahap pemrosesan kueri.
- Kueri berdasarkan rentang durasi — Jumlah kueri pendek, menengah, dan panjang.
- Throughput kueri — Jumlah rata-rata kueri yang diselesaikan per detik.
- Durasi kueri — Jumlah rata-rata waktu untuk menyelesaikan kueri.
- Waktu tunggu antrian rata-rata berdasarkan prioritas — Total waktu kueri yang dihabiskan menunggu di antrian WLM berdasarkan prioritas kueri.

Saat Anda memilih metrik antrian WLM, tab menyertakan grafik berikut:

- Durasi kueri berdasarkan antrian — Durasi kueri rata-rata menurut antrian WLM.
- Hasil kueri berdasarkan antrian — Jumlah rata-rata kueri yang diselesaikan per detik oleh antrian WLM.
- Waktu tunggu kueri berdasarkan antrian — Durasi rata-rata kueri yang dihabiskan menunggu dengan antrian WLM.

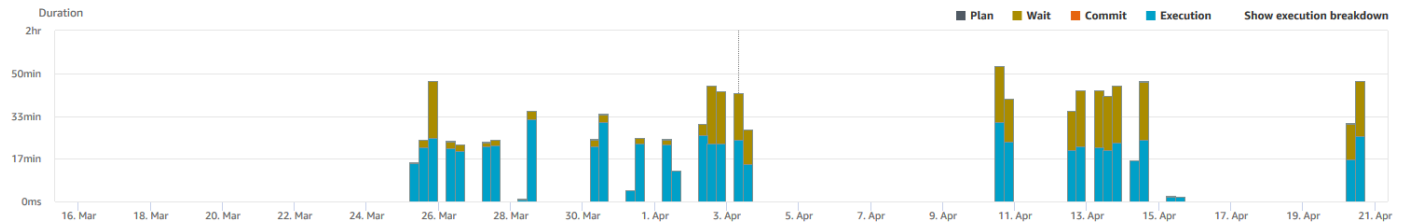
Grafik kinerja basis data

Contoh berikut menunjukkan grafik yang ditampilkan di konsol Amazon Redshift baru.

- Rincian eksekusi beban kerja

Workload execution breakdown

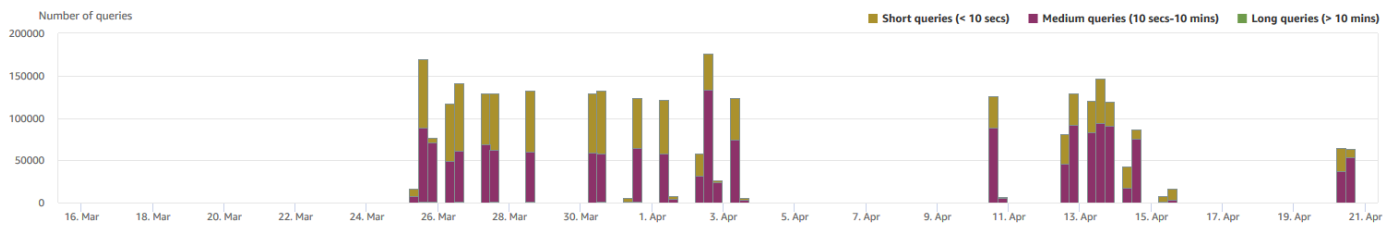
The time used in query processing stages.



- Kueri berdasarkan rentang durasi

Queries by duration range

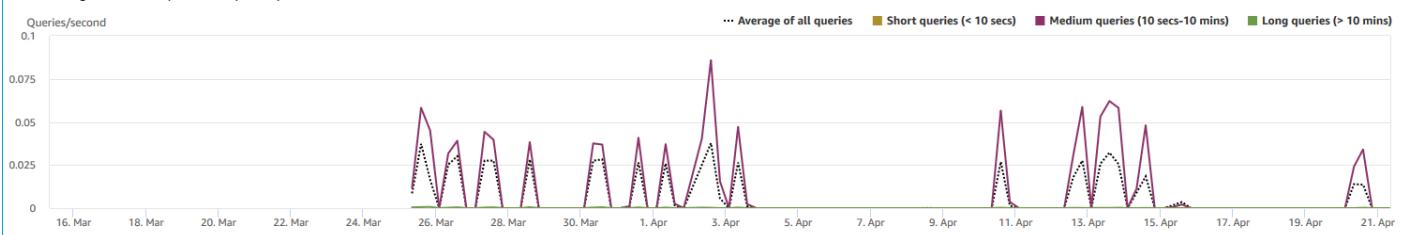
The number of short, medium and long queries.



- Throughput kueri

Query throughput

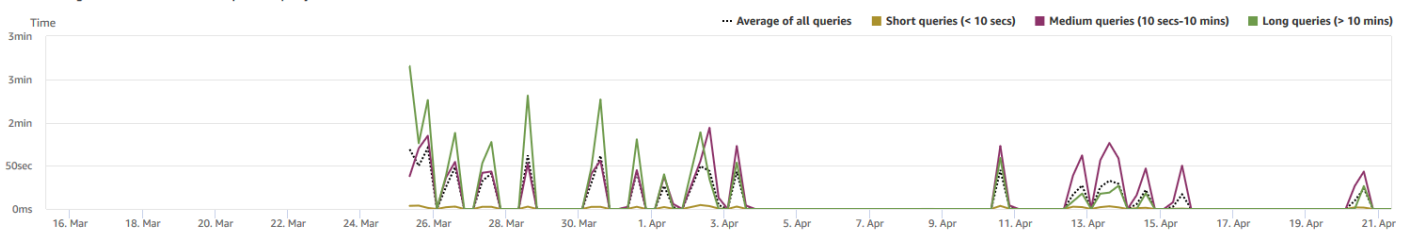
The average number of queries completed per second.



- Durasi kueri

Query Duration

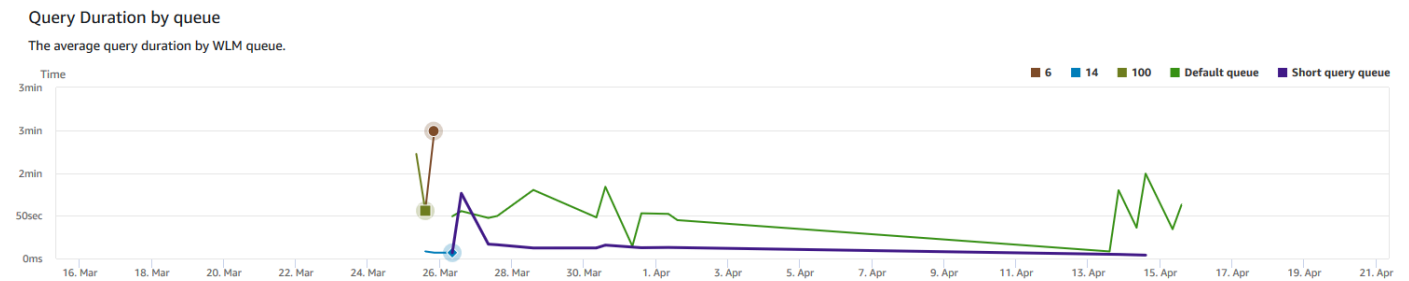
The average amount of time to complete a query.



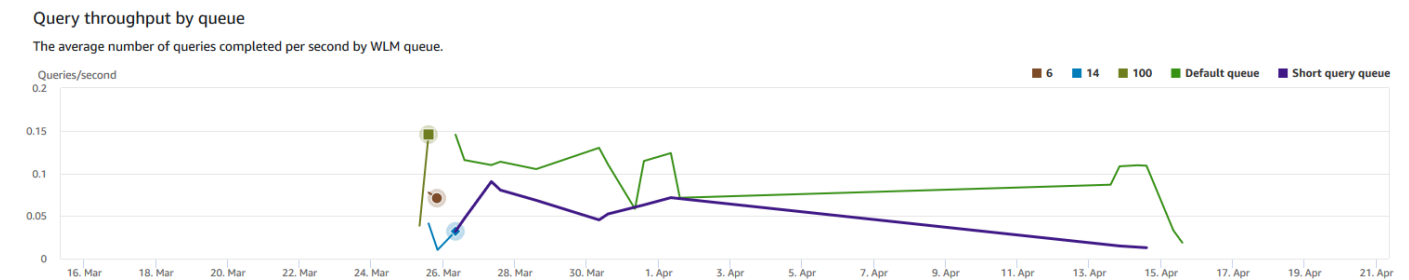
- Waktu tunggu antrian rata-rata berdasarkan prioritas



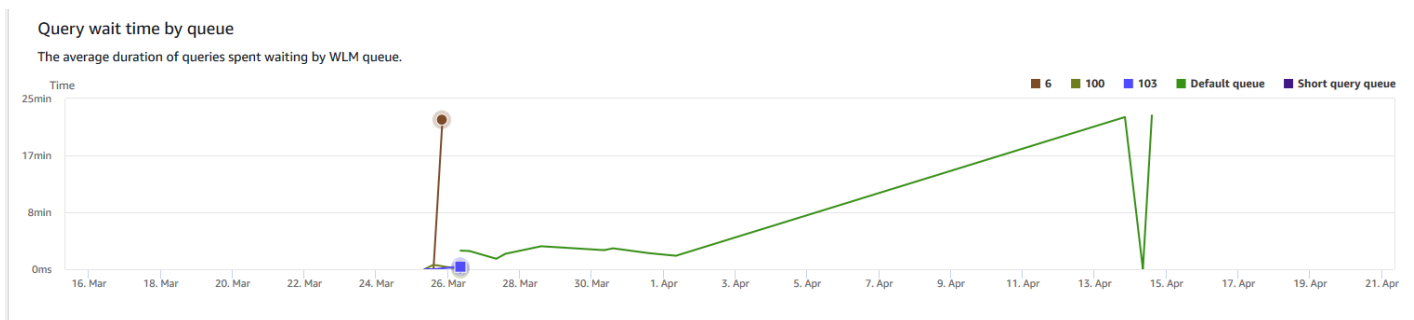
- Durasi kueri berdasarkan antrian



- Hasil kueri berdasarkan antrian



- Kueri waktu tunggu dengan antrian



Melihat data konkurensi beban kerja dan penskalaan konkurensi

Dengan menggunakan metrik penskalaan konkurensi di Amazon Redshift, Anda dapat melakukan hal berikut:

- Analisis apakah Anda dapat mengurangi jumlah kueri antrian dengan mengaktifkan penskalaan konkurensi. Anda dapat membandingkan dengan antrian WLM atau untuk semua antrian WLM.
- Lihat aktivitas penskalaan konkurensi dalam klaster penskalaan konkurensi. Ini dapat memberi tahu Anda jika penskalaan konkurensi dibatasi oleh `max_concurrency_scaling_clusters`. Jika demikian, Anda dapat memilih untuk meningkatkan `max_concurrency_scaling_clusters` parameter DB.
- Lihat total penggunaan penskalaan konkurensi yang dijumlahkan di semua cluster penskalaan konkurensi.

Untuk menampilkan data penskalaan konkurensi

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster, lalu pilih nama cluster dari daftar untuk membuka detailnya. Rincian cluster ditampilkan, yang dapat mencakup kinerja Cluster, pemantauan Kueri, Database, Datashares, Jadwal, Pemeliharaan, dan tab Properti.
3. Pilih tab Pemantauan kueri untuk metrik tentang kueri Anda.
4. Di bagian Pemantauan kueri, pilih tab Konkurensi beban kerja.

Tab ini mencakup grafik berikut:

- Query antrian vs Menjalankan di cluster — Jumlah kueri yang berjalan (dari cluster utama dan cluster penskalaan konkurensi) dibandingkan dengan jumlah kueri yang menunggu di semua antrian WLM di cluster.
- Query antrian vs Menjalankan per antrian — Jumlah kueri yang berjalan (dari cluster utama dan cluster penskalaan konkurensi) dibandingkan dengan jumlah atau kueri yang menunggu di setiap antrian WLM.
- Aktivitas penskalaan konkurensi — Jumlah cluster penskalaan konkurensi yang secara aktif memproses kueri.
- Penggunaan penskalaan konkurensi — Penggunaan cluster penskalaan konkurensi yang memiliki aktivitas pemrosesan kueri aktif.

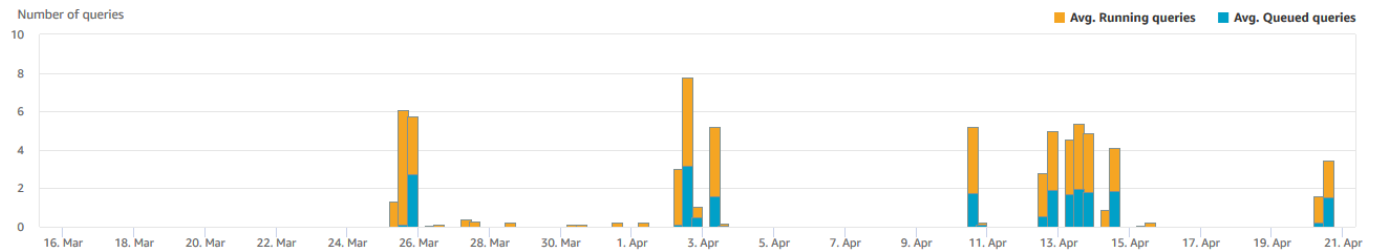
Grafik konkurensi beban kerja

Contoh berikut menunjukkan grafik yang ditampilkan di konsol Amazon Redshift baru. Untuk membuat grafik serupa di Amazon CloudWatch, Anda dapat menggunakan penskalaan konkurensi dan metrik WLM. CloudWatch Untuk informasi selengkapnya tentang CloudWatch metrik untuk Amazon Redshift, lihat. [Data kinerja di Amazon Redshift](#)

- Antrian vs Menjalankan kueri di cluster

Queued vs. Running queries on the cluster

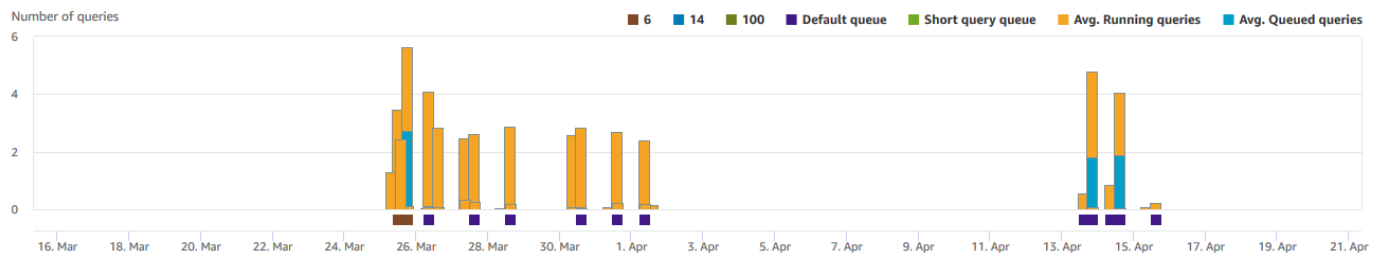
The number of queries running (from the main cluster and concurrency scaling cluster) compared to the number of queries waiting in all WLM queues in the cluster.



- Antrian vs Menjalankan kueri per antrian

Queued vs. Running queries per queue

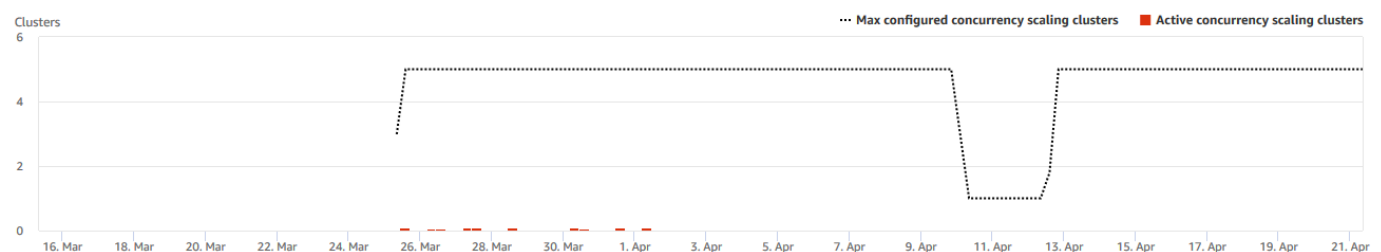
The number of queries running (from the main cluster and concurrency scaling cluster) compared to the number of queries waiting in each WLM queue.



- Aktivitas penskalaan konkurensi

Concurrency scaling activity

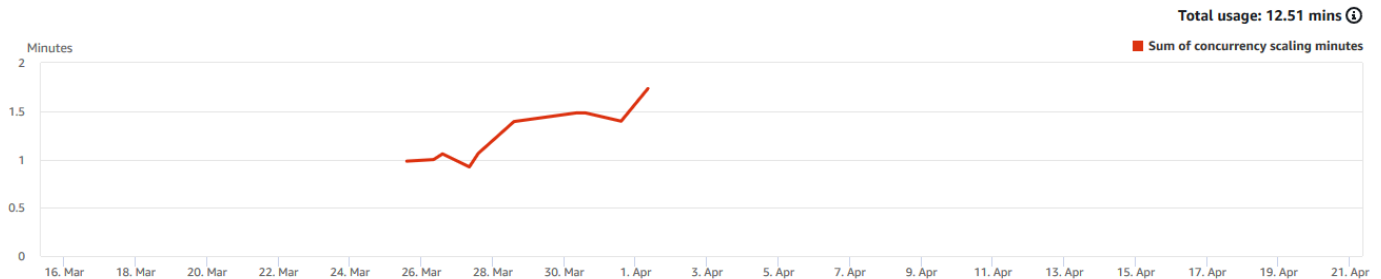
The number of concurrency scaling clusters that are actively processing queries.



- Penggunaan penskalaan konkurensi

Concurrency scaling usage

The usage of concurrency scaling clusters that have active query processing activity.



Melihat data optimasi otomatis

Konsol Amazon Redshift menyediakan informasi tentang pengoptimalan otomatis, atau otonom, yang dijalankan menggunakan sumber daya komputasi tambahan. Anda dapat menggunakan informasi ini untuk melacak penggunaan dan memantau apakah batas penggunaan telah tercapai. Meskipun Amazon Redshift tidak menagih Anda untuk otonom yang dijalankan pada cluster yang disediakan itu sendiri, Amazon Redshift menagih Anda untuk menjalankan otonom menggunakan sumber daya komputasi tambahan. Untuk informasi selengkapnya, lihat [Mengalokasikan sumber daya komputasi tambahan untuk pengoptimalan database otomatis di Panduan Pengembang Database Amazon Redshift](#).

Untuk melihat data otonom komputasi tambahan

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster, lalu pilih nama cluster dari daftar untuk membuka detailnya.
3. Dari halaman detail klaster, pilih Kelola batas penggunaan dari menu tarik-turun Tindakan. Anda juga dapat memilih tab Maintenance untuk cluster, lalu gulir ke bawah dan pilih Buat batas penggunaan.
4. Grafik yang menunjukkan data otonom komputasi tambahan muncul di bawah bagian berjudul Batas penggunaan untuk komputasi ekstra untuk pengoptimalan otomatis. Grafik menampilkan jumlah waktu Amazon Redshift menjalankan otonom menggunakan sumber daya komputasi tambahan dalam periode waktu tertentu.

Melihat kueri dan beban

Konsol Amazon Redshift menyediakan informasi tentang kueri dan beban yang berjalan di database. Anda dapat menggunakan informasi ini untuk mengidentifikasi dan memecahkan masalah kueri yang membutuhkan waktu lama untuk diproses dan yang menciptakan kemacetan yang mencegah kueri lain diproses secara efisien. Anda dapat menggunakan informasi kueri di konsol Amazon Redshift untuk memantau pemrosesan kueri.

Untuk menampilkan data kinerja kueri

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Kueri dan beban untuk menampilkan daftar kueri untuk akun Anda.

Secara default, daftar menampilkan kueri untuk semua cluster Anda selama 24 jam terakhir. Anda dapat mengubah cakupan tanggal yang ditampilkan di konsol.

Important

Daftar Kueri dan beban menampilkan kueri yang berjalan paling lama di sistem, hingga 100 kueri.

Melihat dan menganalisis detail kueri

Dengan pengenalan kueri, Anda dapat melihat detail kueri. Detail dapat mencakup, misalnya, status penyelesaian kueri, durasi, pernyataan SQL, dan apakah itu kueri pengguna atau yang ditulis ulang oleh Amazon Redshift. Kueri pengguna adalah kueri yang dikirimkan ke Amazon Redshift, baik dari klien SQL atau dihasilkan oleh alat intelijen bisnis. Amazon Redshift mungkin menulis ulang kueri untuk mengoptimalkannya, dan ini dapat menghasilkan beberapa kueri yang ditulis ulang. Meskipun prosesnya dilakukan oleh Amazon Redshift, Anda melihat kueri yang ditulis ulang pada halaman detail kueri bersama dengan kueri pengguna.

Untuk melihat kueri

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Kueri dan beban untuk menampilkan daftar kueri untuk akun Anda. Anda mungkin perlu mengubah pengaturan di halaman ini untuk menemukan kueri Anda.

3. Pilih pengenalan Kueri dalam daftar untuk menampilkan Detail kueri.

Halaman Detail kueri mencakup detail Kueri dan tab paket Kueri dengan metrik tentang kueri.

Metrik mencakup detail tentang kueri seperti waktu mulai, ID kueri, status, dan durasi. Detail lainnya termasuk apakah kueri dijalankan pada cluster utama atau klaster penskalaan konkurensi, dan apakah itu kueri induk atau yang ditulis ulang.

Melihat kinerja cluster saat kueri berjalan

Anda dapat memantau kinerja klaster Anda saat kueri dijalankan untuk mengidentifikasi potensi kemacetan dan mengoptimalkan eksekusi kueri. Melihat kinerja klaster saat kueri dijalankan memberikan tampilan real-time dari metrik tingkat sistem, seperti pemanfaatan CPU, I/O disk, dan lalu lintas jaringan, serta detail tingkat kueri seperti waktu eksekusi, data yang diproses, dan langkah kueri. Prosedur berikut memandu Anda dalam mengakses dan menafsirkan metrik kinerja untuk mengelola dan mengoptimalkan kluster yang disediakan secara efektif.

Untuk menampilkan kinerja klaster saat kueri dijalankan

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster, lalu pilih nama cluster dari daftar untuk membuka detailnya. Rincian cluster ditampilkan, yang dapat mencakup kinerja Cluster, pemantauan Kueri, Database, Datashares, Jadwal, Pemeliharaan, dan tab Properti.
3. Pilih tab Pemantauan kueri untuk detail selengkapnya.

Untuk informasi selengkapnya, lihat [Melihat data riwayat kueri](#).

Melihat metrik cluster selama operasi pemuatan

Saat Anda melihat kinerja klaster selama operasi pemuatan, Anda dapat mengidentifikasi kueri yang menghabiskan sumber daya dan bertindak untuk mengurangi efeknya. Anda dapat menghentikan beban jika Anda tidak ingin itu berjalan hingga selesai.

Note

Kemampuan untuk menghentikan kueri dan pemuatan di konsol Amazon Redshift memerlukan izin khusus. Jika Anda ingin pengguna dapat menghentikan kueri dan

pemuatan, pastikan untuk menambahkan `redshift:CancelQuerySession` tindakan ke kebijakan AWS Identity and Access Management (IAM) Anda. Persyaratan ini berlaku baik Anda memilih kebijakan AWS terkelola Amazon Redshift Read Only atau membuat kebijakan kustom di IAM. Pengguna yang memiliki kebijakan Akses Penuh Amazon Redshift sudah memiliki izin yang diperlukan untuk menghentikan kueri dan pemuatan. Untuk informasi selengkapnya tentang tindakan dalam kebijakan IAM untuk Amazon Redshift, lihat [Mengelola akses ke sumber daya](#)

Untuk menampilkan kinerja cluster selama operasi beban

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster, lalu pilih nama cluster dari daftar untuk membuka detailnya. Rincian cluster ditampilkan, yang dapat mencakup kinerja Cluster, pemantauan Kueri, Database, Datashares, Jadwal, Pemeliharaan, dan tab Properti.
3. Pilih tab Pemantauan kueri untuk detail selengkapnya.
4. Di bagian Kueri dan beban, pilih Memuat untuk melihat operasi pemuatan klaster. Jika beban sedang berjalan, Anda dapat mengakhirinya dengan memilih Terminate query.

Melihat bagan rincian beban kerja cluster

Anda bisa mendapatkan tampilan mendetail performa beban kerja Anda dengan melihat bagan rincian eksekusi Workload di konsol. Kami membangun bagan dengan data yang disediakan oleh QueryRuntimeBreakdown metrik. Dengan bagan ini, Anda dapat melihat berapa banyak waktu yang dihabiskan kueri Anda dalam berbagai tahap pemrosesan, seperti menunggu dan merencanakan.

Note

Bagan rincian eksekusi Workload tidak ditampilkan untuk kluster simpul tunggal.

Daftar metrik berikut menjelaskan berbagai tahapan pemrosesan:

- QueryPlanning: Waktu yang dihabiskan mengurai dan mengoptimalkan pernyataan SQL.
- QueryWaiting: Waktu yang dihabiskan menunggu dalam antrian manajemen beban kerja (WLM).

- **QueryExecutingRead:** Waktu yang dihabiskan untuk menjalankan kueri baca.
- **QueryExecutingInsert:** Waktu yang dihabiskan untuk menjalankan kueri sisipan.
- **QueryExecutingDelete:** Waktu yang dihabiskan untuk menjalankan kueri hapus.
- **QueryExecutingUpdate:** Waktu yang dihabiskan untuk menjalankan kueri pembaruan.
- **QueryExecutingCtas:** Waktu yang dihabiskan untuk menjalankan CREATE TABLE AS query.
- **QueryExecutingUnload:** Waktu yang dihabiskan untuk menjalankan kueri bongkar muat.
- **QueryExecutingCopy:** Waktu yang dihabiskan untuk menjalankan kueri salinan.

Misalnya, grafik berikut di konsol Amazon Redshift menunjukkan jumlah waktu yang dihabiskan kueri dalam tahapan paket, tunggu, baca, dan tulis. Anda dapat menggabungkan temuan dari grafik ini dengan metrik lain untuk analisis lebih lanjut. Dalam beberapa kasus, grafik Anda mungkin menunjukkan bahwa kueri dengan durasi pendek (yang diukur dengan `QueryDuration` metrik) menghabiskan waktu lama dalam tahap menunggu. Dalam kasus ini, Anda dapat meningkatkan tingkat konkurensi WLM untuk antrian tertentu untuk meningkatkan throughput.

Berikut ini, adalah contoh bagan pemecahan eksekusi beban kerja. Dalam bagan, nilai sumbu y adalah durasi rata-rata setiap tahap pada waktu yang ditentukan yang ditampilkan sebagai grafik batang bertumpuk.

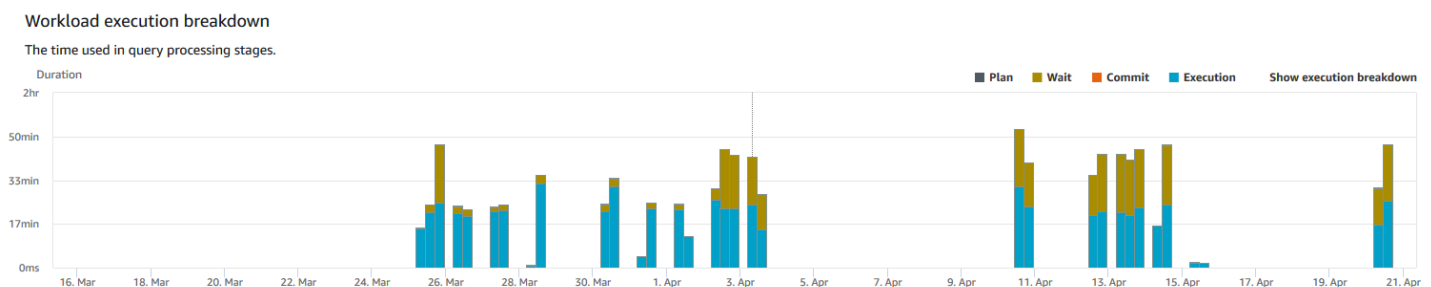
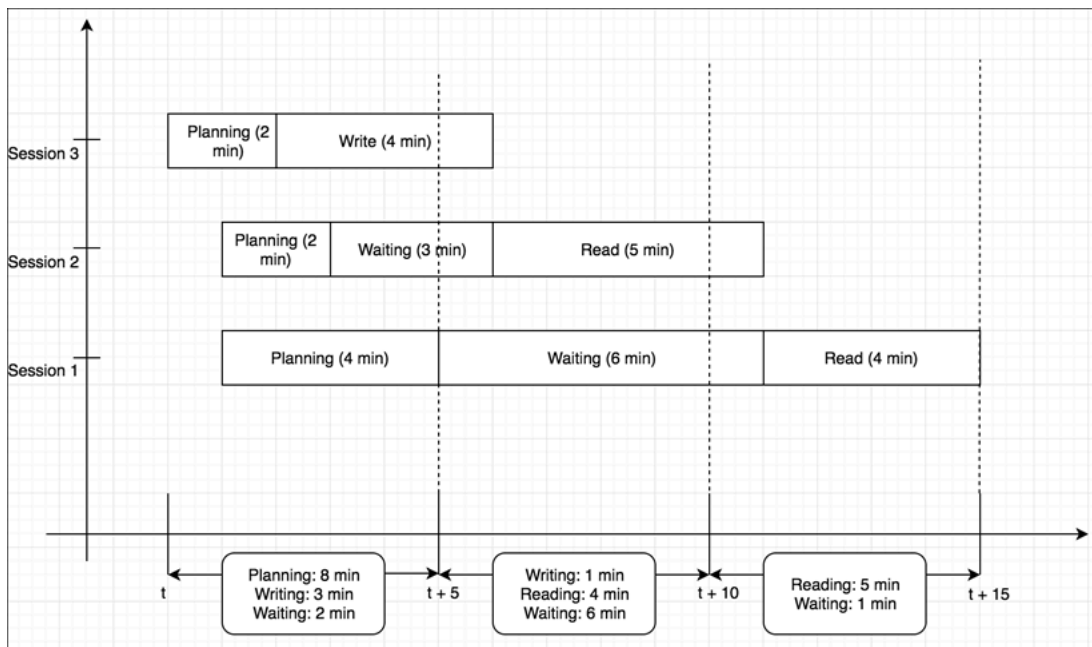


Diagram berikut menggambarkan bagaimana Amazon Redshift menggabungkan pemrosesan kueri untuk sesi bersamaan.



Untuk melihat bagan rincian beban kerja cluster

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster, lalu pilih nama cluster dari daftar untuk membuka detailnya. Rincian cluster ditampilkan, yang dapat mencakup kinerja Cluster, pemantauan Kueri, Database, Datashares, Jadwal, Pemeliharaan, dan tab Properti.
3. Pilih tab Pemantauan kueri untuk metrik tentang kueri Anda.
4. Di bagian Pemantauan kueri, pilih Kinerja database, dan pilih Metrik klaster.

Metrik berikut dibuat grafik untuk rentang waktu yang dipilih sebagai bagan batang bertumpuk:

- Rencanakan waktu
- Tunggu waktu
- Komit waktu
- Waktu eksekusi

Menganalisis eksekusi kueri

Anda dapat menganalisis detail eksekusi kueri untuk memahami bagaimana kinerjanya dan mengidentifikasi area potensial untuk pengoptimalan. Menganalisis kueri memberikan wawasan

ke dalam rencana kueri, termasuk langkah-langkah yang terlibat, waktu yang diambil oleh setiap langkah, dan jumlah data yang diproses. Kasus penggunaan umum termasuk pemecahan masalah kueri yang berjalan lambat, mengoptimalkan strategi distribusi data, dan mengidentifikasi peluang untuk menulis ulang kueri atau pengindeksan.

Untuk menganalisis kueri

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Kueri dan beban untuk menampilkan daftar kueri untuk akun Anda. Anda mungkin perlu mengubah pengaturan di halaman ini untuk menemukan kueri Anda.
3. Pilih pengenalan Kueri dalam daftar untuk menampilkan Detail kueri.

Halaman Detail kueri mencakup detail Kueri dan tab paket Kueri dengan metrik tentang kueri.

Note

Anda juga dapat menavigasi ke halaman Detail kueri dari halaman detail Cluster, tab Riwayat kueri saat Anda menelusuri kueri dalam grafik runtime Kueri.

Halaman detail Query berisi bagian-bagian berikut:

- Daftar kueri yang ditulis ulang, seperti yang ditunjukkan pada gambar berikut.

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	

- Sebuah bagian Query details, seperti yang ditunjukkan pada gambar berikut.

Query details				
Query ID 122927	Cluster ✓ dnd-sudhare-qa	User [redacted]	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

- Tab Query details yang berisi SQL yang dijalankan dan detail Eksekusi tentang proses.
- Tab Query plan yang berisi langkah-langkah Query plan dan informasi lain tentang rencana kueri. Tabel ini juga berisi grafik tentang cluster saat kueri dijalankan.
- Status kesehatan cluster

Cluster health status

Cluster health during the workload.

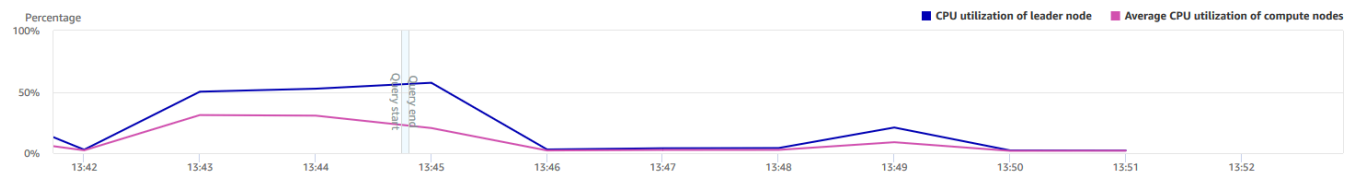
■ Healthy ■ Unhealthy



• Pemanfaatan CPU

CPU utilization

The CPU utilization of the cluster by leader node and average of compute nodes.



• Kapasitas penyimpanan yang digunakan

Storage capacity used

The percent of the storage capacity used.



• Koneksi basis data aktif

Active database connections

The number of active database connections to the cluster.



Membuat alarm

Alarm yang Anda buat di konsol CloudWatch Amazon Redshift adalah alarm. Mereka berguna karena membantu Anda membuat keputusan proaktif tentang cluster atau instance tanpa server Anda. Anda dapat menyetel satu atau beberapa alarm pada salah satu metrik yang tercantum dalam [Data kinerja di Amazon Redshift](#). Misalnya, menyetel alarm untuk high CPUUtilization pada node cluster

membantu menunjukkan kapan node terlalu banyak digunakan. Alarm untuk high DataStorage akan melacak ruang penyimpanan yang digunakan namespace tanpa server Anda untuk data Anda.

Dari Tindakan, Anda dapat memodifikasi atau menghapus alarm. Anda juga dapat membuat peringatan lonceng atau kendur untuk mengirim peringatan dari CloudWatch Slack atau Amazon Chime dengan menentukan URL webhook Slack atau Amazon Chime.

Di bagian ini, Anda dapat menemukan cara membuat alarm menggunakan konsol Amazon Redshift. Anda dapat membuat alarm menggunakan CloudWatch konsol atau cara lain Anda bekerja dengan metrik, seperti dengan AWS CLI atau AWS SDK.

Untuk membuat CloudWatch alarm dengan konsol Amazon Redshift

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>

Jika Anda menggunakan Amazon Redshift Serverless, Pilih Buka Tanpa Server di kanan atas dasbor.

2. Pada menu navigasi, pilih Alarm, lalu pilih Buat alarm.
3. Pada halaman Buat alarm, masukkan properti untuk membuat CloudWatch alarm.
4. Pilih Buat alarm.

Mengakhiri kueri yang sedang berjalan

Anda juga dapat menggunakan halaman Kueri untuk mengakhiri kueri yang sedang berlangsung.

Note

Kemampuan untuk menghentikan kueri dan pemuatan di konsol Amazon Redshift memerlukan izin khusus. Jika Anda ingin pengguna dapat menghentikan kueri dan pemuatan, pastikan untuk menambahkan `redshift:CancelQuerySession` tindakan ke kebijakan AWS Identity and Access Management (IAM) Anda. Persyaratan ini berlaku baik Anda memilih kebijakan AWS terkelola Amazon Redshift Read Only atau membuat kebijakan kustom di IAM. Pengguna yang memiliki kebijakan Akses Penuh Amazon Redshift sudah memiliki izin yang diperlukan untuk menghentikan kueri dan pemuatan. Untuk informasi selengkapnya tentang tindakan dalam kebijakan IAM untuk Amazon Redshift, lihat [Mengelola akses ke sumber daya](#)

Untuk mengakhiri kueri yang sedang berjalan

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Kueri dan beban untuk menampilkan daftar kueri untuk akun Anda.
3. Pilih kueri berjalan yang ingin Anda akhiri dalam daftar, lalu pilih Hentikan kueri.

Metrik kinerja di konsol CloudWatch

Saat bekerja dengan metrik Amazon Redshift di CloudWatch konsol, ingatlah beberapa hal:

- Data kinerja kueri dan muat hanya tersedia di konsol Amazon Redshift.
- Beberapa Metrik CloudWatch memiliki unit yang berbeda dari yang digunakan di konsol Amazon Redshift. Misalnya, `WriteThroughput` ditampilkan dalam GB/s (dibandingkan dengan Bytes/s in CloudWatch), yang merupakan unit yang lebih relevan untuk ruang penyimpanan tipikal node.

Saat bekerja dengan metrik Amazon Redshift di CloudWatch konsol, alat baris perintah, atau Amazon SDK, ingatlah konsep ini:

1. Pertama, tentukan dimensi metrik yang akan digunakan. Dimensi adalah pasangan nama-nilai yang membantu Anda mengidentifikasi metrik secara unik. Dimensi untuk Amazon Redshift adalah `ClusterIdentifier` dan `NodeID`. Di CloudWatch konsol, Redshift `Cluster` dan Redshift `Node` tampilan disediakan untuk dengan mudah memilih dimensi cluster dan node-spesifik. Untuk informasi selengkapnya tentang dimensi, lihat [Dimensi](#) dalam Panduan CloudWatch Pengembang.
2. Kemudian, tentukan nama metrik, seperti `ReadIOPS`.

Tabel berikut merangkum jenis dimensi metrik Amazon Redshift yang tersedia untuk Anda. Bergantung pada metrik, data tersedia dalam interval 1 menit atau 5 menit tanpa biaya. Untuk informasi selengkapnya, lihat [Metrik Pergeseran Merah Amazon](#).

CloudWatch namespace	Dimensi	Deskripsi
AWS/Redshift	NodeID	Filter meminta data yang spesifik untuk node cluster. NodeID adalah "Leader", "Shared", atau "Compute-N" di mana N adalah 0, 1,... untuk jumlah node dalam cluster.

CloudWatch namespace	Dimensi	Deskripsi
		“Shared” berarti bahwa cluster hanya memiliki satu node, yaitu node pemimpin dan node komputasi digabungkan.
AWS/Redshift	ClusterIdentifier	Filter meminta data yang khusus untuk cluster. Metrik yang khusus untuk cluster meliputi HealthStatus, MaintenanceMode, dan DatabaseConnections. Metrik umum untuk dimensi ini (misalnya, ReadIOPS) yang juga merupakan metrik node mewakili agregat data metrik node. Berhati-hatilah dalam menafsirkan metrik ini karena mereka menggabungkan perilaku pemimpin dan node komputasi.

Bekerja dengan metrik gateway dan volume mirip dengan bekerja dengan metrik layanan lainnya. Banyak tugas umum diuraikan dalam CloudWatch dokumentasi, termasuk yang berikut:

- [Lihat metrik yang tersedia](#)
- [Dapatkan statistik untuk metrik](#)
- [Membuat CloudWatch alarm](#)

Profiler kueri

Dokumen ini menjelaskan Query profiler, alat grafis untuk menganalisis komponen dan kinerja kueri.

Query profiler adalah fitur pemantauan dan pemecahan masalah kueri yang dapat dilihat melalui konsol Amazon Redshift. Ini membantu untuk menganalisis kinerja kueri. Tujuan utamanya adalah untuk menunjukkan urutan visual dan grafis, rencana eksekusi, dan statistik tentang kueri, dan membuatnya lebih mudah dipahami dan dipecahkan. Query profiler membantu Anda menganalisis jenis komponen kueri berikut:

- Kueri anak - Kueri anak adalah bagian dari pekerjaan dari kueri. Amazon Redshift dapat membagi kueri menjadi beberapa kueri turunan jika lebih efisien daripada memprosesnya sebagai satu kueri besar. Di profiler Anda dapat melihat properti dari setiap kueri anak. Kueri anak terdiri dari aliran dan sub-komponen tambahan.

Jenis kueri anak yang ditampilkan oleh profiler Query biasanya mencakup yang berikut:

- Kueri tabel temp: Teks kueri anak ini dimulai dengan CREATE TEMP TABLE perintah. Kueri anak ini membuat tabel sementara untuk kueri turunan lainnya untuk diproses.
- Kueri statistik: Profiler kueri menambahkan komentar berikut ke awal kueri anak ini untuk membantu mengidentifikasinya:

```
-- collect statistics of child query queryID
```

Kueri anak ini mengumpulkan informasi yang digunakan mesin kueri Amazon Redshift untuk mengoptimalkan kinerja.

Note

Query Profiler menampilkan kueri yang disediakan pengguna sebagai kueri turunan terakhir yang dijalankan Amazon Redshift.

- Stream — Stream adalah kumpulan segmen yang dibagi di atas irisan node komputasi yang tersedia. Setiap kueri anak terdiri dari satu atau lebih segmen. Di Query profiler, Anda dapat melihat properti dari setiap aliran, seperti waktu eksekusi. Dengan melirik daftar aliran, kemungkinan Anda dapat dengan cepat menemukan kemacetan kinerja.
- Segmen — Segmen adalah kombinasi dari beberapa langkah yang dapat dijalankan oleh satu proses. Segmen juga merupakan unit kompilasi terkecil yang dapat dieksekusi oleh irisan node komputasi. Slice adalah unit pemrosesan paralel di Amazon Redshift. Segmen dalam aliran berjalan secara paralel. Profiler kueri tidak menampilkan segmen secara grafis, tetapi Anda dapat mengakses informasi segmen untuk langkah di panel detail langkah tersebut.
- Langkah - Setiap segmen terdiri dari kumpulan langkah. Langkah adalah bagian dari pekerjaan dalam kueri. Langkah-langkah dapat mencakup hashjoin, misalnya, atau pemindaian, yang merupakan pembacaan catatan dari tabel.

Untuk informasi selengkapnya tentang aliran, segmen, dan langkah, lihat [Alur kerja perencanaan dan eksekusi kueri](#) di Panduan Pengembang Database Amazon Redshift.

Query profiler menampilkan informasi yang dikembalikan oleh SYS_QUERY_HISTORY, SYS_QUERY_DETAIL, SYS_QUERY_EXPLAIN, dan SYS_CHILD_QUERY_TEXT tampilan. [Untuk informasi selengkapnya tentang tampilan](#)

ini, lihat [SYS_QUERY_HISTORY](#), [SYS_QUERY_DETAIL](#), [SYS_QUERY_EXPLOW](#), dan [SYS_CHILD_QUERY_TEXT](#) di Panduan Pengembang [Database Amazon Redshift](#).

Query profiler hanya menampilkan informasi kueri untuk kueri yang baru saja berjalan di database. Kueri yang selesai menggunakan data cache yang telah diisi sebelumnya daripada berjalan terhadap database tidak akan memiliki profil kueri jika informasi sebelumnya tidak tersedia untuk itu. Ini karena Amazon Redshift tidak menghasilkan paket kueri untuk itu.

Prasyarat untuk menggunakan Query profiler

Tampilan pemantauan SYS dirancang untuk kemudahan penggunaan dan mengurangi kompleksitas, menyediakan rangkaian metrik lengkap untuk pemantauan dan pemecahan masalah yang efektif. Tampilan pemantauan SYS juga menjamin riwayat kueri selama tujuh hari terakhir terlepas dari ukuran atau aktivitas cluster. Pengguna hanya memiliki visibilitas untuk kueri yang mereka jalankan, sementara pengguna super memiliki visibilitas untuk kueri dari semua pengguna.

Akun pengguna atau peran IAM Anda memerlukan izin untuk mengakses bagian Kueri dan pemantauan basis data konsol. Bagian ini menjelaskan cara menambahkan izin ke akun atau peran pengguna.

Gunakan kebijakan berikut untuk menambahkan izin minimum ke akun atau peran pengguna IAM Anda:

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:*"
  ]
}
}

```

Topik

- [Memberikan izin pemantauan kueri untuk peran](#)
- [Memberikan izin pemantauan kueri untuk pengguna](#)
- [Kredensi sementara menggunakan identitas IAM Anda](#)

Memberikan izin pemantauan kueri untuk peran

Pengguna dengan peran yang memiliki `sys:monitor` izin dapat melihat semua kueri. Pengguna dengan peran yang memiliki `sys:operator` izin dapat membatalkan kueri, menganalisis riwayat kueri, dan melakukan operasi vakum.

Untuk memberikan izin pemantauan kueri untuk peran

1. Gunakan perintah berikut untuk menyediakan akses monitor sistem, di mana *role-name* nama peran yang ingin Anda berikan aksesnya.

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

2. (Opsional) Gunakan perintah berikut untuk menyediakan akses operator sistem, di mana *role-name* nama peran yang ingin Anda berikan aksesnya.

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

Memberikan izin pemantauan kueri untuk pengguna

Pengguna dengan `sys:monitor` izin dapat melihat semua kueri. Pengguna dengan `sys:operator` izin dapat membatalkan kueri, menganalisis riwayat kueri, dan melakukan operasi vakum.

Untuk memberikan izin pemantauan kueri bagi pengguna

1. Gunakan perintah berikut untuk menyediakan akses monitor sistem, di mana *user-name* nama pengguna yang ingin Anda berikan aksesnya.

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

2. (Opsional) Gunakan perintah berikut untuk menyediakan akses operator sistem, di mana *-name* adalah nama pengguna yang ingin Anda berikan aksesnya.

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

Kredensi sementara menggunakan identitas IAM Anda

Opsi ini hanya tersedia saat menghubungkan ke cluster. Dengan metode ini, Query Profiler memetakan nama pengguna ke identitas IAM Anda dan menghasilkan kata sandi sementara untuk terhubung ke database sebagai identitas IAM Anda. Pengguna yang menggunakan metode ini untuk terhubung harus diizinkan izin IAM untuk `redshift:GetClusterCredentialsWithIAM`. Untuk mencegah pengguna menggunakan metode ini, ubah pengguna atau peran IAM mereka untuk menolak izin ini.

Mengakses profiler Kueri di konsol Amazon Redshift untuk menganalisis kueri

Anda dapat mengakses Query Profiler untuk Amazon Redshift tanpa server atau Amazon Redshift yang disediakan. Lihat bagian berikut untuk detailnya:

Topik

- [Mengakses profiler Kueri di konsol Amazon Redshift untuk Amazon Redshift Tanpa Server](#)
- [Mengakses profiler Kueri di konsol Amazon Redshift untuk Amazon Redshift Disediakan](#)

Mengakses profiler Kueri di konsol Amazon Redshift untuk Amazon Redshift Tanpa Server

Untuk mengakses profiler Query untuk Amazon Redshift Serverless, lakukan hal berikut:

- Buka konsol tanpa server Amazon Redshift.
- Pada panel navigasi, di bawah Monitoring, pilih Query dan pemantauan database.
- Pilih workgroup.

- Pilih Query dan pemantauan database.
- Pilih kueri.
- Pilih tab Query plan di halaman Query details.

Jika paket kueri tersedia, Anda akan melihat daftar kueri anak. Pilih kueri untuk melihatnya di Query profiler.

Mengakses profiler Kueri di konsol Amazon Redshift untuk Amazon Redshift Disediakan

Untuk mengakses profiler Kueri untuk Amazon Redshift Provisioned, lakukan hal berikut:

- Buka dasbor cluster Amazon Redshift Provisioned.
- Pilih cluster
- Pilih Pemantauan kueri.
- Connect ke database
- Pilih Query dan pemantauan database.
- Pilih kueri.

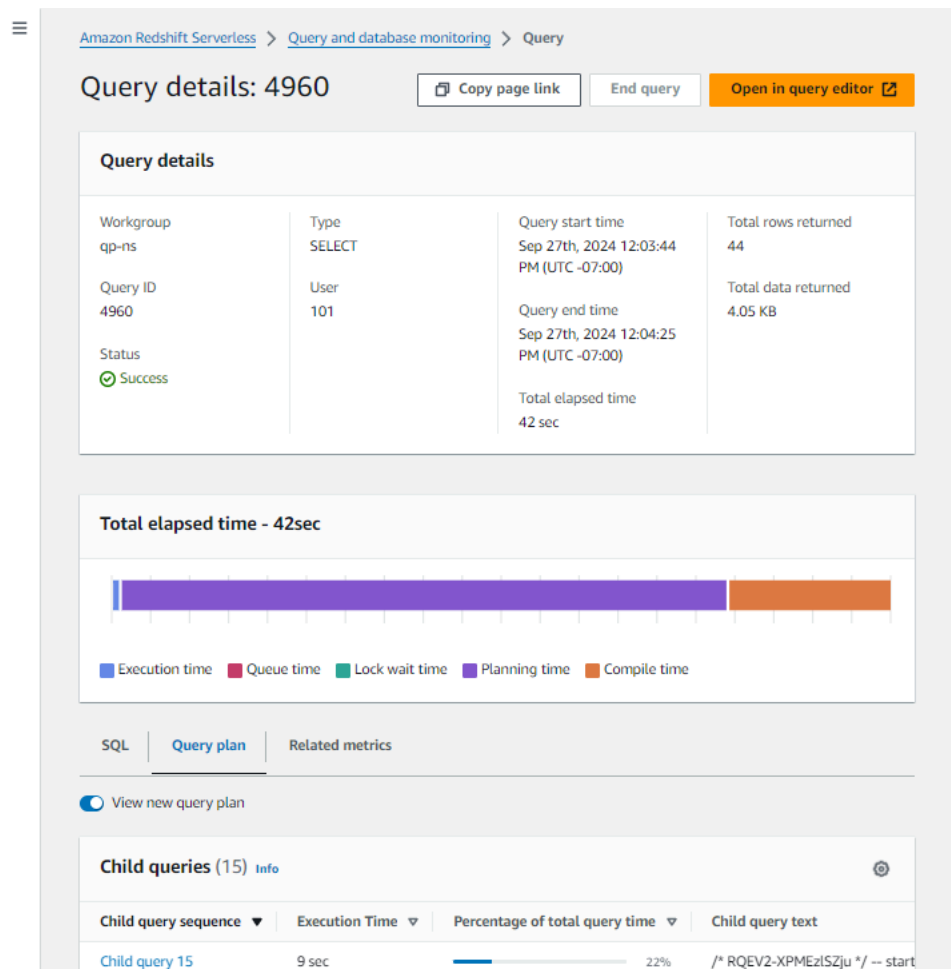
Jika paket kueri tersedia, Anda akan melihat daftar kueri anak. Pilih kueri untuk melihatnya di Query profiler.

Antarmuka pengguna profiler kueri

Query profiler menggunakan halaman berikut untuk menampilkan informasi tentang kueri Anda:

- [Halaman detail kueri](#)— Halaman ini menampilkan statistik dan kueri anak untuk kueri Anda.
- [Halaman kueri anak](#)— Halaman ini menampilkan statistik, aliran, dan representasi visual dari rencana eksekusi untuk kueri anak. Konsol menampilkan halaman ini saat Anda memilih kueri anak dari daftar kueri Anak di halaman Query dan pemantauan database.

Halaman detail kueri



Halaman detail Query memiliki komponen-komponen berikut:

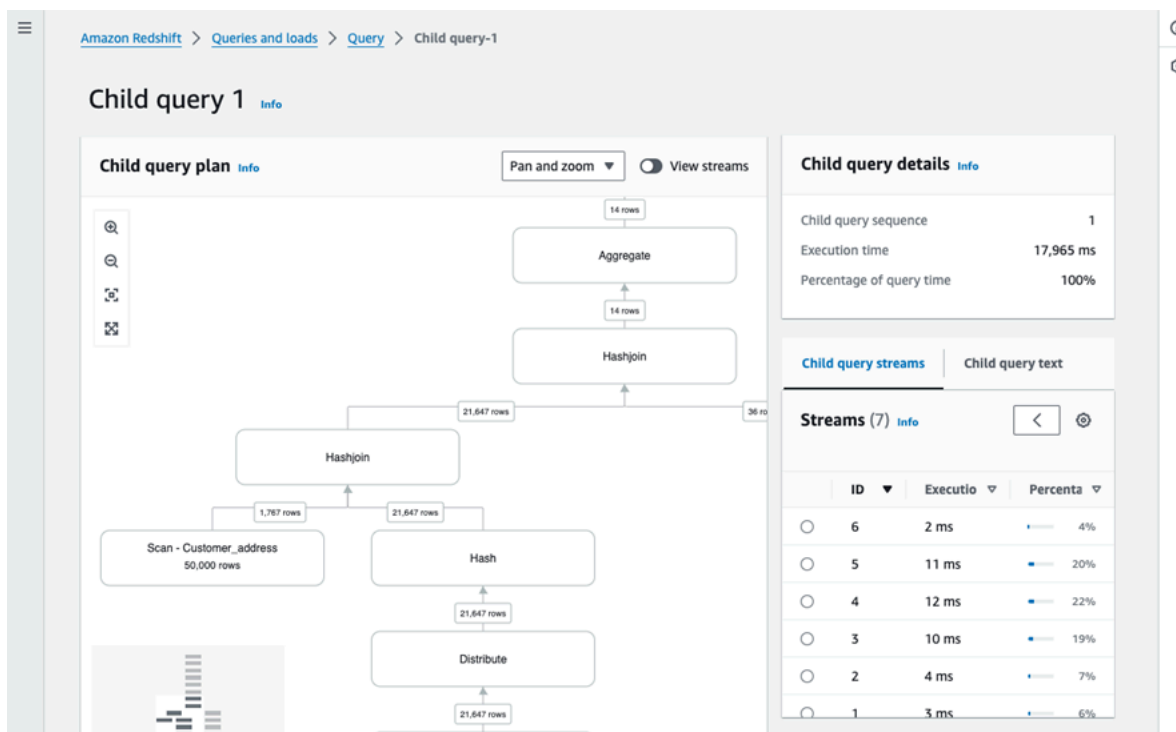
- Panel atas — Panel di bagian atas halaman menampilkan detail tentang kueri, seperti status dan jenis. Untuk informasi tentang sumber informasi yang ditampilkan panel atas, lihat [SYS_QUERY_HISTORY](#) di Panduan Pengembang Database Amazon Redshift.
- Tab SQL - Tab panel bawah ini menunjukkan teks SQL untuk kueri pengguna asli.
- Tab paket kueri — Tab panel bawah ini menampilkan daftar kueri anak yang digunakan Amazon Redshift untuk menyiapkan data dan statistik untuk kueri pengguna. Secara default, daftar kueri Anak menampilkan informasi dan statistik agregat tentang setiap kueri anak. Untuk informasi tentang sumber informasi yang ditampilkan halaman ini, lihat [SYS_QUERY_DETAIL](#) di Panduan Pengembang Database Amazon Redshift.

Anda dapat menambah atau menghapus kolom dari daftar kueri Anak menggunakan menu Preferensi.

- Metrik terkait - Tab panel bawah ini menunjukkan CloudWatch metrik berikut untuk kueri:
 - Kapasitas RPU yang digunakan (untuk kelompok kerja tanpa server): Kapasitas komputasi yang digunakan oleh kueri, diukur dalam Unit Pemrosesan Redshift (). RPU Untuk informasi selengkapnya, lihat [Kapasitas komputasi untuk Amazon Redshift Serverless](#).
 - Status kesehatan cluster, Pemanfaatan CPU, Kapasitas penyimpanan yang digunakan (untuk klaster yang disediakan): Status dan sumber daya sistem kueri yang digunakan.
 - Koneksi database aktif: DatabaseConnections metrik untuk kueri.

Untuk informasi selengkapnya tentang CloudWatch metrik, lihat [Data kinerja di Amazon Redshift](#).

Halaman kueri anak



Halaman kueri Anak memiliki komponen berikut:

- Pulldown kueri anak — Kontrol ini menunjukkan nama urutan dan waktu eksekusi untuk setiap kueri anak. Anda dapat menavigasi ke kueri anak lain dengan memilihnya di kontrol ini.
- Panel samping - Panel ini berisi tab untuk menampilkan aliran kueri anak dan teks kueri anak.
- Tab aliran kueri anak - Tab panel atas ini menampilkan yang berikut:
 - Streams - Panel ini menampilkan daftar aliran dalam kueri anak. Panel ini menampilkan informasi dan data agregat tentang aliran yang digunakan Amazon Redshift untuk

mengoptimalkan kueri. Untuk informasi tentang detail di panel ini, lihat [SYS_QUERY_DETAIL](#) di Panduan Pengembang Database Amazon Redshift.

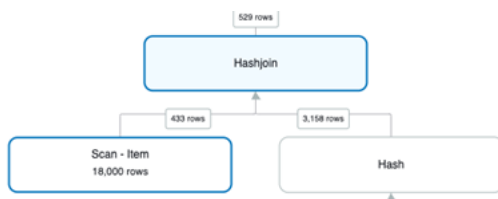
Anda dapat menambah atau menghapus kolom dari panel Streams menggunakan menu Preferensi. Anda mengakses menu Preferensi menggunakan ikon roda gigi di panel Streams.

- Panel kueri anak — Representasi grafis dari langkah-langkah dalam kueri anak. Untuk informasi tentang panel kueri Anak, lihat [Panel kueri anak](#) berikut.
- Tab teks kueri anak - Tab panel atas ini menunjukkan kode SQL untuk kueri anak.
- Panel detail kueri anak — Panel di panel kanan ini menampilkan detail tentang kueri anak. Untuk informasi tentang detail di panel ini, lihat [SYS_QUERY_DETAIL](#) di Panduan Pengembang Database Amazon Redshift.
- Panel detail streaming — Saat Anda memilih aliran di panel Streams, panel Detail Stream menampilkan informasi tentang aliran. Untuk informasi tentang detail di panel ini, lihat [SYS_QUERY_DETAIL](#) di Panduan Pengembang Database Amazon Redshift.
- Panel detail langkah — Bila Anda memilih langkah di panel Streams atau grafik dalam paket kueri Anak, panel Detail langkah menampilkan informasi tentang langkah tersebut, Untuk informasi tentang detail di panel ini, lihat [SYS_QUERY_DETAIL](#) di Panduan Pengembang Database Amazon Redshift. :

Panel kueri anak

Profiler kueri menampilkan kueri anak di panel kueri Anak sebagai representasi grafis dari langkah-langkah dalam kueri anak yang dipilih.

Panel kueri Child menampilkan urutan proses dan hubungan antar langkah-langkahnya. Misalnya, jika sebuah langkah menggabungkan output dari dua langkah lainnya, panel kueri Anak menunjukkan langkah sebagai simpul pohon dengan dua node yang dimasukkan ke dalamnya:



Panel kueri Anak tidak menampilkan aliran yang berisi langkah-langkah secara default. Untuk menampilkan aliran yang digunakan Amazon Redshift untuk secara logis mempartisi langkah-langkah dalam kueri turunan, pilih Lihat aliran. Saat Anda memilih Lihat aliran, panel kueri Anak menunjukkan langkah-langkah yang terkandung dalam aliran kueri.

Panel kueri Anak tidak menampilkan informasi segmen. Untuk melihat segmen untuk satu langkah, pilih langkahnya. Panel Detail langkah kemudian menunjukkan segmen untuk langkah tersebut.

Navigasi di panel kueri Anak

Di panel kueri Anak, Anda dapat memilih langkah-langkah untuk melihat informasi rinci tentang mereka. Anda juga dapat menggeser dan memperbesar ruang kerja untuk memvisualisasikan langkah-langkah dalam rencana kueri dengan lebih baik.

Anda dapat memilih node, menggeser, dan memperbesar paket kueri Child menggunakan metode berikut:

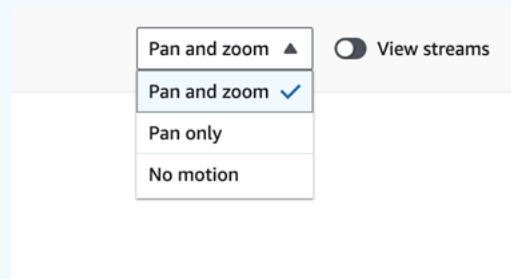
- Menggunakan mouse - Anda dapat memilih node, klik dan seret ruang kerja untuk menggeser, dan tahan **Ctrl** (Windows) atau **Cmd** (Mac) dan roda mouse untuk memperbesar. Memilih node menyebabkan ruang kerja memperbesar dan menggeser untuk menyorot node itu. Jika Anda memilih aliran di ruang kerja, aliran tersebut akan disorot dalam daftar Streams. Jika Anda memilih langkah di ruang kerja, panel Detail langkah menampilkan informasi tentang langkah itu.
- Menggunakan kontrol zoom dan fit di kiri atas ruang kerja - Kontrol ini memungkinkan Anda memperbesar, memperkecil, memperbesar agar sesuai dengan seluruh ruang kerja, dan masuk ke mode layar penuh. Saat Anda memperbesar agar sesuai dengan seluruh rencana kueri, ruang kerja memusatkan rencana kueri baik secara horizontal maupun vertikal.
- Menggunakan minimap di kanan bawah ruang kerja - Anda dapat menggeser atau memperbesar ruang kerja dengan menggunakan kontrol minimap di sudut kiri bawah ruang kerja.
- Memilih aliran di panel Streams — Jika Anda memilih aliran di panel Streams, ruang kerja akan menggeser dan memperbesar untuk menampilkan aliran yang dipilih, dan menampilkan informasi tentang aliran di panel Detail aliran.
- Memilih langkah di panel Streaming details — Jika Anda memilih langkah di panel Streaming details, ruang kerja akan menggeser dan memperbesar untuk menampilkan langkah yang dipilih, dan menampilkan informasi tentang langkah di panel Detail langkah.

Note

Saat Anda memilih langkah, baik di ruang kerja atau panel yang berbeda, ruang kerja mencoba memperbesar dan menggeser sedemikian rupa sehingga langkah yang dipilih paling terlihat.

Saat Anda memilih aliran atau langkah dari ruang kerja atau panel lain, ruang kerja hanya akan memperbesar dan menggeser ke aliran atau langkah itu jika Anda memilih Pan & Zoom

di kontrol kanan atas di ruang kerja. Anda dapat membatasi perilaku ini untuk panning dan zooming, panning saja, atau tidak ada gerakan dengan memilih pengaturan yang sesuai di pulldown.



Memecahkan masalah kueri dengan Query profiler

Jika Anda memecahkan masalah kueri, Anda dapat memilih kueri turunan untuk menentukan aliran mana yang menggunakan Persentase tertinggi dari total waktu kueri. Ini adalah cara cepat untuk menentukan bagian mana dari kueri Anda yang harus dianalisis lebih lanjut.

Setelah Anda mengetahui kueri anak mana yang paling banyak memakan waktu, lihat langkah-langkahnya untuk melihat gabungan atau pemindaian mana yang mungkin menyebabkan kinerja lambat.

Pemantauan Kueri dan Database

Dokumen ini menjelaskan halaman Query dan Database Monitoring, Konsol Manajemen AWS fitur untuk menganalisis kinerja klaster yang disediakan Amazon Redshift atau grup kerja tanpa server, dan kueri dijalankan terhadapnya.

Anda dapat menyelidiki skenario berikut menggunakan halaman Query dan Database Monitoring:

- Memantau metrik gudang data selama periode waktu tertentu
- Bagaimana kueri berkontribusi pada kinerja keseluruhan gudang data
- Lihat rincian waktu proses kueri ke dalam peristiwa siklus hidupnya seperti waktu tunggu kunci, waktu kompilasi, dan waktu eksekusi
- Pengguna mana yang menjalankan kueri paling intensif sumber daya selama periode waktu tertentu
- Pantau bagaimana peristiwa tambalan memengaruhi kinerja kueri

Topik

- [Izin](#)
- [Konsol Pemantauan Kueri dan Database](#)

Izin

Hak istimewa yang Akun AWS Anda gunakan untuk mengakses konsol memengaruhi kueri mana yang ditampilkan halaman Query dan Database Monitoring. Secara default, Anda hanya dapat melihat kueri Anda sendiri. Untuk melihat kueri yang dimiliki oleh pengguna lain, berikan SYS:MONITOR peran tersebut ke akun Anda. Untuk memungkinkan pengguna mengakhiri kueri yang berjalan dari halaman Query dan Database Monitoring, berikan SYS:OPERATOR hak istimewa kepada pengguna.

Untuk menetapkan sys:monitor peran secara otomatis ke pengguna IAM atau peran untuk Amazon Redshift Tanpa Server atau yang disediakan, jalankan perintah berikut:

```
create role monitor;  
grant role sys:monitor to role monitor;
```

Untuk memperbarui peran IAM yang digunakan untuk pemantauan kueri, lakukan hal berikut:

1. Pilih tab Tanda.
2. Pilih Kelola tanda.
3. Tambahkan tag dengan kunci **RedshiftDbRoles** dan nilai **monitor**.
4. Simpan perubahan

Untuk menambahkan kredensi database ke pengguna, jalankan perintah berikut:

```
grant role sys:monitor to <username>
```

Untuk informasi tentang menggunakan perintah GRANT, lihat [GRANT](#) di Panduan Pengembang Database Amazon Redshift.

Untuk menggunakan pemantauan kueri, pengguna IAM Anda memerlukan izin untuk mengakses bidang data Amazon Redshift. Pastikan bahwa pengguna IAM Anda memiliki izin berikut dalam kebijakan izin mereka:

```
{
  "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"
},
```

Kredensi sementara menggunakan identitas IAM Anda

Opsi ini hanya tersedia saat menghubungkan ke cluster. Dengan metode ini, Query and Database Monitoring memetakan nama pengguna ke identitas IAM Anda dan menghasilkan kata sandi sementara untuk terhubung ke database sebagai identitas IAM Anda. Seorang pengguna yang menggunakan metode ini untuk terhubung harus diizinkan izin IAM untuk `redshift:GetClusterCredentialsWithIAM`. Untuk mencegah pengguna menggunakan metode ini, ubah pengguna atau peran IAM mereka untuk menolak izin ini.

Konsol Pemantauan Kueri dan Database

Bagian ini menjelaskan penggunaan halaman konsol Query and Database Monitoring.

Anda dapat menggunakan konsol Query and Database Monitoring untuk mendapatkan gambaran umum kinerja gudang data Anda dengan cepat. Anda dapat memantau kinerja gudang data Anda dari waktu ke waktu, dan memeriksa kinerja klaster yang disediakan gudang data atau kueri individual, sehingga dapat mengidentifikasi kemacetan dan area lain untuk ditingkatkan dengan sebaik-baiknya.

Halaman Query dan Database Monitoring memiliki fitur berikut:

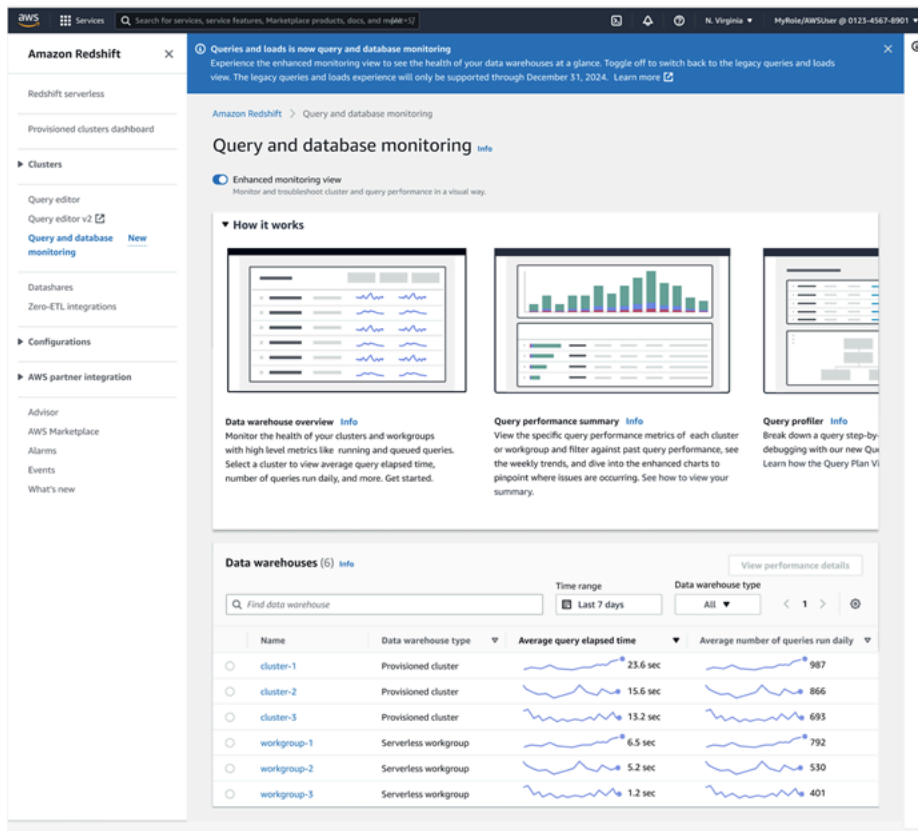
- Peningkatan keamanan — Anda memerlukan hak istimewa yang lebih tinggi untuk memantau kueri bagi pengguna lain. Untuk informasi selengkapnya, lihat [Izin](#).
- Riwayat kueri tujuh hari - Dijamin akses ke tujuh hari riwayat kueri
- Pemantauan kueri — Anda dapat memantau kueri dalam klaster yang disediakan dan grup kerja tanpa server di tingkat kueri pengguna.

- Analisis tren kueri — Anda dapat membandingkan kinerja kueri serupa yang sesuai dengan kriteria yang ditentukan.

Untuk mengakses halaman Query dan Database Monitoring, lakukan hal berikut:

1. Masuk ke konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Pilih Query dan pemantauan database dari panel navigasi.

Halaman konsol pemantauan Query dan database muncul sebagai berikut:



Halaman Query dan database monitoring memiliki komponen-komponen berikut:

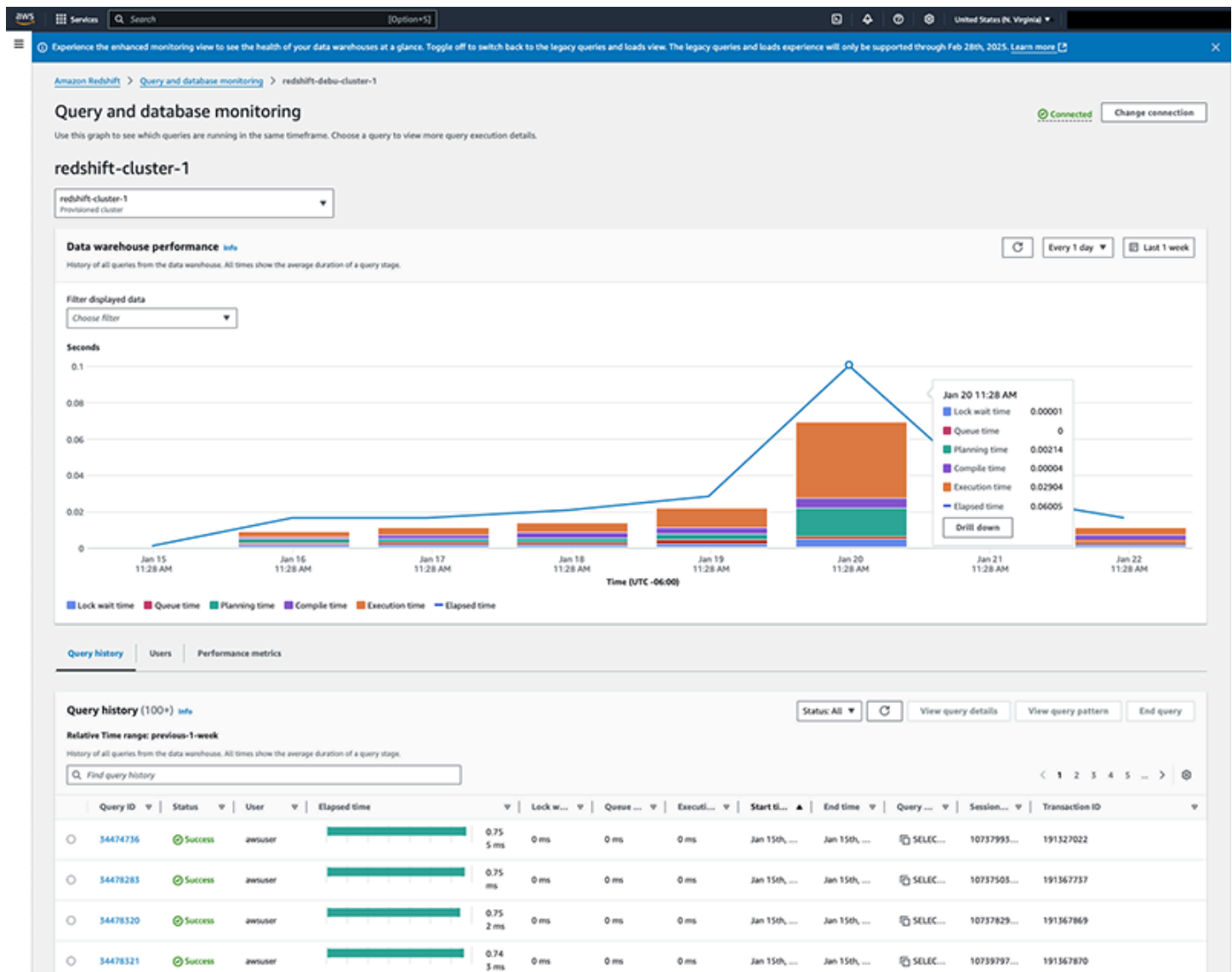
- Ikhtisar gudang data — Pantau kinerja kueri rata-rata untuk klaster yang disediakan dan grup kerja tanpa server Anda. Anda dapat dengan cepat melihat masalah kinerja untuk klaster atau grup kerja tertentu dengan memeriksa statistik di halaman ini untuk lonjakan atau periode aktivitas tinggi.
- Ringkasan kinerja kueri — Pantau kinerja kueri rata-rata untuk klaster tertentu yang disediakan atau grup kerja tanpa server. Anda juga dapat mengakses halaman ringkasan kinerja kueri dengan mengklik salah satu cluster atau kelompok kerja dalam daftar ikhtisar gudang data.

Topik

- [Ringkasan kinerja kueri](#)
- [Detail kueri](#)
- [Pola kueri](#)

Ringkasan kinerja kueri

Saat Anda memilih Ringkasan kinerja kueri dari halaman pemantauan Kueri dan database, atau kluster atau grup kerja dari halaman ikhtisar gudang data, konsol akan menampilkan ringkasan kinerja untuk kluster yang disediakan individu atau grup kerja tanpa server.



Halaman ini memiliki komponen-komponen berikut:

- Dropdown cluster atau workgroup - Pilih cluster atau workgroup yang ingin Anda analisis.
- Kinerja gudang data - Panel ini menampilkan riwayat klaster atau grup kerja dalam periode waktu yang ditentukan, menunjukkan jumlah waktu yang dihabiskan untuk setiap fase kueri. Jika Anda melihat lonjakan dalam fase kueri tertentu, seperti peningkatan waktu perencanaan dan eksekusi pada 20 Januari di grafik sebelumnya, Anda dapat menggunakan informasi ini untuk mengidentifikasi masalah dengan kinerja kueri Anda. Periode waktu default yang ditampilkan adalah tujuh hari terakhir, tetapi Anda dapat menyesuaikan periode waktu sesuai dengan kebutuhan analisis Anda.
- Riwayat kueri - Panel ini menampilkan riwayat kinerja setiap kueri yang dijalankan dalam filter yang ditentukan. Anda dapat menggunakan informasi ini untuk memecahkan masalah kinerja kueri individual. Untuk menganalisis kueri tertentu lebih lanjut, Anda dapat memilihnya dari grafik ini atau dari grafik kinerja gudang data.

Saat Anda mengarahkan kursor ke titik data pada bagan, munculan muncul. Popup ini menampilkan rincian metrik waktu untuk titik data tersebut. Jika titik data berisi data kueri, Anda dapat memilih Telusuri ke bawah untuk memperbarui rentang waktu bagan ke unit waktu terkecil berikutnya dari titik data. Satuan waktu ini adalah sebagai berikut:

- Hari
- Jam
- 15 menit
- 5 menit
- 1 menit

Misalnya, jika Anda memilih Telusuri titik data, rentang grafik berubah menjadi satu hari. Memilih Drill down lagi mengubah rentang grafik menjadi satu jam.

- Query profiler — Alat grafis untuk memantau kinerja kueri. Untuk informasi selengkapnya, lihat [Profilер kueri](#).

Detail kueri

Saat Anda memilih kueri dari panel Performa gudang data atau Riwayat kueri pada halaman detail klaster atau grup kerja, halaman Detail kueri akan terbuka.

316

- Analisis tren kinerja kueri — Grafik batang yang menunjukkan waktu berlalu dari semua kueri yang dipilih pola kueri. Hasilnya dikelompokkan berdasarkan hari.
- Sejarah kueri dengan pola yang sama — Waktu yang dibutuhkan untuk setiap fase kueri yang dipilih oleh pola kueri.

Dengan menggunakan halaman pola Query, Anda bisa mendapatkan wawasan berikut:

- Tren kueri yang berjalan pada waktu tertentu setiap hari
- Lonjakan waktu berjalan untuk kueri yang Anda jalankan secara teratur terhadap gudang data.

Kueri dan Pemantauan Database Berbasis Sys View

Dokumen ini menjelaskan SYS tampilan yang menyediakan data untuk halaman Kueri dan Pemantauan Database di Amazon Redshift Console, alat untuk menganalisis komponen dan kinerja kueri. Untuk informasi tentang halaman Query dan Database Monitoring, lihat [Pemantauan Kueri dan Database](#).

Halaman Query dan Database Monitoring memiliki fungsi yang menampilkan informasi yang disediakan oleh SYS tampilan. Tampilan konsol menyertakan profiler kueri, yang menunjukkan rencana eksekusi grafis kueri. Untuk beralih ke tampilan SYS berbasis, ikuti langkah-langkah di sini untuk memberikan akses dan izin yang tepat untuk halaman Query dan Database Monitoring yang baru.

Fitur tampilan SYS berbasis halaman Queries dan Database Monitoring memiliki fungsionalitas sebagai berikut:

- Peningkatan keamanan - Anda memerlukan hak istimewa yang lebih tinggi untuk memantau kueri bagi pengguna lain
- Riwayat kueri tujuh hari - Dijamin akses ke tujuh hari riwayat kueri
- Query profiler — Alat grafis untuk memantau kinerja kueri. Untuk informasi selengkapnya, lihat [Profiler kueri](#)

Secara default, Anda hanya dapat melihat kueri Anda sendiri. Untuk melihat kueri yang dimiliki oleh pengguna lain, berikan SYS:MONITOR peran tersebut ke akun Anda. Untuk memungkinkan pengguna mengakhiri kueri yang sedang berjalan, berikan SYS:OPERATOR hak istimewa kepada pengguna.

Untuk memberikan hak istimewa untuk melihat kueri yang dimiliki oleh semua pengguna kepada pengguna atau peran database, jalankan perintah berikut:

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

Untuk menetapkan `sys:monitor` peran secara otomatis ke pengguna IAM atau peran untuk Amazon Redshift Tanpa Server atau yang disediakan, jalankan perintah berikut:

```
create role monitor;
grant role sys:monitor to role monitor;
```

Untuk memperbarui peran IAM yang digunakan untuk pemantauan kueri, lakukan hal berikut:

1. Pilih tab Tanda.
2. Pilih Kelola tanda.
3. Tambahkan tag dengan kunci **RedshiftDbRoles** dan nilai **monitor**.
4. Simpan perubahan

Untuk menambahkan kredensi database ke pengguna, jalankan perintah berikut:

```
grant role sys:monitor to <username>
```

Izin

Untuk menggunakan pemantauan kueri, pengguna IAM Anda memerlukan izin untuk mengakses bidang data Amazon Redshift. Pastikan bahwa pengguna IAM Anda memiliki izin berikut dalam kebijakan izin mereka:

```
{
  "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"
},
```

Connect ke database

Sebelum menggunakan fitur pemantauan kueri yang disempurnakan, Anda harus terlebih dahulu terhubung ke database Anda untuk akses ke informasi SYS berbasis tampilan. Untuk terhubung ke database, gunakan salah satu kredensi berikut:

- Nama pengguna dan kata sandi
- Kredensi sementara yang terkait dengan peran IAM Anda
- Seorang pengguna basis data

Perhatikan hal berikut tentang penggunaan pemantauan kueri yang disempurnakan:

- Untuk kluster yang disediakan, Anda harus terhubung ke database, karena pemantauan kueri yang disempurnakan menggunakan tampilan. SYS Pandangan ini telah meningkatkan keamanan, dan memerlukan hak istimewa yang lebih tinggi untuk mengakses data tentang kueri yang dimiliki oleh pengguna lain.
- Saat menggunakan kueri SYS berbasis tampilan dan halaman pemantauan basis data, hanya halaman Anda `user_id` yang terlihat jika akun pengguna Anda tidak memiliki peran superuser database. Nama pengguna disembunyikan dari non-pengguna super.
- Sebagai bagian dari kueri berbasis tampilan sys dan pengalaman halaman pemantauan database, ID proses eksekusi kueri (`p_id`) muncul di bawah judul kolom. `session_id`

Trek untuk kluster yang disediakan Amazon Redshift dan grup kerja tanpa server

Saat Amazon Redshift merilis versi baru, Amazon Redshift akan memperbarui versi gudang data Amazon Redshift Anda (grup kerja tanpa server atau kluster yang disediakan). Anda dapat mengontrol apakah gudang data Anda diperbarui ke rilis terbaru atau ke rilis bersertifikat sebelumnya.

Grup kerja tanpa server atau trek kluster yang disediakan menentukan versi rilis mana yang diterapkan selama pembaruan versi. Amazon Redshift memperbarui kluster yang disediakan selama jendela pemeliharaan yang ditentukan, dan biasanya memperbarui grup kerja tanpa server selama periode idle. Untuk detail tentang kapan Redshift Serverless memperbarui workgroup, lihat.

[Memperbarui grup kerja tanpa server](#)

Saat Amazon Redshift merilis versi baru, versi tersebut ditetapkan ke trek saat ini, dan versi sebelumnya ditetapkan ke trailing track. Untuk mengatur trek untuk gudang data Anda, tentukan salah satu nilai berikut:

- Saat ini - Dengan trek saat ini, Anda mendapatkan versi rilis paling up-to-date bersertifikat dengan fitur terbaru, pembaruan keamanan, dan peningkatan kinerja.
- Trailing — Dengan Trailing track, Anda akan berada di rilis bersertifikat sebelumnya.

Misalnya, workgroup tanpa server Anda saat ini menjalankan versi 1.0.2762 dan Amazon Redshift merilis Redshift Serverless versi 1.0.3072. Jika nilai track Anda Current, workgroup Anda diperbarui ke versi 1.0.3072 (rilis terbaru). Jika Anda menyetel nilai track ke Trailing, workgroup Anda akan diperbarui saat versi trailing track berikutnya dirilis.

Dengan fitur trailing track, Anda memiliki opsi untuk menjalankan subset gudang data Amazon Redshift di trailing track. Hal ini memungkinkan untuk 1-6 minggu pengujian dan validasi integrasi pada gudang data yang disetel ke trek Saat ini sebelum menerapkan rilis ke gudang data di Trailing track. Secara default, Amazon Redshift membuat semua cluster dan grup kerja di trek Saat Ini untuk memanfaatkan rilis tersertifikasi yang paling banyak up-to-date. Namun, menggunakan trailing track Amazon Redshift di lingkungan produksi Anda, dan trek saat ini di lingkungan pengujian dan pengembangan Anda, memberi Anda ketekunan dan waktu tambahan untuk mengevaluasi rilis terbaru. Trailing track memastikan stabilitas maksimum, membuatnya ideal untuk beban kerja yang sangat penting dalam lingkungan produksi.

Note

Versi trailing track mungkin sama dengan versi trek saat ini untuk waktu yang singkat. Ini terjadi ketika trek saat ini belum maju ke versi berikutnya. Biasanya, versi trek saat ini berada di depan versi trailing track.

Beralih antar trek

Mengubah trek untuk sumber daya Amazon Redshift umumnya merupakan keputusan satu kali. Anda harus berhati-hati dalam mengubah trek. Untuk informasi tentang fitur mana yang ada di versi gudang data, lihat [Versi cluster untuk Amazon Redshift](#).

Jika Anda mengubah trek dari Trailing ke Current, kami akan memperbarui gudang data ke versi rilis trek saat ini. Jika Anda mengubah track data warehouse menjadi Trailing, kami akan memperbarui data warehouse Anda sebagai berikut:

- Untuk grup kerja tanpa server, kami memperbarui versi gudang data Anda selama periode idle. Untuk detail selengkapnya tentang cara Redshift Serverless memperbarui versi workgroup Anda, lihat [Memperbarui grup kerja tanpa server](#)
- Untuk kluster yang disediakan, kami tidak akan memperbarui gudang data Anda hingga ada rilis baru setelah versi rilis trek saat ini.

Melacak dan memulihkan

Untuk grup kerja tanpa server, snapshot mewarisi trek gudang data Amazon Redshift target. Misalnya, jika Anda membuat snapshot untuk grup kerja yang disetel ke Trailing track, dan menerapkan snapshot tersebut ke grup kerja yang disetel ke Track saat ini, workgroup akan memiliki setelan trek Current.

Untuk klaster yang disediakan, snapshot mewarisi trek gudang data Amazon Redshift sumber. Jika Anda mengubah trek gudang data sumber setelah mengambil snapshot, snapshot dan gudang data sumber berada di trek yang berbeda. Saat Anda memulihkan dari snapshot, gudang data baru akan berada di trek yang diwarisi dari sumber snapshot. Anda dapat mengubah trek setelah operasi pemulihan selesai.

Mengubah ukuran gudang data tidak memengaruhi lintasannya.

Memperbarui grup kerja tanpa server

Saat versi baru tersedia untuk trek yang dipilih grup kerja, Amazon Redshift Serverless biasanya menerapkan pembaruan selama periode idle selama tidak ada permintaan pembaruan trek yang tertunda. Jika workgroup tidak mengalami periode idle dalam 14 hari, Redshift Serverless memaksa pembaruan versi.

Redshift Tanpa Server hanya memperbarui workgroup Anda ke versi berikutnya yang lebih tinggi. Redshift Tanpa Server tidak melewati versi perantara atau grup kerja downgrade, meskipun versi untuk trek grup kerja yang dipilih kurang dari versi grup kerja saat ini. Workgroup Anda tidak akan menerima upgrade versi utama sampai dan kecuali *Trailing* trek menyusul.

Sebagai contoh, misalkan *Current* trek adalah versi 186, dan versi *Trailing* trek adalah 185. Jika Anda memiliki workgroup yang memiliki *Track* nilai *Current*, yang versinya 186, jika Anda mengubah *Track* nilainya, *Trailing* Redshift Serverless tidak akan menurunkan versi workgroup ke 185. Dalam skenario ini, Redshift Serverless menjaga workgroup pada versi 186 hingga versi *Trailing* track sama dengan atau lebih tinggi dari 186.

Jika perubahan trek tertunda, Redshift Serverless tidak memperbarui workgroup ke versi utama berikutnya di trek yang ada hingga menerapkan perubahan trek. Setelah perubahan trek selesai, Redshift Serverless mengevaluasi kondisi untuk memperbarui workgroup ke versi yang sesuai di bawah trek baru.

Misalnya, jika workgroup Anda disetel ke *Current* track, dan track saat ini adalah 186, dan Anda mengubah workgroup Anda ke *trackTrailing*, Redshift Serverless tidak akan memperbarui workgroup sampai setelah menerapkan perubahan trek, dan setelah versi diperbarui ke *Trailing* versi yang sama dengan atau lebih tinggi dari 186.

Note

Setiap operasi yang ada pada workgroup, seperti memulihkan dari snapshot, mengubah kunci KMS, atau mengubah ukuran, hanya terjadi pada trek yang ada. Redshift Serverless tidak menggunakan track yang tertunda untuk operasi tanpa server.

Jika Anda memiliki permintaan sakelar trek yang tertunda, Anda dapat membatalkan permintaan dengan menyetel `track` parameter kembali ke nilai aslinya menggunakan [UpdateWorkgroup](#).

Mengelola versi

Lagu adalah serangkaian rilis. Anda dapat memutuskan apakah gudang data Amazon Redshift Anda berada di trek saat ini atau trek Trailing. Jika Anda menempatkan gudang data Anda di trek saat ini, itu akan selalu ditingkatkan ke versi rilis terbaru. Jika Anda meletakkan sumber daya Anda di Trailing track, itu akan selalu menjalankan versi rilis yang dirilis segera sebelum versi yang paling baru dirilis.

Untuk kluster yang disediakan, kolom Status rilis di daftar konsol Amazon Redshift dari gudang data Amazon Redshift menunjukkan apakah salah satu sumber daya Anda tersedia untuk ditingkatkan.

Menentukan versi workgroup atau cluster

Anda dapat menentukan versi workgroup tanpa server Amazon Redshift atau engine versi cluster yang disediakan dengan konsol Amazon Redshift.

Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>

Serverless workgroups

Untuk grup kerja tanpa server, pada menu navigasi, pilih Workgroups, lalu pilih nama workgroup dari daftar untuk membuka detailnya. Rincian workgroup ditampilkan.

Provisioned clusters

Untuk kluster yang disediakan, pada menu navigasi, pilih Cluster, lalu pilih nama cluster dari daftar untuk membuka detailnya.

Rincian cluster ditampilkan, yang dapat mencakup kinerja Cluster, pemantauan Kueri, Database, Datashares, Jadwal, Pemeliharaan, dan tab Properti. Pilih tab Pemeliharaan untuk detail selengkapnya.

Di bagian Pemeliharaan, temukan Versi cluster saat ini.

Note

Untuk cluster yang disediakan, konsol menampilkan informasi versi dalam satu bidang, tetapi ada dua parameter di Amazon Redshift API. Parameter ini adalah `ClusterVersion` dan `ClusterRevisionNumber`. Untuk informasi selengkapnya, lihat [Cluster](#) di Referensi Amazon Redshift API.

Integrasi nol-ETL

Integrasi Zero-ETL adalah solusi yang dikelola sepenuhnya yang membuat data transaksional dan operasional tersedia di Amazon Redshift dari berbagai sumber operasional dan transaksional. Dengan solusi ini, Anda dapat mengonfigurasi integrasi dari sumber Anda ke gudang data Amazon Redshift. Anda tidak perlu memelihara pipeline ekstrak, transformasi, dan beban (ETL). Kami menangani ETL untuk Anda dengan mengotomatiskan pembuatan dan pengelolaan replikasi data dari sumber data ke cluster Amazon Redshift atau namespace Redshift Tanpa Server. Anda dapat terus memperbarui dan menanyakan data sumber Anda sekaligus menggunakan Amazon Redshift untuk beban kerja analitik, seperti pelaporan dan dasbor.

Dengan integrasi Zero-ETL, Anda memiliki data yang lebih segar untuk analitik, AI/ML, dan pelaporan. Anda mendapatkan wawasan yang lebih akurat dan tepat waktu untuk kasus penggunaan seperti dasbor bisnis, pengalaman bermain game yang dioptimalkan, pemantauan kualitas data, dan analisis perilaku pelanggan. Anda dapat membuat prediksi berbasis data dengan lebih percaya diri, meningkatkan pengalaman pelanggan, dan mempromosikan wawasan berbasis data di seluruh bisnis.

Sumber-sumber berikut saat ini didukung untuk integrasi nol-ETL:

- 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
- Aplikasi termasuk Salesforce, Salesforce Marketing Cloud Account Engagement, SAP, iklan Instagram, iklan ServiceNow Meta, dan Zendesk
- MySQL, PostgreSQL, SQL Server, dan Oracle yang Dikelola Sendiri

Untuk membuat integrasi nol-ETL, Anda menentukan sumber integrasi dan gudang data Amazon Redshift sebagai target. Setelah pemuatan data awal, integrasi mereplikasi data dari sumber ke gudang data target. Data menjadi tersedia di Amazon Redshift. Anda mengontrol enkripsi data saat membuat sumber integrasi, saat membuat integrasi nol-ETL, dan saat membuat gudang data

Amazon Redshift. Integrasi memantau kondisi pipeline data dan memulihkan dari masalah jika memungkinkan. Anda dapat membuat integrasi dari sumber dengan jenis yang sama ke dalam satu gudang data Amazon Redshift untuk memperoleh wawasan holistik di beberapa aplikasi.

Dengan data di Amazon Redshift, Anda dapat menggunakan analitik yang disediakan Amazon Redshift. Misalnya, pembelajaran mesin bawaan (ML), tampilan terwujud, berbagi data, dan akses langsung ke beberapa penyimpanan data dan data lake. Untuk insinyur data, integrasi nol-ETL menyediakan akses ke data sensitif waktu yang jika tidak dapat ditunda oleh kesalahan intermiten dalam jaringan data yang kompleks. Anda dapat menjalankan kueri analitis dan model ML pada data transaksional untuk mendapatkan wawasan tepat waktu untuk peristiwa dan keputusan bisnis yang sensitif terhadap waktu.

Anda dapat membuat langganan notifikasi peristiwa Amazon Redshift sehingga Anda dapat diberi tahu saat peristiwa terjadi untuk integrasi nol-ETL tertentu. Untuk melihat daftar notifikasi peristiwa terkait integrasi, lihat [Pemberitahuan acara integrasi nol-ETL dengan Amazon EventBridge](#). Cara termudah untuk membuat langganan adalah dengan konsol Amazon SNS. Untuk informasi tentang cara membuat topik Amazon SNS dan berlangganan topik tersebut, lihat [Memulai Amazon SNS](#) di Panduan Pengembang Layanan Pemberitahuan Sederhana Amazon.

Saat mulai menggunakan integrasi nol-ETL, pertimbangkan konsep berikut ini:

- Database sumber adalah database dari mana data direplikasi ke Amazon Redshift.
- Gudang data target adalah klaster yang disediakan Amazon Redshift atau grup kerja Redshift Serverless tempat data direplikasi.
- Database tujuan adalah database yang Anda buat dari integrasi nol-ETL di gudang data target.

Untuk informasi tentang tabel dan tampilan sistem yang dapat Anda gunakan untuk memantau integrasi nol-ETL Anda, lihat [Memantau integrasi nol-ETL dengan tampilan sistem Amazon Redshift](#)

Untuk daftar yang didukung Wilayah AWS oleh setiap sumber untuk integrasi nol-ETL, lihat [Wilayah yang Didukung untuk integrasi nol-ETL](#)

Untuk informasi harga untuk integrasi nol-ETL, lihat halaman harga yang sesuai:

- [Harga Amazon Redshift](#)
- [Harga Amazon Aurora](#)
- [Harga Amazon RDS](#)
- [Harga Amazon DynamoDB](#)

- [AWS Glue harga](#)

Untuk informasi selengkapnya tentang sumber integrasi nol-ETL, lihat topik berikut:

- Untuk integrasi Aurora Zero-ETL, lihat [Manfaat](#), [Konsep Utama](#), [Batasan](#), [Kuota](#), dan Wilayah yang [Didukung dari integrasi Nol-ETL](#) di Panduan Pengguna Amazon Aurora.
- Untuk integrasi RDS nol-ETL, lihat [Manfaat](#), [Konsep Utama](#), [Batasan](#), [Kuota](#), dan Wilayah yang [Didukung dari integrasi Nol-ETL di Panduan Pengguna](#) Amazon RDS.
- Untuk integrasi DynamoDB Zero-ETL, lihat Integrasi DynamoDB [Zero-ETL dengan Amazon Redshift](#) di Panduan Pengembang Amazon DynamoDB.
- [Untuk integrasi nol-ETL dengan aplikasi, lihat Integrasi nol-ETL di Panduan Pengembang.AWS Glue](#)

Topik

- [Pertimbangan saat menggunakan integrasi nol-ETL dengan Amazon Redshift](#)
- [Mulai menggunakan integrasi nol-ETL](#)
- [Membuat database tujuan di Amazon Redshift](#)
- [Menanyakan data yang direplikasi di Amazon Redshift](#)
- [Melihat integrasi nol-ETL](#)
- [Modus sejarah](#)
- [Berbagi data Anda di Amazon Redshift](#)
- [Memantau integrasi nol-ETL](#)
- [Metrik untuk integrasi nol-ETL](#)
- [Memodifikasi integrasi nol-ETL untuk DynamoDB](#)
- [Hapus integrasi nol-ETL untuk DynamoDB](#)
- [Wilayah yang Didukung untuk integrasi nol-ETL](#)
- [Memecahkan masalah integrasi nol-ETL](#)

Pertimbangan saat menggunakan integrasi nol-ETL dengan Amazon Redshift

Pertimbangan berikut berlaku untuk integrasi nol-ETL dengan Amazon Redshift.

- Gudang data Amazon Redshift target Anda harus memenuhi prasyarat berikut:
 - Menjalankan Amazon Redshift Tanpa Server atau tipe node. RA3
 - Terenkripsi (jika menggunakan klaster yang disediakan).
 - Memiliki sensitivitas kasus diaktifkan.
- Jika Anda menghapus sumber yang merupakan sumber integrasi resmi untuk gudang data Amazon Redshift, semua integrasi terkait akan masuk ke status. FAILED Data yang direplikasi sebelumnya tetap ada di database Amazon Redshift Anda dan dapat ditanyakan.
- Database tujuan adalah read-only. Anda tidak dapat membuat tabel, tampilan, atau tampilan terwujud dalam database tujuan. Namun, Anda dapat menggunakan tampilan terwujud pada tabel lain di gudang data target.
- Tampilan terwujud didukung saat digunakan dalam kueri lintas basis data. Untuk informasi tentang membuat tampilan terwujud dengan data yang direplikasi melalui integrasi nol-ETL, lihat [Menanyakan data yang direplikasi dengan tampilan terwujud](#)
- Secara default, Anda dapat menanyakan tabel hanya di gudang data target yang berada dalam Synced status. Untuk menanyakan tabel di negara bagian lain, atur parameter database QUERY_ALL_STATES keTRUE. Untuk informasi tentang pengaturanQUERY_ALL_STATES, lihat [MEMBUAT DATABASE](#) dan [MENGUBAH DATABASE](#) di Panduan Pengembang Database Amazon Redshift. Untuk informasi selengkapnya tentang status database Anda, lihat [SVV_INTEGRATION_TABLE_STATE](#) di Panduan Pengembang Database Amazon Redshift.
- Amazon Redshift hanya menerima karakter UTF-8, jadi mungkin tidak menghormati pemeriksaan yang ditentukan dalam sumber Anda. Aturan penyortiran dan perbandingan mungkin berbeda, yang pada akhirnya dapat mengubah hasil kueri.
- Integrasi nol-ETL dibatasi hingga 50 per target gudang data Amazon Redshift.
- Tabel dalam sumber integrasi harus memiliki kunci utama. Jika tidak, tabel Anda tidak dapat direplikasi ke gudang data target di Amazon Redshift.

Untuk informasi tentang cara menambahkan kunci utama ke Amazon Aurora PostgreSQL, lihat [Menangani tabel tanpa kunci utama saat membuat integrasi Amazon Aurora PostgreSQL Zero-ETL dengan Amazon Redshift di Blog Database.AWS](#) Untuk informasi tentang cara menambahkan kunci utama ke Amazon Aurora MySQL atau RDS untuk MySQL, lihat Menangani tabel [tanpa kunci utama saat membuat Amazon Aurora MySQL atau Amazon RDS untuk MySQL for MySQL atau integrasi Nol-ETL MySQL dengan Amazon Redshift di Blog Database.AWS](#)

- Anda dapat menggunakan pemfilteran data untuk integrasi Aurora Zero-ETL untuk menentukan ruang lingkup replikasi dari cluster Aurora DB sumber ke gudang data Amazon Redshift target. Daripada mereplikasi semua data ke target, Anda dapat menentukan satu atau lebih filter yang

secara selektif menyertakan atau mengecualikan tabel tertentu agar tidak direplikasi. Untuk informasi selengkapnya, lihat [Pemfilteran data untuk integrasi Aurora Zero-ETL dengan Amazon Redshift](#) di Panduan Pengguna Amazon Aurora.

- Untuk integrasi Aurora PostgreSQL Zero-ETL dengan Amazon Redshift, Amazon Redshift mendukung maksimum 100 database dari Aurora PostgreSQL. Setiap database mereplikasi dari sumber ke target secara independen.
- Integrasi nol-ETL tidak mendukung transformasi saat mereplikasi data dari penyimpanan data transaksional ke Amazon Redshift. Data direplikasi apa adanya dari basis data sumber. Namun, Anda dapat menerapkan transformasi pada data yang direplikasi di Amazon Redshift.
- Integrasi nol-ETL berjalan di Amazon Redshift menggunakan koneksi paralel. Ini berjalan menggunakan kredensial pengguna yang membuat database dari integrasi. Saat kueri berjalan, penskalaan konkurensi tidak berlaku untuk koneksi ini selama sinkronisasi (tuliskan). Pembacaan penskalaan konkurensi (dari klien Amazon Redshift) berfungsi untuk objek yang disinkronkan.
- Anda dapat mengatur integrasi nol-ETL untuk mengontrol frekuensi replikasi data ke Amazon Redshift. `REFRESH_INTERVAL` Untuk informasi selengkapnya, lihat [MEMBUAT DATABASE](#) dan [MENGUBAH DATABASE](#) di Panduan Pengembang Database Amazon Redshift.
- Setelah Anda membuat database Amazon Redshift dari integrasi nol-ETL dengan Amazon DynamoDB, status database akan berubah dari Membuat ke Aktif. Ini memulai replikasi data dalam tabel DynamoDB sumber ke tabel Redshift target, yang dibuat di bawah skema publik database tujuan (). `ddb_rs_customerprofiles_zet1_db`

Pertimbangan saat menggunakan mode riwayat pada target

Pertimbangan berikut berlaku saat menggunakan mode riwayat pada database target. Untuk informasi selengkapnya, lihat [Modus sejarah](#).

- Saat Anda menjatuhkan tabel pada sumber, tabel pada target tidak dijatuhkan, tetapi diubah menjadi `DroppedSource` status. Anda dapat menjatuhkan atau mengganti nama tabel dari database Amazon Redshift.
- Saat Anda memotong tabel pada sumber, penghapusan dijalankan pada tabel target. Misalnya, jika semua catatan terpotong pada sumber, catatan yang sesuai pada kolom `_record_is_active` target diubah menjadi `false`.
- Saat Anda menjalankan tabel `TRUNCATE SQL` pada tabel target, baris riwayat aktif ditandai tidak aktif dengan stempel waktu yang sesuai.

- Ketika baris dalam tabel diatur ke tidak aktif, itu dapat dihapus setelah penundaan singkat (sekitar 10 menit). Untuk menghapus baris yang tidak aktif, sambungkan ke database nol-ETL Anda dengan editor kueri v2 atau klien SQL lainnya.
- Anda hanya dapat menghapus baris tidak aktif dari tabel dengan mode riwayat aktif. Misalnya, perintah SQL yang mirip dengan berikut ini hanya menghapus baris yang tidak aktif.

```
delete from schema.user_table where _record_delete_time <= '2024-09-10 12:34:56'
```

Ini setara dengan perintah SQL seperti berikut ini.

```
delete from schema.user_table where _record_delete_time <= '2024-09-10 12:34:56' and  
_record_is_active = False
```

- Saat mematikan mode riwayat untuk tabel, semua data historis disimpan ke tabel yang diberi nama `<schema>.<table-name>_historical_<timestamp>` sementara tabel asli bernama `<schema>.<table-name>` disegarkan.
- Ketika tabel dengan mode riwayat aktif dikecualikan dari replikasi menggunakan filter tabel, semua baris ditetapkan sebagai tidak aktif dan diubah menjadi DroppedSource status. Untuk informasi selengkapnya tentang filter tabel, lihat [Pemfilteran data untuk integrasi Aurora Zero-ETL dengan Amazon Redshift](#) di Panduan Pengguna Amazon Aurora.
- Mode riwayat hanya dapat dialihkan ke `true` atau `false` untuk tabel dalam Synced keadaan.
- Tampilan terwujud untuk tabel dengan mode riwayat aktif dibuat sebagai perhitungan ulang penuh.

Pertimbangan ketika sumber integrasi nol-ETL adalah Aurora atau Amazon RDS

Pertimbangan berikut berlaku untuk integrasi Aurora dan Amazon RDS Zero-ETL dengan Amazon Redshift.

- Anda dapat menggunakan pemfilteran data untuk Aurora dan RDS untuk integrasi MySQL Zero-ETL untuk menentukan ruang lingkup replikasi dari cluster DB sumber ke gudang data Amazon Redshift target. Daripada mereplikasi semua data ke target, Anda dapat menentukan satu atau lebih filter yang secara selektif menyertakan atau mengecualikan tabel tertentu agar tidak direplikasi. Untuk informasi selengkapnya, lihat [Pemfilteran data untuk integrasi Aurora Zero-ETL dengan Amazon Redshift](#) di Panduan Pengguna Amazon Aurora.

- Tabel dalam sumber integrasi harus memiliki kunci utama. Jika tidak, tabel Anda tidak dapat direplikasi ke gudang data target di Amazon Redshift.

Untuk informasi tentang cara menambahkan kunci utama ke Amazon Aurora PostgreSQL, lihat [Menangani tabel tanpa kunci utama saat membuat integrasi Amazon Aurora PostgreSQL Zero-ETL dengan Amazon Redshift di Blog Database.AWS](#) Untuk informasi tentang cara menambahkan kunci utama ke Amazon Aurora MySQL atau RDS untuk MySQL, lihat Menangani tabel [tanpa kunci utama saat membuat Amazon Aurora MySQL atau Amazon RDS untuk MySQL for MySQL atau integrasi Nol-ETL MySQL dengan Amazon Redshift di Blog Database.AWS](#)

- Panjang maksimum tipe data Amazon Redshift VARCHAR adalah 65.535 byte. Ketika konten dari sumber tidak sesuai dengan batas ini, replikasi tidak dilanjutkan dan tabel dimasukkan ke dalam keadaan gagal. Anda dapat mengatur parameter database TRUNCATECOLUMNS TRUE untuk memotong konten agar sesuai dengan kolom. Untuk informasi tentang pengaturan TRUNCATECOLUMNS, lihat [MEMBUAT DATABASE](#) dan [MENGUBAH DATABASE](#) di Panduan Pengembang Database Amazon Redshift.

Untuk informasi selengkapnya tentang perbedaan tipe data antara sumber integrasi nol-ETL dan database Amazon Redshift, lihat Perbedaan [tipe data antara Aurora dan Amazon Redshift di Panduan Pengguna Amazon Aurora](#).

Untuk sumber Aurora, lihat juga [Batasan di Panduan](#) Pengguna Amazon Aurora.

Untuk sumber Amazon RDS, lihat juga [Batasan](#) dalam Panduan Pengguna Amazon RDS.

Pertimbangan ketika sumber integrasi nol-ETL adalah DynamoDB

Pertimbangan berikut berlaku untuk integrasi DynamoDB Zero-ETL dengan Amazon Redshift.

- Nama tabel dari DynamoDB lebih besar dari 127 karakter tidak didukung.
- Data dari integrasi DynamoDB Zero-ETL memetakan ke kolom tipe data SUPER di Amazon Redshift.
- Nama kolom untuk kunci partisi atau kunci sortir yang lebih besar dari 127 karakter tidak didukung.
- Integrasi nol-ETL dari DynamoDB dapat memetakan ke hanya satu database Amazon Redshift.
- Untuk kunci partisi dan sortir, presisi dan skala maksimum adalah (38,18). Tipe data numerik pada DynamoDB mendukung presisi maksimum hingga 38. Amazon Redshift juga mendukung presisi maksimum 38, tetapi desimal default di Amazon precision/scale Redshift adalah (38,10). Itu berarti nilai nilai skala dapat terpotong.

- Untuk integrasi nol-ETL yang berhasil, atribut individual (terdiri dari nama+nilai) dalam item DynamoDB, tidak boleh lebih besar dari 64 KB.
- Pada aktivasi, integrasi nol-ETL mengeksport tabel DynamoDB lengkap untuk mengisi database Amazon Redshift. Waktu yang dibutuhkan untuk proses awal ini untuk menyelesaikan tergantung pada ukuran tabel DynamoDB. Integrasi nol-ETL kemudian secara bertahap mereplikasi pembaruan dari DynamoDB ke Amazon Redshift menggunakan ekspor tambahan DynamoDB. Ini berarti data DynamoDB yang direplikasi di Amazon Redshift disimpan secara otomatis. up-to-date

Saat ini, latensi minimum untuk integrasi DynamoDB Zero-ETL adalah 15 menit. Anda dapat meningkatkannya lebih lanjut dengan menetapkan non-nol REFRESH_INTERVAL untuk integrasi nol-ETL. Untuk informasi selengkapnya, lihat [MEMBUAT DATABASE](#) dan [MENGUBAH DATABASE](#) di Panduan Pengembang Database Amazon Redshift.

Untuk sumber Amazon DynamoDB, [lihat juga Prasyarat dan batasan](#) di Panduan Pengembang Amazon DynamoDB.

Pertimbangan ketika sumber integrasi nol-ETL adalah aplikasi, seperti, Salesforce, SAP,, dan Zendesk ServiceNow

Pertimbangan berikut berlaku untuk sumber adalah aplikasi, seperti, Salesforce, SAP ServiceNow, dan Zendesk dengan Amazon Redshift.

- Nama tabel dan nama kolom dari sumber aplikasi yang lebih besar dari 127 karakter tidak didukung.
- Panjang maksimum tipe data Amazon Redshift VARCHAR adalah 65.535 byte. Ketika konten dari sumber tidak sesuai dengan batas ini, replikasi tidak dilanjutkan dan tabel dimasukkan ke dalam keadaan gagal. Anda dapat mengatur parameter database TRUNCATECOLUMNS TRUE untuk memotong konten agar sesuai dengan kolom. Untuk informasi tentang pengaturan, TRUNCATECOLUMNS lihat [MEMBUAT DATABASE](#) dan [MENGUBAH DATABASE](#) di Panduan Pengembang Database Amazon Redshift.

[Untuk informasi selengkapnya tentang perbedaan tipe data antara sumber aplikasi integrasi nol-ETL dan database Amazon Redshift, lihat Integrasi nol-ETL di Panduan Pengembang.AWS Glue](#)

- Latensi minimum untuk integrasi nol-ETL dengan aplikasi adalah 1 jam. Anda dapat meningkatkannya lebih lanjut dengan menetapkan non-nol REFRESH_INTERVAL untuk integrasi nol-ETL. Untuk informasi selengkapnya, lihat [MEMBUAT DATABASE](#) dan [MENGUBAH DATABASE](#) di Panduan Pengembang Database Amazon Redshift.

Untuk sumber integrasi nol-ETL dengan aplikasi, lihat juga integrasi [nol-ETL di Panduan Pengembangan](#).AWS Glue

Mulai menggunakan integrasi nol-ETL

Kumpulan tugas ini memandu Anda melalui pengaturan integrasi nol-ETL pertama Anda. Pertama, Anda mengonfigurasi sumber integrasi Anda dan mengaturnya dengan parameter dan izin yang diperlukan. Kemudian, Anda melanjutkan ke sisa pengaturan awal dari konsol Amazon Redshift atau. AWS CLI Konsol menyediakan opsi Perbaiki untuk saya untuk memperbaiki beberapa masalah konfigurasi.

Topik

- [Membuat dan mengonfigurasi gudang data Amazon Redshift target](#)
- [Aktifkan sensitivitas kasus untuk gudang data Anda](#)
- [Konfigurasi otorisasi untuk gudang data Amazon Redshift Anda](#)
- [Buat integrasi nol-ETL](#)

Membuat dan mengonfigurasi gudang data Amazon Redshift target

Pada langkah ini, Anda membuat dan mengonfigurasi gudang data Amazon Redshift target, seperti workgroup Redshift Tanpa Server atau klaster yang disediakan. Jika Anda sudah memiliki gudang data Amazon Redshift yang dikonfigurasi untuk digunakan dengan integrasi nol-ETL, Anda dapat melewati langkah ini.

Gudang data target Anda harus memiliki karakteristik sebagai berikut:

- Menjalankan Amazon Redshift Tanpa Server atau klaster tipe node yang disediakan. RA3
- Memiliki sensitivitas kasus (`enable_case_sensitive_identifier`) dihidupkan. Untuk informasi selengkapnya, lihat [Aktifkan sensitivitas kasus untuk gudang data Anda](#).
- Terenkripsi, jika gudang data target Anda adalah klaster yang disediakan Amazon Redshift. Untuk informasi selengkapnya, lihat [Enkripsi basis data Amazon Redshift](#).
- Dibuat di AWS Wilayah yang sama dengan sumber integrasi.

Untuk membuat gudang data target untuk integrasi nol-ETL Anda, lihat salah satu topik berikut tergantung pada jenis penerapan Anda:

- Untuk membuat klaster yang disediakan Amazon Redshift, lihat. [Membuat klaster](#)
- Untuk membuat workgroup Amazon Redshift Tanpa Server dengan namespace, lihat. [Membuat workgroup dengan namespace](#)

Saat Anda membuat klaster yang disediakan, Amazon Redshift juga membuat grup parameter default. Anda tidak dapat mengedit grup parameter default. Namun, Anda dapat membuat grup parameter khusus sebelum membuat cluster baru dan kemudian mengaitkannya dengan cluster. Atau, Anda dapat mengedit grup parameter yang akan dikaitkan dengan cluster yang dibuat. Anda juga harus mengaktifkan sensitivitas huruf besar untuk grup parameter baik saat membuat grup parameter kustom atau saat mengedit grup parameter saat ini untuk menggunakan integrasi nol-ETL.

Untuk membuat grup parameter kustom menggunakan konsol Amazon Redshift atau AWS CLI, lihat [Membuat grup parameter](#).

Aktifkan sensitivitas kasus untuk gudang data Anda

Anda dapat melampirkan grup parameter dan mengaktifkan sensitivitas kasus untuk klaster yang disediakan selama pembuatan. Namun, Anda dapat memperbarui workgroup tanpa server melalui AWS Command Line Interface (AWS CLI) hanya setelah dibuat. Ini diperlukan untuk mendukung sensitivitas kasus tabel sumber dan kolom. `enable_case_sensitive_identifier` ini adalah nilai konfigurasi yang menentukan apakah pengidentifikasi nama database, tabel, dan kolom peka huruf besar/kecil. Parameter ini harus dihidupkan untuk membuat integrasi nol-ETL di gudang data. Untuk informasi selengkapnya, lihat [enable_case_sensitive_identifier](#).

Untuk Amazon Redshift Tanpa Server —. [Aktifkan sensitivitas huruf besar/case untuk Amazon Redshift Serverless menggunakan AWS CLI](#) Perhatikan bahwa Anda dapat mengaktifkan sensitivitas huruf besar untuk Amazon Redshift Serverless hanya dari file. AWS CLI

Untuk klaster yang disediakan Amazon Redshift, aktifkan sensitivitas huruf besar untuk klaster target Anda menggunakan salah satu topik berikut:

- [Aktifkan sensitivitas huruf besar untuk cluster yang disediakan Amazon Redshift menggunakan konsol Amazon Redshift](#)
- [Aktifkan sensitivitas huruf besar/kecil untuk cluster yang disediakan Amazon Redshift menggunakan AWS CLI](#)

Aktifkan sensitivitas huruf besar/case untuk Amazon Redshift Serverless menggunakan AWS CLI

Jalankan AWS CLI perintah berikut untuk mengaktifkan sensitivitas huruf besar/case untuk workgroup Anda.

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

Tunggu status workgroup Active sebelum melanjutkan ke langkah berikutnya.

Aktifkan sensitivitas huruf besar untuk cluster yang disediakan Amazon Redshift menggunakan konsol Amazon Redshift

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Di panel navigasi kiri, pilih Dasbor cluster yang disediakan.
3. Pilih klaster yang disediakan tempat Anda ingin mereplikasi data.
4. Di panel navigasi kiri, pilih Konfigurasi > Manajemen beban kerja.
5. Pada halaman manajemen beban kerja, pilih grup parameter.
6. Pilih tab Parameter.
7. Pilih Edit parameter, lalu ubah enable_case_sensitive_identifier menjadi true.
8. Lalu, pilih Simpan.

Aktifkan sensitivitas huruf besar/kecil untuk cluster yang disediakan Amazon Redshift menggunakan AWS CLI

1. Karena Anda tidak dapat mengedit grup parameter default, dari program terminal Anda, jalankan AWS CLI perintah berikut untuk membuat grup parameter kustom. Nanti, Anda akan mengaitkannya dengan cluster yang disediakan.

```
aws redshift create-cluster-parameter-group \  
    --parameter-group-name zero-etl-params \  
    --parameter-group-family redshift-2.0 \  
    --description "Parameter group for zero-etl-params"
```

```
--description "Param group for zero-ETL integrations"
```

2. Jalankan AWS CLI perintah berikut untuk mengaktifkan sensitivitas huruf besar untuk grup parameter.

```
aws redshift modify-cluster-parameter-group \  
  --parameter-group-name zero-etl-params \  
  --parameters ParameterName=enable_case_sensitive_identifier,ParameterValue=true
```

3. Jalankan perintah berikut untuk mengaitkan grup parameter dengan cluster.

```
aws redshift modify-cluster \  
  --cluster-identifier target-cluster \  
  --cluster-parameter-group-name zero-etl-params
```

4. Tunggu klaster yang disediakan tersedia. Anda dapat memeriksa status cluster dengan menggunakan `describe-cluster` perintah. Kemudian, jalankan perintah berikut untuk me-reboot cluster.

```
aws redshift reboot-cluster \  
  --cluster-identifier target-cluster
```

Konfigurasi otorisasi untuk gudang data Amazon Redshift Anda

Untuk mereplikasi data dari sumber integrasi Anda ke gudang data Amazon Redshift, Anda harus menambahkan dua entitas berikut:

- Prinsipal resmi — mengidentifikasi pengguna atau peran yang dapat membuat integrasi nol-ETL ke dalam gudang data.
- Sumber integrasi resmi - mengidentifikasi basis data sumber yang dapat memperbarui gudang data.

Anda dapat mengonfigurasi prinsip resmi dan sumber integrasi resmi dari tab Kebijakan Sumber Daya di konsol Amazon Redshift atau menggunakan operasi Amazon Redshift API. `PutResourcePolicy`

Tambahkan prinsipal resmi

Untuk membuat integrasi nol-ETL ke dalam workgroup Redshift Serverless atau klaster yang disediakan, otorisasi akses ke namespace terkait atau klaster yang disediakan.

Anda dapat melewati langkah ini jika kedua kondisi berikut benar:

- Akun AWS Yang memiliki grup kerja Redshift Serverless atau cluster yang disediakan juga memiliki database sumber.
- Prinsip tersebut dikaitkan dengan kebijakan IAM berbasis identitas dengan izin untuk membuat integrasi nol-ETL ke dalam namespace Redshift Tanpa Server atau cluster yang disediakan ini.

Tambahkan prinsipal resmi ke namespace Amazon Redshift Tanpa Server

1. Di konsol Amazon Redshift, di panel navigasi kiri, pilih Redshift Serverless.
2. Pilih konfigurasi Namespace, lalu pilih namespace Anda, dan buka tab Kebijakan Sumber Daya.
3. Pilih Tambahkan prinsipal resmi.
4. Untuk setiap prinsipal resmi yang ingin Anda tambahkan, masukkan ke dalam namespace baik ARN AWS pengguna atau peran, atau ID yang ingin Anda berikan akses untuk membuat Akun AWS integrasi nol-ETL. ID akun disimpan sebagai ARN.
5. Pilih Simpan perubahan.

Tambahkan prinsipal resmi ke kluster yang disediakan Amazon Redshift

1. Di konsol Amazon Redshift, di panel navigasi kiri, pilih Dasbor cluster yang disediakan.
2. Pilih Cluster, lalu pilih klaster, dan buka tab Kebijakan Sumber Daya.
3. Pilih Tambahkan prinsipal resmi.
4. Untuk setiap prinsipal resmi yang ingin Anda tambahkan, masukkan ke dalam klaster baik ARN AWS pengguna atau peran, atau ID Akun AWS yang ingin Anda berikan akses untuk membuat integrasi nol-ETL. ID akun disimpan sebagai ARN.
5. Pilih Simpan perubahan.

Tambahkan sumber integrasi resmi

Untuk mengizinkan sumber Anda memperbarui gudang data Amazon Redshift, Anda harus menambahkannya sebagai sumber integrasi resmi ke namespace.

Menambahkan sumber integrasi resmi ke namespace Amazon Redshift Tanpa Server

1. Di konsol Amazon Redshift, buka dasbor Tanpa Server.
2. Pilih nama namespace.
3. Buka tab Kebijakan Sumber Daya.
4. Pilih Tambahkan sumber integrasi resmi.
5. Tentukan ARN sumber untuk integrasi nol-ETL.

Note

Menghapus sumber integrasi resmi menghentikan data mereplikasi ke namespace. Tindakan ini menonaktifkan semua integrasi nol-ETL dari sumber itu ke namespace ini.

Menambahkan sumber integrasi resmi ke klaster yang disediakan Amazon Redshift

1. Di konsol Amazon Redshift, buka dasbor cluster yang disediakan.
2. Pilih nama cluster yang disediakan.
3. Buka tab Kebijakan Sumber Daya.
4. Pilih Tambahkan sumber integrasi resmi.
5. Tentukan ARN dari sumber yang merupakan sumber data untuk integrasi nol-ETL.

Note

Menghapus sumber integrasi resmi menghentikan data mereplikasi ke dalam klaster yang disediakan. Tindakan ini menonaktifkan semua integrasi nol-ETL dari sumber itu ke dalam cluster yang disediakan Amazon Redshift ini.

Konfigurasi otorisasi menggunakan Amazon Redshift API

Anda dapat menggunakan operasi Amazon Redshift API untuk mengonfigurasi kebijakan sumber daya yang berfungsi dengan integrasi nol-ETL.

Untuk mengontrol sumber yang dapat membuat integrasi masuk ke dalam namespace, buat kebijakan sumber daya dan lampirkan ke namespace. Dengan kebijakan sumber daya, Anda dapat

menentukan sumber yang memiliki akses ke integrasi. Kebijakan sumber daya dilampirkan ke namespace gudang data target Anda untuk memungkinkan sumber membuat integrasi masuk untuk mereplikasi data langsung dari sumber ke Amazon Redshift.

Berikut ini adalah contoh kebijakan sumber daya.

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:*"
    }
  ]
}
```

Berikut ini merangkum operasi Amazon Redshift API yang berlaku untuk mengonfigurasi kebijakan sumber daya untuk integrasi:

- Gunakan operasi [PutResourcePolicy](#) API untuk mempertahankan kebijakan sumber daya. Saat Anda memberikan kebijakan sumber daya lain, kebijakan sumber daya sebelumnya pada sumber

daya diganti. Gunakan contoh kebijakan sumber daya sebelumnya, yang memberikan izin untuk tindakan berikut:

- **CreateInboundIntegration**— Memungkinkan prinsipal sumber untuk membuat integrasi inbound untuk data yang akan direplikasi dari sumber ke gudang data target.
- **AuthorizeInboundIntegration**— Memungkinkan Amazon Redshift untuk terus memvalidasi bahwa gudang data target dapat menerima data yang direplikasi dari sumber ARN.
- Gunakan operasi [GetResourcePolicy](#) API untuk melihat kebijakan sumber daya yang ada.
- Gunakan operasi [DeleteResourcePolicy](#) API untuk menghapus kebijakan sumber daya dari sumber daya.

Untuk memperbarui kebijakan sumber daya, Anda juga dapat menggunakan [put-resource-policy](#) AWS CLI perintah. Misalnya, untuk menempatkan kebijakan sumber daya di ruang nama Amazon Redshift ARN untuk sumber DynamoDB, jalankan perintah yang mirip dengan berikut ini. AWS CLI

```
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"
```

Di mana `rs-rp.json` berisi:

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"
    }
  ]
}
```

Buat integrasi nol-ETL

Pertama, Anda membuat integrasi nol-ETL untuk mereplikasi data sumber Anda ke Amazon Redshift.

Sumber data Anda menentukan jenis integrasi nol-ETL yang akan dibuat.

Topik

- [Buat integrasi Nol-ETL untuk Aurora](#)
- [Buat integrasi nol-ETL untuk Amazon RDS](#)
- [Buat integrasi nol-ETL untuk DynamoDB](#)
- [Buat integrasi nol-ETL dengan aplikasi](#)

Buat integrasi Nol-ETL untuk Aurora

Pada langkah ini, Anda membuat integrasi Aurora Zero-ETL dengan Amazon Redshift.

Untuk membuat integrasi Aurora Zero-ETL dengan Amazon Redshift

1. Dari konsol Amazon RDS, [buat grup parameter cluster DB kustom](#) seperti yang dijelaskan dalam Panduan Pengguna Amazon Aurora.
2. Dari konsol Amazon RDS, [buat cluster DB Amazon Aurora sumber](#) seperti yang dijelaskan dalam Panduan Pengguna Amazon Aurora.
3. Dari konsol Amazon Redshift: [Membuat dan mengonfigurasi gudang data Amazon Redshift target](#)

- Dari AWS CLI atau konsol Amazon Redshift:.. [Aktifkan sensitivitas kasus untuk gudang data Anda](#)
 - Dari konsol Amazon Redshift:.. [Konfigurasi otorisasi untuk gudang data Amazon Redshift Anda](#)
4. Dari konsol Amazon RDS, [buat integrasi Nol-ETL](#) seperti yang dijelaskan dalam Panduan Pengguna Amazon Aurora.
 5. Dari konsol Amazon Redshift atau editor kueri v2, [buat database Amazon Redshift](#) dari integrasi Anda.

Kemudian, [kueri dan buat tampilan terwujud dengan data yang direplikasi](#).

Untuk informasi terperinci tentang cara membuat integrasi Aurora Zero-ETL, lihat Membuat [integrasi Nol-ETL Amazon Aurora dengan Amazon Redshift di Panduan Pengguna](#) Amazon Aurora.

Buat integrasi nol-ETL untuk Amazon RDS

Pada langkah ini, Anda membuat integrasi Amazon RDS Zero-ETL dengan Amazon Redshift. Redshift mendukung integrasi dengan RDS untuk MySQL, RDS untuk PostgreSQL, dan RDS untuk Oracle.

Untuk membuat integrasi Amazon RDS Zero-ETL dengan Amazon Redshift

1. Dari konsol Amazon RDS, [buat grup parameter DB khusus](#) seperti yang dijelaskan dalam Panduan Pengguna Amazon RDS.
2. Dari konsol Amazon RDS, [buat instance Amazon RDS sumber](#) seperti yang dijelaskan dalam Panduan Pengguna Amazon RDS.
3. Dari konsol Amazon Redshift:.. [Membuat dan mengonfigurasi gudang data Amazon Redshift target](#)
 - Dari AWS CLI atau konsol Amazon Redshift:.. [Aktifkan sensitivitas kasus untuk gudang data Anda](#)
 - Dari konsol Amazon Redshift:.. [Konfigurasi otorisasi untuk gudang data Amazon Redshift Anda](#)
4. Dari konsol Amazon RDS, [buat integrasi nol-ETL](#) seperti yang dijelaskan dalam Panduan Pengguna Amazon RDS.

5. Dari konsol Amazon Redshift atau editor kueri v2, [buat database Amazon Redshift](#) dari integrasi Anda.

Kemudian, [kueri dan buat tampilan terwujud dengan data yang direplikasi](#).

Konsol Amazon RDS menawarkan alur pembuatan step-by-step integrasi, di mana Anda menentukan database sumber dan gudang data Amazon Redshift target. Jika masalah terjadi, maka Anda dapat memilih untuk meminta Amazon RDS memperbaiki masalah untuk Anda alih-alih memperbaikinya secara manual di konsol Amazon RDS atau Amazon Redshift.

Untuk petunjuk mendetail untuk membuat integrasi RDS Nol-ETL, lihat [Membuat integrasi Amazon RDS Zero-ETL dengan Amazon](#) Redshift di Panduan Pengguna Amazon RDS.

Untuk petunjuk mendetail untuk secara khusus membuat integrasi Amazon RDS for Oracle Zero-ETL, [lihat Menyiapkan](#) integrasi nol-ETL di Panduan Pengguna Oracle Database@AWS

Buat integrasi nol-ETL untuk DynamoDB

Sebelum membuat integrasi nol-ETL, tinjau pertimbangan dan persyaratan yang diuraikan dalam [Pertimbangan saat menggunakan integrasi nol-ETL dengan Amazon Redshift](#) Ikuti alur umum ini untuk membuat integrasi nol-ETL dari DynamoDB ke Amazon Redshift

Untuk mereplikasi data DynamoDB ke Amazon Redshift dengan integrasi nol-ETL

1. Konfirmasikan kredensial masuk Anda memungkinkan izin bekerja dengan integrasi nol-ETL dengan Amazon Redshift dan DynamoDB. Lihat [Kebijakan IAM untuk bekerja dengan integrasi DynamoDB zero-ETL](#) contoh kebijakan IAM.
2. Dari konsol DynamoDB, [konfigurasi tabel DynamoDB Anda agar point-in-time memiliki pemulihan \(PITR\), kebijakan sumber daya, kebijakan berbasis identitas, dan izin kunci enkripsi seperti yang dijelaskan dalam Panduan Pengembang Amazon DynamoDB](#).
3. Dari konsol Amazon Redshift: [Membuat dan mengonfigurasi gudang data Amazon Redshift target](#)
 - Dari AWS CLI atau konsol Amazon Redshift: [Aktifkan sensitivitas kasus untuk gudang data Anda](#)
 - Dari konsol Amazon Redshift: [Konfigurasi otorisasi untuk gudang data Amazon Redshift Anda](#)

4. Dari konsol Amazon Redshift, buat integrasi nol-ETL seperti yang dijelaskan nanti dalam topik ini.
5. Dari konsol Amazon Redshift, buat database tujuan di gudang data Amazon Redshift Anda. Untuk informasi selengkapnya, lihat [Membuat database tujuan di Amazon Redshift](#).
6. Dari konsol Amazon Redshift, kueri data yang direplikasi di gudang data Amazon Redshift. Untuk informasi selengkapnya, lihat [Menanyakan data yang direplikasi di Amazon Redshift](#).

Pada langkah ini, Anda membuat integrasi Amazon DynamoDB Zero-ETL dengan Amazon Redshift.

Amazon Redshift console

Untuk membuat integrasi Amazon DynamoDB Zero-ETL dengan Amazon Redshift menggunakan konsol Amazon Redshift

1. Dari konsol Amazon Redshift, pilih integrasi Nol-ETL. Pada panel dengan daftar integrasi nol-ETL, pilih Buat integrasi nol-ETL, Buat integrasi DynamoDB.
2. Pada halaman untuk membuat integrasi, masukkan informasi tentang integrasi sebagai berikut:
 - Masukkan nama Integrasi — Yang merupakan nama unik yang dapat digunakan untuk mereferensikan integrasi Anda.
 - Masukkan Deskripsi — Yang menjelaskan data yang akan direplikasi dari sumber ke target.
 - Pilih tabel Sumber DynamoDB - Satu tabel DynamoDB dapat dipilih. Point-in-timepemulihan (PITR) harus diaktifkan di atas meja. Hanya tabel dengan ukuran tabel hingga 100 tebibytes (TiB) yang ditampilkan. Tabel DynamoDB sumber harus dienkripsi. Sumber juga harus memiliki kebijakan sumber daya dengan prinsip resmi dan sumber integrasi. Jika ini kebijakan tidak benar, Anda disajikan dengan opsi Perbaiki untuk saya.
 - Pilih gudang data Amazon Redshift target — Gudang data dapat berupa klaster yang disediakan Amazon Redshift atau grup kerja Redshift Serverless. Jika target Amazon Redshift Anda ada di akun yang sama, Anda dapat memilih target. Jika target berada di akun yang berbeda, Anda menentukan ARN gudang data Redshift. Target harus memiliki kebijakan sumber daya dengan prinsip resmi dan sumber integrasi dan `enable_case_sensitive_identifier` parameter disetel ke true. Jika Anda tidak memiliki kebijakan sumber daya yang benar pada target dan target Anda berada di akun yang sama, Anda dapat memilih opsi Perbaiki untuk saya untuk menerapkan kebijakan

sumber daya secara otomatis selama proses integrasi buat. Jika target Anda berbeda Akun AWS, Anda perlu menerapkan kebijakan sumber daya di gudang Amazon Redshift secara manual. Jika gudang data Amazon Redshift target Anda tidak memiliki opsi grup parameter yang benar yang `enable_case_sensitive_identifier` dikonfigurasi `true`, Anda dapat memilih opsi Perbaiki untuk saya untuk memperbarui grup parameter ini secara otomatis dan mem-boot ulang gudang selama proses integrasi pembuatan.

- Masukkan hingga 50 tag Kunci dan dengan Nilai opsional — Untuk memberikan metadata tambahan tentang integrasi. Untuk informasi selengkapnya, lihat [Menandai sumber daya di Amazon Redshift](#).
- Pilih Opsi enkripsi — Untuk mengenkripsi integrasi. Untuk informasi selengkapnya, lihat [Mengekripsi integrasi DynamoDB dengan kunci yang dikelola pelanggan](#).

Saat Anda mengenkripsi integrasi, Anda juga dapat menambahkan konteks enkripsi tambahan. Untuk informasi selengkapnya, lihat [Konteks enkripsi](#).

3. Halaman ulasan ditampilkan di mana Anda dapat memilih Create DynamoDB integrasi.
4. Halaman kemajuan ditampilkan di mana Anda dapat melihat kemajuan berbagai tugas untuk membuat integrasi nol-ETL.
5. Setelah integrasi dibuat dan aktif, pada halaman detail integrasi, pilih Connect to database. Saat gudang data Amazon Redshift Anda pertama kali dibuat, database juga dibuat. Anda perlu terhubung ke database apa pun di gudang data target Anda untuk membuat database lain untuk integrasi. Di halaman Connect to database, tentukan apakah Anda dapat menggunakan koneksi terbaru dan pilih metode Autentikasi. Bergantung pada metode otentikasi Anda, masukkan informasi untuk terhubung ke database yang ada di target Anda. Informasi otentikasi ini dapat mencakup nama Database yang ada (biasanya, `dev`) dan pengguna Database yang ditentukan saat database dibuat dengan gudang data Amazon Redshift.
6. Setelah Anda terhubung ke database, pilih Buat database dari integrasi untuk membuat database yang menerima data dari sumber. Saat Anda membuat database, Anda memberikan ID Integrasi, nama gudang data, dan nama Database.
7. Setelah status integrasi dan database tujuan `Active`, data mulai mereplikasi dari tabel DynamoDB Anda ke tabel target. Saat Anda menambahkan data ke sumber, data tersebut akan mereplikasi secara otomatis ke gudang data Amazon Redshift target.

AWS CLI

Untuk membuat integrasi Amazon DynamoDB Zero-ETL dengan Amazon AWS CLI Redshift menggunakan, gunakan perintah dengan opsi berikut: `create-integration`

- `integration-name` – Tentukan nama untuk integrasi.
- `source-arn`— Tentukan ARN dari sumber DynamoDB.
- `target-arn`— Tentukan ARN namespace dari cluster yang disediakan Amazon Redshift atau target grup kerja Redshift Serverless.

Contoh berikut menciptakan integrasi dengan memberikan nama integrasi, sumber ARN, dan ARN target. Integrasi tidak dienkripsi.

```
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"
  }

```

Contoh berikut membuat integrasi menggunakan kunci yang dikelola pelanggan untuk enkripsi. Sebelum membuat integrasi:

- Buat kunci yang dikelola pelanggan (disebut “CMCMK” dalam contoh) di akun yang sama (disebut “AccountA” dalam contoh) di tabel DynamoDB sumber.
- Pastikan bahwa user/role (disebut “RoleA” dalam contoh) sedang digunakan untuk membuat integrasi `kms:CreateGrant` memiliki `kms:DescribeKey` dan izin pada kunci KMS ini.
- Tambahkan yang berikut ini ke kebijakan utama.

```

{
  "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": []
}

```

Kebijakan IAM untuk bekerja dengan integrasi DynamoDB zero-ETL

Saat membuat integrasi nol-ETL, kredensi masuk Anda harus memiliki izin untuk melakukan tindakan DynamoDB dan Amazon Redshift dan juga pada sumber daya yang terlibat sebagai sumber dan target integrasi. Berikut ini adalah contoh yang menunjukkan izin minimum yang diperlukan.

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>"
    }
]
}

```

Menkripsi integrasi DynamoDB dengan kunci yang dikelola pelanggan

Jika Anda menentukan kunci KMS kustom dan bukan Kunci milik AWS saat Anda membuat integrasi DynamoDB Zero-ETL, kebijakan kunci harus menyediakan akses utama layanan Amazon Redshift ke tindakan tersebut. CreateGrant Selain itu, harus mengizinkan akun pemohon atau izin peran untuk menjalankan DescribeKey dan CreateGrant tindakan.

Contoh pernyataan kebijakan kunci berikut menunjukkan izin yang diperlukan dalam kebijakan Anda. Beberapa contoh menyertakan kunci konteks untuk mengurangi cakupan izin lebih lanjut.

Contoh pernyataan kebijakan utama

Pernyataan kebijakan berikut memungkinkan akun atau peran pemohon untuk mengambil informasi tentang kunci KMS.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::{account-ID}:role/{role-name}"
  },
  "Action": "kms:DescribeKey",
  "Resource": "*"
}
```

Pernyataan kebijakan berikut memungkinkan akun atau peran pemohon untuk menambahkan hibah ke kunci KMS. Kunci [kms:ViaService](#) kondisi membatasi penggunaan kunci KMS untuk permintaan dari Amazon Redshift.

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

Pernyataan kebijakan berikut memungkinkan kepala layanan Amazon Redshift untuk menambahkan hibah ke kunci KMS.

```
{
  "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:*"
  }
}
}
}

```

Untuk informasi selengkapnya, lihat [Membuat kebijakan kunci](#) di Panduan Developer AWS Key Management Service .

Konteks enkripsi

Saat Anda mengenkripsi integrasi nol-ETL, Anda dapat menambahkan pasangan nilai kunci sebagai konteks enkripsi Tambahan. Anda mungkin ingin menambahkan pasangan kunci-nilai ini untuk menambahkan informasi kontekstual tambahan tentang data yang direplikasi. Untuk informasi lebih lanjut, lihat [Konteks enkripsi](#) di Panduan Developer AWS Key Management Service .

Amazon Redshift menambahkan pasangan konteks enkripsi berikut selain yang Anda tambahkan:

- `aws:redshift:integration:arn` - IntegrationArn
- `aws:servicename:id` - Redshift

Ini mengurangi jumlah keseluruhan pasangan yang dapat Anda tambahkan dari 8 menjadi 6, dan berkontribusi pada batas karakter keseluruhan dari batasan hibah. Untuk informasi selengkapnya, lihat [Menggunakan batasan hibah di Panduan Pengembang](#).AWS Key Management Service

Buat integrasi nol-ETL dengan aplikasi

Pada langkah ini, Anda membuat integrasi nol-ETL dengan aplikasi dengan Amazon Redshift.

Untuk membuat integrasi nol-ETL dengan aplikasi dengan Amazon Redshift

1. Dari konsol Amazon Redshift: [Membuat dan mengonfigurasi gudang data Amazon Redshift target](#)
 - Dari AWS CLI atau konsol Amazon Redshift: [Aktifkan sensitivitas kasus untuk gudang data Anda](#)
 - Dari konsol Amazon Redshift: [Konfigurasi otorisasi untuk gudang data Amazon Redshift Anda](#)
2. Dari AWS Glue konsol: [Membuat integrasi seperti yang](#) dijelaskan dalam Panduan AWS Glue Pengembang.
3. Setelah database tujuan dibuat dan data mulai mereplikasi, Anda dapat melakukan kueri dan membuat data terwujud untuk data yang direplikasi. Untuk informasi selengkapnya, lihat [Menanyakan data yang direplikasi di Amazon Redshift](#).

Untuk informasi rinci untuk membuat integrasi nol-ETL dengan aplikasi, lihat Integrasi [nol-ETL di Panduan Pengembang](#).AWS Glue

Membuat database tujuan di Amazon Redshift

Untuk mereplikasi data dari sumber Anda ke Amazon Redshift, Anda harus membuat database dari integrasi Anda di Amazon Redshift.

Connect ke workgroup Redshift Serverless target Anda atau kluster yang disediakan dan buat database dengan referensi ke pengenalan integrasi Anda. Pengidentifikasi ini adalah nilai yang dikembalikan `integration_id` saat Anda menanyakan tampilan [SVV_INTEGRATION](#).

Important

Sebelum membuat database dari integrasi Anda, integrasi nol-ETL Anda harus dibuat dan dalam Active status di konsol Amazon Redshift.

Sebelum Anda dapat mulai mereplikasi data dari sumber Anda ke Amazon Redshift, buat database dari integrasi di Amazon Redshift. Anda dapat membuat database menggunakan konsol Amazon Redshift atau editor kueri v2.

Amazon Redshift console

1. Di panel navigasi kiri, pilih Integrasi nol-ETL.
2. Dari daftar integrasi, pilih integrasi.
3. Jika Anda menggunakan klaster yang disediakan, Anda harus terlebih dahulu terhubung ke database. Pilih Hubungkan ke basis data. Anda dapat terhubung menggunakan koneksi terbaru, atau dengan membuat koneksi baru.
4. Untuk membuat database dari integrasi, pilih Buat database dari integrasi.
5. Masukkan nama database Tujuan. ID Integrasi dan nama gudang Data sudah terisi sebelumnya.

Untuk Aurora PostgreSQL, RDS untuk PostgreSQL, atau RDS untuk sumber Oracle, masukkan database bernama Sumber yang Anda tentukan saat membuat integrasi Zero-ETL Anda. Dalam kasus ini, Anda dapat memetakan maksimum 100 basis data sumber ke database Amazon Redshift.

6. Pilih Buat basis data.

Amazon Redshift query editor v2

1. Arahkan ke konsol Amazon Redshift dan pilih Editor kueri v2.
2. Di panel kiri, pilih workgroup Amazon Redshift Serverless atau klaster yang disediakan Amazon Redshift, lalu sambungkan ke sana.
3. Untuk mendapatkan ID integrasi, navigasikan ke daftar integrasi di konsol Amazon Redshift.

Atau, jalankan perintah berikut untuk mendapatkan `integration_id` nilai:

```
SELECT integration_id FROM SVV_INTEGRATION;
```

4. Kemudian, jalankan perintah berikut untuk membuat database. Dengan menentukan ID integrasi, Anda membuat koneksi antara database dan sumber Anda.

Gantikan `integration_id` dengan nilai yang dikembalikan oleh perintah sebelumnya.

```
CREATE DATABASE destination_db_name FROM INTEGRATION 'integration_id';
```

Untuk sumber PostgreSQL Aurora, Anda juga harus menyertakan referensi ke database bernama dalam cluster yang Anda tentukan saat Anda membuat integrasi. Contoh:

```
CREATE DATABASE "destination_db_name" FROM INTEGRATION 'integration_id'  
DATABASE "named_db";
```

Untuk informasi selengkapnya tentang membuat database untuk target integrasi nol-ETL, lihat [MEMBUAT DATABASE di Panduan Pengembang Database](#) Amazon Redshift. Anda dapat menggunakan ALTER DATABASE untuk mengubah parameter database seperti REFRESH INTERVAL. Untuk informasi selengkapnya tentang mengubah database untuk target integrasi nol-ETL, lihat [MENGUBAH DATABASE di Panduan Pengembang Database](#) Amazon Redshift.

Note

Hanya sumber integrasi Anda yang dapat memperbarui data dalam database yang Anda buat dari integrasi Anda. Untuk mengubah skema tabel, jalankan perintah DDL atau DHTML terhadap tabel di sumbernya. Anda dapat menjalankan perintah DDL dan DHTML terhadap tabel di sumbernya, tetapi Anda hanya dapat menjalankan perintah DDL dan kueri hanya-baca pada database tujuan.

Untuk informasi tentang melihat status database tujuan, lihat [Melihat integrasi nol-ETL](#).

Setelah membuat database tujuan, Anda dapat menambahkan data ke sumber Anda. Untuk menambahkan data ke sumber Anda, lihat salah satu topik berikut:

- Untuk sumber Aurora, lihat [Menambahkan data ke cluster DB sumber di Panduan Pengguna Amazon Aurora](#).
- Untuk sumber Amazon RDS, lihat [Menambahkan data ke instans DB sumber](#) di Panduan Pengguna Amazon RDS.
- Untuk sumber DynamoDB, [lihat Memulai DynamoDB di Panduan Pengembang Amazon DynamoDB](#).
- Untuk integrasi nol-ETL dengan sumber aplikasi, lihat Integrasi [nol-ETL di Panduan Pengembang AWS Glue](#).

Menanyakan data yang direplikasi di Amazon Redshift

Setelah Anda menambahkan data ke sumber Anda, itu direplikasi dalam waktu dekat ke gudang data Amazon Redshift, dan siap untuk kueri. Untuk informasi tentang metrik integrasi dan statistik tabel, lihat [Metrik untuk integrasi nol-ETL](#).

Note

Karena database sama dengan skema di MySQL, tingkat database MySQL memetakan ke tingkat skema Amazon Redshift. Perhatikan perbedaan pemetaan ini saat Anda menanyakan data yang direplikasi dari Aurora MySQL atau RDS untuk MySQL.

Untuk menanyakan data yang direplikasi

1. Arahkan ke konsol Amazon Redshift dan pilih Editor kueri v2.
2. Connect ke workgroup Amazon Redshift Serverless atau klaster yang disediakan Amazon Redshift dan pilih database Anda dari daftar tarik-turun.
3. Gunakan pernyataan SELECT untuk memilih semua data yang direplikasi dari skema dan tabel yang Anda buat di sumber. Untuk sensitivitas kasus, gunakan tanda kutip ganda (") untuk nama skema, tabel, dan kolom. Contoh:

```
SELECT * FROM "schema_name"."table_name";
```

Anda juga dapat melakukan kueri data menggunakan Amazon Redshift Data API.

Menanyakan data yang direplikasi dengan tampilan terwujud

Anda dapat membuat tampilan terwujud di database Amazon Redshift lokal Anda untuk mengubah data yang direplikasi melalui integrasi nol-ETL. Connect ke database lokal Anda dan gunakan kueri lintas basis data untuk mengakses database tujuan. Anda dapat menggunakan nama objek yang sepenuhnya memenuhi syarat dengan notasi tiga bagian (destination-database-name.schema-name.table-name) atau membuat skema eksternal yang mereferensikan pasangan skema database tujuan dan menggunakan notasi dua bagian (.table-name). external-schema-name Untuk informasi selengkapnya tentang kueri lintas basis data, lihat [Mengkueri data di seluruh basis data](#).

Gunakan contoh berikut untuk membuat dan menyisipkan data sampel ke dalam *sales_zetl* dan *event_zetl* tabel dari sumber *tickit_zetl*. Tabel direplikasi ke dalam database Amazon *zetl_int_db* Redshift.

```
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");
```

Anda dapat membuat tampilan terwujud untuk mendapatkan total penjualan per acara menggunakan notasi tiga bagian:

```
--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.tickit_zetl.sales_zetl S, zetl_int_db.tickit_zetl.event_zetl E  
WHERE S.eventid = E.eventid  
GROUP BY 1);
```

Anda dapat membuat tampilan terwujud untuk mendapatkan total penjualan per acara menggunakan notasi dua bagian:

```
--two part notation external-schema-name.table-name notation  
CREATE EXTERNAL schema ext_tickit_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
);

```

Untuk melihat tampilan terwujud yang Anda buat, gunakan contoh berikut.

```
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        |
+-----+-----+

```

Menanyakan data yang direplikasi dari DynamoDB

Saat Anda mereplikasi data dari Amazon DynamoDB ke database Amazon Redshift, data tersebut disimpan dalam tampilan terwujud dalam kolom tipe data SUPER.

Untuk contoh ini, data berikut disimpan di DynamoDB.

```

{
  "key1": {
    "S": "key_1"
  }
}

```

```
{
  "key2": {
    "N": 0
  },
  "payload": {
    "L": [
      {
        "S": "sale1"
      },
      {
        "S": "sale2"
      }
    ]
  }
}
```

Tampilan terwujud Amazon Redshift didefinisikan sebagai berikut.

```
CREATE MATERIALIZED VIEW mv_sales
    BACKUP NO
    AUTO REFRESH YES
    AS
    SELECT "value"."payload"."L"[0]."S"::VARCHAR AS first_payload
    FROM public.sales;
```

Untuk melihat data dalam tampilan terwujud, jalankan perintah SQL.

```
SELECT first_payload FROM mv_sales;
```

Melihat integrasi nol-ETL

Anda dapat melihat integrasi nol-ETL Anda dari konsol Amazon Redshift. Di sini Anda dapat melihat informasi konfigurasi dan status saat ini, dan membuka layar untuk menanyakan dan berbagi data.

Amazon Redshift console

Untuk melihat detail integrasi nol-ETL

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>

2. Dari panel navigasi kiri, pilih dasbor tanpa server atau klaster yang disediakan. Kemudian, pilih integrasi nol-ETL.
3. Pilih integrasi nol-ETL yang ingin Anda lihat. Untuk setiap integrasi, informasi berikut disediakan:
 - **Integration ID** adalah identifier yang dikembalikan ketika integrasi dibuat.
 - **Status** dapat menjadi salah satu dari berikut ini:
 - **Active**—Integrasi Zero-ETL mengirimkan data transaksional ke gudang data Amazon Redshift target.
 - **Syncing**—Integrasi nol-ETL telah mengalami kesalahan yang dapat dipulihkan dan sedang melakukan reseeding data. Tabel yang terpengaruh tidak tersedia untuk kueri di Amazon Redshift hingga selesai disinkronkan ulang.
 - **Failed**—Integrasi nol-ETL mengalami peristiwa atau kesalahan yang tidak dapat dipulihkan yang tidak dapat diperbaiki. Anda harus menghapus dan membuat ulang integrasi nol-ETL.
 - **Creating**—Integrasi Zero-ETL sedang dibuat.
 - **Deleting**—Integrasi nol-ETL sedang dihapus.
 - **Needs attention**—Integrasi nol-ETL mengalami peristiwa atau kesalahan yang memerlukan intervensi manual untuk menyelesaikannya. Untuk memperbaiki masalah, ikuti langkah-langkah dalam pesan kesalahan.
 - **Jenis sumber** adalah jenis data sumber yang mereplikasi ke target. Jenis dapat menentukan pengelola basis data lainnya, seperti Aurora MySQL Compatible Edition, Amazon Aurora PostgreSQL, RDS untuk MySQL, dan dari aplikasi (). GlueSAAS
 - **Sumber ARN** adalah ARN dari sumber data. Untuk sebagian besar sumber, ini adalah ARN dari database sumber atau tabel. Untuk integrasi nol-ETL dengan sumber aplikasi, ini adalah ARN dari objek koneksi. AWS Glue
 - **Target** adalah namespace gudang data Amazon Redshift yang menerima data sumber.
 - **Database** dapat menjadi salah satu dari berikut ini:
 - **No database**—Tidak ada database tujuan untuk integrasi.
 - **Creating**—Amazon Redshift membuat database tujuan untuk integrasi.
 - **Active**—Data sedang direplikasi dari sumber integrasi ke Amazon Redshift.
 - **Error**—Ada kesalahan dengan integrasi.
 - **Recovering**—Integrasi pulih setelah gudang data dimulai ulang.

- Resyncing- Amazon Redshift menyinkronkan ulang tabel dalam integrasi.
- Jenis target adalah jenis gudang data Amazon Redshift.
- Tanggal pembuatan adalah tanggal dan waktu (UTC) saat integrasi dibuat.

Note

Untuk melihat detail integrasi untuk gudang data, pilih halaman detail untuk kluster yang disediakan atau namespace tanpa server, lalu pilih tab integrasi nol-ETL.

Dari daftar integrasi Nol-ETL, Anda dapat memilih Data kueri untuk beralih ke editor kueri Amazon Redshift v2. Basis data target Amazon Redshift mengaktifkan parameter [enable_case_sensitive_identifier](#). <name>Saat Anda menulis SQL, Anda mungkin perlu mengelilingi skema, tabel, dan nama kolom dengan tanda kutip ganda ("""). Untuk informasi selengkapnya tentang kueri data di gudang data Amazon Redshift, lihat. [Menanyakan database menggunakan editor kueri v2](#)

Dari daftar integrasi nol-ETL, Anda dapat memilih Bagikan data untuk membuat datashare. Untuk membuat datashare untuk database Amazon Redshift, ikuti petunjuk pada halaman Create datashare. Sebelum Anda dapat berbagi data dalam database Amazon Redshift Anda, Anda harus terlebih dahulu membuat database tujuan. Untuk informasi selengkapnya tentang berbagi data, lihat [Konsep berbagi data untuk Amazon Redshift](#).

Untuk menyegarkan integrasi Anda, Anda dapat menggunakan perintah [ALTER DATABASE](#). Melakukan hal itu mereplikasi semua data dari sumber integrasi Anda ke database tujuan Anda. Contoh berikut menyegarkan semua tabel yang disinkronkan dan gagal dalam integrasi nol-ETL Anda.

```
ALTER DATABASE sample_integration_db INTEGRATION REFRESH ALL tables;
```

AWS CLI

Untuk menjelaskan integrasi Amazon DynamoDB Zero-ETL dengan Amazon AWS CLI Redshift menggunakan, gunakan perintah dengan opsi berikut: describe-integrations

- `integration-arn`— Tentukan ARN dari integrasi DynamoDB untuk dijelaskan.
- `integration-name`— Tentukan filter opsional yang menentukan satu atau lebih sumber daya untuk dikembalikan.

Contoh berikut menjelaskan integrasi dengan menyediakan ARN integrasi.

```
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"
    }
  ]
}
```

Anda juga dapat memfilter hasil `describe-integrations` dengan `integration-arn`, `source-arn`, `source-types`, atau `status`. Untuk informasi selengkapnya, lihat [deskripsikan integrasi di Panduan CLI Amazon Redshift](#).

Modus sejarah

Dengan mode riwayat, Anda dapat mengonfigurasi integrasi nol-ETL untuk melacak setiap versi (termasuk pembaruan dan penghapusan) catatan Anda di tabel sumber, langsung di Amazon Redshift. Anda dapat menjalankan analitik lanjutan pada semua data Anda, seperti, menjalankan analisis historis, membuat laporan lihat kembali, melakukan analisis tren, dan mengirim pembaruan tambahan ke aplikasi hilir yang dibangun di atas Amazon Redshift. Mode riwayat didukung dengan beberapa integrasi Amazon Redshift Zero-ETL, termasuk Amazon Aurora MySQL, Amazon Aurora

PostgreSQL, Amazon RDS for MySQL, dan Amazon DynamoDB. Mode sejarah juga didukung oleh beberapa aplikasi, seperti Salesforce, SAP ServiceNow, dan Zendesk.

Anda dapat mengaktifkan dan menonaktifkan mode riwayat untuk integrasi Nol-ETL Anda dari konsol Amazon Redshift (). <https://console.aws.amazon.com/redshiftv2/> Gunakan mode riwayat untuk melacak catatan yang telah dihapus atau dimodifikasi dalam sumber integrasi. Pelacakan terjadi di gudang data Amazon Redshift target. Mengaktifkan mode riwayat tidak memengaruhi kinerja kueri analitik reguler pada tabel ini.

Setelah Anda mengaktifkan mode riwayat, tabel yang Anda masukkan ke dalam sumber tidak akan dihapus di Amazon Redshift. Sebagai gantinya, tabel akan muncul dalam DroppedSource keadaan dan Anda masih dapat menanyakan tabel ini. Anda juga masih dapat menggunakan perintah DROP dan RENAME dengan SQL biasa.

Jika Anda ingin menggunakan kembali nama tabel yang sama pada sumbernya, Anda harus DROP atau GANTI NAMA DroppedState tabel yang sesuai sebelum dapat direplikasi ke Amazon Redshift. Pastikan untuk melakukannya sebelum Anda membuat tabel di sumbernya.

Untuk informasi tentang apa yang harus dipertimbangkan saat menggunakan mode riwayat, lihat [Pertimbangan saat menggunakan mode riwayat pada target](#).

Untuk mengelola mode riwayat untuk integrasi nol-ETL

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Dari panel navigasi kiri, pilih dasbor tanpa server atau klaster yang disediakan. Kemudian, pilih integrasi nol-ETL.
3. Pilih integrasi nol-ETL yang ingin Anda kelola, pilih Kelola mode riwayat. Jendela Kelola mode riwayat ditampilkan.
4. Anda dapat Menonaktifkan atau Mengaktifkan mode riwayat untuk tabel target yang direplikasi dari jenis sumber yang memiliki tabel sumber tunggal, seperti Amazon DynamoDB. Jika integrasi nol-ETL memiliki beberapa tabel target yang memungkinkan, Anda dapat menonaktifkan semua tabel yang ada dan yang akan datang, Aktifkan semua tabel yang ada dan yang akan datang, atau Kelola mode riwayat untuk tabel individual. Defaultnya adalah mode riwayat off saat integrasi nol-ETL dibuat.

Saat mode riwayat diputar, kolom berikut ditambahkan ke tabel target Anda untuk melacak perubahan dalam sumber. Mode riwayat on meningkatkan penggunaan dan biaya bulanan karena Amazon Redshift tidak menghapus catatan apa pun di tabel target. Rekaman sumber apa

pun yang dihapus atau diubah membuat catatan baru di target, menghasilkan lebih banyak baris total dalam target dengan beberapa versi rekaman. Rekaman tidak dihapus dari tabel target saat dihapus atau dimodifikasi di sumber. Anda dapat mengelola tabel target dengan menghapus catatan tidak aktif.

Nama kolom	Jenis data	Deskripsi
<code>_record_is_active</code>	Boolean	Menunjukkan jika catatan dalam target saat ini aktif di sumber. Benar menunjukkan catatan aktif.
<code>_record_create_time</code>	Stempel waktu	Waktu mulai (UTC) saat catatan sumber aktif.
<code>_record_delete_time</code>	Stempel waktu	Waktu akhir (UTC) saat catatan sumber diperbarui atau dihapus.

Anda dapat menghapus catatan tidak aktif dari tabel mode riwayat dengan memfilter catatan yang kolomnya `_record_is_active` salah. Perintah SQL DELETE berikut menghapus catatan tidak aktif dari tabel di mana kolom id kurang dari atau sama dengan 100. Setelah Anda menghapus catatan, saat penghapusan vakum otomatis berjalan, penyimpanan untuk catatan yang dihapus direklamasi.

```
DELETE FROM myschema.mytable where not _record_is_active AND id <= 100;
```

Saat mode riwayat diaktifkan off, Amazon Redshift membuat salinan tabel Anda di database target dengan catatan aktif dan tanpa kolom riwayat yang ditambahkan. Amazon Redshift mengganti nama tabel Anda menjadi untuk `table-name_historical_timestamp` Anda gunakan. Anda dapat menjatuhkan salinan tabel ini jika Anda tidak lagi membutuhkannya. Anda dapat mengganti nama tabel ini menggunakan perintah ALTER TABLE. Contoh:

```
ALTER TABLE [schema-name.]table-name_historical_timestamp RENAME TO new_table_name;
```

Untuk informasi selengkapnya, lihat [MENGUBAH TABEL](#) di Panduan Pengembang Database Amazon Redshift.

Anda juga dapat mengelola mode riwayat menggunakan perintah SQL CREATE DATABASE dan ALTER DATABASE. Untuk informasi selengkapnya tentang cara menyetel HISTORY_MODE, lihat [MEMBUAT DATABASE dan MENGUBAH DATABASE di Panduan Pengembang Database](#) Amazon Redshift.

Berbagi data Anda di Amazon Redshift

Setelah Anda menambahkan data ke sumber, itu segera direplikasi ke Amazon Redshift dan siap untuk dibagikan dengan membuat datashares.

Untuk berbagi data, Anda harus membuat database tujuan terlebih dahulu.

Untuk berbagi data di Amazon Redshift Tanpa Server menggunakan konsol Amazon Redshift

1. Di konsol Amazon Redshift, di panel navigasi kiri, pilih dasbor Amazon Redshift Tanpa Server > Tanpa Server.
2. Di panel navigasi sebelah kiri, pilih Integrasi nol-ETL.
3. Pilih Bagikan data.
4. Pada halaman create datashare, ikuti langkah-langkah dalam [Membuat](#) datashares.

Untuk berbagi data di klaster yang disediakan Amazon Redshift menggunakan konsol Amazon Redshift

1. Di konsol Amazon Redshift, di panel navigasi kiri, pilih Dasbor cluster yang disediakan.
2. Di panel navigasi sebelah kiri, pilih Integrasi nol-ETL.
3. Dari daftar integrasi, pilih integrasi.
4. Pada halaman detail integrasi, pilih Connect to database.
5. Pada halaman Koneksi ke database, Anda dapat membuat koneksi baru atau menggunakan koneksi terbaru. Pastikan koneksi dibuat ke database tujuan.
6. Jika Anda membuat koneksi baru, maka masukkan nama Database untuk database. Kemudian, klik Connect.
7. Pada halaman detail integrasi, pilih Bagikan data.
8. Pada halaman create datashare, ikuti langkah-langkah dalam [Membuat](#) datashares.

Memantau integrasi nol-ETL

Anda dapat memantau integrasi nol-ETL Anda dengan menanyakan tampilan sistem atau dengan Amazon. EventBridge

Memantau integrasi nol-ETL dengan tampilan sistem Amazon Redshift

Anda dapat memantau integrasi nol-ETL Anda dengan menanyakan tampilan sistem berikut di Amazon Redshift.

- [SVV_INTEGRATION](#) memberikan informasi tentang detail konfigurasi integrasi nol-ETL.
- [SYS_INTEGRATION_ACTIVITY](#) memberikan informasi tentang integrasi nol-ETL yang lengkap.
- [SVV_INTEGRATION_TABLE_MAPPING](#) memberikan informasi tentang pemetaan nilai metadata dari sumber ke target.
- [SVV_INTEGRATION_TABLE_STATE](#) memberikan informasi tentang status integrasi.
- [SYS_INTEGRATION_TABLE_ACTIVITY](#) menyediakan informasi tentang menyisipkan, menghapus, dan memperbarui aktivitas integrasi.
- [SYS_INTEGRATION_TABLE_STATE_CHANGE](#) memberikan informasi tentang log perubahan status tabel untuk integrasi.

Memantau integrasi nol-ETL dengan Amazon EventBridge

Amazon Redshift mengirimkan peristiwa terkait integrasi ke Amazon. EventBridge Untuk daftar acara dan acara terkait IDs, lihat [Pemberitahuan acara integrasi nol-ETL dengan Amazon EventBridge](#).

Metrik untuk integrasi nol-ETL

Anda dapat menggunakan metrik di konsol Amazon Redshift dan CloudWatch Amazon untuk mempelajari kesehatan dan kinerja integrasi nol-ETL Anda. Anda dapat menyesuaikan metrik untuk menampilkan data untuk durasi yang lebih pendek atau lebih lama, atau memilih untuk melihat metrik. CloudWatch Untuk melihat metrik integrasi Anda di konsol Amazon Redshift, pilih integrasi Nol-ETL di panel navigasi kiri dan pilih ID integrasi Anda.

Bergantung pada data sumber integrasi nol-ETL, Amazon Redshift menyediakan metrik pada halaman detail integrasi untuk integrasi. Metrik yang mungkin termasuk jenis berikut:

- Dari tab Metrik Integrasi, grafik berikut ini tersedia:

Metrik	Nama metrik di konsol Amazon Redshift	Deskripsi
IntegrationLag	Lag	Kelambatan dari data waktu berkomitmen ke sumber Anda hingga saat data tersedia untuk kueri di Amazon Redshift. Unit: detik Dimensi: IntegrationId
IntegrationNumTablesReplicated	Tables replicated	Jumlah tabel yang telah direplikasi dari database sumber Anda ke Amazon Redshift. Unit: Hitungan Dimensi: IntegrationId
IntegrationNumTablesFailedReplication	Tables failed	Jumlah tabel yang gagal replikasi. Unit: Hitungan Dimensi: IntegrationId
IntegrationDataTransferred	Data transferred	Jumlah data yang ditransfer dalam byte logis. Unit: Byte Dimensi: IntegrationId

- Dari tab Statistik tabel, Anda dapat melihat daftar tabel yang saat ini aktif atau memiliki kesalahan. Statistik pada tab ini adalah sebagai berikut (tergantung pada jenis sumber):
 - Nama skema — Nama skema tempat tabel berada.
 - Nama tabel — Nama tabel dalam database sumber.
 - Status — Status tabel. Nilai yang mungkin termasuk adalah Synced, Failed, Deleted, Resync Required, dan Resync Initiated.
 - Database — Database Amazon Redshift tempat tabelnya berada.

- Terakhir diperbarui - Tanggal dan waktu (UTC) saat pembaruan terakhir dilakukan ke tabel.
- Jumlah baris tabel — Jumlah baris dalam tabel.
- Ukuran meja — Ukuran meja.

Anda juga dapat melihat grafik jumlah Baris yang disisipkan, dihapus, dan diperbarui untuk jangka waktu yang dipilih.

Memodifikasi integrasi nol-ETL untuk DynamoDB

Pada langkah ini, Anda memodifikasi integrasi DynamoDB Zero-ETL dengan Amazon Redshift.

Amazon Redshift console

Untuk memodifikasi integrasi Amazon DynamoDB Zero-ETL dengan Amazon Redshift menggunakan konsol Amazon Redshift

1. Dari konsol Amazon Redshift, pilih integrasi Nol-ETL. Pada panel dengan daftar integrasi nol-ETL, lalu pilih integrasi DynamoDB yang ingin Anda modifikasi.
2. Pilih Edit dan buat modifikasi pada nama Integrasi atau Deskripsi.
3. Pilih Simpan perubahan untuk menyimpan perubahan Anda.

AWS CLI

Untuk memodifikasi integrasi Amazon DynamoDB Zero-ETL dengan Amazon AWS CLI Redshift menggunakan, gunakan perintah dengan opsi berikut: `modify-integration`

- `integration-arn`— Tentukan ARN dari integrasi DynamoDB untuk memodifikasi.
- `integration-name`— Tentukan nama baru untuk integrasi.
- `description`— Tentukan deskripsi baru untuk integrasi.

Contoh berikut memodifikasi integrasi dengan menyediakan integrasi ARN, deskripsi baru, dan nama baru.

```
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": []
}
```

Hapus integrasi nol-ETL untuk DynamoDB

Saat Anda menghapus integrasi, gudang data target menyimpan data yang direplikasi sebelumnya. Anda dapat terus berbagi dan menanyakan data ini. Namun, data baru di sumber tidak akan mereplikasi ke target.

Pada langkah ini, Anda menghapus integrasi DynamoDB Zero-ETL dengan Amazon Redshift.

Amazon Redshift console

Untuk menghapus integrasi Amazon DynamoDB Zero-ETL dengan Amazon Redshift menggunakan konsol Amazon Redshift

1. Dari konsol Amazon Redshift, pilih integrasi Nol-ETL. Pada panel dengan daftar integrasi nol-ETL, lalu pilih integrasi DynamoDB yang ingin Anda hapus.
2. Pilih Hapus dan berikan informasi yang diminta.
3. Pilih Hapus untuk menghapus integrasi nol-ETL.

AWS CLI

Untuk menghapus integrasi Amazon DynamoDB Zero-ETL dengan Amazon AWS CLI Redshift menggunakan, gunakan perintah dengan opsi berikut: `delete-integration`

- `integration-arn`— Tentukan ARN dari integrasi DynamoDB yang akan dihapus.

Contoh berikut menghapus integrasi dengan menyediakan ARN integrasi.

```
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": []
}
```

Wilayah yang Didukung untuk integrasi nol-ETL

Integrasi Zero-ETL adalah solusi yang dikelola sepenuhnya yang membuat data transaksional dan operasional tersedia di Amazon Redshift dari berbagai sumber operasional dan transaksional, serta aplikasi perusahaan. Halaman ini mencantumkan Wilayah yang tersedia untuk setiap sumber yang didukung.

Aurora MySQL

Wilayah dan versi mesin berikut tersedia untuk integrasi Aurora MySQL Zero-ETL dengan Amazon Redshift.

Region	Aurora MySQL
Africa (Cape Town)	Available
Asia Pasifik (Hong Kong)	Available
Asia Pasifik (Tokyo)	Available
Asia Pasifik (Seoul)	Available
Asia Pasifik (Osaka)	Available
Asia Pasifik (Mumbai)	Available
Asia Pasifik (Hyderabad)	Available
Asia Pasifik (Singapura)	Available
Asia Pasifik (Sydney)	Available
Asia Pasifik (Jakarta)	Available
Asia Pacific (Melbourne)	Available
Asia Pasifik (Malaysia)	Tidak tersedia
Kanada (Pusat)	Available
Kanada Barat (Calgary)	Available
Tiongkok (Beijing)	Available
Tiongkok (Ningxia)	Available
Eropa (Frankfurt)	Available
Europe (Zurich)	Available

Region	Aurora MySQL
Eropa (Stockholm)	Available
Europe (Milan)	Available
Eropa (Spanyol)	Available
Eropa (Irlandia)	Available
Eropa (London)	Available
Eropa (Paris)	Available
Israel (Tel Aviv)	Available
Timur Tengah (UAE)	Available
Timur Tengah (Bahrain)	Available
Amerika Selatan (Sao Paulo)	Available
AS Timur (Virginia Utara)	Available
AS Timur (Ohio)	Available
AS Barat (California Utara)	Available
AS Barat (Oregon)	Available
AWS GovCloud (AS-Timur)	Tidak tersedia
AWS GovCloud (AS-Barat)	Tidak tersedia

Aurora PostgreSQL

Wilayah berikut tersedia untuk integrasi Aurora PostgreSQL Zero-ETL dengan Amazon Redshift.

Region	Aurora PostgreSQL
Africa (Cape Town)	Available
Asia Pasifik (Hong Kong)	Available
Asia Pasifik (Tokyo)	Available
Asia Pasifik (Seoul)	Available
Asia Pasifik (Osaka)	Available
Asia Pasifik (Mumbai)	Available
Asia Pasifik (Hyderabad)	Available
Asia Pasifik (Singapura)	Available
Asia Pasifik (Sydney)	Available
Asia Pasifik (Jakarta)	Available
Asia Pacific (Melbourne)	Available
Asia Pasifik (Malaysia)	Available
Kanada (Pusat)	Available
Kanada Barat (Calgary)	Available
Tiongkok (Beijing)	Available
Tiongkok (Ningxia)	Available
Eropa (Frankfurt)	Available
Europe (Zurich)	Available
Eropa (Stockholm)	Available
Europe (Milan)	Available

Region	Aurora PostgreSQL
Eropa (Spanyol)	Available
Eropa (Irlandia)	Available
Eropa (London)	Available
Eropa (Paris)	Available
Israel (Tel Aviv)	Available
Timur Tengah (UAE)	Available
Timur Tengah (Bahrain)	Available
Amerika Selatan (Sao Paulo)	Available
AS Timur (Virginia Utara)	Available
AS Timur (Ohio)	Available
AS Barat (California Utara)	Available
AS Barat (Oregon)	Available
AWS GovCloud (AS-Timur)	Tidak tersedia
AWS GovCloud (AS-Barat)	Tidak tersedia

Amazon DynamoDB

Wilayah berikut tersedia untuk integrasi DynamoDB Zero-ETL dengan Amazon Redshift.

Region	DynamoDB
Africa (Cape Town)	Available
Asia Pasifik (Hong Kong)	Available

Region	DynamoDB
Asia Pasifik (Taipei)	Available
Asia Pasifik (Tokyo)	Available
Asia Pasifik (Seoul)	Available
Asia Pasifik (Osaka)	Available
Asia Pasifik (Mumbai)	Available
Asia Pasifik (Hyderabad)	Available
Asia Pasifik (Singapura)	Available
Asia Pasifik (Sydney)	Available
Asia Pasifik (Jakarta)	Available
Asia Pacific (Melbourne)	Available
Asia Pasifik (Malaysia)	Available
Asia Pasifik (Thailand)	Available
Kanada (Pusat)	Available
Kanada Barat (Calgary)	Available
Tiongkok (Beijing)	Available
Tiongkok (Ningxia)	Available
Eropa (Frankfurt)	Available
Europe (Zurich)	Available
Eropa (Stockholm)	Available
Europe (Milan)	Available

Region	DynamoDB
Eropa (Spanyol)	Available
Eropa (Irlandia)	Available
Eropa (London)	Available
Eropa (Paris)	Available
Israel (Tel Aviv)	Available
Timur Tengah (UAE)	Available
Timur Tengah (Bahrain)	Available
Meksiko (Tengah)	Available
Amerika Selatan (Sao Paulo)	Available
AS Timur (Virginia Utara)	Available
AS Timur (Ohio)	Available
AS Barat (California Utara)	Available
AS Barat (Oregon)	Available
AWS GovCloud (AS-Timur)	Available
AWS GovCloud (AS-Barat)	Available

Amazon RDS for MySQL

Wilayah berikut tersedia untuk integrasi Amazon RDS for MySQL nol-ETL dengan Amazon Redshift.

Region	RDS for MySQL
Africa (Cape Town)	Available

Region	RDS for MySQL
Asia Pasifik (Hong Kong)	Available
Asia Pasifik (Tokyo)	Available
Asia Pasifik (Seoul)	Available
Asia Pasifik (Osaka)	Available
Asia Pasifik (Mumbai)	Available
Asia Pasifik (Hyderabad)	Tidak tersedia
Asia Pasifik (Singapura)	Available
Asia Pasifik (Sydney)	Available
Asia Pasifik (Jakarta)	Tidak tersedia
Asia Pacific (Melbourne)	Tidak tersedia
Asia Pasifik (Malaysia)	Tidak tersedia
Kanada (Pusat)	Available
Kanada Barat (Calgary)	Tidak tersedia
Tiongkok (Beijing)	Tidak tersedia
Tiongkok (Ningxia)	Tidak tersedia
Eropa (Frankfurt)	Available
Europe (Zurich)	Tidak tersedia
Eropa (Stockholm)	Available
Europe (Milan)	Available
Eropa (Spanyol)	Tidak tersedia

Region	RDS for MySQL
Eropa (Irlandia)	Available
Eropa (London)	Available
Eropa (Paris)	Available
Israel (Tel Aviv)	Tidak tersedia
Timur Tengah (UAE)	Tidak tersedia
Timur Tengah (Bahrain)	Available
Amerika Selatan (Sao Paulo)	Available
AS Timur (Virginia Utara)	Available
AS Timur (Ohio)	Available
AS Barat (California Utara)	Available
AS Barat (Oregon)	Available
AWS GovCloud (AS-Timur)	Tidak tersedia
AWS GovCloud (AS-Barat)	Tidak tersedia

Aplikasi perusahaan

Wilayah berikut tersedia untuk integrasi nol-ETL aplikasi perusahaan dengan Amazon Redshift.

Region	Aplikasi perusahaan
Africa (Cape Town)	Tidak tersedia
Asia Pasifik (Hong Kong)	Available
Asia Pasifik (Tokyo)	Available

Region	Aplikasi perusahaan
Asia Pasifik (Seoul)	Available
Asia Pasifik (Osaka)	Tidak tersedia
Asia Pasifik (Mumbai)	Tidak tersedia
Asia Pasifik (Hyderabad)	Tidak tersedia
Asia Pasifik (Singapura)	Available
Asia Pasifik (Sydney)	Available
Asia Pasifik (Jakarta)	Tidak tersedia
Asia Pacific (Melbourne)	Tidak tersedia
Asia Pasifik (Malaysia)	Tidak tersedia
Kanada (Pusat)	Available
Kanada Barat (Calgary)	Tidak tersedia
Tiongkok (Beijing)	Tidak tersedia
Tiongkok (Ningxia)	Tidak tersedia
Eropa (Frankfurt)	Available
Europe (Zurich)	Tidak tersedia
Eropa (Stockholm)	Available
Europe (Milan)	Tidak tersedia
Eropa (Spanyol)	Tidak tersedia
Eropa (Irlandia)	Available
Eropa (London)	Available

Region	Aplikasi perusahaan
Eropa (Paris)	Tidak tersedia
Israel (Tel Aviv)	Tidak tersedia
Timur Tengah (UAE)	Tidak tersedia
Timur Tengah (Bahrain)	Tidak tersedia
Amerika Selatan (Sao Paulo)	Available
AS Timur (Virginia Utara)	Available
AS Timur (Ohio)	Available
AS Barat (California Utara)	Tidak tersedia
AS Barat (Oregon)	Available
AWS GovCloud (AS-Timur)	Tidak tersedia
AWS GovCloud (AS-Barat)	Tidak tersedia

Memecahkan masalah integrasi nol-ETL

Gunakan bagian berikut untuk membantu memecahkan masalah yang Anda miliki dengan integrasi nol-ETL.

Memecahkan masalah integrasi nol-ETL dengan Aurora MySQL

Gunakan informasi berikut untuk memecahkan masalah umum dengan integrasi Nol-ETL dengan Aurora MySQL.

Topik

- [Pembuatan integrasi gagal](#)
- [Tabel tidak memiliki kunci utama](#)
- [Tabel Aurora MySQL tidak mereplikasi ke Amazon Redshift](#)
- [Tipe data yang tidak didukung dalam tabel](#)

- [Perintah bahasa manipulasi data gagal](#)
- [Perubahan yang dilacak antar sumber data tidak cocok](#)
- [Otorisasi gagal](#)
- [Jumlah tabel lebih dari 100K atau jumlah skema lebih dari 4950](#)
- [Amazon Redshift tidak dapat memuat data](#)
- [Pengaturan parameter Workgroup salah](#)
- [Database tidak dibuat untuk mengaktifkan integrasi nol-ETL](#)
- [Tabel berada dalam status Resinkronisasi Diperlukan atau Resinkronisasi Inisiasi](#)
- [Integrasi lag tumbuh](#)

Pembuatan integrasi gagal

Jika pembuatan integrasi nol-ETL gagal, status integrasinya adalah. Inactive Pastikan bahwa berikut ini benar untuk cluster Aurora DB sumber Anda:

- Anda membuat cluster Anda di konsol Amazon RDS.
- Cluster Aurora DB sumber Anda menjalankan versi yang didukung. Untuk daftar versi yang didukung, lihat [Wilayah yang Didukung dan engine DB Aurora untuk integrasi Nol-ETL](#) dengan Amazon Redshift. Untuk memvalidasi ini, buka tab Konfigurasi untuk cluster dan periksa versi Engine.
- Anda mengonfigurasi pengaturan parameter binlog dengan benar untuk cluster Anda. Jika parameter binlog Aurora MySQL Anda disetel secara tidak benar atau tidak terkait dengan cluster Aurora DB sumber, pembuatan gagal. Lihat [Mengkonfigurasi parameter cluster DB](#).

Selain itu, pastikan yang berikut ini benar untuk gudang data Amazon Redshift Anda:

- Sensitivitas kasus dihidupkan. Lihat [Aktifkan sensitivitas kasus untuk gudang data Anda](#).
- Anda menambahkan sumber utama dan integrasi resmi yang benar untuk namespace Anda. Lihat [Konfigurasi otorisasi untuk gudang data Amazon Redshift Anda](#).

Tabel tidak memiliki kunci utama

Dalam database tujuan, satu atau beberapa tabel tidak memiliki kunci utama dan tidak dapat disinkronkan.

Untuk mengatasi masalah ini, buka tab Statistik tabel di halaman detail integrasi atau gunakan `SVV_INTEGRATION_TABLE_STATE` untuk melihat tabel yang gagal. Anda dapat menambahkan kunci utama ke tabel dan Amazon Redshift akan menyinkronkan ulang tabel. Atau, meskipun tidak disarankan, Anda dapat menjatuhkan tabel ini di Aurora dan membuat tabel dengan kunci utama. Untuk informasi selengkapnya, lihat [Praktik terbaik Amazon Redshift untuk mendesain tabel](#).

Tabel Aurora MySQL tidak mereplikasi ke Amazon Redshift

Jika Anda tidak melihat satu atau beberapa tabel yang tercermin di Amazon Redshift, Anda dapat menjalankan perintah berikut untuk menyinkronkannya kembali. Ganti *dbname* dengan nama database Amazon Redshift Anda. Dan, ganti *table1* dan *table2* dengan nama-nama tabel yang akan disinkronkan.

```
ALTER DATABASE dbname INTEGRATION REFRESH TABLES table1, table2;
```

Untuk informasi selengkapnya, lihat [MENGUBAH DATABASE](#) di Panduan Pengembang Database Amazon Redshift.

Data Anda mungkin tidak mereplikasi karena satu atau beberapa tabel sumber Anda tidak memiliki kunci utama. Dasbor pemantauan di Amazon Redshift menampilkan status tabel ini sebagai `Failed`, dan status integrasi nol-ETL keseluruhan berubah menjadi `Needs attention`. Untuk mengatasi masalah ini, Anda dapat mengidentifikasi kunci yang ada di tabel Anda yang dapat menjadi kunci utama, atau Anda dapat menambahkan kunci primer sintesis. Untuk solusi mendetail, lihat [Menangani tabel tanpa kunci utama saat membuat Amazon Aurora MySQL atau RDS untuk integrasi MySQL Zero-ETL](#) dengan Amazon Redshift. di Blog AWS Database.

Konfirmasikan juga bahwa jika target Anda adalah cluster Amazon Redshift, kluster tersebut tidak dijeda.

Tipe data yang tidak didukung dalam tabel

Dalam database yang Anda buat dari integrasi di Amazon Redshift dan di mana data direplikasi dari cluster Aurora DB, satu atau beberapa tabel memiliki tipe data yang tidak didukung dan tidak dapat disinkronkan.

Untuk mengatasi masalah ini, buka tab Statistik tabel di halaman detail integrasi atau gunakan `SVV_INTEGRATION_TABLE_STATE` untuk melihat tabel yang gagal. Kemudian, hapus tabel ini dan buat ulang tabel baru di Amazon RDS. Untuk informasi selengkapnya tentang tipe data yang tidak didukung, lihat [Perbedaan tipe data antara database Aurora dan Amazon Redshift di Panduan Pengguna Amazon Aurora](#).

Perintah bahasa manipulasi data gagal

Amazon Redshift tidak dapat menjalankan perintah DHTML pada tabel Redshift. Untuk mengatasi masalah ini, gunakan `SVV_INTEGRATION_TABLE_STATE` untuk melihat tabel yang gagal. Amazon Redshift secara otomatis menyinkronkan ulang tabel untuk mengatasi kesalahan ini.

Perubahan yang dilacak antar sumber data tidak cocok

Kesalahan ini terjadi ketika perubahan antara Amazon Aurora dan Amazon Redshift tidak cocok, yang menyebabkan integrasi memasuki status. `Failed`

Untuk mengatasi ini, hapus integrasi nol-ETL dan buat lagi di Amazon RDS. [Untuk informasi selengkapnya, lihat Membuat integrasi Nol-ETL dan Menghapus integrasi Nol-ETL.](#)

Otorisasi gagal

Otorisasi gagal karena sumber cluster Aurora DB telah dihapus sebagai sumber integrasi resmi untuk gudang data Amazon Redshift.

Untuk mengatasi masalah ini, hapus integrasi nol-ETL dan buat lagi di Amazon RDS. [Untuk informasi selengkapnya, lihat Membuat integrasi Nol-ETL dan Menghapus integrasi Nol-ETL.](#)

Jumlah tabel lebih dari 100K atau jumlah skema lebih dari 4950

Untuk gudang data tujuan, jumlah tabel lebih dari 100K atau jumlah skema lebih dari 4950. Amazon Aurora tidak dapat mengirim data ke Amazon Redshift. Jumlah tabel dan skema melebihi batas yang ditetapkan. Untuk mengatasi masalah ini, hapus skema atau tabel yang tidak perlu dari database sumber.

Amazon Redshift tidak dapat memuat data

Amazon Redshift tidak dapat memuat data ke integrasi Nol-ETL.

Untuk mengatasi masalah ini, hapus integrasi nol-ETL di Amazon RDS dan buat lagi. [Untuk informasi selengkapnya, lihat Membuat integrasi Nol-ETL dan Menghapus integrasi Nol-ETL.](#)

Pengaturan parameter Workgroup salah

Workgroup Anda tidak mengaktifkan sensitivitas huruf besar/case sensitivity.

Untuk mengatasi masalah ini, buka tab Properties pada halaman detail integrasi, pilih grup parameter, dan aktifkan pengenal peka huruf besar/kecil dari tab Properties. Jika Anda tidak memiliki grup parameter yang ada, buat satu dengan pengenal peka huruf besar/kecil diaktifkan. Kemudian,

buat integrasi nol-ETL baru di Amazon RDS. Untuk informasi selengkapnya, lihat [Membuat integrasi nol-ETL](#).

Database tidak dibuat untuk mengaktifkan integrasi nol-ETL

Tidak ada database yang dibuat untuk integrasi nol-ETL untuk mengaktifkannya.

Untuk mengatasi masalah ini, buat database untuk integrasi. Untuk informasi selengkapnya, lihat [Membuat database tujuan di Amazon Redshift](#).

Tabel berada dalam status Resinkronisasi Diperlukan atau Resinkronisasi Inisiasi

Tabel Anda berada dalam status Resinkronisasi Diperlukan atau Resinkronisasi Inisiasi.

Untuk mengumpulkan informasi kesalahan yang lebih rinci tentang mengapa tabel Anda berada dalam status itu, gunakan tampilan [sistem SYS_LOAD_ERROR_DETAIL](#).

Integrasi lag tumbuh

Keterlambatan integrasi integrasi nol-ETL Anda dapat tumbuh jika ada banyak penggunaan SAVEPOINT dalam database sumber Anda.

Memecahkan masalah integrasi nol-ETL dengan Aurora PostgreSQL

Gunakan informasi berikut untuk memecahkan masalah umum dengan integrasi nol-ETL dengan Aurora PostgreSQL.

Topik

- [Pembuatan integrasi gagal](#)
- [Tabel tidak memiliki kunci utama](#)
- [Tabel Aurora PostgreSQL tidak bereplikasi ke Amazon Redshift](#)
- [Tipe data yang tidak didukung dalam tabel](#)
- [Perintah bahasa manipulasi data gagal](#)
- [Perubahan yang dilacak antar sumber data tidak cocok](#)
- [Otorisasi gagal](#)
- [Jumlah tabel lebih dari 100K atau jumlah skema lebih dari 4950](#)
- [Amazon Redshift tidak dapat memuat data](#)
- [Pengaturan parameter Workgroup salah](#)
- [Database tidak dibuat untuk mengaktifkan integrasi nol-ETL](#)

- [Tabel berada dalam status Resinkronisasi Diperlukan atau Resinkronisasi Inisiasi](#)

Pembuatan integrasi gagal

Jika pembuatan integrasi nol-ETL gagal, status integrasinya adalah. Inactive Pastikan bahwa berikut ini benar untuk cluster Aurora DB sumber Anda:

- Anda membuat cluster Anda di konsol Amazon RDS.
- Cluster Aurora DB sumber Anda menjalankan versi yang didukung. Untuk daftar versi yang didukung, lihat [Wilayah yang Didukung dan engine DB Aurora untuk integrasi Nol-ETL](#) dengan Amazon Redshift. Untuk memvalidasi ini, buka tab Konfigurasi untuk cluster dan periksa versi Engine.
- Anda mengonfigurasi pengaturan parameter binlog dengan benar untuk cluster Anda. Jika parameter binlog Aurora PostgreSQL Anda disetel secara tidak benar atau tidak terkait dengan cluster Aurora DB sumber, pembuatan gagal. Lihat [Mengkonfigurasi parameter cluster DB](#).

Selain itu, pastikan yang berikut ini benar untuk gudang data Amazon Redshift Anda:

- Sensitivitas kasus dihidupkan. Lihat [Aktifkan sensitivitas kasus untuk gudang data Anda](#).
- Anda menambahkan sumber utama dan integrasi resmi yang benar untuk endterm=" zero-etl-using .redshift-iam.title" />.

Tabel tidak memiliki kunci utama

Dalam database tujuan, satu atau beberapa tabel tidak memiliki kunci utama dan tidak dapat disinkronkan.

Untuk mengatasi masalah ini, buka tab Statistik tabel di halaman detail integrasi atau gunakan SVV_INTEGRATION_TABLE_STATE untuk melihat tabel yang gagal. Anda dapat menambahkan kunci utama ke tabel dan Amazon Redshift akan menyinkronkan ulang tabel. Atau, meskipun tidak disarankan, Anda dapat menjatuhkan tabel ini di Aurora dan membuat tabel dengan kunci utama. Untuk informasi selengkapnya, lihat [Praktik terbaik Amazon Redshift untuk mendesain tabel](#).

Tabel Aurora PostgreSQL tidak bereplikasi ke Amazon Redshift

Jika Anda tidak melihat satu atau beberapa tabel yang tercermin di Amazon Redshift, Anda dapat menjalankan perintah berikut untuk menyinkronkannya kembali. Ganti *dbname* dengan nama

database Amazon Redshift Anda. Dan, ganti *table1* dan *table2* dengan nama-nama tabel yang akan disinkronkan.

```
ALTER DATABASE dbname INTEGRATION REFRESH TABLES table1, table2;
```

Untuk informasi selengkapnya, lihat [MENGUBAH DATABASE](#) di Panduan Pengembang Database Amazon Redshift.

Data Anda mungkin tidak mereplikasi karena satu atau beberapa tabel sumber Anda tidak memiliki kunci utama. Dasbor pemantauan di Amazon Redshift menampilkan status tabel ini sebagai *Failed*, dan status integrasi nol-ETL keseluruhan berubah menjadi *Needs attention*. Untuk mengatasi masalah ini, Anda dapat mengidentifikasi kunci yang ada di tabel Anda yang dapat menjadi kunci utama, atau Anda dapat menambahkan kunci primer sintetis. Untuk solusi mendetail, lihat [Menangani tabel tanpa kunci utama saat membuat integrasi Amazon Aurora PostgreSQL Zero-ETL dengan Amazon Redshift](#) di Blog AWS Database.

Konfirmasikan juga bahwa jika target Anda adalah cluster Amazon Redshift, klaster tersebut tidak dijeda.

Tipe data yang tidak didukung dalam tabel

Dalam database yang Anda buat dari integrasi di Amazon Redshift dan di mana data direplikasi dari cluster Aurora DB, satu atau beberapa tabel memiliki tipe data yang tidak didukung dan tidak dapat disinkronkan.

Untuk mengatasi masalah ini, buka tab Statistik tabel di halaman detail integrasi atau gunakan `SVV_INTEGRATION_TABLE_STATE` untuk melihat tabel yang gagal. Kemudian, hapus tabel ini dan buat ulang tabel baru di Amazon RDS. Untuk informasi selengkapnya tentang tipe data yang tidak didukung, lihat [Perbedaan tipe data antara database Aurora dan Amazon Redshift di Panduan Pengguna Amazon Aurora](#).

Perintah bahasa manipulasi data gagal

Amazon Redshift tidak dapat menjalankan perintah DHTML pada tabel Redshift. Untuk mengatasi masalah ini, gunakan `SVV_INTEGRATION_TABLE_STATE` untuk melihat tabel yang gagal. Amazon Redshift secara otomatis menyinkronkan ulang tabel untuk mengatasi kesalahan ini.

Perubahan yang dilacak antar sumber data tidak cocok

Kesalahan ini terjadi ketika perubahan antara Amazon Aurora dan Amazon Redshift tidak cocok, yang menyebabkan integrasi memasuki status *Failed*.

Untuk mengatasi ini, hapus integrasi nol-ETL dan buat lagi di Amazon RDS. [Untuk informasi selengkapnya, lihat Membuat integrasi Nol-ETL dan Menghapus integrasi Nol-ETL.](#)

Otorisasi gagal

Otorisasi gagal karena sumber cluster Aurora DB telah dihapus sebagai sumber integrasi resmi untuk gudang data Amazon Redshift.

Untuk mengatasi masalah ini, hapus integrasi nol-ETL dan buat lagi di Amazon RDS. [Untuk informasi selengkapnya, lihat Membuat integrasi Nol-ETL dan Menghapus integrasi Nol-ETL.](#)

Jumlah tabel lebih dari 100K atau jumlah skema lebih dari 4950

Untuk gudang data tujuan, jumlah tabel lebih dari 100K atau jumlah skema lebih dari 4950. Amazon Aurora tidak dapat mengirim data ke Amazon Redshift. Jumlah tabel dan skema melebihi batas yang ditetapkan. Untuk mengatasi masalah ini, hapus skema atau tabel yang tidak perlu dari database sumber.

Amazon Redshift tidak dapat memuat data

Amazon Redshift tidak dapat memuat data ke integrasi Nol-ETL.

Untuk mengatasi masalah ini, hapus integrasi nol-ETL di Amazon RDS dan buat lagi. [Untuk informasi selengkapnya, lihat Membuat integrasi Nol-ETL dan Menghapus integrasi Nol-ETL.](#)

Pengaturan parameter Workgroup salah

Workgroup Anda tidak mengaktifkan sensitivitas huruf besar/case sensitivity.

Untuk mengatasi masalah ini, buka tab Properties pada halaman detail integrasi, pilih grup parameter, dan aktifkan pengenalan peka huruf besar/kecil dari tab Properties. Jika Anda tidak memiliki grup parameter yang ada, buat satu dengan pengenalan peka huruf besar/kecil diaktifkan. Kemudian, buat integrasi nol-ETL baru di Amazon RDS. Untuk informasi selengkapnya, lihat [Membuat integrasi nol-ETL](#).

Database tidak dibuat untuk mengaktifkan integrasi nol-ETL

Tidak ada database yang dibuat untuk integrasi nol-ETL untuk mengaktifkannya.

Untuk mengatasi masalah ini, buat database untuk integrasi. Untuk informasi selengkapnya, lihat [Membuat database tujuan di Amazon Redshift](#).

Tabel berada dalam status Resinkronisasi Diperlukan atau Resinkronisasi Inisiasi

Tabel Anda berada dalam status Resinkronisasi Diperlukan atau Resinkronisasi Inisiasi.

Untuk mengumpulkan informasi kesalahan yang lebih rinci tentang mengapa tabel Anda berada dalam status itu, gunakan tampilan [sistem SYS_LOAD_ERROR_DETAIL](#).

Memecahkan masalah integrasi nol-ETL dengan RDS untuk MySQL

Gunakan informasi berikut untuk memecahkan masalah umum dengan integrasi nol-ETL dengan RDS untuk MySQL.

Topik

- [Pembuatan integrasi gagal](#)
- [Tabel tidak memiliki kunci utama](#)
- [RDS untuk tabel MySQL tidak mereplikasi ke Amazon Redshift](#)
- [Tipe data yang tidak didukung dalam tabel](#)
- [Perintah bahasa manipulasi data gagal](#)
- [Perubahan yang dilacak antar sumber data tidak cocok](#)
- [Otorisasi gagal](#)
- [Jumlah tabel lebih dari 100K atau jumlah skema lebih dari 4950](#)
- [Amazon Redshift tidak dapat memuat data](#)
- [Pengaturan parameter Workgroup salah](#)
- [Database tidak dibuat untuk mengaktifkan integrasi nol-ETL](#)
- [Tabel berada dalam status Resinkronisasi Diperlukan atau Resinkronisasi Inisiasi](#)

Pembuatan integrasi gagal

Jika pembuatan integrasi nol-ETL gagal, status integrasinya adalah. Inactive Pastikan bahwa berikut ini benar untuk instans RDS DB sumber Anda:

- Anda membuat instans Anda di konsol Amazon RDS.
- Instance RDS DB sumber Anda menjalankan versi RDS yang didukung untuk MySQL. Untuk daftar versi yang didukung, lihat [Wilayah yang Didukung dan engine DB untuk integrasi Amazon RDS](#)

[Zero-ETL dengan Amazon Redshift](#). Untuk memvalidasi ini, buka tab Konfigurasi untuk instance dan periksa versi Engine.

- Anda mengonfigurasi pengaturan parameter binlog dengan benar untuk instance Anda. Jika parameter binlog RDS untuk MySQL Anda disetel secara tidak benar atau tidak terkait dengan instans RDS DB sumber, pembuatan gagal. Lihat [Mengkonfigurasi parameter instans DB](#).

Selain itu, pastikan yang berikut ini benar untuk gudang data Amazon Redshift Anda:

- Sensitivitas kasus dihidupkan. Lihat [Aktifkan sensitivitas kasus untuk gudang data Anda](#).
- Anda menambahkan sumber utama dan integrasi resmi yang benar untuk namespace Anda. Lihat [Konfigurasi otorisasi untuk gudang data Amazon Redshift Anda](#).

Tabel tidak memiliki kunci utama

Dalam database tujuan, satu atau beberapa tabel tidak memiliki kunci utama dan tidak dapat disinkronkan.

Untuk mengatasi masalah ini, buka tab Statistik tabel di halaman detail integrasi atau gunakan `SVV_INTEGRATION_TABLE_STATE` untuk melihat tabel yang gagal. Anda dapat menambahkan kunci utama ke tabel dan Amazon Redshift akan menyinkronkan ulang tabel. Atau, meskipun tidak disarankan, Anda dapat menjatuhkan tabel ini pada RDS dan membuat tabel dengan kunci utama. Untuk informasi selengkapnya, lihat [Praktik terbaik Amazon Redshift untuk mendesain tabel](#).

RDS untuk tabel MySQL tidak mereplikasi ke Amazon Redshift

Jika Anda tidak melihat satu atau beberapa tabel yang tercermin di Amazon Redshift, Anda dapat menjalankan perintah berikut untuk menyinkronkannya kembali. Ganti *dbname* dengan nama database Amazon Redshift Anda. Dan, ganti *table1* dan *table2* dengan nama-nama tabel yang akan disinkronkan.

```
ALTER DATABASE dbname INTEGRATION REFRESH TABLES table1, table2;
```

Untuk informasi selengkapnya, lihat [MENGUBAH DATABASE](#) di Panduan Pengembang Database Amazon Redshift.

Data Anda mungkin tidak mereplikasi karena satu atau beberapa tabel sumber Anda tidak memiliki kunci utama. Dasbor pemantauan di Amazon Redshift menampilkan status tabel ini sebagai `Failed`, dan status integrasi nol-ETL keseluruhan berubah menjadi `Needs attention` Untuk mengatasi

masalah ini, Anda dapat mengidentifikasi kunci yang ada di tabel Anda yang dapat menjadi kunci utama, atau Anda dapat menambahkan kunci primer sintetis. Untuk solusi mendetail, lihat [Menangani tabel tanpa kunci utama saat membuat Aurora MySQL Compatible Edition atau RDS untuk integrasi MySQL Zero-ETL dengan Amazon Redshift](#). di Blog AWS Database.

Konfirmasikan juga bahwa jika target Anda adalah cluster Amazon Redshift, klaster tersebut tidak dijeda.

Tipe data yang tidak didukung dalam tabel

Dalam database yang Anda buat dari integrasi di Amazon Redshift dan di mana data direplikasi dari instans RDS DB, satu atau beberapa tabel memiliki tipe data yang tidak didukung dan tidak dapat disinkronkan.

Untuk mengatasi masalah ini, buka tab Statistik tabel di halaman detail integrasi atau gunakan SVV_INTEGRATION_TABLE_STATE untuk melihat tabel yang gagal. Kemudian, hapus tabel ini dan buat ulang tabel baru di Amazon RDS. Untuk informasi selengkapnya tentang tipe data yang tidak didukung, lihat [Perbedaan tipe data antara database RDS dan Amazon Redshift](#) di Panduan Pengguna Amazon RDS.

Perintah bahasa manipulasi data gagal

Amazon Redshift tidak dapat menjalankan perintah DHTML pada tabel Redshift. Untuk mengatasi masalah ini, gunakan SVV_INTEGRATION_TABLE_STATE untuk melihat tabel yang gagal. Amazon Redshift secara otomatis menyinkronkan ulang tabel untuk mengatasi kesalahan ini.

Perubahan yang dilacak antar sumber data tidak cocok

Kesalahan ini terjadi ketika perubahan antara Amazon Aurora dan Amazon Redshift tidak cocok, yang menyebabkan integrasi memasuki status. Failed

Untuk mengatasi ini, hapus integrasi nol-ETL dan buat lagi di Amazon RDS. [Untuk informasi selengkapnya, lihat Membuat integrasi Nol-ETL dan Menghapus integrasi Nol-ETL](#).

Otorisasi gagal

Otorisasi gagal karena instans RDS DB sumber telah dihapus sebagai sumber integrasi resmi untuk gudang data Amazon Redshift.

Untuk mengatasi masalah ini, hapus integrasi nol-ETL dan buat lagi di Amazon RDS. [Untuk informasi selengkapnya, lihat Membuat integrasi Nol-ETL dan Menghapus integrasi Nol-ETL](#).

Jumlah tabel lebih dari 100K atau jumlah skema lebih dari 4950

Untuk gudang data tujuan, jumlah tabel lebih dari 100K atau jumlah skema lebih dari 4950. Amazon Aurora tidak dapat mengirim data ke Amazon Redshift. Jumlah tabel dan skema melebihi batas yang ditetapkan. Untuk mengatasi masalah ini, hapus skema atau tabel yang tidak perlu dari database sumber.

Amazon Redshift tidak dapat memuat data

Amazon Redshift tidak dapat memuat data ke integrasi Nol-ETL.

Untuk mengatasi masalah ini, hapus integrasi nol-ETL di Amazon RDS dan buat lagi. [Untuk informasi selengkapnya, lihat Membuat integrasi Nol-ETL dan Menghapus integrasi Nol-ETL.](#)

Pengaturan parameter Workgroup salah

Workgroup Anda tidak mengaktifkan sensitivitas huruf besar/case sensitivity.

Untuk mengatasi masalah ini, buka tab Properties pada halaman detail integrasi, pilih grup parameter, dan aktifkan pengenal peka huruf besar/kecil dari tab Properties. Jika Anda tidak memiliki grup parameter yang ada, buat satu dengan pengenal peka huruf besar/kecil diaktifkan. Kemudian, buat integrasi nol-ETL baru di Amazon RDS. Untuk informasi selengkapnya, lihat [Membuat integrasi nol-ETL](#).

Database tidak dibuat untuk mengaktifkan integrasi nol-ETL

Tidak ada database yang dibuat untuk integrasi nol-ETL untuk mengaktifkannya.

Untuk mengatasi masalah ini, buat database untuk integrasi. Untuk informasi selengkapnya, lihat [Membuat database tujuan di Amazon Redshift](#).

Tabel berada dalam status Resinkronisasi Diperlukan atau Resinkronisasi Inisiasi

Tabel Anda berada dalam status Resinkronisasi Diperlukan atau Resinkronisasi Inisiasi.

Untuk mengumpulkan informasi kesalahan yang lebih rinci tentang mengapa tabel Anda berada dalam status itu, gunakan tampilan [sistem SYS_LOAD_ERROR_DETAIL](#).

Memecahkan masalah integrasi nol-ETL dengan DynamoDB

Gunakan informasi berikut untuk memecahkan masalah umum dengan integrasi nol-ETL dengan Amazon DynamoDB.

Topik

- [Pembuatan integrasi gagal](#)
- [Tipe data yang tidak didukung dalam tabel](#)
- [Nama tabel dan atribut yang tidak didukung](#)
- [Otorisasi gagal](#)
- [Amazon Redshift tidak dapat memuat data](#)
- [Pengaturan parameter workgroup atau cluster tidak benar](#)
- [Database tidak dibuat untuk mengaktifkan integrasi nol-ETL](#)
- [Point-in-time pemulihan \(PITR\) tidak diaktifkan pada tabel DynamoDB sumber](#)
- [Akses kunci KMS ditolak](#)
- [Amazon Redshift tidak memiliki akses ke tombol tabel DynamoDB](#)

Pembuatan integrasi gagal

Jika pembuatan integrasi nol-ETL gagal, status integrasinya adalah `Inactive`. Pastikan bahwa berikut ini benar untuk gudang data Amazon Redshift dan tabel DynamoDB sumber Anda:

- Sensitivitas kasus diaktifkan untuk gudang data Anda. Lihat [Mengaktifkan sensitivitas huruf besar/kecil](#) di Panduan Manajemen Amazon Redshift.
- Anda menambahkan sumber utama dan integrasi resmi yang benar untuk namespace Anda di Amazon Redshift. Lihat [Mengonfigurasi otorisasi untuk gudang data Amazon Redshift Anda](#) di Panduan Manajemen Amazon Redshift.
- Anda menambahkan kebijakan berbasis sumber daya yang benar ke tabel DynamoDB sumber. Lihat [Kebijakan dan izin di IAM di Panduan Pengguna IAM](#).

Tipe data yang tidak didukung dalam tabel

Nomor DynamoDB diterjemahkan ke DECIMAL (38,10) di Amazon Redshift. Angka yang melebihi rentang presisi ini secara otomatis diubah menjadi (38,10). Hapus integrasi dan satuan presisi angka, lalu buat ulang integrasi.

Nama tabel dan atribut yang tidak didukung

Amazon Redshift mendukung hingga 127 karakter dan nama atribut. Jika nama panjang, seperti nama tabel DynamoDB atau kunci partisi atau nama kolom kunci sortir gagal integrasi Anda, perbaiki dengan menggunakan nama yang lebih pendek dan buat ulang integrasi.

Otorisasi gagal

Otorisasi dapat gagal ketika tabel DynamoDB sumber dihapus sebagai sumber integrasi resmi untuk gudang data Amazon Redshift.

Untuk mengatasi masalah ini, hapus integrasi nol-ETL, dan buat ulang menggunakan Amazon DynamoDB.

Amazon Redshift tidak dapat memuat data

Amazon Redshift tidak dapat memuat data dari integrasi nol-ETL.

Untuk mengatasi masalah ini, segarkan integrasi dengan ALTER DATABASE.

```
ALTER DATABASE sample_integration_db INTEGRATION REFRESH ALL TABLES
```

Pengaturan parameter workgroup atau cluster tidak benar

Workgroup atau cluster Anda tidak mengaktifkan sensitivitas huruf besar/case sensitivity.

Untuk mengatasi masalah ini, buka tab Properties pada halaman detail integrasi, pilih grup parameter, dan aktifkan pengenal peka huruf besar/kecil dari tab Properties. Jika Anda tidak memiliki grup parameter yang ada, buat satu dengan pengenal peka huruf besar/kecil diaktifkan. Kemudian, buat integrasi nol-ETL baru di DynamoDB. Lihat [Mengaktifkan sensitivitas huruf besar/kecil](#) di Panduan Manajemen Amazon Redshift.

Database tidak dibuat untuk mengaktifkan integrasi nol-ETL

Tidak ada database yang dibuat untuk integrasi nol-ETL untuk mengaktifkannya.

Untuk mengatasi masalah ini, buat database untuk integrasi. Lihat [Membuat database tujuan di Amazon](#) Redshift di Panduan Manajemen Pergeseran Merah Amazon.

Point-in-time pemulihan (PITR) tidak diaktifkan pada tabel DynamoDB sumber

Mengaktifkan PITR diperlukan untuk DynamoDB untuk mengekspor data. Pastikan PITR selalu diaktifkan. Jika Anda pernah mematikan PITR saat integrasi aktif, Anda harus mengikuti instruksi dalam pesan kesalahan dan menyegarkan integrasi menggunakan ALTER DATABASE.

```
ALTER DATABASE sample_integration_db INTEGRATION REFRESH ALL TABLES
```

Akses kunci KMS ditolak

Kunci KMS yang digunakan untuk tabel sumber atau integrasi harus dikonfigurasi dengan izin yang memadai. Untuk informasi tentang enkripsi dan dekripsi tabel, lihat [enkripsi DynamoDB saat istirahat di Panduan Pengembang Amazon DynamoDB](#).

Amazon Redshift tidak memiliki akses ke tombol tabel DynamoDB

Jika enkripsi tabel sumber adalah Kunci yang dikelola AWS, maka beralihlah ke kunci yang dikelola pelanggan Kunci milik AWS atau pelanggan. Jika tabel sudah dienkripsi dengan kunci yang dikelola pelanggan, pastikan kebijakan tidak memiliki kunci kondisi apa pun.

Memecahkan masalah integrasi nol-ETL dengan aplikasi

Gunakan informasi berikut untuk memecahkan masalah umum dengan integrasi Zero-ETL dengan aplikasi, seperti, Salesforce, SAP, dan Zendesk. ServiceNow

Topik

- [Pembuatan integrasi gagal](#)
- [Tabel tidak bereplikasi ke Amazon Redshift](#)
- [Tipe data yang tidak didukung dalam tabel](#)
- [Pengaturan parameter Workgroup salah](#)
- [Database tidak dibuat untuk mengaktifkan integrasi nol-ETL](#)
- [Tabel berada dalam status Resinkronisasi Diperlukan atau Resinkronisasi Inisiasi](#)

Pembuatan integrasi gagal

Jika pembuatan integrasi nol-ETL gagal, status integrasinya adalah. Inactive Pastikan bahwa berikut ini benar untuk gudang data Amazon Redshift Anda:

- Sensitivitas kasus dihidupkan. Lihat [Aktifkan sensitivitas kasus untuk gudang data Anda](#).
- Anda menambahkan sumber utama dan integrasi resmi yang benar untuk namespace Anda. Lihat [Konfigurasi otorisasi untuk gudang data Amazon Redshift Anda](#).

Tabel tidak bereplikasi ke Amazon Redshift

Dalam database tujuan, satu atau beberapa tabel tidak memiliki kunci utama dan tidak dapat disinkronkan.

Untuk mengatasi masalah ini, buka tab Statistik tabel di halaman detail integrasi atau gunakan `SVV_INTEGRATION_TABLE_STATE` untuk melihat tabel yang gagal. Anda dapat menambahkan kunci utama ke tabel dan Amazon Redshift akan menyinkronkan ulang tabel. Anda dapat menjalankan perintah berikut untuk menyinkronkan ulang mereka. Ganti *dbname* dengan nama database Amazon Redshift Anda. Dan, ganti *table1* dan *table2* dengan nama-nama tabel yang akan disinkronkan.

```
ALTER DATABASE dbname INTEGRATION REFRESH TABLES table1, table2;
```

Untuk informasi selengkapnya, lihat [MENGUBAH DATABASE](#) di Panduan Pengembang Database Amazon Redshift.

Tipe data yang tidak didukung dalam tabel

Dalam database yang Anda buat dari integrasi di Amazon Redshift dan di mana data direplikasi dari integrasi nol-ETL dengan aplikasi, satu atau beberapa tabel memiliki tipe data yang tidak didukung dan tidak dapat disinkronkan.

Untuk mengatasi masalah ini, buka tab Statistik tabel di halaman detail integrasi atau gunakan `SVV_INTEGRATION_TABLE_STATE` untuk melihat tabel yang gagal. Kemudian, hapus tabel ini dan buat ulang tabel baru di sumbernya. Untuk informasi selengkapnya, lihat [Integrasi nol-ETL](#) di Panduan Pengembang.AWS Glue

Pengaturan parameter Workgroup salah

Workgroup Anda tidak mengaktifkan sensitivitas huruf besar/case sensitivity.

Untuk mengatasi masalah ini, buka tab Properties pada halaman detail integrasi, pilih grup parameter, dan aktifkan pengenal peka huruf besar/kecil dari tab Properties. Jika Anda tidak memiliki grup parameter yang ada, buat satu dengan pengenal peka huruf besar/kecil diaktifkan. Kemudian, buat integrasi nol-ETL baru. Untuk informasi selengkapnya, lihat [Integrasi nol-ETL](#) di Panduan Pengembang.AWS Glue

Database tidak dibuat untuk mengaktifkan integrasi nol-ETL

Tidak ada database yang dibuat untuk integrasi nol-ETL untuk mengaktifkannya.

Untuk mengatasi masalah ini, buat database untuk integrasi. Untuk informasi selengkapnya, lihat [Membuat database tujuan di Amazon Redshift](#).

Tabel berada dalam status Resinkronisasi Diperlukan atau Resinkronisasi Inisiasi

Tabel Anda berada dalam status Resinkronisasi Diperlukan atau Resinkronisasi Inisiasi.

Untuk mengumpulkan informasi kesalahan yang lebih rinci tentang mengapa tabel Anda berada dalam status itu, gunakan tampilan [sistem SYS_LOAD_ERROR_DETAIL](#).

Kueri database

Untuk menanyakan database yang dihosting oleh cluster Amazon Redshift Anda, Anda memiliki dua opsi:

- Hubungkan ke klaster Anda dan jalankan kueri Konsol Manajemen AWS dengan editor kueri.

Jika Anda menggunakan editor kueri di konsol Amazon Redshift, Anda tidak perlu mengunduh dan menyiapkan aplikasi klien SQL.

- Connect ke cluster Anda melalui alat klien SQL, seperti SQL Workbench/J.

Amazon Redshift mendukung alat klien SQL yang terhubung melalui Java Database Connectivity (JDBC) dan Open Database Connectivity (ODBC). Amazon Redshift tidak menyediakan atau menginstal alat atau pustaka klien SQL apa pun, jadi Anda harus menginstalnya di komputer klien atau instans Amazon EC2 untuk menggunakannya. Anda dapat menggunakan sebagian besar alat klien SQL yang mendukung driver JDBC atau ODBC.

Note

Saat Anda menulis prosedur tersimpan, kami merekomendasikan praktik terbaik untuk mengamankan nilai sensitif:

Jangan membuat kode keras informasi sensitif apa pun dalam logika prosedur yang disimpan. Misalnya, jangan tetapkan kata sandi pengguna dalam pernyataan CREATE USER di badan prosedur yang disimpan. Ini menimbulkan risiko keamanan, karena nilai hard-code dapat dicatat sebagai metadata skema dalam tabel katalog. Sebagai gantinya, berikan nilai sensitif, seperti kata sandi, sebagai argumen ke prosedur yang disimpan, melalui parameter. Untuk informasi selengkapnya tentang prosedur tersimpan, lihat [MEMBUAT PROSEDUR](#) dan [Membuat prosedur tersimpan di Amazon Redshift](#). Untuk informasi selengkapnya tentang tabel katalog, lihat [Tabel katalog sistem](#).

Menghubungkan ke Amazon Redshift

Anda dapat terhubung ke database Anda menggunakan sintaks berikut.

```
cluster-name.account-number.aws-region.redshift.amazonaws.com/database-name
```


Elemen sintaks didefinisikan sebagai berikut.

- `cluster-name`

Nama klaster Anda.

- `account-number`

Pengenal unik yang terkait dengan nomor AWS akun Anda di tertentu AWS Region. Semua cluster yang dibuat oleh akun tertentu di tertentu AWS Region memiliki hal yang sama `account-number`.

- `aws-region`

Kode untuk tempat AWS Region cluster berada.

- `database-name`

Nama database Anda.

Misalnya, string koneksi berikut menentukan `my-db` database dalam `my-cluster` cluster di AWS Region `us-east-1`.

```
my-cluster.123456789012.us-east-1.redshift.amazonaws.com/my-db
```

Menanyakan database menggunakan editor kueri v2

Editor kueri v2 adalah aplikasi klien SQL berbasis web terpisah yang Anda gunakan untuk membuat dan menjalankan kueri di gudang data Amazon Redshift Anda. Editor kueri v2 terutama digunakan untuk mengedit dan menjalankan kueri, memvisualisasikan hasil, dan berbagi pekerjaan Anda dengan tim Anda. Dengan query editor v2, Anda dapat membuat database, skema, tabel, dan fungsi yang ditentukan pengguna (). UDFs Dalam panel tampilan pohon, untuk setiap database Anda, Anda dapat melihat skema. Untuk setiap skema, Anda dapat melihat tabel, tampilan UDFs, dan prosedur tersimpan. Editor kueri v2 adalah pengganti editor kueri sebelumnya.

Note

Editor kueri v2 tersedia dalam komersial Wilayah AWS. Untuk daftar Wilayah AWS di mana editor kueri v2 tersedia, lihat titik akhir yang terdaftar untuk [Redshift query editor](#) v2 di.

Referensi Umum Amazon Web Services

Untuk demo query editor v2, tonton video berikut. [Editor kueri Amazon Redshift v2](#).

Untuk demo analisis data, tonton video berikut. [Analisis data menggunakan editor kueri Amazon Redshift v2](#).

Untuk demo menggunakan editor kueri v2 yang menjalankan beberapa kueri dengan koneksi terisolasi atau bersama, tonton video berikut. [Eksekusi Query Concurrent menggunakan Query Editor v2](#).

Editor kueri v2 memiliki serangkaian fitur yang kaya untuk mengelola dan menjalankan pernyataan SQL Anda. Topik di bagian berikut membantu Anda memulai dengan banyak fitur ini. Jelajahi editor kueri v2 Anda sendiri untuk membiasakan diri dengan kemampuannya.

Mengkonfigurasi Akun AWS

Anda dapat melakukan serangkaian tugas ini untuk mengonfigurasi editor kueri v2 untuk menanyakan database Amazon Redshift. Dengan izin yang tepat, Anda dapat mengakses data di kluster Amazon Redshift atau grup kerja yang dimiliki oleh Anda yang ada di saat ini. Akun AWS AWS Region

Pertama kali administrator mengonfigurasi editor kueri v2 untuk Anda Akun AWS, mereka memilih AWS KMS key yang digunakan untuk mengenkripsi sumber daya editor kueri v2. Secara default, kunci yang AWS dimiliki digunakan untuk mengenkripsi sumber daya. Atau, administrator dapat menggunakan kunci yang dikelola pelanggan dengan memilih Amazon Resource Name (ARN) untuk kunci di halaman konfigurasi.

Setelah mengonfigurasi akun, pengaturan AWS KMS enkripsi tidak dapat diubah. Untuk informasi selengkapnya tentang membuat dan menggunakan kunci yang dikelola pelanggan dengan editor kueri v2, lihat [Membuat kunci yang dikelola AWS KMS pelanggan untuk digunakan dengan editor kueri v2](#). Administrator juga dapat secara opsional memilih bucket dan jalur S3 yang digunakan untuk beberapa fitur, seperti memuat data dari file. Untuk informasi selengkapnya, lihat [Memuat data dari pengaturan file lokal dan alur kerja](#).

Editor kueri Amazon Redshift v2 mendukung autentikasi, enkripsi, isolasi, dan kepatuhan untuk menjaga data Anda tetap diam dan data dalam transit aman. Untuk informasi selengkapnya tentang keamanan data dan editor kueri v2, lihat berikut ini:

- [Enkripsi saat diam](#)
- [Enkripsi saat bergerak](#)
- [Analisis konfigurasi dan kerentanan di Amazon Redshift](#)

AWS CloudTrail menangkap panggilan API dan peristiwa terkait yang dibuat oleh atau atas nama Anda Akun AWS dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Anda dapat mengidentifikasi pengguna dan akun mana yang dipanggil AWS, alamat IP sumber dari mana panggilan dilakukan, dan kapan panggilan terjadi. Untuk mempelajari selengkapnya tentang bagaimana editor kueri v2 berjalan AWS CloudTrail, lihat [Logging dengan CloudTrail](#). Untuk informasi selengkapnya CloudTrail, lihat [Panduan AWS CloudTrail Pengguna](#).

Editor kueri v2 memiliki kuota yang dapat disesuaikan untuk beberapa sumber dayanya. Untuk informasi selengkapnya, lihat [Kuota untuk objek Amazon Redshift](#).

Sumber daya dibuat dengan editor kueri v2

Dalam editor kueri v2, Anda dapat membuat sumber daya seperti kueri dan bagan yang disimpan. Semua sumber daya di editor kueri v2 dikaitkan dengan peran IAM atau dengan pengguna. Sebaiknya Anda melampirkan kebijakan ke peran IAM dan menetapkan peran tersebut kepada pengguna.

Di editor kueri v2, Anda dapat menambahkan dan menghapus tag untuk kueri dan bagan yang disimpan. Anda dapat menggunakan tag ini saat menyiapkan kebijakan IAM khusus atau untuk mencari sumber daya. Anda juga dapat mengelola tag dengan menggunakan Editor AWS Resource Groups Tag.

Anda dapat mengatur peran IAM dengan kebijakan IAM untuk berbagi kueri dengan orang lain di bagian yang sama Akun AWS . AWS Region

Membuat kunci yang dikelola AWS KMS pelanggan untuk digunakan dengan editor kueri v2

Untuk membuat kunci terkelola enkripsi simetris yang dikelola pelanggan:

Anda dapat membuat kunci terkelola pelanggan enkripsi simetris untuk mengenkripsi sumber daya editor kueri v2 menggunakan operasi AWS KMS konsol atau AWS KMS API. Untuk petunjuk tentang membuat kunci, lihat [Membuat AWS KMS kunci enkripsi simetris](#) di Panduan AWS Key Management Service Pengembang.

Kebijakan utama

Kebijakan utama mengontrol akses ke kunci yang dikelola pelanggan Anda. Setiap kunci yang dikelola pelanggan harus memiliki persis satu kebijakan utama, yang berisi pernyataan yang menentukan siapa yang dapat menggunakan kunci dan bagaimana mereka dapat menggunakannya. Saat membuat kunci terkelola pelanggan, Anda dapat menentukan kebijakan kunci. Untuk informasi

selengkapnya, lihat [Mengelola akses ke AWS KMS kunci](#) di Panduan AWS Key Management Service Pengembang.

Untuk menggunakan kunci terkelola pelanggan Anda dengan editor kueri Amazon Redshift v2, operasi API berikut harus diizinkan dalam kebijakan kunci:

- `kms:GenerateDataKey`— Menghasilkan kunci data simetris yang unik untuk mengenkripsi data Anda.
- `kms:Decrypt`— Mendekripsi data yang dienkripsi dengan kunci yang dikelola pelanggan.
- `kms:DescribeKey`— Memberikan detail kunci yang dikelola pelanggan untuk memungkinkan layanan memvalidasi kunci.

Berikut ini adalah contoh AWS KMS kebijakan untuk Akun AWS 111122223333. Di bagian pertama, `kms:ViaService` batas penggunaan kunci untuk layanan editor kueri v2 (yang dinamai `sqlworkbench.region.amazonaws.com` dalam kebijakan). Akun AWS Menggunakan kunci harus111122223333. Di bagian kedua, pengguna root dan administrator kunci Akun AWS 111122223333 dapat mengakses ke kunci.

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang disebut pengguna Akun AWS root yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Untuk tugas yang memerlukan kredensial pengguna root, lihat [Tugas yang memerlukan kredensial pengguna root](#) dalam Panduan Pengguna IAM.

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"
}
]
}

```

Sumber daya berikut memberikan informasi lebih lanjut tentang AWS KMS kunci:

- Untuk informasi selengkapnya tentang AWS KMS kebijakan, lihat [Menentukan izin dalam kebijakan di Panduan AWS Key Management Service](#) Pengembang.
- Untuk informasi tentang AWS KMS kebijakan pemecahan masalah, lihat [Memecahkan masalah akses kunci](#) di Panduan Pengembang.AWS Key Management Service
- Untuk informasi selengkapnya tentang kunci, lihat [kunci AWS KMS](#) di Panduan AWS Key Management Service Pengembang.

Mengakses editor kueri v2

Untuk mengakses editor kueri v2, Anda memerlukan izin. Administrator dapat melampirkan salah satu kebijakan AWS terkelola berikut ke peran untuk memberikan izin. (Kami merekomendasikan untuk melampirkan kebijakan ke peran IAM dan menetapkan peran tersebut kepada pengguna.) Kebijakan AWS terkelola ini ditulis dengan opsi berbeda yang mengontrol bagaimana sumber

daya penandaan memungkinkan berbagi kueri. Anda dapat menggunakan konsol IAM (<https://console.aws.amazon.com/iam/>) untuk melampirkan kebijakan IAM.

- **AmazonRedshiftQueryEditorV2 FullAccess** - Memberikan akses penuh ke operasi dan sumber daya editor kueri Amazon Redshift v2. Kebijakan ini juga memberikan akses ke layanan lain yang diperlukan.
- **AmazonRedshiftQueryEditorV2 NoSharing** - Memberikan kemampuan untuk bekerja dengan editor kueri Amazon Redshift v2 tanpa berbagi sumber daya. Kebijakan ini juga memberikan akses ke layanan lain yang diperlukan.
- **AmazonRedshiftQueryEditorV2 ReadSharing** - Memberikan kemampuan untuk bekerja dengan editor kueri Amazon Redshift v2 dengan berbagi sumber daya terbatas. Prinsipal yang diberikan dapat membaca sumber daya yang dibagikan dengan timnya tetapi tidak dapat memperbaruinya. Kebijakan ini juga memberikan akses ke layanan lain yang diperlukan.
- **AmazonRedshiftQueryEditorV2 ReadWriteSharing** - Memberikan kemampuan untuk bekerja dengan editor kueri Amazon Redshift v2 dengan berbagi sumber daya. Kepala sekolah yang diberikan dapat membaca dan memperbarui sumber daya yang dibagikan dengan timnya. Kebijakan ini juga memberikan akses ke layanan lain yang diperlukan.

Anda juga dapat membuat kebijakan sendiri berdasarkan izin yang diizinkan dan ditolak dalam kebijakan terkelola yang disediakan. Jika Anda menggunakan editor kebijakan konsol IAM untuk membuat kebijakan Anda sendiri, pilih SQL Workbench sebagai layanan yang Anda buat kebijakan di editor visual. Query editor v2 menggunakan nama layanan AWS SQL Workbench di editor visual dan IAM Policy Simulator.

Untuk prinsipal (pengguna dengan peran IAM yang ditetapkan) untuk terhubung ke kluster Amazon Redshift, mereka memerlukan izin di salah satu kebijakan terkelola editor kueri v2. Mereka juga membutuhkan salah satu `redshift:GetClusterCredentialsWithIAM` atau `redshift:GetClusterCredentials` izin ke cluster. Untuk mendapatkan izin ini, seseorang dengan izin administratif dapat melampirkan kebijakan ke peran IAM yang digunakan untuk terhubung ke kluster dengan menggunakan kredensial sementara. Anda dapat membuat cakupan kebijakan ke kluster tertentu atau lebih umum. Untuk informasi selengkapnya tentang izin untuk menggunakan kredensial sementara, lihat [Membuat peran IAM atau pengguna dengan izin untuk memanggil IAM atau. GetClusterCredentialsWith GetClusterCredentials](#)

Untuk prinsipal (biasanya pengguna dengan peran IAM yang ditetapkan) untuk mengaktifkan kemampuan di halaman Pengaturan akun untuk orang lain di akun ke set hasil Ekspor, mereka memerlukan `sqlworkbench:UpdateAccountExportSettings` izin yang dilampirkan peran

tersebut. Izin ini termasuk dalam kebijakan `AmazonRedshiftQueryEditorV2FullAccess` AWS terkelola.

Karena fitur baru ditambahkan ke editor kueri v2, kebijakan AWS terkelola diperbarui sesuai kebutuhan. Jika Anda membuat kebijakan sendiri berdasarkan izin yang diizinkan dan ditolak dalam kebijakan terkelola yang disediakan, edit kebijakan Anda agar tetap up to date dengan perubahan pada kebijakan terkelola. Untuk informasi selengkapnya tentang kebijakan terkelola di Amazon Redshift, lihat. [AWS kebijakan terkelola untuk Amazon Redshift](#)

Untuk memberikan akses dan menambahkan izin bagi pengguna, grup, atau peran Anda:

- Pengguna dan grup di AWS IAM Identity Center:

Buat rangkaian izin. Ikuti instruksi di [Buat rangkaian izin](#) dalam Panduan Pengguna AWS IAM Identity Center .

- Pengguna yang dikelola di IAM melalui penyedia identitas:

Buat peran untuk federasi identitas. Ikuti instruksi dalam [Buat peran untuk penyedia identitas pihak ketiga \(federasi\)](#) dalam Panduan Pengguna IAM.

- Pengguna IAM:

- Buat peran yang dapat diambil pengguna Anda. Ikuti instruksi dalam [Buat peran untuk pengguna IAM](#) dalam Panduan Pengguna IAM.
- (Tidak disarankan) Lampirkan kebijakan langsung ke pengguna atau tambahkan pengguna ke grup pengguna. Ikuti petunjuk dalam [Menambahkan izin ke pengguna \(konsol\)](#) dalam Panduan Pengguna IAM.

Note

Jika AWS IAM Identity Center administrator menghapus semua asosiasi set izin untuk set izin tertentu di seluruh akun, akses ke sumber daya editor kueri yang awalnya terkait dengan kumpulan izin yang dihapus tidak lagi dapat diakses. Jika nanti izin yang sama dibuat ulang, pengenalan internal baru dibuat. Karena pengenalan internal telah berubah, akses ke sumber daya editor kueri yang sebelumnya dimiliki oleh pengguna tidak dapat diakses. Sebaiknya sebelum administrator menghapus set izin, pengguna izin tersebut menetapkan sumber daya editor kueri ekspor seperti buku catatan dan kueri sebagai cadangan.

Menyiapkan tag utama untuk menghubungkan cluster atau workgroup dari editor kueri v2

Untuk terhubung ke klaster atau grup kerja menggunakan opsi pengguna federasi, siapkan peran IAM Anda atau pengguna dengan tag utama. Atau, atur penyedia identitas Anda (iDP) untuk masuk RedshiftDbUser dan (opsional). RedshiftDbGroups Untuk informasi selengkapnya tentang menggunakan IAM untuk mengelola tag, lihat [Melewati tag sesi AWS Security Token Service di Panduan Pengguna IAM](#). Untuk mengatur akses menggunakan AWS Identity and Access Management, administrator dapat menambahkan tag menggunakan konsol IAM (<https://console.aws.amazon.com/iam/>).

Untuk menambahkan tag utama ke peran IAM

1. Masuk ke Konsol Manajemen AWS dan buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Pilih Peran di panel navigasi.
3. Pilih peran yang memerlukan akses ke editor kueri v2 menggunakan pengguna federasi.
4. Pilih tab Tanda.
5. Pilih tag Kelola.
6. Pilih Tambah tag dan masukkan Kunci sebagai RedshiftDbUser dan masukkan Nilai nama pengguna federasi.
7. Secara opsional pilih Tambah tag dan masukkan Kunci sebagai RedshiftDbGroups dan masukkan Nilai nama grup untuk dikaitkan dengan pengguna.
8. Pilih Simpan perubahan untuk melihat daftar tag yang terkait dengan peran IAM yang Anda pilih. Menyebarkan perubahan mungkin memakan waktu beberapa detik.
9. Untuk menggunakan pengguna federasi, segarkan halaman v2 editor kueri Anda setelah perubahan disebarkan.

Siapkan penyedia identitas Anda (iDP) untuk meneruskan tag utama

Prosedur untuk mengatur tag menggunakan penyedia identitas (iDP) bervariasi menurut IDP. Lihat dokumentasi IDP Anda untuk petunjuk tentang cara meneruskan informasi pengguna dan grup ke atribut SAMP. Ketika dikonfigurasi dengan benar, atribut berikut muncul dalam respons SAMP Anda yang digunakan oleh AWS Security Token Service untuk mengisi tag utama untuk RedshiftDbUser dan RedshiftDbGroups


```
<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>
```

Opsional **db_groups** harus berupa daftar yang dipisahkan titik dua seperti.
group1:group2:group3

Selain itu, Anda dapat mengatur `TransitiveTagKeys` atribut untuk mempertahankan tag selama rantai peran.

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/TransitiveTagKeys">
  <AttributeValue>RedshiftDbUser</AttributeValue>
  <AttributeValue>RedshiftDbGroups</AttributeValue>
</Attribute>
```

Untuk informasi selengkapnya tentang menyiapkan editor kueri v2, lihat [Izin yang diperlukan untuk menggunakan editor kueri v2](#).

Note

Saat Anda terhubung ke kluster atau grup kerja menggunakan opsi koneksi pengguna Federasi dari editor kueri v2, Penyedia Identitas (iDP) dapat menyediakan tag utama khusus untuk dan. `RedshiftDbUser` `RedshiftDbGroups` Saat ini, AWS IAM Identity Center doesn't support tag utama khusus yang diteruskan langsung ke editor kueri v2.

Membuka editor kueri v2

Dengan Amazon Redshift, Anda dapat menjalankan kueri SQL terhadap kluster gudang data menggunakan editor kueri v2 di konsol Amazon Redshift. Editor kueri v2 adalah alat berbasis web yang menyediakan antarmuka yang ramah pengguna untuk menjalankan kueri ad-hoc, menjelajahi data, dan melakukan tugas analisis data. Bagian berikut memandu Anda melalui proses membuka editor kueri v2 di konsol dan memanfaatkan fungsinya secara efektif.

Untuk membuka editor kueri v2

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Dari menu navigator, pilih Editor, lalu Query editor V2. Editor kueri v2 terbuka di tab browser baru.

Halaman editor kueri memiliki menu navigator tempat Anda memilih tampilan sebagai berikut:

Penyunting



Anda mengelola dan menanyakan data Anda diatur sebagai tabel dan terkandung dalam database. Basis data dapat berisi data yang disimpan atau berisi referensi ke data yang disimpan di tempat lain, seperti Amazon S3. Anda terhubung ke database yang terdapat dalam kluster atau grup kerja tanpa server.

Saat bekerja di tampilan Editor, Anda memiliki kontrol berikut:

- Bidang Cluster atau Workgroup menampilkan nama yang saat ini Anda sambungkan. Bidang Database menampilkan database dalam cluster atau workgroup. Tindakan yang Anda lakukan dalam tampilan Database default untuk bertindak pada database yang telah Anda pilih.
- Tampilan hierarkis tampilan pohon dari cluster atau kelompok kerja, database, dan skema Anda. Di bawah skema, Anda dapat bekerja dengan tabel, tampilan, fungsi, dan prosedur tersimpan. Setiap objek dalam tampilan pohon mendukung menu konteks untuk melakukan tindakan terkait, seperti Refresh atau Drop, untuk objek.

- Tindakan



Buat untuk membuat database, skema, tabel, dan fungsi.

- Tindakan



Muat data untuk memuat data dari Amazon S3 atau dari file lokal ke database Anda.

- Ikon



Simpan untuk menyimpan kueri Anda.

- Ikon



Pintasan untuk menampilkan pintasan keyboard untuk editor.

- Ikon



Lainnya untuk menampilkan lebih banyak tindakan di editor. Seperti:

- Bagikan dengan tim saya untuk membagikan kueri atau buku catatan dengan tim Anda. Untuk informasi selengkapnya, lihat [Berkolaborasi dan berbagi sebagai sebuah tim](#).
- Pintasan untuk menampilkan pintasan keyboard untuk editor.
- Riwayat tab untuk menampilkan riwayat tab di editor.
- Segarkan pelengkapan otomatis untuk menyegarkan saran yang ditampilkan saat membuat SQL.

- Area



Editor tempat Anda dapat memasukkan dan menjalankan kueri Anda.

Setelah Anda menjalankan kueri, tab Hasil muncul dengan hasil. Di sinilah Anda dapat mengaktifkan Bagan untuk memvisualisasikan hasil Anda. Anda juga dapat mengekspor hasil Anda.

- Area



Notebook tempat Anda dapat menambahkan bagian untuk masuk dan menjalankan SQL atau menambahkan Markdown.

Setelah Anda menjalankan kueri, tab Hasil muncul dengan hasil. Di sinilah Anda dapat Mengekspor hasil Anda.

Pertanyaan



Kueri berisi perintah SQL untuk mengelola dan menanyakan data Anda dalam database. Saat Anda menggunakan editor kueri v2 untuk memuat data sampel, itu juga membuat dan menyimpan kueri sampel untuk Anda.

Ketika Anda memilih kueri yang disimpan, Anda dapat membuka, mengganti nama, dan menghapusnya menggunakan menu konteks (klik kanan). Anda dapat melihat atribut seperti ARN Kueri dari kueri yang disimpan dengan memilih Detail kueri. Anda juga dapat melihat riwayat versinya, mengedit tag yang dilampirkan ke kueri, dan membagikannya dengan tim Anda.

Notebook



Notebook SQL berisi sel SQL dan Markdown. Gunakan buku catatan untuk mengatur, membubuhi keterangan, dan berbagi beberapa perintah SQL dalam satu dokumen.

Ketika Anda memilih buku catatan yang disimpan, Anda dapat membuka, mengganti nama, menggandakan, dan menghapusnya menggunakan menu konteks (klik kanan). Anda dapat melihat atribut seperti ARN Notebook buku catatan yang disimpan dengan memilih detail Notebook. Anda juga dapat melihat riwayat versinya, mengedit tag yang dilampirkan ke buku catatan, mengeksponnya, dan membagikannya dengan tim Anda. Untuk informasi selengkapnya, lihat [Notebook di Amazon Redshift](#).

Grafik



Bagan adalah representasi visual dari data Anda. Editor kueri v2 menyediakan alat untuk membuat banyak jenis bagan dan menyimpannya.

Saat Anda memilih bagan yang disimpan, Anda dapat membuka, mengganti nama, dan menghapusnya menggunakan menu konteks (klik kanan). Anda dapat melihat atribut seperti Bagan ARN dari bagan yang disimpan dengan memilih Detail bagan. Anda juga dapat mengedit tag yang dilampirkan pada bagan dan mengeksponnya. Untuk informasi selengkapnya, lihat [Memvisualisasikan hasil kueri](#).

Sejarah



Riwayat kueri adalah daftar kueri yang Anda jalankan menggunakan editor kueri Amazon Redshift v2. Kueri ini berjalan sebagai kueri individual atau sebagai bagian dari notebook SQL. Untuk informasi selengkapnya, lihat [Melihat kueri dan riwayat tab](#).

Pertanyaan terjadwal



Kueri terjadwal adalah kueri yang diatur untuk memulai pada waktu tertentu.

Semua tampilan editor kueri v2 memiliki ikon berikut:

- Ikon mode



Visual untuk beralih antara mode terang dan mode gelap.

- Ikon



Pengaturan untuk menampilkan menu layar pengaturan yang berbeda.

- Ikon preferensi



Editor untuk mengedit preferensi Anda saat Anda menggunakan editor kueri v2. Di sini Anda dapat Mengedit pengaturan ruang kerja untuk mengubah ukuran font, ukuran tab, dan pengaturan tampilan lainnya. Anda juga dapat mengaktifkan (atau menonaktifkan) Pelengkapan Otomatis untuk menampilkan saran saat Anda memasukkan SQL Anda.

- Ikon



Connections untuk melihat koneksi yang digunakan oleh tab editor Anda.

Koneksi digunakan untuk mengambil data dari database. Koneksi dibuat untuk database tertentu. Dengan koneksi terisolasi, hasil perintah SQL yang mengubah database, seperti membuat tabel sementara, dalam satu tab editor, tidak terlihat di tab editor lain. Saat Anda membuka tab editor di editor kueri v2, defaultnya adalah koneksi terisolasi. Saat Anda membuat koneksi bersama, yaitu mematikan sakelar sesi terisolasi, maka hasil koneksi bersama lainnya ke database yang sama akan terlihat satu sama lain. Namun, tab editor yang menggunakan koneksi bersama ke database tidak berjalan secara paralel. Pertanyaan yang menggunakan koneksi yang sama harus menunggu hingga koneksi tersedia. Koneksi ke satu database tidak dapat dibagikan dengan database lain, dan dengan demikian hasil SQL tidak terlihat di seluruh koneksi database yang berbeda.

Jumlah koneksi yang dapat diaktifkan oleh pengguna di akun dikendalikan oleh administrator editor kueri v2.

- Ikon Pengaturan



akun yang digunakan oleh administrator untuk mengubah pengaturan tertentu dari semua pengguna di akun. Untuk informasi selengkapnya, lihat [Pengaturan akun](#).

Pertimbangan saat bekerja dengan editor kueri v2

Pertimbangkan hal berikut saat bekerja dengan editor kueri v2.

- Durasi maksimum kueri adalah 24 jam.
- Ukuran hasil kueri maksimum adalah 100 MB. Jika panggilan mengembalikan lebih dari 100 MB data respons, 100 MB pertama dikembalikan dengan peringatan.
- Anda dapat menjalankan kueri hingga 300.000 karakter.
- Anda dapat menyimpan kueri hingga 30.000 karakter.
- Secara default, editor kueri v2 secara otomatis melakukan setiap perintah SQL individual yang berjalan. Ketika pernyataan BEGIN disediakan, pernyataan dalam blok BEGIN-COMMIT atau BEGIN-ROLLBACK dijalankan sebagai satu transaksi. Untuk informasi selengkapnya tentang transaksi, lihat [MULAI](#) di Panduan Pengembang Database Amazon Redshift.
- Jumlah maksimum peringatan yang ditampilkan oleh editor kueri v2 saat menjalankan pernyataan SQL adalah 10. Misalnya, ketika prosedur tersimpan dijalankan, tidak lebih dari 10 pernyataan RAISE ditampilkan.
- Editor kueri v2 tidak mendukung IAM RoleSessionName yang berisi koma (.).
Anda mungkin melihat kesalahan yang mirip dengan berikut ini: Pesan Kesalahan:
“AROIA123456789Example:MyText, yourtext” bukan nilai yang valid untuk TagValue - ini berisi karakter ilegal” Masalah ini muncul saat Anda mendefinisikan IAM RoleSessionName yang menyertakan koma dan kemudian menggunakan editor kueri v2 dengan peran IAM itu.

Untuk informasi selengkapnya tentang IAMRoleSessionName, lihat [atribut RoleSessionName SAMP](#) di Panduan Pengguna IAM.

Pengaturan akun

Pengguna dengan izin IAM yang tepat dapat melihat dan mengubah pengaturan Akun untuk pengguna lain dalam hal yang sama. Akun AWS Administrator ini dapat melihat atau mengatur hal-hal berikut:

- Koneksi database bersamaan maksimum per pengguna di akun. Ini termasuk koneksi untuk sesi Terisolasi. Saat Anda mengubah nilai ini, perlu waktu 10 menit agar perubahan diterapkan.
- Izinkan pengguna di akun untuk mengekspor seluruh hasil yang ditetapkan dari perintah SQL ke file.
- Muat dan tampilkan database sampel dengan beberapa kueri tersimpan terkait.
- Tentukan jalur Amazon S3 yang digunakan oleh pengguna akun untuk memuat data dari file lokal.
- Lihat ARN kunci KMS yang digunakan untuk mengenkripsi sumber daya editor kueri v2.

Menghubungkan ke database Amazon Redshift

Untuk menyambung ke database, pilih nama cluster atau workgroup di panel tampilan pohon. Jika diminta, masukkan parameter koneksi.

Ketika Anda terhubung ke cluster atau workgroup dan database-nya, Anda biasanya memberikan nama Database. Anda juga menyediakan parameter yang diperlukan untuk salah satu metode otentikasi berikut:

Pusat Identitas IAM

Dengan metode ini, sambungkan ke gudang data Amazon Redshift Anda dengan kredensi masuk tunggal Anda dari penyedia identitas (iDP) Anda. Cluster atau workgroup Anda harus diaktifkan untuk IAM Identity Center di konsol Amazon Redshift. Untuk bantuan menyiapkan koneksi ke Pusat Identitas IAM, lihat [Hubungkan Redshift dengan AWS IAM Identity Center untuk pengalaman masuk tunggal](#).

Pengguna federasi

Dengan metode ini, tag utama peran IAM atau pengguna Anda harus memberikan detail koneksi. Anda mengonfigurasi tag ini di AWS Identity and Access Management atau penyedia identitas Anda (iDP). Editor kueri v2 bergantung pada tag berikut.

- `RedshiftDbUserTag` ini mendefinisikan pengguna database yang digunakan oleh query editor v2. Tag ini diperlukan.

- **RedshiftDbGroups**— Tag ini mendefinisikan grup database yang bergabung saat menghubungkan ke editor kueri v2. Tag ini opsional dan nilainya harus berupa daftar yang dipisahkan titik dua seperti. `group1:group2:group3` Nilai kosong diabaikan, yaitu, `group1:::group2` ditafsirkan sebagai `group1:group2`.

Tag ini diteruskan ke `redshift:GetClusterCredentials` API untuk mendapatkan kredensial untuk klaster Anda. Untuk informasi selengkapnya, lihat [Menyiapkan tag utama untuk menghubungkan cluster atau workgroup dari editor kueri v2](#).

Kredensi sementara menggunakan nama pengguna database

Opsi ini hanya tersedia saat menghubungkan ke cluster. Dengan metode ini, query editor v2, memberikan nama pengguna untuk database. Editor kueri v2 menghasilkan kata sandi sementara untuk terhubung ke database sebagai nama pengguna database Anda. Seorang pengguna yang menggunakan metode ini untuk terhubung harus diizinkan izin IAM untuk `redshift:GetClusterCredentials`. Untuk mencegah pengguna menggunakan metode ini, ubah pengguna atau peran IAM mereka untuk menolak izin ini.

Kredensi sementara menggunakan identitas IAM Anda

Opsi ini hanya tersedia saat menghubungkan ke cluster. Dengan metode ini, editor kueri v2 memetakan nama pengguna ke identitas IAM Anda dan menghasilkan kata sandi sementara untuk terhubung ke database sebagai identitas IAM Anda. Seorang pengguna yang menggunakan metode ini untuk terhubung harus diizinkan izin IAM untuk `redshift:GetClusterCredentialsWithIAM`. Untuk mencegah pengguna menggunakan metode ini, ubah pengguna atau peran IAM mereka untuk menolak izin ini.

Nama pengguna dan kata sandi basis data

Dengan metode ini, berikan juga nama Pengguna dan Kata Sandi untuk database yang Anda sambungkan. Editor kueri v2 membuat rahasia atas nama Anda yang disimpan di AWS Secrets Manager. Rahasia ini berisi kredensial untuk terhubung ke database Anda.

AWS Secrets Manager

Dengan metode ini, alih-alih nama database, Anda memberikan Rahasia yang disimpan di Secrets Manager yang berisi database dan kredensi login Anda. Untuk informasi tentang membuat rahasia, lihat [Membuat rahasia untuk kredensi koneksi database](#).

Ketika Anda memilih klaster atau workgroup dengan editor kueri v2, tergantung pada konteksnya, Anda dapat membuat, mengedit, dan menghapus koneksi menggunakan menu konteks (klik kanan).

Anda dapat melihat atribut seperti ARN Koneksi koneksi dengan memilih Detail koneksi. Anda juga dapat mengedit tag yang dilampirkan ke koneksi.

Menjelajahi database Amazon Redshift

Dalam database, Anda dapat mengelola skema, tabel, tampilan, fungsi, dan prosedur tersimpan di panel tampilan pohon. Setiap objek dalam tampilan memiliki tindakan yang terkait dengannya dalam menu konteks (klik kanan).

Panel tampilan pohon hierarkis menampilkan objek database. Untuk me-refresh panel tampilan pohon untuk menampilkan objek database yang mungkin telah dibuat setelah tampilan pohon terakhir ditampilkan, pilih ikon.



Buka menu konteks (klik kanan) untuk objek untuk melihat tindakan apa yang dapat Anda lakukan.

▼  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	

Setelah Anda memilih tabel, Anda dapat melakukan hal berikut:

- Untuk memulai kueri di editor dengan pernyataan SELECT yang menanyakan semua kolom dalam tabel, gunakan Pilih tabel.
- Untuk melihat atribut atau tabel, gunakan Tampilkan definisi tabel. Gunakan ini untuk melihat nama kolom, jenis kolom, pengkodean, kunci distribusi, kunci pengurutan, dan apakah kolom dapat berisi nilai nol. Untuk informasi selengkapnya tentang atribut tabel, lihat [MEMBUAT TABEL](#) di Panduan Pengembang Database Amazon Redshift.
- Untuk menghapus tabel, gunakan Hapus. Anda dapat menggunakan tabel Truncate untuk menghapus semua baris dari tabel atau Drop table untuk menghapus tabel dari database. Untuk informasi selengkapnya, lihat [TRUNCATE](#) dan [DROP TABLE di Panduan](#) Pengembang Database Amazon Redshift.

Pilih skema untuk Refresh atau Drop schema.

Pilih tampilan untuk Tampilkan definisi tampilan atau Tampilan Jatuhkan.

Pilih fungsi untuk Tampilkan definisi fungsi atau fungsi Drop.

Pilih prosedur tersimpan untuk Tampilkan definisi prosedur atau prosedur Drop.

Membuat objek database

Anda dapat membuat objek database, termasuk database, skema, tabel, dan fungsi yang ditentukan pengguna (). UDFs Anda harus terhubung ke cluster atau workgroup dan database untuk membuat objek database.

Membuat database

Anda dapat menggunakan query editor v2 untuk membuat database di cluster atau workgroup Anda.

Untuk membuat basis data

Untuk informasi tentang database, lihat [MEMBUAT DATABASE di Panduan Pengembang Database](#) Amazon Redshift.

1. Pilih



Buat, lalu pilih Database.

2. Masukkan nama Database.
3. (Opsional) Pilih Pengguna dan grup, dan pilih pengguna Database.
4. (Opsional) Anda dapat membuat database dari datashare atau file. AWS Glue Data Catalog Untuk informasi lebih lanjut tentang AWS Glue, lihat [Apa itu AWS Glue?](#) di Panduan AWS Glue Pengembang.
 - (Opsional) Pilih Buat menggunakan datashare, dan pilih Pilih datashare. Daftar ini mencakup datashares produsen yang dapat digunakan untuk membuat datashare konsumen di cluster atau workgroup saat ini.
 - (Opsional) Pilih Buat menggunakan AWS Glue Data Catalog, dan pilih database Choose an AWS Glue. Dalam skema katalog Data, masukkan nama yang akan digunakan untuk skema saat mereferensikan data dalam nama tiga bagian (database.schema.table).
5. Pilih Buat basis data.

Database baru ditampilkan di panel tampilan pohon.

Bila Anda memilih langkah opsional untuk menanyakan database yang dibuat dari datashare, sambungkan ke database Amazon Redshift di cluster atau workgroup (misalnya, database default`dev`), dan gunakan notasi tiga bagian (database.schema.table) yang mereferensikan nama database yang Anda buat saat memilih Buat menggunakan datashare. Database datasharing tercantum di tab editor editor kueri v2, tetapi tidak diaktifkan untuk koneksi langsung.

Bila Anda memilih langkah opsional untuk menanyakan database yang dibuat dari AWS Glue Data Catalog, sambungkan ke database Amazon Redshift Anda di cluster atau grup kerja (misalnya, database default`dev`), dan gunakan notasi tiga bagian (database.schema.table) yang mereferensikan nama database yang Anda buat saat Anda memilih Buat menggunakan AWS Glue Data Catalog, skema yang Anda beri nama dalam skema katalog Data, dan tabel di. AWS Glue Data Catalog Mirip dengan:

```
SELECT * FROM glue-database.glue-schema.glue-table
```

Note

Konfirmasikan bahwa Anda tersambung ke database default menggunakan metode koneksi Kredensial sementara menggunakan identitas IAM Anda, dan bahwa kredensial IAM Anda telah diberikan hak istimewa penggunaan ke database. AWS Glue

```
GRANT USAGE ON DATABASE glue-database to "IAM:MyIAMUser"
```

AWS Glue Basis data tercantum di tab editor kueri v2 editor, tetapi tidak diaktifkan untuk koneksi langsung.

Untuk informasi selengkapnya tentang kueri AWS Glue Data Catalog, lihat [Bekerja dengan rangkaian data yang dikelola Lake Formation sebagai konsumen dan Bekerja dengan rangkaian data yang dikelola Lake Formation sebagai produsen di Panduan Pengembang Basis Data Amazon Redshift](#).

Contoh membuat database sebagai konsumen datashare

Contoh berikut menjelaskan skenario tertentu yang digunakan untuk membuat database dari datashare menggunakan query editor v2. Tinjau skenario ini untuk mempelajari cara membuat database dari datashare di lingkungan Anda. Skenario ini menggunakan dua cluster, `cluster-base` (cluster produsen) dan `cluster-view` (cluster konsumen).

1. Gunakan konsol Amazon Redshift untuk membuat datashare untuk tabel di cluster. `category2` `cluster-base` Datashare produser diberi nama. `datashare_base`

Untuk informasi tentang membuat rangkaian data, lihat [Berbagi data di seluruh klaster di Amazon Redshift di Panduan Pengembang Database](#) Amazon Redshift.

2. Gunakan konsol Amazon Redshift untuk menerima datashare `datashare_base` sebagai konsumen untuk tabel di cluster. `category2` `cluster-view`
3. Lihat panel tampilan pohon di editor kueri v2 yang menunjukkan hierarki sebagai: `cluster-base`
 - Kluster: `cluster-base`
 - Database: `dev`
 - Skema: `public`
 - Tabel: `category2`

4. Pilih



Buat, lalu pilih Database.

5. Masukkan `see_datashare_base` nama Database.

6. Pilih Buat menggunakan datashare, dan pilih Pilih datashare. Pilih `datashare_base` untuk digunakan sebagai sumber database yang Anda buat.

Panel tampilan pohon di editor kueri v2 menunjukkan hierarki sebagai: `cluster-view`

- Kluster: `cluster-view`
 - Database: `see_datashare_base`
 - Skema: `public`
 - Tabel: `category2`

7. Saat Anda menanyakan data, sambungkan ke database default cluster `cluster-view` (biasanya bernama `dev`), tetapi referensi database datashare `see_datashare_base` di SQL Anda.

Note

Dalam tampilan editor editor kueri v2, cluster yang dipilih adalah `cluster-view`. Database yang dipilih adalah `dev`. Database `see_datashare_base` terdaftar tetapi tidak diaktifkan untuk koneksi langsung. Anda memilih `dev` database dan referensi `see_datashare_base` di SQL yang Anda jalankan.

```
SELECT * FROM "see_datashare_base"."public"."category2";
```

Query mengambil data dari datashare `datashare_base` di cluster. `cluster_base`

Contoh membuat database dari sebuah AWS Glue Data Catalog

Contoh berikut menjelaskan skenario tertentu yang digunakan untuk membuat database dari AWS Glue Data Catalog menggunakan editor query v2. Tinjau skenario ini untuk mempelajari cara membuat database dari AWS Glue Data Catalog lingkungan Anda. Skenario ini menggunakan satu cluster, `cluster-view` untuk memuat database yang Anda buat.

1. Pilih



Buat, lalu pilih Database.

2. Masukkan `data_catalog_database` nama Database.
3. Pilih Buat menggunakan AWS Glue Data Catalog, dan pilih Pilih AWS Glue database. Pilih `glue_db` untuk digunakan sebagai sumber database yang Anda buat.

Pilih Skema katalog data dan masukkan `myschema` sebagai nama skema yang akan digunakan dalam notasi tiga bagian.

Panel tampilan pohon di editor kueri v2 menunjukkan hierarki sebagai: `cluster-view`

- Kluster: `cluster-view`
 - Database: `data_catalog_database`
 - Skema: `myschema`
 - Tabel: `category3`
- 4. Saat Anda menanyakan data, sambungkan ke database default cluster `cluster-view` (biasanya bernama `dev`), tetapi referensi database `data_catalog_database` di SQL Anda.

Note

Dalam tampilan editor editor kueri v2, cluster yang dipilih adalah `cluster-view`. Database yang dipilih adalah `dev`. Database `data_catalog_database` terdaftar tetapi tidak diaktifkan untuk koneksi langsung. Anda memilih `dev` database dan referensi `data_catalog_database` di SQL yang Anda jalankan.

```
SELECT * FROM "data_catalog_database"."myschema"."category3";
```

Kueri mengambil data yang dikatalogkan oleh. AWS Glue Data Catalog

Membuat skema

Anda dapat menggunakan editor kueri v2 untuk membuat skema di klaster atau grup kerja Anda.

Untuk membuat skema

Untuk informasi tentang skema, lihat [Skema](#) di Panduan Pengembang Database Amazon Redshift.

1. Pilih



Buat, lalu pilih Skema.

2. Masukkan nama Skema.

3. Pilih Lokal atau Eksternal sebagai tipe Skema.

Untuk informasi selengkapnya tentang skema lokal, lihat [MEMBUAT SKEMA di Panduan Pengembang](#) Database Amazon Redshift. Untuk informasi selengkapnya tentang skema eksternal, lihat [MEMBUAT SKEMA EKSTERNAL di Panduan](#) Pengembang Database Amazon Redshift.

4. Jika Anda memilih Eksternal, maka Anda memiliki pilihan skema eksternal berikut.

- Glue Data Catalog — untuk membuat skema eksternal di Amazon Redshift yang mereferensikan tabel. AWS Glue Selain memilih AWS Glue database, pilih peran IAM yang terkait dengan cluster dan peran IAM yang terkait dengan Katalog Data.
- PostgreSQL — untuk membuat skema eksternal di Amazon Redshift yang mereferensikan database Amazon RDS for PostgreSQL atau Amazon Aurora PostgreSQL. Juga berikan informasi koneksi ke database. Untuk informasi selengkapnya tentang kueri federasi, lihat [Mengkueri data dengan kueri gabungan di Panduan Pengembang](#) Database Amazon Redshift.
- MySQL — untuk membuat skema eksternal di Amazon Redshift yang mereferensikan Amazon RDS untuk MySQL atau database Amazon Aurora MySQL Edisi yang kompatibel. Juga berikan informasi koneksi ke database. Untuk informasi selengkapnya tentang kueri federasi, lihat [Mengkueri data dengan kueri gabungan di Panduan Pengembang](#) Database Amazon Redshift.

5. Pilih Buat skema.

Skema baru muncul di panel tampilan pohon.

Membuat tabel

Anda dapat menggunakan editor kueri v2 untuk membuat tabel di cluster atau workgroup Anda.

Untuk membuat tabel

Anda dapat membuat tabel berdasarkan file nilai dipisahkan koma (CSV) yang Anda tentukan atau tentukan setiap kolom tabel. Untuk informasi tentang tabel, lihat [Merancang tabel](#) dan [MEMBUAT TABEL](#) di Panduan Pengembang Database Amazon Redshift.

Pilih Buka kueri di editor untuk melihat dan mengedit pernyataan CREATE TABLE sebelum Anda menjalankan kueri untuk membuat tabel.

1. Pilih



Buat, dan pilih Tabel.

2. Pilih skema.
3. Masukkan nama tabel.
4. Pilih



Tambahkan bidang untuk menambahkan kolom.

5. Gunakan file CSV sebagai templat untuk definisi tabel:

- a. Pilih Load dari CSV.
- b. Jelajahi lokasi file.

Jika Anda menggunakan file CSV, pastikan baris pertama file berisi judul kolom.

- c. Pilih file dan pilih Buka. Konfirmasikan bahwa nama kolom dan tipe data adalah apa yang Anda inginkan.
6. Untuk setiap kolom, pilih kolom dan pilih opsi yang Anda inginkan:
 - Pilih nilai untuk Encoding.
 - Pilih nilai Default.
 - Aktifkan Kenaikan secara otomatis jika Anda ingin nilai kolom bertambah. Kemudian tentukan nilai untuk benih kenaikan Otomatis dan langkah kenaikan Otomatis.
 - Aktifkan Not NULL jika kolom harus selalu berisi nilai.
 - Masukkan nilai Ukuran untuk kolom.
 - Aktifkan kunci utama jika Anda ingin kolom menjadi kunci utama.
 - Aktifkan tombol Unik jika Anda ingin kolom menjadi kunci unik.

7. (Opsional) Pilih detail Tabel dan pilih salah satu opsi berikut:
 - Kolom dan gaya kunci distribusi.
 - Urutkan kolom kunci dan jenis sortir.
 - Aktifkan Backup untuk menyertakan tabel dalam snapshot.
 - Nyalakan Tabel sementara untuk membuat tabel sebagai tabel sementara.
8. Pilih Buka kueri di editor untuk melanjutkan menentukan opsi untuk menentukan tabel atau pilih Buat tabel untuk membuat tabel.

Membuat fungsi

Anda dapat menggunakan query editor v2 untuk membuat fungsi di cluster atau workgroup Anda.

Untuk membuat fungsi

1. Pilih



Create, dan pilih Function.

2. Untuk Type, pilih SQL atau Python.
3. Pilih nilai untuk Skema.
4. Masukkan nilai untuk Nama untuk fungsi tersebut.
5. Masukkan nilai Volatilitas untuk fungsi tersebut.
6. Pilih Parameter berdasarkan tipe datanya sesuai urutan parameter input.
7. Untuk Pengembalian, pilih tipe data.
8. Masukkan program SQL atau kode program Python untuk fungsi tersebut.
9. Pilih Buat.

Untuk informasi selengkapnya tentang fungsi yang ditentukan pengguna (UDFs), lihat [Membuat fungsi yang ditentukan pengguna di Panduan Pengembang Database](#) Amazon Redshift.

Melihat kueri dan riwayat tab

Anda dapat melihat riwayat kueri Anda dengan editor kueri v2. Hanya kueri yang Anda jalankan menggunakan editor kueri v2 yang muncul di riwayat kueri. Kedua kueri yang dijalankan dari

menggunakan tab Editor atau tab Notebook ditampilkan. Anda dapat memfilter daftar yang ditampilkan berdasarkan periode waktu, seperti `This week`, di mana seminggu didefinisikan sebagai Senin-Minggu. Daftar kueri mengambil 25 baris kueri yang cocok dengan filter Anda sekaligus. Pilih Muat lebih banyak untuk melihat set berikutnya. Pilih kueri dan dari menu Tindakan. Tindakan yang tersedia tergantung pada apakah kueri yang dipilih telah disimpan. Anda dapat memilih operasi berikut:

- Lihat detail kueri - Menampilkan halaman detail kueri dengan informasi lebih lanjut tentang kueri yang dijalankan.
- Buka kueri di tab baru — Membuka tab editor baru dan memasangnya dengan kueri yang dipilih. Jika masih terhubung, cluster atau workgroup dan database dipilih secara otomatis. Untuk menjalankan kueri, pertama-tama konfirmasi bahwa cluster atau workgroup dan database yang benar dipilih.
- Tab sumber terbuka — Jika masih terbuka, navigasikan ke tab editor atau buku catatan yang berisi kueri saat dijalankan. Isi editor atau buku catatan mungkin telah berubah setelah kueri dijalankan.
- Buka kueri tersimpan — Menavigasi ke tab editor atau buku catatan dan membuka kueri.

Anda juga dapat melihat riwayat kueri yang dijalankan di tab Editor atau riwayat kueri yang dijalankan di tab Notebook. Untuk melihat riwayat kueri di tab, pilih Riwayat tab. Dalam riwayat tab, Anda dapat melakukan operasi berikut:

- Salin kueri - Menyalin konten SQL versi kueri ke clipboard.
- Buka kueri di tab baru — Membuka tab editor baru dan memasangnya dengan kueri yang dipilih. Untuk menjalankan query, Anda harus memilih cluster atau workgroup dan database.
- Lihat detail kueri - Menampilkan halaman detail kueri dengan informasi lebih lanjut tentang kueri yang dijalankan.

Berinteraksi dengan SQL generatif Amazon Q

Note

Dukungan SQL generatif Amazon Q hanya tersedia dalam hal berikut: Wilayah AWS

- Wilayah AS Timur (Virginia N.) (us-east-1)
- Wilayah AS Timur (Ohio) (us-east-2)
- Wilayah AS Barat (Oregon) (us-west-2)

- Wilayah Asia Pasifik (Mumbai) (ap-south-1)
- Wilayah Asia Pasifik (Seoul) (ap-northeast-2)
- Wilayah Asia Pasifik (Singapura) (ap-southeast-1)
- Wilayah Asia Pasifik (Sydney) (ap-southeast-2)
- Wilayah Asia Pasifik (Tokyo) (ap-northeast-1)
- Wilayah Kanada (Tengah) (ca-central-1)
- Wilayah Eropa (Frankfurt) (eu-central-1)
- Wilayah Eropa (Irlandia) (eu-west-1)
- Wilayah Eropa (London) (eu-west-2)
- Wilayah Eropa (Paris) (eu-west-3)
- Wilayah Amerika Selatan (São Paulo) (sa-east-1)

Untuk informasi tentang tempat data Anda diproses, lihat [Inferensi lintas wilayah di Amazon Q Developer](#) di Panduan Pengguna Pengembang Amazon Q.

Anda dapat berinteraksi dengan kemampuan SQL generatif Amazon Q di editor kueri Amazon Redshift v2. Ini adalah asisten pengkodean yang menghasilkan pernyataan SQL berdasarkan prompt dan skema database Anda. Asisten pengkodean ini tersedia saat Anda membuat buku catatan di editor kueri v2. SQL yang dihasilkan adalah untuk database yang terhubung dengan notebook Anda.

Saat berinteraksi dengan SQL generatif Amazon Q, ajukan pertanyaan spesifik, ulangi ketika Anda memiliki permintaan yang rumit, dan verifikasi jawaban untuk akurasi.

Saat memberikan permintaan analisis dalam bahasa alami, sespesifik mungkin untuk membantu asisten pengkodean memahami dengan tepat apa yang Anda butuhkan. Alih-alih bertanya “temukan tempat teratas yang menjual tiket terbanyak,” berikan detail lebih lanjut seperti “temukan names/ids tiga tempat teratas yang menjual tiket terbanyak pada tahun 2008.” Gunakan nama objek yang konsisten dan spesifik dalam database Anda saat Anda mengetahuinya. Seperti skema, tabel, dan nama kolom seperti yang didefinisikan dalam database Anda alih-alih merujuk ke objek yang sama dengan cara yang berbeda, yang dapat membingungkan asisten.

Memecah permintaan kompleks menjadi beberapa pernyataan sederhana yang lebih mudah bagi asisten untuk menafsirkan. Ajukan pertanyaan tindak lanjut secara berulang untuk mendapatkan analisis yang lebih rinci dari asisten. Misalnya, tanyakan dulu “negara bagian mana yang memiliki

tempat paling banyak?” Kemudian berdasarkan tanggapan, tanyakan “tempat mana yang paling populer dari negara bagian ini?”.

Tinjau SQL yang dihasilkan sebelum menjalankannya untuk memastikan akurasi. Jika kueri SQL yang dihasilkan memiliki kesalahan atau tidak cocok dengan maksud Anda, berikan instruksi kepada asisten tentang cara memperbaikinya alih-alih mengulangi seluruh permintaan. Misalnya, jika kueri tidak memiliki klausa predikat pada tahun, tanyakan “Sediakan tempat dari tahun 2008.”

Kirim teks kesalahan yang Anda terima dari menjalankan SQL yang dihasilkan sebagai petunjuk kembali ke SQL generatif Amazon Q. Ia belajar dari kesalahan ini untuk menghasilkan SQL yang lebih baik.

Tambahkan skema Anda ke jalur pencarian SQL untuk memberi sinyal bahwa skema harus digunakan. Misalnya, tambahkan skema tickit saat data berada dalam skema tickit daripada skema publik.

```
set search_path to '$user', tickit;
```

Pertimbangan saat berinteraksi dengan SQL generatif Amazon Q

Pertimbangkan hal berikut saat bekerja di panel obrolan.

- Administrator editor kueri v2 untuk akun Anda harus mengaktifkan kemampuan obrolan di halaman pengaturan SQL Generatif.
- Untuk menggunakan SQL generatif Amazon Q, Anda memerlukan izin `sqlworkbench:GetQSqlRecommendations` dalam kebijakan IAM, selain izin lain yang ditentukan dalam kebijakan AWS terkelola untuk editor kueri v2. Untuk informasi selengkapnya tentang kebijakan AWS terkelola, lihat [Mengakses editor kueri v2](#).
- Pertanyaan Anda harus ditulis dalam bahasa Inggris.
- Pertanyaan Anda harus mengacu pada database yang terhubung di cluster atau workgroup Anda. Untuk menghindari kesalahan status kosong, harus ada setidaknya satu tabel dan beberapa data dalam database.
- Pertanyaan Anda harus mengacu pada data yang disimpan dalam database yang terhubung. Itu tidak dapat merujuk skema eksternal. Untuk informasi selengkapnya tentang skema yang didukung, lihat [Membuat skema](#) di Panduan Pengembang Database Amazon Redshift.
- Setiap pertanyaan yang menghasilkan SQL yang mengubah database yang terhubung dapat menghasilkan peringatan.

- Teknologi AI generatif adalah hal baru dan mungkin ada kesalahan, kadang-kadang disebut halusinasi, dalam tanggapannya. Uji dan tinjau semua kode untuk kesalahan dan kerentanan sebelum menggunakannya di lingkungan atau beban kerja Anda.
- Anda dapat meningkatkan rekomendasi dengan membagikan kueri SQL yang dijalankan oleh pengguna lain di akun Anda. Administrator akun Anda dapat menjalankan perintah SQL berikut untuk memungkinkan akses ke riwayat kueri akun.

```
GRANT ROLE SYS:MONITOR to "IAMR:role-name";  
GRANT ROLE SYS:MONITOR to "IAM:user-name";  
GRANT ROLE SYS:MONITOR to "database-username";
```

Untuk informasi selengkapnya `SYS:MONITOR`, lihat [peran yang ditentukan sistem Amazon Redshift di Panduan](#) Pengembang Database Amazon Redshift.

- Data Anda aman dan pribadi. Data Anda tidak dibagikan di seluruh akun. Kueri, data, dan skema basis data Anda tidak digunakan untuk melatih model fondasi AI generatif (FM). Masukan Anda digunakan sebagai petunjuk kontekstual ke FM untuk menjawab hanya pertanyaan Anda.

Menggunakan SQL generatif

Setelah izin yang benar dikonfigurasi, saat bekerja dengan buku catatan di editor kueri v2, Anda dapat memilih ikon untuk memulai percakapan.

Untuk berinteraksi dengan obrolan SQL generatif Amazon Q untuk menghasilkan SQL

1. Di tab Editor dari editor kueri v2, buka buku catatan.
2. Pilih ikon



Generative SQL, lalu ikuti petunjuk untuk mengajukan pertanyaan Anda tentang SQL generatif Amazon Redshift editor v2 generatif di panel obrolan.

Anda memberikan pertanyaan di bidang prompt dan SQL generatif Amazon Q merespons dengan SQL yang disarankan. Setiap kesalahan yang ditemui dikembalikan kepada Anda di panel obrolan.

3. Pilih Tambahkan ke buku catatan untuk menambahkan sel Markdown dengan prompt Anda dan sel SQL dengan SQL yang disarankan ke buku catatan Anda.

4. (Opsional) Berikan umpan balik mengenai SQL yang dihasilkan dengan memilih ikon umpan balik yang



bermanfaat atau ikon umpan balik yang



tidak membantu. Anda dapat mengkategorikan umpan balik yang tidak membantu sebagai `Incorrect tables/columns`, `Incorrect predicates/literals/group` `bys`, `Incorrect SQL structure`, atau `Other`. Selain itu, Anda dapat memberikan beberapa teks bentuk bebas dengan umpan balik Anda tentang keakuratan SQL.

5. (Opsional) Pilih Regenerate SQL untuk menghasilkan respons lain untuk prompt yang sama. Anda dapat memilih untuk Regenerasi SQL satu kali untuk prompt saat ini.

6. (Opsional) Di panel obrolan SQL generatif, pilih ikon



Lainnya, lalu pilih Refresh database untuk menyegarkan metadata yang menjelaskan database Anda yang terhubung. Metadata ini mencakup definisi skema, tabel, dan kolom dalam database Anda.

Memperbarui pengaturan SQL generatif sebagai administrator

Pengguna dengan izin IAM yang tepat dapat melihat dan mengubah pengaturan SQL Generatif untuk pengguna lain dalam hal yang sama. Akun AWS Administrator ini harus memiliki izin `sqlworkbench:UpdateAccountQSQLSettings` dalam kebijakan IAM mereka, selain izin lain yang ditentukan dalam kebijakan AWS terkelola untuk editor kueri v2. Untuk informasi selengkapnya tentang kebijakan terkelola, lihat [izin yang diperlukan untuk menggunakan editor kueri v2](#).

Agar administrator mengaktifkan obrolan SQL generatif untuk semua pengguna di akun

1. Pilih ikon



Pengaturan untuk menampilkan menu layar pengaturan yang berbeda.

2. Kemudian pilih ikon Pengaturan SQL



Generatif untuk menampilkan halaman pengaturan SQL generatif Q.

3. Pilih pengaturan SQL generatif Q untuk mengaktifkan kemampuan SQL generatif bagi pengguna di akun.

Setelah mengaktifkan SQL generatif Amazon Q, Anda dapat melihat jumlah permintaan yang tersisa di alokasi Anda. Administrator editor kueri v2 dapat memungkinkan pengguna di akun untuk menggunakan Amazon Q Developer Pro tier. Untuk menggunakan tingkat Pro, siapkan pengguna Anda dengan IAM Identity Center dan berlangganan setiap pengguna ke Amazon Q Developer Pro tier. Untuk informasi tentang menyiapkan Pusat Identitas IAM dengan Amazon Redshift, lihat [Hubungkan Redshift dengan AWS IAM Identity Center untuk pengalaman masuk tunggal](#) Untuk informasi tentang harga Pengembang Amazon Q, lihat [harga Pengembang Amazon Q](#).

Saat menggunakan Amazon Q Developer Free tier, jumlah total permintaan dari semua pengguna Akun AWS dibatasi hingga 1.000 dalam sebulan. Saat menggunakan Amazon Q Developer Pro tier, jumlah total permintaan yang dapat dikirimkan oleh setiap pengguna dibatasi hingga 1.000 dalam sebulan. Anda dapat melihat jumlah prompt yang tersedia di halaman Pengaturan. Untuk informasi tentang harga Pengembang Amazon Q, lihat [harga Pengembang Amazon Q](#).

Konteks khusus

Administrator editor kueri v2 dapat menentukan konteks khusus untuk menyesuaikan SQL yang dihasilkan dengan lingkungan Anda. Konteks kustom memberikan pengetahuan dan preferensi domain untuk memberikan kontrol halus atas pembuatan SQL. Konteks kustom didefinisikan dalam file JSON yang dapat diunggah oleh administrator editor kueri v2 ke SQL generatif Amazon Q.

Kunci JSON yang digunakan untuk mempersonalisasi SQL yang dihasilkan untuk gudang data adalah berikut.

Semua referensi tabel harus mengikuti `database.schema.table` notasi tiga bagian.

Sumber daya

Sumber daya menentukan ruang lingkup atau bagian dari aset data yang konteks kustom diterapkan.

ResourceId

Menentukan identifier unik dari sumber daya. Untuk cluster Amazon Redshift, tentukan `cluster id` Untuk grup kerja Redshift Tanpa Server, tentukan `workgroup name`

ResourceType

Nilai valid: REDSHIFT_WAREHOUSE.

TablesToInclude

Menentukan satu set tabel yang dipertimbangkan untuk generasi SQL. Bidang ini sangat penting ketika Anda ingin membatasi ruang lingkup kueri SQL ke subset yang ditentukan dari tabel yang tersedia. Ini dapat membantu mengoptimalkan proses pembuatan dengan mengurangi referensi tabel yang tidak perlu. Anda dapat memasang bidang ini dengan kontrol `TablesToExclude` yang lebih baik atas pembuatan kueri.

TablesToExclude

Menentukan set tabel yang dikecualikan dari generasi SQL. Gunakan ini ketika tabel tertentu tidak relevan atau tidak boleh dipertimbangkan dalam proses pembuatan kueri.

TableAnnotations

Menyediakan metadata atau informasi tambahan tentang tabel yang digunakan. Anotasi ini dapat mencakup deskripsi tabel, catatan penggunaan, atau atribut tambahan apa pun yang membantu SQL generatif Amazon Q lebih memahami konteks atau struktur tabel. Ini berharga untuk meningkatkan akurasi pembuatan SQL dengan menambahkan kejelasan pada definisi tabel.

ColumnsToInclude

Mendefinisikan kolom mana dari tabel yang ditentukan disertakan saat menghasilkan kueri SQL. Bidang ini membantu Amazon Q generatif SQL fokus pada kolom yang relevan dan meningkatkan kinerja dengan mempersempit ruang lingkup pengambilan data. Ini memastikan SQL generatif Amazon Q hanya menarik data yang diperlukan untuk konteks kueri yang diberikan.

ColumnsToExclude

Menentukan kolom yang dihilangkan dari pertimbangan dalam generasi SQL. Ini dapat digunakan ketika kolom tertentu berisi data yang tidak relevan atau berlebihan yang tidak boleh dipertimbangkan oleh SQL generatif Amazon Q. Dengan mengelola penyertaan dan pengecualian kolom, Anda dapat memperbaiki hasil dan mempertahankan kontrol atas data yang diambil.

ColumnAnnotations

Mirip dengan `TableAnnotations`, bidang ini menyediakan metadata atau anotasi khusus untuk masing-masing kolom. Anotasi ini dapat menawarkan wawasan tentang definisi kolom atau

instruksi penanganan khusus. Informasi ini bermanfaat dalam memandu proses pembuatan SQL dan memastikan bahwa kolom digunakan dengan tepat dalam kueri.

CuratedQueries

Satu set contoh pertanyaan dan jawaban yang telah ditentukan, di mana pertanyaan ditulis dalam bahasa alami (NLQ) dan jawabannya adalah kueri SQL yang sesuai. Contoh-contoh ini membantu SQL generatif Amazon Q memahami jenis kueri yang diharapkan akan dihasilkan. Mereka berfungsi sebagai titik referensi untuk meningkatkan akurasi dan relevansi output SQL generatif Amazon Q.

CustomDocuments

Potongan informasi atau petunjuk tambahan yang diberikan kepada SQL generatif Amazon Q, seperti definisi, pengetahuan khusus domain, atau penjelasan. Misalnya, jika unit bisnis Anda menggunakan cara unik untuk menghitung nilai, misalnya “di divisi manufaktur total penjualan adalah harga* pendapatan” ini dapat didokumentasikan di sini. Dokumen-dokumen ini meningkatkan kemampuan SQL generatif Amazon Q untuk menafsirkan input bahasa alami dengan memberikan konteks tambahan.

AdditionalTables

Menentukan setiap tabel tambahan yang harus dipertimbangkan untuk generasi SQL tetapi bukan bagian dari data yang disimpan dalam gudang data. Hal ini memungkinkan SQL generatif Amazon Q untuk mengintegrasikan sumber data eksternal ke dalam logika generasi SQL, memperluas kapasitasnya untuk menangani lingkungan data yang kompleks.

AppendToPrompt

Petunjuk atau pedoman tambahan yang diberikan kepada SQL generatif Amazon Q untuk memandu proses pembuatan SQL. Ini dapat mencakup arahan khusus tentang cara menyusun kueri, preferensi untuk konstruksi SQL tertentu, atau instruksi tingkat tinggi lainnya yang meningkatkan kualitas keluaran SQL generatif Amazon Q.

Contoh konteks kustom berikut menunjukkan format file JSON dan mendefinisikan yang berikut:

- Mendefinisikan konteks kustom untuk gudang data Amazon Redshift untuk kluster. `mycluster`
- Mendefinisikan tabel dan kolom tertentu untuk disertakan dan dikecualikan untuk membantu mengoptimalkan proses pembuatan SQL.
- Mendefinisikan anotasi untuk tabel dan kolom yang dipanggil untuk disertakan.

- Mendefinisikan contoh kueri yang dikuratori untuk SQL generatif Amazon Q untuk digunakan.
- Mendefinisikan dokumen kustom dan pagar untuk digunakan saat membuat SQL.
- Mendefinisikan DDL untuk tabel tambahan untuk digunakan saat menghasilkan SQL.

```
{
  "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."
    }
}

```

Tutorial: Menggunakan kemampuan SQL generatif Amazon Q dengan data TICKIT

Untuk membuat prompt yang efisien untuk menghasilkan SQL, Anda harus belajar tentang skema database dan data Anda. Data TICKIT terdiri dari tujuh tabel: dua tabel fakta dan lima dimensi. Data sampel berisi catatan tentang penjualan kepada petugas acara hiburan yang terjadi pada tahun 2008. Untuk informasi selengkapnya tentang skema data TICKIT, lihat [Contoh database](#) di Panduan Pengembang Database Amazon Redshift. Anda dapat memuat data TICKIT ke dalam

database dengan berbagai metode di konsol Amazon Redshift dan editor kueri v2. Query editor v2 menyediakan metode untuk memuat data TICKIT ke dalam `databasesample_data_dev`. Untuk informasi, lihat [Memuat data ke dalam database](#). Editor kueri v2 juga menyediakan contoh prompt untuk data TICKIT. Skenario berikut menjelaskan percakapan dengan SQL generatif untuk menghasilkan SQL tentang data sampel TICKIT. Dalam skenario ini, data sampel TICKIT sudah dibuat dalam dev database di cluster Amazon Redshift.

Note

Contoh ini adalah untuk menggambarkan percakapan. Tanggapan dari SQL generatif mungkin tidak cocok dengan hasil Anda menggunakan prompt yang sama.

Contoh percakapan dengan SQL generatif Amazon Q

1. Di Editor, sambungkan ke cluster atau workgroup yang berisi data sampel TICKIT.
2. Buat buku catatan kosong dan pilih ikon



Generative SQL untuk membuka panel obrolan.

3. Masukkan prompt berikut untuk menghasilkan SQL untuk memverifikasi jumlah catatan dalam VENUE tabel:

How many venues are there?

```
SELECT
  COUNT(*) AS num_venues
FROM
  tickit.venue
```

Pilih Tambahkan ke buku catatan untuk menambahkan dua sel ke buku catatan yang terbuka. Satu sel Markdown “Ada berapa tempat?” dan satu berisi SQL yang dihasilkan.

Di sel SQL, pilih Jalankan untuk menerima hasilnya:

```
count
-----
```

202

4. Untuk meminta versi SQL lain, pilih Regenerate SQL dan terima jawaban berikut:

```
SELECT
  venuestate,
  COUNT(*) AS num_venues
FROM
  tickit.venue
GROUP BY
  venuestate
```

Pilih Tambahkan ke buku catatan untuk menambahkan dua sel lagi ke buku catatan yang terbuka. Satu sel Markdown “Ada berapa tempat?” dan satu berisi SQL yang dihasilkan.

Di sel SQL, pilih Jalankan untuk menerima status tempat daftar hasil berdasarkan nomor:

```
venuestate num_venues
-----
MA          4
OH          8
MI          5
...
```

5. Asisten SQL generatif dapat membuat asumsi tentang apa yang Anda tanyakan. Meskipun tidak ada widget dalam database, Anda masih bisa bertanya. Dalam hal ini, ini memberi Anda pesan bahwa itu tidak dapat menghasilkan SQL.

How many widgets are there?

I was not able to generate any SQL

6. Saat Anda mempelajari lebih lanjut tentang database Anda, Anda dapat mengajukan pertanyaan yang lebih spesifik untuk menghasilkan SQL. Misalnya, Anda menyadari bahwa database berisi data penjualan selama beberapa bulan di tahun 2008. Masukkan prompt berikut untuk menghasilkan SQL untuk menemukan jumlah penjualan pada bulan November 2008:

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
```

Pilih Tambahkan ke buku catatan untuk menambahkan dua sel ke buku catatan yang terbuka. Satu sel penurunan harga “Berapa banyak penjualan yang ada pada November 2008?” dan satu berisi SQL yang dihasilkan.

Di sel SQL, pilih Jalankan untuk menerima hasilnya:

```
num_sales
-----
0
```

Anda menyadari ini bukan hasil yang Anda harapkan.

- Anda melihat bahwa predikat `date.month = 'November'` mengharapkan bulan dalam tabel DATE diwakili oleh singkatan dari nama bulan. Anda mengubah predikat ke `d.month = 'NOV'` dan menjalankan kembali SQL.

```
SELECT
  COUNT(*)
FROM
  sales
  JOIN date ON sales.dateid = date.dateid
WHERE
  date.month = 'NOV'
  AND date.year = 2008
```

Di sel SQL, pilih Jalankan untuk mendapatkan hasil baru.

```
count
-----
14261
```

8. Jika Anda mengajukan pertanyaan yang mencoba mengubah database yang terhubung, pesan peringatan dikembalikan bersama dengan SQL yang direkomendasikan. Masukkan prompt berikut untuk menghasilkan SQL untuk memasukkan data ke dalam tabel:

```
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.

Jika Anda memilih Tambahkan ke notebook untuk menambahkan dua sel ke notebook yang terbuka dan menjalankan SQL, maka SQL gagal.

```
ERROR: syntax error at or near "," Position: 132 [ErrorId:
1-6546764a-011df2691778846219ce6ec2]
```

Skenario ini hanya menggambarkan beberapa cara dasar untuk berinteraksi dengan SQL generatif Amazon Q. Anda dapat bereksperimen lebih banyak lagi dengan teknologi AI generatif ini untuk membantu Anda mulai membuat SQL untuk menanyakan database Anda.

Memuat data ke dalam database

Anda dapat menggunakan editor kueri v2 untuk memuat data ke dalam database di kluster atau grup kerja Amazon Redshift. Bagian ini mencakup cara memuat data sampel, data dari S3, dan data dari pengaturan file lokal dan alur kerja.

Data sampel

Editor kueri v2 dilengkapi dengan data sampel dan buku catatan yang tersedia untuk dimuat ke dalam database sampel dan skema yang sesuai.

Untuk memuat data sampel, pilih



ikon yang terkait dengan data sampel yang ingin Anda muat. Editor kueri v2 kemudian memuat data ke dalam skema dalam database `sample_data_dev` dan membuat folder notebook yang disimpan.

Kumpulan data sampel berikut tersedia.

tickit

Sebagian besar contoh dalam dokumentasi Amazon Redshift menggunakan data sampel yang disebut `tickit`. Data ini terdiri dari tujuh tabel: dua tabel fakta dan lima dimensi. Saat Anda memuat data ini, skema `tickit` diperbarui dengan data sampel. Untuk informasi selengkapnya tentang `tickit` data, lihat [Contoh database](#) di Panduan Pengembang Database Amazon Redshift.

tpch

Data ini digunakan untuk tolok ukur dukungan keputusan. Saat Anda memuat data ini, skema `tpch` diperbarui dengan data sampel. Untuk informasi lebih lanjut tentang `tpch` data, lihat [TPC-H](#).

tpcds

Data ini digunakan untuk tolok ukur dukungan keputusan. Saat Anda memuat data ini, skema `tpcds` diperbarui dengan data sampel. Untuk informasi lebih lanjut tentang `tpcds` data, lihat [TPC-DS](#).

Memuat data dari Amazon S3

Anda dapat memuat data Amazon S3 ke tabel yang sudah ada atau yang baru.

Untuk memuat data ke dalam tabel yang ada

Perintah `COPY` digunakan oleh editor kueri v2 untuk memuat data dari Amazon S3. Perintah `COPY` yang dihasilkan dan digunakan dalam wizard memuat data editor kueri v2 mendukung banyak

parameter yang tersedia untuk sintaks perintah COPY untuk disalin dari Amazon S3. Untuk informasi tentang perintah COPY dan opsinya yang digunakan untuk memuat data dari Amazon S3, lihat [COPY dari Amazon Simple Storage Service](#) di Panduan Pengembang Database Amazon Redshift.

1. Konfirmasikan bahwa tabel sudah dibuat di database tempat Anda ingin memuat data.
2. Konfirmasikan bahwa Anda terhubung ke database target di panel tampilan pohon editor kueri v2 sebelum melanjutkan. Anda dapat membuat koneksi menggunakan menu konteks (klik kanan) ke cluster atau workgroup tempat data akan dimuat.

Pilih



Muat data.

3. Untuk sumber Data, pilih Muat dari bucket S3.
4. Di S3 URIs, pilih Browse S3 untuk mencari bucket Amazon S3 yang berisi data yang akan dimuat.
5. Jika bucket Amazon S3 yang ditentukan tidak AWS Region sama dengan tabel target, pilih lokasi file S3 untuk lokasi data tersebut AWS Region berada.
6. Pilih File ini adalah file manifes jika file Amazon S3 sebenarnya adalah manifes yang berisi beberapa bucket Amazon S3. URIs
7. Pilih format File untuk file yang akan diunggah. Format data yang didukung adalah CSV, JSON, DELIMITER, FIXEDWIDTH, SHAPEFILE, AVRO, PARQUET, dan ORC. Tergantung pada format file yang ditentukan, Anda dapat memilih opsi File masing-masing. Anda juga dapat memilih Data dienkripsi jika data dienkripsi dan masukkan Nama Sumber Daya Amazon (ARN) dari kunci KMS yang digunakan untuk mengenkripsi data.

Jika Anda memilih CSV atau DELIMITER, Anda juga dapat memilih karakter Delimiter dan apakah akan Abaikan baris header jika jumlah baris yang ditentukan sebenarnya adalah nama kolom dan bukan data yang akan dimuat.

8. Pilih metode kompresi untuk mengompres file Anda. Defaultnya adalah tidak ada kompresi.
9. (Opsional) Pengaturan lanjutan mendukung berbagai parameter konversi Data dan operasi Beban. Masukkan informasi ini sesuai kebutuhan untuk file Anda.

Untuk informasi selengkapnya tentang konversi data dan parameter pemuatan data, lihat [Parameter konversi data](#) dan [Operasi pemuatan data](#) di Panduan Pengembang Database Amazon Redshift.

10. Pilih Berikutnya.

11. Pilih Muat tabel yang ada.
12. Konfirmasikan atau pilih lokasi tabel Target termasuk Cluster atau kelompok kerja, Database, Skema, dan nama Tabel tempat data dimuat.
13. Pilih peran IAM yang memiliki izin yang diperlukan untuk memuat data dari Amazon S3.
14. (Opsional) Pilih nama kolom untuk memasukkannya Pemetaan kolom untuk memetakan kolom dalam urutan file data input.
15. Pilih Muat data untuk memulai pemuatan data.

Saat beban selesai, editor kueri ditampilkan dengan perintah COPY yang dihasilkan yang digunakan untuk memuat data Anda. Hasil COPY ditampilkan. Jika berhasil, Anda sekarang dapat menggunakan SQL untuk memilih data dari tabel yang dimuat. Ketika ada kesalahan, kueri tampilan sistem STL_LOAD_ERRORS untuk mendapatkan detail lebih lanjut. Untuk informasi tentang kesalahan perintah COPY, lihat [STL_LOAD_ERRORS di Panduan Pengembang Database](#) Amazon Redshift.

Saat Anda memuat data ke dalam tabel baru, editor kueri v2 pertama-tama membuat tabel di database, lalu memuat data sebagai tindakan terpisah dalam alur kerja yang sama.

Untuk memuat data ke dalam tabel baru

Perintah COPY digunakan oleh editor kueri v2 untuk memuat data dari Amazon S3. Perintah COPY yang dihasilkan dan digunakan dalam wizard memuat data editor kueri v2 mendukung banyak parameter yang tersedia untuk sintaks perintah COPY untuk disalin dari Amazon S3. Untuk informasi tentang perintah COPY dan opsinya yang digunakan untuk memuat data dari Amazon S3, lihat [COPY dari Amazon Simple Storage Service](#) di Panduan Pengembang Database Amazon Redshift.

1. Konfirmasikan bahwa Anda terhubung ke database target di panel tampilan pohon editor kueri v2 sebelum melanjutkan. Anda dapat membuat koneksi menggunakan menu konteks (klik kanan) ke cluster atau workgroup tempat data akan dimuat.

Pilih



Muat data.

2. Untuk sumber Data, pilih Muat dari bucket S3.
3. Di S3 URIs, pilih Browse S3 untuk mencari bucket Amazon S3 yang berisi data yang akan dimuat.

4. Jika bucket Amazon S3 yang ditentukan tidak AWS Region sama dengan tabel target, pilih lokasi file S3 untuk lokasi data tersebut AWS Region berada.
5. Pilih File ini adalah file manifes jika file Amazon S3 sebenarnya adalah manifes yang berisi beberapa bucket Amazon S3. URIs
6. Pilih format File untuk file yang akan diunggah. Format data yang didukung adalah CSV, JSON, DELIMITER, FIXEDWIDTH, SHAPEFILE, AVRO, PARQUET, dan ORC. Tergantung pada format file yang ditentukan, Anda dapat memilih opsi File masing-masing. Anda juga dapat memilih Data dienkripsi jika data dienkripsi dan masukkan Nama Sumber Daya Amazon (ARN) dari kunci KMS yang digunakan untuk mengenkripsi data.

Jika Anda memilih CSV atau DELIMITER, Anda juga dapat memilih karakter Delimiter dan apakah akan Abaikan baris header jika jumlah baris yang ditentukan sebenarnya adalah nama kolom dan bukan data yang akan dimuat.

7. Pilih metode kompresi untuk mengompres file Anda. Defaultnya adalah tidak ada kompresi.
8. (Opsional) Pengaturan lanjutan mendukung berbagai parameter konversi Data dan operasi Beban. Masukkan informasi ini sesuai kebutuhan untuk file Anda.

Untuk informasi selengkapnya tentang konversi data dan parameter pemuatan data, lihat [Parameter konversi data](#) dan [Operasi pemuatan data](#) di Panduan Pengembang Database Amazon Redshift.

9. Pilih Berikutnya.
10. Pilih Muat tabel baru.

Kolom tabel disimpulkan dari data input. Anda dapat memodifikasi definisi skema tabel dengan menambahkan kolom dan detail tabel. Untuk kembali ke skema tabel yang disimpulkan editor kueri v2, pilih Pulihkan ke default.

11. Konfirmasikan atau pilih lokasi tabel Target termasuk Cluster atau workgroup, Database, dan Skema tempat data dimuat. Masukkan nama Tabel yang akan dibuat.
12. Pilih peran IAM yang memiliki izin yang diperlukan untuk memuat data dari Amazon S3.
13. Pilih Buat tabel untuk membuat tabel menggunakan definisi yang ditampilkan.

Ringkasan tinjauan definisi tabel ditampilkan. Tabel dibuat dalam database. Untuk kemudian menghapus tabel, jalankan perintah DROP TABLE SQL. Untuk informasi selengkapnya, lihat [DROP TABLE](#) di Panduan Pengembang Database Amazon Redshift.

14. Pilih Muat data untuk memulai pemuatan data.

Saat beban selesai, editor kueri ditampilkan dengan perintah COPY yang dihasilkan yang digunakan untuk memuat data Anda. Hasil COPY ditampilkan. Jika berhasil, Anda sekarang dapat menggunakan SQL untuk memilih data dari tabel yang dimuat. Ketika ada kesalahan, kueri tampilan sistem STL_LOAD_ERRORS untuk mendapatkan detail lebih lanjut. Untuk informasi tentang kesalahan perintah COPY, lihat [STL_LOAD_ERRORS di Panduan Pengembang Database](#) Amazon Redshift.

Memuat data dari pengaturan file lokal dan alur kerja

Anda dapat memuat data dari file lokal ke tabel yang sudah ada atau yang baru.

Pengaturan administrator untuk memuat data dari file lokal

Administrator editor kueri v2 Anda harus menentukan bucket Amazon S3 umum di jendela Pengaturan akun. Pengguna akun harus dikonfigurasi dengan izin yang tepat.

- Izin IAM yang diperlukan — pengguna pemuatan dari file lokal harus memilikis3:ListBucket,,s3:GetBucketLocation, s3:putObjects3:getObject, dan s3:deleteObject izin. *optional-prefix* Dapat ditentukan untuk membatasi penggunaan terkait editor kueri v2 dari bucket ini ke objek dengan awalan ini. Anda dapat menggunakan opsi ini saat menggunakan bucket Amazon S3 yang sama ini untuk penggunaan selain editor kueri v2. Untuk informasi selengkapnya tentang bucket dan awalan, lihat [Mengelola akses pengguna ke folder tertentu](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon. Untuk memastikan bahwa akses data lintas pengguna tidak diizinkan, sebaiknya administrator editor kueri v2 menggunakan kebijakan bucket Amazon S3 untuk membatasi akses objek berdasarkan aws:userid Contoh berikut memungkinkan izin Amazon S3 ke a *<staging-bucket-name>* dengan akses baca/tulis hanya ke objek Amazon S3 dengan awalan sebagai. aws:userid
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}/*"
        ]
    }
]
}

```

- Pemisahan data — kami menyarankan agar pengguna tidak memiliki akses ke data satu sama lain (bahkan sebentar). Memuat dari file lokal menggunakan bucket Amazon S3 pementasan yang disiapkan oleh administrator editor kueri v2. Konfigurasi kebijakan bucket untuk bucket staging untuk menyediakan pemisahan data antar pengguna. Contoh berikut menunjukkan kebijakan bucket yang memisahkan data antara pengguna. *<staging-bucket-name>*

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}/*"
            ]
        }
    ]
}

```

Memuat data dari file lokal

Untuk memuat data file lokal ke dalam tabel yang ada

Administrator editor kueri v2 Anda harus menentukan bucket Amazon S3 umum di jendela Pengaturan akun. editor kueri v2 secara otomatis mengunggah file lokal ke bucket Amazon S3 umum yang digunakan oleh akun Anda, lalu menggunakan perintah COPY untuk memuat data. Perintah COPY yang dihasilkan dan dijalankan oleh editor kueri v2 memuat jendela file lokal mendukung banyak parameter yang tersedia untuk sintaks perintah COPY untuk disalin dari Amazon S3. Untuk informasi tentang perintah COPY dan opsinya yang digunakan untuk memuat data dari Amazon S3, lihat [COPY dari Amazon S3](#) di Panduan Pengembang Database Amazon Redshift.

1. Konfirmasikan bahwa tabel sudah dibuat di database tempat Anda ingin memuat data.
2. Konfirmasikan bahwa Anda terhubung ke database target di panel tampilan pohon editor kueri v2. Anda dapat membuat koneksi menggunakan menu konteks (klik kanan) ke cluster atau workgroup tempat data akan dimuat.
3. Pilih  Muat data.
4. Untuk Sumber data, pilih Muat dari file lokal.
5. Pilih Browse untuk menemukan file yang berisi data untuk Memuat file. Secara default, file dengan ekstensi .csv, .avro.parquet, dan .orc ditampilkan, tetapi Anda dapat memilih jenis file lainnya. Ukuran file maksimum adalah 100 MB.
6. Pilih format File untuk file yang akan diunggah. Format data yang didukung adalah CSV, JSON, DELIMITER, FIXEDWIDTH, SHAPEFILE, AVRO, PARQUET, dan ORC. Tergantung pada format file yang ditentukan, Anda dapat memilih opsi File masing-masing. Anda juga dapat memilih Data dienkripsi jika data dienkripsi dan masukkan Nama Sumber Daya Amazon (ARN) dari kunci KMS yang digunakan untuk mengenkripsi data.

Jika Anda memilih CSV atau DELIMITER, Anda juga dapat memilih karakter Delimiter dan apakah akan Abaikan baris header jika jumlah baris yang ditentukan sebenarnya adalah nama kolom dan bukan data yang akan dimuat.

7. (Opsional) Pengaturan lanjutan mendukung berbagai parameter konversi Data dan operasi Beban. Masukkan informasi ini sesuai kebutuhan untuk file Anda.

Untuk informasi selengkapnya tentang konversi data dan parameter pemuatan data, lihat [Parameter konversi data](#) dan [Operasi pemuatan data](#) di Panduan Pengembang Database Amazon Redshift.

8. Pilih Berikutnya.
9. Pilih Muat tabel yang ada.
10. Konfirmasikan atau pilih lokasi tabel Target termasuk Cluster atau kelompok kerja, Database, Skema, dan nama Tabel tempat data dimuat.
11. (Opsional) Anda dapat memilih nama kolom untuk dimasukkan Pemetaan kolom untuk memetakan kolom dalam urutan file data input.
12. Pilih Muat data untuk memulai pemuatan data.

Ketika beban selesai, pesan ditampilkan apakah beban berhasil atau tidak. Jika berhasil, Anda sekarang dapat menggunakan SQL untuk memilih data dari tabel yang dimuat. Ketika ada kesalahan, kueri tampilan sistem STL_LOAD_ERRORS untuk mendapatkan detail lebih lanjut. Untuk informasi tentang kesalahan perintah COPY, lihat [STL_LOAD_ERRORS di Panduan Pengembang Database](#) Amazon Redshift.

Template perintah COPY yang digunakan untuk memuat data muncul di riwayat Query Anda. Template perintah COPY ini menunjukkan beberapa parameter yang digunakan, tetapi tidak dapat dijalankan langsung di tab editor. Untuk informasi selengkapnya tentang riwayat kueri, lihat [Melihat kueri dan riwayat tab](#).

Saat Anda memuat data ke dalam tabel baru, editor kueri v2 pertama-tama membuat tabel di database, lalu memuat data sebagai tindakan terpisah dalam alur kerja yang sama.

Untuk memuat data file lokal ke dalam tabel baru

Administrator editor kueri v2 Anda harus menentukan bucket Amazon S3 umum di jendela Pengaturan akun. File lokal secara otomatis diunggah ke bucket Amazon S3 umum yang digunakan oleh akun Anda, dan kemudian perintah COPY digunakan oleh editor kueri v2 untuk memuat data. Perintah COPY yang dihasilkan dan dijalankan oleh editor kueri v2 memuat jendela file lokal mendukung banyak parameter yang tersedia untuk sintaks perintah COPY untuk disalin dari Amazon S3. Untuk informasi tentang perintah COPY dan opsinya yang digunakan untuk memuat data dari Amazon S3, lihat [COPY dari Amazon S3](#) di Panduan Pengembang Database Amazon Redshift.

1. Konfirmasikan bahwa Anda terhubung ke database target di panel tampilan pohon editor kueri v2. Anda dapat membuat koneksi menggunakan menu konteks (klik kanan) ke cluster atau workgroup tempat data akan dimuat.
2. Pilih  Muat data.
3. Untuk Sumber data, pilih Muat dari file lokal.
4. Pilih Browse untuk menemukan file yang berisi data untuk Memuat file. Secara default, file dengan ekstensi .csv, .avro.parquet, dan .orc ditampilkan, tetapi Anda dapat memilih jenis file lainnya. Ukuran file maksimum adalah 100 MB.
5. Pilih format File untuk file yang akan diunggah. Format data yang didukung adalah CSV, JSON, DELIMITER, FIXEDWIDTH, SHAPEFILE, AVRO, PARQUET, dan ORC. Tergantung pada format file yang ditentukan, Anda dapat memilih opsi File masing-masing. Anda juga dapat memilih Data dienkripsi jika data dienkripsi dan masukkan Nama Sumber Daya Amazon (ARN) dari kunci KMS yang digunakan untuk mengenkripsi data.

Jika Anda memilih CSV atau DELIMITER, Anda juga dapat memilih karakter Delimiter dan apakah akan Abaikan baris header jika jumlah baris yang ditentukan sebenarnya adalah nama kolom dan bukan data yang akan dimuat.

6. (Opsional) Pengaturan lanjutan mendukung berbagai parameter konversi Data dan operasi Beban. Masukkan informasi ini sesuai kebutuhan untuk file Anda.

Untuk informasi selengkapnya tentang konversi data dan parameter pemuatan data, lihat [Parameter konversi data](#) dan [Operasi pemuatan data](#) di Panduan Pengembang Database Amazon Redshift.

7. Pilih Berikutnya.
8. Pilih Muat tabel baru.
9. Konfirmasikan atau pilih lokasi tabel Target termasuk Cluster atau workgroup, Database, dan Skema tempat data dimuat. Masukkan nama Tabel yang akan dibuat.
10. Pilih Buat tabel untuk membuat tabel menggunakan definisi yang ditampilkan.

Ringkasan tinjauan definisi tabel ditampilkan. Tabel dibuat dalam database. Untuk kemudian menghapus tabel, jalankan perintah DROP TABLE SQL. Untuk informasi selengkapnya, lihat [DROP TABLE](#) di Panduan Pengembang Database Amazon Redshift.

11. Pilih Muat data untuk memulai pemuatan data.

Ketika beban selesai, pesan ditampilkan yang menunjukkan apakah beban berhasil atau tidak. Jika berhasil, Anda sekarang dapat menggunakan SQL untuk memilih data dari tabel yang dimuat. Ketika ada kesalahan, kueri tampilan sistem STL_LOAD_ERRORS untuk mendapatkan detail lebih lanjut. Untuk informasi tentang kesalahan perintah COPY, lihat [STL_LOAD_ERRORS di Panduan Pengembang Database](#) Amazon Redshift.

Template perintah COPY yang digunakan untuk memuat data muncul di riwayat Query Anda. Template perintah COPY ini menunjukkan beberapa parameter yang digunakan, tetapi tidak dapat dijalankan langsung di tab editor. Untuk informasi selengkapnya tentang riwayat kueri, lihat [Melihat kueri dan riwayat tab](#).

Membuat kueri dengan Amazon Redshift

Anda dapat memasukkan kueri di editor atau memilih kueri yang disimpan dari daftar Kueri dan memilih Jalankan.

Secara default, Batas 100 diatur untuk membatasi hasil hingga 100 baris. Anda dapat mematikan opsi ini untuk mengembalikan set hasil yang lebih besar. Jika Anda mematikan opsi ini, Anda dapat menyertakan opsi LIMIT dalam pernyataan SQL Anda jika Anda ingin menghindari set hasil yang sangat besar. Untuk informasi selengkapnya, lihat [klausa ORDER BY](#) di Panduan Pengembang Database Amazon Redshift.

Untuk menampilkan rencana kueri di area hasil, aktifkan Jelaskan. Aktifkan grafik Jelaskan untuk hasil untuk juga menampilkan representasi grafis dari rencana penjelasan.

Untuk menyimpan kueri ke folder Kueri, pilih Simpan.

Untuk kueri yang berhasil, pesan sukses muncul. Jika kueri mengembalikan informasi, hasilnya akan ditampilkan di bagian Hasil. Jika jumlah hasil melebihi area tampilan, angka muncul di bagian atas area hasil. Anda dapat memilih angka untuk menampilkan halaman hasil yang berurutan.

Anda dapat memfilter dan mengurutkan Hasil untuk setiap kolom. Untuk memasukkan kriteria filter di header kolom hasil, arahkan kursor ke kolom untuk melihat menu



di mana Anda dapat memasukkan kriteria untuk memfilter kolom.

Jika kueri berisi kesalahan, editor kueri v2 menampilkan pesan kesalahan di area hasil. Pesan tersebut memberikan informasi tentang cara memperbaiki kueri.

Anda dapat mengekspor atau menyalin hasil kueri Anda dengan menggunakan menu konteks (klik kanan) di area hasil sebagai berikut:

- Pilih Salin ke dan JSON atau CSV untuk mengunduh baris yang dipilih ke file.
- Pilih Salin baris untuk menyalin baris yang dipilih ke clipboard.
- Pilih Salin baris dengan header untuk menyalin baris yang dipilih dengan header kolom ke clipboard.

Anda juga dapat memilih Ekspor di area hasil, lalu pilih JSON atau CSV untuk mengunduh seluruh rangkaian hasil baris ke file. Jumlah baris dalam kumpulan hasil mungkin dibatasi oleh opsi Limit atau limit klausa SQL dalam kueri. Ukuran maksimum dari set hasil yang diunduh adalah 5 MB.

Anda juga dapat menggunakan pintasan Ctrl+C di Windows atau Cmd+C di macOS untuk menyalin data dari halaman hasil saat ini ke clipboard. Jika tidak ada baris yang dipilih, maka sel dengan fokus disalin ke clipboard. Jika baris dipilih, maka baris yang dipilih disalin ke clipboard.

Untuk menambahkan tab kueri baru, pilih



ikon, lalu Editor, yang muncul di baris dengan tab kueri. Tab kueri menggunakan `Isolated session` atau tidak. Dengan sesi terisolasi, hasil perintah SQL, seperti membuat tabel sementara di satu tab editor, tidak terlihat di tab editor lain. Saat Anda membuka tab editor di editor kueri v2, defaultnya adalah sesi terisolasi.

Untuk menjalankan kueri

1. Di area kueri, lakukan salah satu hal berikut:
 - Masukkan kueri.
 - Tempelkan kueri yang Anda salin.
 - Pilih folder Kueri, buka menu konteks (klik kanan) kueri yang disimpan, dan pilih Buka kueri.
2. Konfirmasikan bahwa Anda memilih Cluster atau Workgroup yang benar, dan nilai Database untuk SQL yang Anda rencanakan untuk dijalankan.

Awalnya, Anda dapat memilih Cluster atau Workgroup Anda di tampilan pohon. Pilih Database Anda di tampilan pohon juga.

Anda dapat mengubah Cluster atau Workgroup, dan Database dalam setiap tab editor dengan kontrol drop-down yang terletak di dekat header sesi Terisolasi dari setiap tab editor.

Untuk setiap tab editor, Anda memilih apakah akan menjalankan SQL dalam sesi Terisolasi. Sesi terisolasi memiliki koneksi sendiri ke database. Gunakan untuk menjalankan SQL yang terisolasi dari sesi editor kueri lainnya. Untuk informasi selengkapnya tentang koneksi, lihat [Membuka editor kueri v2](#).

3. Pilih Jalankan.

Area Hasil membuka dan menampilkan hasil kueri.

Untuk menampilkan rencana penjelasan untuk kueri

1. Pilih kueri.
2. Aktifkan Jelaskan.

Secara default, grafik Jelaskan juga aktif.

3. Pilih Jalankan.

Kueri berjalan dan rencana penjelasan ditampilkan di area Hasil kueri.

Editor kueri v2 mendukung fitur-fitur berikut:

- Anda dapat membuat kueri dengan beberapa pernyataan SQL dalam satu tab kueri. Kueri dijalankan secara serial dan beberapa tab hasil terbuka untuk setiap kueri.
- Anda dapat membuat kueri dengan variabel sesi dan tabel sementara.
- Anda dapat membuat kueri dengan parameter yang dapat diganti yang ditunjuk oleh `${parameter}`. Anda dapat membuat kueri SQL Anda dengan beberapa parameter yang dapat diganti dan menggunakan parameter yang sama di beberapa tempat dalam pernyataan SQL Anda.

Saat kueri berjalan, sebuah jendela disajikan untuk memasukkan nilai parameter. Setiap kali Anda menjalankan kueri, jendela disajikan untuk memasukkan nilai parameter Anda.

Sebagai contoh, lihat [Contoh: Penjualan lebih besar dari parameter tertentu](#).

- Kueri diversi secara otomatis. Anda dapat memilih versi kueri yang lebih lama untuk dijalankan.
- Anda tidak perlu menunggu kueri selesai sebelum melanjutkan alur kerja Anda. Kueri Anda terus berjalan bahkan jika Anda menutup editor kueri.
- Saat membuat kueri, penyelesaian otomatis nama skema, tabel, dan kolom didukung.

Editor SQL mendukung fitur-fitur berikut:

- Tanda kurung awal dan akhir yang digunakan dalam SQL memiliki warna yang cocok. Garis vertikal ditampilkan di editor untuk membantu Anda mencocokkan tanda kurung.
- Anda dapat menciutkan dan memperluas bagian SQL Anda.
- Anda dapat mencari dan mengganti teks di SQL Anda.
- Anda dapat menggunakan tombol pintas untuk beberapa tugas pengeditan umum.
- Kesalahan SQL disorot di editor untuk lokasi area masalah yang nyaman.

Untuk demo fitur pengeditan, tonton video berikut: [Pengalaman Pengeditan Baru dan Ditingkatkan di editor kueri Amazon Redshift v2](#).

Contoh kueri

Berikut ini, Anda dapat menemukan deskripsi dari berbagai jenis kueri yang dapat Anda jalankan.

Data yang digunakan dalam banyak kueri ini berasal dari skema tickit sampel. Untuk informasi selengkapnya tentang memuat tickit data sampel, lihat [Memuat data ke dalam database](#).

Untuk informasi selengkapnya tentang data tickit sampel, lihat [Contoh database](#) di Panduan Pengembang Database Amazon Redshift.

Saat Anda menjalankan contoh kueri ini, konfirmasi bahwa Anda memilih database yang benar di editor, seperti `sample_data_dev`.

Topik

- [Contoh: Mengatur variabel sesi](#)
- [Contoh: Acara teratas berdasarkan total penjualan](#)
- [Contoh: Penjualan lebih besar dari parameter tertentu](#)
- [Contoh: Buat tabel sementara](#)
- [Contoh: Memilih dari tabel sementara](#)

Contoh: Mengatur variabel sesi

Perintah berikut menetapkan parameter konfigurasi `search_path` server ke publik untuk sesi tersebut. Untuk informasi selengkapnya, lihat [SET](#) dan [search_path](#) di Panduan Pengembang Database Amazon Redshift.

```
set search_path to public;
```

Contoh: Acara teratas berdasarkan total penjualan

Kueri berikut menemukan acara dengan penjualan terbanyak.

```
select eventname, count(salesid) totalorders, sum(pricepaid) totalsales
from sales, event
where sales.eventid=event.eventid
group by eventname
order by 3;
```

Berikut ini adalah sebagian daftar hasil.

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
...		

Contoh: Penjualan lebih besar dari parameter tertentu

Kueri berikut menemukan penjualan di mana jumlah yang terjual lebih besar dari parameter yang ditentukan oleh `${numberoforders}`. Ketika nilai parameternya 7, hasilnya adalah 60 baris.

Saat Anda menjalankan kueri, editor kueri v2 menampilkan jendela formulir kueri Jalankan untuk mengumpulkan nilai parameter dalam pernyataan SQL.

```
select salesid, qtysold
from sales
where qtysold > ${numberoforders}
order by 2;
```

Berikut ini adalah sebagian daftar hasil.

```
salesid qtysold
```

```

20005 8
21279 8
130232 8
42737 8
74681 8
67103 8
105533 8
91620 8
121552 8
...

```

Contoh: Buat tabel sementara

Pernyataan berikut menciptakan tabel `eventsalestemp` sementara dengan memilih informasi dari tabel penjualan dan acara.

```

create temporary table eventsalestemp as
select eventname, count(salesid) totalorders, sum(pricepaid) totalsales
from sales, event
where sales.eventid=event.eventid
group by eventname;

```

Contoh: Memilih dari tabel sementara

Pernyataan berikut memilih acara, total pesanan, dan total penjualan dari `eventsalestemp` tabel sementara, dipesan berdasarkan total pesanan.

```

select eventname, totalorders, totalsales
from eventsalestemp
order by 2;

```

Berikut ini adalah sebagian daftar hasil.

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
...		

Notebook di Amazon Redshift

Anda dapat menggunakan buku catatan untuk mengatur, membuat anotasi, dan berbagi beberapa kueri SQL dalam satu dokumen. Anda dapat menambahkan beberapa kueri SQL dan sel Markdown ke buku catatan. Notebook menyediakan cara untuk mengelompokkan kueri dan penjelasan yang terkait dengan analisis data dalam satu dokumen dengan menggunakan beberapa kueri dan sel Markdown. Anda dapat menambahkan teks dan memformat tampilan menggunakan sintaks Markdown untuk memberikan konteks dan informasi tambahan untuk tugas analisis data Anda. Anda dapat membagikan buku catatan Anda dengan anggota tim.

Untuk menggunakan buku catatan, Anda harus menambahkan izin untuk buku catatan ke kepala IAM Anda (pengguna IAM atau peran IAM). Sebagai praktik terbaik, kami merekomendasikan untuk melampirkan kebijakan izin ke peran IAM dan kemudian menetapkan ke pengguna dan grup sesuai kebutuhan. Untuk informasi selengkapnya, lihat [Identitas dan manajemen akses di Amazon Redshift](#). Anda dapat menambahkan izin ke salah satu kebijakan terkelola editor kueri v2. Untuk informasi selengkapnya, lihat [Mengakses editor kueri v2](#).

Anda dapat menjalankan semua sel notebook secara berurutan. Sel kueri SQL notebook memiliki sebagian besar kemampuan yang sama dengan tab editor kueri. Untuk informasi selengkapnya, lihat [Membuat kueri dengan Amazon Redshift](#). Berikut ini adalah perbedaan antara tab query editor dan sel SQL di notebook.

- Tidak ada kontrol untuk dijalankan Explain pada pernyataan SQL di notebook.
- Anda hanya dapat membuat satu bagan per sel SQL di buku catatan.

Anda dapat mengekspor dan mengimpor buku catatan ke file yang dibuat dengan editor kueri v2. Ekstensi file adalah .ipynb dan ukuran file bisa maksimal 5 MB. Sel SQL dan Markdown disimpan dalam file. Cluster atau workgroup dan database tidak disimpan di notebook yang diekspor. Saat Anda membuka buku catatan yang diimpor, Anda memilih cluster atau workgroup dan database tempat menjalankannya. Setelah menjalankan sel SQL, Anda kemudian dapat memilih di tab hasil apakah akan menampilkan halaman hasil saat ini sebagai bagan. Kumpulan hasil kueri tidak disimpan di buku catatan.

Membuat buku catatan

Anda dapat membuat buku catatan untuk mengatur, membuat anotasi, dan berbagi beberapa kueri SQL dalam satu dokumen.

Untuk membuat buku catatan

1. Dari menu navigator, pilih ikon Editor



).

2. Pilih ikon plus



),

lalu pilih Notebook.

Secara default, sel kueri SQL muncul di buku catatan.

3. Di sel kueri SQL, lakukan salah satu hal berikut:

- Masukkan kueri.
- Tempelkan kueri yang Anda salin.

4. (Opsional) Pilih ikon plus



),

lalu pilih Markdown untuk menambahkan sel Markdown di mana Anda dapat memberikan teks deskriptif atau penjelasan menggunakan sintaks Markdown standar.

5. (Opsional) Pilih ikon plus



),

lalu pilih SQL untuk menyisipkan sel SQL.

Anda dapat mengganti nama notebook menggunakan ikon pensil



.

Dari ikon menu



),

Anda juga dapat melakukan operasi berikut pada buku catatan:

- 

Bagika

dengan tim saya — Untuk berbagi buku catatan dengan tim Anda seperti yang ditentukan oleh tag. Untuk berbagi buku catatan dengan tim Anda, pastikan bahwa Anda memiliki tag utama yang `sqlworkbench-team` disetel ke nilai yang sama dengan anggota tim lainnya di akun Anda.

Misalnya, administrator mungkin menetapkan nilai `accounting-team` untuk semua orang di

departemen akuntansi. Sebagai contoh, lihat [izin yang diperlukan untuk menggunakan editor kueri v2](#).



- Untuk mengekspor buku catatan ke file lokal dengan `.ipynb` ekstensi.



impor - Untuk mengimpor kueri dari file lokal ke sel di buku catatan. Anda dapat mengimpor file dengan `.sql` dan `.txt` ekstensi.



versi — Untuk membuat versi notebook. Untuk melihat versi buku catatan, navigasikan ke buku catatan yang disimpan dan buka Riwayat versi.



— Untuk membuat salinan buku catatan dan membukanya di tab notebook baru.



- Untuk menampilkan pintasan yang tersedia saat membuat buku catatan.

Ekspor

Kueri

Simpan

Duplika

Pintasa

Mengimpor ke notebook

Anda dapat mengimpor seluruh notebook atau sel SQL individual ke dalam notebook editor kueri v2.

Untuk mengimpor seluruh buku catatan dari file lokal ke buku catatan saya, pilih



Impor, lalu pilih Impor buku catatan. Arahkan ke `.ipynb` file yang berisi buku catatan Anda. Notebook diimpor ke folder notebook yang sedang terbuka. Anda kemudian dapat membuka notebook di editor notebook.

Untuk mengimpor kueri dari file lokal ke sel SQL di buku catatan, pilih



Impor, lalu pilih Impor kueri. Pada jendela kueri impor, ikuti petunjuk di layar untuk memilih file dan folder yang dapat diimpor sebagai kueri ke buku catatan baru atau buku catatan yang sudah ada. File harus memiliki ekstensi `.sql` atau `.txt`. Setiap kueri dapat mencapai 10.000 karakter. Saat menambahkan ke buku catatan yang ada, Anda memilih buku catatan mana dari semua buku catatan dalam daftar buku catatan tersimpan. Kueri yang diimpor ditambahkan sebagai sel SQL di akhir notebook. Ketika Anda memilih notebook baru, Anda memilih nama notebook dan itu dibuat di folder notebook yang disimpan saat ini terbuka.

Note

Saat membuat `.sql` file di macOS menggunakan TextEdit aplikasi, Anda mungkin mengalami masalah saat ekstensi tersembunyi tambahan ditambahkan ke file. Misalnya, file bernama `Test.sql` dibuat TextEdit mungkin akan disimpan sebagai `fileTest.sql.rtf`. Editor kueri v2 tidak mendukung file dengan `.rtf` ekstensi. Namun, jika Anda membuat `.sql` file menggunakan TextEdit, dan menyimpannya sebagai file teks biasa, file tersebut memiliki `.txt` ekstensi tersembunyi tambahan. Misalnya, file bernama `Text.sql` mungkin disimpan sebagai `Text.sql.txt`. Tidak seperti `.rtf` ekstensi, editor kueri v2 mendukung file dengan `.txt` ekstensi, sehingga `Text.sql.txt` didukung saat mengimpor kueri ke notebook.

Untuk demo notebook, tonton video berikut: [Amazon Redshift SQL Notebooks](#) di editor kueri v2.

Menanyakan AWS Glue Data Catalog

Anda dapat menggunakan editor kueri v2 untuk menanyakan data yang dikatalogkan di katalog Anda AWS Glue Data Catalog dengan menggunakan perintah SQL tertentu dan memberikan izin yang diuraikan di bagian ini. Secara default, terdaftar sebagai editor kueri v2 database bernama `awsdatacatalog`. AWS Glue Data Catalog Menanyakan tidak AWS Glue Data Catalog tersedia di semua Amazon Wilayah AWS Redshift. Gunakan perintah `SHOW` untuk menentukan apakah kemampuan ini tersedia. Untuk informasi lebih lanjut tentang AWS Glue, lihat [Apa itu AWS Glue?](#) di Panduan AWS Glue Pengembang.

Note

Query hanya AWS Glue Data Catalog didukung di cluster tipe node Amazon RA3 Redshift dan Amazon Redshift Serverless.

Anda dapat mengonfigurasi gudang data dan melihat objek AWS Glue database yang dikatalogkan menggunakan perintah SQL berikut:

- `SHOW` - untuk menampilkan `awsdatacatalog` apakah dipasang untuk gudang data yang saat ini terhubung. Misalnya, untuk menampilkan nilai `data_catalog_auto_mount` parameter, jalankan:

```
SHOW data_catalog_auto_mount;
```

Untuk informasi selengkapnya, lihat [TAMPILKAN](#) di Panduan Pengembang Database Amazon Redshift.

- **ALTER SYSTEM** — untuk mengubah konfigurasi tingkat sistem. `data_catalog_auto_mount` Misalnya, untuk mengubah nilai `data_catalog_auto_mount` parameter menjadion, jalankan:

```
ALTER SYSTEM SET data_catalog_auto_mount = on;
```

Perubahan akan berlaku ketika klaster yang disediakan di-boot ulang atau grup kerja tanpa server secara otomatis dijeda dan dilanjutkan. Untuk informasi selengkapnya, lihat [MENGUBAH SISTEM di Panduan](#) Pengembang Database Amazon Redshift.

- **TAMPILKAN SKEMA** - menunjukkan daftar skema. Skema dalam database bernama `awsdatacatalog` mewakili database yang dikatalogkan AWS Glue dalam. AWS Glue Data Catalog Misalnya, untuk menampilkan skema ini, jalankan:

```
SHOW SCHEMAS FROM DATABASE awsdatacatalog;
```

Untuk informasi selengkapnya, lihat [TAMPILKAN SKEMA](#) di Panduan Pengembang Database Amazon Redshift.

- **TAMPILKAN TABEL** - menunjukkan daftar tabel dalam skema. Misalnya, untuk menampilkan tabel dalam AWS Glue Data Catalog database bernama `awsdatacatalog` yang sedang myglue dijalankan skema:

```
SHOW TABLES FROM SCHEMA awsdatacatalog.myschema;
```

Untuk informasi selengkapnya, lihat [TAMPILKAN TABEL](#) di Panduan Pengembang Database Amazon Redshift.

- **TAMPILKAN KOLOM** - menunjukkan daftar kolom dalam tabel. Misalnya, untuk menampilkan kolom dalam AWS Glue Data Catalog database bernama `awsdatacatalog` yang berada dalam skema myglue dan tabel mytable berjalan:

```
SHOW COLUMNS FROM TABLE awsdatacatalog.myglue.mytable;
```

Untuk informasi selengkapnya, lihat [TAMPILKAN KOLOM](#) di Panduan Pengembang Database Amazon Redshift.

Untuk memberikan izin kepada pengguna atau peran IAM Anda untuk menanyakan, AWS Glue Data Catalog

1. Di panel tampilan pohon, sambungkan ke database awal di klaster yang disediakan atau grup kerja tanpa server menggunakan metode autentikasi nama pengguna dan kata sandi Database. Misalnya, sambungkan ke dev database menggunakan pengguna admin dan kata sandi yang Anda gunakan saat membuat klaster atau grup kerja.
2. Di tab editor, jalankan pernyataan SQL berikut untuk memberikan akses pengguna IAM ke file. AWS Glue Data Catalog

```
GRANT USAGE ON DATABASE awsdatalog to "IAM:myIAMUser"
```

Di *IAM:myIAMUser* mana pengguna IAM yang ingin Anda berikan hak istimewa penggunaan ke. AWS Glue Data Catalog Atau, Anda dapat memberikan hak istimewa penggunaan *IAM:myIAMRole* untuk peran IAM.

3. Di panel tampilan pohon, edit atau hapus sambungan ke klaster atau grup kerja yang sebelumnya Anda buat. Connect ke cluster atau workgroup Anda dengan salah satu cara berikut:
 - Untuk mengakses awsdatalog database dari cluster, Anda harus menggunakan metode otentikasi Kredensial sementara menggunakan identitas IAM Anda. Untuk informasi selengkapnya tentang metode otentikasi ini, lihat [Menghubungkan ke database Amazon Redshift](#). Administrator editor kueri v2 Anda mungkin perlu mengonfigurasi pengaturan Akun untuk akun untuk menampilkan metode otentikasi ini di jendela koneksi.
 - Untuk mengakses awsdatalog database dari workgroup, Anda harus menggunakan metode otentikasi Pengguna Federasi. Untuk informasi selengkapnya tentang metode otentikasi ini, lihat [Menghubungkan ke database Amazon Redshift](#).
4. Dengan hak istimewa yang diberikan, Anda dapat menggunakan identitas IAM Anda untuk menjalankan SQL terhadap Anda. AWS Glue Data Catalog

Setelah menghubungkan, Anda dapat menggunakan editor kueri v2 untuk menanyakan data yang dikatalogkan. AWS Glue Data Catalog Pada panel tampilan pohon editor kueri v2, pilih cluster atau workgroup dan database. awsdatalog Di panel editor atau buku catatan, konfirmasi klaster atau grup kerja yang benar dipilih. Basis data yang dipilih harus menjadi basis data Amazon Redshift awal seperti. dev Untuk informasi tentang membuat kueri, lihat [Membuat kueri dengan Amazon Redshift](#) dan. [Notebook di Amazon Redshift](#) Nama database awsdatalog dicadangkan

untuk referensi database Katalog Data eksternal di akun Anda. Query terhadap `awsdatacatalog` database hanya bisa read-only. Gunakan notasi tiga bagian untuk mereferensikan tabel dalam pernyataan `SELECT` Anda. Di mana bagian pertama adalah nama database, bagian kedua adalah nama AWS Glue database, dan bagian ketiga adalah nama AWS Glue tabel.

```
SELECT * FROM awsdatacatalog.<aws-glue-db-name>.<aws-glue-table-name>;
```

Anda dapat melakukan berbagai skenario yang membaca AWS Glue Data Catalog data dan mengisi tabel Amazon Redshift.

Contoh SQL berikut bergabung dengan dua tabel yang didefinisikan dalam AWS Glue

```
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;
```

Contoh SQL berikut membuat tabel Amazon Redshift dan mengisinya dengan data dari gabungan dua tabel AWS Glue

```
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;
```

Menanyakan tabel Amazon S3 (pratinjau)

Anda dapat menggunakan editor kueri v2 untuk menanyakan data yang disimpan di katalog Amazon S3table yang dipasang ke file. AWS Glue Data Catalog Katalog tabel Amazon S3 dipasang ke saat pembuatan, dan secara otomatis muncul sebagai database eksternal AWS Glue Data Catalog pada semua cluster yang disediakan dan grup kerja tanpa server yang sama di bawah akun yang sama. AWS Region Untuk informasi selengkapnya tentang mengakses tabel Amazon S3 menggunakan Amazon Redshift, lihat Mengakses tabel [Amazon S3 dengan Amazon Redshift di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon](#).

Meminta data lake

Anda dapat meminta data di danau data Amazon S3 dengan mengikuti serangkaian tugas dalam tutorial ini. Pertama, Anda membuat skema eksternal untuk referensi database eksternal di [AWS Glue Data Catalog](#). Kemudian, Anda dapat meminta data di danau data Amazon S3.

Demo: Kueri danau data

Untuk demo tentang cara menanyakan data lake, tonton video berikut. [Kueri data lake Anda dari editor kueri Amazon Redshift v2](#).

Prasyarat

Sebelum Anda bekerja dengan data lake Anda di editor kueri v2, konfirmasikan bahwa hal berikut telah disiapkan di lingkungan Amazon Redshift Anda:

- Merayapi data Amazon S3 Anda AWS Glue menggunakan dan mengaktifkan Katalog Data Anda. AWS Lake Formation
- Buat peran IAM untuk Amazon Redshift menggunakan Katalog Data AWS Glue yang diaktifkan untuk. AWS Lake Formation Untuk detail tentang prosedur ini, lihat [Untuk membuat peran IAM untuk Amazon Redshift menggunakan AWS Glue Data Catalog enabled](#) for. AWS Lake Formation Untuk informasi lebih lanjut tentang penggunaan Redshift Spectrum dan Lake Formation, lihat Menggunakan [Redshift](#) Spectrum dengan. AWS Lake Formation
- Berikan izin SELECT pada tabel untuk kueri di database Lake Formation. Untuk detail tentang prosedur ini, lihat [Untuk memberikan izin SELECT pada tabel untuk kueri di database Lake Formation](#).

Anda dapat memverifikasi di konsol Lake Formation (<https://console.aws.amazon.com/lakeformation/>), bagian Izin, halaman izin data lake, bahwa peran IAM, AWS Glue database, dan tabel memiliki izin yang tepat.

- Konfirmasikan bahwa pengguna Anda yang terhubung memiliki izin untuk membuat skema di database Amazon Redshift dan mengakses data di danau data Anda. Saat Anda terhubung ke database di editor kueri v2, Anda memilih metode otentikasi yang menyertakan kredensial, yang dapat berupa pengguna database atau pengguna IAM. Pengguna yang terhubung harus memiliki izin dan hak istimewa database yang tepat, seperti file. `superuser admin` Pengguna Amazon Redshift yang membuat cluster atau workgroup memiliki `superuser` hak istimewa dan dapat membuat skema dan mengelola database Redshift. Untuk informasi selengkapnya tentang

menghubungkan ke database dengan editor kueri v2, lihat [Menghubungkan ke database Amazon Redshift](#).

Membuat skema eksternal

Untuk melakukan kueri data di danau data Amazon S3, buat skema eksternal terlebih dahulu. Skema eksternal mereferensikan database eksternal di file. [AWS Glue Data Catalog](#)

1. Dalam tampilan Editor editor kueri v2, pilih



Buat, lalu pilih Skema.

2. Masukkan nama Skema.
3. Untuk tipe Skema, pilih Eksternal.
4. Dalam detail Katalog Data, Wilayah default ke tempat database Redshift AWS Region Anda berada.
5. Pilih AWS Glue database yang akan dipetakan skema eksternal dan yang berisi referensi ke AWS Glue tabel.
6. Pilih peran IAM untuk Amazon Redshift yang memiliki izin yang diperlukan untuk menanyakan data di Amazon S3.
7. Secara opsional, pilih peran IAM yang memiliki izin ke Katalog Data.
8. Pilih Buat skema.

Skema muncul di bawah database Anda di panel tampilan pohon.

Saat membuat skema, jika Anda menerima kesalahan izin ditolak untuk database Anda, periksa apakah pengguna yang terhubung memiliki hak istimewa database untuk membuat skema.

Menanyakan data di danau data Amazon S3 Anda

Anda menggunakan skema yang Anda buat dalam prosedur sebelumnya.

1. Di panel tampilan pohon, pilih skema.
2. Untuk melihat definisi tabel, pilih tabel. Kolom tabel dan tipe data ditampilkan.
3. Untuk menanyakan tabel, pilih tabel dan di menu konteks (klik kanan), pilih Pilih tabel untuk menghasilkan kueri.

4. Jalankan kueri di Editor.

Contoh SQL berikut dihasilkan oleh query editor v2 untuk query semua baris dalam AWS Glue tabel bernama `flightscsv`. Kolom dan baris yang ditunjukkan dalam output dipotong untuk kesederhanaan.

```
SELECT * FROM "dev"."mydatalake_schema"."flightscsv";
```

year	quarter	month	dom	day_of_week	fl_date	unique_carrier	airline_id
2016	4	10	19	3	10/19/16	00	20304
2016	4	10	19	3	10/19/16	00	20304
2016	4	10	19	3	10/19/16	00	20304
2016	4	10	19	3	10/19/16	00	20304
...							

Datashares

Anda dapat membuat datashare sehingga pengguna di cluster lain dapat melakukan query data. Cluster yang berisi data yang ingin Anda bagikan disebut cluster produser. Anda membuat datashare di cluster produser untuk objek database yang ingin Anda bagikan. Anda dapat berbagi skema, tabel, tampilan, dan fungsi yang ditentukan pengguna SQL (). UDFs Cluster yang ingin Anda bagikan datanya disebut cluster konsumen. Pada cluster konsumen, Anda membuat database dari datashare. Kemudian, pengguna di cluster konsumen dapat melakukan query data. Untuk informasi selengkapnya, lihat [Memulai berbagi data](#) di Panduan Pengembang Database Amazon Redshift.

Membuat datashares

Anda membuat datashare pada cluster yang ingin Anda gunakan sebagai cluster produser. Untuk mempelajari lebih lanjut tentang pertimbangan data, lihat [Pertimbangan berbagi data di Amazon Redshift di](#) Panduan Pengembang Database Amazon Redshift.

1. Pilih database pada cluster produser yang ingin Anda gunakan.
2. Buat datashare. Contoh:

```
create datashare mysource;
```

3. Tetapkan izin pada datashare. Contoh:

```
grant alter, share on datashare mysource to admin;
```

4. Tetapkan izin pada objek database yang ingin Anda bagikan. Contoh:

```
alter datashare mysource add schema public;
```

```
alter datashare mysource add table public.event;
```

5. Tetapkan izin pada namespace cluster konsumen untuk mengakses datashare. Contoh:

```
grant usage on datashare mysource to namespace '2b12345-1234-5678-9012-  
bb1234567890';
```

Menampilkan datashares

Anda dapat menampilkan datashares yang telah Anda buat di cluster produser.

1. Pilih cluster produser.
2. Tampilkan datashares. Contoh:

```
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
```

Membuat database konsumen

Pada cluster konsumen, Anda membuat database dari datashare. Langkah-langkah ini menjelaskan cara berbagi data antara dua cluster di akun yang sama. Untuk informasi tentang berbagi data di

seluruh AWS akun, lihat [Berbagi data di seluruh AWS akun](#) di Panduan Pengembang Database Amazon Redshift.

Anda dapat menggunakan perintah SQL atau panel tampilan pohon editor kueri v2 untuk membuat database.

Untuk menggunakan SQL

1. Buat database dari datashare untuk akun Anda dan namespace klaster produser. Contoh:

```
create database share_db from datashare mysource of account '123456789012'  
namespace 'p1234567-8765-4321-p10987654321';
```

2. Tetapkan izin sehingga pengguna dapat mengakses database dan skema. Contoh:

```
grant usage on database share_db to usernames;
```

```
grant usage on schema public to usernames;
```

Untuk menggunakan panel tampilan pohon editor kueri v2

1. Pilih



Buat, lalu pilih Database.

2. Masukkan nama Database.
3. (Opsional) Pilih Pengguna dan grup, dan pilih pengguna Database.
4. Pilih Buat menggunakan datashare.
5. Pilih datashare.
6. Pilih Buat basis data.

Database



datashare baru ditampilkan di panel tampilan pohon editor kueri v2.

7. Tetapkan izin sehingga pengguna dapat mengakses database dan skema. Contoh:

```
grant usage on database share_db to usernames;
```

```
grant usage on schema public to usernames;
```

Menanyakan objek datashare

Pada cluster konsumen, Anda dapat menanyakan objek datashare menggunakan nama objek yang sepenuhnya memenuhi syarat yang diekspresikan dengan notasi tiga bagian: database, skema, dan nama objek.

1. Di panel tampilan pohon editor kueri v2, pilih skema.
2. Untuk melihat definisi tabel, pilih tabel.

Kolom tabel dan tipe data ditampilkan.

3. Untuk menanyakan tabel, pilih tabel dan gunakan menu konteks (klik kanan) untuk memilih Pilih tabel.
4. Tabel kueri menggunakan perintah SELECT. Contoh:

```
select top 10 * from test_db.public.event;
```

Kueri terjadwal dengan editor kueri v2

Dengan editor kueri Amazon Redshift v2, Anda dapat mengotomatiskan kueri SQL agar berjalan sesuai jadwal. Kueri terjadwal adalah pernyataan SQL yang berjalan secara otomatis pada waktu atau interval tertentu, memungkinkan Anda mengelola operasi data berulang dan tugas analitik secara efisien. Anda mungkin ingin menjadwalkan kueri jika ingin merampingkan pemrosesan batch, membuat laporan rutin, atau memelihara jalur data dalam lingkungan Amazon Redshift mereka.

Kueri terjadwal memfasilitasi otomatisasi alur kerja ekstrak, transformasi, dan pemuatan (ETL), menyegarkan dasbor dengan up-to-date wawasan, dan mengoperasionalkan berbagai rutinitas manajemen data. Halaman-halaman berikut merinci proses pembuatan, konfigurasi, dan pengelolaan kueri terjadwal untuk mengoptimalkan beban kerja Amazon Redshift Anda.

Membuat jadwal kueri dengan editor kueri v2

Anda dapat membuat jadwal untuk menjalankan pernyataan SQL dengan editor kueri Amazon Redshift v2. Anda membuat jadwal untuk menjalankan pernyataan SQL Anda pada interval waktu yang sesuai dengan kebutuhan bisnis Anda. Saat kueri terjadwal berjalan, kueri dimulai oleh Amazon EventBridge dan menggunakan Amazon Redshift Data API.

Untuk membuat jadwal untuk menjalankan pernyataan SQL

1. Pada



tampilan Editor, pilih



Jadwal untuk membuat jadwal untuk menjalankan pernyataan SQL.

2. Ketika Anda menentukan jadwal, Anda memberikan informasi berikut.

- Peran IAM yang mengasumsikan izin yang diperlukan untuk menjalankan kueri. Peran IAM ini juga dilampirkan ke cluster atau workgroup Anda.
- Nilai otentikasi untuk salah satu AWS Secrets Manager atau kredensi sementara untuk mengotorisasi akses klaster atau grup kerja Anda. Metode otentikasi ini didukung oleh API Data. Untuk informasi selengkapnya, lihat [Mengautentikasi kueri terjadwal](#).
- Cluster atau workgroup tempat database Anda berada.
- Nama database yang berisi data yang akan ditanyakan.
- Nama kueri terjadwal dan deskripsinya. Editor kueri v2 mengawali nama kueri terjadwal yang Anda berikan dengan "QS2-". Editor kueri v1 mengawali nama kueri terjadwal dengan "QS-".
- Pernyataan SQL yang akan dijalankan pada jadwal.
- Frekuensi jadwal dan opsi pengulangan atau nilai diformat cron yang menentukan jadwal. Untuk informasi selengkapnya, lihat [Ekspresi Cron di](#) Panduan Pengguna CloudWatch Acara Amazon.
- Secara opsional, Anda dapat mengaktifkan notifikasi Amazon SNS standar untuk memantau kueri terjadwal. Anda mungkin perlu mengonfirmasi alamat email yang Anda berikan ke notifikasi Amazon SNS. Periksa email Anda untuk tautan untuk mengonfirmasi alamat email untuk notifikasi Amazon SNS. Untuk informasi selengkapnya, lihat [Pemberitahuan email](#) di Panduan Pengembang Layanan Pemberitahuan Sederhana Amazon. Jika kueri Anda sedang

dijalankan tetapi Anda tidak melihat pesan yang dipublikasikan di topik SNS Anda, lihat [Aturan saya berjalan, tetapi saya tidak melihat pesan apa pun yang dipublikasikan ke topik Amazon SNS saya](#) di Panduan Pengguna EventBridge Amazon.

3. Pilih Jadwalkan kueri untuk menyimpan dan mengaktifkan jadwal dan menambahkan jadwal ke daftar kueri dalam tampilan Kueri terjadwal.



Tampil

kueri terjadwal mencantumkan semua kueri terjadwal untuk klaster dan grup kerja Anda. Dengan tampilan ini, Anda dapat menampilkan detail kueri jadwal, mengaktifkan atau menonaktifkan jadwal, mengedit jadwal, dan menghapus kueri terjadwal. Saat melihat detail kueri, Anda juga dapat melihat riwayat menjalankan kueri dengan jadwal.

Note

Proses kueri jadwal hanya tersedia di daftar riwayat Jadwal selama 24 jam. Kueri yang berjalan pada jadwal tidak muncul di tampilan riwayat Kueri editor kueri v2.

Demo penjadwalan kueri

Untuk demo penjadwalan kueri, tonton video berikut. [Demo video penjadwalan kueri](#).

Menyiapkan izin untuk menjadwalkan kueri

Untuk menjadwalkan kueri, pengguna AWS Identity and Access Management (IAM) yang menentukan jadwal dan peran IAM yang terkait dengan jadwal harus dikonfigurasi dengan izin IAM untuk menggunakan Amazon dan Amazon EventBridge Redshift Data API. Untuk menerima email dari kueri terjadwal, notifikasi Amazon SNS yang Anda tentukan secara opsional harus dikonfigurasi juga.

Berikut ini menjelaskan tugas untuk menggunakan kebijakan AWS terkelola untuk memberikan izin, tetapi tergantung pada lingkungan Anda, Anda mungkin ingin mengurangi izin yang diizinkan.

Untuk pengguna IAM yang masuk ke editor kueri v2, edit pengguna IAM menggunakan konsol IAM (). <https://console.aws.amazon.com/iam/>

- Selain izin untuk menjalankan operasi Amazon Redshift dan editor kueri v2, lampirkan `AmazonEventBridgeFullAccess` kebijakan `AmazonRedshiftDataFullAccess` AWS dan terkelola ke pengguna IAM.
- Atau, tetapkan izin ke peran dan tetapkan peran tersebut kepada pengguna.

Lampirkan kebijakan yang mengizinkan `sts:AssumeRole` izin ke ARN sumber daya peran IAM yang Anda tentukan saat Anda menentukan kueri terjadwal. Untuk informasi selengkapnya tentang mengambil peran, lihat [Memberikan izin pengguna untuk beralih peran](#) di Panduan Pengguna IAM.

Contoh berikut menunjukkan kebijakan izin yang mengasumsikan peran IAM `myRedshiftRole` dalam akun. `123456789012` Peran IAM juga `myRedshiftRole` merupakan peran IAM yang dilampirkan ke cluster atau workgroup tempat kueri terjadwal berjalan.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AssumeIAMRole",
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": [
        "arn:aws:iam::123456789012:role/myRedshiftRole"
      ]
    }
  ]
}
```

Perbarui kebijakan kepercayaan peran IAM yang digunakan untuk menjadwalkan kueri agar pengguna IAM mengasumsikan itu.

```
{
  "Sid": "AssumeRole",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::123456789012:user/myIAMUsername"
  },
  "Action": "sts:AssumeRole"
}
```

```
}
```

Untuk peran IAM yang Anda tentukan agar kueri terjadwal dijalankan, edit peran IAM menggunakan konsol IAM (). <https://console.aws.amazon.com/iam/>

- Lampirkan kebijakan `AmazonRedshiftDataFullAccess` dan `AmazonEventBridgeFullAccess` AWS terkelola ke peran IAM. Kebijakan `AmazonRedshiftDataFullAccess` terkelola hanya mengizinkan `redshift-serverless:GetCredentials` izin untuk grup kerja Redshift Tanpa Server yang ditandai dengan kunci. `RedshiftDataFullAccess`

Mengautentikasi kueri terjadwal

Saat menjadwalkan kueri, Anda menggunakan salah satu metode otentikasi berikut saat SQL berjalan. Setiap metode memerlukan kombinasi input yang berbeda pada editor kueri v2. Metode otentikasi ini didukung oleh Data API yang digunakan untuk menjalankan pernyataan SQL Anda.

Pengguna database atau peran yang digunakan untuk menjalankan query harus memiliki hak istimewa database yang diperlukan. Misalnya, untuk memberikan `IAMR:MyRedshiftQEv2Scheduler` hak istimewa ke `tabelmytable`, jalankan perintah SQL berikut.

```
GRANT all ON TABLE mytable TO "IAMR:MyRedshiftQEv2Scheduler";
```

Untuk melihat daftar pengguna database di cluster atau workgroup Anda, kueri tampilan `PG_USER_INFO` sistem.

Note

Setiap grup kerja Redshift Tanpa Server yang Anda jadwalkan kueri akan ditandai dengan kuncinya. `RedshiftDataFullAccess` Untuk informasi selengkapnya, lihat [Mengotorisasi akses ke API Data Amazon Redshift](#).

Sebagai alternatif untuk menandai workgroup, Anda dapat menambahkan kebijakan inline ke peran IAM (yang ditentukan dengan jadwal) yang memungkinkan. `redshift-serverless:GetCredentials` Contoh:

JSON

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

AWS Secrets Manager

Dengan metode ini, berikan nilai rahasia untuk rahasia-arn yang disimpan di AWS Secrets Manager. Rahasia ini berisi kredensial untuk terhubung ke database Anda. Anda mungkin telah membuat rahasia dengan kredensial yang tepat saat membuat kluster atau grup kerja. Rahasiannya harus ditandai dengan kunci `RedshiftDataFullAccess`. Jika kunci tag belum ada, gunakan AWS Secrets Manager konsol untuk menambahkannya. Untuk informasi tentang membuat rahasia, lihat [Membuat rahasia untuk kredensi koneksi database](#).

Untuk informasi selengkapnya tentang izin minimum, lihat [Membuat dan Mengelola Rahasia dengan AWS Secrets Manager](#) di Panduan AWS Secrets Manager Pengguna.

Kredensial sementara

Dengan metode ini, berikan nama Database dan nilai pengguna Database Anda saat menghubungkan ke database dalam sebuah cluster. Anda hanya perlu memberikan nama Database Anda saat menghubungkan ke database di workgroup.

Saat menyambung ke kluster, `AmazonRedshiftDataFullAccess` kebijakan tersebut mengizinkan `redshift_data_api_user` izin bernama pengguna `database:redshift:GetClusterCredentials`. Jika Anda ingin menggunakan pengguna database yang berbeda untuk menjalankan pernyataan SQL, tambahkan kebijakan ke peran IAM

yang dilampirkan ke cluster Anda untuk mengizinkan. `redshift:GetClusterCredentials`
 Contoh kebijakan berikut memungkinkan pengguna database `awsuser` dan `myuser`.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "UseTemporaryCredentialsForAllDbUsers",
      "Effect": "Allow",
      "Action": "redshift:GetClusterCredentials",
      "Resource": [
        "arn:aws:redshift:*:*:dbuser:*/awsuser",
        "arn:aws:redshift:*:*:dbuser:*/myuser"
      ]
    }
  ]
}
```

Menyiapkan izin untuk melihat riwayat kueri jadwal

Untuk memungkinkan pengguna melihat riwayat kueri jadwal, edit peran IAM (yang ditentukan dengan jadwal) Hubungan kepercayaan untuk menambahkan izin.

Berikut ini adalah contoh kebijakan kepercayaan dalam peran IAM yang memungkinkan pengguna IAM *myIAMusername* untuk melihat riwayat kueri jadwal. Alih-alih mengizinkan `sts:AssumeRole` izin pengguna IAM, Anda dapat memilih untuk mengizinkan peran IAM izin ini.

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"
    }
  ]
}

```

Memantau kueri terjadwal

Untuk topik Amazon SNS yang Anda tentukan untuk mengirim notifikasi email, buat topik Amazon SNS menggunakan editor kueri v2 dengan menavigasi ke bagian pemberitahuan SNS, Aktifkan pemantauan, dan buat topik dengan topik Buat SNS. Editor kueri v2 membuat topik Amazon SNS dan menambahkan prinsip layanan ke kebijakan akses untuk Amazon. EventBridge Berikut ini adalah contoh Kebijakan akses yang dibuat dalam topik Amazon SNS. Dalam contoh, topik AWS Region *us-west-2*, Akun AWS *123456789012*, dan Amazon SNS *select-version-pdx-testunload* digunakan.

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"
  }
]
}

```

Saat kueri terjadwal berjalan, Amazon SNS mengirimkan email AWS notifikasi. Contoh berikut menunjukkan email yang dikirim *myemail@example.com* untuk kueri terjadwal *QS2-may25a* yang berjalan Akun AWS *123456789012* menggunakan AWS Region *eu-north-1* topik notifikasi Amazon SNS. *may25a-SNS*

```

{"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>

Pemecahan masalah pengaturan penjadwalan kueri

Pertimbangkan hal berikut jika Anda memiliki masalah dalam menjadwalkan kueri.

Kueri tidak berjalan

Periksa apakah peran IAM yang digunakan dalam jadwal memiliki izin untuk mendapatkan kredensial cluster sementara. Izin untuk cluster yang disediakan adalah. `redshift:GetClusterCredentialsWithIAM` Izin untuk grup kerja Redshift Serverless adalah. `redshift-serverless:GetCredentials`

Riwayat terjadwal tidak ditampilkan

Pengguna IAM atau peran IAM yang digunakan untuk masuk ke AWS konsol tidak ditambahkan ke kebijakan kepercayaan peran IAM yang digunakan untuk menjadwalkan kueri.

Saat menggunakan AWS Secrets Manager kueri terjadwal untuk terhubung, konfirmasi bahwa rahasia ditandai dengan kunci `RedshiftDataFullAccess`.

Jika kueri terjadwal menggunakan AWS Secrets Manager koneksi, peran IAM yang digunakan untuk menjadwalkan kueri harus memiliki kebijakan `SecretsManagerReadWrite` terkelola yang setara dengan peran tersebut.

Status riwayat kueri adalah **Failed**

Lihat tampilan sistem `SYS_QUERY_HISTORY` untuk detail tentang mengapa kueri gagal. Masalah umum adalah bahwa pengguna database atau peran yang digunakan untuk menjalankan kueri mungkin tidak memiliki hak istimewa yang diperlukan untuk menjalankan SQL. Untuk informasi selengkapnya, lihat [Mengautentikasi kueri terjadwal](#).

SQL berikut menanyakan tampilan `SYS_QUERY_HISTORY` untuk mengembalikan kueri yang gagal.

```
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';
```

Untuk mengetahui detail kueri terjadwal tertentu yang gagal, lihat [Melihat hasil kueri terjadwal dengan AWS CloudShell](#).

Melihat hasil kueri terjadwal dengan AWS CloudShell

Anda dapat menggunakan AWS CloudShell untuk mengetahui detail tentang kueri terjadwal. Anda harus memiliki izin yang tepat untuk menjalankan AWS CLI perintah yang ditunjukkan dalam prosedur berikut.

Untuk melihat hasil kueri terjadwal

1. Di AWS konsol, buka prompt AWS CloudShell perintah. Untuk informasi selengkapnya AWS CloudShell, lihat [Apa yang ada AWS CloudShell](#) di Panduan AWS CloudShell Pengguna.

2. Asumsikan peran IAM dari kueri terjadwal. Untuk mengambil peran, temukan peran IAM yang terkait dengan kueri terjadwal di editor kueri v2 dan gunakan dalam AWS CLI perintah di AWS CloudShell. Misalnya, untuk peran `scheduler` masukkan AWS STS perintah untuk mengambil peran yang digunakan oleh kueri terjadwal.

```
aws sts assume-role --role-arn "arn:aws:iam::123456789012:role/scheduler" --role-session-name "scheduler-test"
```

Kredensi yang dikembalikan mirip dengan yang berikut ini.

```
"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. Buat variabel lingkungan dalam AWS CLI menggunakan kredensial yang ditampilkan dari asumsi peran IAM. Anda harus menggunakan token ini sebelum waktu kedaluwarsa. Misalnya, Anda memasukkan yang berikut ini AWS CloudShell.

```
export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE  
export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY  
export AWS_SESSION_TOKEN=je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY...
```

4. Untuk melihat kesalahan kueri yang gagal, jalankan AWS CLI perintah untuk mendeskripsikan pernyataan. Id pernyataan SQL berasal dari ID yang ditampilkan di bagian riwayat Jadwal dari kueri terjadwal di editor kueri v2.

```
aws redshift-data describe-statement --id 130d2620-05d2-439c-b7cf-815d9767f513
```

Dalam contoh ini, SQL terjadwal `select * from users limit 100` menghasilkan kesalahan SQL bahwa `users` tabel tidak ada.

```
{
  "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"
}
```

Memvisualisasikan hasil kueri

Setelah Anda menjalankan kueri dan tampilan hasil, Anda dapat mengaktifkan Bagan untuk menampilkan visualisasi grafis dari halaman hasil saat ini. Anda dapat menggunakan kontrol berikut untuk menentukan konten, struktur, dan tampilan bagan Anda:



Jejak

Merupakan satu set tanda grafis terkait dalam bagan. Anda dapat menentukan beberapa jejak dalam bagan.

Tipe

Anda dapat menentukan jenis jejak untuk mewakili data sebagai salah satu dari berikut ini:

- Bagan sebar untuk plot pencar atau bagan gelembung.
- Bagan batang untuk mewakili kategori data dengan batang vertikal atau horizontal.
- Bagan area untuk menentukan area yang diisi.
- Histogram yang menggunakan bar untuk mewakili distribusi frekuensi.
- Diagram lingkaran untuk representasi melingkar data di mana setiap irisan mewakili persentase dari keseluruhan.
- Bagan Corong atau Area Corong untuk mewakili data melalui berbagai tahap proses.

- Grafik OHLC (open-high-low-close) sering digunakan untuk data keuangan untuk mewakili nilai terbuka, tinggi, rendah, dan dekat sepanjang sumbu x, yang biasanya mewakili interval waktu.
- Bagan kandil untuk mewakili rentang nilai untuk kategori di atas garis waktu.
- Bagan air terjun untuk mewakili bagaimana nilai awal meningkat atau menurun melalui serangkaian nilai menengah. Nilai dapat mewakili interval waktu atau kategori.
- Bagan garis untuk mewakili perubahan nilai dari waktu ke waktu.

Sumbu X

Anda menentukan kolom tabel yang berisi nilai-nilai untuk plot sepanjang sumbu X. Kolom yang berisi nilai deskriptif biasanya mewakili data dimensi. Kolom yang berisi nilai kuantitatif biasanya mewakili data faktual.

Sumbu Y

Anda menentukan kolom tabel yang berisi nilai-nilai untuk plot sepanjang sumbu Y. Kolom yang berisi nilai deskriptif biasanya mewakili data dimensi. Kolom yang berisi nilai kuantitatif biasanya mewakili data faktual.

Subplot

Anda dapat menentukan presentasi tambahan dari data bagan.

Mengubah

Anda dapat menentukan transformasi untuk memfilter data jejak. Anda menggunakan transformasi terpisah untuk menampilkan beberapa jejak dari satu jejak sumber. Anda menggunakan transformasi agregat untuk menyajikan jejak sebagai rata-rata atau minimum. Anda menggunakan transformasi semacam untuk mengurutkan jejak.

Penampilan umum

Anda dapat mengatur default untuk warna latar belakang, warna margin, skala warna untuk merancang palet, gaya dan ukuran teks, gaya dan ukuran judul, dan bilah mode. Anda dapat menentukan interaksi untuk seret, klik, dan arahkan kursor. Anda dapat menentukan teks meta. Anda dapat menentukan tampilan default untuk jejak, sumbu, legenda, dan anotasi.

Untuk membuat bagan

1. Jalankan kueri dan dapatkan hasil.
2. Aktifkan Bagan.

3. Pilih Trace dan mulai memvisualisasikan data Anda.
4. Pilih gaya bagan dari salah satu dari berikut ini:
 - Scatter
 - Bar
 - Bidang
 - Histogram
 - Pie
 - Corong
 - Area Corong
 - OHLC () open-high-low-close
 - Candlestick
 - Air Terjun
 - Baris
5. Pilih Gaya untuk menyesuaikan tampilan, termasuk warna, sumbu, legenda, dan anotasi. Anda dapat menambahkan teks, bentuk, dan gambar.
6. Pilih Anotasi untuk menambahkan teks, bentuk, dan gambar.
7. Pilih Segarkan untuk memperbarui tampilan bagan. Pilih Layar penuh untuk memperluas tampilan bagan.

Contoh: Buat diagram lingkaran untuk memvisualisasikan hasil kueri

Contoh berikut menggunakan tabel Penjualan dari database sampel. Untuk informasi selengkapnya, lihat [Contoh database](#) di Panduan Pengembang Database Amazon Redshift.

Berikut ini adalah query yang Anda jalankan untuk memberikan data untuk pie chart.

```
select top 5 eventname, count(salesid) totalorders, sum(pricepaid) totalsales
from sales, event
where sales.eventid=event.eventid group by eventname
order by 3;
```

Untuk membuat diagram lingkaran untuk acara teratas dengan total penjualan

1. Jalankan kueri.

2. Di area hasil kueri, nyalakan Bagan.
3. Pilih Jejak.
4. Untuk Jenis, pilih Pie.
5. Untuk Nilai, pilih totalsales.
6. Untuk Label, pilih eventname.
7. Pilih Style dan kemudian General.
8. Di bawah Colorscales, pilih Categorical dan kemudian Pastel2.



Contoh: Buat bagan kombinasi untuk membandingkan pendapatan dan penjualan

Lakukan langkah-langkah dalam contoh ini untuk membuat bagan yang menggabungkan diagram batang untuk data pendapatan dan grafik garis untuk data penjualan. Contoh berikut menggunakan tabel Penjualan database sampel tickit. Untuk informasi selengkapnya, lihat [Contoh database](#) di Panduan Pengembang Database Amazon Redshift.

Berikut ini adalah query yang Anda jalankan untuk memberikan data untuk grafik.

```
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;
```

Untuk membuat bagan kombinasi untuk membandingkan pendapatan dan penjualan

1. Jalankan kueri.
2. Di area hasil kueri, nyalakan Bagan.
3. Di bawah jejak o, untuk Jenis, pilih Bar.
4. Untuk X, pilih eventname.
5. Untuk Y, pilih total_price.

Bagan batang ditampilkan dengan nama acara di sepanjang sumbu X.

6. Di bawah Gaya, pilih Jejak.
7. Untuk Nama, masukkan Pendapatan.
8. Di bawah Style, pilih Axes.
9. Untuk Judul, pilih Y dan masukkan Pendapatan.

Label Pendapatan ditampilkan pada sumbu Y kiri.

10. Di bawah Struktur, pilih Jejak.
11. Pilih



Jejak.

Tampilan opsi jejak 1.

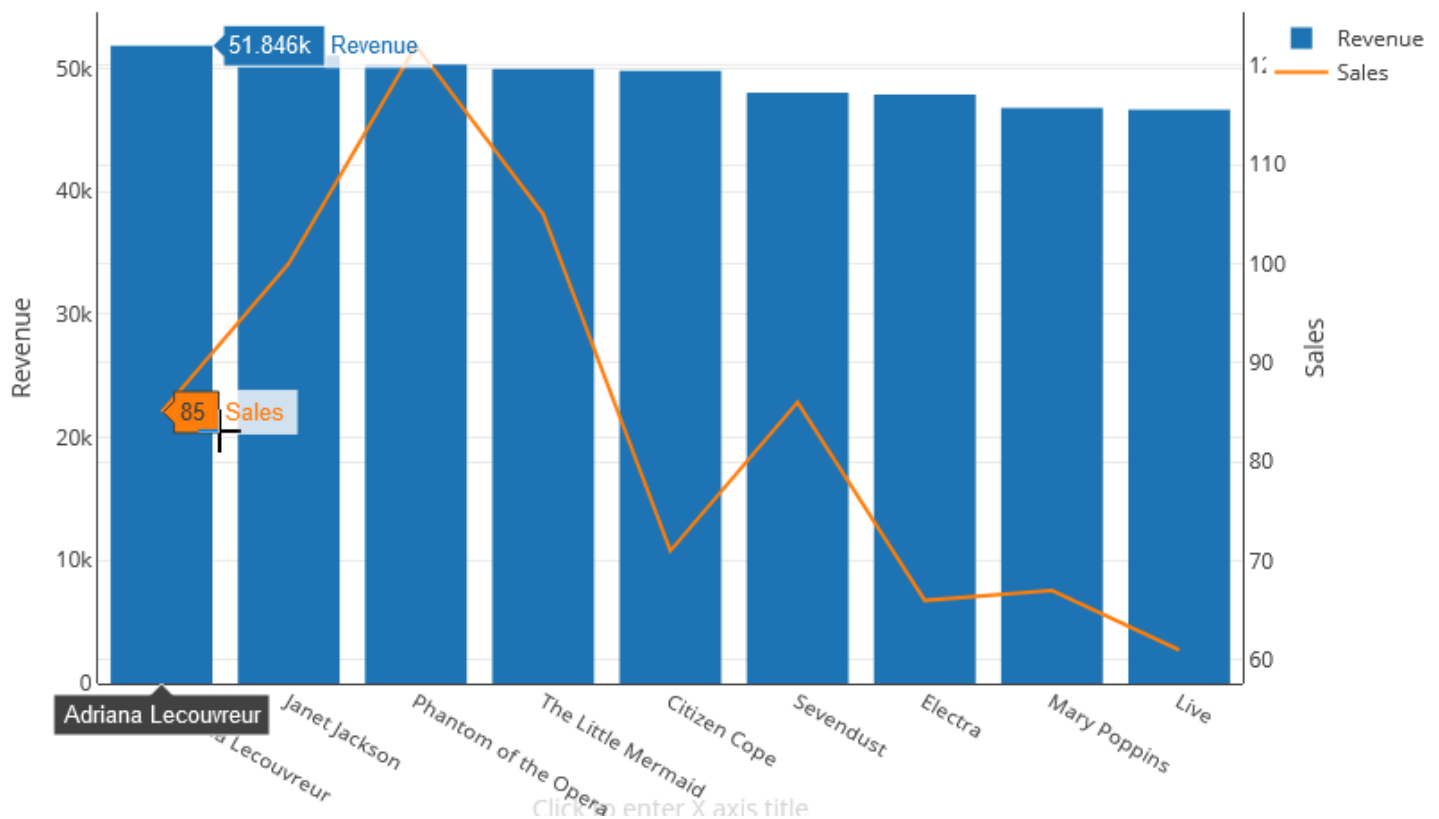
12. Untuk Type, pilih Line.
13. Untuk X, pilih eventname.
14. Untuk Y, pilih total_qty_sold.
15. Di bawah Sumbu Untuk Digunakan, untuk Sumbu Y

pilih

Sumbu Y menampilkan Y2.

16. Di bawah Style, pilih Axes.
17. Di bawah Judul, pilih Y2.
18. Untuk Nama, masukkan Penjualan.
19. Di bawah Garis, pilih Y: Penjualan.
20. Di bawah Garis Sumbu, pilih Tampilkan dan untuk Posisi, pilih Kanan.

Revenue and Sales



Demo: Bangun visualisasi menggunakan editor kueri Amazon Redshift v2

Untuk demo tentang cara membangun visualisasi, tonton video berikut. [Buat visualisasi menggunakan editor kueri Amazon Redshift v2.](#)

Berkolaborasi dan berbagi sebagai sebuah tim

Anda dapat berbagi pertanyaan dengan tim Anda.

Tim didefinisikan untuk sekumpulan pengguna yang berkolaborasi dan berbagi sumber daya editor kueri v2. Administrator dapat membuat tim dengan menambahkan tag ke peran IAM. Untuk informasi selengkapnya, lihat [izin yang diperlukan untuk menggunakan editor kueri v2.](#)

Menyimpan dan menjelajah untuk kueri

Sebelum Anda dapat membagikan kueri Anda dengan tim Anda, simpan kueri Anda. Anda dapat melihat dan menghapus kueri yang disimpan.

Untuk menyimpan kueri

1. Siapkan kueri Anda dan pilih Simpan.
2. Masukkan judul untuk kueri Anda.
3. Pilih Simpan.

Untuk menelusuri kueri yang disimpan

1. Pilih Kueri dari panel navigasi.
2. Anda dapat melihat kueri yang merupakan kueri Saya, Dibagikan oleh saya, atau Dibagikan ke tim saya. Kueri ini dapat muncul sebagai kueri individual atau dalam folder yang Anda buat.

Berbagi kueri

Anda dapat membagikan pertanyaan Anda dengan tim Anda. Anda juga dapat melihat riwayat kueri yang disimpan dan mengelola versi kueri.

Untuk berbagi kueri dengan tim Anda, pastikan bahwa Anda memiliki tag utama yang `sqlworkbench-team` disetel ke nilai yang sama dengan anggota tim lainnya di akun Anda. Misalnya, administrator mungkin menetapkan nilai `accounting-team` untuk semua orang di departemen akuntansi. Sebagai contoh, lihat [Izin yang diperlukan untuk menggunakan editor kueri v2](#).

Untuk berbagi kueri dengan tim

1. Pilih Kueri dari panel navigasi.
2. Buka menu konteks (klik kanan) kueri yang ingin Anda bagikan dan pilih Bagikan dengan tim saya.
3. Pilih tim atau tim yang ingin Anda bagikan kueri, lalu pilih opsi Simpan berbagi.

Mengelola versi kueri

Setiap kali Anda menyimpan kueri SQL, editor kueri v2 menyimpannya sebagai versi baru. Anda dapat menelusuri versi kueri sebelumnya, menyimpan salinan kueri, atau memulihkan kueri.

Untuk mengelola versi kueri

1. Pilih Kueri dari panel navigasi.

2. Buka menu konteks (klik kanan) untuk kueri yang ingin Anda kerjakan.
3. Pilih Riwayat versi untuk membuka daftar versi kueri.
4. Pada halaman Riwayat versi, Anda dapat melakukan hal berikut:
 - Kembalikan ke yang dipilih - Kembalikan ke versi yang dipilih dan lanjutkan pekerjaan Anda dengan versi ini.
 - Simpan dipilih sebagai — Buat kueri baru di editor.

Menanyakan database menggunakan editor kueri Amazon Redshift v1

Menggunakan editor kueri adalah cara mudah untuk menjalankan kueri pada database yang dihosting oleh cluster Amazon Redshift Anda. Setelah membuat klaster, Anda dapat langsung menjalankan kueri dengan menggunakan editor kueri di konsol Amazon Redshift.

Note

Anda tidak dapat melakukan kueri data di Amazon Redshift Tanpa Server menggunakan editor kueri asli ini. Gunakan editor kueri Amazon Redshift v2 sebagai gantinya.

Pada Februari 2021, editor kueri yang diperbarui telah diterapkan dan izin otorisasi untuk menggunakan editor kueri diubah. Editor kueri baru menggunakan Amazon Redshift Data API untuk menjalankan kueri. `AmazonRedshiftQueryEditorKebijakan`, yang merupakan kebijakan AWS terkelola AWS Identity and Access Management (IAM), telah diperbarui untuk menyertakan izin yang diperlukan. Jika Anda memiliki kebijakan IAM khusus, pastikan Anda memperbaruinya. Gunakan `AmazonRedshiftQueryEditor` sebagai panduan. Perubahan untuk `AmazonRedshiftQueryEditor` mencakup yang berikut:

- Izin untuk mengelola hasil pernyataan editor kueri memerlukan pengguna pemilik pernyataan.
- Izin untuk menggunakan Secrets Manager untuk terhubung ke database telah ditambahkan.

Untuk informasi selengkapnya, lihat [Izin diperlukan untuk menggunakan editor kueri konsol Amazon Redshift](#).

Ketika Anda terhubung ke cluster Anda dari editor kueri baru, Anda dapat menggunakan salah satu dari dua metode otentikasi.

Menggunakan editor kueri, Anda dapat melakukan hal berikut:

- Jalankan kueri pernyataan SQL tunggal.
- Hasil unduhan menetapkan sebesar 100 MB ke file nilai dipisahkan koma (CSV).
- Simpan kueri untuk digunakan kembali. Anda tidak dapat menyimpan kueri di Wilayah Eropa (Paris), Wilayah Asia Pasifik (Osaka), Wilayah Asia Pasifik (Hong Kong), atau Wilayah Timur Tengah (Bahrain).
- Lihat detail runtime kueri untuk tabel yang ditentukan pengguna.
- Jadwalkan kueri untuk dijalankan di masa depan.
- Lihat riwayat kueri yang Anda buat di editor kueri.
- Jalankan kueri terhadap cluster menggunakan perutean VPC yang disempurnakan.

Pertimbangan editor kueri

Pertimbangkan hal berikut tentang bekerja dengan kueri saat Anda menggunakan editor kueri:

- Durasi maksimum kueri adalah 24 jam.
- Ukuran hasil kueri maksimum adalah 100 MB. Jika panggilan mengembalikan lebih dari 100 MB data respons, panggilan akan dihentikan.
- Waktu retensi maksimum untuk hasil kueri adalah 24 jam.
- Ukuran pernyataan kueri maksimum adalah 100 KB.
- Cluster harus berada di cloud pribadi virtual (VPC) berdasarkan layanan Amazon VPC.
- Anda tidak dapat menggunakan transaksi di editor kueri. Untuk informasi selengkapnya tentang transaksi, lihat [MULAI](#) di Panduan Pengembang Database Amazon Redshift.
- Anda dapat menyimpan kueri hingga 3.000 karakter.

Menyambung ke gudang data Amazon Redshift menggunakan alat klien SQL

Anda dapat terhubung ke gudang data Amazon Redshift dari alat klien SQL melalui koneksi Java Database Connectivity (JDBC), Python, dan Open Database Connectivity (ODBC). Amazon Redshift

tidak menyediakan atau menginstal alat atau pustaka klien SQL apa pun. Untuk menggunakan alat atau pustaka ini untuk bekerja dengan data di gudang data Anda, instal di komputer klien Anda atau instans Amazon EC2. Anda dapat menggunakan sebagian besar alat klien SQL yang mendukung driver JDBC, Python, atau ODBC.

Gunakan daftar bagian di akhir topik ini untuk membantu Anda menelusuri proses mengonfigurasi komputer klien atau instans Amazon EC2 untuk menggunakan koneksi JDBC, Python, atau ODBC. Topik juga membahas opsi keamanan terkait untuk koneksi klien ke server. Selain itu, temukan informasi tentang pengaturan dan koneksi dari alat klien SQL, seperti [Amazon Redshift](#) RSQL. Anda dapat mencoba alat-alat ini jika Anda belum memiliki alat intelijen bisnis untuk digunakan. Anda juga dapat menggunakan bagian ini untuk mempelajari tentang menghubungkan ke data Anda. Terakhir, jika Anda mengalami masalah saat mencoba terhubung ke gudang data Anda, Anda dapat meninjau informasi pemecahan masalah untuk mengidentifikasi solusi.

Rekomendasi untuk menghubungkan dengan alat klien

Jika Anda terhubung ke kluster Redshift Anda menggunakan alamat IP, itu dapat mengakibatkan downtime tambahan ketika ada pemadaman atau kehilangan koneksi dan cluster dibawa online di Availability Zone (AZ) baru. Namun, jika Anda masih ingin aplikasi Anda terhubung ke Redshift menggunakan alamat IP, gunakan alamat IP pribadi yang dilampirkan ke titik akhir cluster (virtual-private-cloudVPC). Anda dapat menemukan ini di detail cluster di Jaringan dan keamanan, di bawah tab Properties.

Note

Jika aplikasi Anda menggunakan alamat IP node pemimpin untuk mengakses kluster Redshift, praktik terbaik yang disarankan adalah mengubahnya untuk menggunakan URL titik akhir cluster. Untuk informasi selengkapnya, lihat [Mengonfigurasi koneksi di Amazon Redshift](#).

Topik

- [Mengonfigurasi koneksi di Amazon Redshift](#)
- [Mengkonfigurasi opsi keamanan untuk koneksi](#)
- [Menghubungkan dari alat dan kode klien](#)
- [Menggunakan profil otentikasi untuk terhubung ke Amazon Redshift](#)
- [Memecahkan masalah koneksi di Amazon Redshift](#)

Mengonfigurasi koneksi di Amazon Redshift

Di bagian berikut, pelajari cara mengkonfigurasi koneksi JDBC, Python, dan ODBC untuk terhubung ke cluster Anda dari alat klien SQL. Bagian ini menjelaskan cara mengatur koneksi JDBC, Python, dan ODBC. Ini juga menjelaskan cara menggunakan Secure Sockets Layer (SSL) dan sertifikat server untuk mengenkripsi komunikasi antara klien dan server.

Driver JDBC, Python, dan ODBC untuk Amazon Redshift

Untuk bekerja dengan data di cluster Anda, Anda harus memiliki driver JDBC, Python, atau ODBC untuk konektivitas dari komputer klien atau instance Anda. Kode aplikasi Anda untuk menggunakan operasi API akses data JDBC, Python, atau ODBC, dan gunakan alat klien SQL yang mendukung JDBC, Python, atau ODBC.

Amazon Redshift menawarkan driver JDBC, Python, dan ODBC untuk diunduh. Driver ini didukung oleh Dukungan. Driver PostgreSQL tidak diuji dan tidak didukung oleh tim Amazon Redshift. Gunakan driver khusus Amazon Redshift saat menyambungkan ke klaster Amazon Redshift. Driver Amazon Redshift memiliki keuntungan sebagai berikut:

- Support untuk IAM, SSO, dan otentikasi federasi.
- Support untuk tipe data Amazon Redshift baru.
- Support untuk profil otentikasi.
- Peningkatan kinerja dalam hubungannya dengan peningkatan Amazon Redshift.

Untuk informasi selengkapnya tentang cara mengunduh driver JDBC dan ODBC dan mengonfigurasi koneksi ke cluster Anda, lihat [Mengonfigurasi koneksi untuk driver JDBC versi 2.x untuk Amazon Redshift](#), [Konektor Python Amazon Redshift](#) dan [Mengonfigurasi koneksi untuk driver ODBC versi 2.x untuk Amazon Redshift](#)

Untuk informasi selengkapnya tentang mengelola identitas IAM, termasuk praktik terbaik untuk peran IAM, lihat [Manajemen identitas dan akses di Amazon Redshift](#)

Menemukan string koneksi cluster Anda

Untuk terhubung ke cluster Anda dengan alat klien SQL Anda, Anda harus memiliki string koneksi cluster. Anda dapat menemukan string koneksi cluster di konsol Amazon Redshift, di halaman detail cluster.

Untuk menemukan string koneksi untuk sebuah cluster

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster, lalu pilih nama cluster dari daftar untuk membuka detailnya.
3. String koneksi URL JDBC dan URL ODBC tersedia, bersama dengan detail tambahan, di bagian Informasi umum. Setiap string didasarkan pada AWS Wilayah tempat cluster berjalan. Klik ikon di sebelah string koneksi yang sesuai untuk menyalinnya.

Untuk menyambung ke titik akhir cluster, Anda dapat menggunakan URL titik akhir cluster dari permintaan [DescribeClusters API](#). Berikut ini adalah contoh URL endpoint cluster.

```
mycluster.cmeaswqeuae.us-east-2.redshift.amazonaws.com
```

Jika Anda telah menyiapkan nama domain khusus untuk klaster Anda, Anda juga dapat menggunakannya untuk terhubung ke klaster Anda. Untuk informasi selengkapnya tentang membuat nama domain kustom, lihat [Menyiapkan nama domain kustom](#).

Note

Saat Anda terhubung, jangan gunakan alamat IP node cluster atau alamat IP titik akhir VPC. Selalu gunakan titik akhir Redshift untuk menghindari pemadaman yang tidak perlu. Satu-satunya pengecualian untuk menggunakan URL endpoint adalah ketika Anda menggunakan nama domain khusus. Untuk informasi selengkapnya, lihat [Menggunakan nama domain khusus untuk koneksi klien](#).

Mengonfigurasi koneksi untuk driver JDBC versi 2.x untuk Amazon Redshift

Anda dapat menggunakan koneksi driver JDBC versi 2.x untuk terhubung ke cluster Amazon Redshift Anda dari banyak alat klien SQL pihak ketiga. Konektor Amazon Redshift JDBC menyediakan solusi open source. Anda dapat menelusuri kode sumber, meminta penyempurnaan, melaporkan masalah, dan memberikan kontribusi.

Untuk informasi terbaru tentang perubahan driver JDBC, lihat log [perubahan](#).

Secara default, driver Amazon Redshift JDBC dikonfigurasi untuk menggunakan keepalives TCP untuk mencegah koneksi dari kehabisan waktu. Anda dapat menentukan kapan driver mulai

mengirim paket keepalive atau mematikan fitur dengan mengatur properti yang relevan di URL koneksi. Untuk informasi selengkapnya tentang sintaks URL koneksi, lihat [Membangun URL koneksi](#).

Properti	Deskripsi
TCPKeepAlive	Untuk mematikan keepalives TCP, setelah properti ini ke. FALSE

Topik

- [Unduh driver Amazon Redshift JDBC, versi 2.1](#)
- [Menginstal driver Amazon Redshift JDBC, versi 2.2](#)
- [Mendapatkan URL JDBC](#)
- [Membangun URL koneksi](#)
- [Mengkonfigurasi koneksi JDBC dengan Apache Maven](#)
- [Mengkonfigurasi otentikasi dan SSL](#)
- [Mengonfigurasi pencatatan log](#)
- [Konversi jenis data](#)
- [Menggunakan dukungan pernyataan yang disiapkan](#)
- [Perbedaan antara versi 2.2 dan 1.x dari driver JDBC](#)
- [Membuat file inisialisasi \(.ini\) untuk driver JDBC versi 2.x](#)
- [Opsi untuk konfigurasi driver JDBC versi 2.x](#)
- [Versi sebelumnya dari JDBC driver version 2.x](#)

Unduh driver Amazon Redshift JDBC, versi 2.1

Note

Driver Amazon Redshift JDBC 2.x tidak dirancang agar aman untuk thread-safe. Dua atau lebih thread secara bersamaan mencoba menggunakan koneksi yang sama dapat menyebabkan kebuntuan, kesalahan, hasil yang salah, atau perilaku tak terduga lainnya. Jika Anda memiliki aplikasi multi-threaded, kami sarankan Anda menyinkronkan akses ke driver untuk menghindari akses bersamaan.

Amazon Redshift menawarkan driver untuk alat yang kompatibel dengan JDBC 4.2 API. Nama kelas untuk pengemudi ini adalah `com.amazon.redshift.Driver`.

Untuk informasi rinci tentang cara menginstal driver JDBC, referensi perpustakaan driver JDBC, dan mendaftarkan kelas driver, lihat topik berikut.

Untuk setiap komputer tempat Anda menggunakan driver Amazon Redshift JDBC versi 2.x, pastikan Java Runtime Environment (JRE) 8.0 diinstal.

Jika Anda menggunakan driver Amazon Redshift JDBC untuk otentikasi database, pastikan Anda memiliki AWS SDK for Java 1.11.118 atau yang lebih baru di jalur kelas Java Anda. Jika Anda belum AWS SDK for Java menginstal, unduh file ZIP dengan pustaka yang bergantung pada driver dan driver yang kompatibel dengan JDBC 4.2 untuk SDK: AWS

- [JDBC 4.2—driver yang kompatibel dengan versi 2.x dan pustaka yang bergantung pada driver AWS SDK Di Wilayah Tiongkok \(Beijing\), gunakan tautan berikut: JDBC 4.2 — driver yang kompatibel dengan versi 2.x dan pustaka driver SDK AWS](#)

File ZIP ini berisi driver yang kompatibel dengan JDBC 4.2 versi 2.x dan SDK for AWS Java 1.x file pustaka yang bergantung pada driver. Buka zip file jar dependen ke lokasi yang sama dengan driver JDBC. Hanya driver JDBC yang perlu berada di CLASSPATH.

File ZIP ini tidak menyertakan AWS SDK for Java 1.x lengkap. Namun, ini mencakup pustaka yang bergantung pada driver AWS SDK for Java 1.x yang diperlukan AWS Identity and Access Management untuk otentikasi database (IAM).

Gunakan driver Amazon Redshift JDBC ini dengan AWS SDK yang diperlukan untuk autentikasi database IAM.

Untuk menginstal AWS SDK for Java 1.x lengkap, [AWS lihat SDK for Java](#) AWS SDK for Java 1.x di Panduan Pengembang.

- [Driver yang kompatibel dengan JDBC 4.2 versi 2.x \(tanpa SDK AWS \) Di Wilayah Tiongkok \(Beijing\), gunakan tautan berikut: JDBC 4.2—driver versi 2.x \(tanpa SDK\) AWS](#)

Tinjau lisensi perangkat lunak driver JDBC versi 2.x dan ubah file log:

- [Lisensi driver JDBC versi 2.x](#)
- [Driver JDBC versi 2.x perubahan log](#)

Driver JDBC versi 1.2.27.1051 dan yang lebih baru mendukung prosedur tersimpan Amazon Redshift. Untuk informasi selengkapnya, lihat [Membuat prosedur tersimpan di Amazon Redshift](#) di Panduan Pengembang Database Amazon Redshift.

Menginstal driver Amazon Redshift JDBC, versi 2.2

Untuk menginstal Amazon Redshift JDBC 4.2—driver versi 2.x yang kompatibel dengan driver dan library yang bergantung pada driver untuk AWS SDK, ekstrak file dari arsip ZIP ke direktori pilihan Anda.

Untuk menginstal driver yang kompatibel dengan Amazon Redshift JDBC 4.2 versi 2.x (tanpa AWS SDK), salin file JAR ke direktori pilihan Anda.

Untuk mengakses penyimpanan data Amazon Redshift menggunakan driver Amazon Redshift JDBC, Anda perlu melakukan konfigurasi seperti yang dijelaskan berikut.

Topik

- [Merujuk pustaka driver JDBC](#)
- [Mendaftarkan kelas pengemudi](#)

Merujuk pustaka driver JDBC

Aplikasi JDBC atau kode Java yang Anda gunakan untuk terhubung ke data Anda harus mengakses file JAR driver. Dalam aplikasi atau kode, tentukan semua file JAR yang Anda ekstrak dari arsip ZIP.

Menggunakan driver dalam aplikasi JDBC

Aplikasi JDBC biasanya menyediakan serangkaian opsi konfigurasi untuk menambahkan daftar file pustaka driver. Gunakan opsi yang disediakan untuk menyertakan semua file JAR dari arsip ZIP sebagai bagian dari konfigurasi driver dalam aplikasi. Untuk informasi selengkapnya, lihat dokumentasi untuk aplikasi JDBC Anda.

Menggunakan driver dalam kode Java

Anda harus menyertakan semua file pustaka driver di jalur kelas. Ini adalah jalur yang dicari Java Runtime Environment untuk kelas dan file sumber daya lainnya. Untuk informasi selengkapnya, lihat dokumentasi Java SE yang sesuai untuk mengatur jalur kelas untuk sistem operasi Anda.

- Windows: <https://docs.oracle.com/javase/7/.html/docs/technotes/tools/windows/classpath>
- Linux dan Solaris: <https://docs.oracle.com/javase/docs/technotes/tools/solaris/classpath7/.html>

- macOS: Jalur kelas macOS default adalah direktori tempat driver JDBC diinstal.

Mendaftarkan kelas pengemudi

Pastikan Anda mendaftarkan kelas yang sesuai untuk aplikasi Anda. Anda menggunakan kelas berikut untuk menghubungkan driver Amazon Redshift JDBC ke penyimpanan data Amazon Redshift:

- `Driver` kelas diperluas `java.sql.Driver`.
- `DataSource` kelas diperluas `javax.sql.DataSource` dan `javax.sql.ConnectionPoolDataSource`.

Driver mendukung nama kelas berkualifikasi penuh berikut yang independen dari versi JDBC:

- `com.amazon.redshift.jdbc.Driver`
- `com.amazon.redshift.jdbc.DataSource`

Contoh berikut menunjukkan bagaimana menggunakan `DriverManager` kelas untuk membuat koneksi untuk JDBC 4.2.

```
private static Connection connectViaDM() throws Exception
{
    Connection connection = null;
    connection = DriverManager.getConnection(CONNECTION_URL);
    return connection;
}
```

Contoh berikut menunjukkan bagaimana menggunakan `DataSource` kelas untuk membuat koneksi.

```
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;
}
```

Mendapatkan URL JDBC

Sebelum Anda dapat terhubung ke cluster Amazon Redshift Anda dari alat klien SQL, Anda perlu mengetahui URL JDBC klaster Anda. URL JDBC memiliki format berikut:

`jdbc:redshift://endpoint:port/database`

Bidang format sebelumnya memiliki nilai berikut.

Bidang	Nilai
<code>jdbc</code>	Protokol untuk koneksi.
<code>redshift</code>	Subprotokol yang menentukan untuk menggunakan driver Amazon Redshift untuk terhubung ke database.
<code><i>endpoint</i></code>	Titik akhir dari cluster Amazon Redshift.
<code><i>port</i></code>	Nomor port yang Anda tentukan saat Anda meluncurkan cluster. Jika Anda memiliki firewall, pastikan port ini terbuka untuk Anda gunakan.
<code><i>database</i></code>	Database yang Anda buat untuk cluster Anda.

Berikut ini adalah contoh URL JDBC: `jdbc:redshift://examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com:5439/dev`

Jika nilai URL Anda berisi salah satu karakter yang dicadangkan URI berikut, nilainya harus dikodekan URL:

- ;
- +
- {
- }
- [

-]
- &
- =
- ?
- ruang kosong

Misalnya, jika PWD nilai Anda password:password, URL koneksi yang menggunakan nilai itu akan terlihat seperti berikut:

```
jdbc:redshift://redshift.company.us-west-1.redshift.amazonaws.com:9000/  
dev;UID=amazon;PWD=password%3Apassword
```

Untuk informasi tentang cara mendapatkan koneksi JDBC Anda, lihat [Menemukan string koneksi cluster Anda](#)

Jika komputer klien gagal terhubung ke database, Anda dapat memecahkan masalah yang mungkin terjadi. Untuk informasi selengkapnya, lihat [Memecahkan masalah koneksi di Amazon Redshift](#).

Membangun URL koneksi

Gunakan URL koneksi untuk memasok informasi koneksi ke penyimpanan data yang Anda akses. Berikut ini adalah format URL koneksi untuk driver Amazon Redshift JDBC versi 2.x. Di sini, [Host] titik akhir server Amazon Redshift dan [Port] adalah jumlah port Transmission Control Protocol (TCP) yang digunakan server untuk mendengarkan permintaan klien.

```
jdbc:redshift://[Host]:[Port]
```

Berikut ini adalah format URL koneksi yang menentukan beberapa pengaturan opsional.

```
jdbc:redshift://[Host]:[Port]/[database];[Property1]=[Value];  
[Property2]=[Value];
```

Jika nilai URL Anda berisi salah satu karakter yang dicadangkan URI berikut, nilainya harus dikodekan URL:

- ;
- +
- {

- }
- [
-]
- &
- =
- ?
- ruang kosong

Misalnya, jika PWD nilai Anda `password:password`, URL koneksi yang menggunakan nilai itu akan terlihat seperti berikut:

```
jdbc:redshift://redshift.company.us-west-1.redshift.amazonaws.com:9000/  
dev;UID=amazon;PWD=password%3Apassword
```

Misalnya, misalkan Anda ingin terhubung ke port 9000 pada cluster Amazon Redshift di Wilayah AS Barat (California Utara) aktif. AWS Anda juga ingin mengakses database bernama `dev` dan mengotentikasi koneksi menggunakan nama pengguna dan kata sandi database. Dalam hal ini, Anda menggunakan URL koneksi berikut.

```
jdbc:redshift://redshift.company.us-west-1.redshift.amazonaws.com:9000/  
dev;UID=amazon;PWD=amazon
```

Anda dapat menggunakan karakter berikut untuk memisahkan opsi konfigurasi dari sisa string URL:

- ;
- ?

Misalnya, string URL berikut setara:

```
jdbc:redshift://my_host:5439/dev;ssl=true;defaultRowFetchSize=100
```

```
jdbc:redshift://my_host:5439/dev?ssl=true;defaultRowFetchSize=100
```

Anda dapat menggunakan karakter berikut untuk memisahkan opsi konfigurasi satu sama lain dalam string URL:

- ;
- &

Misalnya, string URL berikut setara:

```
jdbc:redshift://my_host:5439/dev;ssl=true;defaultRowFetchSize=100
```

```
jdbc:redshift://my_host:5439/dev;ssl=true&defaultRowFetchSize=100
```

Contoh URL berikut menentukan tingkat log 6 dan jalur untuk log.

```
jdbc:redshift://redshift.amazonaws.com:5439/dev;DSILogLevel=6;LogPath=/home/user/logs;
```

Jangan duplikat properti di URL koneksi.

Untuk daftar lengkap opsi konfigurasi yang dapat Anda tentukan, lihat [Opsi untuk konfigurasi driver JDBC versi 2.x](#).

Note

Saat Anda terhubung, jangan gunakan alamat IP node cluster atau alamat IP titik akhir VPC. Selalu gunakan titik akhir Redshift untuk menghindari pemadaman yang tidak perlu. Satu-satunya pengecualian untuk menggunakan URL endpoint adalah ketika Anda menggunakan nama domain khusus. Untuk informasi selengkapnya, lihat [Menggunakan nama domain khusus untuk koneksi klien](#).

Mengkonfigurasi koneksi JDBC dengan Apache Maven

Apache Maven adalah perangkat lunak manajemen proyek dan alat pemahaman. AWS SDK for Java Mendukung proyek Apache Maven. Untuk informasi selengkapnya, lihat [Menggunakan SDK dengan Apache Maven](#) di Panduan Pengembang AWS SDK for Java

Jika Anda menggunakan Apache Maven, Anda dapat mengonfigurasi dan membangun proyek Anda untuk menggunakan driver Amazon Redshift JDBC untuk terhubung ke cluster Amazon Redshift Anda. Untuk melakukan ini, tambahkan driver JDBC sebagai dependensi dalam file proyek Anda. `pom.xml` Jika Anda menggunakan Maven untuk membangun proyek Anda dan ingin menggunakan koneksi JDBC, ambil langkah-langkah di bagian berikut.

Untuk mengkonfigurasi driver JDBC sebagai dependensi Maven

1. Tambahkan repositori Amazon atau repositori Maven Central ke bagian repositori file Anda. `pom.xml`

Note

URL dalam contoh kode berikut mengembalikan kesalahan jika digunakan dalam browser. Gunakan URL ini hanya dalam konteks proyek Maven.

Untuk terhubung menggunakan Secure Sockets Layer (SSL), tambahkan repositori berikut ke file Anda. `pom.xml`

```
<repositories>
  <repository>
    <id>redshift</id>
    <url>https://s3.amazonaws.com/redshift-maven-repository/release</url>
  </repository>
</repositories>
```

Untuk repositori Maven Central, tambahkan berikut ini ke file Anda. `pom.xml`

```
<repositories>
  <repository>
    <id>redshift</id>
    <url>https://repo1.maven.org/maven2</url>
  </repository>
</repositories>
```

2. Deklarasikan versi driver yang ingin Anda gunakan di bagian dependensi file Anda. `pom.xml`

Amazon Redshift menawarkan driver untuk alat yang kompatibel dengan JDBC 4.2 API. Untuk informasi tentang fungsionalitas yang didukung oleh driver ini, lihat [Unduh driver Amazon Redshift JDBC, versi 2.1](#).

Ganti *driver-version* dalam contoh berikut dengan versi driver Anda, misalnya `2.1.0.1`. Untuk driver yang kompatibel dengan JDBC 4.2, gunakan yang berikut ini.

```
<dependency>
```

```
<groupId>com.amazon.redshift</groupId>
<artifactId>redshift-jdbc42</artifactId>
<version>driver-version</version>
</dependency>
```

Nama kelas untuk pengemudi ini adalah `com.amazon.redshift.Driver`.

Driver Amazon Redshift Maven memerlukan dependensi opsional berikut saat Anda menggunakan otentikasi database IAM.

```
<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>
```

Untuk memutakhirkan atau mengubah driver Amazon Redshift JDBC ke versi terbaru, pertama-tama ubah bagian versi dependensi ke versi driver terbaru. Kemudian bersihkan proyek Anda dengan Plugin Maven Clean, seperti yang ditunjukkan berikut.

```
mvn clean
```

Mengkonfigurasi otentikasi dan SSL

Untuk melindungi data dari akses yang tidak sah, penyimpanan data Amazon Redshift memerlukan semua koneksi untuk diautentikasi menggunakan kredensial pengguna. Beberapa penyimpanan data juga memerlukan koneksi yang harus dibuat melalui protokol Secure Sockets Layer (SSL), baik dengan atau tanpa otentikasi satu arah.

Driver Amazon Redshift JDBC versi 2.x memberikan dukungan penuh untuk protokol otentikasi ini.

Versi SSL yang didukung driver tergantung pada versi JVM yang Anda gunakan. Untuk informasi tentang versi SSL yang didukung oleh setiap versi Java, lihat [Mendiagnosis TLS, SSL, dan HTTPS](#) di Blog Manajemen Produk Grup Platform Java.

Versi SSL yang digunakan untuk koneksi adalah versi tertinggi yang didukung oleh driver dan server, yang ditentukan pada waktu koneksi.

Konfigurasi driver Amazon Redshift JDBC versi 2.x untuk mengautentikasi koneksi Anda sesuai dengan persyaratan keamanan server Redshift yang Anda sambungkan.

Anda harus selalu memberikan nama pengguna dan kata sandi Redshift Anda untuk mengautentikasi koneksi. Bergantung pada apakah SSL diaktifkan dan diperlukan di server, Anda mungkin juga perlu mengonfigurasi driver untuk terhubung melalui SSL. Atau Anda mungkin menggunakan otentikasi SSL satu arah sehingga klien (driver itu sendiri) memverifikasi identitas server.

Anda memberikan informasi konfigurasi ke driver di URL koneksi. Untuk informasi selengkapnya tentang sintaks URL koneksi, lihat [Membangun URL koneksi](#).

SSL menunjukkan TLS/SSL, both Transport Layer Security and Secure Sockets Layer. The driver supports industry-standard versions of TLS/SSL.

Mengkonfigurasi otentikasi IAM

Jika Anda terhubung ke server Amazon Redshift menggunakan autentikasi IAM, setel properti berikut sebagai bagian dari string koneksi sumber data Anda.

Untuk informasi selengkapnya tentang autentikasi IAM, lihat. [Manajemen identitas dan akses di Amazon Redshift](#)

Untuk menggunakan otentikasi IAM, gunakan salah satu format string koneksi berikut:

String koneksi	Deskripsi
<code>jdbc:redshift:iam:// [host]:[port]/[db]</code>	String koneksi biasa. Pengemudi menyimpulkan clusterID dan Region dari host.
<code>jdbc:redshift:iam:// [cluster-id]: [region]/[db]</code>	Pengemudi mengambil informasi host, mengingat ClusterID dan Region.
<code>jdbc:redshift:iam:// [host]/[db]</code>	Driver default ke port 5439, dan menyimpulkan ClusterID dan Region dari host. Bergantung pada port yang Anda pilih saat membuat, memodifikasi, atau memigrasi klaster, izinkan akses ke port yang dipilih.

Menentukan profil

Jika Anda menggunakan otentikasi IAM, Anda dapat menentukan properti koneksi tambahan yang diperlukan atau opsional di bawah nama profil. Dengan melakukan ini, Anda dapat menghindari meletakkan informasi tertentu secara langsung di string koneksi. Anda menentukan nama profil dalam string koneksi Anda menggunakan properti Profil.

Profil dapat ditambahkan ke file AWS kredensial. Lokasi default untuk file ini adalah: `~/.aws/credentials`

Anda dapat mengubah nilai default dengan mengatur jalur dalam variabel lingkungan berikut: `AWS_CREDENTIAL_PROFILES_FILE`

Untuk informasi selengkapnya tentang profil, lihat [Bekerja dengan AWS Kredensial di. AWS SDK for Java](#)

Menggunakan kredensial profil instance

Jika Anda menjalankan aplikasi pada instans Amazon EC2 yang dikaitkan dengan peran IAM, Anda dapat terhubung menggunakan kredensial profil instans.

Untuk melakukan ini, gunakan salah satu format string koneksi IAM di tabel sebelumnya, dan atur properti koneksi dbuser ke nama pengguna Amazon Redshift yang Anda sambungkan sebagai.

Untuk informasi selengkapnya tentang profil instans, lihat [Manajemen Akses](#) di Panduan Pengguna IAM.

Menggunakan penyedia kredensi

Driver juga mendukung plugin penyedia kredensial dari layanan berikut:

- AWS Pusat Identitas IAM
- Layanan Federasi Direktori Aktif (ADFS)
- Layanan Token Web JSON (JWT)
- Layanan Microsoft Azure Active Directory (AD) dan Layanan Browser Microsoft Azure Active Directory (AD)
- Layanan Okta
- PingFederate Layanan
- Browser SAMP untuk layanan SAMP seperti Okta, Ping, atau ADFS

Jika Anda menggunakan salah satu layanan ini, URL koneksi perlu menentukan properti berikut:

- `Plugin_Name` - Jalur kelas yang sepenuhnya memenuhi syarat untuk kelas plugin penyedia kredensial Anda.
- `IDP_host`: — Host untuk layanan yang Anda gunakan untuk mengautentikasi ke Amazon Redshift.
- `IDP_port` — Port tempat host untuk layanan otentikasi mendengarkan. Tidak diperlukan untuk Okta.
- `User` — Nama pengguna untuk server `idp_host`.
- `Kata sandi` — Kata sandi yang terkait dengan nama pengguna `idp_host`.
- `DbUser` — Nama pengguna Amazon Redshift yang Anda sambungkan sebagai.
- `SSL_insecure` - Menunjukkan apakah sertifikat server IDP harus diverifikasi.
- `Client_ID` — ID klien yang terkait dengan nama pengguna di portal Azure AD. Hanya digunakan untuk Azure AD.
- `Client_Secret` — Rahasia klien yang terkait dengan ID klien di portal Azure AD. Hanya digunakan untuk Azure AD.
- `IDP_Tenant` - ID penyewa Azure AD untuk aplikasi Amazon Redshift Anda. Hanya digunakan untuk Azure AD.
- `App_ID` — ID aplikasi Okta untuk aplikasi Amazon Redshift Anda. Hanya digunakan untuk Okta.
- `App_Name` — Nama aplikasi Okta opsional untuk aplikasi Amazon Redshift Anda. Hanya digunakan untuk Okta.
- `Partner_SPID` — Nilai SPID mitra opsional (ID penyedia layanan). Hanya digunakan untuk PingFederate.

- `Idc_Region` — AWS Region Tempat instans AWS IAM Identity Center berada. Hanya digunakan untuk AWS IAM Identity Center.
- `Issuer_Url` — Titik akhir instance server AWS IAM Identity Center. Hanya digunakan untuk AWS IAM Identity Center.

Jika Anda menggunakan plugin browser untuk salah satu layanan ini, URL koneksi juga dapat mencakup:

- `Login_URL` — URL untuk sumber daya di situs web penyedia identitas saat menggunakan layanan Security Assertion Markup Language (SAMP) atau Azure AD melalui plugin browser. Parameter ini diperlukan jika Anda menggunakan plugin browser.
- `Listen_Port` — Port yang digunakan driver untuk mendapatkan respons SAMP dari penyedia identitas saat menggunakan layanan SAMP, Azure AD, atau AWS IAM Identity Center melalui plugin browser.
- `IDP_Response_Timeout` — Jumlah waktu, dalam hitungan detik, driver menunggu respons SAMP dari penyedia identitas saat menggunakan layanan SAMP, Azure AD, atau IAM Identity Center melalui plugin browser. AWS

Untuk informasi tentang properti string koneksi tambahan, lihat [Opsional untuk konfigurasi driver JDBC versi 2.x](#).

Hanya menggunakan nama pengguna dan kata sandi

Jika server yang Anda sambungkan tidak menggunakan SSL, maka Anda hanya perlu memberikan nama pengguna dan kata sandi Redshift Anda untuk mengautentikasi koneksi.

Untuk mengonfigurasi otentikasi menggunakan nama pengguna dan kata sandi Redshift Anda saja

1. Setel UID properti ke nama pengguna Redshift Anda untuk mengakses server Amazon Redshift.
2. Setel properti PWD ke kata sandi yang sesuai dengan nama pengguna Redshift Anda.

Menggunakan SSL tanpa verifikasi identitas

Jika server yang Anda sambungkan menggunakan SSL tetapi tidak memerlukan verifikasi identitas, maka Anda dapat mengonfigurasi driver untuk menggunakan pabrik SSL yang tidak memvalidasi.

Untuk mengonfigurasi koneksi SSL tanpa verifikasi identitas

1. Setel UID properti ke nama pengguna Redshift Anda untuk mengakses server Amazon Redshift.
2. Atur PWD properti ke kata sandi yang sesuai dengan nama pengguna Redshift Anda.
3. Atur SSLFactory properti ke `com.amazon.redshift.ssl.NonValidatingFactory`.

Menggunakan otentikasi SSL satu arah

Jika server yang Anda sambungkan menggunakan SSL dan memiliki sertifikat, maka Anda dapat mengonfigurasi driver untuk memverifikasi identitas server menggunakan otentikasi satu arah.

Otentikasi satu arah memerlukan sertifikat SSL yang ditandatangani dan tepercaya untuk memverifikasi identitas server. Anda dapat mengonfigurasi driver untuk menggunakan sertifikat tertentu atau mengakses TrustStore yang berisi sertifikat yang sesuai. Jika Anda tidak menentukan sertifikat atau TrustStore, maka driver menggunakan Java default TrustStore (biasanya salah satu `jssecacerts` atau `caacerts`).

Untuk mengonfigurasi otentikasi SSL satu arah

1. Setel properti UID ke nama pengguna Redshift Anda untuk mengakses server Amazon Redshift.
2. Setel properti PWD ke kata sandi yang sesuai dengan nama pengguna Redshift Anda.
3. Setel properti SSL ke `true`.
4. Setel properti SSLRoot Cert ke lokasi sertifikat CA root Anda.
5. Jika Anda tidak menggunakan salah satu Java default TrustStores, lakukan salah satu hal berikut:
 - Untuk menentukan sertifikat server, setel properti SSLRoot Cert ke jalur lengkap sertifikat.
 - Untuk menentukan a TrustStore, lakukan hal berikut:
 - a. Gunakan program `keytool` untuk menambahkan sertifikat server ke TrustStore yang ingin Anda gunakan.
 - b. Tentukan TrustStore dan kata sandi yang akan digunakan saat memulai aplikasi Java menggunakan driver. Contoh:

```
-Djavax.net.ssl.trustStore=[TrustStoreName]
-Djavax.net.ssl.trustStorePassword=[TrustStorePassword]
-Djavax.net.ssl.trustStoreType=[TrustStoreType]
```

6. Pilih salah satu:

- Untuk memvalidasi sertifikat, setel SSLMode properti ke verify-ca.
- Untuk memvalidasi sertifikat dan memverifikasi nama host dalam sertifikat, atur SSLMode properti ke verify-full.

Mengonfigurasi pencatatan log

Anda dapat mengaktifkan masuk ke driver untuk membantu mendiagnosis masalah.

Anda dapat mencatat informasi driver dengan menggunakan metode berikut:

- Untuk menyimpan informasi yang dicatat dalam file.log, lihat [Menggunakan file log](#).
- Untuk mengirim informasi yang dicatat ke LogStream atau LogWriter ditentukan dalam DriverManager, lihat [Menggunakan LogStream atau LogWriter](#).

Anda memberikan informasi konfigurasi ke driver di URL koneksi. Untuk informasi selengkapnya tentang sintaks URL koneksi, lihat [Membangun URL koneksi](#).

Menggunakan file log

Aktifkan logging cukup lama untuk menangkap masalah. Logging mengurangi kinerja dan dapat mengkonsumsi sejumlah besar ruang disk.

Atur LogLevel kunci di URL koneksi Anda untuk mengaktifkan logging dan tentukan jumlah detail yang disertakan dalam file log. Tabel berikut mencantumkan level logging yang disediakan oleh driver Amazon Redshift JDBC versi 2.x, dalam urutan dari yang paling sedikit bertele-tele hingga yang paling bertele-tele.

LogLevel nilai	Deskripsi
1	Log peristiwa kesalahan parah yang akan menyebabkan pengemudi membatalkan.
2	Peristiwa kesalahan log yang memungkinkan driver untuk terus berjalan.
3	Log peristiwa yang mungkin mengakibatkan kesalahan jika tindakan tidak diambil. Tingkat

LogLevel nilai	Deskripsi
	logging dan tingkat logging di atas level ini juga mencatat kueri pengguna.
4	Log informasi umum yang menjelaskan kemajuan pengemudi.
5	Log informasi terperinci yang berguna untuk men-debug driver.
6	Catat semua aktivitas pengemudi.

Untuk mengatur logging yang menggunakan file log

1. Atur LogLevel properti ke tingkat informasi yang diinginkan untuk disertakan dalam file log.
2. Atur LogPath properti ke jalur lengkap ke folder tempat Anda ingin menyimpan file log.

Misalnya, URL koneksi berikut memungkinkan logging level 3 dan menyimpan file log di folder `C:\temp: jdbc:redshift://redshift.company.us-west-1.redshift.amazonaws.com:9000/Default;DSILogLevel=3;LogPath=C:\temp`

3. Untuk memastikan bahwa pengaturan baru berlaku, restart aplikasi JDBC Anda dan sambungkan kembali ke server.

Driver Amazon Redshift JDBC menghasilkan file log berikut di lokasi yang ditentukan dalam properti: LogPath

- File `redshift_jdbc.log` yang mencatat aktivitas driver yang tidak spesifik untuk koneksi.
- `redshift_jdbc_connection_ [Number] .log` file untuk setiap koneksi yang dibuat ke database, di mana [Number] adalah nomor yang mengidentifikasi setiap file log. File ini mencatat aktivitas driver yang khusus untuk koneksi.

Jika LogPath nilainya tidak valid, maka driver mengirimkan informasi yang dicatat ke aliran keluaran standar () `System.out`

Menggunakan LogStream atau LogWriter

Aktifkan logging cukup lama untuk menangkap masalah. Logging mengurangi kinerja dan dapat mengkonsumsi sejumlah besar ruang disk.

Atur LogLevel kunci di URL koneksi Anda untuk mengaktifkan logging dan tentukan jumlah detail yang dikirim ke LogStream atau LogWriter ditentukan dalam file DriverManager.

Untuk mengaktifkan logging yang menggunakan LogStream atau LogWriter:

1. Untuk mengonfigurasi driver untuk mencatat informasi umum yang menjelaskan kemajuan driver, atur LogLevel properti ke 1 atau INFO.
2. Untuk memastikan bahwa pengaturan baru berlaku, restart aplikasi JDBC Anda dan sambungkan kembali ke server.

Konversi jenis data

Driver Amazon Redshift JDBC versi 2.x mendukung banyak format data umum, mengonversi antara jenis data Amazon Redshift, SQL, dan Java.

Tabel berikut mencantumkan pemetaan tipe data yang didukung.

Jenis Amazon Redshift	Jenis SQL	Jenis Java
BIGINT	SQL_BIGINT	Panjang
BOOLEAN	SQL_BIT	Boolean
CHAR	SQL_CHAR	String
DATE	SQL_TYPE_DATE	java.sql.date
DECIMAL	SQL_NUMERIK	BigDecimal
DOUBLE PRECISION	SQL_GANDA	Ganda
GEOMETRY	SQL_LONGVARBINARY	byte []
INTEGER	SQL_INTEGER	Bilangan Bulat
OID	SQL_BIGINT	Panjang

Jenis Amazon Redshift	Jenis SQL	Jenis Java
SUPER	SQL_LONGVARCHAR	String
REAL	SQL_REAL	Desimal
SMALLINT	SQL_SMALLINT	Pendek
TEXT	SQL_VARCHAR	String
TIME	SQL_TYPE_TIME	java.sql.waktu
JADWAL	SQL_TYPE_TIME	java.sql.waktu
TIMESTAMP	STEMPEL WAKTU SQL_TYPE_	java.sql.TimeStamp
TIMESTAMPTZ	STEMPEL WAKTU SQL_TYPE_	java.sql.TimeStamp
VARCHAR	SQL_VARCHAR	String

Menggunakan dukungan pernyataan yang disiapkan

Driver Amazon Redshift JDBC mendukung pernyataan yang disiapkan. Anda dapat menggunakan pernyataan yang disiapkan untuk meningkatkan kinerja kueri berparameter yang perlu dijalankan beberapa kali selama koneksi yang sama.

Pernyataan yang disiapkan adalah pernyataan SQL yang dikompilasi di sisi server tetapi tidak segera dijalankan. Pernyataan yang dikompilasi disimpan di server sebagai PreparedStatement objek sampai Anda menutup objek atau koneksi. Sementara objek itu ada, Anda dapat menjalankan pernyataan yang disiapkan sebanyak yang diperlukan menggunakan nilai parameter yang berbeda, tanpa harus mengkompilasi pernyataan lagi. Overhead yang dikurangi ini memungkinkan kumpulan kueri dijalankan lebih cepat.

Untuk informasi lebih lanjut tentang pernyataan yang disiapkan, lihat “Menggunakan Pernyataan yang Disiapkan” di [tutorial JDBC Basics dari Oracle](#).

Anda dapat menyiapkan pernyataan yang berisi beberapa kueri. Misalnya, pernyataan disiapkan berikut berisi dua query INSERT:

```
PreparedStatement pstmt = conn.prepareStatement("INSERT INTO  
MyTable VALUES (1, 'abc'); INSERT INTO CompanyTable VALUES  
(1, 'abc');");
```

Berhati-hatilah agar kueri ini tidak bergantung pada hasil kueri lain yang ditentukan dalam pernyataan yang disiapkan yang sama. Karena kueri tidak berjalan selama langkah persiapan, hasilnya belum dikembalikan, dan tidak tersedia untuk kueri lain dalam pernyataan yang disiapkan yang sama.

Misalnya, pernyataan yang disiapkan berikut, yang membuat tabel dan kemudian menyisipkan nilai ke dalam tabel yang baru dibuat, tidak diperbolehkan:

```
PreparedStatement pstmt = conn.prepareStatement("CREATE  
TABLE MyTable(col1 int, col2 varchar); INSERT INTO myTable  
VALUES (1, 'abc');");
```

Jika Anda mencoba menyiapkan pernyataan ini, server mengembalikan kesalahan yang menyatakan bahwa tabel tujuan (MyTable) belum ada. Query CREATE harus dijalankan sebelum query INSERT dapat disiapkan.

Perbedaan antara versi 2.2 dan 1.x dari driver JDBC

Bagian ini menjelaskan perbedaan informasi yang dikembalikan oleh versi 2.2 dan 1.x dari driver JDBC. Driver JDBC versi 1.x dihentikan.

Tabel berikut mencantumkan DatabaseMetadata informasi yang dikembalikan oleh fungsi `getDatabaseProductName ()` dan `getDatabaseProductVersion ()` untuk setiap versi driver JDBC. Driver JDBC versi 2.2 memperoleh nilai saat membuat koneksi. Driver JDBC versi 1.x memperoleh nilai sebagai hasil dari kueri.

Versi driver JDBC	<code>getDatabaseProductName ()</code> hasil	<code>getDatabaseProductVersion ()</code> hasil
2.2	Redshift	8.0.2
1.x	PostgreSQL	08.00.0002

Tabel berikut mencantumkan DatabaseMetadata informasi yang dikembalikan oleh `getTypeInfo` fungsi untuk setiap versi driver JDBC.

Versi driver JDBC	getTypeInfo hasil
2.2	Konsisten dengan tipe data Redshift
1.x	Konsisten dengan tipe data PostgreSQL

Membuat file inisialisasi (.ini) untuk driver JDBC versi 2.x

Dengan menggunakan file inisialisasi (.ini) untuk driver Amazon Redshift JDBC versi 2.x, Anda dapat menentukan parameter konfigurasi tingkat sistem. Misalnya, parameter otentikasi IDP federasi dapat bervariasi untuk setiap aplikasi. File.ini menyediakan lokasi umum untuk klien SQL untuk mendapatkan parameter konfigurasi yang diperlukan.

Anda dapat membuat file inisialisasi versi 2.x driver JDBC (.ini) yang berisi opsi konfigurasi untuk klien SQL. Nama default file tersebut adalah `rsjdbc.ini`. Driver JDBC versi 2.x memeriksa file.ini di lokasi berikut, terdaftar dalam urutan prioritas:

- `IniFile` parameter di URL koneksi atau di kotak dialog properti koneksi klien SQL. Pastikan `IniFile` parameter berisi path lengkap ke file.ini, termasuk nama file. Untuk informasi tentang `IniFile` parameter, lihat [IniFile](#). Jika `IniFile` parameter salah menentukan lokasi file.ini, kesalahan akan ditampilkan.
- Variabel lingkungan seperti `AMAZON_REDSHIFT_JDBC_INI_FILE` dengan path lengkap, termasuk nama file. Anda dapat menggunakan `rsjdbc.ini` atau menentukan nama file. Jika variabel lingkungan `AMAZON_REDSHIFT_JDBC_INI_FILE` salah menentukan lokasi file.ini, kesalahan akan ditampilkan.
- Direktori tempat file JAR driver berada.
- Direktori home pengguna.
- Direktori temp sistem.

Anda dapat mengatur file.ini menjadi beberapa bagian, misalnya `[DRIVER]`. Setiap bagian berisi pasangan kunci-nilai yang menentukan berbagai parameter koneksi. Anda dapat menggunakan `IniSection` parameter untuk menentukan bagian dalam file.ini. Untuk informasi tentang `IniSection` parameter, lihat [IniSection](#).

Berikut ini adalah contoh dari format file.ini, dengan bagian untuk `[DRIVER]`, `[DEV]`, `[QA]`, dan `[PROD]`. Bagian `[DRIVER]` dapat diterapkan ke koneksi apa pun.

```
[DRIVER]
key1=val1
key2=val2

[DEV]
key1=val1
key2=val2

[QA]
key1=val1
key2=val2

[PROD]
key1=val1
key2=val2
```

Driver JDBC versi 2.x memuat parameter konfigurasi dari lokasi berikut, tercantum dalam urutan prioritas:

- Parameter konfigurasi default dalam kode aplikasi.
- Properti bagian [DRIVER] dari file.ini, jika disertakan.
- Parameter konfigurasi bagian kustom, jika `IniSection` opsi disediakan di URL koneksi atau di kotak dialog properti koneksi klien SQL.
- Properti dari objek properti koneksi yang ditentukan dalam `getConnection` panggilan.
- Parameter konfigurasi dipercepat dalam URL koneksi.

Opsi untuk konfigurasi driver JDBC versi 2.x

Berikut ini, Anda dapat menemukan deskripsi untuk opsi yang dapat Anda tentukan untuk driver Amazon Redshift JDBC versi 2.2. Opsi konfigurasi tidak peka huruf besar/kecil.

Anda dapat mengatur properti konfigurasi menggunakan URL koneksi. Untuk informasi selengkapnya, lihat [Membangun URL koneksi](#).

Topik

- [AccessKeyID](#)
- [Izinkan DBUser Override](#)
- [App_ID](#)

- [App_Nama](#)
- [ApplicationName](#)
- [AuthProfile](#)
- [AutoCreate](#)
- [Client_ID](#)
- [Klien_Rahasia](#)
- [ClusterID](#)
- [Kompresi](#)
- [ConnectTimeout](#)
- [ConnectionTimeZone](#)
- [databaseMetadataCurrentDbOnly](#)
- [DbUser](#)
- [DbGroups](#)
- [DBNAME](#)
- [defaultRowFetchUkuran](#)
- [DisablesValidQuery](#)
- [enableFetchRingPenyangga](#)
- [enableMultiSqlSupport](#)
- [fetchRingBufferUkuran](#)
- [ForceLowercase](#)
- [GroupFederation](#)
- [HOST](#)
- [IAMDisableCache](#)
- [IAMDDuration](#)
- [Idc_Client_Display_Name](#)
- [Wilayah Idc_](#)
- [IDP_Host](#)
- [IDP_partisi](#)
- [IDP_port](#)

- [IDP_penyewa](#)
- [IDP_Response_Timeout](#)
- [IniFile](#)
- [IniSection](#)
- [IServerless](#)
- [Isuer_Url](#)
- [Dengarkan_Port](#)
- [Login_URL](#)
- [loginTimeout](#)
- [loginToRp](#)
- [LogLevel](#)
- [LogPath](#)
- [OverrideSchemaPatternType](#)
- [Partner_SPID](#)
- [Kata Sandi](#)
- [Plugin_Nama](#)
- [PELABUHAN](#)
- [Diutamakan_Peran](#)
- [Profil](#)
- [PWD](#)
- [QueryGroup](#)
- [ReadOnly](#)
- [Region](#)
- [reWriteBatchedSisipan](#)
- [reWriteBatchedInsertsSize](#)
- [roleArn](#)
- [roleSessionName](#)
- [cakupan](#)

- [SecretAccessKey](#)
- [SessionToken](#)
- [serverlessAcctId](#)
- [serverlessWorkGroup](#)
- [Pabrik Socket](#)
- [socketTimeout](#)
- [SSL](#)
- [SSL_tidak aman](#)
- [SSLCert](#)
- [SSLFactory](#)
- [SSLKey](#)
- [SSLMode](#)
- [SSLPassword](#)
- [SSLRootSertifikat](#)
- [StsEndpointUrl](#)
- [tcpKeepAlive](#)
- [token](#)
- [token_type](#)
- [UID](#)
- [Pengguna](#)
- [webIdentityToken](#)

AccessKeyID

- Nilai Default - Tidak Ada
- Tipe Data - String

Anda dapat menentukan parameter ini untuk memasukkan kunci akses IAM untuk pengguna atau peran. Anda biasanya dapat menemukan kunci dengan melihat dan string atau profil pengguna yang ada. Jika Anda menentukan parameter ini, Anda juga harus menentukan SecretAccessKey parameter. Jika diteruskan dalam URL JDBC, AccessKey ID harus dikodekan URL.

Parameter ini bersifat opsional.

Izinkan DBUser Override

- Nilai Default - 0
- Tipe Data - String

Opsi ini menentukan apakah driver menggunakan DbUser nilai dari pernyataan SAMP atau nilai yang ditentukan dalam properti DbUser koneksi di URL koneksi.

Parameter ini bersifat opsional.

1

Pengemudi menggunakan DbUser nilai dari pernyataan SAMP.

Jika pernyataan SAMP tidak menentukan nilai untuk DbUser, driver menggunakan nilai yang ditentukan dalam properti koneksi. DbUser Jika properti koneksi juga tidak menentukan nilai, driver menggunakan nilai yang ditentukan dalam profil koneksi.

0

Driver menggunakan DbUser nilai yang ditentukan dalam properti DbUser koneksi.

Jika properti DbUser koneksi tidak menentukan nilai, driver menggunakan nilai yang ditentukan dalam profil koneksi. Jika profil koneksi juga tidak menentukan nilai, driver menggunakan nilai dari pernyataan SAMP.

App_ID

- Nilai Default - Tidak Ada
- Tipe Data - String

ID unik yang disediakan OKTA yang terkait dengan aplikasi Amazon Redshift Anda.

Parameter ini diperlukan jika mengautentikasi melalui layanan Okta.

App_Nama

- Nilai Default - Tidak Ada

- Tipe Data - String

Nama aplikasi Okta yang Anda gunakan untuk mengautentikasi koneksi ke Amazon Redshift.

Parameter ini bersifat opsional.

ApplicationName

- Nilai Default - null
- Tipe Data - String

Nama aplikasi untuk diteruskan ke Amazon Redshift untuk tujuan audit.

Parameter ini bersifat opsional.

AuthProfile

- Nilai Default - Tidak Ada
- Tipe Data - String

Nama profil otentikasi yang akan digunakan untuk menghubungkan ke Amazon Redshift.

Parameter ini bersifat opsional.

AutoCreate

- Nilai Default - false
- Tipe Data - Boolean

Opsi ini menentukan apakah driver menyebabkan pengguna baru dibuat ketika pengguna tertentu tidak ada.

Parameter ini bersifat opsional.

true

Jika pengguna yang ditentukan oleh salah satu DBUser atau ID unik (UID) tidak ada, pengguna baru dengan nama itu akan dibuat.

false

Driver tidak menyebabkan pengguna baru dibuat. Jika pengguna yang ditentukan tidak ada, otentikasi gagal.

Client_ID

- Nilai Default - Tidak Ada
- Tipe Data - String

ID klien yang akan digunakan saat mengautentikasi koneksi menggunakan layanan Azure AD.

Parameter ini diperlukan jika mengautentikasi melalui layanan Azure AD.

Klien_Rahasia

- Nilai Default - Tidak Ada
- Tipe Data - String

Rahasia Klien untuk digunakan saat mengautentikasi koneksi menggunakan layanan Azure AD.

Parameter ini diperlukan jika mengautentikasi melalui layanan Azure AD.

ClusterID

- Nilai Default - Tidak Ada
- Tipe Data - String

Nama cluster Amazon Redshift yang ingin Anda sambungkan. Pengemudi mencoba mendeteksi parameter ini dari host yang diberikan. Jika Anda menggunakan Network Load Balancer (NLB) dan terhubung melalui IAM, driver akan gagal mendeteksinya, sehingga Anda dapat mengaturnya menggunakan opsi koneksi ini.

Parameter ini bersifat opsional.

Kompresi

- Nilai Default - off
- Tipe Data - String

Metode kompresi yang digunakan untuk komunikasi protokol kawat antara server Amazon Redshift dan klien atau driver.

Parameter ini bersifat opsional.

Anda dapat menentukan salah satu nilai berikut:

- lz4

Menetapkan metode kompresi yang digunakan untuk komunikasi protokol kawat dengan Amazon Redshift ke lz4.

- off

Tidak menggunakan kompresi untuk komunikasi protokol kawat dengan Amazon Redshift.

ConnectTimeout

- Nilai Default - 10
- Tipe Data - Integer

Nilai batas waktu yang digunakan untuk operasi koneksi soket. Jika waktu yang diperlukan untuk membuat koneksi Amazon Redshift melebihi nilai ini, koneksi dianggap tidak tersedia. Batas waktu ditentukan dalam hitungan detik. Nilai 0 berarti tidak ada batas waktu yang ditentukan.

Parameter ini bersifat opsional.

ConnectionTimeZone

- Nilai Default - LOKAL
- Tipe Data - String

Zona waktu tingkat sesi.

Parameter ini bersifat opsional.

Anda dapat menentukan salah satu nilai berikut:

LOKAL

Mengkonfigurasi zona waktu tingkat sesi ke zona waktu JVM LOKAL.

PELADEN

Mengonfigurasi zona waktu tingkat sesi ke zona waktu yang ditetapkan untuk pengguna di server Amazon Redshift. Anda dapat mengonfigurasi zona waktu tingkat sesi untuk pengguna dengan perintah berikut:

```
ALTER USER  
[...]  
SET TIMEZONE TO [...];
```

databaseMetadataCurrentDbOnly

- Nilai Default - benar
- Tipe Data - Boolean

Opsi ini menentukan apakah API metadata mengambil data dari semua database yang dapat diakses atau hanya dari database yang terhubung.

Parameter ini bersifat opsional.

Anda dapat menentukan salah satu nilai berikut:

true

Aplikasi mengambil metadata dari database tunggal.

false

Aplikasi mengambil metadata dari semua database yang dapat diakses.

DbUser

- Nilai Default - Tidak Ada
- Tipe Data - String

ID pengguna untuk digunakan dengan akun Amazon Redshift Anda. Anda dapat menggunakan ID yang saat ini tidak ada jika Anda telah mengaktifkan AutoCreate properti.

Parameter ini bersifat opsional.

DbGroups

- Nilai Default - PUBLIK
- Tipe Data - String

Daftar dipisahkan koma dari nama grup database yang ada yang DBUser bergabung untuk sesi saat ini.

Parameter ini bersifat opsional.

DBNAME

- Nilai Default - null
- Tipe Data - String

Nama database untuk terhubung. Anda dapat menggunakan opsi ini untuk menentukan nama database di URL koneksi JDBC.

Parameter ini diperlukan. Anda harus menentukan nama database, baik di URL koneksi atau di properti koneksi aplikasi klien.

defaultRowFetchUkuran

- Nilai Default - 0
- Tipe Data - Integer

Opsi ini menentukan nilai default untuk getFetchSize.

Parameter ini bersifat opsional.

Anda dapat menentukan salah satu nilai berikut:

0

Ambil semua baris dalam satu operasi.

Integer positif

Jumlah baris yang akan diambil dari database untuk setiap iterasi pengambilan. ResultSet

DisableIsValidQuery

- Nilai Default - Salah
- Tipe Data - Boolean

Opsi ini menentukan apakah driver mengirimkan query database baru saat menggunakan metode `connection.isValid ()` untuk menentukan apakah koneksi database aktif.

Parameter ini bersifat opsional.

`true`

Driver tidak mengirimkan kueri saat menggunakan `Connection.isValid ()` untuk menentukan apakah koneksi database aktif. Hal ini dapat menyebabkan driver salah mengidentifikasi koneksi database sebagai aktif jika server database telah dimatikan secara tak terduga.

`false`

Driver mengirimkan kueri saat menggunakan `connection.isValid ()` untuk menentukan apakah koneksi database aktif.

enableFetchRingPenyangga

- Nilai Default - benar
- Tipe Data - Boolean

Opsi ini menentukan bahwa driver mengambil baris menggunakan buffer cincin pada utas terpisah. Parameter `fetchRingBuffer Ukuran` menentukan ukuran buffer cincin.

Buffer ring mengimplementasikan manajemen memori otomatis di JDBC untuk mencegah kesalahan out-of-memory (OOM) selama operasi pengambilan data. Buffer ring memantau ukuran sebenarnya dari data buffer secara real-time, memastikan penggunaan memori total oleh driver tetap dalam batas yang ditentukan. Ketika kapasitas buffer tercapai, driver menghentikan operasi pengambilan data, mencegah memori meluap tanpa memerlukan intervensi manual. Pengamanan bawaan ini menghilangkan kesalahan OOM secara otomatis, tanpa konfigurasi yang diperlukan dari pengguna.

Jika transaksi mendeteksi Pernyataan yang berisi beberapa perintah SQL yang dipisahkan oleh titik koma, buffer ring fetch untuk transaksi tersebut disetel ke `false`. `enableFetchRingNilai` buffer tidak berubah.

Parameter ini bersifat opsional.

 Note

Ketika buffer cincin dinonaktifkan dan ukuran pengambilan tidak dikonfigurasi dengan benar, masalah out-of-memory (OOM) dapat terjadi. [Untuk informasi selengkapnya tentang mengonfigurasi ukuran pengambilan, lihat di sini.](#)

enableMultiSqlSupport

- Nilai Default - benar
- Tipe Data - Boolean

Opsi ini menentukan apakah akan memproses beberapa perintah SQL dipisahkan oleh titik koma dalam Pernyataan.

Parameter ini bersifat opsional.

Anda dapat menentukan salah satu nilai berikut:

true

Driver memproses beberapa perintah SQL, dipisahkan oleh titik koma, dalam objek Pernyataan.

false

Driver mengembalikan kesalahan untuk beberapa perintah SQL dalam satu Pernyataan.

fetchRingBufferUkuran

- Nilai Default - 1G
- Tipe Data - String

Opsi ini menentukan ukuran buffer cincin yang digunakan saat mengambil set hasil. Anda dapat menentukan ukuran dalam byte, misalnya 1K untuk 1 KB, 5000 untuk 5.000 byte, 1M untuk 1 MB, 1G untuk 1 GB, dan seterusnya. Anda juga dapat menentukan persentase memori heap. Pengemudi berhenti mengambil baris setelah mencapai batas. Mengambil dilanjutkan saat aplikasi membaca baris dan membebaskan ruang di buffer ring.

Parameter ini bersifat opsional.

ForceLowercase

- Nilai Default - false
- Tipe Data - Boolean

Opsi ini menentukan apakah driver huruf kecil semua grup database (DbGroups) yang dikirim dari penyedia identitas ke Amazon Redshift saat menggunakan autentikasi masuk tunggal.

Parameter ini bersifat opsional.

true

Driver huruf kecil semua grup database yang dikirim dari penyedia identitas.

false

Pengemudi tidak mengubah grup basis data.

GroupFederation

- Nilai Default - false
- Tipe Data - Boolean

Opsi ini menentukan apakah akan menggunakan grup IDP Amazon Redshift. Ini didukung oleh API GetClusterCredentials V2.

Parameter ini bersifat opsional.

true

Gunakan grup Amazon Redshift Identity Provider (IDP).

false

Gunakan STS API dan GetClusterCredentials untuk federasi pengguna dan tentukan secara eksplisit DbGroups untuk koneksi.

HOST

- Nilai Default - null
- Tipe Data - String

Nama host dari server Amazon Redshift untuk terhubung. Anda dapat menggunakan opsi ini untuk menentukan nama host di URL koneksi JDBC.

Parameter ini diperlukan. Anda harus menentukan nama host, baik di URL koneksi atau di properti koneksi aplikasi klien.

IAMDisableCache

- Nilai Default - false
- Tipe Data - Boolean

Opsi ini menentukan apakah kredensyal IAM di-cache.

Parameter ini bersifat opsional.

true

Kredensi IAM tidak di-cache.

false

Kredensi IAM di-cache. Ini meningkatkan kinerja saat permintaan ke gateway API dibatasi, misalnya.

IAMDURATION

- Nilai Default - 900
- Tipe Data - Integer

Lamanya waktu, dalam hitungan detik, hingga kredensyal IAM sementara kedaluwarsa.

- Nilai minimum - 900
- Nilai maksimum - 3.600

Parameter ini bersifat opsional.

`Idc_Client_Display_Name`

- Nilai Default - Driver Amazon Redshift JDBC
- Tipe Data - String

Nama tampilan yang akan digunakan untuk klien yang menggunakan `BrowserIdcAuthPlugin`.

Parameter ini bersifat opsional.

`Wilayah Idc_`

- Nilai Default - Tidak Ada
- Tipe Data - String

AWS Wilayah di mana instans IAM Identity Center berada.

Parameter ini diperlukan hanya ketika mengautentikasi menggunakan `BrowserIdcAuthPlugin` dalam opsi konfigurasi `plugin_name`.

`IDP_Host`

- Nilai Default - Tidak Ada
- Tipe Data - String

Host iDP (penyedia identitas) yang Anda gunakan untuk mengautentikasi ke Amazon Redshift. Ini dapat ditentukan baik dalam string koneksi atau dalam profil.

Parameter ini bersifat opsional.

`IDP_partisi`

- Nilai Default - Tidak Ada
- Tipe Data - String

Menentukan partisi cloud tempat penyedia identitas Anda (iDP) dikonfigurasi. Ini menentukan titik akhir otentikasi IDP mana yang terhubung ke driver.

Jika parameter ini dibiarkan kosong, driver default ke partisi komersial. Kemungkinan nilainya adalah:

- `us-gov`: Gunakan nilai ini jika IDP Anda dikonfigurasi di Azure Government. Misalnya, Azure AD Government menggunakan `login.microsoftonline.us` endpoint.
- `cn`: Gunakan nilai ini jika IDP Anda dikonfigurasi di partisi cloud Tiongkok. Misalnya, Azure AD Tiongkok menggunakan titik akhir. `login.chinacloudapi.cn`

Parameter ini bersifat opsional.

IDP_port

- Nilai Default - Tidak Ada
- Tipe Data - String

Port yang digunakan oleh iDP (penyedia identitas). Anda dapat menentukan port baik dalam string koneksi atau dalam profil. Port default adalah 5439. Bergantung pada port yang Anda pilih saat membuat, memodifikasi, atau memigrasi klaster, izinkan akses ke port yang dipilih.

Parameter ini bersifat opsional.

IDP_penyewa

- Nilai Default - Tidak Ada
- Tipe Data - String

ID penyewa Azure AD untuk aplikasi Amazon Redshift Anda.

Parameter ini diperlukan jika mengautentikasi melalui layanan Azure AD.

IDP_Response_Timeout

- Nilai Default - 120
- Tipe Data - Integer

Jumlah waktu, dalam hitungan detik, driver menunggu respons SAMP dari penyedia identitas saat menggunakan layanan SAMP atau Azure AD melalui plugin browser.

Parameter ini bersifat opsional.

IniFile

- Nilai Default - Tidak Ada
- Tipe Data - String

Jalur lengkap dari file.ini, termasuk nama file. Contoh:

```
IniFile="C:\tools\rsjdbc.ini"
```

Untuk informasi tentang file.ini, lihat [Membuat file inisialisasi \(.ini\) untuk driver JDBC versi 2.x](#).

Parameter ini bersifat opsional.

IniSection

- Nilai Default - Tidak Ada
- Tipe Data - String

Nama bagian dalam file.ini yang berisi opsi konfigurasi. Untuk informasi tentang file.ini, lihat [Membuat file inisialisasi \(.ini\) untuk driver JDBC versi 2.x](#).

Contoh berikut menentukan bagian [Prod] dari file.ini:

```
IniSection="Prod"
```

Parameter ini bersifat opsional.

IServerless

- Nilai Default - false
- Tipe Data - Boolean

Opsi ini menentukan apakah host endpoint Amazon Redshift adalah instance tanpa server. Pengemudi mencoba mendeteksi parameter ini dari host yang diberikan. Jika Anda menggunakan Network Load Balancer (NLB), driver akan gagal mendeteksinya, sehingga Anda dapat mengaturnya di sini.

Parameter ini bersifat opsional.

true

Host endpoint Amazon Redshift adalah instance tanpa server.

false

Host endpoint Amazon Redshift adalah cluster yang disediakan.

Isuer_Url

- Nilai Default - Tidak Ada
- Tipe Data - String

Menunjuk ke titik akhir instance server AWS IAM Identity Center.

Parameter ini diperlukan hanya ketika mengautentikasi menggunakan BrowserIdcAuthPlugin dalam opsi konfigurasi plugin_name.

Dengarkan_Port

- Nilai default - 7890
- Tipe Data - Integer

Port yang digunakan driver untuk menerima respons SAMP dari penyedia identitas atau kode otorisasi saat menggunakan layanan SAMP, Azure AD, atau AWS Identity Center melalui plugin browser.

Parameter ini bersifat opsional.

Login_URL

- Nilai Default - Tidak Ada
- Tipe Data - String

URL untuk sumber daya di situs web penyedia identitas saat menggunakan layanan SAMP atau Azure AD melalui plugin browser.

Parameter ini diperlukan jika mengautentikasi dengan layanan SAMP atau Azure AD melalui plugin browser.

loginTimeout

- Nilai Default - 0
- Tipe Data - Integer

Jumlah detik untuk menunggu sebelum waktu habis saat menghubungkan dan mengautentikasi ke server. Jika membuat koneksi membutuhkan waktu lebih lama dari ambang ini, maka koneksi dibatalkan.

Ketika properti ini disetel ke 0, koneksi tidak akan habis waktu.

Parameter ini bersifat opsional.

loginToRp

- Nilai Default - `urn:amazon:webservices`
- Tipe Data - String

Pihak yang mengandalkan percaya bahwa Anda ingin menggunakan untuk jenis otentikasi AD FS.

Parameter ini bersifat opsional.

LogLevel

- Nilai Default - 0
- Tipe Data - Integer

Gunakan properti ini untuk mengaktifkan atau mematikan logging di driver dan untuk menentukan jumlah detail yang disertakan dalam file log.

Aktifkan logging hanya cukup lama untuk menangkap masalah. Logging mengurangi kinerja dan dapat mengkonsumsi sejumlah besar ruang disk.

Parameter ini bersifat opsional.

Atur parameter ke salah satu nilai berikut:

0

Nonaktifkan semua logging.

1

Aktifkan logging pada level FATAL, yang mencatat peristiwa kesalahan yang sangat parah yang akan menyebabkan pengemudi membatalkan.

2

Aktifkan logging pada level ERROR, yang mencatat peristiwa kesalahan yang mungkin masih memungkinkan driver untuk terus berjalan.

3

Aktifkan pencatatan pada tingkat PERINGATAN, yang mencatat peristiwa yang mungkin mengakibatkan kesalahan jika tindakan tidak dilakukan.

4

Aktifkan logging pada tingkat INFO, yang mencatat informasi umum yang menjelaskan kemajuan driver.

5

Aktifkan logging pada level DEBUG, yang mencatat informasi terperinci yang berguna untuk men-debug driver.

6

Aktifkan logging pada level TRACE, yang mencatat semua aktivitas driver.

Saat logging diaktifkan, driver menghasilkan file log berikut di lokasi yang ditentukan dalam LogPath properti:

- **redshift_jdbc.log**— File yang mencatat aktivitas driver yang tidak spesifik untuk koneksi.
- **redshift_jdbc_connection_[Number].log**— File untuk setiap koneksi yang dibuat ke database, di mana [Number] adalah nomor yang membedakan setiap file log dari yang lain. File ini mencatat aktivitas driver yang khusus untuk koneksi.

Jika LogPath nilainya tidak valid, driver mengirimkan informasi yang dicatat ke aliran output standar, `System.out`

LogPath

- Nilai Default - Direktori kerja saat ini.

- Tipe Data - String

Jalur lengkap ke folder tempat driver menyimpan file log saat properti DSILog Level diaktifkan.

Untuk memastikan bahwa URL koneksi kompatibel dengan semua aplikasi JDBC, kami sarankan Anda melarikan diri dari garis miring terbalik (\) di jalur file Anda dengan mengetik garis miring terbalik lainnya.

Parameter ini bersifat opsional.

OverrideSchemaPatternType

- Nilai Default - null
- Tipe Data - Integer

Opsi ini menentukan apakah akan mengganti jenis query yang digunakan dalam panggilan getTables.

0

Tidak ada Skema Universal Query

1

Kueri Skema Lokal

2

Kueri Skema Eksternal

Parameter ini bersifat opsional.

Partner_SPID

- Nilai Default - Tidak Ada
- Tipe Data - String

Nilai SPID mitra (ID penyedia layanan) yang akan digunakan saat mengautentikasi koneksi menggunakan layanan. PingFederate

Parameter ini bersifat opsional.

Kata Sandi

- Nilai Default - Tidak Ada
- Tipe Data - String

Saat menghubungkan menggunakan otentikasi IAM melalui IDP, ini adalah kata sandi untuk server IDP_Host. Saat menggunakan otentikasi standar, ini dapat digunakan untuk kata sandi database Amazon Redshift alih-alih PWD.

Parameter ini bersifat opsional.

Plugin_Nama

- Nilai Default - Tidak Ada
- Tipe Data - String

Nama kelas yang sepenuhnya memenuhi syarat untuk mengimplementasikan plugin penyedia kredensial tertentu.

Parameter ini bersifat opsional.

Opsi penyedia berikut didukung:

- **AdfsCredentialsProvider**— Layanan Federasi Direktori Aktif.
- **AzureCredentialsProvider**— Layanan Microsoft Azure Active Directory (AD).
- **BasicJwtCredentialsProvider**— Layanan Token Web JSON (JWT).
- **BasicSamlCredentialsProvider**— Kredensial Security Assertion Markup Language (SAMP) yang dapat Anda gunakan dengan banyak penyedia layanan SAMP.
- **BrowserAzureCredentialsProvider**— Browser Layanan Microsoft Azure Active Directory (AD).
- **BrowserAzureOAuth2CredentialsProvider**— Layanan Browser Microsoft Azure Active Directory (AD) untuk Otentikasi Asli.
- **BrowserIdcAuthPlugin**- Plugin otorisasi menggunakan AWS IAM Identity Center.
- **BrowserSamlCredentialsProvider**— Browser SAMP untuk layanan SAMP seperti Okta, Ping, atau ADFS.

- **IdpTokenAuthPlugin**- Plugin otorisasi yang menerima token AWS IAM Identity Center atau OpenID Connect (OIDC) JSON berbasis token identitas (JWT) dari penyedia identitas web mana pun yang terhubung ke IAM Identity Center. AWS
- **OktaCredentialsProvider**— Layanan Okta.
- **PingCredentialsProvider**— PingFederate Layanan.

PELABUHAN

- Nilai Default - null
- Tipe Data - Integer

Port server Amazon Redshift untuk terhubung. Anda dapat menggunakan opsi ini untuk menentukan port di URL koneksi JDBC.

Parameter ini bersifat opsional.

Diutamakan_Peran

- Nilai Default - Tidak Ada
- Tipe Data - String

Peran IAM yang ingin Anda asumsikan selama koneksi ke Amazon Redshift.

Parameter ini bersifat opsional.

Profil

- Nilai Default - Tidak Ada
- Tipe Data - String

Nama profil yang akan digunakan untuk otentikasi IAM. Profil ini berisi properti koneksi tambahan yang tidak ditentukan dalam string koneksi.

Parameter ini bersifat opsional.

PWD

- Nilai Default - Tidak Ada

- Tipe Data - String

Kata sandi yang sesuai dengan nama pengguna Amazon Redshift yang Anda berikan menggunakan UID properti.

Parameter ini bersifat opsional.

QueryGroup

- Nilai Default - null
- Tipe Data - String

Opsi ini menetapkan kueri ke antrian saat runtime dengan menetapkan kueri Anda ke grup kueri yang sesuai. Grup kueri diatur untuk sesi. Semua kueri yang berjalan pada koneksi milik grup kueri ini.

Parameter ini bersifat opsional.

ReadOnly

- Nilai Default - false
- Tipe Data - Boolean

Properti ini menentukan apakah driver dalam mode read-only.

Parameter ini bersifat opsional.

true

Koneksi dalam mode read-only dan tidak dapat menulis ke penyimpanan data.

false

Koneksi tidak dalam mode read-only dan dapat menulis ke penyimpanan data.

Region

- Nilai Default - null
- Tipe Data - String

Opsi ini menentukan AWS Wilayah di mana cluster berada. Jika Anda menentukan StsEndPoint opsi, opsi Wilayah diabaikan. Operasi Redshift GetClusterCredentials API juga menggunakan opsi Region.

Parameter ini bersifat opsional.

reWriteBatchedSisipan

- Nilai Default - false
- Tipe Data - Boolean

Opsi ini memungkinkan pengoptimalan untuk menulis ulang dan menggabungkan pernyataan INSERT yang kompatibel ke dalam batch.

Parameter ini bersifat opsional.

reWriteBatchedInsertsSize

- Nilai Default - 128
- Tipe Data - Integer

Opsi ini memungkinkan pengoptimalan untuk menulis ulang dan menggabungkan pernyataan INSERT yang kompatibel ke dalam batch. Nilai ini harus meningkat secara eksponensial dengan kekuatan 2.

Parameter ini bersifat opsional.

roleArn

- Nilai Default - Tidak Ada
- Tipe Data - String

Nama Sumber Daya Amazon (ARN) peran. Pastikan untuk menentukan parameter ini saat Anda menentukan opsi BasicJwtCredentialsProvider Plugin_Name. Anda menentukan ARN dalam format berikut:

arn:*partition:service:region:account-id:resource-id*

Parameter ini diperlukan jika Anda menentukan BasicJwtCredentialsProvider untuk opsi Plugin_Name.

roleSessionName

- Nilai Default - jwt_redshift_session
- Tipe Data - String

Sebuah pengenalan untuk sesi peran yang diambil. Biasanya, Anda melewati nama atau pengenalan yang terkait dengan pengguna aplikasi Anda. Kredensial keamanan sementara yang digunakan aplikasi Anda terkait dengan pengguna tersebut. Anda dapat menentukan parameter ini ketika Anda menentukan BasicJwtCredentialsProvider untuk opsi Plugin_Name.

Parameter ini bersifat opsional.

cakupan

- Nilai Default - Tidak Ada
- Tipe Data - String

Daftar cakupan yang dipisahkan ruang yang dapat disetujui pengguna. Anda menentukan parameter ini sehingga aplikasi Microsoft Azure Anda bisa mendapatkan persetujuan untuk APIs yang ingin Anda panggil. Anda dapat menentukan parameter ini ketika Anda menentukan BrowserAzure OAuth2 CredentialsProvider untuk opsi Plugin_Name.

Parameter ini diperlukan untuk BrowserAzure OAuth2 CredentialsProvider plug-in.

SecretAccessKey

- Nilai Default - Tidak Ada
- Tipe Data - String

Kunci akses IAM untuk pengguna atau peran. Jika ini ditentukan, maka AccessKey ID juga harus ditentukan. Jika diteruskan dalam URL JDBC, SecretAccessKey harus dikodekan URL.

Parameter ini bersifat opsional.

SessionToken

- Nilai Default - Tidak Ada
- Tipe Data - String

Token sesi IAM sementara yang terkait dengan peran IAM yang Anda gunakan untuk mengautentikasi. Jika diteruskan dalam URL JDBC, token sesi IAM sementara harus dikodekan URL.

Parameter ini bersifat opsional.

`serverlessAcctId`

- Nilai Default - null
- Tipe Data - String

ID akun Tanpa Server Amazon Redshift. Pengemudi mencoba mendeteksi parameter ini dari host yang diberikan. Jika Anda menggunakan Network Load Balancer (NLB), driver akan gagal mendeteksinya, sehingga Anda dapat mengaturnya di sini.

Parameter ini bersifat opsional.

`serverlessWorkGroup`

- Nilai Default - null
- Tipe Data - String

Nama workgroup Amazon Redshift Tanpa Server. Pengemudi mencoba mendeteksi parameter ini dari host yang diberikan. Jika Anda menggunakan Network Load Balancer (NLB), driver akan gagal mendeteksinya, sehingga Anda dapat mengaturnya di sini.

Parameter ini bersifat opsional.

`Pabrik Socket`

- Nilai Default - null
- Tipe Data - String

Opsi ini menentukan pabrik soket untuk pembuatan soket.

Parameter ini bersifat opsional.

`socketTimeout`

- Nilai Default - 0

- Tipe Data - Integer

Jumlah detik untuk menunggu selama operasi pembacaan soket sebelum waktu habis. Jika operasi memakan waktu lebih lama dari ambang ini, maka koneksi ditutup. Ketika properti ini disetel ke 0, koneksi tidak akan habis waktu.

Parameter ini bersifat opsional.

SSL

- Nilai Default - BENAR
- Tipe Data - String

Gunakan properti ini untuk mengaktifkan atau mematikan SSL untuk koneksi.

Parameter ini bersifat opsional.

Anda dapat menentukan salah satu nilai berikut:

BETUL

Driver terhubung ke server melalui SSL.

SALAH

Driver terhubung ke server tanpa menggunakan SSL. Opsi ini tidak didukung dengan otentikasi IAM.

Atau, Anda dapat mengkonfigurasi AuthMech properti.

SSL_tidak aman

- Nilai Default - benar
- Tipe Data - String

Properti ini menunjukkan apakah sertifikat server host IDP harus diverifikasi.

Parameter ini bersifat opsional.

Anda dapat menentukan salah satu nilai berikut:

true

Pengemudi tidak memeriksa keaslian sertifikat server IDP.

false

Pengemudi memeriksa keaslian sertifikat server IDP.

SSLCert

- Nilai Default - Tidak Ada
- Tipe Data - String

Jalur lengkap file.pem atau .crt yang berisi sertifikat CA tepercaya tambahan untuk memverifikasi instance server Amazon Redshift saat menggunakan SSL.

Parameter ini diperlukan jika SSLKey ditentukan.

SSLFactory

- Nilai Default - Tidak Ada
- Tipe Data - String

Pabrik SSL untuk digunakan saat menghubungkan ke server melalui TLS/SSL tanpa menggunakan sertifikat server.

SSLKey

- Nilai Default - Tidak Ada
- Tipe Data - String

Jalur lengkap dari file.der yang berisi file PKCS8 kunci untuk memverifikasi sertifikat yang ditentukan dalam SSLCert file.

Parameter ini diperlukan jika SSLCert ditentukan.

SSLMode

- Nilai Default - verify-ca

- Tipe Data - String

Gunakan properti ini untuk menentukan bagaimana driver memvalidasi sertifikat saat TLS/SSL diaktifkan.

Parameter ini bersifat opsional.

Anda dapat menentukan salah satu nilai berikut:

`verify-ca`

Pengemudi memverifikasi bahwa sertifikat tersebut berasal dari otoritas sertifikat tepercaya (CA).

`verify-full`

Pengemudi memverifikasi bahwa sertifikat berasal dari CA tepercaya dan bahwa nama host dalam sertifikat cocok dengan nama host yang ditentukan dalam URL koneksi.

`SSLPassword`

- Nilai Default - 0
- Tipe Data - String

Kata sandi untuk file kunci terenkripsi yang ditentukan dalam `SSLKey`

Parameter ini diperlukan jika `SSLKey` ditentukan dan file kunci dienkripsi.

`SSLRootSertifikat`

- Nilai Default - Tidak Ada
- Tipe Data - String

Jalur lengkap file.pem atau .crt yang berisi sertifikat CA root untuk memverifikasi instance Amazon Redshift Server saat menggunakan SSL.

`StsEndpointUrl`

- Nilai Default - null
- Tipe Data - String

Anda dapat menentukan titik akhir AWS Security Token Service (AWS STS). Jika Anda menentukan opsi ini, opsi Wilayah diabaikan. Anda hanya dapat menentukan protokol aman (HTTPS) untuk titik akhir ini.

tcpKeepAlive

- Nilai Default - BENAR
- Tipe Data - String

Gunakan properti ini untuk mengaktifkan atau mematikan keepalives TCP.

Parameter ini bersifat opsional.

Anda dapat menentukan salah satu nilai berikut:

BETUL

Driver menggunakan TCP keepalives untuk mencegah koneksi dari timing out.

SALAH

Driver tidak menggunakan TCP keepalives.

token

- Nilai Default - Tidak Ada
- Tipe Data - String

Pusat AWS Identitas IAM menyediakan token akses atau OpenID Connect (OIDC) JSON Web Token (JWT) yang disediakan oleh penyedia identitas web yang terhubung dengan IAM Identity Center. AWS Aplikasi Anda harus menghasilkan token ini dengan mengautentikasi pengguna aplikasi Anda dengan AWS IAM Identity Center atau penyedia identitas yang ditautkan dengan AWS IAM Identity Center.

Parameter ini bekerja dengan `IdpTokenAuthPlugin`.

token_type

- Nilai Default - Tidak Ada
- Tipe Data - String

Jenis token yang digunakan di `IdpTokenAuthPlugin`.

Anda dapat menentukan salah satu nilai berikut:

`ACCESS_TOKEN`

Masukkan ini jika Anda menggunakan token akses yang disediakan Pusat Identitas AWS IAM.

`EXT_JWT`

Masukkan ini jika Anda menggunakan OpenID Connect (OIDC) JSON Web Token (JWT) yang disediakan oleh penyedia identitas berbasis web yang terintegrasi dengan IAM Identity Center.

`AWS`

Parameter ini bekerja dengan `IdpTokenAuthPlugin`.

`UID`

- Nilai Default - Tidak Ada
- Tipe Data - String

Nama pengguna database yang Anda gunakan untuk mengakses database.

Parameter ini diperlukan.

`Pengguna`

- Nilai Default - Tidak Ada
- Tipe Data - String

Saat menghubungkan menggunakan otentikasi IAM melalui IDP, ini adalah nama pengguna untuk server `idp_host`. Saat menggunakan otentikasi standar, ini dapat digunakan untuk nama pengguna database Amazon Redshift.

Parameter ini bersifat opsional.

`webIdentityToken`

- Nilai Default - Tidak Ada
- Tipe Data - String

Token akses OAuth 2.1 atau token OpenID Connect ID yang disediakan oleh penyedia identitas. Aplikasi Anda harus mendapatkan token ini dengan mengautentikasi pengguna aplikasi Anda dengan penyedia identitas web. Pastikan untuk menentukan parameter ini saat Anda menentukan opsi `BasicJwtCredentialsProvider Plugin_Name`.

Parameter ini diperlukan jika Anda menentukan `BasicJwtCredentialsProvider` untuk opsi `Plugin_Name`.

Versi sebelumnya dari JDBC driver version 2.x

Unduh versi sebelumnya dari driver Amazon Redshift JDBC versi 2.x hanya jika alat Anda memerlukan versi driver tertentu.

Ini adalah driver JDBC versi 2.x versi 2.x yang kompatibel dengan JDBC 4.2 sebelumnya:

- <https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.2.4/redshift-jdbc42-2.2.4.zip> <https://s3.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/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.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/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.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/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.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/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.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/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.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/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.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/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.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/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.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/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.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/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.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/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.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/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.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/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.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/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.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/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.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/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.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/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.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/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.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/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.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/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.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/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.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/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.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/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.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/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.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/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.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/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.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/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.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/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.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/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.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/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>
<https://s3.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/drivers/jdbc/2.1.0.7/redshift-jdbc42-2.1.0.7.zip>

Konektor Python Amazon Redshift

Dengan menggunakan konektor Amazon Redshift untuk Python, Anda dapat mengintegrasikan pekerjaan dengan [AWS SDK for Python \(Boto3\)](#), dan juga [panda](#) dan [Numerical Python \(\)](#). NumPy Untuk informasi lebih lanjut tentang panda, lihat repositori [panda GitHub](#). Untuk informasi lebih lanjut tentang NumPy, lihat [NumPy GitHubrepositori](#).

Konektor Amazon Redshift Python menyediakan solusi open source. Anda dapat menelusuri kode sumber, meminta penyempurnaan, melaporkan masalah, dan memberikan kontribusi.

Untuk menggunakan konektor Amazon Redshift Python, pastikan Anda memiliki Python versi 3.6 atau yang lebih baru. Untuk informasi selengkapnya, lihat [perjanjian SIM Amazon Redshift Python](#).

Konektor Amazon Redshift Python menyediakan yang berikut:

- AWS Identity and Access Management (IAM) otentikasi. Untuk informasi selengkapnya, lihat [Manajemen identitas dan akses di Amazon Redshift](#).
- Otentikasi penyedia identitas menggunakan akses API federasi. Akses API federasi didukung untuk penyedia identitas perusahaan seperti berikut ini:
 - Azure AD. Untuk informasi selengkapnya, lihat postingan blog AWS Big Data Akses [Federate Amazon Redshift dengan sistem masuk tunggal Microsoft Azure AD](#).
 - Layanan Federasi Direktori Aktif. Untuk informasi selengkapnya, lihat posting blog AWS Big Data [akses Federasi ke kluster Amazon Redshift Anda dengan Active Directory Federation Services \(AD FS\)](#): Bagian 1.
 - Okta. Untuk informasi lebih lanjut, lihat posting blog AWS Big Data [akses Federate Amazon Redshift dengan Okta sebagai](#) penyedia identitas.
 - PingFederate. Untuk informasi lebih lanjut, lihat [PingFederate situs](#).
 - JumpCloud. Untuk informasi lebih lanjut, lihat [JumpCloud situs](#).
- Jenis data Amazon Redshift.

Konektor Amazon Redshift Python mengimplementasikan Python Database API Specification 2.0. Untuk informasi selengkapnya, lihat [PEP 249—Spesifikasi API Database Python v2.0](#) di situs web Python.

Topik

- [Memasang konektor Amazon Redshift Python](#)
- [Opsis konfigurasi untuk konektor Amazon Redshift Python](#)

- [Mengimpor konektor Python](#)
- [Mengintegrasikan konektor Python dengan NumPy](#)
- [Mengintegrasikan konektor Python dengan panda](#)
- [Menggunakan plugin penyedia identitas](#)
- [Contoh penggunaan konektor Amazon Redshift Python](#)
- [Referensi API untuk konektor Amazon Redshift Python](#)

Memasang konektor Amazon Redshift Python

Anda dapat menggunakan salah satu metode berikut untuk menginstal konektor Amazon Redshift Python:

- Indeks Package Python (PyPI)
- Conda
- Mengkloning repositori GitHub

Menginstal konektor Python dari PyPI

Untuk menginstal konektor Python dari Python Package Index (PyPI), Anda dapat menggunakan pip. Untuk melakukan ini, jalankan perintah berikut.

```
>>> pip install redshift_connector
```

Anda dapat menginstal konektor dalam lingkungan virtual. Untuk melakukan ini, jalankan perintah berikut.

```
>>> pip install redshift_connector
```

Secara opsional, Anda dapat menginstal panda dan NumPy dengan konektor.

```
>>> pip install 'redshift_connector[full]'
```

Untuk informasi lebih lanjut tentang pip, lihat situs [pip](#).

Menginstal konektor Python dari Conda

Anda dapat menginstal konektor Python dari Anaconda.org.

```
>>>conda install -c conda-forge redshift_connector
```

Menginstal konektor Python dengan mengkloning repositori dari GitHub AWS

Untuk menginstal konektor Python dari sumber, kloning repositori dari GitHub AWS Setelah Anda menginstal Python dan virtualenv, atur lingkungan Anda dan instal dependensi yang diperlukan dengan menjalankan perintah berikut.

```
$ 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
```

Opsi konfigurasi untuk konektor Amazon Redshift Python

Berikut ini, Anda dapat menemukan deskripsi untuk opsi yang dapat Anda tentukan untuk konektor Amazon Redshift Python. Opsi di bawah ini berlaku untuk versi konektor terbaru yang tersedia kecuali ditentukan lain.

`access_key_id`

- Nilai default - Tidak ada
- Tipe data - String

Kunci akses untuk peran IAM atau pengguna yang dikonfigurasi untuk otentikasi database IAM.

Parameter ini bersifat opsional.

`allow_db_user_override`

- Nilai default - Salah
- Tipe data — Boolean

True

Menentukan bahwa konektor menggunakan `DbUser` nilai dari pernyataan Security Assertion Markup Language (SAMP).

False

Menentukan bahwa nilai dalam parameter DbUser koneksi digunakan.

Parameter ini bersifat opsional.

app_name

- Nilai default - Tidak ada
- Tipe data - String

Nama aplikasi penyedia identitas (IDP) yang digunakan untuk otentikasi.

Parameter ini bersifat opsional.

application_name

- Nilai default - Tidak ada
- Tipe data - String

Nama aplikasi klien untuk diteruskan ke Amazon Redshift untuk tujuan audit. [Nama aplikasi yang Anda berikan muncul di kolom 'application_name' dari tabel SYS_CONNECTION_LOG](#). Ini membantu melacak dan memecahkan masalah sumber koneksi saat men-debug masalah.

Parameter ini bersifat opsional.

auth_profil

- Nilai default - Tidak ada
- Tipe data - String

Nama profil otentikasi Amazon Redshift yang memiliki properti koneksi sebagai JSON. Untuk informasi selengkapnya tentang penamaan parameter koneksi, lihat `RedshiftProperty` kelas. `RedshiftPropertyKelas` menyimpan parameter koneksi yang disediakan oleh pengguna akhir dan, jika berlaku, dihasilkan selama proses otentikasi IAM (misalnya, kredensial IAM sementara). Untuk informasi lebih lanjut, lihat [RedshiftProperty kelas](#).

Parameter ini bersifat opsional.

auto_create

- Nilai default - Salah
- Tipe data — Boolean

Nilai yang menunjukkan apakah akan membuat pengguna jika pengguna tidak ada.

Parameter ini bersifat opsional.

client_id

- Nilai default - Tidak ada
- Tipe data - String

ID klien dari Azure iDP.

Parameter ini bersifat opsional.

client_secret

- Nilai default - Tidak ada
- Tipe data - String

Rahasia klien dari Azure iDP.

Parameter ini bersifat opsional.

cluster_identifier

- Nilai default - Tidak ada
- Tipe data - String

Pengidentifikasi cluster dari cluster Amazon Redshift.

Parameter ini bersifat opsional.

credentials_provider

- Nilai default - Tidak ada

- Tipe data - String

IDP yang digunakan untuk otentikasi dengan Amazon Redshift. Berikut ini adalah nilai yang valid:

- AdfsCredentialsProvider
- AzureCredentialsProvider
- BrowserAzureCredentialsProvider
- BrowserAzureOAuth2CredentialsProvider
- BrowserIdcAuthPlugin- Plugin otorisasi menggunakan AWS IAM Identity Center.
- BrowserSamlCredentialsProvider
- IdpTokenAuthPlugin- Plugin otorisasi yang menerima token AWS IAM Identity Center atau OpenID Connect (OIDC) JSON berbasis token identitas (JWT) dari penyedia identitas web mana pun yang terhubung ke Pusat Identitas IAM. AWS
- PingCredentialsProvider
- OktaCredentialsProvider

Parameter ini bersifat opsional.

basis data

- Nilai default - Tidak ada
- Tipe data - String

Nama database yang ingin Anda sambungkan.

Parameter ini diperlukan.

database_metadata_current_db_only

- Nilai default - Benar
- Tipe data — Boolean

Nilai yang menunjukkan apakah aplikasi mendukung katalog datashare multidatabase. Nilai default True menunjukkan bahwa aplikasi tidak mendukung katalog datashare multidatabase untuk kompatibilitas mundur.

Parameter ini bersifat opsional.

db_group

- Nilai default - Tidak ada
- Tipe data - String

Daftar dipisahkan koma dari nama grup database yang ada yang ditunjukkan oleh pengguna dengan DbUser bergabung untuk sesi saat ini.

Parameter ini bersifat opsional.

db_pengguna

- Nilai default - Tidak ada
- Tipe data - String

ID pengguna untuk digunakan dengan Amazon Redshift.

Parameter ini bersifat opsional.

endpoint_url

- Nilai default - Tidak ada
- Tipe data - String

URL titik akhir Amazon Redshift. Opsi ini hanya untuk penggunaan AWS internal.

Parameter ini bersifat opsional.

group_federasi

- Nilai default - Salah
- Tipe data — Boolean

Opsi ini menentukan apakah akan menggunakan grup IDP Amazon Redshift.

Parameter ini bersifat opsional.

true

Gunakan grup Amazon Redshift Identity Provider (IDP).

false

Gunakan STS API dan GetClusterCredentials untuk federasi pengguna dan tentukan db_groups untuk koneksi.

host

- Nilai default - Tidak ada
- Tipe data - String

Nama host dari cluster Amazon Redshift.

Parameter ini bersifat opsional.

iam

- Nilai default - Salah
- Tipe data — Boolean

Autentikasi IAM diaktifkan.

Parameter ini diperlukan.

iam_disable_cache

- Nilai default - Salah
- Tipe data — Boolean

Opsi ini menentukan apakah kredensial IAM di-cache. Secara default, kredensial IAM di-cache. Ini meningkatkan kinerja saat permintaan ke gateway API dibatasi.

Parameter ini bersifat opsional.

idc_client_display_name

- Nilai Default - Konektor Amazon Redshift Python

- Tipe Data - String

Nama tampilan yang akan digunakan untuk klien yang menggunakan BrowserIdcAuthPlugin.

Parameter ini bersifat opsional.

`idc_region`

- Nilai Default - Tidak Ada
- Tipe Data - String

AWS Wilayah di mana instans AWS IAM Identity Center berada.

Parameter ini diperlukan hanya ketika mengautentikasi menggunakan BrowserIdcAuthPlugin dalam opsi konfigurasi `credentials_provider`.

`idp_partisi`

- Nilai Default - Tidak Ada
- Tipe Data - String

Menentukan partisi cloud tempat penyedia identitas Anda (iDP) dikonfigurasi. Ini menentukan titik akhir otentikasi IDP mana yang terhubung ke driver.

Jika parameter ini dibiarkan kosong, driver default ke partisi komersial. Kemungkinan nilainya adalah:

- `us-gov`: Gunakan nilai ini jika IDP Anda dikonfigurasi di Azure Government. Misalnya, Azure AD Government menggunakan `login.microsoftonline.us` endpoint.
- `cn`: Gunakan nilai ini jika IDP Anda dikonfigurasi di partisi cloud Tiongkok. Misalnya, Azure AD Tiongkok menggunakan titik akhir `login.chinacloudapi.cn`

Parameter ini bersifat opsional.

`IDPport`

- Nilai default - 7890
- Tipe data - Integer

Port mendengarkan tempat iDP mengirimkan pernyataan SAMP.

Parameter ini diperlukan.

`idp_response_timeout`

- Nilai default - 120
- Tipe data - Integer

Batas waktu untuk mengambil pernyataan SAMP dari iDP.

Parameter ini diperlukan.

`idp_penyewa`

- Nilai default - Tidak ada
- Tipe data - String

Penyewa iDP.

Parameter ini bersifat opsional.

`issuer_url`

- Nilai Default - Tidak Ada
- Tipe Data - String

Menunjuk ke titik akhir instance server AWS IAM Identity Center.

Parameter ini diperlukan hanya ketika mengautentikasi menggunakan `BrowserIdcAuthPlugin` dalam opsi konfigurasi `credentials_provider`.

`listen_port`

- Nilai default - 7890
- Tipe data - Integer

Port yang digunakan driver untuk menerima respons SAFL dari penyedia identitas atau kode otorisasi saat menggunakan layanan SAFL, Azure AD, atau AWS IAM Identity Center melalui plugin browser.

Parameter ini bersifat opsional.

`login_url`

- Nilai default - Tidak ada
- Tipe data - String

Url masuk tunggal untuk iDP.

Parameter ini bersifat opsional.

`max_prepared_statement`

- Nilai default - 1000
- Tipe data - Integer

Jumlah maksimum pernyataan disiapkan yang akan di-cache per koneksi. Menyetel parameter ini ke 0 menonaktifkan mekanisme caching. Memasukkan angka negatif untuk parameter ini menetapkan ke nilai default.

Parameter ini bersifat opsional.

`numeric_to_float`

- Nilai default - Salah
- Tipe data — Boolean

Opsi ini menentukan apakah konektor mengkonversi nilai tipe data numerik dari Decimal.Decimal untuk float. Secara default, konektor menerima nilai tipe data numerik sebagai desimal. Desimal dan tidak mengonversinya.

Kami tidak menyarankan mengaktifkan `numeric_to_float` untuk kasus penggunaan yang memerlukan presisi, karena hasilnya dapat dibulatkan.

Untuk informasi lebih lanjut tentang desimal. Desimal dan pengorbanan antara itu dan float, lihat desimal [— Titik tetap desimal](#) dan aritmatika floating point di situs web Python.

Parameter ini bersifat opsional.

partner_sp_id

- Nilai default - Tidak ada
- Tipe data - String

ID SP Mitra yang digunakan untuk otentikasi dengan Ping.

Parameter ini bersifat opsional.

password

- Nilai default - Tidak ada
- Tipe data - String

Kata sandi yang digunakan untuk otentikasi.

Parameter ini bersifat opsional.

port

- Nilai default - 5439
- Tipe data - Integer

Nomor port cluster Amazon Redshift.

Parameter ini diperlukan.

preferred_role

- Nilai default - Tidak ada
- Tipe data - String

Peran IAM lebih disukai untuk koneksi saat ini.

Parameter ini bersifat opsional.

principal_arn

- Nilai default - Tidak ada

- Tipe data - String

Nama Sumber Daya Amazon (ARN) dari pengguna atau peran IAM tempat Anda membuat kebijakan. Anda disarankan untuk melampirkan kebijakan ke peran dan kemudian menetapkan peran tersebut ke pengguna Anda, untuk akses.

Parameter ini bersifat opsional.

profile

- Nilai default - Tidak ada
- Tipe data - String

Nama profil dalam file kredensial yang berisi AWS AWS kredensial.

Parameter ini bersifat opsional.

provider_name

- Nilai default - Tidak ada
- Tipe data - String

Nama Penyedia Otentikasi Asli Redshift.

Parameter ini bersifat opsional.

region

- Nilai default - Tidak ada
- Tipe data - String

Di AWS Region mana cluster berada.

Parameter ini bersifat opsional.

role_arn

- Nilai default - Tidak ada
- Tipe data - String

Nama Sumber Daya Amazon (ARN) dari peran yang diasumsikan oleh penelepon. Parameter ini digunakan oleh penyedia yang ditunjukkan oleh `JwtCredentialsProvider`.

Untuk `JwtCredentialsProvider` penyedia, parameter ini wajib. Jika tidak, parameter ini opsional.

`role_session_name`

- Nilai default - `jwt_redshift_session`
- Tipe data - String

Sebuah pengenalan untuk sesi peran yang diambil. Biasanya, Anda meneruskan nama atau pengenalan yang terkait dengan pengguna yang menggunakan aplikasi Anda. Kredensial keamanan sementara yang digunakan aplikasi Anda terkait dengan pengguna tersebut. Parameter ini digunakan oleh penyedia yang ditunjukkan oleh `JwtCredentialsProvider`.

Parameter ini bersifat opsional.

`cakupan`

- Nilai default - Tidak ada
- Tipe data - String

Daftar cakupan yang dipisahkan ruang yang dapat disetujui pengguna. Anda menentukan parameter ini sehingga aplikasi Anda bisa mendapatkan persetujuan untuk APIs yang ingin Anda panggil. Anda dapat menentukan parameter ini ketika Anda menentukan `BrowserAzure OAuth2 CredentialsProvider` untuk opsi `credentials_provider`.

Parameter ini diperlukan untuk `BrowserAzure OAuth2 CredentialsProvider` plug-in.

`secret_access_key_id`

- Nilai default - Tidak ada
- Tipe data - String

Kunci akses rahasia untuk peran IAM atau pengguna yang dikonfigurasi untuk otentikasi database IAM.

Parameter ini bersifat opsional.

`session_token`

- Nilai default - Tidak ada
- Tipe data - String

Kunci akses untuk peran IAM atau pengguna yang dikonfigurasi untuk otentikasi database IAM. Parameter ini diperlukan jika AWS kredensyal sementara sedang digunakan.

Parameter ini bersifat opsional.

`serverless_acct_id`

- Nilai default - Tidak ada
- Tipe data - String

ID akun Tanpa Server Amazon Redshift.

Parameter ini bersifat opsional.

`serverless_work_group`

- Nilai default - Tidak ada
- Tipe data - String

Nama workgroup Amazon Redshift Tanpa Server.

Parameter ini bersifat opsional.

`ssl`

- Nilai default - Benar
- Tipe data — Boolean

Secure Sockets Layer (SSL) diaktifkan.

Parameter ini diperlukan.

`ssl_insecure`

- Nilai default - Salah

- Tipe data — Boolean

Nilai yang menentukan apakah akan menonaktifkan verifikasi sertifikat SSL server idP host.

Menyetel parameter ini ke True akan menonaktifkan verifikasi sertifikat SSL server host iDP. Kami menyarankan Anda menyimpan nilai default False di lingkungan produksi.

Parameter ini bersifat opsional.

sslmode

- Nilai default - verify-ca
- Tipe data - String

Keamanan koneksi ke Amazon Redshift. Anda dapat menentukan salah satu dari berikut ini:

- verify-ca
- verify-full

Parameter ini diperlukan.

tcp_keepalive

- Nilai default - Benar
- Tipe data — Boolean

Apakah akan menggunakan TCP keepalives untuk menjaga koneksi dari waktu habis. Anda dapat menentukan salah satu nilai berikut:

- Benar: Driver akan menggunakan TCP keepalives untuk menjaga koneksi dari waktu habis.
- Salah: Driver tidak akan menggunakan TCP keepalives.

Parameter ini bersifat opsional.

tcp_keepalive_count

- Nilai default - Tidak ada
- Tipe data - Integer

Jumlah probe yang tidak diakui untuk dikirim sebelum mempertimbangkan koneksi tidak aktif. Misalnya, menetapkan nilai ke 3 berarti bahwa driver akan mengirim 3 paket keepalive yang belum dijawab sebelum menentukan bahwa koneksi tidak lagi aktif.

Jika parameter ini tidak ditentukan, Amazon Redshift menggunakan nilai default sistem.

Parameter ini bersifat opsional.

`tcp_keepalive_interval`

- Nilai default - Tidak ada
- Tipe data - Integer

Interval, dalam hitungan detik, antara probe keepalive berikutnya jika pengemudi tidak menerima pengakuan untuk probe sebelumnya. Jika Anda menentukan parameter ini, itu harus bilangan bulat positif.

Jika parameter ini tidak ditentukan, Amazon Redshift menggunakan nilai default sistem.

Parameter ini bersifat opsional.

`tcp_keepalive_idle`

- Nilai default - Tidak ada
- Tipe data - Integer

Durasi tidak aktif, dalam hitungan detik, setelah itu pengemudi mengirimkan probe keepalive pertama. Misalnya, menetapkan nilai ke 120 berarti bahwa driver akan menunggu selama 2 menit tidak aktif sebelum mengirim paket keepalive pertama. Jika Anda menentukan parameter ini, itu harus bilangan bulat positif.

Jika parameter ini tidak ditentukan, Amazon Redshift menggunakan nilai default sistem.

Parameter ini bersifat opsional.

`batas waktu`

- Nilai default - Tidak ada
- Tipe data - Integer

Jumlah detik sebelum koneksi ke server habis.

Parameter ini bersifat opsional.

token

- Nilai Default - Tidak Ada
- Tipe Data - String

Pusat AWS Identitas IAM menyediakan token akses atau OpenID Connect (OIDC) JSON Web Token (JWT) yang disediakan oleh penyedia identitas web yang terhubung dengan IAM Identity Center. AWS Aplikasi Anda harus menghasilkan token ini dengan mengautentikasi pengguna aplikasi Anda dengan AWS IAM Identity Center atau penyedia identitas yang ditautkan dengan AWS IAM Identity Center.

Parameter ini bekerja dengan `IdpTokenAuthPlugin`.

token_type

- Nilai Default - Tidak Ada
- Tipe Data - String

Jenis token yang digunakan di `IdpTokenAuthPlugin`.

Anda dapat menentukan salah satu nilai berikut:

`ACCESS_TOKEN`

Masukkan ini jika Anda menggunakan token akses yang disediakan Pusat Identitas AWS IAM.

`EXT_JWT`

Masukkan ini jika Anda menggunakan OpenID Connect (OIDC) JSON Web Token (JWT) yang disediakan oleh penyedia identitas berbasis web yang terintegrasi dengan IAM Identity Center.
AWS

Parameter ini bekerja dengan `IdpTokenAuthPlugin`.

user

- Nilai default - Tidak ada

- Tipe data - String

Nama pengguna yang akan digunakan untuk otentikasi.

Parameter ini bersifat opsional.

`web_identity_token`

- Nilai default - Tidak ada
- Tipe data - String

Token akses OAuth 2.0 atau token OpenID Connect ID yang disediakan oleh penyedia identitas. Pastikan aplikasi Anda mendapatkan token ini dengan mengautentikasi pengguna yang menggunakan aplikasi Anda dengan penyedia identitas web. Penyedia yang ditunjukkan dengan `JwtCredentialsProvider` menggunakan parameter ini.

Untuk `JwtCredentialsProvider` penyedia, parameter ini wajib. Jika tidak, parameter ini opsional.

Mengimpor konektor Python

Untuk mengimpor konektor Python, jalankan perintah berikut.

```
>>> import redshift_connector
```

Untuk menyambung ke klaster Amazon Redshift menggunakan AWS kredensial, jalankan perintah berikut.

```
conn = redshift_connector.connect(  
    host='examplecluster.abc123xyz789.us-west-1.redshift.amazonaws.com',  
    port=5439,  
    database='dev',  
    user='awsuser',  
    password='my_password'  
)
```

Mengintegrasikan konektor Python dengan NumPy

Berikut ini adalah contoh mengintegrasikan konektor Python dengan NumPy

```
>>> 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)
```

Berikut ini adalah hasilnya.

```
[[ 'One Hundred Years of Solitude' 'Gabriel García Márquez']
[ 'A Brief History of Time' 'Stephen Hawking']]
```

Mengintegrasikan konektor Python dengan panda

Berikut ini adalah contoh mengintegrasikan konektor Python dengan panda.

```
>>> 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)
```

Menggunakan plugin penyedia identitas

Untuk informasi umum tentang cara menggunakan plugin penyedia identitas, lihat [Opsinya untuk memberikan kredensi IAM](#). Untuk informasi selengkapnya tentang mengelola identitas IAM, termasuk praktik terbaik untuk peran IAM, lihat [Manajemen identitas dan akses di Amazon Redshift](#)

Otentikasi menggunakan plugin penyedia identitas ADFS

Berikut ini adalah contoh penggunaan plugin penyedia identitas Active Directory Federation Service (ADFS) untuk mengautentikasi pengguna yang terhubung ke database Amazon Redshift.

```
>>> 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'
)
```

Otentikasi menggunakan plugin penyedia identitas Azure

Berikut ini adalah contoh otentikasi menggunakan plugin penyedia identitas Azure. Anda dapat membuat nilai untuk `client_id` dan `client_secret` untuk aplikasi Azure Enterprise seperti yang ditunjukkan berikut.

```
>>> 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'
)

```

Otentikasi menggunakan plugin penyedia identitas Pusat AWS Identitas IAM

Berikut ini adalah contoh otentikasi menggunakan plugin penyedia identitas AWS IAM Identity Center.

```

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
)

```

Otentikasi menggunakan plugin penyedia identitas Azure Browser

Berikut ini adalah contoh penggunaan plugin penyedia identitas Azure Browser untuk mengautentikasi pengguna yang terhubung ke database Amazon Redshift.

Otentikasi multi-faktor terjadi di browser, di mana kredensi masuk disediakan oleh pengguna.

```

>>>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',
)

```

Otentikasi menggunakan plugin penyedia identitas Okta

Berikut ini adalah contoh otentikasi menggunakan plugin penyedia identitas Okta. Anda dapat memperoleh nilai untuk `idp_host`, `app_id` dan `app_name` melalui aplikasi Okta.

```

>>> 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'  
)
```

Otentikasi menggunakan JumpCloud dengan plugin penyedia identitas browser SAMP generik

Berikut ini adalah contoh penggunaan JumpCloud dengan plugin penyedia identitas browser SAMP generik untuk otentikasi.

Parameter kata sandi diperlukan. Namun, Anda tidak perlu memasukkan parameter ini karena otentikasi multi-faktor terjadi di browser.

```
>>> 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'  
)
```

Contoh penggunaan konektor Amazon Redshift Python

Berikut ini adalah contoh cara menggunakan konektor Amazon Redshift Python. Untuk menjalankannya, Anda harus menginstal konektor Python terlebih dahulu. Untuk informasi selengkapnya tentang menginstal konektor Amazon Redshift Python, lihat. [Memasang konektor Amazon Redshift Python](#) Untuk informasi lebih lanjut tentang opsi konfigurasi yang dapat Anda gunakan dengan konektor Python, lihat. [Opsi konfigurasi untuk konektor Amazon Redshift Python](#)

Topik

- [Menyambung ke dan menanyakan klaster Amazon Redshift menggunakan kredensial AWS](#)

- [Mengaktifkan komit otomatis](#)
- [Mengkonfigurasi paramstyle kursor](#)
- [Menggunakan COPY untuk menyalin data dari bucket Amazon S3 dan UNLOAD untuk menulis data ke dalamnya](#)

Menyambung ke dan menanyakan klaster Amazon Redshift menggunakan kredensial AWS

Contoh berikut memandu Anda untuk menghubungkan ke klaster Amazon Redshift menggunakan AWS kredensial Anda, lalu menanyakan tabel dan mengambil hasil kueri.

```
#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'])
```

Mengaktifkan komit otomatis

Properti autocommit dinonaktifkan secara default, mengikuti Spesifikasi API Database Python. Anda dapat menggunakan perintah berikut untuk mengaktifkan properti autocommit koneksi setelah melakukan perintah rollback untuk memastikan bahwa transaksi tidak sedang berlangsung.

```
#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
```

Mengkonfigurasi paramstyle kursor

Paramstyle untuk kursor dapat dimodifikasi melalui `cursor.paramstyle`. Paramstyle default yang digunakan adalah `format`. Nilai yang valid untuk paramstyle adalah `qmark`, `numeric`, `namedformat`, dan `pyformat`.

Berikut ini adalah contoh menggunakan berbagai paramstyles untuk meneruskan parameter ke pernyataan SQL sampel.

```
# 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)'\ncursor.execute(sql, {"bar": 1, "jar": "hello world"})
```

Menggunakan COPY untuk menyalin data dari bucket Amazon S3 dan UNLOAD untuk menulis data ke dalamnya

Contoh berikut menunjukkan cara menyalin data dari bucket Amazon S3 ke dalam tabel dan kemudian membongkar dari tabel itu kembali ke bucket.

File teks bernama `category_csv.txt` berisi data berikut diunggah ke bucket Amazon S3:

```
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"
```

Berikut ini adalah contoh kode Python, yang pertama kali terhubung ke database Amazon Redshift. Kemudian membuat tabel yang disebut `category` dan menyalin data CSV dari bucket S3 ke dalam tabel.

```
#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'])
```

Jika Anda belum `autocommit` menyetel ke `true`, komit dengan `conn.commit()` setelah menjalankan `execute()` pernyataan.

Data diturunkan ke file `unloaded_category_csv.text0000_part00` di bucket S3, dengan konten berikut:

```
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"
```

Referensi API untuk konektor Amazon Redshift Python

Berikut ini, Anda dapat menemukan deskripsi operasi API konektor Amazon Redshift Python.

`redshift_connector`

Berikut ini, Anda dapat menemukan deskripsi operasi `redshift_connector` API.

`connect(user, database, password[, port, ...])`

Membuat koneksi ke cluster Amazon Redshift. Fungsi ini memvalidasi input pengguna, secara opsional mengautentikasi menggunakan plugin penyedia identitas, dan kemudian membangun objek koneksi.

`apilevel`

Level DBAPI didukung, saat ini “2.0”.

`paramstyle, str(object='') -> str str(bytes_or_buffer[, encoding[, errors]])`
`-> str`

Gaya parameter API database untuk digunakan secara global.

Koneksi

Berikut ini, Anda dapat menemukan deskripsi operasi API koneksi untuk konektor Amazon Redshift Python.

```
__init__(user, password, database[, host, ...])
```

Menginisialisasi objek koneksi mentah.

```
cursor
```

Membuat objek kursor terikat pada koneksi ini.

```
commit
```

Melakukan transaksi database saat ini.

```
rollback
```

Menggulung kembali transaksi database saat ini.

```
close
```

Menutup koneksi database.

```
execute(cursor, operation, vals)
```

Menjalankan perintah SQL yang ditentukan. Anda dapat memberikan parameter sebagai urutan atau sebagai pemetaan, tergantung pada nilai `redshift_connector.paramstyle`

```
run(sql[, stream])
```

Menjalankan perintah SQL yang ditentukan. Secara opsional, Anda dapat menyediakan aliran untuk digunakan dengan perintah COPY.

```
xid(format_id, global_transaction_id, ...)
```

Buat ID transaksi. Hanya `global_transaction_id` parameter yang digunakan di postgres. `format_id` dan `branch_qualifier` tidak digunakan di postgres. `global_transaction_id` Dapat berupa pengidentifikasi string apa pun yang didukung oleh postgres yang mengembalikan tuple `(format_id,,global_transaction_id).branch_qualifier`

```
tpc_begin(xid)
```

Memulai transaksi TPC dengan ID transaksi `xid` yang terdiri dari ID format, ID transaksi global, dan kualifikasi cabang.

```
tpc_prepare
```

Melakukan fase pertama transaksi yang dimulai dengan `tpc_begin`.

`tpc_commit([xid])`

Ketika dipanggil tanpa argumen, `.tpc_commit` melakukan transaksi TPC yang sebelumnya disiapkan dengan `.tpc_prepare ()`.

`tpc_rollback([xid])`

Ketika dipanggil tanpa argumen, `.tpc_rollback` mengembalikan transaksi TPC.

`tpc_recover`

Mengembalikan daftar transaksi tertunda yang IDs cocok untuk digunakan dengan `.tpc_commit (xid)` atau `.tpc_rollback (xid)`.

Kursor

Berikut ini, Anda dapat menemukan deskripsi operasi API kursor.

`__init__(connection[, paramstyle])`

Menginisialisasi objek kursor mentah.

`insert_data_bulk(filename, table_name, parameter_indices, column_names, delimiter, batch_size)`

Menjalankan pernyataan INSERT massal.

`execute(operation[, args, stream, ...])`

Menjalankan operasi database.

`executemany(operation, param_sets)`

Mempersiapkan operasi database, dan kemudian menjalankannya untuk semua urutan parameter atau pemetaan yang disediakan.

`fetchone`

Mengambil baris berikutnya dari hasil query set.

`fetchmany([num])`

Mengambil set baris berikutnya dari hasil query.

`fetchall`

Mengambil semua baris yang tersisa dari hasil query.

close

Tutup kursor sekarang.

__iter__

Sebuah objek kursor dapat diiterasi untuk mengambil baris dari query.

fetch_dataframe([num])

Mengembalikan kerangka data dari hasil query terakhir.

write_dataframe(df, table)

Menulis kerangka data struktur yang sama ke dalam database Amazon Redshift.

fetch_numpy_array([num])

Mengembalikan NumPy array hasil query terakhir.

get_catalogs

Amazon Redshift tidak mendukung beberapa katalog dari satu koneksi. Amazon Redshift hanya mengembalikan katalog saat ini.

get_tables([catalog, schema_pattern, ...])

Mengembalikan tabel publik unik yang ditentukan pengguna dalam sistem.

get_columns([catalog, schema_pattern, ...])

Mengembalikan daftar semua kolom dalam tabel tertentu dalam database Amazon Redshift.

AdfsCredentialsProvider plugin

Berikut ini adalah sintaks untuk operasi API AdfsCredentialsProvider plugin untuk konektor Amazon Redshift Python.

```
redshift_connector.plugin.AdfsCredentialsProvider()
```

AzureCredentialsProvider plugin

Berikut ini adalah sintaks untuk operasi API AzureCredentialsProvider plugin untuk konektor Amazon Redshift Python.

```
redshift_connector.plugin.AzureCredentialsProvider()
```


BrowserAzureCredentialsProvider plugin

Berikut ini adalah sintaks untuk operasi API BrowserAzureCredentialsProvider plugin untuk konektor Amazon Redshift Python.

```
redshift_connector.plugin.BrowserAzureCredentialsProvider()
```

BrowserSamlCredentialsProvider plugin

Berikut ini adalah sintaks untuk operasi API BrowserSamlCredentialsProvider plugin untuk konektor Amazon Redshift Python.

```
redshift_connector.plugin.BrowserSamlCredentialsProvider()
```

OktaCredentialsProvider plugin

Berikut ini adalah sintaks untuk operasi API OktaCredentialsProvider plugin untuk konektor Amazon Redshift Python.

```
redshift_connector.plugin.OktaCredentialsProvider()
```

PingCredentialsProvider plugin

Berikut ini adalah sintaks untuk operasi API PingCredentialsProvider plugin untuk konektor Amazon Redshift Python.

```
redshift_connector.plugin.PingCredentialsProvider()
```

SamlCredentialsProvider plugin

Berikut ini adalah sintaks untuk operasi API SamlCredentialsProvider plugin untuk konektor Amazon Redshift Python.

```
redshift_connector.plugin.SamlCredentialsProvider()
```

Integrasi Amazon Redshift untuk Apache Spark

[Apache Spark](#) adalah kerangka pemrosesan terdistribusi dan model pemrograman yang membantu Anda melakukan pembelajaran mesin, pemrosesan aliran, atau analisis grafik. Mirip dengan Apache Hadoop, Spark merupakan sistem pemrosesan terdistribusi sumber terbuka, yang biasa digunakan untuk beban kerja big data. Spark memiliki mesin eksekusi grafik asiklik terarah (DAG) yang dioptimalkan dan secara aktif menyimpan data dalam memori. Ini dapat meningkatkan kinerja, terutama untuk algoritma tertentu dan kueri interaktif.

Integrasi ini memberi Anda konektor Spark yang dapat Anda gunakan untuk membangun aplikasi Apache Spark yang membaca dan menulis ke data di Amazon Redshift dan Amazon Redshift Serverless. Aplikasi ini tidak berkompromi pada kinerja aplikasi atau konsistensi transaksional data. Integrasi ini secara otomatis disertakan dalam [Amazon EMR](#) dan [AWS Glue](#), sehingga Anda dapat segera menjalankan pekerjaan Apache Spark yang mengakses dan memuat data ke Amazon Redshift sebagai bagian dari pipa penyerapan dan transformasi data Anda.

Saat ini, Anda dapat menggunakan versi 3.3.x, 3.4.x, 3.5.x, dan 4.0.0 dari Spark dengan integrasi ini.

Integrasi ini menyediakan yang berikut:

- AWS Identity and Access Management (IAM) otentikasi. Untuk informasi selengkapnya, lihat [Manajemen identitas dan akses di Amazon Redshift](#).
- Predikat dan kueri pushdown untuk meningkatkan kinerja.
- Jenis data Amazon Redshift.
- Konektivitas ke Amazon Redshift dan Amazon Redshift Tanpa Server.

Pertimbangan dan batasan saat menggunakan konektor Spark

- URI tempdir menunjuk ke lokasi Amazon S3. Direktori temp ini tidak dibersihkan secara otomatis dan dapat menambah biaya tambahan. Sebaiknya gunakan [kebijakan siklus hidup Amazon S3 di](#) Panduan Pengguna Layanan Penyimpanan Sederhana Amazon untuk menentukan aturan penyimpanan bucket Amazon S3.
- Secara default, salinan antara Amazon S3 dan Redshift tidak berfungsi jika bucket S3 dan cluster Redshift berada di Wilayah yang berbeda. AWS Untuk menggunakan AWS Regions terpisah, atur tempdir_region parameter ke Region bucket S3 yang digunakan untuk tempdir

- Cross-Region menulis antara S3 dan Redshift jika menulis data Parquet menggunakan parameter `tempformat`
- Sebaiknya gunakan [enkripsi sisi server Amazon S3 untuk mengenkripsi bucket](#) Amazon S3 yang digunakan.
- Kami merekomendasikan untuk [memblokir akses publik ke bucket Amazon S3](#).
- Kami menyarankan agar cluster Amazon Redshift tidak dapat diakses publik.
- Sebaiknya aktifkan pencatatan [audit Amazon Redshift](#).
- Sebaiknya aktifkan enkripsi saat [istirahat Amazon Redshift](#).
- Sebaiknya aktifkan SSL untuk koneksi JDBC dari Spark di Amazon EMR ke Amazon Redshift.
- Kami merekomendasikan untuk meneruskan peran IAM menggunakan parameter `aws_iam_role` untuk parameter autentikasi Amazon Redshift.

Otentikasi dengan konektor Spark

Diagram berikut menjelaskan otentikasi antara Amazon S3, Amazon Redshift, driver Spark, dan pelaksana Spark.

Otentikasi antara Redshift dan Spark

Anda dapat menggunakan driver versi 2.x driver JDBC yang disediakan Amazon Redshift untuk terhubung ke Amazon Redshift dengan konektor Spark dengan menentukan kredensial masuk. Untuk menggunakan IAM, [konfigurasi url JDBC Anda untuk menggunakan](#) otentikasi IAM. Untuk menyambung ke kluster Redshift dari Amazon EMR atau AWS Glue, pastikan bahwa peran IAM Anda memiliki izin yang diperlukan untuk mengambil kredensial IAM sementara. Daftar berikut menjelaskan semua izin yang diperlukan peran IAM Anda untuk mengambil kredensial dan menjalankan operasi Amazon S3.

- [Redshift: GetClusterCredentials](#) (untuk cluster Redshift yang disediakan)
- [Redshift: DescribeClusters](#) (untuk cluster Redshift yang disediakan)
- [Redshift: GetWorkgroup](#) (untuk grup kerja Amazon Redshift Tanpa Server)
- [Redshift: GetCredentials](#) (untuk grup kerja Amazon Redshift Tanpa Server)
- [s3: ListBucket](#)
- [s3: GetBucket](#)
- [s3: GetObject](#)

- [s3: PutObject](#)
- [s3: GetBucketLifecycleConfiguration](#)

Untuk informasi selengkapnya GetClusterCredentials, lihat [kebijakan IAM untuk GetClusterCredentials](#).

Anda juga harus memastikan bahwa Amazon Redshift dapat mengambil peran IAM selama COPY dan operasi. UNLOAD

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "redshift.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Jika Anda menggunakan driver JDBC terbaru, driver akan secara otomatis mengelola transisi dari sertifikat yang ditandatangani sendiri Amazon Redshift ke sertifikat ACM. Namun, Anda harus [menentukan opsi SSL ke url JDBC](#).

Berikut ini adalah contoh cara menentukan URL driver JDBC dan terhubung `aws_iam_role` ke Amazon Redshift.

```
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()
```

Otentikasi antara Amazon S3 dan Spark

Jika Anda menggunakan peran IAM untuk mengautentikasi antara Spark dan Amazon S3, gunakan salah satu metode berikut:

- AWS SDK for Java akan secara otomatis mencoba AWS menemukan kredensial dengan menggunakan rantai penyedia kredensi default yang diimplementasikan oleh kelas DefaultAWSCredentials ProviderChain Untuk informasi selengkapnya, lihat [Menggunakan Rantai Penyedia Kredensial Default](#).
- Anda dapat menentukan AWS kunci melalui properti [konfigurasi Hadoop](#). Misalnya, jika tempdir konfigurasi Anda menunjuk ke s3n:// sistem file, setel fs.s3n.awsAccessKeyId dan fs.s3n.awsSecretAccessKey properti dalam file konfigurasi XMLHadoop atau panggil sc.hadoopConfiguration.set() untuk mengubah konfigurasi Hadoop global Spark.

Misalnya, jika Anda menggunakan sistem file s3n, tambahkan:

```
sc.hadoopConfiguration.set("fs.s3n.awsAccessKeyId", "YOUR_KEY_ID")
sc.hadoopConfiguration.set("fs.s3n.awsSecretAccessKey", "YOUR_SECRET_ACCESS_KEY")
```

Untuk sistem file s3a, tambahkan:

```
sc.hadoopConfiguration.set("fs.s3a.access.key", "YOUR_KEY_ID")
sc.hadoopConfiguration.set("fs.s3a.secret.key", "YOUR_SECRET_ACCESS_KEY")
```

Jika Anda menggunakan Python, gunakan operasi berikut:

```
sc._jsc.hadoopConfiguration().set("fs.s3n.awsAccessKeyId", "YOUR_KEY_ID")
sc._jsc.hadoopConfiguration().set("fs.s3n.awsSecretAccessKey",
    "YOUR_SECRET_ACCESS_KEY")
```

- Encode kunci otentikasi di URL. tempdir Misalnya, URI s3n:// ACCESSKEY:SECRETKEY@bucket/path/to/temp/dir mengkodekan key pair (ACCESSKEY,SECRETKEY).

Otentikasi antara Redshift dan Amazon S3

Jika Anda menggunakan perintah COPY dan UNLOAD dalam kueri, Anda juga harus memberikan Amazon S3 akses ke Amazon Redshift untuk menjalankan kueri atas nama Anda. Untuk

melakukannya, pertama-tama [otorisasi Amazon Redshift untuk mengakses layanan AWS lain](#), lalu otorisasi operasi [COPY dan UNLOAD](#) menggunakan peran IAM.

Sebagai praktik terbaik, kami menyarankan untuk melampirkan kebijakan izin ke peran IAM dan kemudian menetapkan ke pengguna dan grup sesuai kebutuhan. Untuk informasi selengkapnya, lihat [Manajemen identitas dan akses di Amazon Redshift](#).

Integrasi dengan AWS Secrets Manager

Anda dapat mengambil nama pengguna Redshift dan kredensial kata sandi Anda dari rahasia yang disimpan di AWS Secrets Manager Untuk secara otomatis menyediakan kredensial Redshift, gunakan parameter `secret.id` Untuk informasi selengkapnya tentang cara membuat rahasia kredensial Redshift, lihat [Membuat](#) rahasia database. AWS Secrets Manager

GroupID	ArtifactID	Revisi yang Didukung	Deskripsi
com.amazonaws.secretsmanager	aws-secretsmanager-jdbc	1.0.12	AWS Secrets Manager SQL Connection Library for Java memungkinkan Pengembang Java untuk dengan mudah terhubung ke database SQL menggunakan rahasia yang disimpan di dalamnya. AWS Secrets Manager

Note

Pengakuan: Dokumentasi ini berisi kode contoh dan bahasa yang dikembangkan oleh [Apache Software Foundation](#) yang dilisensikan di bawah lisensi [Apache](#) 2.0.

Peningkatan kinerja dengan pushdown

Konektor Spark secara otomatis menerapkan predikat dan pushdown kueri untuk mengoptimalkan kinerja. Dukungan ini berarti bahwa jika Anda menggunakan fungsi yang didukung dalam kueri Anda, konektor Spark akan mengubah fungsi menjadi kueri SQL dan menjalankan kueri di Amazon Redshift. Optimalisasi ini menghasilkan lebih sedikit data yang diambil, sehingga Apache Spark dapat memproses lebih sedikit data dan memiliki kinerja yang lebih baik. Secara default, pushdown diaktifkan secara otomatis. Untuk menonaktifkannya, atur `autopushdown` ke `false`.

```
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()
```

Fungsi-fungsi berikut didukung dengan pushdown. Jika Anda menggunakan fungsi yang tidak ada dalam daftar ini, konektor Spark akan menjalankan fungsi di Spark alih-alih Amazon Redshift, menghasilkan kinerja yang tidak dioptimalkan. Untuk daftar lengkap fungsi di Spark, lihat [Fungsi bawaan](#).

- Fungsi agregasi
 - rata-rata
 - count
 - max
 - min
 - sum
 - stddev_samp
 - stddev_pop
 - var_samp
 - var_pop
- Operator Boolean
 - in
 - isnull

- isnotnull
- mengandung
- berakhir dengan
- mulaidengan
- Operator logis
 - and
 - atau
 - tidak (atau!)
- Fungsi matematika
 - +
 - -
 - *
 - /
 - - (unary)
 - perut
 - acos
 - asin
 - atan
 - ceil
 - cos
 - exp
 - lantai
 - terbesar
 - paling sedikit
 - log10
 - pi
 - pow
 - bulat
 - dosa
 - sqrt

- tan
- Fungsi lain-lain
 - melemparkan
 - coalesce
 - desimal
 - jika
 - in
- Operator relasional
 - !=
 - =
 - >
 - >=
 - <
 - <=
- Fungsi string
 - ascii
 - lpad
 - rpad
 - menerjemahkan
 - atas
 - menurunkan
 - length
 - memangkas
 - ltrim
 - rtrim
 - suka
 - substring
 - concat

- **Fungsi waktu dan tanggal**

Mengonfigurasi koneksi di Amazon Redshift

- add_months

- date
- date_add
- date_sub
- date_trunc
- timestamp
- batang
- Operasi matematika
 - CheckOverflow
 - PromotePrecision
- Operasi relasional
 - Alias (misalnya, AS)
 - CaseWhen
 - Berbeda
 - InSet
 - Bergabung dan bergabung silang
 - Batas
 - Serikat pekerja, serikat semua
 - ScalarSubquery
 - Sortir (naik dan turun)
 - UnscaledValue

Opsi konfigurasi lainnya

Di halaman ini, Anda dapat menemukan deskripsi untuk opsi yang dapat Anda tentukan untuk konektor Amazon Redshift Spark.

Ukuran maksimum kolom string

Redshift membuat kolom string sebagai kolom teks saat membuat tabel, yang disimpan sebagai VARCHAR (256). Jika Anda menginginkan kolom yang mendukung ukuran yang lebih besar, Anda dapat menggunakan maxlength untuk menentukan panjang maksimum kolom string. Berikut ini adalah contoh cara menentukanmaxlength.

```
columnLengthMap.foreach { case (colName, length) =>
```

```
val metadata = new MetadataBuilder().putLong("maxlength", length).build()
df = df.withColumn(colName, df(colName).as(colName, metadata))
}
```

Jenis kolom

Untuk mengatur jenis kolom, gunakan `redshift_type` bidang.

```
columnTypeMap.foreach { case (colName, colType) =>
  val metadata = new MetadataBuilder().putString("redshift_type", colType).build()
  df = df.withColumn(colName, df(colName).as(colName, metadata))
}
```

Pengkodean kompresi pada kolom

Untuk menggunakan pengkodean kompresi tertentu pada kolom, gunakan bidang pengkodean.

Untuk daftar lengkap penyandian kompresi dukungan, lihat Pengkodean [kompresi](#).

Deskripsi untuk kolom

Untuk mengatur deskripsi, gunakan `description` bidang.

Otentikasi antara Redshift dan Amazon S3

Secara default, hasilnya diturunkan ke Amazon S3 dalam format `parquet`. Untuk membongkar hasilnya sebagai file teks yang dibatasi pipa, tentukan opsi berikut.

```
.option("unload_s3_format", "TEXT")
```

Pernyataan pushdown

Parameter	Diperlukan	Default	Deskripsi
<code>spark.datasource.redshift.community.autoPushdown.lazyMode</code>	Tidak	True	Menentukan apakah konektor harus malas menjalankan pernyataan pushdown Redshift. Jika benar, konektor percikan mengambil

Parameter	Diperlukan	Default	Deskripsi
			semua model dan informasi terkait sebelum menjalankan kueri, yang umumnya menghasilkan kinerja yang lebih baik.
			Jika salah, konektor percikan menjalankan pernyataan pushdown segera di atas driver Spark utama dan diserialisasikan di seluruh ekspresi.

Parameter konektor

Peta parameter atau `OPTIONS` di Spark SQL mendukung pengaturan berikut.

Parameter	Diperlukan	Default	Deskripsi
<code>dbtable</code>	Ya, kecuali kueri ditentukan	N/A	Tabel untuk membuat atau membaca dari Redshift. Parameter ini diperlukan saat menyimpan data kembali ke Redshift.
<code>kueri</code>	Ya, kecuali <code>dbtable</code> ditentukan	N/A	Kueri untuk dibaca di Redshift.
<code>user</code>	Tidak	N/A	Nama pengguna Redshift. Harus

Parameter	Diperlukan	Default	Deskripsi
			digunakan dengan parameter kata sandi. Hanya berlaku jika pengguna dan kata sandi bukan parameter dalam URL. Menggunakan keduanya akan menyebabkan kesalahan.
password	Tidak	N/A	Kata sandi Redshift. Harus digunakan dengan parameter pengguna. Hanya berlaku jika pengguna dan kata sandi bukan parameter dalam URL. Menggunakan keduanya akan menyebabkan kesalahan.

Parameter	Diperlukan	Default	Deskripsi
url	Tidak	N/A	URL JDBC. Formatnya adalah jdbc:subprotocol: // host:port/database? user=username&password=password. Subprotokol dapat berupa postgresql atau Redshift, tergantung pada driver JDBC mana yang telah Anda muat. Perhatikan bahwa satu driver yang kompatibel dengan Redshift harus berada di classpath dan cocok dengan URL ini. Host dan port harus mengarah ke node master Redshift, jadi Anda harus mengonfigurasi grup keamanan and/or VPC untuk mengizinkan akses dari aplikasi driver Anda. Database adalah nama database Redshift.

Parameter	Diperlukan	Default	Deskripsi
			Pengguna dan kata sandi adalah kredensial untuk mengakses database, yang harus disematkan dalam URL ini untuk JDBC, dan pengguna database harus memiliki izin yang diperlukan untuk mengakses tabel.
aws_iam_role	Hanya jika menggunakan peran IAM untuk mengotorisasi operasi Redshift COPY/UNLOAD	N/A	ARN yang ditentukan sepenuhnya dari Peran IAM yang melekat pada cluster Redshift.

Parameter	Diperlukan	Default	Deskripsi
<code>forward_spark_s3_credentials</code>	Tidak	False	Menunjukkan apakah pustaka ini akan secara otomatis menemukan kredensial yang digunakan Spark untuk terhubung ke Amazon S3, dan apakah akan meneruskan kredensialnya ke Redshift melalui driver JDBC. Kredensial ini dikirim sebagai bagian dari kueri JDBC. Oleh karena itu kami menyarankan Anda mengaktifkan enkripsi SSL dengan koneksi JDBC saat menggunakan opsi ini.
<code>temporary_aws_access_key_id</code>	Tidak	N/A	AWS kunci akses. Harus memiliki izin menulis ke bucket S3.
<code>temporary_aws_secret_access_key</code>	Tidak	N/A	AWS kunci akses rahasia yang sesuai dengan kunci akses.

Parameter	Diperlukan	Default	Deskripsi
temporary_aws_session_token	Tidak	N/A	AWS token sesi sesuai dengan kunci akses yang disediakan.
tempdir	Tidak	N/A	Lokasi yang dapat ditulis di Amazon S3. Digunakan untuk membongkar data saat membaca dan data Avro untuk dimuat ke Redshift saat menulis. Jika Anda menggunakan sumber data Redshift untuk Spark sebagai bagian dari pipeline ETL biasa, akan berguna untuk menyetel kebijakan siklus hidup pada bucket dan menggunakannya sebagai lokasi sementara untuk data ini.

Parameter	Diperlukan	Default	Deskripsi
jdbcdriver	Tidak	Ditentukan oleh subprotokol URL JDBC	Nama kelas driver JDBC yang akan digunakan. Kelas ini harus berada di classpath. Dalam kebanyakan kasus, tidak perlu menentukan opsi ini, karena nama kelas driver yang sesuai harus secara otomatis ditentukan oleh subprotokol URL JDBC.
diststyle	Tidak	Bahkan	Gaya Distribusi Redshift untuk digunakan saat membuat tabel. Opsi yang valid adalah EVEN, KEY atau ALL. Saat menggunakan KEY, Anda juga harus mengatur kunci distribusi dengan opsi distkey.
distkey	Tidak, kecuali menggunakan DISTYLE_KEY	N/A	Nama kolom dalam tabel untuk digunakan sebagai kunci distribusi saat membuat tabel.

Parameter	Diperlukan	Default	Deskripsi
sortkeyspec	Tidak	N/A	Sebuah definisi Redshift Sort Key lengkap.
include_column_list	Tidak	False	Menunjukkan apakah pustaka ini harus secara otomatis mengekstrak kolom dari skema dan menambahkannya ke perintah COPY sesuai dengan opsi pemetaan Kolom .
deskripsi	Tidak	N/A	Deskripsi untuk tabel. Deskripsi diatur dengan perintah SQL COMMENT, dan muncul di sebagian besar alat kueri. Lihat <code>description</code> metadata untuk mengatur deskripsi pada kolom individual.

Parameter	Diperlukan	Default	Deskripsi
preaksi	Tidak	N/A	Sebuah daftar perintah SQL yang dibatasi titik koma untuk dijalankan sebelum memuat perintah COPY. Mungkin berguna untuk menjalankan perintah DELETE atau serupa sebelum memuat data baru. Jika perintah berisi %s, nama tabel akan diformat sebelum runtime (jika Anda menggunakan tabel pementasan). Jika perintah ini gagal, itu diperlakukan sebagai pengecualian. Jika Anda menggunakan tabel pementasan, perubahan akan dikembalikan dan mengembalikan tabel cadangan jika preaksi gagal.

Parameter	Diperlukan	Default	Deskripsi
ekstrakooptions	Tidak	N/A	Daftar opsi tambahan untuk ditambahkan ke perintah Redshift COPY saat memuat data (TRUNCATECOLUMNS seperti MAXERROR n atau). Lihat Parameter opsional untuk daftar lengkap parameter yang tersedia. Perhatikan bahwa karena opsi ini ditambahkan ke akhir COPY perintah, Anda hanya dapat menggunakan opsi yang masuk akal di akhir perintah. Itu harus mencakup sebagian besar kasus penggunaan yang mungkin.

Parameter	Diperlukan	Default	Deskripsi
sse_kms_key	Tidak	N/A	ID AWS KMS kunci yang digunakan untuk enkripsi sisi server di S3 selama operasi Redshift UNLOAD daripada enkripsi default. AWS Peran IAM Redshift harus memiliki akses ke kunci KMS untuk menulis dengannya , dan peran IAM Spark harus memiliki akses ke kunci untuk operasi baca. Membaca data terenkripsi tidak memerlukan perubahan (AWS menanganinya) selama peran IAM Spark memiliki akses yang tepat.
tempformat	Tidak	AVRO	Format untuk menyimpan file sementara di Amazon S3 saat menulis ke Redshift. Nilai yang valid adalah AVRO, CSV, CSV GZIP (CSV terkompresi), dan PARQUET.

Parameter	Diperlukan	Default	Deskripsi
csvnullstring (eksperimental)	Tidak	Null	Nilai string untuk menulis untuk nulls saat menggunakan tempformat CSV. Ini harus menjadi nilai yang tidak muncul dalam data aktual Anda.
autopushdown	Tidak	True	Menunjukkan apakah akan menerapkan predikat dan pushdown kueri dengan menangkap dan menganalisis rencana logis Spark untuk operasi SQL. Operasi diterjemahkan ke dalam kueri SQL dan kemudian dijalankan di Redshift untuk meningkatkan kinerja.

Parameter	Diperlukan	Default	Deskripsi
autopushdown.s3_result_cache	Tidak	False	Cache SQL kueri untuk membongkar data pemetaan jalur Amazon S3 di memori, sehingga kueri yang sama tidak perlu dijalankan lagi dalam sesi Spark yang sama. Hanya didukung saat autopushdown dihidupkan. Kami tidak menyarankan menggunakan parameter ini saat mencampur operasi baca dan tulis karena hasil cache mungkin berisi informasi basi.
unload_s3_format	Tidak	Parquet	Format yang digunakan untuk membongkar hasil kueri. Opsi yang valid adalah Parquet dan Teks, yang menentukan untuk membongkar hasil kueri dalam format teks yang dibatasi pipa.

Parameter	Diperlukan	Default	Deskripsi
extraunloadoptions	Tidak	N/A	Opsi tambahan untuk ditambahkan ke perintah Redshift UNLOAD . Tidak semua opsi dijamin berfungsi karena beberapa opsi mungkin bertentangan dengan opsi lain yang ditetapkan dalam konektor.
copydelay	Tidak	30000	Penundaan (dalam ms) antara percobaan ulang untuk operasi Redshift COPY.
copyretrycount	Tidak	2	Berapa kali untuk mencoba lagi operasi Redshift COPY.

Parameter	Diperlukan	Default	Deskripsi
tempdir_region	Tidak	N/A	AWS Wilayah di mana tempdir berada. Menyetel opsi ini meningkatkan kinerja konektor untuk interaksi dengan tempdir serta secara otomatis memberikan nilai ini sebagai bagian dari operasi COPY dan UNLOAD selama operasi baca dan tulis konektor. Pengaturan ini direkomendasikan dalam situasi berikut: 1) Ketika konektor berjalan di luar AWS, karena penemuan Wilayah otomatis akan gagal dan berdampak negatif pada kinerja konektor. 2) Kapan tempdir berada di Wilayah yang berbeda dari cluster Redshift, karena menggunakan pengaturan ini mengurangi

Parameter	Diperlukan	Default	Deskripsi
			kebutuhan untuk memasok Wilayah secara manual menggunakan parameter <code>extracopyoptions</code> <code>extraunl</code> <code>adoption</code> <code>tempdir</code> tidak bisa berada di Wilayah yang berbeda dari cluster Redshift PARQUET saat menggunakan <code>tempformat</code> meskipun ini menggunakan parameter ini. 3) Ketika konektor berjalan di Wilayah yang berbeda dari <code>tempdir</code>, karena meningkatkan kinerja akses konektor <code>tempdir</code>.

Parameter	Diperlukan	Default	Deskripsi
secret.id	Tidak	N/A	Nama atau ARN rahasia Anda disimpan di. AWS Secrets Manager Anda dapat menggunakan parameter ini untuk secara otomatis menyediakan kredensial Redshift, tetapi hanya jika pengguna, kata sandi, dan DbUser kredensialnya tidak diteruskan ke URL JDBC atau sebagai opsi lain.

Parameter	Diperlukan	Default	Deskripsi
secret.region	Tidak	N/A	AWS Wilayah utama, seperti US East (Virginia N.), untuk mencari secret.id nilai. Jika Anda tidak menentukan Wilayah ini, konektor akan mencoba menggunakan rantai penyedia kredensi Default untuk menyelesaikan Wilayah. secret.id Dalam beberapa kasus, seperti jika Anda menggunakan konektor di luar konektor tidak akan dapat menemukan Wilayah. Sebaiknya gunakan pengaturan ini dalam situasi berikut: 1) Saat konektor berjalan di luar AWS, karena penemuan Wilayah otomatis akan gagal dan mencegah otentikasi dengan Redshift Ketika konektor berjalan di Wilayah

Parameter	Diperlukan	Default	Deskripsi
			yang berbeda dari <code>secret.id</code> , karena meningkatkan kinerja akses konektor rahasia.
<code>rahasia.vpcEndpointUrl</code>	Tidak	N/A	URL endpoint PrivateLink DNS untuk AWS Secrets Manager saat mengganti rantai penyedia kredensi Default .
<code>rahasia.vpcEndpointRegion</code>	Tidak	N/A	Wilayah titik akhir PrivateLink DNS untuk AWS Secrets Manager saat mengganti rantai penyedia kredensi Default .

Parameter	Diperlukan	Default	Deskripsi
jdbc. *	Tidak	N/A	Parameter tambahan untuk diteruskan ke driver JDBC yang mendasarinya di mana wildcard adalah nama parameter JDBC, seperti jdbc.ssl. Perhatikan bahwa jdbc awalan akan dihapus sebelum diteruskan ke driver JDBC. Untuk melihat semua opsi yang mungkin untuk driver Redshift JDBC, lihat Opsi untuk konfigurasi driver JDBC versi 2.x.

Parameter	Diperlukan	Default	Deskripsi
label	Tidak	" "	Pengidentifikasi untuk disertakan dalam kumpulan kueri yang disetel saat menjalankan kueri dengan konektor. Harus 100 atau kurang karakter, dan semua karakter harus validunicodeIdentifierParts . Jika pengenalan Anda memiliki lebih dari 100 karakter, kelebihannya akan dihapus. Saat menjalankan kueri dengan konektor, grup kueri akan ditetapkan sebagai string format JSON, seperti <pre>{ "spark-redshift-connector" : { "svc": " ", "ver": "5.1.0-amzn-1-spark_3.3", "op": "Read", "lbl": "" } }</pre> . Opsi ini mengganti nilai lbl kunci.

Note

Pengakuan: Dokumentasi ini berisi kode contoh dan bahasa yang dikembangkan oleh [Apache Software Foundation](#) yang dilisensikan di bawah lisensi [Apache](#) 2.0.

Jenis data yang didukung

Tipe data berikut di Amazon Redshift didukung dengan konektor Spark. Untuk daftar lengkap tipe data yang didukung di Amazon Redshift, lihat Jenis [data](#). Jika tipe data tidak ada dalam tabel di bawah ini, itu tidak didukung di konektor Spark.

Jenis data	Alias
SMALLINT	INT2
INTEGER	INT, INT4
BIGINT	INT8
DECIMAL	NUMERIC
REAL	FLOAT4
DOUBLE PRECISION	FLOAT8, MENGAPUNG
BOOLEAN	BOOL
CHAR	KARAKTER, NCHAR, BPCHAR
VARCHAR	KARAKTER BERVARIASI, NVARCHAR, TEKS
DATE	
TIMESTAMP	Stempel waktu tanpa zona waktu
TIMESTAMPTZ	Stempel waktu dengan zona waktu
SUPER	

Jenis data	Alias
TIME	Waktu tanpa zona waktu
JADWAL	Waktu dengan zona waktu
VARBYTE	VARBINARY, BINER BERVARIASI

Tipe data yang kompleks

Anda dapat menggunakan konektor percikan untuk membaca dan menulis tipe data kompleks Spark seperti `ArrayType`, `MapType`, dan `StructType` ke dan dari kolom tipe data Redshift SUPER. Jika Anda memberikan skema selama operasi baca, data di kolom akan dikonversi ke tipe kompleks yang sesuai di Spark, termasuk semua jenis bersarang. Selain itu, jika `autopushdown` diaktifkan, proyeksi atribut bersarang, nilai peta, dan indeks array didorong ke Redshift sehingga seluruh struktur data bersarang tidak perlu lagi dibongkar saat mengakses hanya sebagian data.

Saat Anda menulis `DataFrames` dari konektor, kolom jenis apa pun `MapType` (menggunakan `StringType`), `StructType`, atau `ArrayType` ditulis ke kolom tipe data Redshift SUPER. Saat menulis struktur data bersarang ini, `tempformat` parameter harus bertipe `CSV`, `CSV GZIP`, atau `PARQUET`. Menggunakan `AVRO` akan menyebabkan pengecualian. Menulis struktur `MapType` data yang memiliki tipe kunci selain juga `StringType` akan menyebabkan pengecualian.

StructType

Contoh berikut menunjukkan cara membuat tabel dengan tipe data SUPER yang berisi struct

```
create table contains_super (a super);
```

Anda kemudian dapat menggunakan konektor untuk menanyakan `StringType` bidang `hello` dari kolom SUPER `a` dalam tabel menggunakan skema seperti pada contoh berikut.

```
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")
```

Contoh berikut menunjukkan bagaimana menulis struct ke kolom. a

```
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

Jika Anda lebih suka menggunakan a MapType untuk mewakili data Anda, maka Anda dapat menggunakan struktur MapType data dalam skema Anda dan mengambil nilai yang sesuai dengan kunci di peta. Perhatikan bahwa semua kunci dalam struktur MapType data Anda harus bertipe String, dan semua nilai harus dari jenis yang sama, seperti int.

Contoh berikut menunjukkan bagaimana untuk mendapatkan nilai kunci hello di koloma.

```
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

Jika kolom berisi array bukan struct, Anda dapat menggunakan konektor untuk query elemen pertama dalam array.

```
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]")
```

Batasan

Menggunakan tipe data yang kompleks dengan konektor percikan memiliki batasan berikut:

- Semua nama bidang struct bersarang dan kunci peta harus huruf kecil. Jika menanyakan nama bidang kompleks dengan huruf besar, Anda dapat mencoba menghilangkan skema dan menggunakan fungsi `from_json` percikan untuk mengonversi string yang dikembalikan secara lokal sebagai solusi.
- Setiap bidang peta yang digunakan dalam operasi baca atau tulis harus hanya memiliki `StringType` kunci.
- Hanya CSV, CSV GZIP, dan PARQUET didukung nilai tempformat untuk menulis tipe kompleks ke Redshift. Mencoba menggunakan AVRO akan menimbulkan pengecualian.

Mengonfigurasi koneksi untuk driver ODBC versi 2.x untuk Amazon Redshift

Anda dapat menggunakan koneksi ODBC untuk terhubung ke kluster Amazon Redshift Anda dari banyak alat dan aplikasi klien SQL pihak ketiga. Jika alat klien Anda mendukung JDBC, Anda dapat memilih untuk menggunakan jenis koneksi itu daripada ODBC karena kemudahan konfigurasi yang disediakan JDBC. Namun, jika alat klien Anda tidak mendukung JDBC, Anda dapat mengikuti langkah-langkah di bagian ini untuk mengatur koneksi ODBC di komputer klien atau instans Amazon EC2.

Amazon Redshift menyediakan driver ODBC 64-bit untuk sistem operasi Linux, Windows dan Mac; driver ODBC 32-bit dihentikan. Pembaruan lebih lanjut untuk driver ODBC 32-bit tidak akan dirilis, kecuali untuk patch keamanan yang mendesak.

Untuk informasi terbaru tentang perubahan driver ODBC, lihat [log perubahan](#).

Topik

- [Mendapatkan URL ODBC](#)
- [Menggunakan driver Amazon Redshift ODBC di Microsoft Windows](#)
- [Menggunakan driver Amazon Redshift ODBC di Linux](#)
- [Menggunakan driver Amazon Redshift ODBC di Apple macOS](#)
- [Metode autentikasi](#)
- [Konversi tipe data](#)
- [Opsional driver ODBC](#)
- [Versi driver ODBC sebelumnya](#)

Mendapatkan URL ODBC

Amazon Redshift menampilkan URL ODBC untuk kluster Anda di konsol Amazon Redshift. URL ini berisi informasi yang diperlukan untuk mengatur koneksi antara komputer klien Anda dan database.

URL ODBC memiliki format berikut:

```
Driver={driver}; Server=endpoint_host; Database=database_name; UID=user_name;  
PWD=password; Port=port_number
```

Bidang format sebelumnya memiliki nilai-nilai berikut:

Bidang	Nilai
<i>Driver</i>	Nama driver ODBC 64-bit yang akan digunakan: Amazon Redshift ODBC Driver (x64)
<i>Server</i>	Host endpoint dari cluster Amazon Redshift.
<i>Database</i>	Database yang Anda buat untuk cluster Anda.
<i>UID</i>	Nama pengguna dari akun pengguna database yang memiliki izin untuk terhubung ke database. Meskipun nilai ini adalah izin tingkat basis data dan bukan izin tingkat cluster, Anda dapat menggunakan akun pengguna admin Redshift yang Anda atur saat meluncurkan cluster.
<i>PWD</i>	Kata sandi untuk akun pengguna database untuk terhubung ke database.
<i>Port</i>	Nomor port yang Anda tentukan saat meluncurkan cluster. Jika Anda memiliki firewall, pastikan port ini terbuka untuk Anda gunakan.

Berikut ini adalah contoh URL ODBC:

```
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
```

Untuk informasi tentang tempat menemukan URL ODBC, lihat [Menemukan string koneksi cluster Anda](#).

Menggunakan driver Amazon Redshift ODBC di Microsoft Windows

Anda harus menginstal driver Amazon Redshift ODBC di komputer klien yang mengakses gudang data Amazon Redshift. Untuk setiap komputer tempat Anda menginstal driver, ada persyaratan minimum berikut:

- Hak administrator pada mesin.
- Mesin memenuhi persyaratan sistem berikut:
 - Salah satu sistem operasi berikut:

- Windows 10 atau 8.1.
- Windows Server 2019, 2016, atau 2012.
- 100 MB ruang disk yang tersedia.
- Visual C ++ Redistributable untuk Visual Studio 2015 untuk Windows 64-bit diinstal. Anda dapat mengunduh paket instalasi di [Download Visual C++ Redistributable for Visual Studio 2022](#) di situs web Microsoft.

Mengunduh dan menginstal driver Amazon Redshift ODBC

Gunakan prosedur berikut untuk mengunduh dan menginstal driver Amazon Redshift ODBC untuk sistem operasi Windows. Hanya gunakan driver yang berbeda jika Anda menjalankan aplikasi pihak ketiga yang disertifikasi untuk digunakan dengan Amazon Redshift, dan aplikasi itu memerlukan driver khusus itu.

Untuk mengunduh dan menginstal driver ODBC:

1. [Unduh driver berikut: 64-bit ODBC driver versi 2.1.15.0 Di 64-bit ODBC driver versi 2.1.15.0](#)

Nama driver ini adalah Amazon Redshift ODBC Driver (x64).

2. Tinjau lisensi [driver Amazon Redshift ODBC versi 2.x](#).
3. Klik dua kali file.msi, lalu ikuti langkah-langkah di wizard untuk menginstal driver.

Membuat entri DSN sistem untuk koneksi ODBC

Setelah Anda mengunduh dan menginstal driver ODBC, tambahkan entri nama sumber data (DSN) ke komputer klien atau instans Amazon EC2. Alat klien SQL dapat menggunakan sumber data ini untuk terhubung ke database Amazon Redshift.

Kami menyarankan Anda membuat sistem DSN bukan DSN pengguna. Beberapa aplikasi memuat data menggunakan akun pengguna database yang berbeda, dan mungkin tidak dapat mendeteksi pengguna DSNs yang dibuat di bawah akun pengguna database lain.

Note

Untuk autentikasi menggunakan kredensial AWS Identity and Access Management (IAM) atau kredensial penyedia identitas (iDP), diperlukan langkah-langkah tambahan.

Untuk informasi selengkapnya, lihat [Mengkonfigurasi koneksi JDBC atau ODBC untuk menggunakan](#) kredensial IAM.

Untuk membuat entri DSN sistem untuk koneksi ODBC:

1. Di menu Mulai, ketik “Sumber Data ODBC.” Pilih Sumber Data ODBC.

Pastikan Anda memilih Administrator Sumber Data ODBC yang memiliki bitness yang sama dengan aplikasi klien yang Anda gunakan untuk terhubung ke Amazon Redshift.

2. Di Administrator Sumber Data ODBC, pilih tab Driver dan cari folder driver berikut: Amazon Redshift ODBC Driver (x64).
3. Pilih tab Sistem DSN untuk mengonfigurasi driver untuk semua pengguna di komputer, atau tab DSN Pengguna untuk mengonfigurasi driver hanya untuk akun pengguna database Anda.
4. Pilih Tambahkan. Jendela Create New Data Source terbuka.
5. Pilih driver Amazon Redshift ODBC (x64), lalu pilih Selesai. Jendela Pengaturan DSN Driver Amazon Redshift ODBC terbuka.
6. Di bawah bagian Pengaturan Koneksi, masukkan informasi berikut:

- Nama sumber data

Masukkan sebuah nama untuk sumber data. Misalnya, jika Anda mengikuti Panduan Memulai Amazon Redshift, Anda dapat mengetik `exampleclusterdsn` agar mudah mengingat cluster yang Anda kaitkan dengan DSN ini.

- Server

Tentukan host endpoint untuk cluster Amazon Redshift Anda. Anda dapat menemukan informasi ini di konsol Amazon Redshift di halaman detail cluster. Untuk informasi selengkapnya, lihat [Mengonfigurasi koneksi di Amazon Redshift](#).

- Port

Masukkan nomor port yang digunakan database. Bergantung pada port yang Anda pilih saat membuat, memodifikasi, atau memigrasi klaster, izinkan akses ke port yang dipilih.

- Basis Data

Masukkan nama database Amazon Redshift. Jika Anda meluncurkan cluster Anda tanpa menentukan nama database, masukkan `dev`. Jika tidak, gunakan nama yang Anda pilih selama proses peluncuran. Jika Anda mengikuti Panduan Memulai Amazon Redshift, masukkan `dev`.

7. Di bawah bagian Otentikasi, tentukan opsi konfigurasi untuk mengonfigurasi otentikasi standar atau IAM.

8. Pilih Opsi SSL dan tentukan nilai untuk yang berikut ini:

- Mode autentikasi

Pilih mode untuk menangani Secure Sockets Layer (SSL). Dalam lingkungan pengujian, Anda mungkin menggunakan `prefer`. Namun, untuk lingkungan produksi dan ketika pertukaran data yang aman diperlukan, gunakan `verify-ca` atau `verify-full`.

- Min TLS

Secara opsional, pilih versi minimum TLS/SSL yang memungkinkan penyimpanan data digunakan oleh driver untuk mengenkripsi koneksi. Misalnya, jika Anda menentukan TLS 1.2, TLS 1.1 tidak dapat digunakan untuk mengenkripsi koneksi. Versi defaultnya adalah TLS 1.2.

9. Di tab Proxy, tentukan pengaturan koneksi proxy apa pun.

10 Di tab Cursor, tentukan opsi tentang cara mengembalikan hasil kueri ke alat atau aplikasi klien SQL Anda.

11 Di Opsi Lanjutan, tentukan nilai untuk `LogLevel`, `logPath`, `compression`, dan opsi lainnya.

12 Pilih Uji. Jika komputer klien dapat terhubung ke database Amazon Redshift, pesan berikut akan muncul: Koneksi berhasil. Jika komputer klien gagal terhubung ke database, Anda dapat memecahkan masalah yang mungkin terjadi dengan membuat file log dan menghubungi AWS dukungan. Untuk informasi tentang membuat log, lihat (LINK).

13 Pilih OK.

Menggunakan driver Amazon Redshift ODBC di Linux

Anda harus menginstal driver Amazon Redshift ODBC di komputer klien yang mengakses gudang data Amazon Redshift. Untuk setiap komputer tempat Anda menginstal driver, ada persyaratan minimum berikut:

- Akses root pada mesin.
- Salah satu distribusi berikut:
 - Red Hat® Enterprise Linux® (RHEL) 8 atau lebih baru
 - CentOS 8 atau yang lebih baru.
- 150 MB ruang disk yang tersedia.
- UnixODBC 2.2.14 atau yang lebih baru.

- glibc 2.26 atau yang lebih baru.

Mengunduh dan menginstal driver Amazon Redshift ODBC

Untuk mengunduh dan menginstal driver Amazon Redshift ODBC versi 2.x untuk Linux:

1. Unduh driver berikut:

- [x86 64-bit RPM driver versi 2.1.15.0](#) Di [versi driver RPM 64-bit 2.1.15.0](#)
- [Versi driver ARM 64-bit RPM 2.1.15.0](#) versi 2.1.15.0

Note

Driver ODBC 32-bit dihentikan. Pembaruan lebih lanjut tidak akan dirilis, kecuali untuk patch keamanan yang mendesak.

2. Pergi ke lokasi di mana Anda mengunduh paket, dan kemudian jalankan salah satu perintah berikut. Gunakan perintah yang sesuai dengan distribusi Linux Anda.

Pada sistem operasi RHEL dan CentOS, jalankan perintah berikut:

```
yum --nogpgcheck localinstall RPMFileName
```

Ganti *RPMFileName* dengan nama file paket RPM. Misalnya, perintah berikut menunjukkan menginstal driver 64-bit:

```
yum --nogpgcheck localinstall AmazonRedshiftODBC-64-bit-2.x.xx.xxxx.x86_64.rpm
```

Menggunakan manajer driver ODBC untuk mengonfigurasi driver ODBC

Di Linux, Anda menggunakan manajer driver ODBC untuk mengonfigurasi pengaturan koneksi ODBC. Manajer driver ODBC menggunakan file konfigurasi untuk menentukan dan mengkonfigurasi sumber data dan driver ODBC. Manajer driver ODBC yang Anda gunakan bergantung pada sistem operasi yang Anda gunakan.

Mengkonfigurasi driver ODBC menggunakan pengelola driver UnixODBC

File-file berikut diperlukan untuk mengonfigurasi driver Amazon Redshift ODBC:

- `amazon.redshiftodbc.ini`
- `odbc.ini`
- `odbcinst.ini`

Jika Anda menginstal ke lokasi default, file `amazon.redshiftodbc.ini` konfigurasi terletak di `/opt/amazon/redshiftodbcx64`.

Selain itu, di bawah `/opt/amazon/redshiftodbcx64`, Anda dapat menemukan sampel `odbc.ini` dan `odbcinst.ini` file. Anda dapat menggunakan file-file ini sebagai contoh untuk mengonfigurasi driver Amazon Redshift ODBC dan nama sumber data (DSN).

Kami tidak menyarankan menggunakan direktori instalasi driver Amazon Redshift ODBC untuk file konfigurasi. File sampel dalam direktori yang diinstal hanya untuk tujuan contoh. Jika Anda menginstal ulang driver Amazon Redshift ODBC di lain waktu, atau meningkatkan ke versi yang lebih baru, direktori instalasi akan ditimpa. Anda akan kehilangan perubahan apa pun yang mungkin telah Anda buat pada file di direktori instalasi.

Untuk menghindari hal ini, salin `amazon.redshiftodbc.ini` file ke direktori selain direktori instalasi. Jika Anda menyalin file ini ke direktori home pengguna, tambahkan titik (.) ke awal nama file untuk menjadikannya file tersembunyi.

Untuk `odbcinst.ini` file `odbc.ini` dan, gunakan file konfigurasi di direktori home pengguna atau buat versi baru di direktori lain. Secara default, sistem operasi Linux Anda harus memiliki `odbc.ini` file dan `odbcinst.ini` file di direktori home pengguna (`/home/$USER` atau `~/.`). File default ini adalah file tersembunyi, yang ditunjukkan oleh titik (.) di depan setiap nama file. File-file ini hanya ditampilkan ketika Anda menggunakan `-a` bendera untuk daftar isi direktori.

Opsi apa pun yang Anda pilih untuk `odbcinst.ini` file `odbc.ini` dan, ubah file untuk menambahkan informasi konfigurasi driver dan DSN. Jika Anda membuat file baru, Anda juga perlu mengatur variabel lingkungan untuk menentukan di mana file konfigurasi ini berada.

Secara default, manajer driver ODBC dikonfigurasi untuk menggunakan versi tersembunyi dari file `odbc.ini` dan `odbcinst.ini` konfigurasi (bernama `.odbc.ini` dan `.odbcinst.ini`) yang terletak di direktori home. Mereka juga dikonfigurasi untuk menggunakan `amazon.redshiftodbc.ini` file di direktori instalasi driver. Jika Anda menyimpan file konfigurasi

ini di tempat lain, atur variabel lingkungan yang dijelaskan berikut ini sehingga manajer driver dapat menemukan file.

Jika Anda menggunakan UnixODBC, lakukan hal berikut:

- Atur ODBCINI ke path lengkap dan nama file `odbc.ini` file.
- Setel ODBCSYSINI ke path lengkap direktori yang berisi `odbcinst.ini` file.
- Atur AMAZONREDSHIFTODBCINI ke path lengkap dan nama file `amazon.redshiftodbc.ini` file.

Berikut ini adalah contoh pengaturan nilai-nilai di atas:

```
export ODBCINI=/usr/local/odbc/odbc.ini
export ODBCSYSINI=/usr/local/odbc
export AMAZONREDSHIFTODBCINI=/etc/amazon.redshiftodbc.ini
```

Mengkonfigurasi koneksi menggunakan nama sumber data (DSN) di Linux

Saat menghubungkan ke penyimpanan data Anda menggunakan nama sumber data (DSN), konfigurasi `odbc.ini` file untuk menentukan nama sumber data (DSNs). Tetapkan properti dalam `odbc.ini` file untuk membuat DSN yang menentukan informasi koneksi untuk penyimpanan data Anda.

Pada sistem operasi Linux, gunakan format berikut:

```
[ODBC Data Sources]
driver_name=dsn_name

[dsn_name]
Driver=path/driver_file
Host=cluster_endpoint
Port=port_number
Database=database_name
locale=locale
```

Contoh berikut menunjukkan konfigurasi untuk `odbc.ini` dengan driver ODBC 64-bit pada sistem operasi Linux.

```
[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
```

Mengkonfigurasi koneksi tanpa DSN di Linux

Untuk terhubung ke penyimpanan data Anda melalui koneksi yang tidak memiliki DSN, tentukan driver dalam `odbcinst.ini` file. Kemudian berikan string koneksi tanpa DSN di aplikasi Anda.

Pada sistem operasi Linux, gunakan format berikut:

```
[ODBC Drivers]
driver_name=Installed
...

[driver_name]
Description=driver_description
Driver=path/driver_file
...
```

Contoh berikut menunjukkan konfigurasi untuk `odbcinst.ini` dengan driver ODBC 64-bit pada sistem operasi Linux.

```
[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
```

Menggunakan driver Amazon Redshift ODBC di Apple macOS

Anda harus menginstal driver Amazon Redshift ODBC di komputer klien yang mengakses gudang data Amazon Redshift. Untuk setiap komputer tempat Anda menginstal driver, ini adalah persyaratan minimum berikut:

- Akses root pada mesin.
- Persyaratan Sistem Apple macOS:
 - Versi 64-bit Apple macOS versi 11.7 atau lebih tinggi (seperti Apple macOS Big Sur, Monterey, Ventura atau yang lebih baru) diperlukan. Driver Redshift ODBC hanya mendukung aplikasi klien 64-bit.
 - 150 MB ruang disk yang tersedia.
 - Driver mendukung aplikasi yang dibangun dengan iODBC 3.52.9+ atau UnixODBC 2.3.7+.

Mengunduh dan menginstal driver Amazon Redshift ODBC

Gunakan prosedur berikut untuk mengunduh dan menginstal driver Amazon Redshift ODBC di Apple macOS. Hanya gunakan driver yang berbeda jika Anda menjalankan aplikasi pihak ketiga yang disertifikasi untuk digunakan dengan Amazon Redshift, dan aplikasi itu memerlukan driver khusus itu.

Untuk mengunduh dan menginstal driver ODBC:

1. [Unduh driver berikut: 64-bit ODBC driver versi 2.1.15.0 Di 64-bit ODBC driver versi 2.1.15.0](#)

Driver ini didukung pada arsitektur x86_64 dan arm64. Nama driver ini adalah Amazon Redshift ODBC Driver (x64).

2. Tinjau lisensi [driver Amazon Redshift ODBC versi 2.x](#).
3. Klik dua kali file.pkg, lalu ikuti langkah-langkah di wizard untuk menginstal driver. Atau, jalankan perintah berikut:

```
sudo installer -pkg PKGFileName -target /
```

Ganti *PKGFileName* dengan nama file paket pkg. Misalnya, perintah berikut menunjukkan menginstal driver 64-bit:

```
sudo installer -pkg ./AmazonRedshiftODBC-64-bit.X.X.XX.X.universal.pkg -target /
```

Menggunakan manajer driver ODBC untuk mengonfigurasi driver ODBC

Di Mac, Anda menggunakan pengelola driver ODBC untuk mengonfigurasi pengaturan koneksi ODBC. Manajer driver ODBC menggunakan file konfigurasi untuk menentukan dan mengkonfigurasi sumber data dan driver ODBC. Manajer driver ODBC yang Anda gunakan bergantung pada sistem operasi yang Anda gunakan.

Mengkonfigurasi driver ODBC menggunakan manajer driver iodBC atau UnixODBC

File-file berikut diperlukan untuk mengonfigurasi driver Amazon Redshift ODBC:

- `amazon.redshiftodbc.ini`
- `odbc.ini`
- `odbcinst.ini`

Jika Anda menginstal ke lokasi default, file `amazon.redshiftodbc.ini` konfigurasi terletak di `/opt/amazon/redshiftodbcx64`.

Selain itu, di bawah `/opt/amazon/redshiftodbcx64`, Anda dapat menemukan sampel `odbc.ini` dan `odbcinst.ini` file. Anda dapat menggunakan file-file ini sebagai contoh untuk mengonfigurasi driver Amazon Redshift ODBC dan nama sumber data (DSN). File sampel dalam direktori yang diinstal hanya untuk tujuan contoh.

Kami tidak menyarankan menggunakan direktori instalasi driver Amazon Redshift ODBC untuk file konfigurasi. Jika Anda menginstal ulang driver Amazon Redshift ODBC di lain waktu, atau meningkatkan ke versi yang lebih baru, direktori instalasi akan ditimpa. Anda akan kehilangan perubahan apa pun yang mungkin telah Anda buat pada file di direktori instalasi.

Untuk menghindari hal ini `iodbc.ini`, salin, `odbcinst.ini` dan `amazon.redshiftodbc.ini` file ke direktori selain direktori instalasi. Jika Anda menyalin file-file ini ke direktori home pengguna, tambahkan titik (.) ke awal nama file ini untuk menjadikannya file tersembunyi.

Ubah file untuk menambahkan informasi konfigurasi DSN. Saat Anda membuat file baru, Anda juga perlu mengatur variabel lingkungan untuk menentukan di mana file konfigurasi ini berada.

Berikut ini adalah contoh pengaturan variabel lingkungan:

```
export ODBCINI=/Library/ODBC/odbc.ini
export ODBCSYSINI=/Library/ODBC
export ODBCINSTINI=${ODBCSYSINI}/odbcinst.ini
```

Untuk aplikasi baris perintah: Tambahkan perintah ekspor ke file startup shell Anda (misalnya, `~/.bash_profile` atau `~/.zshrc`).

Untuk versi manajer driver yang didukung, lihat [di sini](#)

Mengonfigurasi koneksi menggunakan nama sumber data (DSN) di Apple macOS

Saat menghubungkan ke penyimpanan data Anda menggunakan nama sumber data (DSN), konfigurasi `odbc.ini` file untuk menentukan nama sumber data (DSNs). Tetapkan properti dalam `odbc.ini` file untuk membuat DSN yang menentukan informasi koneksi untuk gudang data Redshift Anda.

Di Apple macOS, gunakan format berikut:

```
[ODBC Data Sources]
driver_name=dsn_name

[dsn_name]
Driver=path/driver_file
Host=cluster_endpoint
Port=port_number
Database=database_name
locale=locale
```

Contoh berikut menunjukkan konfigurasi untuk `odbc.ini` dengan driver ODBC 64-bit di Apple macOS.

```
[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
```

Mengkonfigurasi koneksi tanpa DSN di Apple macOS

Untuk terhubung ke gudang data Redshift Anda melalui koneksi yang tidak memiliki DSN, tentukan driver dalam file `odbcinst.ini`. Kemudian berikan string koneksi tanpa DSN di aplikasi Anda.

Di Apple macOS, gunakan format berikut:

```
[ODBC Drivers]
driver_name=Installed
...

[driver_name]
Description=driver_description
Driver=path/driver_file
...
```

Contoh berikut menunjukkan konfigurasi untuk `odbcinst.ini` dengan driver ODBC 64-bit di Apple macOS.

```
[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
```

Metode autentikasi

Untuk melindungi data dari akses yang tidak sah, penyimpanan data Amazon Redshift memerlukan semua koneksi untuk diautentikasi menggunakan kredensial pengguna.

Tabel berikut mengilustrasikan opsi koneksi yang diperlukan dan opsional untuk setiap metode otentikasi yang dapat digunakan untuk menyambung ke driver Amazon Redshift ODBC versi 2.x:

Metode Otentikasi	Diperlukan	Opsional
Standar	• Host • Port • Basis Data • UID • Kata Sandi	
Profil IAM	• Host	• ClusterID

Metode Otentikasi	Diperlukan	Opsional
	• Port • Basis Data • IAM • Profil	• Region • AutoCreate • EndpointURL • StsEndpointURL • InstanceProfile  Note ClusterID dan Region harus disetel di Host jika tidak disetel secara terpisah.
Kredensi IAM	• Host • Port • Basis Data • IAM • AccessKeyID • SecretAccessKey	• ClusterID • Region • AutoCreate • EndpointURL • StsEndpointURL • SessionToken • UID  Note ClusterID dan Region harus disetel di Host jika tidak disetel secara terpisah.

Metode Otentikasi	Diperlukan	Opsional
IKLAN FS	• Host • Port • Basis Data • IAM • plugin_name • UID • Kata Sandi • IDP_host • IDP_port	• ClusterID • Region • AutoCreate • EndpointUrl • StsEndpointUrl • Diutamakan_Peran • loginToRp • SSL_tidak aman  Note ClusterID dan Region harus disetel di Host jika tidak disetel secara terpisah.
Azure AD	• Host • Port • Basis Data • IAM • plugin_name • UID • Kata Sandi • IDP_penyewa • Client_ID • Klien_Rahasia	• ClusterID • Region • AutoCreate • EndpointUrl • StsEndpointUrl • Diutamakan_Peran • dbgroups_filter  Note ClusterID dan Region harus disetel di Host jika tidak disetel secara terpisah.

Metode Otentikasi	Diperlukan	Opsional
JWT	• Host • Port • Basis Data • IAM • plugin_name • web_identity_token	• provider_name
Okta	• Host • Port • Basis Data • IAM • plugin_name • UID • Kata Sandi • IDP_host • App_Nama • App_ID	• ClusterID • Region • AutoCreate • EndpointUrl • StsEndpointUrl • Diutamakan_Peran  Note ClusterID dan Region harus disetel di Host jika tidak disetel secara terpisah.

Metode Otentikasi	Diperlukan	Opsional
Federasi Ping	• Host • Port • Basis Data • IAM • plugin_name • UID • Kata Sandi • IDP_host • IDP_port	• ClusterID • Region • AutoCreate • EndpointUrl • StsEndpointUrl • Diutamakan_Peran • SSL_tidak aman • partner_spid  Note ClusterID dan Region harus disetel di Host jika tidak disetel secara terpisah.

Metode Otentikasi	Diperlukan	Opsional
Browser Azure AD	• Host • Port • Basis Data • IAM • plugin_name • IDP_penyewa • Client_ID • UID	• ClusterID • Region • AutoCreate • EndpointUrl • StsEndpointUrl • Diutamakan_Peran • dbgroups_filter • IDP_Response_Timeout • listen_port  Note ClusterID dan Region harus disetel di Host jika tidak disetel secara terpisah.

Metode Otentikasi	Diperlukan	Opsional
Browser SAMP	• Host • Port • Basis Data • IAM • plugin_name • login_url • UID	• ClusterID • Region • AutoCreate • EndpointUrl • StsEndpointUrl • Diutamakan_Peran • dbgroups_filter • IDP_Response_Timeout • listen_port  Note ClusterID dan Region harus disetel di Host jika tidak disetel secara terpisah.
Profil Auth	• Host • Port • Basis Data • AccessKeyID • SecretAccessKey	

Metode Otentikasi	Diperlukan	Opsional
Browser Azure AD OAUTH2	• Host • Port • Basis Data • IAM • plugin_name • IDP_penyewa • Client_ID • UID	• ClusterID • Region • EndpointUrl • IDP_Response_Timeout • listen_port • cakupan • provider_name  Note ClusterID dan Region harus disetel di Host jika tidak disetel secara terpisah.
AWS Pusat Identitas IAM	• Host • Basis Data • plugin_name • idc_region • issuer_url	• idc_client_display_name • idp_response_timeout • listen_port

Menggunakan layanan kredensyal eksternal

Selain dukungan bawaan untuk AD FS, Azure AD, dan Okta, versi Windows dari driver Amazon Redshift ODBC juga menyediakan dukungan untuk layanan kredensil lainnya. Driver dapat mengautentikasi koneksi menggunakan plugin penyedia kredensyal berbasis SAML pilihan Anda.

Untuk mengkonfigurasi layanan kredensial eksternal pada Windows:

1. Buat profil IAM yang menentukan plugin penyedia kredensyal dan parameter otentikasi lainnya sesuai kebutuhan. Profil harus dikodekan ASCII, dan harus berisi pasangan kunci-nilai berikut, di mana jalur lengkap ke aplikasi PluginPath plugin:


```
plugin_name = PluginPath
```

Contoh:

```
plugin_name = C:\Users\kjson\myapp\CredServiceApp.exe
```

Untuk informasi tentang cara membuat profil, lihat [Menggunakan Profil Konfigurasi di Panduan Manajemen Cluster Amazon Redshift](#).

2. Konfigurasi driver untuk menggunakan profil ini. Driver mendeteksi dan menggunakan pengaturan otentikasi yang ditentukan dalam profil.

Konversi tipe data

Driver Amazon Redshift ODBC versi 2.x mendukung banyak format data umum, mengonversi antara Amazon Redshift dan tipe data SQL.

Tabel berikut mencantumkan pemetaan tipe data yang didukung.

Jenis Amazon Redshift	Jenis SQL
BIGINT	SQL_BIGINT
BOOLEAN	SQL_BIT
CHAR	SQL_CHAR
DATE	SQL_TYPE_DATE
DECIMAL	SQL_NUMERIK
DOUBLE PRECISION	SQL_GANDA
GEOGRAPHY	SQL_LONGVARBINARY
GEOMETRY	SQL_LONGVARBINARY
INTEGER	SQL_INTEGER

Jenis Amazon Redshift	Jenis SQL
REAL	SQL_REAL
SMALLINT	SQL_SMALLINT
SUPER	SQL_LONGVARCHAR
TEXT	SQL_LONGVARCHAR
TIME	SQL_TYPE_TIME
JADWAL	SQL_TYPE_TIME
TIMESTAMP	STEMPEL WAKTU SQL_TYPE_
TIMESTAMPTZ	STEMPEL WAKTU SQL_TYPE_
VARBYTE	SQL_LONGVARBINARY
VARCHAR	SQL_VARCHAR

Opsi driver ODBC

Anda dapat menggunakan opsi konfigurasi driver untuk mengontrol perilaku driver Amazon Redshift ODBC. Opsi driver tidak peka huruf besar/kecil.

Di Microsoft Windows, Anda biasanya mengatur opsi driver saat mengonfigurasi nama sumber data (DSN). Anda juga dapat mengatur opsi driver dalam string koneksi saat Anda terhubung secara terprogram, atau dengan menambahkan atau mengubah kunci registri. `HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\your_DSN`

Di Linux, Anda mengatur opsi konfigurasi driver di `amazon.redshiftodbc.ini` file `odbc.ini` dan Anda. Opsi konfigurasi yang diatur dalam `amazon.redshiftodbc.ini` file berlaku untuk semua koneksi. Sebaliknya, opsi konfigurasi yang diatur dalam `odbc.ini` file khusus untuk koneksi. Opsi konfigurasi yang disetel lebih `odbc.ini` diutamakan daripada opsi konfigurasi yang ditetapkan. `amazon.redshiftodbc.ini`

Berikut ini adalah deskripsi untuk opsi yang dapat Anda tentukan untuk driver Amazon Redshift ODBC versi 2.x:

AccessKeyID

- Nilai Default - Tidak Ada
- Tipe Data - String

Kunci akses IAM untuk pengguna atau peran. Jika Anda mengatur parameter ini, Anda juga harus menentukan SecretAccessKey.

Parameter ini bersifat opsional.

app_id

- Nilai Default - Tidak Ada
- Tipe Data - String

ID unik yang disediakan OKTA yang terkait dengan aplikasi Amazon Redshift Anda.

Parameter ini bersifat opsional.

ApplicationName

- Nilai default - Tidak ada
- Tipe data - String

Nama aplikasi klien untuk diteruskan ke Amazon Redshift untuk tujuan audit. [Nama aplikasi yang Anda berikan muncul di kolom 'application_name' dari tabel SYS_CONNECTION_LOG](#). Ini membantu melacak dan memecahkan masalah sumber koneksi saat men-debug masalah.

Parameter ini bersifat opsional.

app_name

- Nilai Default - Tidak Ada
- Tipe Data - String

Nama aplikasi Okta yang Anda gunakan untuk mengautentikasi koneksi ke Amazon Redshift.

Parameter ini bersifat opsional.

AuthProfile

- Nilai Default - Tidak Ada
- Tipe Data - String

Profil otentikasi yang digunakan untuk mengelola pengaturan koneksi. Jika Anda mengatur parameter ini, Anda juga harus mengatur AccessKeyID dan SecretAccessKey.

Parameter ini bersifat opsional.

AuthType

- Nilai Default - Standar
- Tipe Data - String

Opsi ini menentukan mode otentikasi yang digunakan driver saat Anda mengonfigurasi DSN menggunakan kotak dialog Amazon Redshift ODBC Driver DSN Setup:

- Standar: Otentikasi standar menggunakan nama pengguna dan kata sandi Amazon Redshift Anda.
- AWS Profil: Autentikasi IAM menggunakan profil.
- AWS IAM Credentials: Autentikasi IAM menggunakan kredensi IAM.
- Penyedia Identitas: AD FS: Autentikasi IAM menggunakan Layanan Federasi Direktori Aktif (AD FS).
- Penyedia Identitas: Plugin Auth: Plugin otorisasi yang menerima token AWS IAM Identity Center atau token identitas berbasis JSON (OIDC) OpenID Connect (OIDC) dari penyedia identitas web mana pun yang ditautkan ke IAM Identity Center. AWS
- Penyedia Identitas: Azure AD: Autentikasi IAM menggunakan portal Azure AD.
- Penyedia Identitas: JWT: Autentikasi IAM menggunakan JSON Web Token (JWT).
- Penyedia Identitas: Okta: Otentikasi IAM menggunakan Okta.
- Penyedia Identitas: PingFederate: Autentikasi IAM menggunakan. PingFederate

Opsi ini hanya tersedia ketika Anda mengonfigurasi DSN menggunakan kotak dialog Amazon Redshift ODBC Driver DSN Setup di driver Windows. Saat Anda mengonfigurasi koneksi menggunakan string koneksi atau mesin non-Windows, driver secara otomatis menentukan apakah akan menggunakan otentikasi Kredensial Standar, AWS Profil, atau AWS IAM berdasarkan kredensi

yang Anda tentukan. Untuk menggunakan penyedia identitas, Anda harus mengatur properti `plugin_name`.

Parameter ini diperlukan.

AutoCreate

- Nilai Default - 0
- Tipe Data - Boolean

Boolean yang menentukan apakah driver membuat pengguna baru ketika pengguna yang ditentukan tidak ada.

- 1 | BENAR: Jika pengguna yang ditentukan oleh UID tidak ada, driver membuat pengguna baru.
- 0 | SALAH: Driver tidak membuat pengguna baru. Jika pengguna yang ditentukan tidak ada, otentikasi gagal.

Parameter ini bersifat opsional.

CaFile

- Nilai Default - Tidak Ada
- Tipe Data - String

Jalur file ke file sertifikat CA yang digunakan untuk beberapa bentuk otentikasi IAM.

Parameter ini hanya tersedia di Linux.

Parameter ini bersifat opsional.

client_id

- Nilai Default - Tidak Ada
- Tipe Data - String

ID klien yang terkait dengan aplikasi Amazon Redshift Anda di Azure AD.

Parameter ini diperlukan jika mengautentikasi melalui layanan Azure AD.

rahasia klien_

- Nilai Default - Tidak Ada
- Tipe Data - String

Kunci rahasia yang terkait dengan aplikasi Amazon Redshift Anda di Azure AD.

Parameter ini diperlukan jika mengautentikasi melalui layanan Azure AD.

ClusterId

- Nilai Default - Tidak Ada
- Tipe Data - String

Nama cluster Amazon Redshift yang ingin Anda sambungkan. Ini digunakan dalam otentikasi IAM. ID Cluster tidak ditentukan dalam parameter Server.

Parameter ini bersifat opsional.

Kompresi

- Nilai Default - off
- Tipe Data - String

Metode kompresi yang digunakan untuk komunikasi protokol kawat antara server Amazon Redshift dan klien atau driver.

Anda dapat menentukan salah satu nilai berikut:

- lz4: Mengatur metode kompresi yang digunakan untuk komunikasi protokol kawat dengan Amazon Redshift ke. `lz4`
- zstd: Menyetel metode kompresi yang digunakan untuk komunikasi protokol kawat dengan Amazon Redshift ke. `zstd`
- off: Tidak menggunakan kompresi untuk komunikasi protokol kawat dengan Amazon Redshift.

Parameter ini bersifat opsional.

Basis Data

- Nilai Default - Tidak Ada
- Tipe Data - String

Nama database Amazon Redshift yang ingin Anda akses.

Parameter ini diperlukan.

DatabaseMetadataCurrentDbOnly

- Nilai Default - 1
- Tipe Data - Boolean

Boolean yang menentukan apakah driver mengembalikan metadata dari beberapa database dan cluster.

- 1 | BENAR: Driver hanya mengembalikan metadata dari database saat ini.
- 0 | SALAH. Driver mengembalikan metadata di beberapa database dan cluster Amazon Redshift.

Parameter ini bersifat opsional.

dbgroups_filter

- Nilai Default - Tidak Ada
- Tipe Data - String

Ekspresi reguler yang dapat Anda tentukan untuk memfilter DbGroups yang diterima dari respons SAFL ke Amazon Redshift saat menggunakan jenis autentikasi Azure, Browser Azure, dan Browser SALL.

Parameter ini bersifat opsional.

Driver

- Nilai Default - Driver Amazon Redshift ODBC (x64)
- Tipe Data - String

Nama sopirnya. Satu-satunya nilai yang didukung adalah Amazon Redshift ODBC Driver (x64).

Parameter ini diperlukan jika Anda tidak mengatur DSN.

DSN

- Nilai Default - Tidak Ada
- Tipe Data - String

Nama nama sumber data driver. Aplikasi menentukan DSN di SQLDriver Connect API.

Parameter ini diperlukan jika Anda tidak mengatur Driver. .

EndpointUrl

- Nilai Default - Tidak Ada
- Tipe Data - String

Titik akhir utama yang digunakan untuk berkomunikasi dengan Amazon Redshift Coral Service untuk otentikasi IAM.

Parameter ini bersifat opsional.

ForceLowercase

- Nilai Default - 0
- Tipe Data - Boolean

Boolean yang menentukan apakah driver huruf kecil semua DbGroups dikirim dari penyedia identitas ke Amazon Redshift saat menggunakan autentikasi masuk tunggal.

- 1 | BENAR: Driver huruf kecil semua DbGroups yang dikirim dari penyedia identitas.
- 0 | SALAH: Pengemudi tidak berubah DbGroups.

Parameter ini bersifat opsional.

group_federasi

- Nilai Default - 0

- Tipe Data - Boolean

Boolean yang menentukan apakah `getClusterCredentialsWithIAM` API digunakan untuk mendapatkan kredensial kluster sementara di kluster yang disediakan. Opsi ini memungkinkan pengguna IAM berintegrasi dengan peran database Redshift dalam cluster yang disediakan. Perhatikan bahwa opsi ini tidak berlaku untuk ruang nama Redshift Tanpa Server.

- 1 | BENAR: Driver menggunakan `getClusterCredentialsWithIAM` API untuk mendapatkan kredensial cluster sementara di cluster yang disediakan.
- 0 | FALSE: Driver menggunakan `getClusterCredentials` API default untuk mendapatkan kredensial cluster sementara di cluster yang disediakan.

Parameter ini bersifat opsional.

`http_proxy_host`

- Nilai Default - Tidak Ada
- Tipe Data - String

Nama host atau alamat IP server proxy tempat Anda ingin melewati proses otentikasi IAM.

Parameter ini bersifat opsional.

`http_proxy_password`

- Nilai Default - Tidak Ada
- Tipe Data - String

Kata sandi yang Anda gunakan untuk mengakses server proxy. Ini digunakan untuk otentikasi IAM.

Parameter ini bersifat opsional.

`http_proxy_port`

- Nilai Default - Tidak Ada
- Tipe Data - Integer

Jumlah port yang digunakan server proxy untuk mendengarkan koneksi klien. Ini digunakan untuk otentikasi IAM.

Parameter ini bersifat opsional.

http_proxy_nama pengguna

- Nilai Default - Tidak Ada
- Tipe Data - String

Nama pengguna yang Anda gunakan untuk mengakses server proxy. Ini digunakan untuk otentikasi IAM.

Parameter ini bersifat opsional.

IAM

- Nilai Default - 0
- Tipe Data - Boolean

Boolean yang menentukan apakah driver menggunakan metode otentikasi IAM untuk mengautentikasi koneksi.

- 1 | BENAR: Driver menggunakan salah satu metode autentikasi IAM (menggunakan kunci akses dan secret key pair, atau profil, atau layanan kredensial).
- 0 | SALAH. Driver menggunakan otentikasi standar (menggunakan nama pengguna dan kata sandi database Anda).

Parameter ini bersifat opsional.

idc_client_display_name

- Nilai Default - Driver Amazon Redshift ODBC
- Tipe Data - String

Nama tampilan yang akan digunakan untuk klien yang menggunakan BrowserIdcAuthPlugin.

Parameter ini bersifat opsional.

idc_region

- Nilai Default - Tidak Ada
- Tipe Data - String

AWS Wilayah di mana instans AWS IAM Identity Center berada.

Parameter ini diperlukan hanya ketika mengautentikasi menggunakan `BrowserIdcAuthPlugin` dalam opsi konfigurasi `plugin_name`.

idp_host

- Nilai Default - Tidak Ada
- Tipe Data - String

Host IDP (penyedia identitas) yang Anda gunakan untuk mengautentikasi ke Amazon Redshift.

Parameter ini bersifat opsional.

idp_port

- Nilai Default - Tidak Ada
- Tipe Data - Integer

Port untuk IDP (penyedia identitas) yang Anda gunakan untuk mengautentikasi ke Amazon Redshift. Bergantung pada port yang Anda pilih saat membuat, memodifikasi, atau memigrasi klaster, izinkan akses ke port yang dipilih.

Parameter ini bersifat opsional.

idp_response_timeout

- Nilai Default - 120
- Tipe Data - Integer

Jumlah detik driver menunggu respons SAMP dari penyedia identitas saat menggunakan layanan SAMP atau Azure AD melalui plugin browser.

Parameter ini bersifat opsional.

`idp_penyewa`

- Nilai Default - Tidak Ada
- Tipe Data - String

ID penyewa Azure AD yang terkait dengan aplikasi Amazon Redshift Anda.

Parameter ini diperlukan jika mengautentikasi melalui layanan Azure AD.

`idp_partisi`

- Nilai Default - Tidak Ada
- Tipe Data - String

Menentukan partisi cloud tempat penyedia identitas Anda (iDP) dikonfigurasi. Ini menentukan titik akhir otentikasi IDP mana yang terhubung ke driver.

Jika parameter ini dibiarkan kosong, driver default ke partisi komersial. Kemungkinan nilainya adalah:

- `us-gov`: Gunakan nilai ini jika IDP Anda dikonfigurasi di Azure Government. Misalnya, Azure AD Government menggunakan `login.microsoftonline.us` endpoint.
- `cn`: Gunakan nilai ini jika IDP Anda dikonfigurasi di partisi cloud Tiongkok. Misalnya, Azure AD Tiongkok menggunakan titik akhir. `login.chinacloudapi.cn`

Parameter ini bersifat opsional.

`idp_use_https_proxy`

- Nilai Default - 0
- Tipe Data - Boolean

Boolean yang menentukan apakah driver melewati proses otentikasi untuk penyedia identitas (iDP) melalui server proxy.

- 1 | BENAR: Driver melewati proses otentikasi IDP melalui server proxy.

- 0 | SALAH. Driver tidak melewati proses otentikasi IDP melalui server proxy.

Parameter ini bersifat opsional.

InstanceProfile

- Nilai Default - 0
- Tipe Data - Boolean

Boolean yang menentukan apakah driver menggunakan profil instans Amazon EC2, saat dikonfigurasi untuk menggunakan profil untuk otentikasi.

- 1 | BENAR: Driver menggunakan profil instans Amazon EC2.
- 0 | SALAH. Pengemudi menggunakan profil peran berantai yang ditentukan oleh opsi Nama Profil (Profil) sebagai gantinya.

Parameter ini bersifat opsional.

issuer_url

- Nilai Default - Tidak Ada
- Tipe Data - String

Menunjuk ke titik akhir instance server AWS IAM Identity Center.

Parameter ini diperlukan hanya ketika mengautentikasi menggunakan `BrowserIdcAuthPlugin` dalam opsi konfigurasi `plugin_name`.

KeepAlive

- Nilai Default - 1
- Tipe Data - Boolean

Boolean yang menentukan apakah driver menggunakan TCP keepalives untuk mencegah koneksi dari timing out.

- 1 | BENAR: Driver menggunakan TCP keepalives untuk mencegah koneksi dari timing out.

- 0 | SALAH. Pengemudi tidak menggunakan keepalives TCP.

Parameter ini bersifat opsional.

KeepAliveCount

- Nilai Default - 0
- Tipe Data - Integer

Jumlah paket TCP keepalive yang dapat hilang sebelum koneksi dianggap rusak. Ketika parameter ini diatur ke 0, driver menggunakan sistem default untuk pengaturan ini.

Parameter ini bersifat opsional.

KeepAliveInterval

- Nilai Default - 0
- Tipe Data - Integer

Jumlah detik antara setiap transmisi ulang TCP keepalive. Ketika parameter ini diatur ke 0, driver menggunakan sistem default untuk pengaturan ini.

Parameter ini bersifat opsional.

KeepAliveTime

- Nilai Default - 0
- Tipe Data - Integer

Jumlah detik tidak aktif sebelum driver mengirim paket TCP keepalive. Ketika parameter ini diatur ke 0, driver menggunakan sistem default untuk pengaturan ini.

Parameter ini bersifat opsional.

listen_port

- Nilai Default - 7890
- Tipe Data - Integer

Port yang digunakan driver untuk menerima respons SAFL dari penyedia identitas atau kode otorisasi saat menggunakan layanan SAFL, Azure AD, atau AWS IAM Identity Center melalui plugin browser.

Parameter ini bersifat opsional.

login_url

- Nilai Default - Tidak Ada
- Tipe Data - String

URL untuk sumber daya di situs web penyedia identitas saat menggunakan plugin SAML Browser generik.

Parameter ini diperlukan jika mengautentikasi dengan layanan SAMP atau Azure AD melalui plugin browser.

loginToRp

- Nilai default - urn:amazon:webservices
- Tipe Data - String

Pihak yang mengandalkan percaya bahwa Anda ingin menggunakan untuk jenis otentikasi AD FS.

String ini opsional.

LogLevel

- Nilai Default - 0
- Tipe Data - Integer

Gunakan properti ini untuk mengaktifkan atau menonaktifkan logging di driver dan untuk menentukan jumlah detail yang disertakan dalam log. file. Kami menyarankan Anda hanya mengaktifkan logging cukup lama untuk menangkap masalah, karena logging mengurangi kinerja dan dapat menghabiskan sejumlah besar ruang disk.

Atur properti ke salah satu nilai berikut:

- 0: MATI. Nonaktifkan semua logging.

- 1: KESALAHAN. Mencatat peristiwa kesalahan yang memungkinkan driver untuk terus berjalan tetapi menghasilkan kesalahan.
- 2: API_CALL. Log panggilan fungsi API ODBC dengan nilai argumen fungsi.
- 3: INFO. Log informasi umum yang menggambarkan kemajuan pengemudi.
- 4: MSG_PROTOCOL. Log informasi rinci dari pesan pengemudi protocol.
- 5: DEBUG. Mencatat semua aktivitas pengemudi
- 6: DEBUG_APPEND. Terus tambahkan log untuk semua aktivitas pengemudi.

Saat logging diaktifkan, driver menghasilkan file log berikut di lokasi yang Anda tentukan di LogPath properti:

- `redshift_odbc.log.1` File yang mencatat aktivitas driver yang terjadi selama jabat tangan koneksi.
- `redshift_odbc.log` File untuk semua aktivitas driver setelah koneksi dibuat ke database.

Parameter ini bersifat opsional.

LogPath

- Nilai Default - Direktori TEMP khusus OS
- Tipe Data - String

Jalur lengkap ke folder tempat driver menyimpan file log saat LogLevel lebih tinggi dari 0.

Parameter ini bersifat opsional.

Min_tls

- Nilai Default - 1.2
- Tipe Data - String

Versi minimum dari driver TLS/SSL yang memungkinkan penyimpanan data untuk digunakan untuk mengenkripsi koneksi. Misalnya, jika TLS 1.2 ditentukan, TLS 1.1 tidak dapat digunakan untuk mengenkripsi koneksi.

Min_tls menerima nilai-nilai berikut:

- 1.0: Koneksi harus menggunakan setidaknya TLS 1.0.
- 1.1: Koneksi harus menggunakan setidaknya TLS 1.1.
- 1.2: Koneksi harus menggunakan setidaknya TLS 1.2.

Parameter ini bersifat opsional.

partner_spid

- Nilai Default - Tidak Ada
- Tipe Data - String

Nilai SPID mitra (ID penyedia layanan) yang akan digunakan saat mengautentikasi koneksi menggunakan layanan. PingFederate

Parameter ini bersifat opsional.

Kata Sandi | PWS

- Nilai Default - Tidak Ada
- Tipe Data - String

Kata sandi yang sesuai dengan nama pengguna database yang Anda berikan di bidang User (UID | User | LogonId).

Parameter ini bersifat opsional.

plugin_name

- Nilai Default - Tidak Ada
- Tipe Data - String

Nama plugin penyedia kredensial yang ingin Anda gunakan untuk otentikasi.

Nilai berikut didukung:

- ADFS: Gunakan Layanan Federasi Direktori Aktif untuk otentikasi.
- AzureAD: Gunakan Layanan Microsoft Azure Active Directory (AD) untuk otentikasi.

- `BrowserAzureAD`: Gunakan plugin browser untuk Layanan Microsoft Azure Active Directory (AD) untuk otentikasi.
- `BrowserIdcAuthPlugin` : Plugin otorisasi menggunakan AWS IAM Identity Center.
- `BrowserSAML`: Gunakan plugin browser untuk layanan SAMP seperti Okta atau Ping untuk otentikasi.
- `IdpTokenAuthPlugin`: Plugin otorisasi yang menerima token AWS IAM Identity Center atau OpenID Connect (OIDC) JSON berbasis token identitas (JWT) dari penyedia identitas web mana pun yang ditautkan ke IAM Identity Center. AWS
- `JWT`: Gunakan JSON Web Token (JWT) untuk otentikasi.
- `Ping`: Gunakan `PingFederate` layanan untuk otentikasi.
- `Okta`: Gunakan layanan Okta untuk otentikasi.

Parameter ini bersifat opsional.

Pelabuhan | `PortNumber`

- Nilai default - 5439
- Tipe Data - Integer

Jumlah port TCP yang digunakan server Amazon Redshift untuk mendengarkan koneksi klien.

Parameter ini bersifat opsional.

`preferred_role`

- Nilai Default - Tidak Ada
- Tipe Data - String

Peran yang ingin Anda asumsikan selama koneksi ke Amazon Redshift. Ini digunakan untuk otentikasi IAM.

Parameter ini bersifat opsional.

Profil

- Nilai Default - Tidak Ada
- Tipe Data - String

Nama AWS profil pengguna yang digunakan untuk mengautentikasi ke Amazon Redshift.

- Jika parameter Use Instance Profile (InstanceProfileproperty) disetel ke 1 | TRUE, setelah tersebut akan diutamakan dan driver menggunakan profil instans Amazon EC2 sebagai gantinya.
- Lokasi default untuk file kredensial yang berisi profil adalah. ~/.aws/Credentials Variabel AWS_SHARED_CREDENTIALS_FILE lingkungan dapat digunakan untuk menunjuk ke file kredensial yang berbeda.

Parameter ini bersifat opsional.

provider_name

- Nilai Default - Tidak Ada
- Tipe Data - String

Penyedia otentikasi yang dibuat oleh pengguna menggunakan kueri CREATE IDENTITY PROVIDER. Ini digunakan dalam otentikasi Amazon Redshift asli.

Parameter ini bersifat opsional.

ProxyHost

- Nilai Default - Tidak Ada
- Tipe Data - String

Nama host atau alamat IP dari server proxy yang ingin Anda sambungkan.

Parameter ini bersifat opsional.

ProxyPort

- Nilai Default - Tidak Ada
- Tipe Data - Integer

Jumlah port yang digunakan server proxy untuk mendengarkan koneksi klien.

Parameter ini bersifat opsional.

ProxyPwd

- Versi driver ValPrevious ODBC default — Tidak ada
- Tipe Data - String

Kata sandi yang Anda gunakan untuk mengakses server proxy.

Parameter ini bersifat opsional.

ProxyUid

- Nilai Default - Tidak Ada
- Tipe Data - String

Nama pengguna yang Anda gunakan untuk mengakses server proxy.

Parameter ini bersifat opsional.

ReadOnly

- Nilai Default - 0
- Tipe Data - Boolean

Boolean yang menentukan apakah driver dalam mode read-only.

- 1 | BENAR: Koneksi dalam mode hanya-baca, dan tidak dapat menulis ke penyimpanan data.
- 0 | SALAH: Koneksi tidak dalam mode read-only, dan dapat menulis ke penyimpanan data.

Parameter ini bersifat opsional.

region

- Nilai Default - Tidak Ada
- Tipe Data - String

AWS Wilayah tempat cluster Anda berada.

Parameter ini bersifat opsional.

SecretAccessKey

- Nilai Default - Tidak Ada
- Tipe Data - String

Kunci rahasia IAM untuk pengguna atau peran. Jika Anda mengatur parameter ini, Anda juga harus mengatur AccessKeyID.

Parameter ini bersifat opsional.

SessionToken

- Nilai Default - Tidak Ada
- Tipe Data - String

Token sesi IAM sementara yang terkait dengan peran IAM yang Anda gunakan untuk mengautentikasi.

Parameter ini bersifat opsional.

Server | HostName | Tuan rumah

- Nilai Default - Tidak Ada
- Tipe Data - String

Server endpoint untuk terhubung ke.

Parameter ini diperlukan.

ssl_insecure

- Nilai Default - 0
- Tipe Data - Boolean

Boolean yang menentukan apakah driver memeriksa keaslian sertifikat server iDP.

- 1 | BENAR: Driver tidak memeriksa keaslian sertifikat server iDP.
- 0 | FALSE: Driver memeriksa keaslian sertifikat server iDP

Parameter ini bersifat opsional.

SSLMode

- Nilai Default - `verify-ca`
- Tipe Data - String

Mode verifikasi sertifikat SSL untuk digunakan saat menghubungkan ke Amazon Redshift. Nilai-nilai berikut dimungkinkan:

- `verify-full`: Connect hanya menggunakan SSL, otoritas sertifikat terpercaya, dan nama server yang cocok dengan sertifikat.
- `verify-ca`: Connect hanya menggunakan SSL dan otoritas sertifikat terpercaya.
- `require`: Connect hanya menggunakan SSL.
- `prefer`: Connect menggunakan SSL jika tersedia. Jika tidak, sambungkan tanpa menggunakan SSL.
- `allow`: Secara default, sambungkan tanpa menggunakan SSL. Jika server membutuhkan koneksi SSL, maka gunakan SSL.
- `disable`: Connect tanpa menggunakan SSL.

Parameter ini bersifat opsional.

StsConnectionTimeout

- Nilai Default - 0
- Tipe Data - Integer

Waktu tunggu maksimum untuk koneksi IAM, dalam hitungan detik. Jika disetel ke 0 atau tidak ditentukan, driver menunggu 60 detik untuk setiap AWS STS panggilan.

Parameter ini bersifat opsional.

StsEndpointUrl

- Nilai Default - Tidak Ada
- Tipe Data - String

Opsi ini menentukan endpoint utama yang digunakan untuk berkomunikasi dengan (). AWS Security Token Service AWS STS

Parameter ini bersifat opsional.

token

- Nilai Default - Tidak Ada
- Tipe Data - String

Pusat AWS Identitas IAM menyediakan token akses atau OpenID Connect (OIDC) JSON Web Token (JWT) yang disediakan oleh penyedia identitas web yang terhubung dengan IAM Identity Center. AWS Aplikasi Anda harus menghasilkan token ini dengan mengautentikasi pengguna aplikasi Anda dengan AWS IAM Identity Center atau penyedia identitas yang ditautkan dengan AWS IAM Identity Center.

Parameter ini bekerja dengan `IdpTokenAuthPlugin`.

token_type

- Nilai Default - Tidak Ada
- Tipe Data - String

Jenis token yang digunakan di `IdpTokenAuthPlugin`.

Anda dapat menentukan salah satu nilai berikut:

`ACCESS_TOKEN`

Masukkan ini jika Anda menggunakan token akses yang disediakan Pusat Identitas AWS IAM.

`EXT_JWT`

Masukkan ini jika Anda menggunakan OpenID Connect (OIDC) JSON Web Token (JWT) yang disediakan oleh penyedia identitas berbasis web yang terintegrasi dengan IAM Identity Center.
AWS

Parameter ini bekerja dengan `IdpTokenAuthPlugin`.

UID | Pengguna | LogonID

- Nilai Default - Tidak Ada
- Tipe Data - String

Nama pengguna yang Anda gunakan untuk mengakses server Amazon Redshift.

Parameter ini diperlukan jika Anda menggunakan otentikasi database.

UseUnicode

- Nilai Default - 0
- Tipe Data - Boolean

Boolean yang menentukan apakah driver mengembalikan data Redshift sebagai Unicode atau tipe SQL biasa.

- 1 | BENAR: Driver mengembalikan tipe SQL lebar untuk tipe data karakter.
 - SQL_WCHAR dikembalikan bukan SQL_CHAR.
 - SQL_WVARCHAR dikembalikan bukan SQL_VARCHAR.
 - SQL_WLONGVARCHAR dikembalikan bukan SQL_LONGVARCHAR.
- 0 | FALSE: Driver mengembalikan tipe SQL normal untuk tipe data karakter.
 - SQL_CHAR dikembalikan bukan SQL_WCHAR.
 - SQL_VARCHAR dikembalikan bukan SQL_WVARCHAR.
 - SQL_LONGVARCHAR dikembalikan bukan SQL_WLONGVARCHAR.

Parameter ini bersifat opsional. Ini tersedia dalam versi driver 2.1.15 dan yang lebih baru.

web_identity_token

- Nilai Default - Tidak Ada
- Tipe Data - String

Token OAUTH yang disediakan oleh penyedia identitas. Ini digunakan dalam plugin JWT.

Parameter ini diperlukan jika Anda mengatur parameter `plugin_name` ke `BasicJwtCredentialsProvider`

Versi driver ODBC sebelumnya

Unduh versi driver Amazon Redshift ODBC versi 2.x sebelumnya hanya jika alat Anda memerlukan versi driver tertentu.

Menggunakan versi driver ODBC sebelumnya untuk Microsoft Windows

Berikut ini adalah versi sebelumnya dari driver Amazon Redshift ODBC versi 2.x untuk Microsoft Windows:

- <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>

Gunakan versi driver ODBC sebelumnya untuk Linux

Berikut ini adalah versi sebelumnya dari driver Amazon Redshift ODBC versi 2.x untuk Linux:

- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.14.0/AmazonRedshiftODBC-64-bit-2.1.14.0.x86_64.rpm -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 -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 -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 -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 -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 -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 -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 -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 -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 Di Wilayah Tiongkok-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 -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 -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 -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 -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/2.0.1.0/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/2.0.0.11/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/2.0.0.9/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/2.0.0.8/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/2.0.0.7/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/2.0.0.6/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/2.0.0.5/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/2.0.0.3/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 https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.1/AmazonRedshiftODBC-64-bit-2.0.0.1.x86_64.rpm

Menggunakan versi driver ODBC sebelumnya untuk Apple macOS

Berikut ini adalah versi sebelumnya dari driver Amazon Redshift ODBC versi 2.x untuk Apple macOS:

- <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>

Mengkonfigurasi koneksi driver ODBC versi 1.x

Anda dapat menggunakan koneksi ODBC untuk terhubung ke kluster Amazon Redshift Anda dari banyak alat dan aplikasi klien SQL pihak ketiga. Untuk melakukan ini, atur koneksi di komputer klien Anda atau instans Amazon EC2. Jika alat klien Anda mendukung JDBC, Anda mungkin memilih untuk menggunakan jenis koneksi itu daripada ODBC karena kemudahan konfigurasi yang disediakan JDBC. Namun, jika alat klien Anda tidak mendukung JDBC, ikuti langkah-langkah di bagian ini untuk mengonfigurasi koneksi ODBC.

Amazon Redshift menyediakan driver ODBC 64-bit untuk sistem operasi Linux, Windows, dan macOS X. Driver ODBC 32-bit dihentikan. Pembaruan lebih lanjut tidak akan dirilis, kecuali untuk patch keamanan yang mendesak.

[Untuk informasi terbaru tentang fungsionalitas dan prasyarat driver ODBC, lihat catatan rilis driver Amazon Redshift ODBC.](#)

Untuk informasi penginstalan dan konfigurasi untuk driver Amazon Redshift ODBC, lihat panduan pemasangan dan konfigurasi konektor [Amazon Redshift ODBC](#).

Topik

- [Mendapatkan URL ODBC](#)
- [Menggunakan driver Amazon Redshift ODBC di Microsoft Windows](#)
- [Menggunakan driver Amazon Redshift ODBC di Linux](#)
- [Menggunakan driver Amazon Redshift ODBC di macOS X](#)
- [Opsi driver ODBC](#)
- [Versi driver ODBC sebelumnya](#)

Mendapatkan URL ODBC

Amazon Redshift menampilkan URL ODBC untuk klaster Anda di konsol Amazon Redshift. URL ini berisi informasi untuk mengatur koneksi antara komputer klien Anda dan database.

URL ODBC memiliki format berikut:

`Driver={driver};Server=endpoint;Database=database_name;UID=user_name;PWD=password`

Bidang format yang ditunjukkan sebelumnya memiliki nilai berikut.

Bidang	Nilai
Driver	Nama driver ODBC 64-bit yang akan digunakan: Amazon Redshift (x64). Nama driver ODBC 32-bit: Amazon Redshift (x86).
Server	Titik akhir dari cluster Amazon Redshift.
Database	Database yang Anda buat untuk cluster Anda.
UID	Nama pengguna akun pengguna yang memiliki izin untuk terhubung ke database. Nilai ini adalah izin database, bukan izin Amazon Redshift, meskipun Anda dapat menggunakan akun pengguna admin yang Anda atur saat meluncurkan cluster.
PWD	Kata sandi untuk akun pengguna untuk terhubung ke database.
Port	Nomor port yang Anda tentukan saat Anda meluncurkan cluster. Jika Anda memiliki firewall, pastikan port ini terbuka untuk Anda gunakan.

Bidang dalam tabel sebelumnya dapat berisi karakter khusus berikut:

```
[ ] { } ( ) , ; ? * = ! @
```

Jika Anda menggunakan karakter khusus ini, Anda harus melampirkan nilainya dalam kurung kurawal. Misalnya, nilai kata sandi `Your;password123` dalam string koneksi direpresentasikan sebagai `PWD={Your;password123};`.

Karena `Field=value` pasangan dipisahkan oleh titik koma, kombinasi `}` dan `;` dengan sejumlah spasi di antaranya dianggap sebagai akhir dari pasangan. `Field={value};` Kami menyarankan

Anda menghindari urutan `};` dalam nilai bidang Anda. Misalnya, jika Anda menetapkan nilai kata sandi Anda sebagai `PWD={This is a passwor} ;d};`, kata sandi Anda akan menjadi `This is a passwor} ;` dan URL akan error.

Berikut ini adalah contoh URL ODBC.

```
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
```

Untuk informasi tentang cara mendapatkan koneksi ODBC Anda, lihat [Menemukan string koneksi cluster Anda](#).

Menggunakan driver Amazon Redshift ODBC di Microsoft Windows

Anda menginstal driver Amazon Redshift ODBC di komputer klien yang mengakses gudang data Amazon Redshift. Setiap komputer tempat Anda menginstal driver harus memenuhi daftar persyaratan sistem minimum. Untuk informasi tentang persyaratan sistem minimum, lihat panduan [pemasangan dan konfigurasi konektor Amazon Redshift ODBC](#).

Topik

- [Mengunduh dan menginstal driver Amazon Redshift ODBC](#)
- [Membuat entri DSN sistem untuk koneksi ODBC](#)

Mengunduh dan menginstal driver Amazon Redshift ODBC

Gunakan prosedur berikut untuk mengunduh driver Amazon Redshift ODBC untuk sistem operasi Windows. Hanya gunakan driver selain ini jika Anda menjalankan aplikasi pihak ketiga yang disertifikasi untuk digunakan dengan Amazon Redshift dan yang memerlukan driver tertentu.

Untuk menginstal driver ODBC

1. Unduh salah satu dari berikut ini, tergantung pada arsitektur sistem alat atau aplikasi klien SQL Anda:
 - [64-bit ODBC driver versi 1.6.3](#) driver versi 1.6.3.

Nama driver ini adalah Amazon Redshift (x64).

- [32-bit ODBC driver versi 1.4.52 Di AWS](#) [32-bit ODBC driver versi 1.4.52](#)

Nama driver ini adalah Amazon Redshift (x86). Driver ODBC 32-bit dihentikan. Pembaruan lebih lanjut tidak akan dirilis, kecuali untuk patch keamanan yang mendesak.

 Note

Unduh paket MSI yang sesuai dengan arsitektur sistem alat atau aplikasi klien SQL Anda. Misalnya, jika alat klien SQL Anda 64-bit, instal driver 64-bit.

Kemudian unduh dan tinjau perjanjian [SIM Amazon Redshift ODBC dan JDBC](#).

2. Klik dua kali file.msi, lalu ikuti langkah-langkah di wizard untuk menginstal driver.

Membuat entri DSN sistem untuk koneksi ODBC

Setelah Anda mengunduh dan menginstal driver ODBC, tambahkan entri nama sumber data (DSN) ke komputer klien atau instans Amazon EC2. Alat klien SQL menggunakan sumber data ini untuk terhubung ke database Amazon Redshift.

Kami menyarankan Anda membuat sistem DSN bukan DSN pengguna. Beberapa aplikasi memuat data menggunakan akun pengguna yang berbeda. Aplikasi ini mungkin tidak dapat mendeteksi pengguna DSNs yang dibuat di bawah akun pengguna lain.

 Note

Untuk autentikasi menggunakan kredensial AWS Identity and Access Management (IAM) atau kredensial penyedia identitas (iDP), diperlukan langkah-langkah tambahan. Untuk informasi selengkapnya, lihat [Langkah 5: Konfigurasi koneksi JDBC atau ODBC untuk menggunakan kredensial IAM](#).

Untuk informasi tentang cara membuat entri DSN sistem, lihat panduan [pemasangan dan konfigurasi konektor Amazon Redshift ODBC](#).

Untuk membuat entri DSN sistem untuk koneksi ODBC pada Windows

1. Di menu Start, buka Sumber Data ODBC.

Pastikan Anda memilih Administrator Sumber Data ODBC yang memiliki bitness yang sama dengan aplikasi klien yang Anda gunakan untuk terhubung ke Amazon Redshift.

2. Di Administrator Sumber Data ODBC, pilih tab Driver dan cari folder driver:
 - Driver Amazon Redshift ODBC (64-bit)
 - Driver Amazon Redshift ODBC (32-bit)
3. Pilih tab DSN Sistem untuk mengonfigurasi driver untuk semua pengguna di komputer, atau tab DSN Pengguna untuk mengonfigurasi driver hanya untuk akun pengguna Anda.
4. Pilih Tambahkan. Jendela Create New Data Source terbuka.
5. Pilih driver Amazon Redshift ODBC, lalu pilih Selesai. Jendela Pengaturan DSN Driver Amazon Redshift ODBC terbuka.
6. Di bawah Pengaturan Koneksi, masukkan informasi berikut:

Nama sumber data

Masukkan sebuah nama untuk sumber data. Anda dapat menggunakan nama apa pun yang ingin Anda identifikasi sumber data nanti saat Anda membuat koneksi ke cluster. Misalnya, jika Anda mengikuti Panduan Memulai Amazon Redshift, Anda dapat mengetik `exampleclusterdsn` agar mudah mengingat cluster yang Anda kaitkan dengan DSN ini.

Server

Tentukan titik akhir untuk cluster Amazon Redshift Anda. Anda dapat menemukan informasi ini di konsol Amazon Redshift di halaman detail cluster. Untuk informasi selengkapnya, lihat [Mengonfigurasi koneksi di Amazon Redshift](#).

Port

Masukkan nomor port yang digunakan database. Gunakan port yang cluster dikonfigurasi untuk digunakan ketika diluncurkan atau dimodifikasi.

Basis Data

Masukkan nama database Amazon Redshift. Jika Anda meluncurkan cluster Anda tanpa menentukan nama database, masukkan *dev*. Jika tidak, gunakan nama yang Anda pilih selama proses peluncuran. Jika Anda mengikuti Panduan Memulai Amazon Redshift, masukkan. *dev*

7. Di bawah Otentikasi, tentukan opsi konfigurasi untuk mengonfigurasi otentikasi standar atau IAM. Untuk informasi tentang opsi otentikasi, lihat “Mengonfigurasi Otentikasi di Windows” di Panduan Pemasangan dan Konfigurasi Konektor Amazon Redshift ODBC.
8. Di bawah Pengaturan SSL, tentukan nilai untuk hal berikut:

Otentikasi SSL

Pilih mode untuk menangani Secure Sockets Layer (SSL). Dalam lingkungan pengujian, Anda mungkin menggunakan *prefer*. Namun, untuk lingkungan produksi dan ketika pertukaran data yang aman diperlukan, gunakan *verify-ca* atau *verify-full*. Untuk informasi selengkapnya tentang penggunaan SSL di Windows, lihat “Mengonfigurasi Verifikasi SSL di Windows” di Panduan Pemasangan dan Konfigurasi Konektor Amazon Redshift ODBC.

9. Di bawah Opsi Tambahan, tentukan opsi tentang cara mengembalikan hasil kueri ke alat atau aplikasi klien SQL Anda. Untuk informasi selengkapnya, lihat “Mengonfigurasi Opsi Tambahan di Windows” di Panduan Pemasangan dan Konfigurasi Konektor Amazon Redshift ODBC.
10. Di Opsi Logging, tentukan nilai untuk opsi logging. Untuk informasi selengkapnya, lihat “Mengonfigurasi Opsi Pencatatan di Windows” di Panduan Pemasangan dan Konfigurasi Konektor Amazon Redshift ODBC.

Lalu pilih OK.

11. Di bawah Opsi Tipe Data, tentukan nilai untuk tipe data. Untuk informasi selengkapnya, lihat “Mengonfigurasi Opsi Jenis Data di Windows” di Panduan Pemasangan dan Konfigurasi Konektor Amazon Redshift ODBC.

Lalu pilih OK.

12. Pilih Uji. Jika komputer klien dapat terhubung ke database Amazon Redshift, Anda melihat pesan berikut: Koneksi berhasil.

Jika komputer klien gagal terhubung ke database, Anda dapat memecahkan masalah yang mungkin terjadi. Untuk informasi selengkapnya, lihat [Memecahkan masalah koneksi di Amazon Redshift](#).

13. Konfigurasi keepalives TCP di Windows untuk mencegah koneksi dari waktu habis. Untuk informasi tentang cara mengonfigurasi keepalives TCP di Windows, lihat [Panduan Pemasangan dan Konfigurasi Konektor Amazon Redshift ODBC](#).
14. Untuk membantu pemecahan masalah, konfigurasi logging. Untuk informasi tentang cara mengonfigurasi logging di Windows, lihat [Panduan Instalasi dan Konfigurasi Konektor Amazon Redshift ODBC](#).

Menggunakan driver Amazon Redshift ODBC di Linux

Anda menginstal driver Amazon Redshift ODBC di komputer klien yang mengakses gudang data Amazon Redshift. Setiap komputer tempat Anda menginstal driver harus memenuhi daftar persyaratan sistem minimum. Untuk informasi tentang persyaratan sistem minimum, lihat [panduan pemasangan dan konfigurasi konektor Amazon Redshift ODBC](#).

Topik

- [Mengunduh dan menginstal driver Amazon Redshift ODBC](#)
- [Menggunakan manajer driver ODBC untuk mengkonfigurasi driver](#)

Mengunduh dan menginstal driver Amazon Redshift ODBC

Gunakan langkah-langkah di bagian ini untuk mengunduh dan menginstal driver Amazon Redshift ODBC pada distribusi Linux yang didukung. Proses instalasi menginstal file driver di direktori berikut:

- `/opt/amazon/redshiftodbc/lib/64`(untuk driver 64-bit)
- `/opt/amazon/redshiftodbc/ErrorMessage`s
- `/opt/amazon/redshiftodbc/Setup`
- `/opt/amazon/redshiftodbc/lib/32`(untuk driver 32-bit)

Untuk menginstal driver Amazon Redshift ODBC

1. Unduh salah satu dari berikut ini, tergantung pada arsitektur sistem alat atau aplikasi klien SQL Anda:
 - [Versi driver RPM 64-bit 1.6.3](#) versi 1.6.3.
 - [64-bit Debian driver versi 1.6.3](#) versi 1.6.3.
 - [Versi driver 32-bit 1.4.52](#) versi 1.6.3.

Nama untuk masing-masing driver ini adalah driver Amazon Redshift ODBC. Driver ODBC 32-bit dihentikan. Pembaruan lebih lanjut tidak akan dirilis, kecuali untuk patch keamanan yang mendesak.

 Note

Unduh paket yang sesuai dengan arsitektur sistem alat atau aplikasi klien SQL Anda. Misalnya, jika alat klien Anda 64-bit, instal driver 64-bit.

Kemudian unduh dan tinjau perjanjian [SIM Amazon Redshift ODBC dan JDBC](#).

2. Pergi ke lokasi di mana Anda mengunduh paket, dan kemudian jalankan salah satu perintah berikut. Gunakan perintah yang sesuai dengan distribusi Linux Anda.
- Pada sistem operasi RHEL dan CentOS, jalankan perintah berikut.

```
yum -nogpgcheck localinstall RPMFileName
```

Ganti *RPMFileName* dengan nama file paket RPM. Misalnya, perintah berikut menunjukkan menginstal driver 64-bit.

```
yum -nogpgcheck localinstall AmazonRedshiftODBC-64-bit-1.x.xx.xxxx-x.x86_64.rpm
```

- Pada SLES, jalankan perintah berikut.

```
zypper install RPMFileName
```

Ganti *RPMFileName* dengan nama file paket RPM. Misalnya, perintah berikut menunjukkan menginstal driver 64-bit.

```
zypper install AmazonRedshiftODBC-1.x.x.xxxx-x.x86_64.rpm
```

- Pada Debian, jalankan perintah berikut.

```
sudo apt install ./DEBFileName.deb
```

Ganti *DEBFileName.deb* dengan nama file paket Debian. Misalnya, perintah berikut menunjukkan menginstal driver 64-bit.

```
sudo apt install ./AmazonRedshiftODBC-1.x.x.xxxx-x.x86_64.deb
```

 Important

Setelah Anda selesai menginstal driver, konfigurasikan untuk digunakan pada sistem Anda. Untuk informasi selengkapnya tentang konfigurasi driver, lihat [Menggunakan manajer driver ODBC untuk mengkonfigurasi driver](#).

Menggunakan manajer driver ODBC untuk mengkonfigurasi driver

Pada sistem operasi Linux, Anda menggunakan manajer driver ODBC untuk mengonfigurasi pengaturan koneksi ODBC. Manajer driver ODBC menggunakan file konfigurasi untuk menentukan dan mengkonfigurasi sumber data dan driver ODBC. Manajer driver ODBC yang Anda gunakan bergantung pada sistem operasi yang Anda gunakan. Untuk Linux, ini adalah manajer driver UnixODBC.

Untuk informasi selengkapnya tentang manajer driver ODBC yang didukung untuk mengonfigurasi driver Amazon Redshift ODBC, [Menggunakan driver Amazon Redshift ODBC di Linux](#) lihat untuk sistem operasi Linux. Juga, lihat “Menentukan Manajer Driver ODBC di Mesin Non-Windows” di panduan pemasangan dan [konfigurasi konektor Amazon Redshift ODBC](#).

Tiga file diperlukan untuk mengonfigurasi driver Amazon Redshift ODBC: `amazon.redshiftodbc.ini`, `odbc.ini`, dan `odbcinst.ini`.

Jika Anda menginstal ke lokasi default, file `amazon.redshiftodbc.ini` konfigurasi terletak di salah satu direktori berikut:

- `/opt/amazon/redshiftodbc/lib/64` (untuk driver 64-bit pada sistem operasi Linux)
- `/opt/amazon/redshiftodbc/lib/32` (untuk driver 32-bit pada sistem operasi Linux)

Selain itu, `/opt/amazon/redshiftodbc/Setup` di Linux, ada sampel `odbc.ini` dan `odbcinst.ini` file. Anda dapat menggunakan file-file ini sebagai contoh untuk mengonfigurasi driver Amazon Redshift ODBC dan nama sumber data (DSN).

Kami tidak menyarankan menggunakan direktori instalasi driver Amazon Redshift ODBC untuk file konfigurasi. File sampel dalam Setup direktori hanya untuk tujuan contoh. Jika Anda menginstal ulang driver Amazon Redshift ODBC di lain waktu, atau meningkatkan ke versi yang lebih baru, direktori instalasi akan ditimpa. Anda kemudian kehilangan perubahan apa pun yang mungkin Anda buat pada file-file itu.

Untuk menghindari hal ini, salin `amazon.redshiftodbc.ini` file ke direktori selain direktori instalasi. Jika Anda menyalin file ini ke direktori home pengguna, tambahkan titik (.) ke awal nama file untuk menjadikannya file tersembunyi.

Untuk `odbcinst.ini` file `odbc.ini` dan, gunakan file konfigurasi di direktori home pengguna atau buat versi baru di direktori lain. Secara default, sistem operasi Linux Anda harus memiliki `odbc.ini` file dan `odbcinst.ini` file di direktori home pengguna (`/home/$USER` atau `~/.`). File default ini adalah file tersembunyi, yang ditunjukkan oleh titik (.) di depan setiap nama file. File-file ini hanya ditampilkan ketika Anda menggunakan `-a` bendera untuk daftar isi direktori.

Opsi apa pun yang Anda pilih untuk `odbcinst.ini` file `odbc.ini` dan, ubah file untuk menambahkan informasi konfigurasi driver dan DSN. Jika Anda membuat file baru, Anda juga perlu mengatur variabel lingkungan untuk menentukan di mana file konfigurasi ini berada.

Secara default, manajer driver ODBC dikonfigurasi untuk menggunakan versi tersembunyi dari file `odbc.ini` dan `odbcinst.ini` konfigurasi (bernama `odbc.in` dan `odbcinst.ini`) terletak di direktori home. Mereka juga dikonfigurasi untuk menggunakan `amazon.redshiftodbc.ini` file di `/lib` subfolder direktori instalasi driver. Jika Anda menyimpan file konfigurasi ini di tempat lain, atur variabel lingkungan yang dijelaskan berikut sehingga manajer driver dapat menemukan file. Untuk informasi selengkapnya, lihat “Menentukan Lokasi File Konfigurasi Driver” di panduan [pemasangan dan konfigurasi konektor Amazon Redshift ODBC](#).

Membuat nama sumber data pada sistem operasi Linux

Saat menghubungkan ke penyimpanan data Anda menggunakan nama sumber data (DSN), konfigurasikan `odbc.ini` file yang akan ditentukan DSNs. Tetapkan properti dalam `odbc.ini` file untuk membuat DSN yang menentukan informasi koneksi untuk penyimpanan data Anda.

Untuk informasi tentang cara mengonfigurasi `odbc.ini` file, lihat “Membuat Nama Sumber Data pada Mesin Non-Windows” di panduan [pemasangan dan konfigurasi konektor Amazon Redshift ODBC](#) ODBC

Gunakan format berikut pada sistem operasi Linux.

```
[ODBC Data Sources]
driver_name=dsn_name

[dsn_name]
Driver=path/driver_file

Host=cluster_endpoint
Port=port_number
Database=database_name
locale=locale
```

Contoh berikut menunjukkan konfigurasi untuk `odbc.ini` dengan driver ODBC 64-bit pada sistem operasi Linux.

```
[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
```

Contoh berikut menunjukkan konfigurasi untuk `odbc.ini` dengan driver ODBC 32-bit pada sistem operasi Linux.

```
[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
```

Mengkonfigurasi koneksi tanpa DSN pada sistem operasi Linux

Untuk terhubung ke penyimpanan data Anda melalui koneksi yang tidak memiliki DSN, tentukan driver dalam `odbcinst.ini` file. Kemudian berikan string koneksi tanpa DSN di aplikasi Anda.

Untuk informasi tentang cara mengonfigurasi `odbcinst.ini` file dalam kasus ini, lihat “Mengonfigurasi Koneksi Tanpa DSN pada Mesin Non-Windows” di panduan pemasangan dan konfigurasi konektor Amazon Redshift [ODBC](#).

Gunakan format berikut pada sistem operasi Linux.

```
[ODBC Drivers]
driver_name=Installed
...

[driver_name]
Description=driver_description
Driver=path/driver_file
...
```

Contoh berikut menunjukkan `odbcinst.ini` konfigurasi untuk driver 64-bit yang diinstal di direktori default pada sistem operasi Linux.

```
[ODBC Drivers]
Amazon Redshift (x64)=Installed

[Amazon Redshift (x64)]
Description=Amazon Redshift ODBC Driver (64-bit)
Driver=/opt/amazon/redshiftodbc/lib/64/libamazonredshiftodbc64.so
```

Contoh berikut menunjukkan `odbcinst.ini` konfigurasi untuk driver 32-bit yang diinstal di direktori default pada sistem operasi Linux.

```
[ODBC Drivers]
Amazon Redshift (x86)=Installed

[Amazon Redshift (x86)]
Description=Amazon Redshift ODBC Driver (32-bit)
Driver=/opt/amazon/redshiftodbc/lib/32/libamazonredshiftodbc32.so
```

Mengonfigurasi variabel lingkungan

Gunakan manajer driver ODBC yang benar untuk memuat driver yang benar. Untuk melakukan ini, atur variabel lingkungan jalur perpustakaan. Untuk informasi selengkapnya, lihat “Menentukan

Manajer Driver ODBC di Mesin Non-Windows” di panduan pemasangan dan konfigurasi [konektor Amazon Redshift ODBC](#).

Secara default, manajer driver ODBC dikonfigurasi untuk menggunakan versi tersembunyi dari file `odbc.ini` dan `odbcinst.ini` konfigurasi (bernama `odbc.inidan.odbcinst.ini`) terletak di direktori home. Mereka juga dikonfigurasi untuk menggunakan `amazon.redshiftoDBC.ini` file di `/lib` subfolder direktori instalasi driver. Jika Anda menyimpan file konfigurasi ini di tempat lain, variabel lingkungan sehingga manajer driver dapat menemukan file. Untuk informasi selengkapnya, lihat “Menentukan Lokasi File Konfigurasi Driver” di Panduan Pemasangan dan Konfigurasi Konektor Amazon Redshift ODBC.

Mengkonfigurasi fitur koneksi

Anda dapat mengonfigurasi fitur koneksi berikut untuk pengaturan ODBC Anda:

- Konfigurasi driver ODBC untuk memberikan kredensi dan mengautentikasi koneksi ke database Amazon Redshift.
- Konfigurasi driver ODBC untuk menyambung ke soket yang diaktifkan dengan Secure Sockets Layer (SSL), jika Anda terhubung ke server Amazon Redshift yang mengaktifkan SSL.
- Konfigurasi driver ODBC untuk terhubung ke Amazon Redshift melalui server proxy.
- Konfigurasi driver ODBC untuk menggunakan mode pemrosesan kueri untuk mencegah kueri menghabiskan terlalu banyak memori.
- Konfigurasi driver ODBC untuk meneruskan proses otentikasi IAM melalui server proxy.
- Konfigurasi driver ODBC untuk menggunakan keepalives TCP untuk mencegah koneksi dari waktu habis.

Untuk informasi tentang fitur koneksi ini, lihat panduan [pemasangan dan konfigurasi konektor Amazon Redshift ODBC](#).

Menggunakan driver Amazon Redshift ODBC di macOS X

Anda menginstal driver di komputer klien yang mengakses gudang data Amazon Redshift. Setiap komputer tempat Anda menginstal driver harus memenuhi daftar persyaratan sistem minimum. Untuk informasi tentang persyaratan sistem minimum, lihat panduan [pemasangan dan konfigurasi konektor Amazon Redshift ODBC](#).

Topik

- [Mengunduh dan menginstal driver Amazon Redshift ODBC](#)

- [Gunakan manajer driver ODBC untuk mengonfigurasi driver](#)

Mengunduh dan menginstal driver Amazon Redshift ODBC

Gunakan langkah-langkah di bagian ini untuk mengunduh dan menginstal driver Amazon Redshift ODBC pada versi macOS X yang didukung. Proses instalasi menginstal file driver di direktori berikut:

- /opt/amazon/redshift/lib/universal
- /opt/amazon/redshift/ErrorMessage
- /opt/amazon/redshift/Setup

Untuk menginstal driver Amazon Redshift ODBC di macOS X

1. [Untuk menginstal driver Amazon Redshift ODBC di macOS X, unduh driver macOS versi 1.6.3.](#)

Kemudian unduh dan tinjau perjanjian [SIM Amazon Redshift ODBC dan JDBC](#).

2. Klik dua kali AmazonRedshiftODBC.pkg untuk menjalankan installer.
3. Ikuti langkah-langkah di installer untuk menyelesaikan proses instalasi driver. Untuk melakukan instalasi, setuju ketentuan perjanjian lisensi.

Important

Setelah Anda selesai menginstal driver, konfigurasikan untuk digunakan pada sistem Anda. Untuk informasi selengkapnya tentang konfigurasi driver, lihat [Gunakan manajer driver ODBC untuk mengonfigurasi driver](#).

Gunakan manajer driver ODBC untuk mengonfigurasi driver

Pada sistem operasi macOS X, Anda menggunakan pengelola driver ODBC untuk mengonfigurasi setelan koneksi ODBC. Manajer driver ODBC menggunakan file konfigurasi untuk menentukan dan mengkonfigurasi sumber data dan driver ODBC. Manajer driver ODBC yang Anda gunakan bergantung pada sistem operasi yang Anda gunakan. Untuk sistem operasi macOS X, ini adalah manajer driver lodBC.

Untuk informasi selengkapnya tentang pengelola driver ODBC yang didukung untuk mengonfigurasi driver Amazon Redshift ODBC, lihat [Menggunakan driver Amazon Redshift ODBC di macOS X](#) untuk

sistem operasi macOS X. Juga, lihat “Menentukan Manajer Driver ODBC di Mesin Non-Windows” di panduan pemasangan dan [konfigurasi konektor Amazon Redshift ODBC](#).

Tiga file diperlukan untuk mengonfigurasi driver Amazon Redshift ODBC: `amazon.redshiftodbc.ini`, `odbc.ini`, dan `odbcinst.ini`.

Jika Anda menginstal ke lokasi default, file `amazon.redshiftodbc.ini` konfigurasi terletak di `/opt/amazon/redshift/lib`.

Selain itu, `/opt/amazon/redshift/Setup` di bawah macOS X, ada sampel `odbc.ini` dan `odbcinst.ini` file. Anda dapat menggunakan file-file ini sebagai contoh untuk mengonfigurasi driver Amazon Redshift ODBC dan nama sumber data (DSN).

Kami tidak menyarankan menggunakan direktori instalasi driver Amazon Redshift ODBC untuk file konfigurasi. File sampel dalam `Setup` direktori hanya untuk tujuan contoh. Jika Anda menginstal ulang driver Amazon Redshift ODBC di lain waktu, atau meningkatkan ke versi yang lebih baru, direktori instalasi akan ditimpa. Anda kemudian kehilangan perubahan apa pun yang mungkin Anda buat pada file-file itu.

Untuk menghindari hal ini, salin `amazon.redshiftodbc.ini` file ke direktori selain direktori instalasi. Jika Anda menyalin file ini ke direktori home pengguna, tambahkan titik (.) ke awal nama file untuk menjadikannya file tersembunyi.

Untuk `odbcinst.ini` file `odbc.ini` dan, gunakan file konfigurasi di direktori home pengguna atau buat versi baru di direktori lain. Secara default, sistem operasi macOS X Anda harus memiliki `odbc.ini` file dan `odbcinst.ini` file di direktori home pengguna (`/home/$USER` atau `~/.`). File default ini adalah file tersembunyi, yang ditunjukkan oleh titik (.) di depan setiap nama file. File-file ini hanya ditampilkan ketika Anda menggunakan `-a` bendera untuk daftar isi direktori.

Opsi apa pun yang Anda pilih untuk `odbcinst.ini` file `odbc.ini` dan, ubah file untuk menambahkan informasi konfigurasi driver dan DSN. Jika Anda membuat file baru, Anda juga perlu mengatur variabel lingkungan untuk menentukan di mana file konfigurasi ini berada.

Secara default, manajer driver ODBC dikonfigurasi untuk menggunakan versi tersembunyi dari file `odbc.ini` dan `odbcinst.ini` konfigurasi (bernama `odbc.in` dan `odbcinst.in`) terletak di direktori home. Mereka juga dikonfigurasi untuk menggunakan `amazon.redshiftodbc.ini` file di `/lib` subfolder direktori instalasi driver. Jika Anda menyimpan file konfigurasi ini di tempat lain, atur variabel lingkungan yang dijelaskan berikut sehingga manajer driver dapat menemukan file. Untuk informasi selengkapnya, lihat “Menentukan Lokasi File Konfigurasi Driver” di panduan [pemasangan dan konfigurasi konektor Amazon Redshift ODBC](#).

Membuat nama sumber data sistem operasi macOS X

Saat menghubungkan ke penyimpanan data Anda menggunakan nama sumber data (DSN), konfigurasi `odbc.ini` file yang akan ditentukan DSNs. Tetapkan properti dalam `odbc.ini` file untuk membuat DSN yang menentukan informasi koneksi untuk penyimpanan data Anda.

Untuk informasi tentang cara mengonfigurasi `odbc.ini` file, lihat “Membuat Nama Sumber Data pada Mesin Non-Windows” di panduan [pemasangan dan konfigurasi konektor Amazon Redshift ODBC](#) ODBC

Gunakan format berikut pada sistem operasi macOS X.

```
[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
```

Contoh berikut menunjukkan konfigurasi untuk `odbc.ini` pada sistem operasi macOS X.

```
[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
```

Mengonfigurasi koneksi tanpa DSN pada sistem operasi macOS X

Untuk terhubung ke penyimpanan data Anda melalui koneksi yang tidak memiliki DSN, tentukan driver dalam `odbcinst.ini` file. Kemudian berikan string koneksi tanpa DSN di aplikasi Anda.

Untuk informasi tentang cara mengonfigurasi `odbcinst.ini` file dalam kasus ini, lihat “Mengonfigurasi Koneksi Tanpa DSN pada Mesin Non-Windows” di panduan pemasangan dan konfigurasi konektor Amazon Redshift [ODBC](#).

Gunakan format berikut pada sistem operasi macOS X.

```
[ODBC Drivers]
driver_name=Installed
...

[driver_name]
Description=driver_description
Driver=path/lib/amazonredshiftodbc.dylib

...
```

Contoh berikut menunjukkan `odbcinst.ini` konfigurasi untuk driver yang diinstal di direktori default pada sistem operasi macOS X.

```
[ODBC Drivers]
Amazon RedshiftODBC DSN=Installed

[Amazon RedshiftODBC DSN]
Description=Amazon Redshift ODBC Driver for macOS X
Driver=/opt/amazon/redshift/lib/amazonredshiftodbc.dylib
```

Mengonfigurasi variabel lingkungan

Gunakan manajer driver ODBC yang benar untuk memuat driver yang benar. Untuk melakukan ini, atur variabel lingkungan jalur perpustakaan. Untuk informasi selengkapnya, lihat “Menentukan Manajer Driver ODBC di Mesin Non-Windows” di panduan pemasangan dan konfigurasi [konektor Amazon Redshift ODBC](#).

Secara default, manajer driver ODBC dikonfigurasi untuk menggunakan versi tersembunyi dari file `odbc.ini` dan `odbcinst.ini` konfigurasi (bernama `odbc.inidan. odbcinst.ini`) terletak di direktori home. Mereka juga dikonfigurasi untuk menggunakan `amazon.redshiftodbc.ini` file di `/lib` subfolder direktori instalasi driver. Jika Anda menyimpan file konfigurasi ini di tempat lain, variabel lingkungan sehingga manajer driver dapat menemukan file. Untuk informasi selengkapnya, lihat “Menentukan Lokasi File Konfigurasi Driver” di Panduan Pemasangan dan Konfigurasi Konektor Amazon Redshift ODBC.

Mengkonfigurasi fitur koneksi

Anda dapat mengonfigurasi fitur koneksi berikut untuk pengaturan ODBC Anda:

- Konfigurasikan driver ODBC untuk memberikan kredensi dan mengautentikasi koneksi ke database Amazon Redshift.
- Konfigurasikan driver ODBC untuk menyambung ke soket yang diaktifkan dengan Secure Sockets Layer (SSL), jika Anda terhubung ke server Amazon Redshift yang mengaktifkan SSL.
- Konfigurasikan driver ODBC untuk terhubung ke Amazon Redshift melalui server proxy.
- Konfigurasikan driver ODBC untuk menggunakan mode pemrosesan kueri untuk mencegah kueri menghabiskan terlalu banyak memori.
- Konfigurasikan driver ODBC untuk meneruskan proses otentikasi IAM melalui server proxy.
- Konfigurasikan driver ODBC untuk menggunakan keepalives TCP untuk mencegah koneksi dari waktu habis.

Untuk informasi tentang fitur koneksi ini, lihat panduan [pemasangan dan konfigurasi konektor Amazon Redshift ODBC](#).

Opsi driver ODBC

Anda dapat menggunakan opsi konfigurasi untuk mengontrol perilaku driver Amazon Redshift ODBC.

Di Microsoft Windows, Anda biasanya mengatur opsi driver saat mengonfigurasi nama sumber data (DSN). Anda juga dapat mengatur opsi driver dalam string koneksi saat Anda terhubung secara terprogram, atau dengan menambahkan atau mengubah kunci registri. `HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\your_DSN` Untuk informasi selengkapnya tentang mengonfigurasi DSN, lihat. [Menggunakan driver Amazon Redshift ODBC di Microsoft Windows](#)

Di macOS X, Anda mengatur opsi konfigurasi driver di `amazon.redshiftodbc.ini` file `odbc.ini` dan file Anda, seperti yang dijelaskan dalam. [Gunakan manajer driver ODBC untuk mengonfigurasi driver](#) Opsi konfigurasi yang diatur dalam `amazon.redshiftodbc.ini` file berlaku untuk semua koneksi. Sebaliknya, opsi konfigurasi yang diatur dalam `odbc.ini` file khusus untuk koneksi. Opsi konfigurasi yang disetel lebih `odbc.ini` diutamakan daripada opsi konfigurasi yang ditetapkan. `amazon.redshiftodbc.ini`

Untuk informasi tentang cara mengatur opsi konfigurasi driver ODBC, lihat panduan [pemasangan dan konfigurasi konektor Amazon Redshift ODBC](#).

Versi driver ODBC sebelumnya

Unduh versi driver Amazon Redshift ODBC sebelumnya hanya jika alat Anda memerlukan versi driver tertentu.

Versi driver ODBC sebelumnya untuk Windows

Berikut ini adalah driver 64-bit:

- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.6.3.1006/AmazonRedshiftODBC64-1.6.3.1006.msi> Di Wilayah AWS
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.6.1.1000/AmazonRedshiftODBC64-1.6.1.1000.msi> Di Wilayah AWS
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.5.20.1024/AmazonRedshiftODBC64-1.5.20.1024.msi> Di Wilayah AWS
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.5.16.1019/AmazonRedshiftODBC64-1.5.16.1019.msi> Di Wilayah AWS
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.5.9.1011/AmazonRedshiftODBC64-1.5.9.1011.msi> Di Wilayah AWS
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.5.7.1007/AmazonRedshiftODBC64-1.5.7.1007.msi> Di Wilayah AWS
- <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>

Driver 32-bit dihentikan dan versi sebelumnya tidak didukung.

Versi driver ODBC sebelumnya untuk Linux

Berikut ini adalah versi driver 64-bit:

- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.6.3.1006/AmazonRedshiftODBC-64-bit-1.6.3.1006-1.x86_64.rpm -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 -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 -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 -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 -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 -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 -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 -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 -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 -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

Driver 32-bit dihentikan dan versi sebelumnya tidak didukung.

Versi driver ODBC sebelumnya untuk macOS X

Berikut ini adalah versi driver Amazon Redshift ODBC untuk macOS X:

- <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> -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 -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 -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 -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 -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> -1.4.65.1000.dmg
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.62.1000/AmazonRedshiftODBC-1.4.62.1000.dmg> -1.4.62.1000.dmg
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.59.1000/AmazonRedshiftODBC-1.4.59.1000.dmg> -1.4.59.1000.dmg
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.56.1000/AmazonRedshiftODBC-1.4.56.1000.dmg> -1.4.56.1000.dmg
- <https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.52.1000/AmazonRedshiftODBC-1.4.52.1000.dmg> -1.4.52.1000.dmg

Mengkonfigurasi opsi keamanan untuk koneksi

Amazon Redshift mendukung koneksi Secure Sockets Layer (SSL) untuk mengenkripsi data dan sertifikat server untuk memvalidasi sertifikat server yang terhubung dengan klien.

SSL

Untuk mendukung koneksi SSL, Amazon Redshift membuat dan menginstal sertifikat SSL ([ACM](#))AWS Certificate Manager yang dikeluarkan pada setiap cluster. Sertifikat ACM dipercaya publik oleh sebagian besar sistem operasi, browser web, dan klien. Anda mungkin perlu mengunduh bundel sertifikat jika klien atau aplikasi SQL Anda terhubung ke Amazon Redshift menggunakan SSL dengan `sslmode` opsi koneksi disetel `require` `verify-ca`, atau `verify-full`. Jika klien Anda memerlukan sertifikat, Amazon Redshift menyediakan sertifikat bundel sebagai berikut:

- Unduh bundel dari <https://s3.amazonaws.com/redshift-downloads/amazon-trust-ca-bundle.crt>.
 - Nomor MD5 checksum yang diharapkan adalah `418dea9b6d5d5de7a8f1ac42e164cdcf`.
 - Nomor checksum sha256 adalah `36dba8e4b8041cd14b9d60158893963301bcbb92e1c456847784de2acb5bd550`.

Jangan gunakan bundel sertifikat sebelumnya yang berada di `https://s3.amazonaws.com/redshift-downloads/redshift-ca-bundle.crt`.

- Di AWS Region Tiongkok, unduh bundel <https://s3.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/amazon-trust-ca-bundle.crt>.
 - Nomor MD5 checksum yang diharapkan adalah `418dea9b6d5d5de7a8f1ac42e164cdcf`.
 - Nomor checksum sha256 adalah `36dba8e4b8041cd14b9d60158893963301bcbb92e1c456847784de2acb5bd550`.

Jangan gunakan bundel sertifikat sebelumnya yang berada di `https://s3.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/redshift-ca-bundle.crt` dan `https://s3.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/redshift-ssl-ca-cert.pem`

Important

Amazon Redshift telah mengubah cara sertifikat SSL dikelola. Anda mungkin perlu memperbarui sertifikat CA root kepercayaan Anda saat ini untuk terus terhubung ke cluster Anda menggunakan SSL. Untuk informasi selengkapnya, lihat [Transisi ke sertifikat ACM untuk koneksi SSL](#).

Secara default, database cluster menerima koneksi apakah menggunakan SSL atau tidak. Untuk mengonfigurasi klaster Anda agar memerlukan koneksi SSL, atur `require_ssl` parameter ke `true` dalam grup parameter yang terkait dengan cluster.

Amazon Redshift mendukung mode SSL yang sesuai dengan Federal Information Processing Standard (FIPS) 140-2. Mode SSL yang sesuai dengan FIPS dinonaktifkan secara default.

 Important

Aktifkan mode SSL yang sesuai dengan FIPS hanya jika sistem Anda harus sesuai dengan FIPS.

Untuk mengaktifkan mode SSL yang sesuai dengan FIPS, setel `use_fips_ssl` parameter dan parameter ke `true` dalam grup `require_ssl` parameter yang terkait dengan cluster Amazon Redshift atau grup kerja Redshift Serverless. Untuk informasi tentang memodifikasi grup parameter pada klaster, lihat [Grup parameter Amazon Redshift](#). Untuk informasi tentang memodifikasi grup parameter pada workgroup, lihat [Mengonfigurasi koneksi SSL yang sesuai dengan FIPS ke Amazon Redshift Tanpa Server](#).

Amazon Redshift mendukung protokol perjanjian kunci Elliptic Curve Diffie—Hellman Ephemeral (ECDHE). Dengan ECDHE, klien dan server masing-masing memiliki elliptic curve public-private key pair yang digunakan untuk membangun rahasia bersama melalui saluran yang tidak aman. Anda tidak perlu mengonfigurasi apa pun di Amazon Redshift untuk mengaktifkan ECDHE. Jika Anda terhubung dari alat klien SQL yang menggunakan ECDHE untuk mengenkripsi komunikasi antara klien dan server, Amazon Redshift menggunakan daftar sandi yang disediakan untuk membuat koneksi yang sesuai. [Untuk informasi lebih lanjut, lihat Elliptic curve diffie—hellman di Wikipedia dan Ciphers di situs web OpenSSL.](#)

SSL dan percaya sertifikat CA di ODBC

Jika Anda terhubung menggunakan driver Amazon Redshift ODBC terbaru (versi 1.3.7.1000 atau yang lebih baru), Anda dapat melewati bagian ini. Untuk mengunduh driver terbaru, lihat [Mengonfigurasi koneksi untuk driver ODBC versi 2.x untuk Amazon Redshift](#).

Anda mungkin perlu memperbarui sertifikat CA root kepercayaan Anda saat ini untuk terus terhubung ke cluster Anda menggunakan SSL. Untuk informasi selengkapnya, lihat [SSL](#).

Anda dapat memverifikasi bahwa sertifikat yang Anda unduh cocok dengan nomor MD5 checksum yang diharapkan. Untuk melakukan ini, Anda dapat menggunakan program MD5sum pada sistem operasi Linux, atau alat lain pada sistem operasi Windows dan macOS X.

ODBC DSNs berisi `sslmode` pengaturan yang menentukan cara menangani enkripsi untuk koneksi klien dan verifikasi sertifikat server. Amazon Redshift mendukung `sslmode` nilai-nilai berikut dari koneksi klien:

- `disable`

SSL dinonaktifkan dan koneksi tidak dienkripsi.

- `allow`

SSL digunakan jika server membutuhkannya.

- `prefer`

SSL digunakan jika server mendukungnya. Amazon Redshift mendukung SSL, jadi SSL digunakan saat Anda menyetelnya. `sslmode prefer`

- `require`

SSL diperlukan.

- `verify-ca`

SSL harus digunakan dan sertifikat server harus diverifikasi.

- `verify-full`

SSL harus digunakan. Sertifikat server harus diverifikasi dan nama host server harus cocok dengan atribut `hostname` pada sertifikat.

Anda dapat menentukan apakah SSL digunakan dan sertifikat server diverifikasi dalam koneksi antara klien dan server. Untuk melakukan ini, Anda perlu meninjau `sslmode` pengaturan untuk ODBC DSN Anda pada klien dan `require_SSL` pengaturan untuk cluster Amazon Redshift di server. Tabel berikut menjelaskan hasil enkripsi untuk berbagai kombinasi pengaturan klien dan server:

sslmode (klien)	Require_SSL (server)	Hasil
disable	false	Koneksi tidak dienkripsi.
disable	true	Koneksi tidak dapat dibuat karena server memerlukan SSL dan klien memiliki SSL dinonaktifkan untuk koneksi.
allow	true	Koneksi dienkripsi.
allow	false	Koneksi tidak dienkripsi.
prefer atau require	true	Koneksi dienkripsi.
prefer atau require	false	Koneksi dienkripsi.
verify-ca	true	Koneksi dienkripsi dan sertifikat server diverifikasi.
verify-ca	false	Koneksi dienkripsi dan sertifikat server diverifikasi.
verify-full	true	Koneksi dienkripsi dan sertifikat server dan nama host diverifikasi.
verify-full	false	Koneksi dienkripsi dan sertifikat server dan nama host diverifikasi.

Connect menggunakan sertifikat server dengan ODBC di Microsoft Windows

Jika Anda ingin terhubung ke cluster menggunakan SSL dan sertifikat server, unduh sertifikat terlebih dahulu ke komputer klien Anda atau instans Amazon EC2. Kemudian konfigurasi ODBC DSN.

1. Unduh bundel otoritas sertifikat Amazon Redshift ke komputer klien Anda di `lib` folder di direktori instalasi driver Anda, dan simpan file sebagai `root.crt` Untuk informasi unduhan, lihat [SSL](#).

2. Buka Administrator Sumber Data ODBC, dan tambahkan atau edit entri DSN sistem untuk koneksi ODBC Anda. Untuk Mode SSL, pilih `verify-full` kecuali Anda menggunakan alias DNS. Jika Anda menggunakan alias DNS, pilih `verify-ca`. Lalu, pilih Simpan.

Untuk informasi selengkapnya tentang mengonfigurasi ODBC DSN, lihat [Mengonfigurasi koneksi untuk driver ODBC versi 2.x untuk Amazon Redshift](#)

Sertifikat SSL dan server di Java

SSL menyediakan satu lapisan keamanan dengan mengenkripsi data yang bergerak antara klien dan cluster Anda. Menggunakan sertifikat server memberikan lapisan keamanan tambahan dengan memvalidasi bahwa klaster tersebut adalah klaster Amazon Redshift. Ia melakukannya dengan memeriksa sertifikat server yang secara otomatis diinstal pada semua cluster yang Anda sediakan. Untuk informasi lebih lanjut tentang menggunakan sertifikat server dengan JDBC, buka [Mengonfigurasi klien dalam](#) dokumentasi PostgreSQL.

Connect menggunakan sertifikat CA trust di Java

Important

Amazon Redshift telah mengubah cara sertifikat SSL dikelola. Anda mungkin perlu memperbarui sertifikat CA root kepercayaan Anda saat ini untuk terus terhubung ke cluster Anda menggunakan SSL. Untuk informasi selengkapnya, lihat [SSL](#).

Untuk terhubung menggunakan sertifikat CA trust

Anda dapat menggunakan `redshift-keytool.jar` file untuk mengimpor sertifikat CA di bundel Amazon Redshift Certificate Authority ke Java TrustStore atau private Anda. TrustStore

1. Jika Anda menggunakan `-Djavax.net.ssl.trustStore` opsi baris perintah Java, hapus dari baris perintah, jika memungkinkan.
2. Unduh [redshift-keytool.jar](#).
3. Lakukan salah satu tindakan berikut:
 - Untuk mengimpor bundel Amazon Redshift Certificate Authority ke Java TrustStore, jalankan perintah berikut.

```
java -jar redshift-keytool.jar -s
```

- Untuk mengimpor bundel Amazon Redshift Certificate Authority ke private Anda TrustStore, jalankan perintah berikut:

```
java -jar redshift-keytool.jar -k <your_private_trust_store> -  
p <keystore_password>
```

Transisi ke sertifikat ACM untuk koneksi SSL

Amazon Redshift mengganti sertifikat SSL di klaster Anda dengan sertifikat yang dikeluarkan [AWS Certificate Manager \(ACM\)](#). ACM adalah otoritas sertifikat publik terpercaya (CA) yang dipercaya oleh sebagian besar sistem saat ini. Anda mungkin perlu memperbarui sertifikat CA root kepercayaan Anda saat ini untuk terus terhubung ke cluster Anda menggunakan SSL.

Perubahan ini hanya memengaruhi Anda jika semua hal berikut berlaku:

- Klien atau aplikasi SQL Anda terhubung ke klaster Amazon Redshift menggunakan SSL dengan `sslMode` opsi koneksi disetel `require` `keverify-ca`, atau opsi konfigurasi `verify-full`
- Anda tidak menggunakan driver Amazon Redshift ODBC atau JDBC, atau Anda menggunakan driver Amazon Redshift sebelum ODBC versi 1.3.7.1000 atau JDBC versi 1.2.8.1005.

Jika perubahan ini memengaruhi Anda di Wilayah Amazon Redshift komersial, maka Anda harus memperbarui sertifikat CA root kepercayaan Anda saat ini sebelum 23 Oktober 2017. Amazon Redshift akan mentransisikan cluster Anda untuk menggunakan sertifikat ACM antara sekarang dan 23 Oktober 2017. Perubahan seharusnya sangat sedikit atau tidak berpengaruh pada kinerja atau ketersediaan cluster Anda.

Jika perubahan ini memengaruhi Anda di Wilayah AWS GovCloud (US) (AS), maka Anda harus memperbarui sertifikat CA root kepercayaan Anda saat ini sebelum 1 April 2020 untuk menghindari gangguan layanan. Mulai tanggal ini, klien yang terhubung ke klaster Amazon Redshift menggunakan koneksi terenkripsi SSL memerlukan otoritas sertifikat terpercaya (CA) tambahan. Klien menggunakan otoritas sertifikat terpercaya untuk mengonfirmasi identitas klaster Amazon Redshift saat mereka terhubung dengannya. Tindakan Anda diperlukan untuk memperbarui klien dan aplikasi SQL Anda untuk menggunakan bundel sertifikat yang diperbarui yang menyertakan CA terpercaya baru.

⚠ Important

Di Wilayah Tiongkok pada 5 Januari 2021, Amazon Redshift mengganti sertifikat SSL di AWS Certificate Manager klaster Anda dengan sertifikat yang dikeluarkan (ACM). Jika perubahan ini memengaruhi Anda di Wilayah Tiongkok (Beijing) atau Wilayah Tiongkok (Ningxia), maka Anda harus memperbarui sertifikat CA root trust Anda saat ini sebelum 5 Januari 2021 untuk menghindari gangguan layanan. Mulai tanggal ini, klien yang terhubung ke klaster Amazon Redshift menggunakan koneksi terenkripsi SSL memerlukan otoritas sertifikat tepercaya (CA) tambahan. Klien menggunakan otoritas sertifikat tepercaya untuk mengonfirmasi identitas klaster Amazon Redshift saat mereka terhubung dengannya. Tindakan Anda diperlukan untuk memperbarui klien dan aplikasi SQL Anda untuk menggunakan bundel sertifikat yang diperbarui yang menyertakan CA tepercaya baru.

- [Menggunakan driver Amazon Redshift ODBC atau JDBC terbaru](#)
- [Menggunakan driver Amazon Redshift ODBC atau JDBC sebelumnya](#)
- [Menggunakan jenis koneksi SSL lainnya](#)

Menggunakan driver Amazon Redshift ODBC atau JDBC terbaru

Metode yang disukai adalah menggunakan driver Amazon Redshift ODBC atau JDBC terbaru. Driver Amazon Redshift dimulai dengan ODBC versi 1.3.7.1000 dan JDBC versi 1.2.8.1005 secara otomatis mengelola transisi dari sertifikat yang ditandatangani sendiri Amazon Redshift ke sertifikat ACM. Untuk mengunduh driver terbaru, lihat [Mengonfigurasi koneksi untuk driver JDBC versi 2.x untuk Amazon Redshift](#).

Jika Anda menggunakan driver Amazon Redshift JDBC terbaru, sebaiknya jangan gunakan `-Djavax.net.ssl.trustStore` dalam opsi JVM. Jika Anda harus menggunakan `-Djavax.net.ssl.trustStore`, impor bundel otoritas sertifikat Redshift ke truststore yang ditunjuknya. Untuk informasi unduhan, lihat [SSL](#). Untuk informasi selengkapnya, lihat [Mengimpor bundel otoritas sertifikat Amazon Redshift ke dalam TrustStore](#).

Menggunakan driver Amazon Redshift ODBC atau JDBC sebelumnya

- Jika ODBC DSN Anda dikonfigurasi `SSLCertPath`, timpa file sertifikat di jalur yang ditentukan.
- Jika tidak `SSLCertPath` disetel, timpa file sertifikat yang disebutkan `root.crt` di lokasi DLL driver.

Jika Anda harus menggunakan driver Amazon Redshift JDBC sebelum versi 1.2.8.1005, lakukan salah satu hal berikut:

- Jika string koneksi JDBC Anda menggunakan `sslCert` opsi, hapus opsi. `sslCert` Kemudian impor bundel otoritas sertifikat Redshift ke Java Anda. TrustStore Untuk informasi unduhan, lihat [SSL](#). Untuk informasi selengkapnya, lihat [Mengimpor bundel otoritas sertifikat Amazon Redshift ke dalam TrustStore](#).
- Jika Anda menggunakan `-Djavax.net.ssl.trustStore` opsi baris perintah Java, hapus dari baris perintah, jika memungkinkan. Kemudian impor bundel otoritas sertifikat Redshift ke Java Anda. TrustStore Untuk informasi unduhan, lihat [SSL](#). Untuk informasi selengkapnya, lihat [Mengimpor bundel otoritas sertifikat Amazon Redshift ke dalam TrustStore](#).

Mengimpor bundel otoritas sertifikat Amazon Redshift ke dalam TrustStore

Anda dapat menggunakan `redshift-keytool.jar` untuk mengimpor sertifikat CA di bundel Amazon Redshift Certificate Authority ke Java TrustStore atau truststore pribadi Anda.

Untuk mengimpor bundel otoritas sertifikat Amazon Redshift ke dalam TrustStore

1. Unduh [redshift-keytool.jar](#).
2. Lakukan salah satu tindakan berikut:
 - Untuk mengimpor bundel Amazon Redshift Certificate Authority ke Java TrustStore, jalankan perintah berikut.

```
java -jar redshift-keytool.jar -s
```

- Untuk mengimpor bundel Amazon Redshift Certificate Authority ke private Anda TrustStore, jalankan perintah berikut:

```
java -jar redshift-keytool.jar -k <your_private_trust_store> -  
p <keystore_password>
```

Menggunakan jenis koneksi SSL lainnya

Ikuti langkah-langkah di bagian ini jika Anda terhubung menggunakan salah satu dari berikut ini:

- Driver ODBC sumber terbuka

- Driver JDBC sumber terbuka
- Antarmuka [baris perintah Amazon Redshift RSQL](#)
- Binding bahasa apa pun berdasarkan libpq, seperti psycopg2 (Python) dan ruby-pg (Ruby)

Untuk menggunakan sertifikat ACM dengan jenis koneksi SSL lainnya:

1. Unduh bundel otoritas sertifikat Amazon Redshift. Untuk informasi unduhan, lihat [SSL](#).
2. Tempatkan sertifikat dari bundel di `root.crt` file Anda.
 - Pada sistem operasi Linux dan macOS X, file tersebut adalah. `~/.postgresql/root.crt`
 - Di Microsoft Windows, filenya adalah `%APPDATA%\postgresql\root.crt`.

Menghubungkan dari alat dan kode klien

Amazon Redshift menyediakan editor kueri Amazon Redshift v2 untuk terhubung ke cluster dan grup kerja Anda. Untuk informasi selengkapnya, lihat [Menanyakan database menggunakan editor kueri v2](#).

Bagian ini menyediakan beberapa opsi untuk alat pihak ketiga untuk terhubung. Selain itu, ini menjelaskan cara terhubung ke cluster Anda secara terprogram.

Topik

- [Menghubungkan dengan Amazon Redshift RSQL](#)
- [Connect ke cluster dengan Amazon Redshift RSQL](#)
- [Perintah meta Amazon Redshift RSQL](#)
- [Variabel Amazon Redshift RSQL](#)
- [Kode kesalahan Amazon Redshift RSQL](#)
- [Variabel lingkungan Amazon Redshift RSQL](#)

Menghubungkan dengan Amazon Redshift RSQL

Amazon Redshift RSQL adalah klien baris perintah untuk berinteraksi dengan cluster dan database Amazon Redshift. Anda dapat terhubung ke klaster Amazon Redshift, menjelaskan objek database, data kueri, dan melihat hasil kueri dalam berbagai format keluaran.

Amazon Redshift RSQL mendukung kemampuan alat baris perintah PostgreSQL psql dengan serangkaian kemampuan tambahan khusus untuk Amazon Redshift. Sumber daya yang dimaksud meliputi:

- Anda dapat menggunakan otentikasi masuk tunggal menggunakan AD FS,, Okta PingIdentity, Azure ADm atau penyedia identitas berbasis lainnya. SAML/JWT Anda juga dapat menggunakan penyedia identitas SAMP berbasis browser untuk otentikasi multi-faktor (MFA).
- Anda dapat mendeskripsikan properti atau atribut objek Amazon Redshift seperti kunci distribusi tabel, kunci pengurutan tabel, tampilan pengikatan akhir (LBVs), dan tampilan terwujud. Anda juga dapat menjelaskan properti atau atribut tabel eksternal dalam AWS Glue katalog atau Apache Hive Metastore, database eksternal di Amazon RDS untuk PostgreSQL, Amazon Aurora PostgreSQL Edisi yang kompatibel dengan PostgreSQL, RDS untuk MySQL (pratinjau) dan Amazon Aurora Edisi yang kompatibel dengan MySQL (pratinjau), dan tabel yang dibagikan menggunakan Amazon Redshift Berbagi data.
- Anda juga dapat menggunakan perintah aliran kontrol yang disempurnakan seperti IF (\ELSEIF, \ELSE, \ENDIF), \GOTO dan \LABEL.

Dengan mode batch Amazon Redshift RSQL, yang menjalankan skrip yang diteruskan sebagai parameter input, Anda dapat menjalankan skrip yang menyertakan SQL dan logika bisnis yang kompleks. Jika Anda memiliki gudang data lokal yang dikelola sendiri, Anda dapat menggunakan Amazon Redshift RSQL untuk mengganti skrip ekstrak, transformasi, pemuatan (ETL), dan otomatisasi yang ada, seperti skrip Teradata BTEQ. Menggunakan RSQL membantu Anda menghindari penerapan ulang skrip secara manual dalam bahasa prosedural.

Amazon Redshift RSQL tersedia untuk sistem operasi Linux, Windows, dan macOS X.

<Untuk melaporkan masalah Amazon Redshift RSQL, tulis ke [@amazon .com. redshift-](mailto:@amazon.com.redshift-)

Topik

- [Memulai dengan Amazon Redshift RSQL](#)
- [Log perubahan Amazon Redshift RSQL](#)

Memulai dengan Amazon Redshift RSQL

Instal Amazon Redshift RSQL di komputer dengan sistem operasi Linux, macOS, atau Microsoft Windows.

Unduh RSQL

-
-
-
-
-

Lihat log perubahan dan unduhan untuk versi sebelumnya di [Log perubahan Amazon Redshift RSQL](#).

Instal RSQL untuk Linux

Ikuti langkah-langkah di bawah ini untuk menginstal RSQL untuk Linux.

1. Instal manajer driver dengan perintah berikut:

```
sudo yum install unixODBC
```

2. Instal driver ODBC: [Mengunduh dan menginstal driver Amazon Redshift ODBC](#).
3. Salin file ini ke direktori home Anda:

```
cp /opt/amazon/redshiftodbcx64/odbc.ini ~/.odbc.ini
```

4. Atur variabel lingkungan untuk menunjuk ke lokasi file:

```
export ODBCINI=~/.odbc.ini
export ODBCSYSINI=/opt/amazon/redshiftodbcx64/
export AMAZONREDSHIFTODBCINI=/opt/amazon/redshiftodbcx64/amazon.redshiftodbc.ini
```

5. Anda sekarang dapat menginstal RSQL dengan menjalankan perintah berikut.

```
sudo rpm -i AmazonRedshiftRsql-<version>.rhel.x86_64.rpm
```

Instal RSQL untuk Mac

Ikuti langkah-langkah di bawah ini untuk menginstal RSQL for Mac OSX.

1. Instal manajer driver dengan perintah berikut:

```
brew install unixodbc --build-from-source
```

2. Instal driver ODBC: [Mengunduh dan menginstal driver Amazon Redshift ODBC](#).
3. Salin file ini ke direktori home Anda:

```
cp /opt/amazon/redshift/Setup/odbc.ini ~/.odbc.ini
```

4. Atur variabel lingkungan untuk menunjuk ke lokasi file:

```
export ODBCINI=~/.odbc.ini
export ODBCSYSINI=/opt/amazon/redshift/Setup
export AMAZONREDSHIFTODBCINI=/opt/amazon/redshift/lib/amazon.redshiftodbc.ini
```

5. Setel DYLD_LIBRARY_PATH ke lokasi libodbc.dylib Anda jika tidak masuk. /usr/local/lib

```
export DYLD_LIBRARY_PATH=$DYLD_LIBRARY_PATH:/usr/local/lib
```

6. Klik dua kali file pkg untuk menjalankan installer.
7. Ikuti langkah-langkah di installer untuk menyelesaikan instalasi. Setuju dengan ketentuan perjanjian lisensi.

Instal RSQL untuk Windows

Ikuti langkah-langkah di bawah ini untuk menginstal RSQL untuk Windows.

1. Instal driver ODBC: [Mengunduh dan menginstal driver Amazon Redshift ODBC](#).
2. Klik dua kali file unduhan RSQL untuk menjalankan penginstal, lalu ikuti petunjuk untuk menyelesaikan instalasi.

Log perubahan Amazon Redshift RSQL

1.1.2 (2025-12-11)

Perbaiki Bug

- Memperbaiki bug yang menyebabkan kegagalan dengan perintah\ goto dan\ label.
- Memperbaiki bug yang mencegah RSQL mencetak nilai variabel saat variabel diapit tanda kutip.

- Memperbaiki bug yang menyebabkan RSQL crash saat hasil kueri melebihi ukuran parameter ODBC DSN Fetch saat diaktifkan. UseDeclareFetch
- Memperbaiki masalah di mana beberapa halaman hasil dikembalikan sekaligus meskipun pager diaktifkan.
- Memperbaiki bug yang memicu crash RSQL saat kueri gagal dalam blok transaksi.

1.1.1 (2025-11-20)

Perbaikan Bug

- Memperbaiki masalah di mana RSQL salah mengurai kueri saat menggunakan flag -c. Perbaikan ini berlaku untuk semua platform.
- Perbaiki bug di Mac yang mencegah pengguna menggunakan perintah\ s di RSQL.

1.1.0 (2025-11-11)

Perbaikan Bug

- Menyelesaikan masalah kebocoran memori yang menyebabkan crash tak terduga di RSQL.
- Dihapus ketergantungan OpenSSL dari RSQL.
- Konflik penautan tetap dengan libpq/psql instalasi di lingkungan yang sama.
- Kompatibilitas platform yang ditingkatkan untuk Amazon Linux 2023, Windows, dan macOS.
- Memperbaiki masalah di mana output akan terpotong saat melebihi ukuran tampilan saat ini.

1.0.8 (2023-06-19)

Perbaikan Bug

- Memperbaiki masalah saat output terpotong dengan perintah SHOW.
- Menambahkan dukungan ke\ de untuk menjelaskan aliran Kinesis eksternal dan topik Kafka.

1.0.7 (2023-03-22)

Perbaikan Bug

- Memperbaiki masalah di mana RSQL tidak dapat menggambarkan tampilan terwujud.

- Memperbaiki kesalahan penolakan izin pada `stl_connection_log` saat menggunakan Amazon Redshift Serverless.
- Memperbaiki masalah di mana RSQL dapat memproses label\ GOTO secara tidak benar.
- Memperbaiki masalah saat pesan SSL dicetak dalam mode senyap.
- Memperbaiki masalah dengan karakter acak yang ditampilkan saat menjelaskan prosedur yang disimpan.
- Memperbaiki masalah dengan mencetak ERROR/INFO pesan duplikat.

Baru

- RSQL sekarang mendapatkan informasi SSL langsung dari driver ODBC.

1.0.6 (2023-02-21)

Perbaikan Bug

- Memperbaiki masalah di mana\ d melempar kesalahan - sintaks input tidak valid untuk integer: "xid" - pada patch Redshift 1.0.46086 (P173).

Baru

- Berganti nama file instalasi untuk mencerminkan arsitektur yang didukung.

1.0.5 (2022-06-27)

Perbaikan Bug

- Kirim pesan kesalahan SQL ke kesalahan standar (stderr).
- Memperbaiki masalah dengan kode keluar saat menggunakan `ON_ERROR_STOP`. Skrip sekarang berakhir setelah mengalami kesalahan dan mengembalikan kode keluar yang benar.
- `Maxerror` sekarang tidak peka huruf besar/kecil.

Baru

- Menambahkan dukungan untuk driver ODBC 2.x.

1.0.4 (2022-03-19)

- Tambahkan dukungan untuk variabel lingkungan RSPASSWORD. Tetapkan kata sandi untuk terhubung ke Amazon Redshift. Misalnya, `export RSPASSWORD=TestPassw0rd`.

1.0.3 (2021-12-08)

Perbaikan Bug

- Dialog tetap muncul saat menggunakan `\c` atau `\l` untuk beralih antar database di OS Windows.
- Memperbaiki kerusakan saat memeriksa informasi ssl.

Amazon Redshift RSQL versi sebelumnya

Pilih salah satu tautan untuk mengunduh versi Amazon Redshift RSQL yang Anda butuhkan, berdasarkan sistem operasi Anda.

Linux 64-bit RPM

- [RSQL Versi 1.1.1](#)
- [RSQL Versi 1.1.0](#)
- [RSQL Versi](#)
- [RSQL Versi](#)
- [RSQL Versi](#)
- [RSQL Versi 1.0.5](#)
- [RSQL Versi](#)
- [RSQL Versi 1.0.3](#) [64-bit RSQL Versi 1.0.3](#)
- [RSQL Versi 1.0.1](#) [64-bit RSQL Versi 1.0.1](#)

Mac OS 64-bit DMG/PKG

- [RSQL Versi 1.1.1](#)
- [RSQL Versi 1.1.0](#)

- [RSQL Versi](#)
- [RSQL Versi](#)
- [RSQL Versi](#)
- [RSQL Versi 1.0.5](#) [64-bit RSQL Versi 1.0.5](#)
- [RSQL Versi 1.0.4](#) [Di : 64-bit RSQL Versi 1.0.4](#)
- [RSQL Versi 1.0.3](#) [64-bit RSQL Versi 1.0.3](#)
- [RSQL Versi 1.0.1](#) [64-bit RSQL Versi 1.0.1](#)

Windows 64-bit MSI

- [RSQL Versi 1.1.1](#)
- [RSQL Versi 1.1.0](#)
- [RSQL Versi](#)
- [RSQL Versi](#)
- [RSQL Versi](#)
- [RSQL Versi 1.0.5](#)
- [RSQL Versi](#)
- [RSQL Versi 1.0.3](#) [64-bit RSQL Versi 1.0.3](#)
- [RSQL Versi 1.0.1](#) [64-bit RSQL Versi 1.0.1](#)

Connect ke cluster dengan Amazon Redshift RSQL

Dengan Amazon Redshift, Anda dapat terhubung ke cluster dan berinteraksi dengannya menggunakan RSQL. Ini adalah alat baris perintah yang menyediakan cara aman untuk menanyakan data, membuat objek database, dan mengelola cluster Amazon Redshift Anda. Bagian berikut memandu Anda melalui langkah-langkah untuk membuat koneksi ke cluster Anda menggunakan RSQL dengan dan tanpa nama sumber data (DSN).

Menghubungkan tanpa DSN

1. Di konsol Amazon Redshift, pilih cluster yang ingin Anda sambungkan dan catat titik akhir, database, dan port.
2. Pada prompt perintah, tentukan informasi koneksi dengan menggunakan parameter baris perintah.

```
rsql -h <endpoint> -U <username> -d <databasename> -p <port>
```

Di sini, berikut ini berlaku:

- **<endpoint>** adalah Endpoint yang Anda rekam pada langkah sebelumnya.
- **<username>** adalah nama pengguna dengan izin untuk terhubung ke cluster.
- **<databasename>** adalah Nama Database yang Anda rekam pada langkah sebelumnya.
- **<port>** adalah Port yang Anda rekam pada langkah sebelumnya. **<port>** adalah parameter opsional.

Berikut contohnya.

```
rsql -h testcluster.example.amazonaws.com -U user1 -d dev -p 5439
```

3. Pada prompt kata sandi, masukkan kata sandi untuk **<username>** pengguna.

Respons koneksi yang berhasil terlihat seperti berikut ini.

```
% 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=#
```

Perintah untuk menghubungkan memiliki parameter yang sama di Linux, Mac OS, dan Windows.

Menghubungkan dengan DSN

Anda dapat menghubungkan RSQL ke Amazon Redshift dengan menggunakan DSN untuk menyederhanakan organisasi properti koneksi. Topik ini mencakup instruksi untuk instalasi ODBC-driver dan deskripsi untuk properti DSN.

Menggunakan koneksi DSN dengan kata sandi

Berikut ini menunjukkan contoh konfigurasi koneksi DSN yang menggunakan kata sandi.

Default <path to driver> untuk Mac OSX adalah /opt/amazon/redshift/lib/libamazonredshiftodbc.dylib dan untuk Linux adalah /opt/amazon/redshiftodbc/lib/64/libamazonredshiftodbc64.so.

```
[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
```

Hasil output berikut dari koneksi yang berhasil.

```
% 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=#
```

Menggunakan DSN masuk tunggal

Anda dapat mengonfigurasi DSN untuk otentikasi masuk tunggal. Berikut ini menunjukkan contoh konfigurasi koneksi DSN yang menggunakan tanda tunggal Okta.

```
[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>
```

Contoh output dari koneksi yang berhasil.

```
% 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=#
```

Contoh berikut menunjukkan contoh konfigurasi koneksi DSN yang menggunakan tanda tunggal Azure.

```
[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>
```

Menggunakan koneksi DSN dengan profil IAM

Anda dapat terhubung ke Amazon Redshift menggunakan profil IAM yang telah dikonfigurasi. Profil IAM harus memiliki hak istimewa untuk menelepon. `GetClusterCredentials` Contoh berikut menunjukkan properti DSN untuk digunakan. `RegionParameter ClusterID` dan diperlukan hanya jika Host bukan titik akhir yang disediakan Amazon seperti `examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com`.

```
[testiam]
Driver=Default
Host=testcluster.example.com
Database=dev
DbUser=testuser
ClusterID=rsqltestcluster
Region=us-east-1
IAM=1
Profile=default
```

Nilai untuk `Profile` kunci adalah profil bernama yang Anda pilih dari kredensial AWS CLI Anda. Contoh ini menunjukkan kredensial untuk profil bernama `default`

```
$ cat .aws/credentials
[default]
aws_access_key_id = ASIAIOSFODNN7EXAMPLE
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxrFcYEXAMPLEKEY
```

Berikut ini menunjukkan respon koneksi.

```
$ 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=>
```

Menggunakan koneksi DSN dengan profil Instance

Anda dapat terhubung ke Amazon Redshift menggunakan profil instans Amazon EC2. Profil instance harus memiliki hak istimewa untuk menelepon `GetClusterCredentials`. Lihat contoh di bawah ini untuk properti DSN yang akan digunakan. `RegionParameter ClusterID` dan diperlukan hanya jika Host bukan titik akhir yang disediakan Amazon seperti `examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com`.

```
[testinstanceprofile]
Driver=Default
Host=testcluster.example.com
Database=dev
DbUser=testuser
ClusterID=rsqltestcluster
Region=us-east-1
IAM=1
Instanceprofile=1
```

Berikut ini menunjukkan respon koneksi.

```
$ 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=>
```

Menggunakan koneksi DSN dengan rantai penyedia kredensi default

Untuk menyambung menggunakan rantai penyedia kredensi default, tentukan hanya properti IAM, dan Amazon Redshift RSQL akan mencoba memperoleh kredensial dalam urutan yang dijelaskan dalam Bekerja dengan Kredensial di SDK for [Java](#). AWS AWS Setidaknya salah satu penyedia dalam

rantai harus memiliki `GetClusterCredentials` izin. Ini berguna untuk menghubungkan dari wadah ECS, misalnya.

```
[iamcredentials]
Driver=Default
Host=testcluster.example.com
Database=dev
DbUser=testuser
ClusterID=rsqltestcluster
Region=us-east-1
IAM=1
```

Perintah meta Amazon Redshift RSQL

Perintah meta Amazon Redshift RSQL mengembalikan catatan informasi tentang database atau objek database tertentu. Hasil dapat mencakup berbagai kolom dan metadata. Perintah lain melakukan tindakan tertentu. Perintah ini didahului dengan garis miring terbalik.

`\d [S +]`

Daftar tabel buatan pengguna lokal, tampilan reguler, tampilan yang mengikat akhir, dan tampilan terwujud. `\dS` juga mencantumkan tabel dan tampilan `\d`, seperti, tetapi objek sistem disertakan dalam catatan yang dikembalikan. +Hasil di kolom metadata tambahan `description` untuk semua objek yang terdaftar. Berikut ini menunjukkan catatan sampel dikembalikan sebagai hasil dari perintah.

```
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+] NAMA`

Menjelaskan tabel, tampilan, atau indeks. Termasuk nama kolom dan jenis. Ini juga menyediakan `diststyle`, konfigurasi cadangan, membuat tanggal (tabel dibuat setelah Oktober 2018), dan kendala.

Misalnya, `\dS+ sample` mengembalikan properti objek. Menambahkan `S+` hasil di kolom tambahan termasuk dalam catatan yang dikembalikan.

```
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)
```

Gaya distribusi, atau `Diststyle`, dari tabel bisa `KEY`, `AUTO`, `EVEN` atau `ALL`.

Backup menunjukkan apakah tabel dicadangkan saat snapshot diambil. Nilai-nilai yang valid adalah `YES` atau `NO`.

Dibuat adalah stempel waktu untuk saat tabel dibuat. Tanggal pembuatan tidak tersedia untuk tabel Amazon Redshift yang dibuat sebelum November 2018. Tabel yang dibuat sebelum tanggal ini ditampilkan `n/a` (Tidak Tersedia).

Kendala Unik mencantumkan kendala kunci unik dan primer di atas meja.

Kendala kunci asing mencantumkan kendala kunci asing di atas meja.

\dC [+] [POLA]

Daftar gips. Termasuk jenis sumber, tipe target, dan apakah pemeran itu implisit.

Berikut ini menunjukkan subset hasil dari\dC+.

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				
bigint		character varying	text	in
assignment				
bigint		double precision	float8	yes

bigint	integer	int4	in
assignment			
bigint	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			
bigint	super	int8_partiql	in
assignment			

\dd [S] [POLA]

Menampilkan deskripsi objek yang tidak ditampilkan di tempat lain.

\de

Daftar tabel eksternal. Ini termasuk tabel di AWS Glue Data Catalog, Hive Metastore dan tabel federasi dari Amazon MySQL, Amazon PostgreSQL, dan tabel data RDS/Aurora Amazon Redshift. RDS/Aurora

\NAMA

Menjelaskan tabel eksternal.

Contoh berikut menunjukkan tabel AWS Glue eksternal.

```
# \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_linenum	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"}

Meja Hive Metastore.

\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_linenum	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 tabel eksternal.

```
# \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+] [POLA]`

Daftar fungsi dari berbagai jenis. Perintah `\ df`, misalnya, mengembalikan daftar fungsi. Hasil mencakup properti seperti nama, tipe data yang dikembalikan, hak akses, dan metadata tambahan. Jenis fungsi dapat mencakup pemicu, prosedur tersimpan, fungsi jendela, dan jenis lainnya. Ketika Anda menambahkan `S+` ke perintah, misalnya `\ dfantS+`, kolom metadata tambahan disertakan, seperti `owner`, dan `security access privileges`

`\ dL [S+] [POLA]`

Daftar data tentang bahasa prosedural yang terkait dengan database. Informasi mencakup nama, seperti `plpgsql`, dan metadata tambahan, yang mencakup apakah itu tepercaya, hak akses, dan deskripsi. Contoh panggilan, misalnya `\ dLS+`, yang mencantumkan bahasa dan propertinya. Ketika Anda menambahkan `S+` ke perintah, kolom metadata tambahan disertakan, seperti `call handler access privileges`

Hasil sampel:

List of languages

name	trusted	internal language	call handler	access privileges	description
validator					
-----+-----+-----+-----+-----+-----					
+-----+-----+-----+-----+-----+-----					
+-----+-----+-----+-----+-----+-----					
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+] [POLA]

Daftar tampilan terwujud. Misalnya, \ dmS+ daftar tampilan terwujud dan propertinya. Saat Anda menambahkan S+ ke perintah, kolom metadata tambahan disertakan.

\ dn [S+] [POLA]

Daftar skema. Ketika Anda menambahkan S+ ke perintah, misalnya \ dnS+, kolom metadata tambahan disertakan, seperti dan. description access privileges

\ dp [POLA]

Daftar hak akses tabel, tampilan, dan urutan.

`\ dt [S+] [POLA]`

Daftar tabel. Ketika Anda menambahkan S+ ke perintah, misalnya `\ dt S+`, kolom metadata tambahan disertakan, seperti `description` dalam kasus ini.

`\ du`

Daftar pengguna untuk database. Termasuk nama dan peran mereka, seperti Superuser, dan atribut.

`\ dv [S+] [POLA]`

Daftar tampilan. Termasuk skema, jenis, dan data pemilik. Ketika Anda menambahkan S+ ke perintah, misalnya `\ dv S+`, kolom metadata tambahan disertakan.

`\ H`

Menghidupkan output HTML. Ini berguna untuk mengembalikan hasil yang diformat dengan cepat. Misalnya, `select * from sales; \H` mengembalikan hasil dari tabel penjualan, dalam HTML. Untuk beralih kembali ke hasil tablular, gunakan `\q`, atau diam.

`\ saya`

Menjalankan perintah dari file. Misalnya, dengan asumsi Anda memiliki `rsql_steps.sql` di direktori kerja Anda, berikut ini menjalankan perintah dalam file: `\i rsql_steps.sql`.

`\ I [+] [POLA]`

Daftar database. Termasuk pemilik, pengkodean, dan informasi tambahan.

`\ q`

Keluar, atau `\q` perintah, log off sesi database dan keluar dari RSQL.

`\ sv [+] NAMA TAMPILAN`

Menunjukkan definisi tampilan.

`\ waktu`

Menunjukkan waktu berjalan, untuk kueri, misalnya.

`\ z [POLA]`

Output yang sama dengan `\ dp`.

\?

Menampilkan informasi bantuan. Parameter opsional menentukan item untuk menjelaskan.

\ KELUAR

Log off semua sesi database dan keluar dari Amazon Redshift RSQL. Selain itu, Anda dapat menentukan kode keluar opsional. Misalnya, \EXIT 15 akan keluar dari terminal Amazon Redshift RSQL dan mengembalikan kode keluar 15.

Contoh berikut menunjukkan output dari koneksi dan keluar dari RSQL.

```
% rsq1 -D testuser
DSN Connected
DBMS Name: Amazon Redshift
Driver Name: Amazon Redshift ODBC Driver
Driver Version: 1.4.34.1000
Rsqli Version: 1.0.1
Redshift Version: 1.0.29306
Type "help" for help.

(testcluster) user1@dev=# \exit 15

% echo $?
15
```

\ EKSPOR

Menentukan nama file ekspor yang RSQL gunakan untuk menyimpan informasi database yang dikembalikan oleh pernyataan SQL SELECT berikutnya.

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
```

Keluaran konsol

```
Rformat is off.  
Target width is 1500.  
Heading is set to: General Title  
Titledashes is on.  
(exported 40 rows)
```

\MASUK

Terhubung ke database. Anda dapat menentukan parameter koneksi menggunakan sintaks posisi atau sebagai string koneksi.

Sintaks perintah adalah sebagai berikut: `\logon {[DBNAME]- USERNAME|- HOST|- PORT|- [PASSWORD]] | conninfo}`

DBNAMEIni adalah nama database untuk terhubung ke. USERNAMEItu adalah nama pengguna untuk terhubung sebagai. HOSTDefaultnya adalahlocalhost. PORTDefaultnya adalah5439.

Ketika nama host ditentukan dalam \LOGON perintah, itu menjadi nama host default untuk \LOGON perintah tambahan. Untuk mengubah nama host default, tentukan yang baru HOST dalam \LOGON perintah tambahan.

Contoh output dari \LOGON perintah untuk user1 berikut.

```
(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=#
```

Output sampel untuk pengguna2.

```
(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=#
```

\ KOMENTAR

Perpanjangan `\echo` perintah. `\REMARK` mencetak string yang ditentukan ke aliran output. `\REMARK` meluas `\echo` dengan menambahkan kemampuan untuk memecah output melalui jalur terpisah.

Contoh berikut menunjukkan output dari perintah.

```
(testcluster) user1@dev=# \remark 'hello//world'
hello
world
```

\ RSET

Perintah `\rset` menetapkan parameter perintah dan variabel. `\rset` memiliki mode interaktif dan batch. `<arg>` Itu tidak mendukung opsi sebagai opsi bash, seperti `-x`, atau argumen, misalnya `--`.

Ini menetapkan variabel, seperti berikut ini:

- TINGKAT KESALAHAN
- HEADING dan RTITLE
- RFORMAT
- MAXERROR
- TITLEDASHES
- LEBAR

Contoh berikut menentukan judul.

```
\rset heading "Winter Sales Report"
```

Untuk lebih banyak contoh cara menggunakan `\rset`, Anda dapat menemukan beberapa [Variabel Amazon Redshift RSQL](#) topik.

\ LARI

Menjalankan skrip Amazon Redshift RSQL yang terdapat dalam file yang ditentukan.

`\RUN` memperluas `\i` perintah dengan menambahkan opsi untuk melewati baris header dalam file.

Jika nama file menyertakan koma, titik koma, atau spasi, lampirkan dalam tanda kutip tunggal. Selain itu, jika teks mengikuti nama file, lampirkan dalam tanda kutip. Di UNIX, nama file peka huruf besar/kecil. Di Windows, nama file tidak peka huruf besar/kecil.

Contoh berikut menunjukkan output dari perintah.

```
(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

Sebuah alias untuk \! perintah. \OS menjalankan perintah sistem operasi yang diteruskan sebagai parameter. Kontrol kembali ke Amazon Redshift RSQL setelah perintah dijalankan. Misalnya, Anda dapat menjalankan perintah berikut untuk mencetak waktu tanggal sistem saat ini dan kembali ke terminal RSQL:.

```
\os date
```

```
(testcluster) user1@dev=# \os date
Tue Sep 7 20:47:54 UTC 2021
```

\GOTO

Perintah baru untuk Amazon Redshift RSQL. \GOTO melewati semua perintah intervensi dan melanjutkan pemrosesan pada yang ditentukan. \LABEL \LABEL Harus menjadi referensi ke depan. Anda tidak dapat melompat ke \LABEL yang secara leksikal mendahului. \GOTO

Berikut menampilkan contoh output.

```
(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

Perintah baru untuk Amazon Redshift RSQL. \LABEL menetapkan titik masuk untuk menjalankan program, sebagai target untuk \GOTO perintah.

Berikut ini menunjukkan contoh output dari perintah.

```
(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` dan perintah terkait menjalankan bagian dari skrip input secara kondisional. Perpanjangan dari perintah PSQL `\ if (\ elif, \ else, \ endif)`. `\ IF` dan `\ ELSEIF` mendukung ekspresi boolean termasuk `AND`, `OR` dan `NOT` kondisi.

Berikut ini menunjukkan contoh output dari perintah.

```

(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

```

Gunakan `ERRORCODE` dalam logika percabangan Anda.

```
\if :'ERRORCODE' = '00000'
    \remark 'The statement was executed without error'
\else
    \remark :LAST_ERROR_MESSAGE
\endif
```

Gunakan `\GOTO` dalam `\IF` blok untuk mengontrol bagaimana kode dijalankan.

Variabel Amazon Redshift RSQL

Beberapa kata kunci bertindak sebagai variabel dalam RSQL. Anda dapat mengatur masing-masing ke nilai tertentu, atau mengatur ulang nilainya. Sebagian besar diatur dengan `\rset`, yang memiliki mode interaktif dan mode batch. Perintah dapat didefinisikan dalam huruf kecil atau huruf besar.

JUMLAH AKTIVITAS

Menunjukkan jumlah baris yang dipengaruhi oleh permintaan terakhir yang dikirimkan. Untuk permintaan pengembalian data, ini adalah jumlah baris yang dikembalikan ke RSQL dari database. Nilainya adalah 0 atau bilangan bulat positif. Nilai maksimum adalah 18.446.744.073.709.551.615.

Variabel `ACTIVITYCOUNT` yang diperlakukan secara khusus mirip dengan variabel `ROW_COUNT`. Namun, `ROW_COUNT` tidak melaporkan jumlah baris yang terpengaruh ke aplikasi klien pada penyelesaian perintah untuk `SELECT`, `COPY` atau `UNLOAD`. Tapi `ACTIVITYCOUNT` tidak.

activitycount_01.sql:

```
select viewname, schemaname
from pg_views
where schemaname = 'not_existing_schema';
\if :ACTIVITYCOUNT = 0
\remark 'views do not exist'
\endif
```

Keluaran konsol:

```
viewname | schemaname
-----+-----
(0 rows)
```

```
views do not exist
```

TINGKAT KESALAHAN

Menetapkan tingkat keparahan untuk kesalahan. Gunakan tingkat keparahan untuk menentukan tindakan. Jika `ERRORLEVEL` perintah belum digunakan, nilainya secara default ON.

`errorlevel_01.sql`:

```
\rset errorlevel 42P01 severity 0

select * from tbl;

select 1 as col;

\echo exit
\quit
```

Keluaran konsol:

```
Errorlevel is on.
rsql: ERROR: relation "tbl" does not exist
(1 row)

col
1

exit
```

HEADING dan RTITLE

Memungkinkan pengguna untuk menentukan header yang muncul di bagian atas laporan. Header yang ditentukan oleh `RSET RTITLE` perintah secara otomatis menyertakan tanggal sistem saat ini dari komputer klien.

konten `rset_heading_rtitle_02.rsq`:

```
\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...
```

Keluaran konsol:

```
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

Menentukan tingkat keparahan kesalahan maksimum di luar mana RSQL menghentikan pemrosesan pekerjaan. Kode pengembalian adalah nilai integer yang RSQL kembali ke sistem operasi klien setelah menyelesaikan setiap pekerjaan atau tugas. Nilai kode pengembalian menunjukkan status penyelesaian pekerjaan atau tugas. Jika skrip berisi pernyataan yang menghasilkan tingkat keparahan kesalahan lebih besar dari `maxerror` nilai yang ditentukan, RSQL segera keluar. Oleh karena itu, untuk memiliki RSQL keluar pada tingkat keparahan kesalahan 8, gunakan. `RSET MAXERROR 7`

konten `maxerror_01.sql`:

```
\rset maxerror 0

select 1 as col;

\quit
```

Keluaran konsol:

```
Maxerror is default.
(1 row)

col
1
```

RFORMAT

Memungkinkan pengguna untuk menentukan apakah akan menerapkan pengaturan untuk perintah pemformatan.

konten rset_rformat.rsq:

```
\remark Starting...
\pset border 2
\pset format wrapped
\pset expanded on
\pset title 'Great Title'
select * from rsq_test.tbl_long where id = 500;
\rset rformat
select * from rsq_test.tbl_long where id = 500;
\rset rformat off
select * from rsq_test.tbl_long where id = 500;
\rset rformat on
select * from rsq_test.tbl_long where id = 500;
\exit
\remark Finishing...
```

Keluaran konsol:

```
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

Mendapat jumlah catatan yang dipengaruhi oleh kueri sebelumnya. Ini biasanya digunakan untuk memeriksa hasil, seperti dalam fragmen kode berikut:

```

SET result = ROW_COUNT;

IF result = 0

```

...

TITLEDASHES

Kontrol ini memungkinkan pengguna untuk menentukan apakah garis karakter dasbor akan dicetak di atas data kolom yang dikembalikan untuk pernyataan SQL.

Contoh:

```
\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;
```

Keluaran konsol:

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

LEBAR

Menetapkan format output untuk dibungkus dan menentukan lebar target untuk setiap baris dalam laporan. Tanpa parameter, ia mengembalikan pengaturan saat ini untuk format dan lebar target.

konten rset_width_01.rsq1:

```
\echo Starting...
\rset width
\rset width 50
\rset width
\quit
\echo Finishing...
```

Keluaran konsol:

```
Starting...
Target width is 75.
Target width is 50.
Target width is 50.
Press any key to continue . . .
```

Contoh dengan parameter:

```
\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...
```

Keluaran konsol:

```
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 . . .
```

Kode kesalahan Amazon Redshift RSQL**Pesan sukses, peringatan, dan pengecualian:**

Kode Kesalahan	Kelas Kesalahan	Nama Kondisi
00000	Kelas 00 — Penyelesaian Berhasil	successful_completion
01000	Kelas 01 - Peringatan	memperingati
0100C	Kelas 01 - Peringatan	dynamic_result_sets_returned
01008	Kelas 01 - Peringatan	implicit_zero_bit_padding
01003	Kelas 01 - Peringatan	null_value_eliminated_in_set_function
01007	Kelas 01 - Peringatan	privilege_not_granted
01006	Kelas 01 - Peringatan	privilege_not_revoked
01004	Kelas 01 - Peringatan	string_data_right_truncation
01P01	Kelas 01 - Peringatan	deprecated_feature
02000	Kelas 02 - Tidak Ada Data	tidak data_data
02001	Kelas 02 - Tidak Ada Data	no_additional_dynamic_result_sets_returned
03000	Kelas 03 - Pernyataan SQL Belum Lengkap	sql_statement_not_yet_complete
08000	Kelas 08 - Pengecualian Koneksi	connection_exception
08003	Kelas 08 - Pengecualian Koneksi	connection_does_not_exist
08006	Kelas 08 - Pengecualian Koneksi	connection_failure
08001	Kelas 08 - Pengecualian Koneksi	sqlclient_unable_to_establish_sqlconnection

Kode Kesalahan	Kelas Kesalahan	Nama Kondisi
08004	Kelas 08 - Pengecualian Koneksi	sqlserver_rejected_establishment_of_sqlconnection
08007	Kelas 08 - Pengecualian Koneksi	transaction_resolution_unknown
08P01	Kelas 08 - Pengecualian Koneksi	protocol_violation
09000	Kelas 09 - Pengecualian Tindakan yang Dipicu	triggered_action_exception
0A000	Kelas 0A - Fitur Tidak Didukung	feature_not_supported
0A000	Kelas 0A - Fitur Tidak Didukung	feature_not_supported
0B000	Kelas 0B — Inisiasi Transaksi Tidak Valid	invalid_transaction_initiation
0F000	Kelas 0F - Pengecualian Pencari Lokasi	locator_exception
0F001	Kelas 0F - Pengecualian Pencari Lokasi	invalid_locator_specification
0L000	Kelas 0L - Pemberi Tidak Valid	invalid_grantor
0 LP01	Kelas 0L - Pemberi Tidak Valid	invalid_grant_operation
0P000	Kelas 0P - Spesifikasi Peran Tidak Valid	invalid_role_specification

Kode Kesalahan	Kelas Kesalahan	Nama Kondisi
0Z000	Kelas 0Z - Pengecualian Diagnostik	diagnostics_exception
0Z002	Kelas 0Z - Pengecualian Diagnostik	stacked_diagnostics_accessed_without_active_handler
20000	Kelas 20 — Kasus Tidak Ditemukan	kasus_not_found
21000	Kelas 21 — Pelanggaran Kardinalitas	kardinalitas_pelanggaran

Pengecualian data:

Kode Kesalahan	Kelas Kesalahan	Nama Kondisi
22000	Kelas 22 - Pengecualian Data	data_exception
2202E	Kelas 22 - Pengecualian Data	array_subscript_error
22021	Kelas 22 - Pengecualian Data	character_not_in_repertoar
22008	Kelas 22 - Pengecualian Data	datetime_field_overflow
22012	Kelas 22 - Pengecualian Data	division_by_zero
22005	Kelas 01 - Peringatan	error_in_assignment
2200B	Kelas 01 - Peringatan	escape_character_conflict
22022	Kelas 01 - Peringatan	indicator_overflow
22015	Kelas 01 - Peringatan	interval_field_overflow
2201E	Kelas 01 - Peringatan	invalid_argument_for_logaritma

Kode Kesalahan	Kelas Kesalahan	Nama Kondisi
2201F	Kelas 01 - Peringatan	invalid_argument_for_power_function
2201G	Kelas 01 - Peringatan	invalid_argument_for_width_bucket_function
22018	Kelas 01 - Peringatan	invalid_character_value_for_cast
22007	Kelas 01 - Peringatan	format invalid_datetime_
22019	Kelas 01 - Peringatan	invalid_escape_character
2200D	Kelas 01 - Peringatan	invalid_escape_octet
22025	Kelas 01 - Peringatan	invalid_escape_sequence
22P06	Kelas 01 - Peringatan	nonstandard_use_of_escape_character
22010	Kelas 01 - Peringatan	invalid_indicator_parameter_value
22023	Kelas 01 - Peringatan	invalid_parameter_value
2201B	Kelas 01 - Peringatan	invalid_regular_expression
22009	Kelas 01 - Peringatan	invalid_time_zone_displacement_value
2200C	Kelas 01 - Peringatan	invalid_use_of_escape_character
2200G	Kelas 01 - Peringatan	most_specific_type_mismatch
22004	Kelas 01 - Peringatan	null_value_not_allowed
22002	Kelas 01 - Peringatan	null_value_no_indicator_parameter
22003	Kelas 01 - Peringatan	numeric_value_out_of_range
22026	Kelas 01 - Peringatan	string_data_length_mismatch
22001	Kelas 01 - Peringatan	string_data_right_truncation
22011	Kelas 01 - Peringatan	substring_error

Kode Kesalahan	Kelas Kesalahan	Nama Kondisi
22027	Kelas 01 - Peringatan	trim_error
22024	Kelas 01 - Peringatan	unterminated_c_string
2200F	Kelas 01 - Peringatan	zero_length_character_string
22P01	Kelas 01 - Peringatan	floating_point_exception
22P02	Kelas 01 - Peringatan	invalid_text_representation
22P03	Kelas 01 - Peringatan	invalid_binary_representasi
22P04	Kelas 01 - Peringatan	bad_copy_file_format
22P05	Kelas 01 - Peringatan	untranslatable_character

Pelanggaran kendala integritas:

Kode Kesalahan	Kelas Kesalahan	Nama Kondisi
23000	Kelas 23 — Pelanggaran Kendala Integritas	integrity_constraint_violation
23001	Kelas 23 — Pelanggaran Kendala Integritas	restrict_violation
23502	Kelas 23 — Pelanggaran Kendala Integritas	not_null_violation
23503	Kelas 23 — Pelanggaran Kendala Integritas	foreign_key_violor
23505	Kelas 23 — Pelanggaran Kendala Integritas	unik_pelanggaran

Kode Kesalahan	Kelas Kesalahan	Nama Kondisi
23514	Kelas 23 — Pelanggaran Kendala Integritas	check_violation
24000	Kelas 24 - Status Kursor Tidak Valid	invalid_cursor_state
01004	Kelas 01 - Peringatan	string_data_right_truncation
25000	Kelas 25 - Status Transaksi Tidak Valid	invalid_transaction_state
25001	Kelas 25 - Status Transaksi Tidak Valid	active_sql_transaction
25002	Kelas 25 - Status Transaksi Tidak Valid	invalid_transaction_state
25008	Kelas 25 - Status Transaksi Tidak Valid	held_cursor_requires_same_isolation_level
25003	Kelas 25 - Status Transaksi Tidak Valid	tidak sesuai_access_mode_for_branch_transaction
25004	Kelas 25 - Status Transaksi Tidak Valid	tidak sesuai_isolation_level_for_branch_transaction
25005	Kelas 25 - Status Transaksi Tidak Valid	no_active_sql_transaction_for_branch_transaction
25006	Kelas 25 - Status Transaksi Tidak Valid	read_only_sql_transaction
25007	Kelas 25 - Status Transaksi Tidak Valid	no_active_sql_transaction_for_branch_transaction
25P01	Kelas 25 - Status Transaksi Tidak Valid	no_active_sql_transaction

Kode Kesalahan	Kelas Kesalahan	Nama Kondisi
25P02	Kelas 25 - Status Transaksi Tidak Valid	dalam_failed_sql_transaction
26000	Kelas 26 - Nama Pernyataan SQL Tidak Valid	invalid_sql_statement_name
28000	Kelas 28 - Spesifikasi Otorisasi Tidak Valid	invalid_authorization_specification
2B000	Kelas 2B - Deskriptor Hak Istimewa Dependen Masih Ada	dependent_privilege_descriptors_still_exist
2 BP01	Kelas 2B - Deskriptor Hak Istimewa Dependen Masih Ada	dependent_objects_still_exist
2D000	Kelas 2D — Pengakhiran Transaksi Tidak Valid	invalid_transaction_termination
2F000	Kelas 2F - Pengecualian Rutin SQL	sql_routine_exception
2F005	Kelas 2F - Pengecualian Rutin SQL	function_executed_no_return_statement
2F002	Kelas 2F - Pengecualian Rutin SQL	modifying_sql_data_not_permitted
2F003	Kelas 2F - Pengecualian Rutin SQL	prohibited_sql_statement_trial
2F004	Kelas 2F - Pengecualian Rutin SQL	reading_sql_data_not_permitted

Kode Kesalahan	Kelas Kesalahan	Nama Kondisi
34000	Kelas 34 - Nama Kursor Tidak Valid	invalid_cursor_name
38000	Kelas 38 - Pengecualian Rutin Eksternal	external_routine_exception
38001	Kelas 38 - Pengecualian Rutin Eksternal	containing_sql_not_permitted
38002	Kelas 38 - Pengecualian Rutin Eksternal	modifying_sql_data_not_permitted
38003	Kelas 38 - Pengecualian Rutin Eksternal	prohibited_sql_statement_trial
38004	Kelas 38 - Pengecualian Rutin Eksternal	reading_sql_data_not_permitted
39000	Kelas 39 - Pengecualian Doa Rutin Eksternal	external_routine_invocation_exception
39001	Kelas 39 - Pengecualian Doa Rutin Eksternal	invalid_sqlstate_returned
39004	Kelas 39 - Pengecualian Doa Rutin Eksternal	null_value_not_allowed
39P01	Kelas 39 - Pengecualian Doa Rutin Eksternal	trigger_protocol_dilanggar
39P02	Kelas 39 - Pengecualian Doa Rutin Eksternal	srf_protocol_dilanggar
3D000	Kelas 3D - Nama Katalog Tidak Valid	invalid_catalog_name

Kode Kesalahan	Kelas Kesalahan	Nama Kondisi
3F000	Kelas 3F - Nama Skema Tidak Valid	invalid_schema_name
42000	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	syntax_error_or_access_rule_violation
42601	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	syntax_error
42501	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	insufficient_privilege
42846	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	cannot_memaksa
42803	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	grouping_error
42830	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	invalid_foreign_key
42602	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	invalid_name
42622	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	name_too_long

Kode Kesalahan	Kelas Kesalahan	Nama Kondisi
42939	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	reserved_name
42804	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	datatype_mismatch
42P18	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	indeterminate_datatype
42809	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	wrong_object_type
42703	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	undefined_column
42883	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	undefined_function
42P01	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	undefined_table
42P02	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	undefined_parameter
42704	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	undefined_object

Kode Kesalahan	Kelas Kesalahan	Nama Kondisi
42701	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	duplikat_kolom
42P03	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	duplikat_kursor
42P04	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	duplikate_database
42723	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	duplikate_function
42P05	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	duplikate_prepared_statement
42P06	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	duplikate_skema
42P07	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	duplikate_table
42712	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	duplikate_alias
42710	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	duplikate_objek

Kode Kesalahan	Kelas Kesalahan	Nama Kondisi
42702	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	ambiguous_column
42725	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	ambiguous_function
42P08	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	ambigu_parameter
42P09	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	ambigu_alias
42P10	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	invalid_column_reference
42611	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	invalid_column_definition
42P11	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	invalid_cursor_definition
42P12	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	invalid_database_definition
42P13	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	invalid_function_definition

Kode Kesalahan	Kelas Kesalahan	Nama Kondisi
42P14	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	invalid_prepared_statement_definition
42P15	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	invalid_schema_definition
42P16	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	invalid_table_definition
42P17	Kelas 42 — Kesalahan Sintaks atau Pelanggaran Aturan Akses	invalid_object_definition
44000	Kelas 44 — DENGAN CEK OPTION Pelanggaran	dengan_check_option_violation
53000	Kelas 53 - Sumber Daya Tidak Cukup	insufficient_resources
53100	Kelas 53 - Sumber Daya Tidak Cukup	disk_penuh
53200	Kelas 53 - Sumber Daya Tidak Cukup	keluar_of_memori
53300	Kelas 53 - Sumber Daya Tidak Cukup	too_many_connections
54000	Kelas 54 — Batas Program Terlampaui	program_limit_exceeded
54001	Kelas 54 — Batas Program Terlampaui	statement_too_complex

Kode Kesalahan	Kelas Kesalahan	Nama Kondisi
54011	Kelas 54 — Batas Program Terlampaui	too_many_columns
54023	Kelas 54 — Batas Program Terlampaui	too_many_arguments
55000	Kelas 55 — Objek Tidak Dalam Keadaan Prasyarat	object_not_in_prerequisite_state
55006	Kelas 55 — Objek Tidak Dalam Keadaan Prasyarat	object_in_use
55P02	Kelas 55 — Objek Tidak Dalam Keadaan Prasyarat	cant_change_runtime_param
55P03	Kelas 55 — Objek Tidak Dalam Keadaan Prasyarat	lock_not_available
57000	Kelas 57 — Intervensi Operator	operator_intervensi
57014	Kelas 57 — Intervensi Operator	query_canceled
57P01	Kelas 57 — Intervensi Operator	admin_shutdown
57P02	Kelas 57 — Intervensi Operator	crash_shutdown
57P03	Kelas 57 — Intervensi Operator	cannot_connect_now
58000	Kelas 58 - Kesalahan Sistem (kesalahan eksternal ke PostgreSQL)	system_error

Kode Kesalahan	Kelas Kesalahan	Nama Kondisi
58030	Kelas 58 - Kesalahan Sistem (kesalahan eksternal ke PostgreSQL)	io_error
58P01	Kelas 58 - Kesalahan Sistem (kesalahan eksternal ke PostgreSQL)	undefined_file
58P02	Kelas 58 - Kesalahan Sistem (kesalahan eksternal ke PostgreSQL)	duplikat_file
F0000	Kelas F0 - Kesalahan File Konfigurasi	duplikat_file
F0001	Kelas F0 - Kesalahan File Konfigurasi	lock_file_exists
P0000	Kelas P0 - Kesalahan PL/pgSQL	plpgsql_error
P0001	Kelas P0 - Kesalahan PL/pgSQL	raise_exception
P0002	Kelas P0 - Kesalahan PL/pgSQL	no_data_found
P0003	Kelas P0 - Kesalahan PL/pgSQL	too_many_rows
XX000	Kelas XX - Kesalahan Internal	internal_error
XX001	Kelas XX - Kesalahan Internal	data_rusak
XX002	Kelas XX - Kesalahan Internal	index_rusak

Variabel lingkungan Amazon Redshift RSQL

Amazon Redshift RSQL dapat menggunakan variabel lingkungan untuk memilih nilai parameter default.

RSPASSWORD

Important

Kami tidak menyarankan menggunakan variabel lingkungan ini untuk alasan keamanan, karena beberapa sistem operasi memungkinkan pengguna non-administratif untuk melihat variabel lingkungan proses.

Menetapkan kata sandi untuk Amazon Redshift RSQL untuk digunakan saat menghubungkan ke Amazon Redshift. Variabel lingkungan ini memerlukan Amazon Redshift RSQL 1.0.4 ke atas.

RSQL memprioritaskan RSPASSWORD jika ada yang disetel. Jika RSPASSWORD tidak disetel dan Anda terhubung menggunakan DSN, RSQL mengambil kata sandi dari parameter file DSN. Akhirnya, jika RSPASSWORD tidak diatur dan Anda tidak menggunakan DSN, RSQL menyediakan prompt kata sandi setelah mencoba untuk terhubung.

Berikut ini adalah contoh pengaturan RSPASSWORD:

```
export RSPASSWORD=TestPassw0rd
```

Menggunakan profil otentikasi untuk terhubung ke Amazon Redshift

Jika Anda memiliki banyak koneksi ke Amazon Redshift, mungkin sulit untuk mengelola pengaturan untuk semuanya. Seringkali, setiap koneksi JDBC atau ODBC menggunakan opsi konfigurasi tertentu. Dengan menggunakan profil otentikasi, Anda dapat menyimpan opsi koneksi bersama-sama. Dengan cara ini, pengguna Anda dapat memilih profil untuk terhubung dan menghindari pengaturan pengaturan untuk opsi individual. Profil dapat diterapkan ke berbagai skenario dan jenis pengguna.

Setelah Anda membuat profil otentikasi, pengguna dapat menambahkan ready-to-use profil ke string koneksi. Dengan melakukan ini, mereka dapat terhubung ke Amazon Redshift dengan pengaturan yang tepat untuk setiap peran dan kasus penggunaan.

Untuk informasi Amazon Redshift API, lihat. [CreateAuthenticationProfile](#)

Membuat profil otentikasi

Menggunakan AWS CLI, Anda membuat profil otentikasi dengan `create-authentication-profile` perintah. Ini mengasumsikan bahwa Anda memiliki cluster Amazon Redshift yang sudah ada dan database yang ada. Kredensial Anda harus memiliki izin untuk terhubung ke database Amazon Redshift dan hak untuk mengambil profil otentikasi. Anda memberikan opsi konfigurasi sebagai string JSON, atau referensi file yang berisi string JSON Anda.

```
create-authentication-profile --authentication-profile-name<value: String> --  
authentication-profile-content<value: String>
```

Contoh berikut membuat profil yang disebut `ExampleProfileName`. Di sini, Anda dapat menambahkan kunci dan nilai yang menentukan nama cluster Anda dan pengaturan opsi lainnya, sebagai string JSON.

```
create-authentication-profile --authentication-profile-name "ExampleProfileName"  
--authentication-profile-content "{\"AllowDBUserOverride\":\"1\", \"Client_ID\"  
\": \"ExampleClientID\", \"App_ID\": \"ExampleAppID\", \"AutoCreate\": false,  
\"enableFetchRingBuffer\": true, \"databaseMetadataCurrentDbOnly\": true}"  
}
```

Perintah ini membuat profil dengan pengaturan JSON yang ditentukan. Berikut ini dikembalikan, yang menunjukkan bahwa profil dibuat.

```
{"AuthenticationProfileName": "ExampleProfileName",  
"AuthenticationProfileContent": "{\"AllowDBUserOverride\":\"1\",  
\"Client_ID\": \"ExampleClientID\", \"App_ID\": \"ExampleAppID\",  
\"AutoCreate\": false, \"enableFetchRingBuffer\": true,  
\"databaseMetadataCurrentDbOnly\": true}" }
```

Batasan dan kuota untuk membuat profil otentikasi

Setiap pelanggan memiliki kuota sepuluh (10) profil otentikasi.

Kesalahan tertentu dapat terjadi dengan profil otentikasi. Contohnya adalah jika Anda membuat profil baru dengan nama yang sudah ada, atau jika Anda melebihi kuota profil Anda. Untuk informasi selengkapnya, lihat [CreateAuthenticationProfile](#).

Anda tidak dapat menyimpan kunci dan nilai opsi tertentu untuk string koneksi JDBC, ODBC, dan Python di toko profil otentikasi:

- AccessKeyID
- access_key_id
- SecretAccessKey
- secret_access_key_id
- PWD
- Password
- password

Anda tidak dapat menyimpan kunci atau nilai AuthProfile di toko profil, untuk string koneksi JDBC atau ODBC. Untuk koneksi Python, Anda tidak dapat menyimpan. auth_profile

Profil otentikasi disimpan di Amazon DynamoDB dan dikelola oleh. AWS

Menghubungkan dengan profil otentikasi

Setelah Anda membuat profil otentikasi, Anda dapat menyertakan nama profil sebagai opsi koneksi untuk JDBC versi 2.0. AuthProfile Menggunakan opsi koneksi ini mengambil pengaturan yang disimpan.

```
jdbc:redshift:iam://endpoint:port/database?AuthProfile=<Profile-Name>&AccessKeyID=<Caller-Access-Key>&SecretAccessKey=<Caller-Secret-Key>
```

Berikut ini adalah contoh string URL JDBC.

```
jdbc:redshift:iam://examplecluster:us-west-2/dev?AuthProfile="ExampleProfile"&AccessKeyID="AKIAIOSFODNN7EXAMPLE"&SecretAccessKey="wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY"
```

Tentukan AccessKeyID dan SecretAccessKey di URL JDBC, bersama dengan nama profil otentikasi.

Anda juga dapat memisahkan opsi konfigurasi dengan pembatas titik koma, seperti pada contoh berikut, yang mencakup opsi untuk logging.

```
jdbc:redshift:iam://my_redshift_end_point:5439/dev?LogLevel=6;LogPath=/tmp;AuthProfile=my_profile;AccessKeyID="AKIAIOSFODNN7EXAMPLE";SecretAccessKey="wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY"
```

 Note

Jangan menambahkan informasi rahasia ke profil otentikasi. Misalnya, jangan menyimpan `SecretAccessKey` nilai `AccessKeyID` atau dalam profil otentikasi. Toko profil otentikasi memiliki aturan untuk melarang penyimpanan kunci rahasia. Anda mendapatkan kesalahan jika Anda mencoba menyimpan kunci dan nilai yang terkait dengan informasi sensitif.

Mendapatkan profil otentikasi

Untuk membuat daftar profil otentikasi yang ada, hubungi perintah berikut.

```
describe-authentication-profiles --authentication-profile-name <value: String>
```

Contoh berikut menunjukkan dua profil diambil. Semua profil dikembalikan jika Anda tidak menentukan nama profil.

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

Memecahkan masalah koneksi di Amazon Redshift

Jika Anda memiliki masalah dengan menghubungkan ke cluster Anda dari alat klien SQL, ada beberapa hal yang dapat Anda periksa untuk mempersempit masalah. Jika Anda menggunakan sertifikat SSL atau server, pertama-tama hapus kompleksitas ini saat Anda memecahkan masalah koneksi. Kemudian tambahkan ini kembali ketika Anda telah menemukan solusi. Untuk informasi selengkapya, lihat [Mengkonfigurasi opsi keamanan untuk koneksi](#).

Untuk informasi tentang perubahan perilaku dalam fungsionalitas Amazon Redshift yang dapat memengaruhi aplikasi Anda, lihat. [Perubahan perilaku di Amazon Redshift](#)

⚠ Important

Amazon Redshift telah mengubah cara sertifikat SSL dikelola. Jika Anda mengalami masalah saat menghubungkan menggunakan SSL, Anda mungkin perlu memperbarui sertifikat CA root kepercayaan Anda saat ini. Untuk informasi selengkapnya, lihat [Transisi ke sertifikat ACM untuk koneksi SSL](#).

Bagian berikut memiliki beberapa contoh pesan kesalahan dan kemungkinan solusi untuk masalah koneksi. Karena alat klien SQL yang berbeda menyediakan pesan kesalahan yang berbeda, ini bukan daftar lengkap, tetapi harus menjadi titik awal yang baik untuk memecahkan masalah.

Menghubungkan dari luar Amazon EC2 dan menghadapi masalah batas waktu firewall

Koneksi klien Anda ke database tampak hang atau batas waktu saat menjalankan kueri panjang, seperti perintah COPY. Dalam kasus ini, Anda mungkin mengamati bahwa konsol Amazon Redshift menampilkan bahwa kueri telah selesai, tetapi alat klien itu sendiri tampaknya masih menjalankan kueri. Hasil kueri mungkin hilang atau tidak lengkap tergantung kapan koneksi berhenti.

Solusi yang mungkin

Masalah ini terjadi saat Anda terhubung ke Amazon Redshift dari mesin selain instans Amazon EC2. Dalam hal ini, koneksi idle diakhiri oleh komponen jaringan perantara, seperti firewall, setelah periode tidak aktif. Perilaku ini khas ketika Anda masuk dari jaringan pribadi virtual (VPN) atau jaringan lokal Anda.

Untuk menghindari batas waktu ini, kami merekomendasikan perubahan berikut:

- Tingkatkan nilai sistem klien yang berhubungan dengan TCP/IP batas waktu. Buat perubahan ini di komputer yang Anda gunakan untuk terhubung ke cluster Anda. Periode batas waktu harus disesuaikan untuk klien dan jaringan Anda. Untuk informasi selengkapnya, lihat [Ubah TCP/IP pengaturan batas waktu](#).
- Secara opsional, atur perilaku keepalive di tingkat DSN. Untuk informasi selengkapnya, lihat [Ubah pengaturan batas waktu DSN](#).

Ubah TCP/IP pengaturan batas waktu

Untuk mengubah pengaturan TCP/IP batas waktu, konfigurasi pengaturan batas waktu sesuai dengan sistem operasi yang Anda gunakan untuk terhubung ke cluster Anda.

- Linux — Jika klien Anda berjalan di Linux, jalankan perintah berikut sebagai pengguna root untuk mengubah pengaturan batas waktu untuk sesi saat ini:

```
/sbin/sysctl -w net.ipv4.tcp_keepalive_time=200 net.ipv4.tcp_keepalive_intvl=200  
net.ipv4.tcp_keepalive_probes=5
```

Untuk mempertahankan pengaturan, membuat atau memodifikasi file `/etc/sysctl.conf` dengan nilai-nilai berikut kemudian reboot sistem Anda.

```
net.ipv4.tcp_keepalive_time=200  
net.ipv4.tcp_keepalive_intvl=200  
net.ipv4.tcp_keepalive_probes=5
```

- Windows — Jika klien Anda berjalan pada Windows, edit nilai untuk pengaturan registri berikut di bawah `HKEY_LOCAL_MACHINE\SYSTEM\ Services\ Tcpip\ CurrentControlSet Parameters\`:
 - `KeepAliveTime`: 30000
 - `KeepAliveInterval`: 1000
 - `TcpMaxDataRetransmissions`: 10

Pengaturan ini menggunakan tipe data `DWORD`. Jika tidak ada di bawah jalur registri, Anda dapat membuat pengaturan dan menentukan nilai yang disarankan ini. Untuk informasi lebih lanjut tentang mengedit registri Windows, lihat dokumentasi Windows.

Setelah Anda menetapkan nilai-nilai ini, restart komputer Anda agar perubahan diterapkan.

- Mac - Jika klien Anda berjalan di Mac, jalankan perintah berikut untuk mengubah pengaturan batas waktu untuk sesi saat ini:

```
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
```

Untuk mempertahankan pengaturan, membuat atau memodifikasi file `/etc/sysctl.conf` dengan nilai-nilai berikut:

```
net.inet.tcp.keepidle=200000  
net.inet.tcp.keepintvl=200000  
net.inet.tcp.keepinit=200000  
net.inet.tcp.always_keepalive=1
```

Nyalakan kembali komputer Anda, lalu jalankan perintah berikut untuk memverifikasi bahwa nilainya ditetapkan.

```
sysctl net.inet.tcp.keepidle  
sysctl net.inet.tcp.keepintvl  
sysctl net.inet.tcp.keepinit  
sysctl net.inet.tcp.always_keepalive
```

Ubah pengaturan batas waktu DSN

Anda dapat mengatur perilaku keepalive di tingkat DSN jika Anda mau. Anda melakukan ini dengan menambahkan atau memodifikasi parameter berikut dalam file `odbc.ini`:

KeepAlivesCount

Jumlah paket TCP keepalive yang dapat hilang sebelum koneksi dianggap rusak.

KeepAlivesIdle

Jumlah detik tidak aktif sebelum driver mengirim paket TCP keepalive.

KeepAlivesInterval

Jumlah detik antara setiap transmisi ulang TCP keepalive.

Jika parameter ini tidak ada, atau jika mereka memiliki nilai 0, sistem menggunakan parameter keepalive yang ditentukan TCP/IP untuk menentukan perilaku DSN keepalive. Di Windows, Anda dapat menemukan TCP/IP parameter di registri di `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\`. Di Linux dan macOS, Anda dapat menemukan TCP/IP parameter dapat ditemukan di file `sysctl.conf`.

Koneksi ditolak atau gagal

Ketika koneksi Anda ditolak atau gagal, Anda mungkin menerima kesalahan yang mirip dengan salah satu dari berikut ini.

- “Gagal membuat koneksi ke<endpoint>.”
- “Tidak dapat terhubung ke server: Waktu koneksi habis. Apakah server berjalan pada host '<endpoint>' dan menerima koneksi TCP/IP pada port?" '<port>'
- “Koneksi ditolak. Periksa apakah nama host dan port sudah benar dan bahwa postmaster menerima TCP/IP koneksi.”

Solusi yang mungkin

Umumnya, ketika Anda menerima pesan kesalahan yang menunjukkan bahwa ada kegagalan untuk membuat koneksi, itu adalah masalah dengan izin untuk mengakses cluster atau dengan lalu lintas jaringan yang mencapai cluster.

Untuk terhubung ke klaster dari alat klien di luar jaringan tempat klaster berada, Anda menambahkan aturan masuk ke grup keamanan klaster. Konfigurasi aturan tergantung pada apakah cluster Amazon Redshift dibuat di cloud pribadi virtual (VPC):

- Jika Anda membuat klaster Amazon Redshift di cloud pribadi virtual (VPC) berdasarkan Amazon VPC, tambahkan aturan masuk ke grup keamanan VPC yang menentukan alamat klien, di Amazon VPC. CIDR/IP Untuk informasi selengkapnya tentang mengonfigurasi grup keamanan VPC untuk klaster Anda dan opsi yang dapat diakses publik, lihat. [Sumber daya Redshift dalam VPC](#)
- Jika Anda membuat klaster Amazon Redshift di luar VPC, tambahkan CIDR/IP alamat klien Anda ke grup keamanan klaster di Amazon Redshift. Untuk informasi selengkapnya tentang mengonfigurasi grup keamanan klaster, lihat [Grup keamanan Amazon Redshift](#).

Jika Anda mencoba menyambung ke klaster dari alat klien yang berjalan pada instans Amazon EC2, Anda juga menambahkan aturan masuk. Dalam hal ini, tambahkan aturan ke grup keamanan cluster. Aturan harus menentukan grup keamanan Amazon EC2 yang terkait dengan instans Amazon EC2 alat klien.

Dalam beberapa kasus, Anda mungkin memiliki lapisan antara klien dan server Anda, seperti firewall. Dalam kasus ini, pastikan firewall menerima koneksi masuk melalui port yang Anda konfigurasikan untuk cluster Anda.

Klien dan driver tidak kompatibel

Jika klien dan driver Anda tidak kompatibel, Anda mungkin menerima kesalahan yang mengatakan, “DSN yang ditentukan berisi ketidakcocokan arsitektur antara Driver dan Aplikasi.”

Solusi yang mungkin

Ketika Anda mencoba untuk menghubungkan dan mendapatkan kesalahan tentang ketidakcocokan arsitektur, ini berarti bahwa alat klien dan driver tidak kompatibel. Ini terjadi karena arsitektur sistem mereka tidak cocok. Misalnya, ini dapat terjadi jika Anda memiliki alat klien 32-bit tetapi telah menginstal versi 64-bit driver. Terkadang alat klien 64-bit dapat menggunakan driver 32-bit, tetapi Anda tidak dapat menggunakan aplikasi 32-bit dengan driver 64-bit. Pastikan driver dan alat klien menggunakan versi arsitektur sistem yang sama.

Kueri tampaknya hang dan terkadang gagal mencapai cluster

Anda mengalami masalah dengan penyelesaian kueri, di mana kueri tampaknya berjalan tetapi hang di alat klien SQL. Terkadang kueri gagal muncul di cluster, seperti di tabel sistem atau konsol Amazon Redshift.

Solusi yang mungkin

Masalah ini bisa terjadi karena packet drop. Dalam hal ini, ada perbedaan ukuran unit transmisi maksimum (MTU) di jalur jaringan antara dua host Internet Protocol (IP). Ukuran MTU menentukan ukuran maksimum, dalam byte, dari paket yang dapat ditransfer dalam satu bingkai Ethernet melalui koneksi jaringan. Pada tahun AWS, beberapa jenis instans Amazon EC2 mendukung MTU 1500 (frame Ethernet v2) dan jenis instans lainnya mendukung MTU 9001 (bingkai jumbo TCP/IP).

Untuk menghindari masalah yang dapat terjadi dengan perbedaan ukuran MTU, sebaiknya lakukan salah satu hal berikut:

- Jika klaster Anda menggunakan platform EC2-VPC, konfigurasi grup keamanan Amazon VPC dengan aturan Internet Control Message Protocol (ICMP) kustom masuk yang muncul. `Destination Unreachable` Aturan dengan demikian menginstruksikan host asal untuk menggunakan ukuran MTU terendah di sepanjang jalur jaringan. Untuk detail tentang pendekatan ini, lihat [Mengkonfigurasi grup keamanan untuk memungkinkan ICMP "tujuan tidak dapat dijangkau"](#).
- Jika cluster Anda menggunakan platform EC2-Classic, atau Anda tidak dapat mengizinkan aturan masuk ICMP, nonaktifkan bingkai TCP/IP jumbo sehingga frame Ethernet v2 digunakan. Untuk detail tentang pendekatan ini, lihat [Mengkonfigurasi MTU dari sebuah instance](#).

Mengkonfigurasi grup keamanan untuk memungkinkan ICMP “tujuan tidak dapat dijangkau”

Ketika ada perbedaan ukuran MTU di jaringan antara dua host, pertama-tama pastikan bahwa pengaturan jaringan Anda tidak memblokir jalur penemuan MTU (PMTUD). PMTUD memungkinkan host penerima untuk menanggapi host asal dengan pesan ICMP berikut: `Destination Unreachable: fragmentation needed and DF set` (ICMP Type 3, Code 4) Pesan ini menginstruksikan host asal untuk menggunakan ukuran MTU terendah di sepanjang jalur jaringan untuk mengirim ulang permintaan. Tanpa negosiasi ini, paket drop dapat terjadi karena permintaan terlalu besar untuk diterima oleh host penerima. Untuk informasi lebih lanjut tentang pesan ICMP ini, kunjungi situs web Internet Engineering Task Force (IETF). [RFC792](#)

Jika Anda tidak secara eksplisit mengonfigurasi aturan masuk ICMP ini untuk grup keamanan VPC Amazon Anda, PMTUD diblokir. Dalam AWS, grup keamanan adalah firewall virtual yang menentukan aturan untuk lalu lintas masuk dan keluar ke suatu instans. Untuk informasi tentang grup keamanan klaster Amazon Redshift, lihat. [Grup keamanan Amazon Redshift](#) Untuk cluster yang menggunakan platform EC2-VPC, Amazon Redshift menggunakan grup keamanan VPC untuk mengizinkan atau menolak lalu lintas ke cluster. Secara default, grup keamanan dikunci dan menolak semua lalu lintas masuk. Untuk informasi tentang cara menyetel aturan masuk dan keluar untuk instans EC2-Classic atau EC2-VPC, lihat Perbedaan [antara instans di EC2-Classic dan VPC di Panduan Pengguna Amazon](#) EC2.

Untuk informasi selengkapnya tentang cara menambahkan aturan ke grup keamanan VPC, lihat. [Grup keamanan VPC](#) Untuk informasi selengkapnya tentang setelan PMTUD tertentu yang diperlukan dalam aturan ini, lihat [Penemuan Path MTU di Panduan Pengguna](#) Amazon EC2.

Mengkonfigurasi MTU dari sebuah instance

Dalam beberapa kasus, klaster Anda mungkin menggunakan platform EC2-Classic atau Anda tidak dapat mengizinkan aturan ICMP khusus untuk lalu lintas masuk. Dalam kasus ini, kami menyarankan Anda menyesuaikan MTU ke 1500 pada antarmuka jaringan (NIC) dari instans EC2 yang Anda sambungkan ke cluster Amazon Redshift Anda. Penyesuaian ini menonaktifkan bingkai TCP/IP jumbo untuk memastikan bahwa koneksi secara konsisten menggunakan ukuran paket yang sama. Namun, opsi ini mengurangi throughput jaringan maksimum untuk instans sepenuhnya, tidak hanya untuk koneksi ke Amazon Redshift. Untuk informasi selengkapnya, lihat prosedur berikut ini.

Untuk mengatur MTU pada sistem operasi Microsoft Windows

Jika klien Anda berjalan di sistem operasi Microsoft Windows, Anda dapat meninjau dan mengatur nilai MTU untuk adaptor Ethernet dengan menggunakan perintah. `netsh`

1. Jalankan perintah berikut untuk menentukan nilai MTU saat ini:

```
netsh interface ipv4 show subinterfaces
```

2. Tinjau MTU nilai Ethernet adaptor dalam output.
3. Jika nilainya tidak 1500, jalankan perintah berikut untuk mengaturnya:

```
netsh interface ipv4 set subinterface "Ethernet" mtu=1500 store=persistent
```

Setelah Anda menetapkan nilai ini, restart komputer Anda agar perubahan diterapkan.

Untuk mengatur MTU pada sistem operasi Linux

Jika klien Anda berjalan di sistem operasi Linux, Anda dapat meninjau dan mengatur nilai MTU dengan menggunakan `ip` perintah.

1. Jalankan perintah berikut untuk menentukan nilai MTU saat ini:

```
$ ip link show eth0
```

2. Tinjau nilai berikut `mtu` dalam output.
3. Jika nilainya tidak 1500, jalankan perintah berikut untuk mengaturnya:

```
$ sudo ip link set dev eth0 mtu 1500
```

Untuk mengatur MTU pada sistem operasi Mac

- Ikuti petunjuk tentang situs dukungan macOS. [How to change the MTU for troubleshooting purposes](#) Untuk informasi lebih lanjut, cari [situs dukungan](#).

Mengatur parameter ukuran pengambilan JDBC

Secara default, driver Redshift JDBC menggunakan buffer ring untuk mengelola memori secara efisien dan mencegah kesalahan. `out-of-memory` Parameter ukuran pengambilan hanya berlaku ketika buffer cincin dinonaktifkan secara eksplisit. Untuk informasi lebih lanjut, tinjau [tautannya](#). Dalam konfigurasi ini, Anda harus mengatur ukuran pengambilan untuk mengontrol berapa banyak baris yang diambil di setiap batch.

Kapan Menggunakan Ukuran Ambil?

Gunakan parameter ukuran ambil saat:

- Anda membutuhkan kontrol halus atas batching berbasis baris
- Bekerja dengan aplikasi lama yang memerlukan perilaku ukuran pengambilan tradisional

Pengaturan Ukuran Ambil

Ketika buffer dering dinonaktifkan, driver JDBC mengumpulkan semua hasil untuk kueri pada satu waktu secara default. Kueri yang mengembalikan set hasil besar dapat menghabiskan memori yang berlebihan. Untuk mengambil set hasil dalam batch, bukan sekaligus, atur parameter ukuran pengambilan JDBC di aplikasi Anda.

Note

Ukuran pengambilan tidak didukung untuk ODBC.

Untuk kinerja terbaik, atur ukuran pengambilan ke nilai tertinggi yang tidak menyebabkan kesalahan memori. Nilai ukuran pengambilan yang lebih rendah menghasilkan lebih banyak perjalanan server, yang memperpanjang waktu eksekusi. Server menyimpan sumber daya, termasuk slot kueri WLM dan memori terkait, hingga klien mengambil seluruh kumpulan hasil atau kueri dibatalkan. Saat Anda menyetel ukuran pengambilan dengan tepat, sumber daya tersebut dirilis lebih cepat, membuatnya tersedia untuk kueri lain.

Note

Jika Anda perlu mengekstrak kumpulan data besar, sebaiknya gunakan pernyataan [UNLOAD](#) untuk mentransfer data ke Amazon S3. Saat Anda menggunakan UNLOAD, node komputasi bekerja secara paralel untuk mempercepat transfer data.

Untuk informasi selengkapnya tentang pengaturan parameter ukuran pengambilan JDBC, buka [Mendapatkan hasil berdasarkan kursor dalam dokumentasi PostgreSQL](#).

Menggunakan API Data Amazon Redshift

Amazon Redshift Data API menyederhanakan akses ke gudang data Amazon Redshift Anda dengan menghilangkan kebutuhan untuk mengelola driver database, koneksi, konfigurasi jaringan, buffering data, kredensi, dan banyak lagi. Anda dapat menjalankan pernyataan SQL menggunakan operasi Data API dengan AWS SDK. Untuk informasi selengkapnya tentang operasi Data API, lihat [Referensi API Data Amazon Redshift](#).

Data API tidak memerlukan koneksi persisten ke database Anda. Sebaliknya, ia menyediakan titik akhir HTTP yang aman dan integrasi dengan AWS SDKs. Anda dapat menggunakan titik akhir untuk menjalankan pernyataan SQL tanpa mengelola koneksi. Panggilan ke API Data bersifat asinkron. Data API dapat menggunakan kredensial yang disimpan dalam AWS Secrets Manager atau kredensial database sementara. Anda tidak perlu meneruskan kata sandi dalam panggilan API dengan salah satu metode otorisasi. Untuk informasi lebih lanjut tentang AWS Secrets Manager, lihat [Apa itu AWS Secrets Manager?](#) dalam AWS Secrets Manager User Guide. Anda juga dapat menggunakan AWS IAM Identity Center untuk otorisasi.

Dengan API Data, Anda dapat mengakses data Amazon Redshift secara terprogram dengan aplikasi berbasis layanan web, AWS Lambda termasuk notebook Amazon AI, dan SageMaker AWS Cloud9. Untuk informasi lebih lanjut tentang aplikasi ini, lihat [AWS Lambda](#), [Amazon SageMaker AI](#), dan [AWS Cloud9](#).

Untuk mempelajari API Data selengkapnya, lihat [Memulai API Data Amazon Redshift](#) di Blog AWS Big Data.

Bekerja dengan Amazon Redshift Data API

Sebelum Anda menggunakan Amazon Redshift Data API, tinjau langkah-langkah berikut:

1. Tentukan apakah Anda, sebagai pemanggil API Data, diotorisasi. Untuk informasi selengkapnya tentang otorisasi, lihat [Mengotorisasi akses ke API Data Amazon Redshift](#).
2. Tentukan apakah Anda berencana untuk memanggil Data API dengan kredensial otentikasi dari Secrets Manager, kredensial sementara, atau penggunaan AWS IAM Identity Center. Untuk informasi selengkapnya, lihat [Memilih kredensial otentikasi database saat memanggil Amazon Redshift Data API](#).
3. Siapkan rahasia jika Anda menggunakan Secrets Manager untuk kredensial otentikasi. Untuk informasi selengkapnya, lihat [Menyimpan kredensial database di AWS Secrets Manager](#).

4. Tinjau pertimbangan dan batasan saat memanggil Data API. Untuk informasi selengkapnya, lihat [Pertimbangan saat memanggil Amazon Redshift Data API](#).
5. Panggil API Data dari AWS Command Line Interface (AWS CLI), dari kode Anda sendiri, atau menggunakan editor kueri di konsol Amazon Redshift. Untuk contoh panggilan dari AWS CLI, lihat [Memanggil API Data](#).

Pertimbangan saat memanggil Amazon Redshift Data API

Pertimbangkan hal berikut saat memanggil Data API:

- Amazon Redshift Data API dapat mengakses database di kluster yang disediakan Amazon Redshift dan grup kerja Tanpa Server Redshift. Untuk daftar Wilayah AWS tempat Redshift Data API tersedia, lihat titik akhir yang tercantum untuk [Redshift](#) Data API di. Referensi Umum Amazon Web Services
- Durasi maksimum kueri adalah 24 jam.
- Jumlah maksimum kueri aktif (STARTED dan SUBMITTED kueri) per cluster Amazon Redshift adalah 500.
- Ukuran hasil kueri maksimum adalah 500 MB (setelah kompresi gzip). Jika panggilan mengembalikan lebih dari 500 MB data respons, panggilan akan berakhir.
- Waktu retensi maksimum untuk hasil kueri adalah 24 jam.
- Ukuran pernyataan kueri maksimum adalah 100 KB.
- Data API tersedia untuk kueri cluster single-node dan multiple-node dari tipe node berikut:
 - dc2.large
 - dc2.8xlarge
 - ra3. besar
 - ra3.xlplus
 - ra3.4xlarge
 - ra3.16xlarge
- Cluster harus berada di cloud pribadi virtual (VPC) berdasarkan layanan Amazon VPC.
- Secara default, pengguna dengan peran IAM yang sama dengan pelari operasi ExecuteStatement atau BatchExecuteStatement API dapat bertindak berdasarkan pernyataan yang sama dengan CancelStatement,, DescribeStatement GetStatementResultGetStatementResultV2, dan operasi ListStatements API. Untuk

bertindak pada pernyataan SQL yang sama dari pengguna lain, pengguna harus dapat mengambil peran IAM dari pengguna yang menjalankan pernyataan SQL. Untuk informasi selengkapnya tentang cara mengambil peran, lihat [Mengotorisasi akses ke API Data Amazon Redshift](#).

- Pernyataan SQL dalam `Sqls` parameter operasi `BatchExecuteStatement` API dijalankan sebagai satu transaksi. Mereka berjalan secara serial dalam urutan array. Pernyataan SQL berikutnya tidak dimulai sampai pernyataan sebelumnya dalam array selesai. Jika pernyataan SQL gagal, maka karena mereka dijalankan sebagai satu transaksi, semua pekerjaan digulirkan kembali.
- Waktu retensi maksimum untuk token klien yang digunakan dalam `ExecuteStatement` atau operasi `BatchExecuteStatement` API adalah 8 jam.
- Jika kluster yang disediakan Amazon Redshift dan grup kerja Redshift Serverless dienkripsi menggunakan kunci yang dikelola pelanggan Redshift membuat hibah yang memungkinkan Redshift Data API menggunakan kunci untuk operasinya. Untuk informasi lebih lanjut, lihat [Menggunakan AWS KMS dengan Amazon Redshift Data API](#).
- Setiap API di Redshift Data API memiliki kuota transaksi per detik sebelum membatasi permintaan. Untuk kuota, lihat [Kuota untuk Amazon Redshift Data API](#). Jika tingkat permintaan melebihi kuota, a `ThrottlingException` dengan Kode Status HTTP: 400 dikembalikan. Untuk merespons pelambatan, gunakan strategi coba lagi seperti yang dijelaskan dalam [Perilaku Coba lagi di Panduan Referensi](#) Alat AWS SDKs dan Alat. Strategi ini diimplementasikan secara otomatis untuk membatasi kesalahan di beberapa. AWS SDKs

Note

Secara default di AWS Step Functions, percobaan ulang tidak diaktifkan. Jika Anda perlu memanggil Redshift Data API di mesin status Step Functions, sertakan parameter `ClientToken` idempotency dalam panggilan Redshift Data API Anda. Nilai `ClientToken` kebutuhan untuk bertahan di antara percobaan ulang. Dalam contoh cuplikan permintaan `ExecuteStatement` API berikut, ekspresi `States.ArrayGetItem(States.StringSplit($$.Execution.Id, ':'), 7)` menggunakan fungsi intrinsik untuk mengekstrak bagian `UUID$$$.Execution.Id`, yang unik untuk setiap eksekusi mesin status. Untuk informasi selengkapnya, lihat [Fungsi intrinsik](#) di Panduan AWS Step Functions Pengembang.

```
{
  "Database": "dev",
  "Sql": "select 1;",
  "ClusterIdentifier": "MyCluster",
```

```
"ClientToken.$": "States.ArrayGetItem(States.StringSplit($$.Execution.Id,
':'), 7)"
}
```

Memilih kredensi otentikasi database saat memanggil Amazon Redshift Data API

Saat memanggil Data API, Anda menggunakan salah satu metode otentikasi berikut untuk beberapa operasi API. Setiap metode membutuhkan kombinasi parameter yang berbeda.

AWS IAM Identity Center

Data API dapat diakses dengan satu pengguna masuk yang terdaftar di AWS IAM Identity Center. Untuk informasi tentang langkah-langkah untuk mengatur Pusat Identitas IAM, lihat [Menggunakan Data API dengan propagasi identitas tepercaya](#).

AWS Secrets Manager

Dengan metode ini, berikan rahasia `secret-arn` yang disimpan di AWS Secrets Manager mana memiliki `username` dan `password`. Rahasia yang ditentukan berisi kredensial untuk terhubung ke database Anda tentukan. Ketika Anda menghubungkan ke cluster, Anda juga menyediakan nama database. Jika Anda memberikan identifier cluster (`dbClusterIdentifier`), itu harus cocok dengan identifier cluster yang disimpan dalam rahasia. Saat Anda menghubungkan ke workgroup tanpa server, Anda juga menyediakan nama database. Untuk informasi selengkapnya, lihat [Menyimpan kredensi database di AWS Secrets Manager](#).

Dengan metode ini, Anda juga dapat memberikan `region` nilai yang menentukan di AWS Region mana data Anda berada.

Kredensial sementara

Dengan metode ini, pilih salah satu opsi berikut:

- Saat menghubungkan ke workgroup tanpa server, tentukan nama workgroup dan nama database. Nama pengguna database berasal dari identitas IAM. Misalnya, `arn:iam::123456789012:user:foo` memiliki nama pengguna database `IAM:foo`. Juga, izin untuk memanggil `redshift-serverless:GetCredentials` operasi diperlukan.
- Saat menghubungkan ke cluster sebagai identitas IAM, tentukan pengidentifikasi cluster dan nama database. Nama pengguna database berasal dari identitas IAM. Misalnya,

`arn:iam::123456789012:user:foo` memiliki nama pengguna database IAM: `foo`. Juga, izin untuk memanggil `redshift:GetClusterCredentialsWithIAM` operasi diperlukan.

- Saat menghubungkan ke cluster sebagai pengguna database, tentukan pengidentifikasi cluster, nama database, dan nama pengguna database. Juga, izin untuk memanggil `redshift:GetClusterCredentials` operasi diperlukan. Untuk informasi tentang cara bergabung dengan grup database saat menghubungkan dengan metode ini, lihat [Bergabung dengan grup basis data saat menghubungkan ke klaster](#).

Dengan metode ini, Anda juga dapat memberikan `region` nilai yang menentukan di AWS Region mana data Anda berada.

Memetakan tipe data JDBC saat memanggil Amazon Redshift Data API

Tabel berikut memetakan tipe data Java Database Connectivity (JDBC) ke jenis data yang Anda tentukan dalam panggilan API Data.

Jenis data JDBC	Jenis data API Data
INTEGER, SMALLINT, BIGINT	LONG
FLOAT, REAL, DOUBLE	DOUBLE
DECIMAL	STRING
BOOLEAN, BIT	BOOLEAN
BLOB, BINARY, LONGVARBINARY	BLOB
VARBINARY	STRING
CLOB	STRING
Jenis lainnya (termasuk jenis yang terkait dengan tanggal dan waktu)	STRING

Nilai string diteruskan ke database Amazon Redshift dan secara implisit diubah menjadi tipe data database.

 Note

Saat ini, Data API tidak mendukung array pengidentifikasi unik universal (UUIDs).

Menjalankan pernyataan SQL dengan parameter saat memanggil Amazon Redshift Data API

Anda dapat mengontrol teks SQL yang dikirimkan ke mesin database dengan memanggil operasi Data API menggunakan parameter untuk bagian dari pernyataan SQL. Parameter bernama menyediakan cara yang fleksibel untuk meneruskan parameter tanpa hardcoding mereka dalam teks SQL. Mereka membantu Anda menggunakan kembali teks SQL dan menghindari masalah injeksi SQL.

Contoh berikut menunjukkan parameter bernama dari parameters bidang execute-statement AWS CLI perintah.

```
--parameters "[{\"name\": \"id\", \"value\": \"1\"},{\"name\": \"address\", \"value\": \"Seattle\"}]"
```

Pertimbangkan hal berikut saat menggunakan parameter bernama:

- Parameter bernama hanya dapat digunakan untuk mengganti nilai dalam pernyataan SQL.
- Anda dapat mengganti nilai dalam pernyataan INSERT, seperti `INSERT INTO mytable VALUES (:val1)`.

Parameter bernama dapat dalam urutan apa pun dan parameter dapat digunakan lebih dari satu kali dalam teks SQL. Opsi parameter yang ditunjukkan dalam contoh sebelumnya, nilai-nilai 1 dan Seattle dimasukkan ke dalam kolom tabel id dan address. Dalam teks SQL, Anda menentukan parameter bernama sebagai berikut:

```
--sql "insert into mytable values (:id, :address)"
```

- Anda dapat mengganti nilai dalam klausa kondisi, seperti `WHERE attr >= :val1`, `WHERE attr BETWEEN :val1 AND :val2`, dan `HAVING COUNT(attr) > :val`.
- Anda tidak dapat mengganti nama kolom dalam pernyataan SQL, seperti `SELECT column-name`, `ORDER BY column-name`, atau `GROUP BY column-name`.

Misalnya, pernyataan SELECT berikut gagal dengan sintaks yang tidak valid.

```
--sql "SELECT :colname, FROM event" --parameters "[{\"name\": \"colname\", \"value\": \"eventname\"}]"
```

Jika Anda menjelaskan (`describe-statement` operasi) pernyataan dengan kesalahan sintaks, yang `QueryString` dikembalikan tidak menggantikan nama kolom untuk parameter ("`QueryString`": "SELECT :colname, FROM event"), dan kesalahan dilaporkan (ERROR: kesalahan sintaks di atau dekat "FROM" \nPosisi: 12).

- Anda tidak dapat mengganti nama kolom dalam fungsi agregat, seperti `COUNT(column-name)`, `AVG(column-name)`, atau `SUM(column-name)`.
- Anda tidak dapat mengganti nama kolom dalam klausa JOIN.
- Ketika SQL berjalan, data secara implisit dilemparkan ke tipe data. Untuk informasi selengkapnya tentang casting tipe data, lihat Tipe [data](#) di Panduan Pengembang Database Amazon Redshift.
- Anda tidak dapat menetapkan nilai ke NULL. Data API menafsirkannya sebagai string NULL literal. Contoh berikut menggantikan id dengan string `null` literal. Bukan nilai SQL NULL.

```
--parameters "[{\"name\": \"id\", \"value\": \"null\"}]"
```

- Anda tidak dapat menetapkan nilai panjang nol. Pernyataan Data API SQL gagal. Contoh berikut mencoba untuk mengatur id dengan nilai panjang nol dan menghasilkan kegagalan pernyataan SQL.

```
--parameters "[{\"name\": \"id\", \"value\": \"\"}]"
```

- Anda tidak dapat mengatur nama tabel dalam pernyataan SQL dengan parameter. Data API mengikuti aturan `JDBCPreparedStatement`.
- Output dari `describe-statement` operasi mengembalikan parameter query dari pernyataan SQL.
- Hanya `execute-statement` operasi yang mendukung pernyataan SQL dengan parameter.

Menjalankan pernyataan SQL dengan token idempotensi saat memanggil Amazon Redshift Data API

Saat Anda membuat permintaan API yang bermutasi, permintaan biasanya menampilkan hasil sebelum alur kerja asinkron operasi selesai. Operasi mungkin juga habis atau mengalami masalah server lain sebelum selesai, meskipun permintaan telah mengembalikan hasilnya. Hal ini dapat membuat sulit untuk menentukan apakah permintaan berhasil atau tidak, dan dapat menyebabkan beberapa percobaan ulang untuk memastikan bahwa operasi selesai dengan sukses. Namun, jika permintaan asli dan percobaan ulang berikutnya berhasil, operasi selesai beberapa kali. Ini berarti Anda dapat memperbarui lebih banyak sumber daya daripada yang Anda inginkan.

Idempotency memastikan bahwa permintaan API selesai tidak lebih dari satu kali. Dengan permintaan idempoten, jika permintaan asli berhasil diselesaikan, percobaan ulang berikutnya berhasil diselesaikan tanpa melakukan tindakan lebih lanjut. Data API `ExecuteStatement` dan `BatchExecuteStatement` operasi memiliki parameter `ClientToken` idempoten opsional. `ClientToken` kedaluwarsa setelah 8 jam.

Important

Jika Anda memanggil `ExecuteStatement` dan `BatchExecuteStatement` mengoperasikannya dari AWS SDK, secara otomatis akan menghasilkan token klien untuk digunakan saat mencoba lagi. Dalam hal ini, kami tidak menyarankan menggunakan `client-token` parameter dengan `ExecuteStatement` dan `BatchExecuteStatement` operasi. Lihat CloudTrail log untuk melihat `ClientToken`. Untuk contoh CloudTrail log, lihat [Contoh API Data Amazon Redshift](#).

`execute-statement` AWS CLI Perintah berikut menggambarkan `client-token` parameter opsional untuk idempotensi.

```
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
```


Tabel berikut menunjukkan beberapa tanggapan umum yang mungkin Anda dapatkan untuk permintaan API idempoten, dan memberikan rekomendasi coba ulang.

Respons	Rekomendasi	Komentar
200 (OK)	Jangan coba lagi	Permintaan asli berhasil diselesaikan. Setiap percobaan ulang berikutnya berhasil kembali.
Kode respons 400 seri	Jangan coba lagi	Ada masalah dengan permintaan, dari antara yang berikut: • Ini termasuk parameter atau kombinasi parameter yang tidak valid. • Ini menggunakan tindakan atau sumber daya yang Anda tidak memiliki izin. • Ini menggunakan sumber daya yang sedang dalam proses mengubah keadaan. Jika permintaan melibatkan sumber daya yang sedang dalam proses mengubah status, mencoba kembali permintaan mungkin berhasil.
Kode respons 500 seri	Coba lagi	Kesalahan ini disebabkan oleh masalah AWS sisi server dan umumnya bersifat sementara . Ulangi permintaan dengan strategi backoff yang sesuai.

Untuk informasi tentang kode respons Amazon Redshift, lihat [Kesalahan Umum](#) di Referensi API Amazon Redshift.

Menjalankan pernyataan SQL dengan penggunaan kembali sesi saat memanggil Amazon Redshift Data API

Ketika Anda membuat permintaan API untuk menjalankan pernyataan SQL, sesi di mana SQL berjalan biasanya dihentikan ketika SQL selesai. Untuk menjaga sesi tetap aktif selama

beberapa detik tertentu, API `ExecuteStatement` dan `BatchExecuteStatement` operasi memiliki `SessionKeepAliveSeconds` parameter opsional. Bidang `SessionId` respons berisi identitas sesi yang kemudian dapat digunakan dalam `BatchExecuteStatement` operasi berikutnya `ExecuteStatement`. Dalam panggilan berikutnya Anda dapat menentukan yang lain `SessionKeepAliveSeconds` untuk mengubah waktu batas waktu idle. Jika `SessionKeepAliveSeconds` tidak diubah, pengaturan batas waktu idle awal tetap ada. Pertimbangkan hal berikut saat menggunakan penggunaan kembali sesi:

- Nilai maksimum `SessionKeepAliveSeconds` adalah 24 jam.
- Sesi ini dapat berlangsung paling lama 24 jam. Setelah 24 jam sesi ditutup secara paksa dan kueri yang sedang berlangsung dihentikan.
- Jumlah maksimum sesi per cluster Amazon Redshift atau grup kerja Redshift Serverless adalah 500.
- Anda hanya dapat menjalankan satu kueri pada satu waktu dalam satu sesi. Anda harus menunggu sampai kueri selesai untuk menjalankan kueri berikutnya di sesi yang sama. Artinya, Anda tidak dapat menjalankan kueri secara paralel dalam sesi yang disediakan.
- Data API tidak dapat mengantri kueri untuk sesi tertentu.

Untuk mengambil `SessionId` yang digunakan oleh panggilan ke `ExecuteStatement` dan `BatchExecuteStatement` operasi, panggilan `DescribeStatement` dan `ListStatements` operasi.

Contoh berikut menunjukkan penggunaan `SessionKeepAliveSeconds` dan `SessionId` parameter untuk menjaga sesi tetap hidup dan digunakan kembali. Pertama, panggil `execute-statement` AWS CLI perintah dengan `session-keep-alive-seconds` parameter opsional diatur ke 2.

```
aws redshift-data execute-statement
  --session-keep-alive-seconds 2
  --sql "select 1"
  --database dev
  --workgroup-name mywg
```

Respons berisi pengenalan sesi.

```
{
  "WorkgroupName": "mywg",
```

```
"CreatedAt": 1703022996.436,  
"Database": "dev",  
"DbUser": "awsuser",  
"Id": "07c5ffea-76d6-4786-b62c-4fe3ef529680",  
"SessionId": "5a254dc6-4fc2-4203-87a8-551155432ee4"  
}
```

Kemudian, panggil `execute-statement` AWS CLI perintah dengan yang `SessionId` dikembalikan dari panggilan pertama. Dan secara opsional, tentukan `session-keep-alive-seconds` parameter yang disetel 10 untuk mengubah nilai batas waktu idle.

```
aws redshift-data execute-statement  
--sql "select 1"  
--session-id 5a254dc6-4fc2-4203-87a8-551155432ee4  
--session-keep-alive-seconds 10
```

Mengambil hasil pernyataan SQL

Anda menggunakan operasi Data API yang berbeda untuk mengambil hasil SQL tergantung pada format hasil. Saat Anda memanggil `ExecuteStatement` dan `BatchExecuteStatement` operasi, Anda dapat menentukan apakah hasilnya diformat sebagai JSON atau CSV. Jika Anda tidak menentukan, defaultnya adalah JSON. Untuk mengambil hasil JSON, gunakan operasi. `GetStatementResult` Untuk mengambil hasil CSV, gunakan operasi. `GetStatementResultV2`

Hasil yang dikembalikan dalam format JSON adalah catatan yang menyertakan metadata tentang setiap kolom. Setiap rekaman dalam format JSON. Misalnya, respons dari `GetStatementResult` terlihat mirip dengan ini:

```
{  
  "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>
}

```

Hasil yang dikembalikan dalam format CSV adalah catatan yang menyertakan metadata tentang setiap kolom. Hasil dikembalikan dalam potongan 1 MB, di mana setiap potongan dapat menyimpan sejumlah baris dalam format CSV. Setiap permintaan mengembalikan hasil hingga 15 MB. Jika hasilnya lebih besar dari 15 MB, maka token halaman berikutnya dikembalikan untuk melanjutkan pengambilan hasilnya. Misalnya, respons dari `GetStatementResultV2` terlihat mirip dengan ini:

```

{
  "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>
}

```

Mengotorisasi akses ke API Data Amazon Redshift

Untuk mengakses Data API, pengguna harus diberi wewenang. Anda dapat mengotorisasi pengguna untuk mengakses Data API dengan menambahkan kebijakan terkelola, yang merupakan kebijakan standar AWS Identity and Access Management (IAM), ke pengguna tersebut. Sebagai praktik terbaik, kami merekomendasikan untuk melampirkan kebijakan izin ke peran IAM dan kemudian menetapkan ke pengguna dan grup sesuai kebutuhan. Untuk informasi selengkapnya, lihat [Identitas dan manajemen akses di Amazon Redshift](#). Untuk melihat izin yang diizinkan dan ditolak oleh kebijakan terkelola, lihat konsol IAM () <https://console.aws.amazon.com/iam/>.

Mengonfigurasi izin IAM

Amazon Redshift menyediakan kebijakan `AmazonRedshiftDataFullAccess` terkelola. Kebijakan ini menyediakan akses penuh ke operasi Amazon Redshift Data API. Kebijakan ini juga memungkinkan akses tercakup ke Amazon Redshift tertentu AWS Secrets Manager,, dan operasi IAM API yang diperlukan untuk mengautentikasi dan mengakses kluster Amazon Redshift atau grup kerja Tanpa Server Redshift.

Anda juga dapat membuat kebijakan IAM Anda sendiri yang memungkinkan akses ke sumber daya tertentu. Untuk membuat kebijakan Anda, gunakan `AmazonRedshiftDataFullAccess` kebijakan sebagai templat awal Anda. Setelah membuat kebijakan, tambahkan ke setiap pengguna yang memerlukan akses ke API Data.

Pertimbangkan persyaratan berikut dari kebijakan IAM yang terkait dengan pengguna:

- Jika Anda menggunakan AWS Secrets Manager untuk mengautentikasi, konfirmasi kebijakan mengizinkan penggunaan `secretsmanager:GetSecretValue` tindakan untuk mengambil rahasia yang ditandai dengan kunci. `RedshiftDataFullAccess`
- Jika Anda menggunakan kredensi sementara untuk mengautentikasi ke kluster, konfirmasi kebijakan tersebut mengizinkan penggunaan `redshift:GetClusterCredentials` tindakan ke nama pengguna database `redshift_data_api_user` untuk database apa pun di kluster. Nama pengguna ini harus sudah dibuat di database Anda.
- Jika Anda menggunakan kredensi sementara untuk mengautentikasi ke grup kerja tanpa server, konfirmasi kebijakan tersebut mengizinkan penggunaan `redshift-serverless:GetCredentials` tindakan untuk mengambil grup kerja yang diberi tag dengan kunci tersebut. `RedshiftDataFullAccess` Pengguna database dipetakan 1:1 ke identitas sumber AWS Identity and Access Management (IAM). Misalnya, pengguna `sample_user` dipetakan ke pengguna database `IAM:sample_user`, dan peran IAM `sample_role` dipetakan ke.

IAM: `sample_role` Untuk informasi selengkapnya tentang identitas IAM, lihat [Identitas IAM \(pengguna, grup pengguna, dan peran\)](#) di Panduan Pengguna IAM.

- Tindakan IAM `redshift-data:GetStatementResult` memungkinkan akses ke operasi keduanya `GetStatementResult` dan `GetStatementResultV2` API.

Tautan berikut memberikan informasi lebih lanjut tentang AWS Identity and Access Management Panduan Pengguna IAM.

- Untuk informasi tentang membuat peran IAM, lihat [Membuat peran IAM](#).
- Untuk informasi tentang membuat kebijakan IAM, lihat [Membuat kebijakan IAM](#).
- Untuk informasi tentang menambahkan kebijakan IAM ke pengguna, lihat [Menambahkan dan menghapus izin identitas IAM](#).

Jalankan kueri pada klaster yang dimiliki oleh akun lain

Untuk menjalankan kueri pada klaster yang dimiliki oleh akun lain, akun pemilik harus menyediakan peran IAM yang dapat diasumsikan oleh API Data di akun panggilan. Misalnya, Akun B memiliki klaster yang perlu diakses oleh Akun A. Akun B dapat melampirkan kebijakan AWS terkelola `AmazonRedshiftDataFullAccess` ke peran IAM Akun B. Kemudian Akun B mempercayai Akun A menggunakan kebijakan kepercayaan seperti berikut:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:role/someRoleA"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Terakhir, peran IAM Akun A harus dapat mengambil peran Akun B IAM.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource": "arn:aws:iam::111122223333:role/someRoleB"
  }
}
```

Tentukan peran IAM yang membatasi sumber daya ke grup kerja Redshift Tanpa Server dan kluster Amazon Redshift di Akun AWS

Anda dapat menentukan sumber daya ARNs dalam kebijakan berbasis identitas untuk mengontrol akses ke grup kerja Redshift Tanpa Server dan kluster Amazon Redshift di file. Akun AWS Contoh ini menunjukkan cara Anda membuat kebijakan yang mengizinkan akses ke Data API hanya untuk workgroup dan cluster dalam yang ditentukan. Akun AWS

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:*"
    ]
}

```

Mengonfigurasi kebijakan IAM yang membatasi akses ke informasi pernyataan SQL hanya untuk pemilik pernyataan

Secara default, Amazon Redshift Data API memperlakukan peran IAM yang digunakan saat memanggil `ExecuteStatement` dan `BatchExecuteStatement` sebagai pemilik pernyataan SQL. Siapa pun yang diizinkan untuk mengambil peran dapat mengakses informasi tentang pernyataan SQL, termasuk hasilnya. Untuk membatasi akses informasi pernyataan SQL ke sesi peran IAM dengan pemilik tertentu, tambahkan kondisi. `redshift-data:statement-owner-iam-userid: "${aws:userid}"` Kebijakan IAM berikut membatasi akses.

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}"
        }
      }
    }
  ]
}

```

Anda dapat menggunakan kondisi `statement-owner-iam-userid` dengan `CancelStatement`, `DescribeStatement`, `GetStatementResult`, dan `ListStatements`. Untuk informasi selengkapnya, lihat [Tindakan yang ditentukan oleh Amazon Redshift Data API](#).

Konfigurasi kebijakan IAM yang membatasi akses ke hasil SQL hanya untuk pemilik sesi

Secara default, Amazon Redshift Data API memperlakukan peran IAM yang digunakan saat memanggil `ExecuteStatement` dan `BatchExecuteStatement` sebagai pemilik sesi database yang menjalankan pernyataan SQL. Siapa pun yang diizinkan untuk mengambil peran dapat mengirimkan kueri ke sesi database. Untuk membatasi akses sesi ke sesi peran IAM dengan pemilik tertentu, tambahkan kondisi. `redshift-data:session-owner-iam-userid: "${aws:userid}"` Kebijakan IAM berikut membatasi akses.

Kebijakan IAM berikut hanya memungkinkan pemilik sesi untuk mendapatkan hasil pernyataan. Kondisi `session-owner-iam-userid` ini digunakan untuk membatasi akses sumber daya ke yang ditentukan `userid`.

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}"
        }
      }
    }
  ]
}
```

Anda dapat menggunakan kondisi `session-owner-iam-userid` dengan `ExecuteStatement` dan `BatchExecuteStatement`. Untuk informasi selengkapnya, lihat [Tindakan yang ditentukan oleh Amazon Redshift Data API](#).

Menyimpan kredensi database di AWS Secrets Manager

Saat memanggil Data API, Anda dapat meneruskan kredensial untuk klaster atau grup kerja tanpa server dengan menggunakan secret in. AWS Secrets Manager Untuk memberikan kredensial dengan cara ini, Anda menentukan nama rahasia atau Amazon Resource Name (ARN) rahasia tersebut.

Untuk menyimpan kredensial dengan Secrets Manager, Anda memerlukan izin kebijakan `SecretManagerReadWrite` terkelola. Untuk informasi selengkapnya tentang izin minimum, lihat [Membuat dan Mengelola Rahasia dengan AWS Secrets Manager](#) di Panduan AWS Secrets Manager Pengguna.

Untuk menyimpan kredensial Anda secara rahasia untuk cluster Amazon Redshift

1. Gunakan AWS Secrets Manager konsol untuk membuat rahasia yang berisi kredensial untuk klaster Anda:
 - Saat Anda memilih Simpan rahasia baru, pilih Credentials for Redshift cluster.
 - Simpan nilai Anda untuk nama Pengguna (pengguna basis data), Kata Sandi, dan cluster DB (pengidentifikasi cluster) dalam rahasia Anda.
 - Tandai rahasia dengan kuncinya `RedshiftDataFullAccess`. Kebijakan AWS terkelola `AmazonRedshiftDataFullAccess` hanya mengizinkan tindakan `secretsmanager:GetSecretValue` untuk rahasia yang ditandai dengan kunci `RedshiftDataFullAccess`.

Untuk instruksi, lihat [Membuat Rahasia Basic](#) di Panduan Pengguna AWS Secrets Manager .

2. Gunakan AWS Secrets Manager konsol untuk melihat detail rahasia yang Anda buat, atau jalankan `aws secretsmanager describe-secret` AWS CLI perintah.

Catat nama dan ARN rahasia. Anda dapat menggunakan ini dalam panggilan ke Data API.

Untuk menyimpan kredensial Anda secara rahasia untuk grup kerja tanpa server

1. Gunakan AWS Secrets Manager AWS CLI perintah untuk menyimpan rahasia yang berisi kredensial untuk workgroup tanpa server Anda:

- Buat rahasia Anda dalam sebuah file, misalnya file JSON bernama `mycreds.json`. Berikan nilai untuk nama Pengguna (pengguna database) dan Kata Sandi dalam file.

```
{
  "username": "myusername",
  "password": "mypassword"
}
```

- Simpan nilai-nilai Anda dalam rahasia Anda dan tandai rahasia dengan kuncinya `RedshiftDataFullAccess`.

```
aws secretsmanager create-secret --name MyRedshiftSecret --tags
  Key="RedshiftDataFullAccess",Value="serverless" --secret-string file://
mycreds.json
```

Berikut ini menunjukkan output.

```
{
  "ARN":
    "arn:aws:secretsmanager:region:accountId:secret:MyRedshiftSecret-mvLHxf",
  "Name": "MyRedshiftSecret",
  "VersionId": "a1603925-e8ea-4739-9ae9-e509eEXAMPLE"
}
```

Untuk informasi selengkapnya, lihat [Membuat Rahasia Dasar dengan AWS CLI](#) di Panduan AWS Secrets Manager Pengguna.

2. Gunakan AWS Secrets Manager konsol untuk melihat detail rahasia yang Anda buat, atau jalankan `aws secretsmanager describe-secret` AWS CLI perintah.

Catat nama dan ARN rahasia. Anda dapat menggunakan ini dalam panggilan ke Data API.

Membuat endpoint (AWS PrivateLink) Amazon VPC untuk API Data

Amazon Virtual Private Cloud (Amazon VPC) memungkinkan Anda meluncurkan AWS sumber daya, seperti kluster dan aplikasi Amazon Redshift, ke cloud pribadi virtual (VPC). AWS PrivateLink menyediakan konektivitas pribadi antara virtual private cloud (VPCs) dan AWS layanan dengan aman di jaringan Amazon. Dengan menggunakan AWS PrivateLink, Anda dapat membuat titik akhir

VPC, yang dapat Anda gunakan sambungkan ke layanan di berbagai akun dan VPCs berdasarkan Amazon VPC. Untuk informasi selengkapnya AWS PrivateLink, lihat [Layanan Titik Akhir VPC \(AWS PrivateLink\)](#) di Panduan Pengguna Amazon Virtual Private Cloud.

Anda dapat memanggil titik akhir API Data dengan titik akhir Amazon VPC. Dengan menggunakan titik akhir Amazon VPC, Anda akan menjaga lalu lintas antara aplikasi di Amazon VPC Anda dan API Data di jaringan AWS, tanpa menggunakan alamat IP publik. Titik akhir Amazon VPC dapat membantu Anda memenuhi persyaratan kepatuhan dan peraturan yang berkaitan dengan membatasi konektivitas internet publik. Misalnya, jika Anda menggunakan titik akhir Amazon VPC, Anda dapat mempertahankan lalu lintas antara aplikasi yang berjalan pada instans Amazon EC2 dan API Data di VPCs yang memuatnya.

Setelah membuat titik akhir Amazon VPC, Anda dapat mulai menggunakannya tanpa membuat kode atau perubahan konfigurasi apa pun di aplikasi Anda.

Membuat titik akhir Amazon VPC untuk API Data

1. Masuk ke Konsol Manajemen AWS dan buka konsol VPC Amazon di <https://console.aws.amazon.com/vpc/>
2. Pilih Titik Akhir, lalu pilih Buat Titik Akhir.
3. Di halaman Buat Titik Akhir, untuk Kategori layanan, pilih Layanan AWS. Untuk Nama Layanan, pilih `redshift-data()`. `com.amazonaws.region.redshift-data`
4. Untuk VPC, pilih VPC tempat membuat titik akhir.

Pilih VPC yang berisi aplikasi yang membuat panggilan API Data.

5. Untuk Subnet, pilih subnet untuk setiap Availability Zone (AZ) yang digunakan oleh AWS layanan yang menjalankan aplikasi Anda.

Untuk membuat titik akhir VPC Amazon, tentukan rentang alamat IP pribadi tempat titik akhir dapat diakses. Untuk melakukannya, pilih subnet untuk setiap Zona Ketersediaan. Hal tersebut membatasi titik akhir VPC ke rentang alamat IP pribadi khusus untuk setiap Zona Ketersediaan dan membuat titik akhir Amazon VPC di setiap Zona Ketersediaan.

6. Untuk Aktifkan nama DNS, pilih Aktifkan untuk titik akhir ini.

DNS Pribadi menyelesaikan nama host DNS API Data standar (`https://redshift-data.region.amazonaws.com`) ke alamat IP pribadi yang dikaitkan dengan nama host DNS khusus untuk titik akhir Amazon VPC Anda. Akibatnya, Anda dapat mengakses titik akhir

VPC API Data menggunakan AWS CLI atau AWS SDKs tanpa membuat perubahan kode atau konfigurasi apa pun untuk memperbarui URL titik akhir API Data.

7. Untuk Grup keamanan, pilih satu grup keamanan untuk dikaitkan dengan titik akhir Amazon VPC.

Pilih grup keamanan yang memungkinkan akses ke AWS layanan yang menjalankan aplikasi Anda. Misalnya, jika instans Amazon EC2 menjalankan aplikasi Anda, pilih grup keamanan yang mengizinkan akses ke instans Amazon EC2. Grup keamanan memungkinkan Anda mengontrol lalu lintas ke titik akhir Amazon VPC dari sumber daya di VPC Anda.

8. Pilih Buat titik akhir.

Setelah titik akhir dibuat, pilih tautan di Konsol Manajemen AWS untuk melihat detail titik akhir.

Tab Detail titik akhir menunjukkan nama host DNS yang dibuat saat membuat titik akhir Amazon VPC.

Anda dapat menggunakan titik akhir standar (`redshift-data.region.amazonaws.com`) atau salah satu titik akhir khusus VPC untuk memanggil API Data dalam Amazon VPC. Titik akhir API Data standar secara otomatis merutekan ke titik akhir Amazon VPC. Perutean ini terjadi karena nama host DNS Pribadi diaktifkan saat titik akhir Amazon VPC dibuat.

Saat Anda menggunakan titik akhir VPC Amazon dalam panggilan API Data, semua lalu lintas antara aplikasi Anda dan API Data tetap berada di Amazon VPCs yang mengandungnya. Anda dapat menggunakan titik akhir Amazon VPC untuk semua jenis panggilan API Data. Lihat informasi selengkapnya tentang memanggil API Data di [Pertimbangan saat memanggil Amazon Redshift Data API](#).

Bergabung dengan grup basis data saat menghubungkan ke kluster

Grup database adalah kumpulan pengguna database. Hak istimewa database dapat diberikan kepada grup. Administrator dapat mengonfigurasi peran IAM sedemikian rupa sehingga grup database ini diperhitungkan saat SQL Anda berjalan dengan API Data. Untuk informasi selengkapnya tentang grup database, lihat [Grup](#) di Panduan Pengembang Database Amazon Redshift.

Anda dapat mengonfigurasi peran IAM pemanggil API Data sehingga pengguna database yang ditentukan dalam panggilan bergabung dengan grup database saat API Data terhubung ke kluster. Kemampuan ini hanya didukung saat menghubungkan ke cluster yang disediakan. Itu tidak didukung saat menghubungkan ke grup kerja Redshift Serverless. Peran IAM dari pemanggil Data API juga harus memungkinkan tindakan. `redshift:JoinGroup`

Konfigurasi ini dengan menambahkan tag ke peran IAM. Administrator peran IAM memanggil menambahkan tag dengan kunci `RedshiftDbGroups` dan nilai kunci dari daftar grup database. Nilainya adalah daftar nama titik dua (:) dipisahkan dari kelompok database hingga total panjang 256 karakter. Grup database harus didefinisikan sebelumnya dalam database yang terhubung. Jika grup tertentu tidak ditemukan dalam database, itu diabaikan. Misalnya, untuk grup database `accounting` dan `retail`, nilai kunci adalah `accounting:retail`. Pasangan nilai kunci tag `{"Key": "RedshiftDbGroups", "Value": "accounting:retail"}` digunakan oleh Data API untuk menentukan grup database mana yang terkait dengan pengguna database yang disediakan dalam panggilan ke Data API.

Untuk bergabung dengan grup basis data

1. Masuk ke Konsol Manajemen AWS dan buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di panel navigasi konsol, pilih Peran lalu pilih nama peran yang ingin Anda edit.
3. Pilih tab Tag, lalu pilih Kelola tag.
4. Pilih Tambah tag, lalu tambahkan kunci `RedshiftDbGroups` dan nilai yang merupakan daftar *database-groups-colon-separated*.
5. Pilih Simpan perubahan.

Sekarang ketika prinsipal IAM (dengan peran IAM ini terpasang) memanggil Data API, pengguna database yang ditentukan bergabung dengan grup database yang ditentukan dalam peran IAM.

Untuk informasi selengkapnya tentang cara melampirkan tag ke prinsipal, termasuk peran IAM dan pengguna IAM, lihat [Menandai sumber daya IAM](#) di Panduan Pengguna IAM.

Menggunakan Data API dengan propagasi identitas tepercaya

Sebagai administrator akun Amazon Redshift, Anda dapat mengintegrasikan kluster atau grup kerja Amazon Redshift, yang membantu mengelola akses tenaga kerja Anda ke Amazon Redshift AWS IAM Identity Center dengan sistem masuk tunggal. Untuk informasi selengkapnya, lihat [Menyiapkan integrasi Pusat AWS Identitas IAM dengan Amazon Redshift](#). Amazon Redshift Data API mendukung penyebaran identitas pengguna IAM Identity Center ke cluster atau workgroup Amazon Redshift, dan ke layanan lain, seperti, di bawah rantai. AWS Lake Formation Anda dapat mengatur dan melakukan kueri menggunakan API Data dengan mengikuti langkah-langkah dalam [AWS layanan Access secara terprogram menggunakan propagasi identitas tepercaya](#).

Saat Anda memanggil API Data menggunakan identitas pengguna IAM Identity Center dari sesi peran IAM yang disempurnakan identitas, Anda hanya dapat mengakses pernyataan dan hasil pernyataan yang dihasilkan menggunakan pengguna Pusat Identitas IAM yang sama. Misalnya, AWS CLI perintah berikut memanggil `execute-statement` operasi untuk menjalankan perintah SQL dengan propagasi identitas terpercaya.

```
aws redshift-data execute-statement
--sql "select current_user;"
--cluster-id mycluster
--database dev
```

AWS CLI Perintah berikut memanggil `batch-execute-statement` operasi untuk menjalankan dua perintah SQL.

```
aws redshift-data batch-execute-statement
--sqls "select current_user;" "select current_date;"
--cluster-id mycluster
--database dev
```

Untuk mengakses pernyataan dengan `cancel-statement`, `describe-statement`, `get-statement-result`, dan `get-statement-result-v2` dikirimkan oleh sesi peran IAM yang disempurnakan identitas, pengguna Pusat Identitas IAM dan peran IAM harus cocok dengan kredensial yang digunakan untuk menjalankan atau. `execute-statement` `batch-execute-statement` Misalnya, AWS CLI perintah berikut mendapatkan hasil dari pernyataan SQL.

```
aws redshift-data get-statement-result
--id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Untuk membuat daftar pernyataan, `workgroup-name` parameter `cluster-identifier` atau harus disediakan untuk memastikan bahwa pengguna Pusat Identitas IAM hanya memiliki akses ke aplikasi Pusat Identitas IAM Amazon Redshift tempat mereka ditugaskan. Misalnya, AWS CLI perintah berikut mencantumkan pernyataan untuk cluster tertentu.

```
aws redshift-data list-statements
--cluster-identifier mycluster
```

Anda juga dapat menjalankan operasi Data API yang mengakses objek database dalam kluster atau grup kerja menggunakan propagasi identitas terpercaya. Ini termasuk `list-databases`, `list-schemas`, `list-tables`, dan `describe-table` operasi.

Panggilan API yang dilakukan oleh pengguna IAM Identity Center dapat dilacak. AWS CloudTrail onBehalfOf bagian dari CloudTrail acara menunjukkan id pengguna IAM Identity Center dan ARN toko identitas. Contoh berikut menunjukkan cuplikan CloudTrail peristiwa yang menunjukkan onBehalfOf bagian dengan ID pengguna IAM Identity Center a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 dan ARN penyimpanan Identitas dari. arn:aws:identitystore::123456789012:identitystore/d-9067bc44d2

```
{
    "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"
}
```

Anda dapat menjalankan perintah SQL berikut untuk memeriksa kueri yang dikirimkan oleh pengguna IAM Identity Center. Dalam contoh ini, email yang terdaftar di Pusat Identitas adalah username@example.com.

```
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;
```

Memanggil API Data

Anda dapat memanggil Data API atau AWS CLI untuk menjalankan pernyataan SQL pada kluster atau grup kerja tanpa server. Operasi utama untuk menjalankan pernyataan SQL adalah [ExecuteStatement](#) dan [BatchExecuteStatement](#) di Referensi API Data Amazon Redshift. Data API mendukung bahasa pemrograman yang didukung oleh AWS SDK. Untuk informasi selengkapnya tentang ini, lihat [Alat untuk Dibangun AWS](#).

Untuk melihat contoh kode pemanggilan Data API, lihat [Memulai dengan Redshift Data API](#) di GitHub Repositori ini memiliki contoh penggunaan AWS Lambda untuk mengakses data Amazon Redshift dari Amazon EC2 AWS Glue Data Catalog, dan Amazon Runtime. SageMaker Contoh bahasa pemrograman termasuk Python, Go, Java, dan Javascript.

Anda dapat memanggil API Data menggunakan AWS CLI.

Contoh berikut menggunakan AWS CLI untuk memanggil Data API. Untuk menjalankan contoh, edit nilai parameter agar sesuai dengan lingkungan Anda. Dalam banyak contoh, a `cluster-identifier` disediakan untuk dijalankan melawan cluster. Saat Anda menjalankan workgroup tanpa server, Anda menyediakan sebagai gantinya. `workgroup-name` Contoh-contoh ini menunjukkan beberapa operasi Data API. Untuk informasi selengkapnya, lihat Referensi Perintah AWS CLI .

Perintah dalam contoh berikut telah dibagi dan diformat agar mudah dibaca. Tidak semua parameter dan respons ditampilkan dalam semua contoh. Untuk definisi API dari sintaks permintaan lengkap, parameter permintaan, sintaks respons, dan elemen respons, lihat [Referensi API Data Amazon Redshift](#).

Meneruskan pernyataan SQL ke gudang data Amazon Redshift

Contoh di halaman ini mencakup berbagai cara untuk meneruskan pernyataan SQL ke gudang data Anda

Jalankan pernyataan SQL

Untuk menjalankan pernyataan SQL, gunakan `aws redshift-data execute-statement` AWS CLI perintah.

AWS CLI Perintah berikut menjalankan pernyataan SQL terhadap cluster dan mengembalikan identifier untuk mengambil hasil. Contoh ini menggunakan metode AWS Secrets Manager otentikasi.

```
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
```

Berikut adalah contoh respons tersebut.

```
{
  "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"
}
```

AWS CLI Perintah berikut menjalankan pernyataan SQL terhadap cluster dan mengembalikan identifier untuk mengambil hasil. Contoh ini menggunakan metode otentikasi kredensial sementara.

```
aws redshift-data execute-statement
  --db-user myuser
  --cluster-identifier mycluster-test
  --database dev
  --sql "select * from stl_query limit 1"
```

Berikut adalah contoh respons tersebut.

```
{
  "ClusterIdentifier": "mycluster-test",
  "CreatedAt": 1598306924.632,
  "Database": "dev",
  "DbUser": "myuser",
  "Id": "d9b6c0c9-0747-4bf4-b142-e8883122f766"
}
```

AWS CLI Perintah berikut menjalankan pernyataan SQL terhadap workgroup tanpa server dan mengembalikan identifier untuk mengambil hasilnya. Contoh ini menggunakan metode otentikasi kredensial sementara.

```
aws redshift-data execute-statement
  --database dev
  --workgroup-name myworkgroup
  --sql "select 1;"
```

Berikut adalah contoh respons tersebut.

```
{
  "CreatedAt": "2022-02-11T06:25:28.748000+00:00",
  "Database": "dev",
  "DbUser": "IAMR:RoleName",
  "Id": "89dd91f5-2d43-43d3-8461-f33aa093c41e",
  "WorkgroupName": "myworkgroup"
}
```

AWS CLI Perintah berikut menjalankan pernyataan SQL terhadap cluster dan mengembalikan identifier untuk mengambil hasil. Contoh ini menggunakan metode AWS Secrets Manager otentikasi dan token idempotensi.

```
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
```

Berikut adalah contoh respons tersebut.

```
{
  "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"
}
```

Jalankan pernyataan SQL dengan parameter

Untuk menjalankan pernyataan SQL, gunakan `aws redshift-data execute-statement` AWS CLI perintah.

AWS CLI Perintah berikut menjalankan pernyataan SQL terhadap cluster dan mengembalikan identifier untuk mengambil hasil. Contoh ini menggunakan metode AWS Secrets Manager otentikasi. Teks SQL memiliki parameter `distance` bernama. Dalam hal ini, jarak yang digunakan dalam predikat adalah 5. Dalam pernyataan `SELECT`, parameter bernama untuk nama kolom hanya dapat digunakan dalam predikat. Nilai untuk parameter bernama untuk pernyataan SQL ditentukan dalam `parameters` opsi.

```
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
```

Berikut adalah contoh respons tersebut.

```
{
  "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"
}
```

Contoh berikut menggunakan `EVENT` tabel dari database sampel. Untuk informasi selengkapnya, lihat [tabel EVENT](#) di Panduan Pengembang Database Amazon Redshift.

Jika Anda belum memiliki `EVENT` tabel dalam database Anda, Anda dapat membuatnya menggunakan Data API sebagai berikut:

```
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)"
```

Perintah berikut menyisipkan satu baris ke dalam EVENT tabel.

```
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"}]"
```

Perintah berikut menyisipkan baris kedua ke dalam EVENT tabel. Contoh ini menunjukkan hal berikut:

- Parameter bernama id digunakan empat kali dalam teks SQL.
- Konversi tipe implisit diterapkan secara otomatis saat memasukkan parameter. starttime
- venueidKolom adalah tipe cast ke tipe data SMALLINT.
- String karakter yang mewakili tipe data DATE secara implisit diubah menjadi tipe data TIMESTAMP.
- Komentar dapat digunakan dalam teks SQL.

```
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\"}]"
```

Berikut ini menunjukkan dua baris yang disisipkan:

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

Perintah berikut menggunakan parameter bernama dalam klausa WHERE untuk mengambil baris di mana eventid adalah 1

```
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\"}]"
```

Jalankan perintah berikut untuk mendapatkan hasil SQL dari pernyataan SQL sebelumnya:

```
aws redshift-data get-statement-result --id 7529ad05-b905-4d71-9ec6-8b333836eb5a
```

Memberikan hasil sebagai berikut:

```
{
```

```
"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
}

```

Jalankan beberapa pernyataan SQL

Untuk menjalankan beberapa pernyataan SQL dengan satu perintah, gunakan `aws redshift-data batch-execute-statement` AWS CLI perintah.

AWS CLI Perintah berikut menjalankan tiga pernyataan SQL terhadap cluster dan mengembalikan identifier untuk mengambil hasil. Contoh ini menggunakan metode otentikasi kredensial sementara.

```

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"

```

Berikut adalah contoh respons tersebut.

```

{
  "ClusterIdentifier": "mycluster-test",
  "CreatedAt": 1598306924.632,

```

```
"Database": "dev",
"DbUser": "myuser",
"Id": "d9b6c0c9-0747-4bf4-b142-e8883122f766"
}
```

Daftar metadata tentang pernyataan SQL

Untuk daftar metadata tentang pernyataan SQL, gunakan perintah `aws redshift-data list-statements` AWS CLI Otorisasi untuk menjalankan perintah ini didasarkan pada izin IAM pemanggil.

AWS CLI Perintah berikut mencantumkan pernyataan SQL yang berjalan.

```
aws redshift-data list-statements
--status ALL
```

Berikut adalah contoh respons tersebut.

```
{
  "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  
  }  
]  
}
```

Jelaskan metadata tentang pernyataan SQL

Untuk mendapatkan deskripsi metadata untuk pernyataan SQL, gunakan perintah. `aws redshift-data describe-statement` AWS CLI Otorisasi untuk menjalankan perintah ini didasarkan pada izin IAM pemanggil.

AWS CLI Perintah berikut menjelaskan pernyataan SQL.

```
aws redshift-data describe-statement  
  --id d9b6c0c9-0747-4bf4-b142-e8883122f766
```

Berikut adalah contoh respons tersebut.

```
{  
  "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  
}
```

Berikut ini adalah contoh `describe-statement` respon setelah menjalankan `batch-execute-statement` perintah dengan beberapa pernyataan SQL.

```
{  
  "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
}

```

Ambil hasil dari pernyataan SQL

Untuk mengambil hasil dari pernyataan SQL yang berjalan, gunakan perintah `redshift-data get-statement-result` or `redshift-data get-statement-result-v2` AWS CLI. Hasil dari `get-statement-result` dalam format JSON. Hasil dari `get-statement-result-v2` dalam

format CSV. Anda dapat memberikan Id yang Anda terima sebagai tanggapan terhadap `execute-statement` atau `batch-execute-statement`. IdNilai untuk pernyataan SQL dijalankan oleh `batch-execute-statement` dapat diambil dalam hasil `describe-statement` dan diakhiran oleh titik dua dan nomor urut seperti. `b2906c76-fa6e-4cdf-8c5f-4de1ff9b7652:2` Jika Anda menjalankan beberapa pernyataan SQL dengan `batch-execute-statement`, setiap pernyataan SQL memiliki Id nilai seperti yang ditunjukkan pada. `describe-statement` Otorisasi untuk menjalankan perintah ini didasarkan pada izin IAM pemanggil.

Pernyataan berikut mengembalikan hasil dari pernyataan SQL dijalankan oleh `execute-statement` yang membiarkan `ResultFormat` default untuk `JSON`. Untuk mengambil hasilnya, hubungi `get-statement-result` operasi.

```
aws redshift-data get-statement-result
--id d9b6c0c9-0747-4bf4-b142-e8883122f766
```

Pernyataan berikut mengembalikan hasil dari pernyataan SQL kedua dijalankan oleh `batch-execute-statement`.

```
aws redshift-data get-statement-result
--id b2906c76-fa6e-4cdf-8c5f-4de1ff9b7652:2
```

Berikut ini adalah contoh dari respon untuk panggilan ke `get-statement-result` mana hasil SQL dikembalikan dalam format JSON dalam `Records` kunci respon.

```
{
  "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
}

```

Contoh berikut menunjukkan pernyataan SQL dijalankan oleh `execute-statement` untuk mengembalikan hasil sebagai JSON. Tabel `testingtable` memiliki tiga kolom integer (`col1`, `col2`, `col3`) dan ada tiga baris dengan nilai (1, 2, 3), (4, 5, 6), dan (7, 8, 9).

```

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"
}

```

Berikut ini adalah contoh dari respon untuk panggilan ke `get-statement-result` mana hasil SQL dikembalikan dalam format JSON dalam `Records` kunci respon.

```
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
}
```

Contoh berikut menunjukkan pernyataan SQL dijalankan oleh `execute-statement` untuk mengembalikan hasil sebagai CSV. Tabel `testingtable` memiliki tiga kolom integer (`col1`, `col2`, `col3`) dan ada tiga baris dengan nilai (1, 2, 3), (4, 5, 6), dan (7, 8, 9).

```
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"
}
```

Berikut ini adalah contoh respons terhadap panggilan ke `get-statement-result-v2` tempat hasil SQL dikembalikan dalam format CSV di `Records` kunci respons. Baris dipisahkan oleh carriage return dan newline (`\r\n`). Baris pertama yang dikembalikan `Records` adalah header kolom. Hasil yang dikembalikan dalam format CSV dikembalikan dalam 1 MB di mana setiap potongan dapat menyimpan sejumlah baris hingga 1MB.

```
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"
```

```
}
```

Jelaskan tabel

Untuk mendapatkan metadata yang menggambarkan tabel, gunakan perintah `aws redshift-data describe-table` AWS CLI

AWS CLI Perintah berikut menjalankan pernyataan SQL terhadap cluster dan mengembalikan metadata yang menggambarkan tabel. Contoh ini menggunakan metode AWS Secrets Manager otentikasi.

```
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
```

Berikut adalah contoh respons tersebut.

```
{
  "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"
    }
]
}

```

AWS CLI Perintah berikut menjalankan pernyataan SQL terhadap cluster yang menggambarkan tabel. Contoh ini menggunakan metode otentikasi kredensial sementara.

```

aws redshift-data describe-table
--db-user myuser
--cluster-identifier mycluster-test
--database dev
--schema information_schema
--table sql_features

```

Berikut adalah contoh respons tersebut.

```

{
  "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"
    }
]
}

```

Daftar database dalam sebuah cluster

Untuk membuat daftar database dalam sebuah cluster, gunakan `aws redshift-data list-databases` AWS CLI perintah.

AWS CLI Perintah berikut menjalankan pernyataan SQL terhadap cluster untuk daftar database. Contoh ini menggunakan metode AWS Secrets Manager otentikasi.

```
aws redshift-data list-databases
```

```
--secret-arn arn:aws:secretsmanager:us-west-2:123456789012:secret:myuser-secret-
hKgPWn
--cluster-identifier mycluster-test
--database dev
```

Berikut adalah contoh respons tersebut.

```
{
  "Databases": [
    "dev"
  ]
}
```

AWS CLI Perintah berikut menjalankan pernyataan SQL terhadap cluster untuk daftar database. Contoh ini menggunakan metode otentikasi kredensial sementara.

```
aws redshift-data list-databases
--db-user myuser
--cluster-identifier mycluster-test
--database dev
```

Berikut adalah contoh respons tersebut.

```
{
  "Databases": [
    "dev"
  ]
}
```

Daftar skema dalam database

Untuk membuat daftar skema dalam database, gunakan `aws redshift-data list-schemas` AWS CLI perintah.

AWS CLI Perintah berikut menjalankan pernyataan SQL terhadap cluster untuk daftar skema dalam database. Contoh ini menggunakan metode AWS Secrets Manager otentikasi.

```
aws redshift-data list-schemas
  --secret-arn arn:aws:secretsmanager:us-west-2:123456789012:secret:myuser-secret-
hKgPWn
  --cluster-identifier mycluster-test
  --database dev
```

Berikut adalah contoh respons tersebut.

```
{
  "Schemas": [
    "information_schema",
    "pg_catalog",
    "pg_internal",
    "public"
  ]
}
```

AWS CLI Perintah berikut menjalankan pernyataan SQL terhadap cluster untuk daftar skema dalam database. Contoh ini menggunakan metode otentikasi kredensial sementara.

```
aws redshift-data list-schemas
  --db-user mysuser
  --cluster-identifier mycluster-test
  --database dev
```

Berikut adalah contoh respons tersebut.

```
{
  "Schemas": [
    "information_schema",
    "pg_catalog",
    "pg_internal",
    "public"
  ]
}
```

Daftar tabel dalam database

Untuk membuat daftar tabel dalam database, gunakan `aws redshift-data list-tables` AWS CLI perintah.

AWS CLI Perintah berikut menjalankan pernyataan SQL terhadap cluster untuk daftar tabel dalam database. Contoh ini menggunakan metode AWS Secrets Manager otentikasi.

```
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
```

Berikut adalah contoh respons tersebut.

```
{
  "Tables": [
    {
      "name": "sql_features",
      "schema": "information_schema",
      "type": "SYSTEM TABLE"
    },
    {
      "name": "sql_implementation_info",
      "schema": "information_schema",
      "type": "SYSTEM TABLE"
    }
  ]
}
```

AWS CLI Perintah berikut menjalankan pernyataan SQL terhadap cluster untuk daftar tabel dalam database. Contoh ini menggunakan metode otentikasi kredensial sementara.

```
aws redshift-data list-tables
  --db-user myuser
  --cluster-identifier mycluster-test
  --database dev
  --schema information_schema
```

Berikut adalah contoh respons tersebut.

```
{
  "Tables": [
    {
      "name": "sql_features",
      "schema": "information_schema",
      "type": "SYSTEM TABLE"
    },
    {
      "name": "sql_implementation_info",
      "schema": "information_schema",
      "type": "SYSTEM TABLE"
    }
  ]
}
```

Memecahkan masalah untuk Amazon Redshift Data API

Gunakan bagian berikut, yang berjudul pesan kesalahan umum, untuk membantu memecahkan masalah yang Anda alami dengan API Data.

Topik

- [Paket untuk kueri terlalu besar](#)
- [Respons basis data melebihi batas ukuran](#)

Paket untuk kueri terlalu besar

Jika Anda melihat kesalahan yang menunjukkan bahwa paket untuk kueri terlalu besar, umumnya set hasil yang dikembalikan untuk baris terlalu besar. Batas ukuran API Data adalah 64 KB per baris dalam set hasil yang ditampilkan oleh basis data.

Untuk mengatasi masalah ini, pastikan setiap baris dalam set hasil berukuran 64 KB atau kurang.

Respons basis data melebihi batas ukuran

Jika Anda melihat kesalahan yang menunjukkan bahwa respons database telah melebihi batas ukuran, umumnya ukuran kumpulan hasil yang dikembalikan oleh database terlalu besar. Batas Data API adalah 500 MB dalam kumpulan hasil yang dikembalikan oleh database.

Untuk mengatasi masalah ini, pastikan panggilan ke Data API mengembalikan 500 MB data atau kurang. Jika Anda perlu mengembalikan lebih dari 500 MB, Anda dapat menjalankan beberapa panggilan pernyataan dengan LIMIT klausa dalam kueri Anda.

Menjadwalkan operasi Amazon Redshift Data API dengan Amazon EventBridge

Anda dapat membuat aturan yang cocok dengan peristiwa yang dipilih dan mengarahkannya ke target untuk mengambil tindakan. Anda juga dapat menggunakan aturan untuk mengambil tindakan pada jadwal yang telah ditentukan. Untuk informasi selengkapnya, lihat [Panduan EventBridge Pengguna Amazon](#).

Untuk menjadwalkan operasi Data API dengan EventBridge, peran IAM terkait harus mempercayai prinsipal untuk CloudWatch Acara (events.amazonaws.com). Peran ini harus setara dengan kebijakan terkelola yang AmazonEventBridgeFullAccess dilampirkan. Itu juga harus memiliki izin AmazonRedshiftDataFullAccess kebijakan yang dikelola oleh API Data. Anda dapat membuat peran IAM dengan izin ini di konsol IAM. Saat membuat peran di konsol IAM, pilih entitas terpercaya AWS layanan untuk CloudWatch Acara. Tentukan peran IAM dalam nilai RoleArn JSON di target. EventBridge Untuk informasi selengkapnya tentang membuat peran IAM, lihat [Membuat Peran untuk AWS Layanan \(Konsol\)](#) di Panduan Pengguna IAM.

Aturan name yang Anda buat di Amazon EventBridge harus cocok dengan StatementName diRedshiftDataParameters.

Contoh berikut menunjukkan variasi pembuatan EventBridge aturan dengan satu atau beberapa pernyataan SQL dan dengan cluster Amazon Redshift atau grup kerja Amazon Redshift Tanpa Server sebagai gudang data.

Memanggil dengan satu pernyataan SQL dan cluster

Contoh berikut menggunakan AWS CLI untuk membuat EventBridge aturan yang digunakan untuk menjalankan pernyataan SQL terhadap cluster Amazon Redshift.

```
aws events put-rule
--name test-redshift-cluster-data
--schedule-expression "rate(1 minute)"
```

Kemudian EventBridge target dibuat untuk berjalan pada jadwal yang ditentukan dalam aturan.


```
aws events put-targets
--cli-input-json file:///data.json
```

File input data.json adalah sebagai berikut. Kunci Sql JSON menunjukkan ada pernyataan SQL tunggal. Nilai Arn JSON berisi pengidentifikasi cluster. Nilai RoleArn JSON berisi peran IAM yang digunakan untuk menjalankan SQL seperti yang dijelaskan sebelumnya.

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

Memanggil dengan satu pernyataan SQL dan workgroup

Contoh berikut menggunakan AWS CLI untuk membuat EventBridge aturan yang digunakan untuk menjalankan pernyataan SQL terhadap workgroup Amazon Redshift Tanpa Server.

```
aws events put-rule
--name test-redshift-serverless-workgroup-data
--schedule-expression "rate(1 minute)"
```

Kemudian EventBridge target dibuat untuk berjalan pada jadwal yang ditentukan dalam aturan.

```
aws events put-targets
--cli-input-json file:///data.json
```

File input data.json adalah sebagai berikut. Kunci `Sql` JSON menunjukkan ada pernyataan SQL tunggal. Nilai `Arn` JSON berisi nama workgroup. Nilai `RoleArn` JSON berisi peran IAM yang digunakan untuk menjalankan SQL seperti yang dijelaskan sebelumnya.

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

Memanggil dengan beberapa pernyataan SQL dan cluster

Contoh berikut menggunakan AWS CLI untuk membuat EventBridge aturan yang digunakan untuk menjalankan beberapa pernyataan SQL terhadap cluster Amazon Redshift.

```
aws events put-rule
--name test-redshift-cluster-data
--schedule-expression "rate(1 minute)"
```

Kemudian EventBridge target dibuat untuk berjalan pada jadwal yang ditentukan dalam aturan.

```
aws events put-targets
--cli-input-json file://data.json
```

File input data.json adalah sebagai berikut. Kunci `Sqls` JSON menunjukkan ada beberapa pernyataan SQL. Nilai `Arn` JSON berisi pengidentifikasi cluster. Nilai `RoleArn` JSON berisi peran IAM yang digunakan untuk menjalankan SQL seperti yang dijelaskan sebelumnya.

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

Memanggil dengan beberapa pernyataan SQL dan workgroup

Contoh berikut menggunakan AWS CLI untuk membuat EventBridge aturan yang digunakan untuk menjalankan beberapa pernyataan SQL terhadap workgroup Amazon Redshift Tanpa Server.

```
aws events put-rule
--name test-redshift-serverless-workgroup-data
--schedule-expression "rate(1 minute)"
```

Kemudian EventBridge target dibuat untuk berjalan pada jadwal yang ditentukan dalam aturan.

```
aws events put-targets
--cli-input-json file://data.json
```

File input data.json adalah sebagai berikut. Kunci `Sqls` JSON menunjukkan ada beberapa pernyataan SQL. Nilai `Arn` JSON berisi nama workgroup. Nilai `RoleArn` JSON berisi peran IAM yang digunakan untuk menjalankan SQL seperti yang dijelaskan sebelumnya.

```
{
  "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
  }
}
```

Memantau API Data

Pemantauan adalah bagian penting untuk menjaga keandalan, ketersediaan, dan kinerja API Data dan AWS solusi Anda yang lain. AWS menyediakan alat pemantauan berikut untuk menonton API Data, melaporkan ketika ada sesuatu yang salah, dan mengambil tindakan otomatis bila perlu:

- Amazon EventBridge dapat digunakan untuk mengotomatiskan AWS layanan Anda dan merespons secara otomatis peristiwa sistem, seperti masalah ketersediaan aplikasi atau perubahan sumber daya. Acara dari AWS layanan dikirimkan ke EventBridge dalam waktu nyaris nyata. Anda dapat menuliskan aturan sederhana untuk menunjukkan peristiwa mana yang sesuai kepentingan Anda, dan tindakan otomatis mana yang diambil ketika suatu peristiwa sesuai dengan suatu aturan. Untuk informasi selengkapnya, lihat [Panduan EventBridge Pengguna Amazon](#).
- AWS CloudTrail menangkap panggilan API dan peristiwa terkait yang dibuat oleh atau atas nama AWS akun Anda dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Anda dapat mengidentifikasi pengguna dan akun mana yang dipanggil AWS, alamat IP sumber dari mana panggilan dilakukan, dan kapan panggilan terjadi. Untuk mempelajari selengkapnya tentang cara Amazon Redshift terintegrasi AWS CloudTrail, lihat [Logging](#) with CloudTrail. Untuk informasi selengkapnya CloudTrail, lihat [Panduan AWS CloudTrail Pengguna](#).

Topik

- [Memantau peristiwa untuk Amazon Redshift Data API di Amazon EventBridge](#)

Memantau peristiwa untuk Amazon Redshift Data API di Amazon EventBridge

Anda dapat memantau peristiwa API Data di EventBridge, yang mengirimkan aliran data real-time dari aplikasi, aplikasi software-as-a-service (SaaS), dan layanan Anda sendiri. AWS EventBridge merutekan data tersebut ke target seperti AWS Lambda dan Amazon SNS. Peristiwa ini sama dengan yang muncul di CloudWatch Peristiwa, yang memberikan aliran peristiwa sistem yang mendekati waktu nyata yang menggambarkan perubahan sumber daya. AWS Acara dikirim ke akun yang berisi database Amazon Redshift. Misalnya, jika Anda berperan di akun lain, acara akan dikirim ke akun tersebut. Untuk informasi selengkapnya, lihat [EventBridge Acara Amazon](#) di Panduan EventBridge Pengguna Amazon. .

Peristiwa API data dikirim saat operasi `ExecuteStatement` atau `BatchExecuteStatement` API menetapkan `WithEvent` opsi `kettrue`. `stateBidang` acara berisi salah satu nilai berikut:

- DIBATALKAN - Proses kueri dihentikan oleh pengguna.
- GAGAL — Query run gagal.
- SELESAI - Kueri telah selesai berjalan.

Acara disampaikan secara terjamin. Untuk informasi selengkapnya, lihat [Acara dari AWS layanan](#) di Panduan EventBridge Pengguna Amazon.

Contoh untuk acara selesai Data API

Contoh berikut menunjukkan peristiwa untuk Data API saat operasi `ExecuteStatement` API selesai. Dalam contoh ini, pernyataan bernama `test.testtable` selesai berjalan.

```
{
  "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
  }
}
```

Menggunakan AWS KMS dengan Amazon Redshift Data API

Saat Anda mengenkripsi kluster Amazon Redshift atau grup kerja Redshift Tanpa Server dengan kunci yang dikelola pelanggan, Amazon Redshift Data API menggunakan kunci yang dikelola pelanggan yang sama untuk menyimpan dan mengenkripsi kueri dan hasil Anda.

Data API mengenkripsi data Anda secara default untuk melindungi informasi sensitif, seperti teks kueri dan hasil kueri. Ini menggunakan kunci AWS KMS enkripsi yang dimiliki oleh AWS untuk perlindungan ini.

Enkripsi default untuk data saat istirahat mengurangi overhead operasional dan kompleksitas saat Anda melindungi data sensitif. Pendekatan ini membantu Anda membangun aplikasi aman yang memenuhi kepatuhan enkripsi yang ketat dan persyaratan peraturan.

Menggunakan hibah di AWS KMS

Data API memerlukan hibah untuk menggunakan kunci yang dikelola pelanggan Anda.

Saat Anda menelepon `ExecuteStatement` atau `BatchExecuteStatement` melawan kluster yang dienkripsi dengan kunci yang dikelola pelanggan, Amazon Redshift membuat hibah atas nama Anda dengan mengirimkan [CreateGrant](#) permintaan ke AWS KMS. AWS KMS menggunakan hibah untuk memberikan akses Data API ke kunci KMS di akun Anda.

Data API memerlukan hibah untuk menggunakan kunci terkelola pelanggan Anda untuk operasi berikut:

- Kirim [Encrypt](#) permintaan AWS KMS untuk mengenkripsi metadata kueri dengan kunci terkelola pelanggan Anda.
- Kirim [GenerateDataKey](#) permintaan AWS KMS untuk menghasilkan kunci data yang dienkripsi oleh kunci terkelola pelanggan Anda.
- Kirim [Decrypt](#) permintaan AWS KMS untuk mendekripsi kunci data terenkripsi sehingga mereka dapat mengenkripsi data Anda.

Anda dapat mencabut akses ke hibah atau menghapus akses Amazon Redshift ke kunci yang dikelola pelanggan kapan saja. Jika Anda melakukannya, API Data tidak dapat lagi mengakses data yang dienkripsi oleh kunci yang dikelola pelanggan Anda, yang memengaruhi operasi yang bergantung pada data tersebut. Misalnya, jika Anda mencoba mengambil hasil kueri atau melacak status kueri setelah mencabut hibah, API Data akan mengembalikan file. `AccessDeniedException`

Kebijakan utama untuk kunci terkelola pelanggan Anda

Kebijakan utama mengontrol akses ke kunci yang dikelola pelanggan Anda. Setiap kunci yang dikelola pelanggan harus memiliki persis satu kebijakan utama, yang berisi pernyataan yang menentukan siapa yang dapat menggunakan kunci dan bagaimana mereka dapat menggunakannya. Saat membuat kunci terkelola pelanggan, Anda dapat menentukan kebijakan kunci. Untuk informasi selengkapnya, lihat [Kunci terkelola pelanggan](#) di Panduan AWS Key Management Service Pengembang.

Untuk menggunakan kunci yang dikelola pelanggan dengan API Data, Anda harus terlebih dahulu mengizinkan akses ke Amazon Redshift. Operasi API berikut harus diizinkan dalam kebijakan utama:

- `kms:CreateGrant`— Menambahkan hibah ke kunci yang dikelola pelanggan. Memberikan akses kontrol ke AWS KMS kunci tertentu, yang memungkinkan akses untuk memberikan operasi yang diperlukan Amazon Redshift. Untuk informasi selengkapnya, lihat [Menggunakan hibah di AWS KMS](#).

Berikut ini adalah contoh kebijakan kunci:

```
"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":""
}
]

```

Konteks enkripsi API data

Konteks enkripsi adalah kumpulan opsional pasangan kunci-nilai yang berisi informasi kontekstual tambahan tentang data. AWS KMS menggunakan konteks enkripsi sebagai data otentikasi tambahan untuk mendukung enkripsi yang diautentikasi. Bila Anda menyertakan konteks enkripsi dalam permintaan untuk mengenkripsi data, AWS KMS mengikat konteks enkripsi ke data terenkripsi. Untuk mendekripsi data, Anda harus menyertakan konteks enkripsi yang sama dalam permintaan.

Data API menggunakan tiga pasangan nilai kunci konteks enkripsi yang sama di semua operasi AWS KMS kriptografi untuk kluster yang disediakan:

- `aws:redshift:arn`— Nama Sumber Daya Amazon (ARN) cluster

- `aws:redshift:createtime`— Stempel waktu saat Anda meminta pembuatan cluster
- `serviceName` – `RedshiftDataAPI`

```
"EncryptionContextSubset": {  
  "aws:redshift:arn": "arn:aws:redshift:us-east-1:123456789012:cluster:redshift-cluster",  
  "aws:redshift:createtime": "20250815T0000Z",  
  "serviceName": "RedshiftDataAPI",  
}
```

Data API menggunakan dua pasangan nilai kunci konteks enkripsi di semua operasi AWS KMS kriptografi untuk grup kerja tanpa server:

- `aws:redshift-serverless:arn`— Nama Sumber Daya Amazon (ARN) namespace
- `serviceName`— `RedshiftDataAPI`

```
"EncryptionContextSubset": {  
  "aws:redshift-serverless:arn": "arn:aws:redshift-serverless:us-east-1:123456789012:namespace:12345678-1234-1234-1234-123456789012",  
  "serviceName": "RedshiftDataAPI"  
}
```

Untuk informasi selengkapnya tentang enkripsi, lihat [Pengantar detail kriptografi. AWS KMS](#) Untuk informasi selengkapnya tentang Amazon Redshift dan AWS KMS integrasi, lihat [Cara Amazon Redshift menggunakan. AWS KMS](#)

Menggunakan Amazon SageMaker Unified Studio untuk menanyakan database Anda di Amazon Redshift dan lakehouse SageMaker

Amazon SageMaker Unified Studio menyediakan lingkungan pengembangan di luar konsol dan mendukung analisis SQL pada data di SageMaker lakehouse, Amazon Redshift, dan Amazon Athena untuk analitik SQL. Arahkan ke Amazon SageMaker Unified Studio menggunakan URL dari admin Anda dan gunakan SSO atau AWS kredensial Anda untuk masuk. Untuk informasi selengkapnya tentang menyiapkan project pertama Anda, lihat [Memulai](#) di Panduan Pengguna Amazon SageMaker Unified Studio.

[Di Amazon SageMaker Unified Studio, Anda dapat melakukan analisis SQL dengan menjalankan Amazon Redshift dan Amazon Athena dengan editor kueri.](#) Gunakan editor kueri untuk menulis dan menjalankan kueri, melihat hasil, dan berbagi pekerjaan Anda dengan tim Anda. Jalankan kueri terhadap gudang data Redshift di (dalam Akun AWS akun yang sama dan di akun Akun AWS lainnya), buat kueri SQL untuk Redshift dan Athena menggunakan antarmuka yang sama, dan jadwalkan kueri SQL menggunakan Alur Kerja Terkelola Amazon untuk Apache Airflow. Anda juga dapat menggunakan SQL generatif Amazon Q untuk menghasilkan SQL dari bahasa alami.

Grup parameter Amazon Redshift

Di Amazon Redshift, Anda mengaitkan grup parameter dengan setiap cluster yang Anda buat. Grup parameter adalah sekelompok parameter yang berlaku untuk semua database yang Anda buat di cluster. Parameter ini mengkonfigurasi pengaturan database seperti batas waktu kueri dan gaya tanggal. Saat Anda meluncurkan cluster, Anda harus mengaitkannya dengan grup parameter. Jika Anda ingin mengubah grup parameter nanti, Anda dapat memodifikasi cluster dan memilih grup parameter yang berbeda.

Setiap grup parameter memiliki beberapa parameter untuk mengkonfigurasi pengaturan untuk database. Daftar parameter yang tersedia tergantung pada keluarga kelompok parameter tempat grup parameter berada. Keluarga grup parameter default adalah `redshift-2.0`.

Amazon Redshift menyediakan satu grup parameter default untuk setiap keluarga grup parameter. Grup parameter default memiliki nilai preset untuk masing-masing parameternya, dan tidak dapat dimodifikasi. Format nama grup parameter default adalah `default.parameter_group_family`. Misalnya, grup parameter default untuk keluarga grup `redshift-2.0` parameter adalah `default.redshift-2.0`.

Jika Anda ingin menggunakan nilai parameter yang berbeda dari grup parameter default, Anda harus membuat grup parameter khusus dan kemudian mengaitkan cluster Anda dengannya. Awalnya, nilai parameter dalam grup parameter kustom sama seperti pada grup parameter default. Awal `source` untuk semua parameter adalah `engine-default` karena nilainya telah ditetapkan oleh Amazon Redshift. Setelah Anda mengubah nilai parameter, `source` perubahan user untuk menunjukkan bahwa nilai telah dimodifikasi dari nilai defaultnya.

Note

Konsol Amazon Redshift tidak menampilkan setiap `source` parameter. Anda harus menggunakan Amazon Redshift API, the AWS CLI, atau salah satu AWS SDKs untuk melihat `source`.

Untuk grup parameter yang Anda buat, Anda dapat mengubah nilai parameter kapan saja, atau Anda dapat mengatur ulang semua nilai parameter ke defaultnya. Anda juga dapat mengaitkan grup parameter yang berbeda dengan cluster. Dalam beberapa kasus, Anda dapat mengubah nilai parameter dalam grup parameter yang sudah dikaitkan dengan cluster atau mengaitkan grup parameter yang berbeda dengan cluster. Dalam kasus ini, Anda mungkin perlu me-restart cluster.

agar nilai parameter yang diperbarui diterapkan. Jika klaster gagal dan dimulai ulang oleh Amazon Redshift, perubahan Anda akan diterapkan pada saat itu. Perubahan tidak diterapkan jika klaster Anda dimulai ulang selama pemeliharaan. Untuk informasi selengkapnya, lihat [Sifat dinamis dan statis WLM](#).

Nilai parameter default

Note

Pada 10 Januari 2025, nilai default untuk `require_ssl` parameter adalah benar. Jika Anda tidak ingin klaster Anda memerlukan SSL, Anda dapat menggunakan grup parameter kustom saat membuat cluster, atau memodifikasi cluster untuk mengaitkannya dengan grup parameter kustom setelah membuat cluster dengan default.

Tabel berikut menunjukkan nilai parameter default sekilas dengan link ke informasi yang lebih mendalam tentang setiap parameter. Ini adalah nilai default untuk keluarga grup `redshift-2.0` parameter.

Nama parameter	Nilai	Informasi selengkapnya
<code>auto_analisis</code>	true	auto_analysis di Panduan Pengembang Basis Data Amazon Redshift
<code>auto_mv</code>	true	Tampilan terwujud otomatis dalam Panduan Pengembang Basis Data Amazon Redshift
<code>datestyle</code>	ISO, MDY	datestyle di Panduan Pengembang Database Amazon Redshift
<code>enable_case_sensitive_identifier</code>	false	enable_case_sensitive_identifier di Panduan Pengembang Basis Data Amazon Redshift
<code>enable_user_activity_logging</code>	false	Pencatatan audit basis data dalam panduan ini
<code>extra_float_digits</code>	0	extra_float_digits di Panduan Pengembang Basis Data Amazon Redshift

Nama parameter	Nilai	Informasi selengkapnya
max_concurrency_scaling_clusters	1	max_concurrency_scaling_clusters di Panduan Pengembang Basis Data Amazon Redshift
query_group	default	query_group di Panduan Pengembang Basis Data Amazon Redshift
require_ssl	true	Mengkonfigurasi opsi keamanan untuk koneksi dalam panduan ini
search_path	\$ pengguna, publik	search_path di Panduan Pengembang Basis Data Amazon Redshift
statement_timeout	0	statement_timeout di Panduan Pengembang Basis Data Amazon Redshift
wlm_json_configuration	[{"auto_wlm":true}]	Manajemen beban kerja dalam panduan ini
gunakan_fips_ssl	false	Aktifkan mode SSL yang sesuai dengan FIPS hanya jika sistem Anda harus sesuai dengan FIPS.

Note

max_cursor_result_set_sizeParameter tidak digunakan lagi. Untuk informasi selengkapnya tentang ukuran set hasil kursor, lihat Batasan [kursor di Panduan Pengembang Database](#) Amazon Redshift.

Anda dapat mengganti sementara parameter dengan menggunakan SET perintah dalam database. SETPerintah mengesampingkan parameter untuk durasi sesi Anda saat ini saja. Selain parameter yang tercantum dalam tabel sebelumnya, Anda juga dapat menyesuaikan jumlah slot sementara dengan mengatur wlm_query_slot_count di database. wlm_query_slot_countParameter tidak tersedia untuk konfigurasi dalam grup parameter. Untuk informasi selengkapnya tentang menyesuaikan jumlah slot, lihat [wlm_query_slot_count](#) di Panduan Pengembang Database Amazon

Redshift. Untuk informasi selengkapnya tentang mengganti sementara parameter lain, lihat [Memodifikasi konfigurasi server di Panduan Pengembang](#) Database Amazon Redshift.

Manajemen beban kerja

Di Amazon Redshift, Anda menggunakan manajemen beban kerja (WLM) untuk menentukan jumlah antrian kueri yang tersedia, dan cara kueri dirutekan ke antrian tersebut untuk diproses. WLM adalah bagian dari konfigurasi grup parameter. Sebuah cluster menggunakan konfigurasi WLM yang ditentukan dalam kelompok parameter terkait.

Saat Anda membuat grup parameter, konfigurasi WLM default berisi satu antrian yang dapat menjalankan hingga lima kueri secara bersamaan. Anda dapat menambahkan antrian tambahan dan mengkonfigurasi properti WLM di masing-masing jika Anda ingin lebih banyak kontrol atas pemrosesan kueri. Setiap antrian yang Anda tambahkan memiliki konfigurasi WLM default yang sama hingga Anda mengonfigurasi propertinya.

Saat Anda menambahkan antrian tambahan, antrian terakhir dalam konfigurasi adalah antrian default. Kecuali kueri dirutekan ke antrian lain berdasarkan kriteria dalam konfigurasi WLM, itu diproses oleh antrian default. Anda dapat menentukan mode dan tingkat konkurensi (slot kueri) untuk antrian default, tetapi Anda tidak dapat menentukan grup pengguna atau grup kueri untuk antrian default.

Seperti parameter lainnya, Anda tidak dapat memodifikasi konfigurasi WLM di grup parameter default. Cluster yang terkait dengan grup parameter default selalu menggunakan konfigurasi WLM default. Untuk memodifikasi konfigurasi WLM, buat grup parameter baru dan kemudian kaitkan grup parameter itu dengan cluster apa pun yang memerlukan konfigurasi WLM kustom Anda.

Sifat dinamis dan statis WLM

Properti konfigurasi WLM bersifat dinamis atau statis. Anda dapat menerapkan properti dinamis ke database tanpa reboot cluster, tetapi properti statis memerlukan reboot cluster agar perubahan diterapkan. Untuk informasi selengkapnya tentang properti statis dan dinamis, lihat properti [konfigurasi dinamis dan statis WLM](#).

Properti untuk parameter konfigurasi WLM

Anda dapat mengonfigurasi WLM dengan menggunakan konsol Amazon Redshift, AWS CLI Amazon Redshift API, atau salah satunya. AWS SDKs Konfigurasi WLM menggunakan beberapa properti

untuk menentukan perilaku antrian, seperti alokasi memori di seluruh antrian, jumlah kueri yang dapat berjalan secara bersamaan dalam antrian, dan sebagainya.

 Note

Properti berikut muncul dengan nama konsol Amazon Redshift mereka, dengan nama properti JSON yang sesuai dalam deskripsi.

Tabel berikut merangkum apakah properti berlaku untuk WLM otomatis atau WLM manual.

Properti WLM	WLM otomatis	Panduan WLM
WLM otomatis	Ya	Ya
Aktifkan akselerasi kueri singkat	Ya	Ya
Waktu berjalan maksimum untuk kueri singkat	Ya	Ya
Prioritas	Ya	Tidak
Jenis antrian	Ya	Ya
Nama antrian	Ya	Ya
Mode Penskalaan Konkurensi	Ya	Ya
Bersamaan	Tidak	Ya
Grup pengguna	Ya	Ya
Wildcard grup pengguna	Ya	Ya
Grup kueri	Ya	Ya
Wildcard grup kueri	Ya	Ya
Peran pengguna	Ya	Ya

Properti WLM	WLM otomatis	Panduan WLM
Wildcard peran pengguna	Ya	Ya
Waktu habis	Tidak	Usang
Memori	Tidak	Ya
Aturan Pemantauan Kueri	Ya	Ya

Daftar berikut menjelaskan properti WLM yang dapat Anda konfigurasi.

WLM otomatis

Auto WLM diatur untuk `true` mengaktifkan WLM otomatis. WLM otomatis menetapkan nilai untuk Concurrency pada main dan Memory (%) ke. Auto Amazon Redshift mengelola konkurensi kueri dan alokasi memori. Nilai default-nya `true`.

Properti JSON: `auto_wlm`

Aktifkan akselerasi kueri singkat

Akselerasi kueri singkat (SQA) memprioritaskan kueri jangka pendek yang dipilih sebelum kueri yang berjalan lebih lama. SQA mengeksekusi kueri jangka pendek di ruang khusus, sehingga kueri SQA tidak dipaksa untuk menunggu dalam antrian di belakang kueri yang lebih panjang. Dengan SQA, kueri jangka pendek mulai dijalankan lebih cepat dan pengguna melihat hasilnya lebih cepat. Ketika Anda mengaktifkan SQA, Anda juga dapat menentukan waktu berjalan maksimum untuk kueri singkat. Untuk mengaktifkan SQA, tentukan `true`. Nilai default-nya `false`. Pengaturan ini diterapkan untuk setiap kelompok parameter daripada antrian.

Properti JSON: `short_query_queue`

Waktu berjalan maksimum untuk kueri singkat

Saat Anda mengaktifkan SQA, Anda dapat menentukan 0 untuk membiarkan WLM secara dinamis mengatur waktu berjalan maksimum untuk kueri singkat. Atau, Anda dapat menentukan nilai 1-20 detik, dalam milidetik. Nilai default-nya adalah 0.

Properti JSON: `max_execution_time`

Prioritas

Prioritas menetapkan prioritas kueri yang berjalan dalam antrian. Untuk mengatur prioritas, mode WLM harus diatur ke Auto WLM; artinya, harus. `auto_wlm true` Nilai prioritas dapat berupa `highest`, `high`, `normal`, `low`, dan `lowest`. Nilai default-nya `normal`.

Properti JSON: `priority`

Jenis antrian

Jenis antrian menunjuk antrian seperti yang digunakan baik oleh Auto WLM atau Manual WLM. Setel `queue_type` ke salah satu `auto` atau `manual`. Jika tidak ditentukan, default-nya adalah `manual`.

Properti JSON: `queue_type`

Nama antrian

Nama antrian. Anda dapat mengatur nama antrian berdasarkan kebutuhan bisnis Anda. Nama antrian harus unik dalam konfigurasi WLM, hingga 64 karakter alfanumerik, garis bawah atau spasi, dan tidak dapat berisi tanda kutip. Misalnya, jika Anda memiliki antrian untuk kueri ETL Anda, Anda dapat menamainya. `ETL_queue` Nama ini digunakan dalam metrik, nilai tabel sistem, dan konsol Amazon Redshift untuk mengidentifikasi antrian. Kueri dan laporan yang menggunakan nama dari sumber-sumber ini harus dapat menangani perubahan nama. Sebelumnya, nama antrian dihasilkan oleh Amazon Redshift. Nama default antrian adalah `Queue 1`, `Queue 2`, ke antrian terakhir bernama. `Default_queue`

Important

Jika Anda mengubah nama antrian, nilai `QueueName` dimensi metrik antrian WLM (seperti, `WLMQueue Panjang`, `WLMQuery Durasi`, `WLMRunning Kueri WLMQueue WaitTime` `WLMQueriesCompletedPerSecond`, dan sebagainya) juga berubah. Jadi, jika Anda mengubah nama antrian, Anda mungkin perlu mengubah CloudWatch alarm yang telah Anda atur.

Properti JSON: `name`

Mode Penskalaan Konkurensi

Untuk mengaktifkan penskalaan konkurensi pada antrian, setel mode Penskalaan Konkurensi ke. `auto` Ketika jumlah kueri yang dirutekan ke antrian melebihi konkurensi antrian yang

dikonfigurasi, kueri yang memenuhi syarat akan masuk ke klaster penskalaan. Ketika slot tersedia, kueri berjalan di cluster utama. Nilai default-nya off.

Properti JSON: `concurrency_scaling`

Konkurensi

Jumlah kueri yang dapat berjalan secara bersamaan dalam antrian WLM manual. Properti ini hanya berlaku untuk WLM manual. Jika penskalaan konkurensi diaktifkan, kueri yang memenuhi syarat akan masuk ke klaster penskalaan saat antrian mencapai tingkat konkurensi (slot kueri). Jika penskalaan konkurensi tidak diaktifkan, kueri menunggu dalam antrian sampai slot tersedia. Kisarannya antara 1 dan 50.

Properti JSON: `query_concurrency`

Grup Pengguna

Daftar nama grup pengguna yang dipisahkan koma. Ketika anggota grup pengguna menjalankan kueri dalam database, kueri mereka dirutekan ke antrian yang terkait dengan grup pengguna mereka.

Properti JSON: `user_group`

Grup Pengguna Wildcard

Nilai Boolean yang menunjukkan apakah akan mengaktifkan wildcard untuk grup pengguna. Jika ini 0, wildcard dinonaktifkan; jika ini adalah 1, wildcard diaktifkan. Ketika wildcard diaktifkan, Anda dapat menggunakan "*" atau "?" untuk menentukan beberapa grup pengguna saat menjalankan kueri. Untuk informasi selengkapnya, lihat [Wildcard](#).

Properti JSON: `user_group_wild_card`

Grup Kueri

Daftar grup kueri yang dipisahkan koma. Saat anggota grup kueri menjalankan kueri dalam database, kueri mereka dirutekan ke antrian yang terkait dengan grup kueri mereka.

Properti JSON: `query_group`

Wildcard Grup Kueri

Nilai Boolean yang menunjukkan apakah akan mengaktifkan wildcard untuk grup kueri. Jika ini 0, wildcard dinonaktifkan; jika ini adalah 1, wildcard diaktifkan. Ketika wildcard diaktifkan, Anda dapat menggunakan "*" atau "?" untuk menentukan beberapa grup kueri saat menjalankan kueri. Untuk informasi selengkapnya, lihat [Wildcard](#).

Properti JSON: `query_group_wild_card`

Peran Pengguna

Daftar peran pengguna yang dipisahkan koma. Ketika anggota dengan peran pengguna tersebut menjalankan kueri dalam database, kueri mereka dirutekan ke antrian yang terkait dengan peran pengguna mereka. Untuk informasi selengkapnya tentang peran pengguna, lihat [Kontrol akses berbasis peran \(RBAC\)](#).

Properti JSON: `user_role`

Peran Pengguna Wildcard

Nilai Boolean yang menunjukkan apakah akan mengaktifkan wildcard untuk grup kueri. Jika ini 0, wildcard dinonaktifkan; jika ini adalah 1, wildcard diaktifkan. Ketika wildcard diaktifkan, Anda dapat menggunakan "*" atau "?" untuk menentukan beberapa grup kueri saat menjalankan kueri. Untuk informasi selengkapnya, lihat [Wildcard](#).

Properti JSON: `user_role_wild_card`

Batas waktu (ms)

WLM timeout (`max_execution_time`) tidak digunakan lagi. Ini tidak tersedia saat menggunakan WLM otomatis. Sebagai gantinya, buat aturan pemantauan kueri (QMR) menggunakan `query_execution_time` untuk membatasi waktu eksekusi yang telah berlalu untuk kueri. Untuk informasi selengkapnya, lihat aturan [pemantauan kueri WLM](#).

Waktu maksimum, dalam milidetik, kueri dapat berjalan sebelum dibatalkan. Dalam beberapa kasus, kueri hanya-baca, seperti pernyataan SELECT, mungkin dibatalkan karena batas waktu WLM. Dalam kasus ini, WLM mencoba merutekan kueri ke antrian pencocokan berikutnya berdasarkan aturan penetapan antrian WLM. Jika kueri tidak cocok dengan definisi antrian lainnya, kueri dibatalkan; kueri tidak ditetapkan ke antrian default. Untuk informasi selengkapnya, lihat lompatan [antrian kueri WLM](#). Batas waktu WLM tidak berlaku untuk kueri yang telah mencapai status. `returning` Untuk melihat status kueri, lihat tabel sistem [STV_WLM_QUERY_STATE](#).

Properti JSON: `max_execution_time`

Memori (%)

Persentase memori untuk dialokasikan ke antrian. Jika Anda menentukan persentase memori untuk setidaknya satu antrian, Anda harus menentukan persentase untuk semua antrian lainnya, hingga total 100 persen. Jika alokasi memori Anda di bawah 100 persen di semua antrian, memori

yang tidak terisi dikelola oleh layanan. Layanan sementara dapat memberikan memori yang tidak terisi ini ke antrian yang meminta memori tambahan untuk diproses.

Properti JSON: `memory_percent_to_use`

Aturan Pemantauan Kueri

Anda dapat menggunakan aturan pemantauan kueri WLM untuk terus memantau antrian WLM Anda untuk kueri berdasarkan kriteria, atau predikat, yang Anda tentukan. Misalnya, Anda dapat memantau kueri yang cenderung menggunakan sumber daya sistem yang berlebihan, dan kemudian memulai tindakan tertentu ketika kueri melebihi batas kinerja yang ditentukan.

Note

Jika Anda memilih untuk membuat aturan secara terprogram, kami sangat menyarankan menggunakan konsol untuk menghasilkan JSON yang Anda sertakan dalam definisi grup parameter.

Anda mengaitkan aturan pemantauan kueri dengan antrian kueri tertentu. Anda dapat memiliki hingga 25 aturan per antrian, dan batas total untuk semua antrian adalah 25 aturan.

Properti JSON: `rules`

Hirarki properti JSON:

```
rules
  rule_name
  predicate
    metric_name
    operator
    value
  action
    value
```

Untuk setiap aturan, Anda menentukan properti berikut:

- `rule_name`— Nama aturan harus unik dalam konfigurasi WLM. Nama aturan dapat mencapai 32 karakter alfanumerik atau garis bawah, dan tidak dapat berisi spasi atau tanda kutip.
- `predicate`— Anda dapat memiliki hingga tiga predikat per aturan. Untuk setiap predikat, tentukan properti berikut.

- **metric_name**— Untuk daftar metrik, lihat Metrik [pemantauan kueri di Panduan Pengembang Database Amazon Redshift](#).
- **operator** Operasi adalah =, <, dan >
- **value**— Nilai ambang batas untuk metrik tertentu yang memicu tindakan.
- **action** Setiap aturan dikaitkan dengan satu tindakan. Tindakan yang valid adalah:
 - log
 - hop (hanya tersedia dengan WLM manual)
 - abort
 - change_query_priority (hanya tersedia dengan WLM otomatis)

Contoh berikut menunjukkan JSON untuk aturan pemantauan kueri WLM bernama `rule_1`, dengan dua predikat dan tindakan. hop

```
"rules": [  
  {  
    "rule_name": "rule_1",  
    "predicate": [  
      {  
        "metric_name": "query_execution_time",  
        "operator": ">",  
        "value": 100000  
      },  
      {  
        "metric_name": "query_blocks_read",  
        "operator": ">",  
        "value": 1000  
      }  
    ],  
    "action": "hop"  
  }  
]
```

Untuk informasi selengkapnya tentang masing-masing properti dan strategi untuk mengonfigurasi antrian kueri, lihat [Menerapkan manajemen beban kerja](#) di Panduan Pengembang Database Amazon Redshift.

Mengkonfigurasi parameter WLM menggunakan AWS CLI

Untuk mengkonfigurasi WLM, Anda memodifikasi parameter. `wlm_json_configuration` Ukuran maksimum nilai `wlm_json_configuration` properti adalah 8000 karakter. Nilai diformat dalam JavaScript Object Notation (JSON). Jika Anda mengonfigurasi WLM dengan menggunakan AWS CLI, Amazon Redshift API, atau salah AWS SDKs satunya, gunakan bagian lainnya untuk mempelajari cara membuat struktur JSON untuk parameter. `wlm_json_configuration`

Note

Jika Anda mengonfigurasi WLM dengan menggunakan konsol Amazon Redshift, Anda tidak perlu memahami pemformatan JSON karena konsol menyediakan cara mudah untuk menambahkan antrian dan mengonfigurasi propertinya. Untuk informasi selengkapnya tentang mengonfigurasi WLM menggunakan konsol, lihat. [Mengubah grup parameter](#)

Contoh

Contoh berikut adalah konfigurasi WLM default, yang mendefinisikan satu antrian dengan WLM otomatis.

```
{
  "auto_wlm": true
}
```

Contoh

Contoh berikut adalah konfigurasi WLM kustom, yang mendefinisikan satu antrian WLM manual dengan tingkat konkurensi (slot kueri) lima.

```
{
  "query_concurrency":5
}
```

Sintaksis

Konfigurasi WLM default sangat sederhana, dengan hanya antrian dan satu properti. Anda dapat menambahkan lebih banyak antrian dan mengkonfigurasi beberapa properti untuk setiap

antrian dalam struktur JSON. Sintaks berikut mewakili struktur JSON yang Anda gunakan untuk mengkonfigurasi beberapa antrian dengan beberapa properti:

```
[
  {
    "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",
          ...
        }
      ]"
  }
]
```

Dalam contoh sebelumnya, properti representatif yang dimulai dengan q1 adalah objek dalam array untuk antrian pertama. Masing-masing objek ini adalah name/value pasangan; name dan value bersama-sama mengatur properti WLM untuk antrian pertama. Properti representatif yang dimulai dengan q2 adalah objek dalam array untuk antrian kedua. Jika Anda memerlukan lebih banyak antrian, Anda menambahkan array lain untuk setiap antrian tambahan dan mengatur properti untuk setiap objek.

Ketika Anda memodifikasi konfigurasi WLM, Anda harus menyertakan seluruh struktur untuk antrian Anda, bahkan jika Anda hanya ingin mengubah satu properti dalam antrian. Ini karena seluruh struktur JSON diteruskan sebagai string sebagai nilai untuk `wlm_json_configuration` parameter.

Memformat perintah AWS CLI

`wlm_json_configurationParameter` memerlukan format tertentu saat Anda menggunakan file AWS CLI. Format yang Anda gunakan tergantung pada sistem operasi klien Anda. Sistem operasi memiliki cara yang berbeda untuk melampirkan struktur JSON sehingga diteruskan dengan benar dari baris perintah. Untuk detail tentang cara membuat perintah yang sesuai di sistem operasi Linux, Mac OS X, dan Windows, lihat bagian berikut. Untuk informasi selengkapnya tentang perbedaan

dalam melampirkan struktur data JSON secara umum, lihat [Mengutip string](#) di Panduan Pengguna. AWS CLI AWS Command Line Interface

Contoh

Contoh perintah berikut mengkonfigurasi WLM manual untuk kelompok parameter yang disebut. `example-parameter-group` Konfigurasi ini memungkinkan akselerasi kueri pendek dengan waktu berjalan maksimum untuk kueri pendek yang disetel ke 0, yang menginstruksikan WLM untuk mengatur nilai secara dinamis. `ApplyTypePengaturannya` adalah `dynamic`. Pengaturan ini berarti bahwa setiap perubahan yang dibuat pada properti dinamis dalam parameter diterapkan segera kecuali perubahan statis lainnya telah dilakukan pada konfigurasi. Konfigurasi mendefinisikan tiga antrian dengan yang berikut:

- Antrian pertama memungkinkan pengguna untuk menentukan `report` sebagai label (seperti yang ditentukan dalam `query_group` properti) dalam kueri mereka untuk membantu dalam merutekan kueri ke antrian tersebut. Pencarian wildcard diaktifkan untuk `report*` label, sehingga label tidak perlu tepat untuk kueri yang akan diarahkan ke antrian. Misalnya, `reports` dan `reporting` keduanya cocok dengan grup kueri ini. Antrian dialokasikan 25 persen dari total memori di semua antrian, dan dapat menjalankan hingga empat kueri pada saat yang bersamaan. Kueri dibatasi hingga waktu maksimum 20000 milidetik (ms). mode diatur ke `auto`, jadi ketika slot kueri antrian penuh kueri yang memenuhi syarat dikirim ke cluster penskalaan.
- Antrian kedua memungkinkan pengguna yang merupakan anggota `admin` atau `dba` grup dalam database untuk mengarahkan kueri mereka ke antrian untuk diproses. Pencarian wildcard dinonaktifkan untuk grup pengguna, sehingga pengguna harus dicocokkan persis dengan grup dalam database agar kueri mereka dialihkan ke antrian. Antrian dialokasikan 40 persen dari total memori di semua antrian, dan dapat menjalankan hingga lima kueri pada saat yang sama. mode diatur ke `off`, sehingga semua kueri yang dikirim oleh anggota `admin` atau grup `dba` berjalan di cluster utama.
- Antrian terakhir dalam konfigurasi adalah antrian default. Antrian ini dialokasikan 35 persen dari total memori di semua antrian, dan dapat memproses hingga lima kueri sekaligus. Mode diatur ke `auto`.

Note

Contoh ditampilkan pada beberapa baris untuk tujuan demonstrasi. Perintah yang sebenarnya seharusnya tidak memiliki jeda baris.


```
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 }
]
```

Berikut ini adalah contoh konfigurasi aturan pemantauan kueri WLM untuk konfigurasi WLM otomatis. Contoh membuat grup parameter bernama `example-monitoring-rules`. Konfigurasi mendefinisikan tiga antrian yang sama seperti contoh sebelumnya, tetapi `query_concurrency` dan tidak `memory_percent_to_use` ditentukan lagi. Konfigurasi juga menambahkan aturan dan prioritas kueri berikut:

- Antrian pertama mendefinisikan aturan bernama `rule_1`. Aturan ini memiliki dua predikat: `query_cpu_time > 10000000` dan `query_blocks_read > 1000`. Tindakan aturannya adalah `log`. Prioritas antrian ini adalah `Normal`.
- Antrian kedua mendefinisikan aturan bernama `rule_2`. Aturan ini memiliki dua predikat: `query_execution_time > 600000000` dan `scan_row_count > 1000000000`. Tindakan aturannya adalah `abort`. Prioritas antrian ini adalah `Highest`.
- Antrian terakhir dalam konfigurasi adalah antrian default. Prioritas antrian ini adalah `Low`.

Note

Contoh ditampilkan pada beberapa baris untuk tujuan demonstrasi. Perintah yang sebenarnya seharusnya tidak memiliki jeda baris.

```
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": 1000000 },
        { "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

```

```
} ]'
```

Mengkonfigurasi WLM dengan menggunakan AWS CLI di baris perintah dengan file JSON

Anda dapat memodifikasi `wlm_json_configuration` parameter menggunakan AWS CLI dan meneruskan nilai `parameters` argumen sebagai file JSON.

```
aws redshift modify-cluster-parameter-group --parameter-group-name
myclusterparaametergroup --parameters file://modify_pg.json
```

Argumen untuk `--parameters` disimpan dalam file `modify_pg.json`. Lokasi file ditentukan dalam format untuk sistem operasi Anda. Untuk informasi selengkapnya, lihat [Memuat parameter dari file](#). Berikut ini menunjukkan contoh isi dari file `modify_pg.json` JSON.

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

Aturan untuk mengkonfigurasi WLM dengan menggunakan baris perintah AWS CLI pada sistem operasi Linux dan macOS X

Ikuti aturan ini untuk menjalankan AWS CLI perintah dengan parameter pada satu baris:

- Seluruh struktur JSON harus dilampirkan dalam tanda kutip tunggal (') dan satu set tanda kurung ([]).
- Semua nama parameter dan nilai parameter harus diapit tanda kutip ganda (").
- Dalam ParameterValue nilai, Anda harus melampirkan seluruh struktur bersarang dalam tanda kutip ganda (") dan tanda kurung ([]).
- Dalam struktur bersarang, masing-masing properti dan nilai untuk setiap antrian harus diapit kurung kurawal ({}).
- Dalam struktur bersarang, Anda harus menggunakan karakter escape backslash (\) sebelum setiap tanda kutip ganda (").
- Untuk name/value pasangan, titik dua (:) memisahkan setiap properti dari nilainya.
- Setiap name/value pasangan dipisahkan dari yang lain dengan koma (,).
- Beberapa antrian dipisahkan oleh koma (,) antara akhir kurung kurawal antrian (}) dan awal kurung kurawal antrian berikutnya ({}).

Aturan untuk mengkonfigurasi WLM dengan menggunakan di PowerShell Windows AWS CLI pada sistem operasi Microsoft Windows

Ikuti aturan ini untuk menjalankan AWS CLI perintah dengan parameter pada satu baris:

- Seluruh struktur JSON harus dilampirkan dalam tanda kutip tunggal (') dan satu set tanda kurung ([]).
- Semua nama parameter dan nilai parameter harus diapit tanda kutip ganda (").
- Dalam ParameterValue nilai, Anda harus melampirkan seluruh struktur bersarang dalam tanda kutip ganda (") dan tanda kurung ([]).
- Dalam struktur bersarang, masing-masing properti dan nilai untuk setiap antrian harus diapit kurung kurawal ({}).
- Dalam struktur bersarang, Anda harus menggunakan karakter escape backslash (\) sebelum setiap tanda kutip ganda (") dan karakter escape backslash (\). Persyaratan ini berarti bahwa Anda akan menggunakan tiga garis miring terbalik dan tanda kutip ganda untuk memastikan bahwa properti diteruskan dengan benar (\\").
- Untuk name/value pasangan, titik dua (:) memisahkan setiap properti dari nilainya.
- Setiap name/value pasangan dipisahkan dari yang lain dengan koma (,).
- Beberapa antrian dipisahkan oleh koma (,) antara akhir kurung kurawal antrian (}) dan awal kurung kurawal antrian berikutnya ({}).

Aturan untuk mengkonfigurasi WLM dengan menggunakan command prompt pada sistem operasi Windows

Ikuti aturan ini untuk menjalankan AWS CLI perintah dengan parameter pada satu baris:

- Seluruh struktur JSON harus diapit tanda kutip ganda (") dan satu set tanda kurung ([]).
- Semua nama parameter dan nilai parameter harus diapit tanda kutip ganda (").
- Dalam ParameterValue nilai, Anda harus melampirkan seluruh struktur bersarang dalam tanda kutip ganda (") dan tanda kurung ([]).
- Dalam struktur bersarang, masing-masing properti dan nilai untuk setiap antrian harus diapit kurung kurawal ({}).
- Dalam struktur bersarang, Anda harus menggunakan karakter escape backslash (\) sebelum setiap tanda kutip ganda (") dan karakter escape backslash (\). Persyaratan ini berarti bahwa Anda akan menggunakan tiga garis miring terbalik dan tanda kutip ganda untuk memastikan bahwa properti diteruskan dengan benar (\\").
- Untuk name/value pasangan, titik dua (:) memisahkan setiap properti dari nilainya.
- Setiap name/value pasangan dipisahkan dari yang lain dengan koma (,).
- Beberapa antrian dipisahkan oleh koma (,) antara akhir kurung kurawal antrian (}) dan awal kurung kurawal antrian berikutnya ({}).

Membuat grup parameter

Jika Anda ingin mengatur nilai parameter yang berbeda dari grup parameter default, Anda dapat membuat grup parameter Anda sendiri,

Untuk membuat grup parameter

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Konfigurasi, lalu pilih Manajemen beban kerja untuk menampilkan halaman Manajemen beban kerja.
3. Pilih Buat untuk menampilkan jendela Buat grup parameter.
4. Masukkan nilai untuk nama grup Parameter dan Deskripsi.
5. Pilih Buat untuk membuat grup parameter.

Mengubah grup parameter

Anda dapat melihat salah satu grup parameter Anda untuk melihat ringkasan nilai untuk konfigurasi parameter dan manajemen beban kerja (WLM). Anda dapat memodifikasi parameter untuk mengubah pengaturan parameter dan properti konfigurasi WLM.

Note

Anda tidak dapat mengubah grup parameter default.

Konsol Manajemen AWS

Di konsol, parameter grup muncul di tab Parameter, dan antrian beban kerja muncul di tab Manajemen Beban Kerja.

Untuk memodifikasi grup parameter

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Konfigurasi, lalu pilih Manajemen beban kerja untuk menampilkan halaman Manajemen beban kerja.
3. Pilih grup parameter yang ingin Anda modifikasi untuk menampilkan halaman detail, dengan tab untuk Parameter dan manajemen Beban Kerja.
4. Pilih tab Parameter untuk melihat pengaturan parameter saat ini.
5. Pilih Edit parameter untuk mengaktifkan perubahan pengaturan untuk parameter ini:
 - 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`

Untuk informasi selengkapnya tentang parameter ini, lihat [Grup parameter Amazon Redshift](#).

6. Masukkan perubahan Anda dan kemudian pilih Simpan untuk memperbarui grup parameter.

Untuk memodifikasi konfigurasi WLM untuk grup parameter

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Konfigurasi, lalu pilih Manajemen beban kerja untuk menampilkan halaman Manajemen beban kerja.
3. Pilih grup parameter yang ingin Anda modifikasi untuk menampilkan halaman detail dengan tab untuk Parameter dan manajemen Beban Kerja.
4. Pilih tab Manajemen beban kerja untuk melihat konfigurasi WLM saat ini.
5. Pilih Edit antrian beban kerja untuk mengedit konfigurasi WLM.
6. (Opsional) Pilih Aktifkan akselerasi kueri pendek untuk mengaktifkan akselerasi kueri pendek (SQA).

Saat Anda mengaktifkan SQA, Waktu berjalan maksimum untuk kueri singkat (1 hingga 20 detik) diatur ke Dinamis secara default. Untuk mengatur runtime maksimum ke nilai tetap, pilih nilai 1-20.

7. Lakukan satu atau beberapa hal berikut untuk mengubah konfigurasi antrian:
 - Pilih mode Switch WLM untuk memilih antara WLM Otomatis dan WLM Manual.

Dengan WLM Otomatis, Memori dan Konkurensi pada nilai utama diatur ke auto.

- Untuk membuat antrian, pilih Edit antrian beban kerja, lalu pilih Tambah Antrian.
- Untuk memodifikasi antrian, ubah nilai properti dalam tabel. Tergantung pada jenis antrian, properti dapat mencakup yang berikut:
 - Nama antrian dapat diubah.
 - Memori (%)

- Konkurensi pada cluster utama
- Mode penskalaan konkurensi dapat dimatikan atau otomatis
- Batas waktu (ms)
- Grup pengguna
- Grup kueri
- Peran pengguna

Untuk informasi selengkapnya tentang properti ini, lihat [Properti untuk parameter konfigurasi WLM](#).

 Important

Jika Anda mengubah nama antrian, nilai QueueName dimensi metrik antrian WLM (seperti, WLMQueue Panjang,, WLMQuery Durasi, WLMRunning Kueri WLMQueue WaitTime WLMQueriesCompletedPerSecond, dan sebagainya) juga berubah. Jadi, jika Anda mengubah nama antrian, Anda mungkin perlu mengubah CloudWatch alarm yang telah Anda atur.

- Untuk mengubah urutan antrian, pilih tombol panah Atas dan Bawah.
 - Untuk menghapus antrian, pilih Hapus di baris antrian dalam tabel.
8. (Opsional) Pilih Tunda perubahan dinamis hingga reboot agar perubahan diterapkan ke cluster setelah reboot berikutnya.

 Note

Beberapa perubahan memerlukan reboot cluster terlepas dari pengaturan ini. Untuk informasi selengkapnya, lihat [Sifat dinamis dan statis WLM](#).

9. Pilih Simpan.

AWS CLI

Untuk mengonfigurasi parameter Amazon Redshift dengan menggunakan AWS CLI, Anda menggunakan [modify-cluster-parameter-group](#) perintah untuk grup parameter tertentu. Anda menentukan kelompok parameter untuk memodifikasi parameter-group-name. Anda menggunakan parameters parameter (untuk modify-cluster-parameter-group perintah)

untuk menentukan name/value pasangan untuk setiap parameter yang ingin Anda modifikasi dalam kelompok parameter.

 Note

Ada pertimbangan khusus saat mengkonfigurasi `wlm_json_configuration` parameter dengan menggunakan AWS CLI. Contoh di bagian ini berlaku untuk semua parameter kecuali `wlm_json_configuration`. Untuk informasi selengkapnya tentang mengonfigurasi `wlm_json_configuration` dengan menggunakan AWS CLI, lihat [Manajemen beban kerja](#).

Setelah Anda memodifikasi nilai parameter, Anda harus me-reboot cluster apa pun yang terkait dengan grup parameter yang dimodifikasi. Status cluster ditampilkan `applying` untuk `ParameterApplyStatus` sementara nilai-nilai sedang diterapkan, dan kemudian `pending-reboot` setelah nilai-nilai telah diterapkan. Setelah Anda reboot, database di cluster Anda mulai menggunakan nilai parameter baru. Untuk informasi selengkapnya tentang me-reboot cluster, lihat [Melakukan boot ulang klaster](#).

 Note

`wlm_json_configurationParameter` berisi beberapa properti yang dinamis dan tidak mengharuskan Anda untuk me-reboot cluster terkait untuk perubahan yang akan diterapkan. Untuk informasi selengkapnya tentang properti dinamis dan statis, lihat [Sifat dinamis dan statis WLM](#).

Sintaks berikut menunjukkan cara menggunakan `modify-cluster-parameter-group` perintah untuk mengkonfigurasi parameter. Anda menentukan *parameter_group_name* dan mengganti keduanya *parameter_name* dan *parameter_value* dengan parameter aktual untuk memodifikasi dan nilai untuk parameter itu. Jika Anda ingin memodifikasi lebih dari satu parameter pada saat yang sama, pisahkan setiap parameter dan nilai yang ditetapkan dari parameter berikutnya dengan spasi.

```
aws redshift modify-cluster-parameter-group --parameter-group-name parameter_group_name --parameters
ParameterName=parameter_name,ParameterValue=parameter_value
```

Contoh berikut menunjukkan cara mengkonfigurasi `statement_timeout` dan `enable_user_activity_logging` parameter untuk kelompok `myclusterparametergroup` parameter.

 Note

Untuk tujuan keterbacaan, contoh ditampilkan pada beberapa baris, tetapi sebenarnya AWS CLI ini adalah satu baris.

```
aws redshift modify-cluster-parameter-group
--parameter-group-name myclusterparametergroup
--parameters ParameterName=statement_timeout,ParameterValue=20000
ParameterName=enable_user_activity_logging,ParameterValue=true
```

Membuat aturan pemantauan kueri

Anda dapat menggunakan konsol Amazon Redshift untuk membuat dan memodifikasi aturan pemantauan kueri WLM. Aturan pemantauan kueri adalah bagian dari parameter konfigurasi WLM untuk grup parameter. Jika Anda memodifikasi aturan pemantauan kueri (QMR), perubahan terjadi secara otomatis tanpa perlu memodifikasi cluster. Untuk informasi selengkapnya, lihat [aturan pemantauan kueri WLM](#).

Saat Anda membuat aturan, Anda menentukan nama aturan, satu atau beberapa predikat, dan tindakan.

Ketika Anda menyimpan konfigurasi WLM yang menyertakan aturan, Anda dapat melihat kode JSON untuk definisi aturan sebagai bagian dari JSON untuk parameter konfigurasi WLM.

Untuk membuat aturan pemantauan kueri

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Konfigurasi, lalu pilih Manajemen beban kerja untuk menampilkan halaman Manajemen beban kerja.
3. Pilih grup parameter yang ingin Anda modifikasi untuk menampilkan halaman detail dengan tab untuk Parameter dan manajemen Beban Kerja.

4. Pilih tab Manajemen beban kerja, dan pilih Edit antrian beban kerja untuk mengedit konfigurasi WLM.
5. Tambahkan aturan baru baik dengan menggunakan template yang telah ditentukan atau dari awal.

Untuk menggunakan template yang telah ditentukan, lakukan hal berikut:

1. Pilih Tambahkan aturan dari templat di grup Aturan pemantauan kueri. Daftar template aturan ditampilkan.
2. Pilih satu atau beberapa templat aturan. Ketika Anda memilih Simpan, WLM membuat satu aturan untuk setiap template yang Anda pilih.
3. Masukkan atau konfirmasi nilai untuk aturan, termasuk nama Aturan, Predikat, dan Tindakan.
4. Pilih Simpan.

Untuk menambahkan aturan baru dari awal, lakukan hal berikut:

1. Untuk menambahkan predikat tambahan, pilih Tambahkan predikat. Anda dapat memiliki hingga tiga predikat untuk setiap aturan. Jika semua predikat terpenuhi, WLM memicu tindakan terkait.
2. Pilih Tindakan. Setiap aturan memiliki satu tindakan.
3. Pilih Simpan.

Amazon Redshift menghasilkan parameter konfigurasi WLM Anda dalam format JSON dan menampilkannya di bagian JSON.

Menghapus grup parameter

Anda dapat menghapus grup parameter jika Anda tidak lagi membutuhkannya dan tidak terkait dengan cluster apa pun. Anda hanya dapat menghapus grup parameter khusus.

Untuk menghapus grup parameter

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>

2. Pada menu navigasi, pilih Konfigurasi, lalu pilih Manajemen beban kerja untuk menampilkan halaman Manajemen beban kerja.
3. Untuk grup Parameter, pilih grup parameter yang ingin Anda ubah.

 Note

Anda tidak dapat menghapus grup parameter default.

4. Pilih Hapus dan konfirmasikan bahwa Anda ingin menghapus grup parameter.

Integrasikan Amazon Redshift dengan Mitra AWS

Dengan bekerja sama dengan Amazon Redshift, Anda dapat berintegrasi dengan AWS Partner di konsol Amazon Redshift. Dari halaman detail Cluster, Anda dapat mempercepat orientasi data ke gudang data Amazon Redshift AWS dengan aplikasi Partner. Anda juga dapat bergabung dan menganalisis data dari berbagai sumber bersama dengan data yang ada di cluster Anda. Sebelum menyelesaikan integrasi dengan Informatica, Anda harus menambahkan alamat IP mitra ke daftar lalu lintas masuk yang diizinkan. AWS Mitra berikut dapat berintegrasi dengan Amazon Redshift:

- [Datacoral](#)
- [Etleap](#)
- [Fivetran](#)
- [SnapLogic](#)
- [Jahitan](#)
- [Upsolver](#)
- [Matillion \(pratinjau\)](#)
- [Sisense \(pratinjau\)](#)
- [Thoughtspot](#)

AWS Mitra dapat berintegrasi dengan Amazon Redshift menggunakan operasi API Amazon Redshift AWS CLI atau Amazon. Untuk informasi selengkapnya, lihat Referensi AWS CLI Perintah atau Referensi API Amazon Redshift.

Gunakan prosedur berikut untuk mengintegrasikan cluster dengan AWS Partner.

Untuk mengintegrasikan cluster Amazon Redshift dengan Partner AWS

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster.
3. Pilih cluster yang ingin Anda integrasikan.
4. Pilih Tambahkan integrasi mitra. Halaman Pilih mitra terbuka dengan detail tentang AWS Mitra yang tersedia.
5. Pilih AWS Mitra, lalu pilih Berikutnya.

Rincian lebih lanjut tentang AWS Partner yang dipilih muncul, bersama dengan detail tentang cluster yang Anda integrasikan. Bagian detail Cluster mencakup informasi yang Anda berikan di situs web AWS Mitra seperti pengenalan Cluster, Endpoint, nama Database, dan nama Pengguna (yang merupakan nama pengguna database). Informasi ini dikirim ke mitra yang Anda pilih.

6. Pilih Tambah mitra untuk membuka situs web AWS Mitra.
7. Konfigurasi integrasi dengan cluster Amazon Redshift Anda di situs web mitra. Di situs web mitra, Anda dapat memilih dan mengonfigurasi sumber data yang dimuat ke cluster Amazon Redshift Anda. Anda juga dapat menentukan transformasi ekstrak, pemuatan, dan transformasi (ELT) tambahan untuk memproses data bisnis Anda, menggabungkannya dengan kumpulan data lain, dan membangun tampilan konsolidasi untuk analisis dan pelaporan.

Anda dapat melihat dan mengelola integrasi AWS Partner dari tab Properties detail cluster. Bagian Integrasi mencantumkan nama Mitra yang dapat Anda gunakan untuk menautkan ke situs web AWS Mitra, Status integrasi, Database yang menerima data, dan koneksi terakhir yang berhasil yang mungkin telah memperbarui klaster.

Nilai status yang mungkin adalah sebagai berikut:

- Aktif — AWS Mitra dapat terhubung ke cluster dan menyelesaikan tugas yang dikonfigurasi.
- Tidak Aktif — Integrasi AWS Mitra tidak ada.
- Kegagalan runtime — AWS Partner dapat terhubung ke cluster tetapi tidak dapat menyelesaikan tugas yang dikonfigurasi.
- Kegagalan koneksi — AWS Mitra tidak dapat terhubung ke cluster.

Setelah Anda menghapus integrasi AWS Partner dari Amazon Redshift, data terus mengalir ke klaster Anda. Selesaikan penghapusan di situs web mitra.

Memuat data dengan AWS mitra

Selain mengintegrasikan mitra dengan cluster Amazon Redshift, Anda juga dapat memindahkan data dari lebih dari 30 sumber ke cluster Amazon Redshift menggunakan alat pemuatan data mitra kami. Sebelum melakukannya, Anda harus menambahkan alamat IP mitra (ditemukan di bawah) ke daftar aturan masuk yang diizinkan. Untuk informasi selengkapnya tentang menambahkan aturan ke grup keamanan Amazon EC2, lihat [Mengotorisasi Lalu Lintas Masuk untuk Instans Anda di](#) Panduan

Pengguna Amazon EC2. Perhatikan bahwa meskipun alat Pemuat Data Informatica gratis, biaya masuknya data mungkin berlaku tergantung pada sumber data dan target yang Anda pilih.

Anda dapat memuat data dari mitra berikut:

- [Informatika](#)

Untuk memuat data ke cluster Amazon Redshift dengan mitra

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih integrasi AWS mitra, lalu pilih mitra yang ingin Anda integrasikan dengan cluster Anda.
3. Pilih Integrasi lengkap<partner-name>. Anda akan diarahkan ke situs integrasi mitra.
4. Masukkan detail yang diperlukan di situs mitra dan selesaikan integrasi.

Simpul terpesan

Di AWS, biaya yang Anda dapatkan untuk menggunakan Amazon Redshift didasarkan pada node komputasi. Setiap node komputasi ditagih dengan tarif per jam. Tarif per jam bervariasi tergantung pada faktor-faktor seperti wilayah, tipe node, dan apakah node menerima harga node sesuai permintaan atau harga node cadangan.

Harga node sesuai permintaan adalah opsi yang paling mahal, tetapi paling fleksibel di Amazon Redshift. Dengan tarif sesuai permintaan, Anda hanya dikenakan biaya untuk node komputasi yang Anda miliki di cluster yang sedang berjalan. Jika Anda mematikan atau menghapus klaster, Anda tidak lagi dikenakan biaya untuk node komputasi yang ada di cluster itu. Anda ditagih hanya untuk node komputasi yang Anda gunakan, dan tidak lebih. Tarif per jam yang dikenakan biaya untuk setiap node komputasi bervariasi tergantung pada faktor-faktor seperti wilayah dan tipe node.

Harga node cadangan lebih murah daripada harga sesuai permintaan karena node komputasi ditagih dengan tarif per jam diskon. Namun, untuk menerima tarif diskon ini, Anda harus membeli penawaran node yang dipesan. Ketika Anda membeli penawaran, Anda membuat reservasi. Reservasi menetapkan tarif diskon untuk setiap node yang Anda pesan selama reservasi. Tarif diskon dalam penawaran bervariasi tergantung pada faktor-faktor seperti wilayah, jenis node, durasi, dan opsi pembayaran.

Anda dapat menetapkan node sebagai node cadangan dengan memanggil operasi `PurchaseReservedNodeOffering` API atau memilih Purchase reserved node di konsol Amazon Redshift. Saat Anda membeli node yang dicadangkan, Anda harus menentukan AWS Region, tipe node, istilah, jumlah node, dan tipe penawaran untuk tipe node cadangan yang berlaku. Node cadangan hanya dapat digunakan di AWS Wilayah yang ditunjuk.

Topik ini membahas apa itu penawaran node cadangan dan bagaimana Anda dapat membelinya untuk mengurangi biaya menjalankan cluster Amazon Redshift Anda. Topik ini membahas tarif secara umum sebagai sesuai permintaan atau diskon sehingga Anda dapat memahami konsep penetapan harga dan bagaimana harga memengaruhi penagihan. Untuk informasi selengkapnya tentang tarif tertentu, buka Harga [Amazon Redshift](#).

Penawaran node cadangan

Jika Anda berniat untuk menjaga klaster Amazon Redshift Anda berjalan terus menerus untuk waktu yang lama, Anda harus mempertimbangkan untuk membeli penawaran node cadangan. Penawaran ini memberikan penghematan yang signifikan atas harga sesuai permintaan, tetapi

mereka mengharuskan Anda untuk memesan node komputasi dan berkomitmen untuk membayar node tersebut untuk durasi satu tahun atau tiga tahun.

Node cadangan adalah konsep penagihan yang digunakan secara ketat untuk menentukan tarif di mana Anda dikenakan biaya untuk node. Reservasi node sebenarnya tidak membuat node apa pun untuk Anda. Anda dikenakan biaya untuk node cadangan terlepas dari penggunaan, yang berarti bahwa Anda harus membayar untuk setiap node yang Anda pesan selama durasi reservasi, terlepas dari apakah Anda memiliki node dalam klaster yang berjalan di mana tarif diskon berlaku atau tidak.

Pada tahap evaluasi proyek Anda atau ketika Anda mengembangkan bukti konsep, harga sesuai permintaan memberi Anda fleksibilitas untuk membayar saat Anda pergi, membayar hanya untuk apa yang Anda gunakan, dan berhenti membayar kapan saja dengan mematikan atau menghapus cluster. Setelah Anda menetapkan kebutuhan lingkungan produksi Anda dan memulai fase implementasi, Anda harus mempertimbangkan untuk memesan node komputasi dengan membeli satu atau lebih penawaran.

Penawaran dapat diterapkan ke satu atau lebih node komputasi. Anda menentukan jumlah node komputasi yang akan dipesan saat Anda membeli penawaran. Anda dapat memilih untuk membeli satu penawaran untuk beberapa node komputasi, atau Anda dapat memilih untuk membeli beberapa penawaran dan menentukan sejumlah node komputasi dalam setiap penawaran.

Misalnya, salah satu dari berikut ini adalah cara yang valid untuk membeli penawaran untuk tiga node komputasi:

- Beli satu penawaran dan tentukan tiga node komputasi.
- Beli dua penawaran, dan tentukan satu node komputasi untuk penawaran pertama dan dua node komputasi untuk penawaran kedua.
- Beli tiga penawaran, dan tentukan satu node komputasi untuk setiap penawaran.

Membandingkan harga di antara penawaran node yang dicadangkan

Amazon Redshift menyediakan beberapa opsi pembayaran untuk penawaran. Opsi pembayaran yang Anda pilih memengaruhi jadwal pembayaran dan tarif diskon yang dikenakan untuk reservasi. Semakin banyak Anda membayar di muka untuk reservasi, semakin baik penghematan keseluruhan.

Opsi pembayaran berikut tersedia untuk penawaran. Penawaran terdaftar dalam urutan dari penghematan paling sedikit hingga paling banyak dibandingkan tingkat permintaan.

Note

Anda dikenakan tarif per jam yang berlaku untuk setiap jam dalam durasi reservasi yang ditentukan, terlepas dari apakah Anda menggunakan node yang dipesan atau tidak. Opsi pembayaran hanya menentukan frekuensi pembayaran dan diskon yang akan diterapkan. Untuk informasi selengkapnya, lihat [Penawaran node cadangan](#).

Opsi pembayaran	Jadwal pembayaran	Tabungan komparatif	Durasi	Biaya di muka	Biaya bulanan berulang
Tidak ada di muka	Angsuran bulanan selama reservasi. Tidak ada pembayaran di muka.	Sekitar 20% discount over on-demand rates.	Jangka waktu satu tahun atau tiga tahun	Tidak ada	Ya
Sebagian di muka	Pembayaran dimuka sebagian, dan cicilan bulanan selama reservasi.	Diskon hingga 41 persen hingga 73 persen tergantung durasi.	Jangka waktu satu tahun atau tiga tahun	Ya	Ya
Semua di muka	Pembayaran dimuka penuh untuk reservasi. Tidak ada biaya bulanan.	Diskon hingga 42 persen hingga 76 persen tergantung durasi.	Jangka waktu satu tahun atau tiga tahun	Ya	Tidak ada

Opsi dan durasi tertentu tergantung pada ketersediaan.

 Note

Jika sebelumnya Anda membeli penawaran Pemanfaatan Berat untuk Amazon Redshift, penawaran yang sebanding adalah penawaran Partial Upfront.

Cara kerja node yang dicadangkan

Dengan penawaran node cadangan, Anda membayar sesuai dengan ketentuan pembayaran seperti yang dijelaskan di bagian sebelumnya. Anda membayar dengan cara ini apakah Anda sudah memiliki cluster yang sedang berjalan atau Anda meluncurkan klaster setelah Anda memiliki reservasi.

Ketika Anda membeli penawaran, reservasi Anda memiliki status pembayaran-tertunda sampai reservasi diproses. Jika reservasi gagal diproses, status ditampilkan sebagai pembayaran gagal dan Anda dapat mencoba prosesnya lagi. Setelah reservasi Anda berhasil diproses, statusnya berubah menjadi aktif. Tarif diskon yang berlaku dalam reservasi Anda tidak berlaku untuk tagihan Anda sampai status berubah menjadi aktif. Setelah durasi reservasi berlalu, status berubah menjadi pensiunan tetapi Anda dapat terus mengakses informasi tentang reservasi untuk tujuan historis. Ketika reservasi dihentikan, klaster Anda terus berjalan tetapi Anda mungkin ditagih dengan tarif sesuai permintaan kecuali Anda memiliki reservasi lain yang menerapkan harga diskon ke node.

Node yang dicadangkan khusus untuk wilayah tempat Anda membeli penawaran. Jika Anda membeli penawaran dengan menggunakan konsol Amazon Redshift, pilih AWS wilayah tempat Anda ingin membeli penawaran, lalu selesaikan proses reservasi. Jika Anda membeli penawaran secara terprogram, wilayah ditentukan oleh titik akhir Amazon Redshift yang Anda sambungkan. Untuk informasi selengkapnya tentang wilayah Amazon Redshift, buka [Wilayah dan Titik Akhir](#) di Referensi Umum Amazon Web Services

Untuk memastikan bahwa tingkat diskon diterapkan ke semua node saat Anda meluncurkan klaster, pastikan bahwa wilayah, jenis node, dan jumlah node yang Anda pilih cocok dengan satu atau lebih reservasi aktif. Jika tidak, Anda akan dikenakan tarif sesuai permintaan untuk node yang tidak cocok dengan reservasi aktif.

Dalam klaster yang sedang berjalan, jika Anda melebihi jumlah node yang telah Anda pesan, Anda mulai menambah biaya untuk node tambahan tersebut dengan tarif sesuai permintaan. Akrua ini berarti Anda dapat dikenakan tarif yang bervariasi untuk node di cluster yang sama tergantung pada berapa banyak node yang telah Anda pesan. Anda dapat membeli penawaran lain untuk menutupi

node tambahan tersebut, dan kemudian tarif diskon diterapkan ke node tersebut selama sisa durasi setelah status reservasi menjadi aktif.

Jika Anda mengubah ukuran cluster Anda menjadi tipe node yang berbeda dan Anda belum memesan node dari jenis itu, Anda akan dikenakan biaya pada tingkat permintaan. Anda dapat membeli penawaran lain dengan tipe node baru jika Anda ingin menerima tarif diskon untuk klaster yang diubah ukurannya. Namun, Anda juga terus membayar reservasi awal hingga durasinya berlalu. [Jika Anda perlu mengubah reservasi sebelum jangka waktu berakhir, buat kasus dukungan menggunakan Konsol.AWS](#)

Note

Konsol menunjukkan jumlah node cadangan yang digunakan dan yang tidak digunakan. Namun, konsol hanya menampilkan jumlah node seperti yang digunakan yang digunakan akun pengguna saat ini. Jika akun pengguna lain di bawah akun pembayar yang sama menggunakan node, konsol menunjukkan node tersebut sebagai tidak digunakan.

Contoh

- Akun pembayar menyimpan 20 node
- Akun pengguna saat ini menggunakan enam node
- Akun pengguna lain di bawah akun pembayar yang sama juga menggunakan enam node

Dalam contoh ini, konsol hanya menampilkan enam node yang digunakan, dan empat belas node yang tidak digunakan.

Node cadangan dan penagihan terkonsolidasi

Manfaat harga dari Node Cadangan dibagikan ketika akun pembelian merupakan bagian dari serangkaian akun yang ditagih di bawah satu akun pembayar tagihan konsolidasi. Penggunaan per jam di semua sub-akun dikumpulkan di akun pembayar setiap bulan. Ini biasanya berguna untuk perusahaan di mana ada tim atau kelompok fungsional yang berbeda; kemudian, logika Node Cadangan normal diterapkan untuk menghitung tagihan. Untuk informasi selengkapnya, lihat [Penagihan Konsolidasi](#) di AWS Billing Panduan Pengguna.

Contoh simpul cadangan

Skenario di bagian ini menunjukkan bagaimana node memperoleh biaya berdasarkan tarif sesuai permintaan dan diskon menggunakan detail reservasi berikut:

- Wilayah: AS Barat (Oregon)
- Jenis Node: ra3.xlplus
- Opsi Pembayaran: Tidak Ada di Muka
- Durasi: satu tahun
- Jumlah Node Cadangan: 16

Contoh 1

Anda memiliki satu cluster di wilayah AS Barat (Oregon) dengan 20 node.

Dalam skenario ini, 16 node menerima tarif diskon dari reservasi, tetapi 4 node tambahan di cluster ditagih pada tingkat permintaan.

Contoh 2

Anda memiliki satu cluster di wilayah AS Barat (Oregon) dengan 12 node.

Dalam skenario ini, semua 12 node di cluster menerima tarif diskon dari reservasi. Namun, Anda juga membayar untuk node cadangan yang tersisa di reservasi meskipun saat ini Anda tidak memiliki klaster yang sedang berjalan yang mereka terapkan.

Contoh 3

Anda memiliki satu cluster di wilayah AS Barat (Oregon) dengan 12 node. Anda menjalankan cluster selama beberapa bulan dengan konfigurasi ini, dan kemudian Anda perlu menambahkan node ke cluster. Anda mengubah ukuran cluster, memilih jenis node yang sama dan menentukan total 16 node.

Dalam skenario ini, Anda ditagih tarif diskon untuk 16 node. Biaya Anda tetap sama untuk durasi setahun penuh karena jumlah node yang Anda miliki di cluster sama dengan jumlah node yang telah Anda pesan.

Contoh 4

Anda memiliki satu cluster di wilayah AS Barat (Oregon) dengan 16 node. Anda menjalankan cluster selama beberapa bulan dengan konfigurasi ini, dan kemudian Anda perlu menambahkan node. Anda mengubah ukuran cluster, memilih jenis node yang sama dan menentukan total 20 node.

Dalam skenario ini, Anda akan ditagih tarif diskon untuk semua node sebelum mengubah ukuran. Setelah mengubah ukuran, Anda akan ditagih tarif diskon untuk 16 node selama sisa tahun ini, dan Anda ditagih pada tingkat permintaan untuk 4 node tambahan yang Anda tambahkan ke cluster.

Contoh 5

Anda memiliki dua kelompok di wilayah AS Barat (Oregon). Salah satu cluster memiliki 6 node, dan yang lainnya memiliki 10 node.

Dalam skenario ini, Anda ditagih pada tingkat diskon untuk semua node karena jumlah total node di kedua cluster sama dengan jumlah node yang telah Anda pesan.

Contoh 6

Anda memiliki dua kelompok di wilayah AS Barat (Oregon). Salah satu cluster memiliki 4 node, dan yang lainnya memiliki 6 node.

Dalam skenario ini, Anda ditagih tarif diskon untuk 10 node yang Anda miliki dalam menjalankan cluster, dan Anda juga membayar tarif diskon untuk tambahan 6 node yang telah Anda pesan meskipun saat ini Anda tidak memiliki cluster berjalan yang mereka terapkan.

Membeli simpul terpesan

Anda dapat menggunakan Konsol Manajemen AWS atau AWS CLI untuk membeli penawaran node yang dipesan, dan untuk melihat reservasi saat ini dan sebelumnya.

Konsol Manajemen AWS

Untuk membeli node yang dicadangkan

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster, lalu pilih Node cadangan untuk menampilkan daftar node yang dicadangkan.

3. Pilih Purchase reserved nodes untuk menampilkan halaman untuk memilih properti node yang ingin Anda beli.
4. Masukkan properti node, lalu pilih Purchase reserved nodes.

Setelah Anda membeli penawaran, daftar Node Cadangan menampilkan reservasi Anda dan detail masing-masing, seperti jenis node, jumlah node, dan status reservasi. Untuk informasi selengkapnya tentang detail reservasi, lihat [Cara kerja node yang dicadangkan](#).

Untuk memutakhirkan node yang dicadangkan, gunakan file AWS CLI.

Anda tidak dapat mengonversi semua jenis node menjadi node yang dicadangkan, dan mungkin juga node cadangan yang ada tidak tersedia untuk pembaruan. Ini mungkin karena tipe node dihentikan. Hubungi dukungan pelanggan untuk memperbarui tipe node yang dihentikan.

AWS CLI

Untuk memutakhirkan reservasi node yang dipesan dengan AWS CLI

1. Dapatkan daftar ReservedNodeOffering ID untuk penawaran yang memenuhi persyaratan Anda untuk jenis pembayaran, jangka waktu, dan biaya. Contoh berikut menggambarkan langkah ini.

```
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. Panggil `accept-reserved-node-exchange` dan berikan ID untuk node DC1 cadangan yang ingin Anda tukarkan bersama dengan `ReservedNodeOffering` ID yang Anda peroleh pada langkah sebelumnya.

Contoh berikut menggambarkan langkah ini.

```

aws redshift accept-reserved-node-exchange --reserved-node-id xxxxxxxx-xxxx-
xxxx-xxxx-xxxxxxxxxxxxxx --target-reserved-node-offering-id yyyyyyyy-yyyy-yyyy-
yyyy-yyyyyyyyyyyyyy
{
  "ExchangedReservedNode": {
    "UsagePrice": 0.0,
    "OfferingType": "No Upfront",
    "State": "exchanging",
    "FixedPrice": 0.0,
    "CurrencyCode": "USD",
    "ReservedNodeId": "zzzzzzzz-zzzz-zzzz-zzzz-zzzzzzzzzzzz",
    "NodeType": "dc2.large",
    "NodeCount": 1,
    "RecurringCharges": [
      {
        "RecurringChargeFrequency": "Hourly",
        "RecurringChargeAmount": 0.2
      }
    ],
    "ReservedNodeOfferingType": "Regular",
    "StartTime": "2018-06-27T18:02:58Z",
    "ReservedNodeOfferingId": "yyyyyyyy-yyyy-yyyy-yyyy-yyyyyyyyyyyyyy",
    "Duration": 31536000
  }
}

```

Anda dapat mengonfirmasi bahwa pertukaran selesai dengan menelepon [describe-reserved-nodes](#) dan memeriksa nilai `node type`.

Keamanan di Amazon Redshift

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. [Model tanggung jawab bersama](#) menggambarkan hal ini sebagai keamanan dari cloud dan keamanan di cloud:

- **Keamanan cloud** — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Efektivitas keamanan kami diuji dan diverifikasi secara rutin oleh auditor pihak ketiga sebagai bagian dari [program kepatuhan AWS](#). Untuk mempelajari tentang program kepatuhan yang berlaku untuk Amazon Redshift, lihat [AWS layanan dalam cakupan berdasarkan program kepatuhan](#).
- **Keamanan di cloud** — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, termasuk sensitivitas data, persyaratan perusahaan, serta hukum dan peraturan yang berlaku.

Akses ke sumber daya Amazon Redshift dikendalikan pada empat level:

- **Manajemen cluster** — Kemampuan untuk membuat, mengkonfigurasi, dan menghapus cluster dikendalikan oleh izin yang diberikan kepada pengguna atau akun yang terkait dengan kredensi AWS keamanan Anda. Pengguna dengan izin yang tepat dapat menggunakan Konsol Manajemen AWS, AWS Command Line Interface (CLI), atau Amazon Redshift Application Programming Interface (API) untuk mengelola cluster mereka. Akses ini dikelola dengan menggunakan kebijakan IAM.

Important

Amazon Redshift memiliki kumpulan praktik terbaik untuk mengelola izin, identitas, dan akses aman. Kami menyarankan Anda untuk membiasakan diri dengan ini saat Anda memulai dengan Amazon Redshift. Untuk informasi selengkapnya, lihat [Manajemen identitas dan akses di Amazon Redshift](#).

- **Konektivitas klaster** — Grup keamanan Amazon Redshift menentukan AWS instans yang diizinkan untuk terhubung ke klaster Amazon Redshift dalam format Classless Inter-Domain Routing (CIDR).

Untuk informasi tentang membuat Amazon Redshift, Amazon EC2, dan grup keamanan Amazon VPC dan mengaitkannya dengan cluster, lihat [Grup keamanan Amazon Redshift](#)

- Akses database — Kemampuan untuk mengakses objek database, seperti tabel dan tampilan, dikendalikan oleh akun pengguna database di database Amazon Redshift. Pengguna hanya dapat mengakses sumber daya dalam database yang akun pengguna mereka telah diberikan izin untuk mengakses. Anda membuat akun pengguna Amazon Redshift ini dan mengelola izin dengan menggunakan pernyataan CREATE [USER](#), CREATE [GROUP](#), [GRANT](#), dan [REVOKE](#) SQL. Untuk informasi selengkapnya, lihat [Mengelola keamanan database](#) di Panduan Pengembang Database Amazon Redshift.
- Kredensi database sementara dan sistem masuk tunggal — Selain membuat dan mengelola pengguna database menggunakan perintah SQL, seperti CREATE USER dan ALTER USER, Anda dapat mengonfigurasi klien SQL Anda dengan driver Amazon Redshift JDBC atau ODBC khusus. Driver ini mengelola proses pembuatan pengguna database dan kata sandi sementara sebagai bagian dari proses logon database.

Driver mengautentikasi pengguna database berdasarkan otentikasi AWS Identity and Access Management (IAM). Jika Anda sudah mengelola identitas pengguna di luar AWS, Anda dapat menggunakan penyedia identitas (IDP) yang sesuai dengan SAMP 2.0 untuk mengelola akses ke sumber daya Amazon Redshift. Anda menggunakan peran IAM untuk mengonfigurasi IDP Anda AWS dan mengizinkan pengguna federasi Anda menghasilkan kredensi database sementara dan masuk ke database Amazon Redshift. Untuk informasi selengkapnya, lihat [Menggunakan otentikasi IAM untuk menghasilkan kredensi pengguna basis data](#).

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan Amazon Redshift. Topik berikut menunjukkan cara mengonfigurasi Amazon Redshift untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga mempelajari cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan sumber daya Amazon Redshift Anda.

Topik

- [Perlindungan data di Amazon Redshift](#)
- [Manajemen identitas dan akses di Amazon Redshift](#)
- [Mengelola kata sandi admin Amazon Redshift menggunakan AWS Secrets Manager](#)
- [Pencatatan dan pemantauan di Amazon Redshift](#)
- [Validasi kepatuhan untuk Amazon Redshift](#)

- [Ketahanan dalam Pergeseran Merah Amazon](#)
- [Keamanan infrastruktur di Amazon Redshift](#)
- [Analisis konfigurasi dan kerentanan di Amazon Redshift](#)

Perlindungan data di Amazon Redshift

[Model tanggung jawab AWS bersama model](#) berlaku untuk perlindungan data di Amazon Redshift. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk mempertahankan kendali atas konten yang di-host pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Lihat informasi yang lebih lengkap tentang privasi data dalam [Pertanyaan Umum Privasi Data](#). Lihat informasi tentang perlindungan data di Eropa di pos blog [Model Tanggung Jawab Bersama dan GDPR AWS](#) di Blog Keamanan AWS.

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensial dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan AWS sumber daya. Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan pencatatan aktivitas pengguna dengan AWS CloudTrail. Untuk informasi tentang penggunaan CloudTrail jejak untuk menangkap AWS aktivitas, lihat [Bekerja dengan CloudTrail jejak](#) di AWS CloudTrail Panduan Pengguna.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola tingkat lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-3 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi selengkapnya tentang titik akhir FIPS yang tersedia di [Standar Pemrosesan Informasi Federal \(FIPS\) 140-3](#).

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan Amazon Redshift atau lainnya Layanan AWS menggunakan konsol, APIAWS CLI, atau. AWS SDKs Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

Enkripsi data

Perlindungan data mengacu pada melindungi data saat transit (saat melakukan perjalanan ke dan dari Amazon Redshift) dan saat istirahat (saat disimpan di disk di pusat data Amazon Redshift). Anda dapat melindungi data dalam perjalanan dengan menggunakan SSL atau dengan menggunakan enkripsi sisi klien. Anda memiliki opsi berikut untuk melindungi data saat istirahat di Amazon Redshift.

- Gunakan enkripsi sisi server — Anda meminta Amazon Redshift untuk mengenkripsi data Anda sebelum menyimpannya di disk di pusat datanya dan mendekripsi ketika Anda mengunduh objek.
- Gunakan enkripsi sisi klien — Anda dapat mengenkripsi data sisi klien dan mengunggah data terenkrpsi ke Amazon Redshift. Dalam hal ini, Anda mengelola proses enkripsi, kunci enkripsi, dan alat terkait.

Enkripsi saat diam

Enkripsi sisi server adalah tentang enkripsi data saat istirahat—yaitu, Amazon Redshift secara opsional mengenkripsi data Anda saat menulisnya di pusat datanya dan mendekripsi untuk Anda saat Anda mengaksesnya. Selama Anda mengautentikasi permintaan Anda dan memiliki izin akses, tidak ada perbedaan dalam cara Anda mengakses data terenkrpsi atau tidak terenkrpsi.

Amazon Redshift melindungi data saat istirahat melalui enkripsi. Secara opsional, Anda dapat melindungi semua data yang disimpan pada disk dalam cluster dan semua cadangan di Amazon S3 dengan Advanced Encryption Standard AES-256.

[Untuk mengelola kunci yang digunakan untuk mengenkripsi dan mendekripsi sumber daya Amazon Redshift, Anda menggunakan \(\). AWS Key Management ServiceAWS KMS](#)

AWS KMS menggabungkan perangkat keras dan perangkat lunak yang aman dan sangat tersedia untuk menyediakan sistem manajemen kunci yang diskalakan untuk cloud. Dengan menggunakan AWS KMS, Anda dapat membuat kunci enkripsi dan menentukan kebijakan yang mengontrol bagaimana

kunci ini dapat digunakan. AWS KMS mendukung AWS CloudTrail, sehingga Anda dapat mengaudit penggunaan kunci untuk memverifikasi bahwa kunci sedang digunakan dengan tepat. Anda dapat menggunakan AWS KMS tombol Anda dalam kombinasi dengan Amazon Redshift dan layanan yang didukung.. AWS Untuk daftar layanan yang mendukung AWS KMS, lihat [Cara Penggunaan AWS Layanan AWS KMS](#) di Panduan AWS Key Management Service Pengembang.

Jika Anda memilih untuk mengelola kata sandi admin cluster atau namespace tanpa server yang disediakan, Amazon AWS Secrets Manager Redshift juga menerima kunci KMS tambahan AWS yang digunakan untuk mengenkripsi kredensial Anda. AWS Secrets Manager Kunci tambahan ini dapat berupa kunci yang dihasilkan secara otomatis dari AWS Secrets Manager, atau kunci khusus yang Anda berikan.

Editor kueri Amazon Redshift v2 dengan aman menyimpan informasi yang dimasukkan ke dalam editor kueri sebagai berikut:

- Nama Sumber Daya Amazon (ARN) dari kunci KMS yang digunakan untuk mengenkripsi data editor kueri v2.
- Informasi koneksi database.
- Nama dan isi file dan folder.

Editor kueri Amazon Redshift v2 mengenkripsi informasi menggunakan enkripsi tingkat blok dengan kunci KMS Anda atau kunci KMS akun layanan. Enkripsi data Amazon Redshift Anda dikendalikan oleh properti kluster Amazon Redshift Anda.

Topik

- [Enkripsi basis data Amazon Redshift](#)

Enkripsi basis data Amazon Redshift

Di Amazon Redshift, database Anda dienkripsi secara default untuk melindungi data Anda saat istirahat. Enkripsi database berlaku untuk cluster dan juga untuk snapshot-nya.

Anda dapat memodifikasi kluster yang tidak terenkripsi untuk menggunakan enkripsi AWS Key Management Service (AWS KMS). Untuk melakukannya, Anda dapat menggunakan kunci yang dimiliki atau kunci yang dikelola pelanggan. Saat Anda memodifikasi kluster untuk mengaktifkan AWS KMS enkripsi, Amazon Redshift secara otomatis memigrasikan data Anda ke kluster terenkripsi baru. Snapshot yang dibuat dari cluster terenkripsi juga dienkripsi. Anda juga dapat memigrasikan kluster

terenkripsi ke klaster yang tidak terenkripsi dengan memodifikasi klaster dan mengubah opsi Enkripsi database. Untuk informasi selengkapnya, lihat [Mengubah enkripsi cluster](#).

Meskipun Anda masih dapat mengubah cluster terenkripsi default menjadi tidak terenkripsi setelah membuat cluster, kami sarankan Anda menyimpan cluster yang berisi data sensitif sebagai terenkripsi. Selain itu, Anda mungkin diminta untuk menggunakan enkripsi tergantung pada pedoman atau peraturan yang mengatur data Anda. Misalnya, Standar Keamanan Data Industri Kartu Pembayaran (PCI DSS), Sarbanes-Oxley Act (SOX), Health Insurance Portability and Accountability Act (HIPAA), dan peraturan lainnya memberikan pedoman untuk menangani jenis data tertentu.

Amazon Redshift menggunakan hierarki kunci enkripsi untuk mengenkripsi database. Anda dapat menggunakan AWS Key Management Service (AWS KMS) atau modul keamanan perangkat keras (HSM) untuk mengelola kunci enkripsi tingkat atas dalam hierarki ini. Proses yang digunakan Amazon Redshift untuk enkripsi berbeda tergantung pada cara Anda mengelola kunci. Amazon Redshift secara otomatis terintegrasi dengan AWS KMS tetapi tidak dengan HSM. Saat Anda menggunakan HSM, Anda harus menggunakan sertifikat klien dan server untuk mengonfigurasi koneksi tepercaya antara Amazon Redshift dan HSM Anda.

Important

Amazon Redshift dapat kehilangan akses ke kunci KMS untuk klaster yang disediakan atau namespace tanpa server saat Anda menonaktifkan kunci KMS yang dikelola pelanggan. Dalam kasus ini, Amazon Redshift mengambil cadangan gudang data Amazon Redshift dan menempatkannya dalam keadaan selama 14 `inaccessible-kms-key` hari. Jika Anda mengembalikan kunci KMS dalam periode itu, Amazon Redshift akan memulihkan akses dan gudang akan berfungsi secara normal. Jika periode 14 hari berakhir tanpa kunci KMS dipulihkan, Amazon Redshift akan menghapus gudang data. Sementara gudang dalam `inaccessible-kms-key` keadaan, ia memiliki karakteristik sebagai berikut:

- Anda tidak dapat menjalankan kueri apa pun di gudang data.
- Jika gudang data adalah gudang produsen dari datashare, Anda tidak dapat menjalankan kueri pembagian data terhadapnya dari gudang konsumen.
- Anda tidak dapat membuat salinan snapshot lintas wilayah.

Untuk informasi tentang memulihkan kunci KMS yang dinonaktifkan, lihat [Mengaktifkan dan menonaktifkan kunci](#) di Panduan AWS Key Management Service Pengembang. Jika kunci

KMS gudang dihapus, Anda dapat menggunakan cadangan untuk membuat gudang data baru sebelum gudang `inaccessible-kms-key` status dihapus.

Peningkatan proses enkripsi untuk kinerja dan ketersediaan yang lebih baik

Enkripsi dengan RA3 node

Pembaruan proses enkripsi untuk RA3 node telah membuat pengalaman jauh lebih baik. Kueri baca dan tulis dapat berjalan selama proses dengan dampak kinerja yang lebih sedikit dari enkripsi. Juga, enkripsi selesai jauh lebih cepat. Langkah-langkah proses yang diperbarui mencakup operasi pemulihan dan migrasi metadata cluster ke cluster target. Pengalaman yang ditingkatkan berlaku untuk jenis enkripsi seperti AWS KMS, misalnya. Ketika Anda memiliki volume data skala petabyte, operasi telah berkurang dari minggu ke hari.

Sebelum mengenkripsi klaster Anda, jika Anda berencana untuk terus menjalankan beban kerja database, Anda dapat meningkatkan kinerja dan mempercepat proses dengan menambahkan node dengan pengubahan ukuran elastis. Anda tidak dapat menggunakan pengubahan ukuran elastis saat enkripsi sedang dalam proses, jadi lakukan sebelum Anda mengenkripsi. Perhatikan bahwa menambahkan node biasanya menghasilkan biaya yang lebih tinggi.

Enkripsi dengan tipe node lainnya

Saat Anda mengenkripsi cluster dengan DC2 node, Anda tidak memiliki kemampuan untuk menjalankan kueri tulis, seperti dengan RA3 node. Hanya kueri baca yang dapat dijalankan.

Catatan penggunaan untuk enkripsi dengan RA3 node

Wawasan dan sumber daya berikut membantu Anda mempersiapkan enkripsi dan memantau prosesnya.

- Menjalankan kueri setelah memulai enkripsi — Setelah enkripsi dimulai, membaca dan menulis tersedia dalam waktu sekitar lima belas menit. Berapa lama waktu yang dibutuhkan proses enkripsi penuh untuk menyelesaikan tergantung pada jumlah data pada cluster dan tingkat beban kerja.
- Berapa lama enkripsi? — Waktu untuk mengenkripsi data Anda tergantung pada beberapa faktor: Ini termasuk jumlah beban kerja yang berjalan, sumber daya komputasi yang digunakan, jumlah node, dan jenis node. Kami menyarankan agar Anda awalnya melakukan enkripsi di lingkungan pengujian. Sebagai aturan praktis, jika Anda bekerja dengan volume data dalam petabyte, kemungkinan akan memakan waktu 1-3 hari untuk menyelesaikan enkripsi.

- Bagaimana saya tahu enkripsi selesai? — Setelah Anda mengaktifkan enkripsi, penyelesaian snapshot pertama mengonfirmasi bahwa enkripsi selesai.
- Menggulung kembali enkripsi - Jika Anda perlu memutar kembali operasi enkripsi, cara terbaik untuk melakukannya adalah memulihkan dari cadangan terbaru yang diambil sebelum enkripsi dimulai. Anda harus menerapkan kembali setiap pembaruan baru (updates/deletes/inserts) setelah pencadangan terakhir.
- Melakukan pemulihan tabel — Perhatikan bahwa Anda tidak dapat memulihkan tabel dari kluster yang tidak terenkripsi ke kluster terenkripsi.
- Mengenkripsi kluster simpul tunggal — Mengenkripsi kluster simpul tunggal memiliki keterbatasan kinerja. Dibutuhkan waktu lebih lama dari enkripsi untuk cluster multi-node.
- Membuat cadangan setelah enkripsi — Saat Anda mengenkripsi data di kluster Anda, cadangan tidak dibuat sampai cluster sepenuhnya dienkripsi. Jumlah waktu yang dibutuhkan dapat bervariasi. Waktu yang dibutuhkan untuk pencadangan bisa berjam-jam hingga sehari-hari, tergantung pada ukuran cluster. Setelah enkripsi selesai, mungkin ada penundaan sebelum Anda dapat membuat cadangan.

Perhatikan bahwa karena backup-and-restore operasi terjadi selama proses enkripsi, setiap tabel atau tampilan terwujud yang dibuat dengan BACKUP NO tidak dipertahankan. Untuk informasi selengkapnya, lihat [MEMBUAT TABEL](#) atau [MEMBUAT TAMPILAN TERWUJUD](#).

Topik

- [Enkripsi menggunakan AWS KMS](#)
- [Enkripsi menggunakan modul keamanan perangkat keras](#)
- [Rotasi kunci enkripsi](#)
- [Mengubah enkripsi cluster](#)
- [Migrasi ke cluster terenkripsi HSM](#)
- [Memutar kunci enkripsi](#)

Enkripsi menggunakan AWS KMS

Saat Anda memilih AWS KMS untuk manajemen kunci dengan Amazon Redshift, ada hierarki kunci enkripsi empat tingkat. Kunci ini, dalam urutan hierarkis, adalah kunci root, kunci enkripsi cluster (CEK), kunci enkripsi database (DEK), dan kunci enkripsi data.

Saat Anda meluncurkan cluster Anda, Amazon Redshift mengembalikan daftar Amazon Redshift atau akun AWS Anda telah dibuat atau memiliki izin untuk digunakan. AWS KMS keys AWS KMS Anda memilih kunci KMS untuk digunakan sebagai kunci root Anda dalam hierarki enkripsi.

Secara default, Amazon Redshift memilih kunci AWS milik yang dibuat secara otomatis sebagai kunci root untuk AWS akun Anda untuk digunakan di Amazon Redshift.

Jika Anda tidak ingin menggunakan kunci default, Anda harus memiliki (atau membuat) kunci KMS yang dikelola pelanggan secara terpisah AWS KMS sebelum meluncurkan kluster di Amazon Redshift. Kunci yang dikelola pelanggan memberi Anda lebih banyak fleksibilitas, termasuk kemampuan untuk membuat, memutar, menonaktifkan, menentukan kontrol akses, dan mengaudit kunci enkripsi yang digunakan untuk membantu melindungi data Anda. Untuk informasi selengkapnya tentang membuat kunci KMS, lihat [Membuat Kunci](#) di Panduan AWS Key Management Service Pengembang.

Jika Anda ingin menggunakan AWS KMS kunci dari AWS akun lain, Anda harus memiliki izin untuk menggunakan kunci dan menentukan Nama Sumber Daya Amazon (ARN) di Amazon Redshift. Untuk informasi selengkapnya tentang akses ke kunci AWS KMS, lihat [Mengontrol Akses ke Kunci Anda](#) di Panduan AWS Key Management Service Pengembang.

Setelah Anda memilih kunci root, Amazon Redshift meminta yang AWS KMS menghasilkan kunci data dan mengenkripsinya menggunakan kunci root yang dipilih. Kunci data ini digunakan sebagai CEK di Amazon Redshift. AWS KMS mengeksport CEK terenkripsi ke Amazon Redshift, di mana ia disimpan secara internal pada disk dalam jaringan terpisah dari cluster bersama dengan hibah ke kunci KMS dan konteks enkripsi untuk CEK. Hanya CEK terenkripsi yang diekspor ke Amazon Redshift; kunci KMS tetap ada. AWS KMS Amazon Redshift juga meneruskan CEK terenkripsi melalui saluran aman ke cluster dan memuatnya ke dalam memori. Kemudian, Amazon Redshift memanggil AWS KMS untuk mendekripsi CEK dan memuat CEK yang didekripsi ke dalam memori. Untuk informasi selengkapnya tentang hibah, konteks enkripsi, dan konsep AWS KMS terkait lainnya, lihat [Konsep dalam Panduan AWS Key Management Service](#) Pengembang.

Selanjutnya, Amazon Redshift secara acak menghasilkan kunci untuk digunakan sebagai DEK dan memuatnya ke dalam memori di cluster. CEK yang didekripsi digunakan untuk mengenkripsi DEK, yang kemudian dilewatkan melalui saluran aman dari cluster untuk disimpan secara internal oleh Amazon Redshift pada disk di jaringan terpisah dari cluster. Seperti CEK, versi DEK yang dienkripsi dan didekripsi dimuat ke dalam memori di cluster. Versi DEK yang didekripsi kemudian digunakan untuk mengenkripsi kunci enkripsi individu yang dihasilkan secara acak untuk setiap blok data dalam database.

Saat cluster reboot, Amazon Redshift dimulai dengan versi CEK dan DEK yang tersimpan secara internal dan terenkripsi, memuat ulang ke dalam memori, dan kemudian AWS KMS memanggil untuk mendekripsi CEK dengan kunci KMS lagi sehingga dapat dimuat ke dalam memori. CEK yang didekripsi kemudian digunakan untuk mendekripsi DEK lagi, dan DEK yang didekripsi dimuat ke dalam memori dan digunakan untuk mengenkripsi dan mendekripsi kunci blok data sesuai kebutuhan.

Untuk informasi selengkapnya tentang membuat kluster Amazon Redshift yang dienkripsi dengan kunci, lihat. AWS KMS [Membuat kluster](#)

Menyalin AWS KMS—snapshot terenkripsi ke yang lain AWS Region

AWS KMS kunci khusus untuk sebuah AWS Region. Jika Anda ingin mengaktifkan penyalinan snapshot Amazon Redshift dari cluster sumber terenkripsi ke yang AWS Region lain, tetapi ingin menggunakan kunci Anda AWS KMS sendiri untuk snapshot di tujuan, Anda perlu mengonfigurasi hibah untuk Amazon Redshift untuk menggunakan kunci root di akun Anda di tujuan. AWS Region Hibah ini memungkinkan Amazon Redshift mengenkripsi snapshot di tujuan. AWS Region Jika Anda ingin snapshot di tujuan dienkripsi oleh kunci yang AWS Region dimiliki, Anda tidak perlu mengonfigurasi hibah apa pun di tujuan. AWS Region Untuk informasi selengkapnya tentang salinan snapshot lintas wilayah, lihat. [Menyalin snapshot ke Wilayah lain AWS](#)

 Note

Jika Anda mengaktifkan penyalinan snapshot dari cluster terenkripsi dan digunakan AWS KMS untuk kunci root Anda, Anda tidak dapat mengganti nama cluster Anda karena nama cluster adalah bagian dari konteks enkripsi. Jika Anda harus mengganti nama cluster Anda, Anda dapat menonaktifkan penyalinan snapshot di AWS Wilayah sumber, mengganti nama cluster, dan kemudian mengkonfigurasi dan mengaktifkan penyalinan snapshot lagi.

Proses untuk mengonfigurasi hibah untuk menyalin snapshot adalah sebagai berikut.

1. Di AWS Wilayah tujuan, buat hibah salinan snapshot dengan melakukan hal berikut:
 - Jika Anda belum memiliki AWS KMS kunci untuk digunakan, buat satu. Untuk informasi selengkapnya tentang membuat AWS KMS kunci, lihat [Membuat Kunci](#) di Panduan AWS Key Management Service Pengembang.
 - Tentukan nama untuk hibah salinan snapshot. Nama ini harus unik di AWS Wilayah itu untuk AWS akun Anda.

- Tentukan ID AWS KMS kunci tempat Anda membuat hibah. Jika Anda tidak menentukan ID kunci, hibah berlaku untuk kunci default Anda.
2. Di AWS wilayah sumber, aktifkan penyalinan snapshot dan tentukan nama hibah salinan snapshot yang Anda buat di Wilayah tujuan. AWS

Proses sebelumnya ini hanya diperlukan jika Anda mengaktifkan penyalinan snapshot menggunakan, AWS CLI Amazon Redshift API, atau SDKs. Jika Anda menggunakan konsol, Amazon Redshift menyediakan alur kerja yang tepat untuk mengonfigurasi hibah saat Anda mengaktifkan salinan snapshot lintas wilayah. Untuk informasi selengkapnya tentang mengonfigurasi salinan snapshot lintas wilayah untuk cluster AWS KMS-enkripsi menggunakan konsol, lihat [Mengonfigurasi salinan snapshot lintas wilayah untuk kluster —terenkripsi AWS KMS](#)

Sebelum snapshot disalin ke AWS Wilayah tujuan, Amazon Redshift mendekripsi snapshot menggunakan kunci root di Wilayah AWS sumber dan mengenkripsi ulang sementara menggunakan kunci RSA yang dibuat secara acak yang dikelola Amazon Redshift secara internal. Amazon Redshift kemudian menyalin snapshot melalui saluran aman ke AWS Wilayah tujuan, mendekripsi snapshot menggunakan kunci RSA yang dikelola secara internal, dan kemudian mengenkripsi ulang snapshot menggunakan kunci root di Wilayah tujuan. AWS

Enkripsi menggunakan modul keamanan perangkat keras

Jika Anda tidak menggunakan AWS KMS untuk manajemen kunci, Anda dapat menggunakan modul keamanan perangkat keras (HSM) untuk manajemen kunci dengan Amazon Redshift.

 Important

Enkripsi HSM tidak didukung untuk DC2 dan tipe RA3 node.

HSMs adalah perangkat yang memberikan kontrol langsung atas pembuatan dan manajemen kunci. Mereka memberikan keamanan yang lebih besar dengan memisahkan manajemen kunci dari lapisan aplikasi dan database. Amazon Redshift mendukung AWS CloudHSM Classic untuk manajemen kunci. Proses enkripsi berbeda ketika Anda menggunakan HSM untuk mengelola kunci enkripsi Anda alih-alih. AWS KMS

 Important

Amazon Redshift hanya AWS CloudHSM mendukung Klasik. Kami tidak mendukung AWS CloudHSM layanan yang lebih baru.

AWS CloudHSM Klasik tertutup untuk pelanggan baru. Untuk informasi selengkapnya, lihat Harga [Classic CloudHSM](#). AWS CloudHSM Klasik tidak tersedia di semua AWS Wilayah.

Untuk informasi selengkapnya tentang AWS Wilayah yang tersedia, lihat [Tabel AWS Wilayah](#).

Saat Anda mengonfigurasi klaster Anda untuk menggunakan HSM, Amazon Redshift mengirimkan permintaan ke HSM untuk menghasilkan dan menyimpan kunci yang akan digunakan sebagai CEK. Namun, tidak seperti AWS KMS, HSM tidak mengeksport CEK ke Amazon Redshift. Sebaliknya, Amazon Redshift secara acak menghasilkan DEK di cluster dan meneruskannya ke HSM untuk dienkripsi oleh CEK. HSM mengembalikan DEK terenkripsi ke Amazon Redshift, di mana ia dienkripsi lebih lanjut menggunakan kunci root internal yang dihasilkan secara acak dan disimpan secara internal pada disk di jaringan terpisah dari cluster. Amazon Redshift juga memuat versi DEK yang didekripsi dalam memori di cluster sehingga DEK dapat digunakan untuk mengenkripsi dan mendekripsi kunci individual untuk blok data.

Jika klaster di-boot ulang, Amazon Redshift mendekripsi DEK terenkripsi ganda yang disimpan secara internal menggunakan kunci root internal untuk mengembalikan DEK yang disimpan secara internal ke status terenkripsi CEK. DEK yang dienkripsi CEK kemudian diteruskan ke HSM untuk didekripsi dan diteruskan kembali ke Amazon Redshift, di mana ia dapat dimuat dalam memori lagi untuk digunakan dengan kunci blok data individual.

Mengonfigurasi koneksi tepercaya antara Amazon Redshift dan HSM

Ketika Anda memilih untuk menggunakan HSM untuk pengelolaan kunci klaster Anda, Anda perlu mengonfigurasi tautan jaringan tepercaya antara Amazon Redshift dan HSM Anda. Melakukan hal ini memerlukan konfigurasi sertifikat klien dan server. Koneksi tepercaya digunakan untuk meneruskan kunci enkripsi antara HSM dan Amazon Redshift selama operasi enkripsi dan dekripsi.

Amazon Redshift membuat sertifikat klien publik dari key pair private dan public key pair yang dibuat secara acak. Ini dienkripsi dan disimpan secara internal. Anda mengunduh dan mendaftarkan sertifikat klien publik di HSM Anda, dan menetakannya ke partisi HSM yang berlaku.

Anda memberikan Amazon Redshift dengan alamat IP HSM, nama partisi HSM, kata sandi partisi HSM, dan sertifikat server HSM publik, yang dienkripsi dengan menggunakan kunci root internal. Amazon Redshift menyelesaikan proses konfigurasi dan memverifikasi bahwa ia dapat terhubung ke

HSM. Jika tidak bisa, cluster dimasukkan ke dalam status `INCOMPATIBLE_HSM` dan cluster tidak dibuat. Dalam hal ini, Anda harus menghapus cluster yang tidak lengkap dan coba lagi.

Important

Ketika Anda memodifikasi klaster Anda untuk menggunakan partisi HSM yang berbeda, Amazon Redshift memverifikasi bahwa ia dapat terhubung ke partisi baru, tetapi tidak memverifikasi bahwa kunci enkripsi yang valid ada. Sebelum Anda menggunakan partisi baru, Anda harus mereplikasi kunci Anda ke partisi baru. Jika cluster dimulai ulang dan Amazon Redshift tidak dapat menemukan kunci yang valid, restart gagal. Untuk informasi selengkapnya, lihat [Mereplikasi Kunci Di Seluruh HSMs](#).

Setelah konfigurasi awal, jika Amazon Redshift gagal terhubung ke HSM, peristiwa dicatat. Untuk informasi selengkapnya tentang peristiwa ini, lihat [Pemberitahuan Acara Amazon Redshift](#).

Rotasi kunci enkripsi

Di Amazon Redshift, Anda dapat memutar kunci enkripsi untuk cluster terenkripsi. Saat Anda memulai proses rotasi kunci, Amazon Redshift memutar CEK untuk cluster yang ditentukan dan untuk snapshot cluster otomatis atau manual apa pun. Amazon Redshift juga memutar DEK untuk cluster yang ditentukan, tetapi tidak dapat memutar DEK untuk snapshot saat disimpan secara internal di Amazon Simple Storage Service (Amazon S3) dan dienkripsi menggunakan DEK yang ada.

Sementara rotasi sedang berlangsung, cluster dimasukkan ke dalam status `ROTATING_KEYS` sampai selesai, pada saat itu cluster kembali ke status `AVAILABLE`. Amazon Redshift menangani dekripsi dan enkripsi ulang selama proses rotasi kunci.

Note

Anda tidak dapat memutar tombol untuk snapshot tanpa cluster sumber. Sebelum Anda menghapus cluster, pertimbangkan apakah snapshot-nya bergantung pada rotasi tombol.

Karena klaster sesaat tidak tersedia selama proses rotasi kunci, Anda harus memutar kunci hanya sesering yang dibutuhkan data Anda atau ketika Anda mencurigai kunci mungkin telah disusupi. Sebagai praktik terbaik, Anda harus meninjau jenis data yang Anda simpan dan merencanakan seberapa sering memutar kunci yang mengenkripsi data tersebut. Frekuensi untuk memutar

kunci bervariasi tergantung pada kebijakan perusahaan Anda untuk keamanan data, dan standar industri apa pun mengenai data sensitif dan kepatuhan terhadap peraturan. Pastikan paket Anda menyeimbangkan kebutuhan keamanan dengan pertimbangan ketersediaan untuk kluster Anda.

Untuk informasi selengkapnya tentang tombol putar, lihat [Memutar kunci enkripsi](#).

Mengubah enkripsi cluster

Anda dapat memodifikasi kluster yang tidak terenkripsi untuk menggunakan enkripsi AWS Key Management Service (AWS KMS) menggunakan kunci yang AWS miliki atau kunci yang dikelola pelanggan. Saat Anda memodifikasi kluster untuk mengaktifkan AWS KMS enkripsi, Amazon Redshift secara otomatis memigrasikan data Anda ke kluster terenkripsi baru. Anda juga dapat memigrasikan kluster terenkripsi ke cluster yang tidak terenkripsi dengan memodifikasi cluster dengan, tetapi tidak dengan cluster. AWS CLI Konsol Manajemen AWS

Selama operasi migrasi, kluster Anda tersedia dalam mode hanya-baca, dan status kluster muncul sebagai perubahan ukuran.

Jika kluster Anda dikonfigurasi untuk mengaktifkan salinan snapshot lintas AWS Wilayah, Anda harus menonaktifkannya sebelum mengubah enkripsi. Untuk informasi selengkapnya, lihat [Menyalin snapshot ke Wilayah lain AWS](#) dan [Mengonfigurasi salinan snapshot lintas wilayah untuk kluster — terenkripsi AWS KMS](#). Anda tidak dapat mengaktifkan enkripsi modul keamanan perangkat keras (HSM) dengan memodifikasi cluster. Sebagai gantinya, buat cluster baru yang dienkripsi HSM dan migrasi data Anda ke cluster baru. Untuk informasi selengkapnya, lihat [Migrasi ke cluster terenkripsi HSM](#).

Amazon Redshift console

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster, lalu pilih cluster yang ingin Anda ubah enkripsi.
3. Pilih Properti.
4. Di bagian Konfigurasi basis data, pilih Edit, lalu pilih Edit enkripsi.
5. Pilih salah satu opsi enkripsi dan pilih Simpan perubahan.

AWS CLI

Untuk memodifikasi cluster yang tidak terenkripsi untuk digunakan AWS KMS, jalankan perintah `modify-cluster` CLI dan tentukan `--encrypted`, seperti yang ditunjukkan berikut. Secara

default, kunci KMS default Anda digunakan. Untuk menentukan kunci yang dikelola pelanggan, sertakan `--kms-key-id` opsi.

```
aws redshift modify-cluster --cluster-identifier <value> --encrypted --kms-key-id <value>
```

Untuk menghapus enkripsi dari cluster Anda, jalankan perintah CLI berikut.

```
aws redshift modify-cluster --cluster-identifier <value> --no-encrypted
```

Migrasi ke cluster terenkripsi HSM

Untuk memigrasikan klaster yang tidak terenkripsi ke kluster yang dienkripsi menggunakan modul keamanan perangkat keras (HSM), Anda membuat klaster terenkripsi baru dan memindahkan data Anda ke klaster baru. Anda tidak dapat bermigrasi ke klaster terenkripsi HSM dengan memodifikasi klaster.

Untuk bermigrasi dari klaster yang tidak terenkripsi ke klaster terenkripsi HSM, pertama-tama Anda membongkar data Anda dari kluster sumber yang ada. Kemudian Anda memuat ulang data dalam cluster target baru dengan pengaturan enkripsi yang dipilih. Untuk informasi selengkapnya tentang meluncurkan cluster terenkripsi, lihat [Enkripsi basis data Amazon Redshift](#)

Selama proses migrasi, kluster sumber Anda tersedia untuk kueri hanya-baca hingga langkah terakhir. Langkah terakhir adalah mengganti nama kluster target dan sumber, yang mengalihkan titik akhir sehingga semua lalu lintas diarahkan ke cluster target yang baru. Cluster target tidak tersedia sampai Anda reboot mengikuti penggantian nama. Tangguhkan semua beban data dan operasi penulisan lainnya di cluster sumber saat data sedang ditransfer.

Untuk mempersiapkan migrasi

1. Identifikasi semua sistem dependen yang berinteraksi dengan Amazon Redshift, misalnya alat intelijen bisnis (BI) dan mengekstrak, mengubah, dan memuat (ETL) sistem.
2. Identifikasi kueri validasi untuk menguji migrasi.

Misalnya, Anda dapat menggunakan kueri berikut untuk menemukan jumlah tabel yang ditentukan pengguna.

```
select count(*)
```



```
from pg_table_def
where schemaname != 'pg_catalog';
```

Query berikut mengembalikan daftar semua tabel yang ditentukan pengguna dan jumlah baris di setiap tabel.

```
select "table", tbl_rows
from svv_table_info;
```

3. Pilih waktu yang tepat untuk migrasi Anda. Untuk menemukan waktu ketika penggunaan cluster terendah, pantau metrik cluster seperti pemanfaatan CPU dan jumlah koneksi database. Untuk informasi selengkapnya, lihat [Melihat data kinerja cluster](#).
4. Jatuhkan tabel yang tidak digunakan.

Untuk membuat daftar tabel dan berapa kali setiap tabel telah ditanyakan, jalankan kueri berikut.

```
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. Luncurkan cluster baru yang dienkripsi.

Gunakan nomor port yang sama untuk cluster target seperti untuk cluster sumber. Untuk informasi selengkapnya tentang meluncurkan cluster terenkripsi, lihat. [Enkripsi basis data Amazon Redshift](#)

6. Atur proses bongkar muat dan muat.

Anda dapat menggunakan [Amazon Redshift Unload/Copy Utility](#) untuk membantu memigrasikan data antar cluster. Utilitas mengekspor data dari cluster sumber ke lokasi di Amazon S3. Data dienkripsi dengan AWS KMS Utilitas kemudian secara otomatis mengimpor data ke target. Secara opsional, Anda dapat menggunakan utilitas untuk membersihkan Amazon S3 setelah migrasi selesai.

7. Jalankan tes untuk memverifikasi proses Anda dan memperkirakan berapa lama operasi penulisan harus ditangguhkan.

Selama operasi bongkar muat, pertahankan konsistensi data dengan menangguhkan beban data dan operasi penulisan lainnya. Menggunakan salah satu tabel terbesar Anda, jalankan proses bongkar muat untuk membantu Anda memperkirakan waktu.

8. Buat objek database, seperti skema, tampilan, dan tabel. Untuk membantu Anda menghasilkan pernyataan bahasa definisi data (DDL) yang diperlukan, Anda dapat menggunakan skrip di [AdminViews](#) dalam repositori. AWS GitHub

Untuk memigrasikan klaster

1. Hentikan semua proses ETL pada cluster sumber.

Untuk mengonfirmasi bahwa tidak ada operasi penulisan dalam proses, gunakan Amazon Redshift Management Console untuk memantau penulisan IOPS. Untuk informasi selengkapnya, lihat [Melihat data kinerja cluster](#).

2. Jalankan kueri validasi yang Anda identifikasi sebelumnya untuk mengumpulkan informasi tentang kluster sumber yang tidak terenkripsi sebelum migrasi.
3. (Opsional) Buat satu antrian manajemen beban kerja (WLM) untuk menggunakan sumber daya maksimum yang tersedia di cluster sumber dan target. Misalnya, buat antrian bernama `data_migrate` dan konfigurasi antrian dengan memori 95 persen dan konkurensi 4. Untuk informasi selengkapnya, lihat [Perutean Kueri ke Antrian Berdasarkan Grup Pengguna dan Grup Kueri](#) di Panduan Pengembang Database Amazon Redshift.
4. Menggunakan `data_migrate` antrian, jalankan file. `UnloadCopyUtility`

Pantau proses UNLOAD dan COPY menggunakan Amazon Redshift Console.

5. Jalankan kueri validasi lagi dan verifikasi bahwa hasilnya cocok dengan hasil dari cluster sumber.

6. Ganti nama sumber dan kluster target Anda untuk menukar titik akhir. Untuk menghindari gangguan, lakukan operasi ini di luar jam kerja.
7. Verifikasi bahwa Anda dapat terhubung ke cluster target menggunakan semua klien SQL Anda, seperti ETL dan alat pelaporan.
8. Matikan cluster sumber yang tidak terenkripsi.

Memutar kunci enkripsi

Anda dapat menggunakan prosedur berikut untuk memutar kunci enkripsi dengan menggunakan konsol Amazon Redshift.

Untuk memutar kunci enkripsi untuk sebuah cluster

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster, lalu pilih cluster yang ingin Anda perbarui kunci enkripsi.
3. Untuk Tindakan, pilih Putar enkripsi untuk menampilkan halaman Putar kunci enkripsi.
4. Pada halaman Putar kunci enkripsi, pilih Putar kunci enkripsi.

Enkripsi saat bergerak

Anda dapat mengonfigurasi lingkungan Anda untuk melindungi kerahasiaan dan integritas data dalam perjalanan.

Rincian berikut berlaku untuk enkripsi data dalam perjalanan antara kluster Amazon Redshift dan klien SQL melalui JDBC/ODBC:

- Anda dapat terhubung ke cluster Amazon Redshift dari alat klien SQL melalui koneksi Java Database Connectivity (JDBC) dan Open Database Connectivity (ODBC).
- Amazon Redshift mendukung koneksi Secure Sockets Layer (SSL) untuk mengenkripsi data dan sertifikat server untuk memvalidasi sertifikat server yang terhubung dengan klien. Klien terhubung ke node pemimpin cluster Amazon Redshift. Untuk informasi selengkapnya, lihat [Mengkonfigurasi opsi keamanan untuk koneksi](#).
- Untuk mendukung koneksi SSL, Amazon Redshift membuat dan AWS Certificate Manager menginstal (ACM) mengeluarkan sertifikat di setiap kluster. Untuk informasi selengkapnya, lihat [Transisi ke sertifikat ACM untuk koneksi SSL](#).

- Untuk melindungi data Anda saat transit di dalam AWS Cloud, Amazon Redshift menggunakan SSL yang dipercepat perangkat keras untuk berkomunikasi dengan Amazon S3 atau Amazon DynamoDB untuk operasi COPY, UNLOAD, backup, dan restore.

Detail berikut berlaku untuk enkripsi data dalam perjalanan antara cluster Amazon Redshift dan Amazon S3 atau DynamoDB:

- Amazon Redshift menggunakan SSL yang dipercepat perangkat keras untuk berkomunikasi dengan Amazon S3 atau DynamoDB untuk operasi COPY, UNLOAD, backup, dan restore.
- Redshift Spectrum mendukung enkripsi sisi server (SSE) Amazon S3 menggunakan kunci default akun Anda yang dikelola oleh (KMS). AWS Key Management Service
- Anda dapat mengenkripsi beban Amazon Redshift dengan Amazon S3 dan. AWS KMS Untuk informasi selengkapnya, lihat [Mengekripsi Beban Amazon Redshift Anda dengan Amazon S3](#) dan. AWS KMS

Detail berikut berlaku untuk enkripsi dan penandatanganan data dalam transit antara AWS CLI, SDK, atau klien API dan titik akhir Amazon Redshift:

- Amazon Redshift menyediakan titik akhir HTTPS untuk mengenkripsi data dalam perjalanan.
- Untuk melindungi integritas permintaan API ke Amazon Redshift, panggilan API harus ditandatangani oleh pemanggil. Panggilan ditandatangani oleh sertifikat X.509 atau kunci akses AWS rahasia pelanggan sesuai dengan Proses Penandatanganan Versi Tanda Tangan 4 (Sigv4). Untuk informasi selengkapnya, lihat [Proses Penandatanganan Versi Tanda Tangan 4](#) di Referensi Umum AWS.
- Gunakan AWS CLI atau salah satu AWS SDKs untuk membuat permintaan AWS. Alat-alat ini secara otomatis menandatangani permintaan untuk Anda dengan kunci akses yang Anda tentukan saat Anda mengonfigurasi alat.

Rincian berikut berlaku untuk enkripsi data dalam perjalanan antara kluster Amazon Redshift dan editor kueri Amazon Redshift v2:

- Data ditransmisikan antara editor kueri v2 dan cluster Amazon Redshift melalui saluran terenkripsi TLS.

Kontrol enkripsi VPC dengan Amazon Redshift

Amazon Redshift mendukung [kontrol enkripsi VPC](#), fitur keamanan yang membantu Anda menerapkan enkripsi saat transit untuk semua lalu lintas di dalam dan di seluruh Wilayah. VPCs Dokumen ini menjelaskan cara menggunakan kontrol enkripsi VPC dengan kluster Amazon Redshift dan grup kerja tanpa server.

Kontrol enkripsi VPC menyediakan kontrol terpusat untuk memantau dan menegakkan enkripsi saat transit di dalam Anda. VPCs Ketika diaktifkan dalam mode penegakan, ini memastikan bahwa semua lalu lintas jaringan dienkripsi baik di lapisan perangkat keras (menggunakan Sistem AWS Nitro) atau pada lapisan aplikasi (menggunakan TLS/SSL).

Amazon Redshift terintegrasi dengan kontrol enkripsi VPC untuk membantu Anda memenuhi persyaratan kepatuhan untuk industri seperti layanan kesehatan (HIPAA), pemerintah (FedRAMP), dan keuangan (PCI DSS).

Cara kerja kontrol enkripsi VPC dengan Amazon Redshift

Kontrol enkripsi VPC beroperasi dalam dua mode:

- **Mode Monitor:** Memberikan visibilitas ke dalam status enkripsi arus lalu lintas dan membantu mengidentifikasi sumber daya yang memungkinkan lalu lintas tidak terenkripsi.
- **Menegakkan Mode:** Mencegah pembuatan atau penggunaan sumber daya yang memungkinkan lalu lintas tidak terenkripsi dalam VPC. Semua lalu lintas harus dienkripsi baik pada lapisan perangkat keras (instance berbasis Nitro) atau lapisan aplikasi (TLS/SSL).

Persyaratan untuk menggunakan kontrol enkripsi VPC

Persyaratan jenis instans

Amazon Redshift memerlukan instans berbasis Nitro untuk mendukung kontrol enkripsi VPC. Semua jenis instans Redshift modern mendukung kemampuan enkripsi yang diperlukan.

Persyaratan SSL/TLS

Ketika kontrol enkripsi VPC diaktifkan dalam mode enforce, parameter `require_ssl` harus disetel ke `true` dan tidak dapat dinonaktifkan. Ini memastikan bahwa semua koneksi klien menggunakan koneksi TLS terenkripsi.

Migrasi ke kontrol enkripsi VPC

Untuk cluster dan kelompok kerja yang ada

Anda tidak dapat mengaktifkan kontrol enkripsi VPC dalam mode penerapan pada VPC yang berisi klaster Redshift atau grup kerja tanpa server yang ada. Lihat langkah-langkah berikut untuk menggunakan kontrol enkripsi jika Anda memiliki klaster atau grup kerja yang ada:

1. Buat snapshot dari cluster atau namespace yang ada
2. Buat VPC baru dengan kontrol enkripsi VPC yang diaktifkan dalam mode penegakkan
3. Kembalikan dari snapshot ke VPC baru menggunakan salah satu operasi ini:
 - Untuk cluster yang disediakan: Gunakan operasi `restore-from-cluster-snapshot`
 - Untuk tanpa server: Gunakan `restore-from-snapshot` operasi di workgroup Anda

Saat membuat cluster atau workgroup baru di VPC dengan kontrol enkripsi diaktifkan, parameter `require_ssl` harus disetel ke `true`.

Amazon Redshift memerlukan instans berbasis Nitro untuk mendukung kontrol enkripsi VPC. Semua jenis instans Redshift modern mendukung kemampuan enkripsi yang diperlukan.

Persyaratan SSL/TLS

Ketika kontrol enkripsi VPC diaktifkan dalam mode `enforce`, parameter `require_ssl` harus disetel ke `true` dan tidak dapat dinonaktifkan. Ini memastikan bahwa semua koneksi klien menggunakan koneksi TLS terenkripsi.

Pertimbangan dan batasan

Saat menggunakan kontrol enkripsi VPC di Amazon Redshift, pertimbangkan hal berikut:

Pembatasan Negara VPC

- Pembuatan cluster dan workgroup diblokir saat kontrol enkripsi VPC dalam status `enforce-in-progress`
- Anda harus menunggu hingga VPC mencapai `enforce` mode sebelum membuat sumber daya baru

Konfigurasi SSL

- parameter `require_ssl`: Harus selalu `true` untuk cluster dan grup kerja yang dibuat dalam enkripsi yang diberlakukan VPCs
- Setelah klaster atau grup kerja dibuat dalam `require_ssl` VPC yang diberlakukan enkripsi, tidak dapat dinonaktifkan selama masa pakainya

Ketersediaan Wilayah

Fitur ini tidak tersedia dalam mode penegakan dengan Amazon Redshift Tanpa Server di Wilayah berikut:

- Amerika Selatan (Sao Paulo)
- Europe (Zurich)

Manajemen kunci

Anda dapat mengonfigurasi lingkungan Anda untuk melindungi data dengan kunci:

- Amazon Redshift secara otomatis terintegrasi dengan AWS Key Management Service (AWS KMS) untuk manajemen kunci. AWS KMS menggunakan enkripsi amplop. Untuk informasi selengkapnya, lihat [Enkripsi Amplop](#).
- Saat kunci enkripsi dikelola AWS KMS, Amazon Redshift menggunakan arsitektur berbasis kunci empat tingkat untuk enkripsi. Arsitektur terdiri dari kunci enkripsi data AES-256 yang dihasilkan secara acak, kunci database, kunci cluster, dan kunci root. Untuk informasi selengkapnya, lihat [Cara Penggunaan Amazon Redshift](#). AWS KMS
- Anda dapat membuat kunci yang dikelola pelanggan Anda sendiri AWS KMS. Untuk informasi selengkapnya, lihat [Membuat Kunci](#).
- Anda juga dapat mengimpor materi kunci Anda sendiri untuk yang baru AWS KMS keys. Untuk informasi selengkapnya, lihat [Mengimpor Materi Utama di AWS Key Management Service \(AWS KMS\)](#).
- Amazon Redshift mendukung pengelolaan kunci enkripsi dalam modul keamanan perangkat keras eksternal (HSMs). HSM dapat berada di tempat atau bisa. AWS CloudHSM Saat Anda menggunakan HSM, Anda harus menggunakan sertifikat klien dan server untuk mengonfigurasi koneksi tepercaya antara Amazon Redshift dan HSM Anda. Amazon Redshift hanya mendukung AWS CloudHSM Classic untuk manajemen kunci. Untuk informasi selengkapnya, lihat [Enkripsi menggunakan modul keamanan perangkat keras](#). Untuk informasi tentang AWS CloudHSM, lihat [Apa itu AWS CloudHSM?](#)

- Anda dapat memutar kunci enkripsi untuk cluster terenkripsi.. Untuk informasi selengkapnya, lihat [Rotasi kunci enkripsi](#).

Tokenisasi data

Tokenisasi adalah proses mengganti nilai aktual dengan nilai buram untuk tujuan keamanan data. Aplikasi yang sensitif terhadap keamanan menggunakan tokenisasi untuk mengganti data sensitif seperti informasi identitas pribadi (PII) atau informasi kesehatan yang dilindungi (PHI) dengan token untuk mengurangi risiko keamanan. Detokenisasi membalikkan token dengan nilai aktual untuk pengguna yang berwenang dengan kebijakan keamanan yang sesuai.

Untuk integrasi dengan layanan tokenisasi pihak ketiga, Anda dapat menggunakan fungsi UDFs () yang ditentukan pengguna Amazon Redshift yang Anda buat. [AWS Lambda](#) Untuk informasi selengkapnya, lihat [Fungsi yang ditentukan pengguna Lambda](#) di Panduan Pengembang Database Amazon Redshift. Misalnya, lihat [Protegrity](#).

Amazon Redshift mengirimkan permintaan tokenisasi ke server tokenisasi yang diakses melalui REST API atau titik akhir yang telah ditentukan sebelumnya. Dua atau lebih fungsi Lambda gratis memproses permintaan tokenisasi dan detokenisasi. Untuk pemrosesan ini, Anda dapat menggunakan fungsi Lambda yang disediakan oleh penyedia tokenisasi pihak ketiga. Anda juga dapat menggunakan fungsi Lambda yang Anda daftarkan sebagai Lambda di Amazon UDFs Redshift.

Misalnya, misalkan kueri dikirimkan yang memanggil UDF tokenisasi atau detokenisasi pada kolom. Cluster Amazon Redshift menggulung baris argumen yang berlaku dan mengirimkan baris tersebut dalam batch ke fungsi Lambda secara paralel. Transfer data antara node komputasi Amazon Redshift dan Lambda dalam koneksi jaringan terpisah dan terisolasi yang tidak dapat diakses oleh klien. Fungsi Lambda meneruskan data ke titik akhir server tokenisasi. Server tokenisasi memberi token atau mendetokenisasi data seperlunya dan mengembalikannya. Fungsi Lambda kemudian mengirimkan hasilnya ke cluster Amazon Redshift untuk diproses lebih lanjut, jika perlu, dan kemudian mengembalikan hasil kueri.

Merutekan lalu lintas internetwork di Amazon Redshift

Anda dapat merutekan lalu lintas melalui rute jaringan yang dikenal dan pribadi di Amazon Redshift. Halaman ini mencakup cara merutekan lalu lintas di jaringan perusahaan dan antara sumber daya yang sama AWS Region.

Untuk merutekan lalu lintas antara Amazon Redshift dan klien dan aplikasi di jaringan perusahaan:

- Siapkan koneksi pribadi antara virtual private cloud (VPC) dan jaringan perusahaan Anda. Siapkan koneksi IPsec VPN melalui internet atau koneksi fisik pribadi menggunakan Direct Connect koneksi. Direct Connect memungkinkan Anda untuk membuat antarmuka virtual pribadi dari jaringan lokal Anda langsung ke VPC Amazon Anda, memberi Anda koneksi jaringan bandwidth tinggi pribadi antara jaringan Anda dan VPC Anda. Dengan beberapa antarmuka virtual, Anda bahkan dapat membangun konektivitas pribadi ke beberapa VPCs sambil mempertahankan isolasi jaringan. Untuk informasi selengkapnya, lihat [Apa itu AWS Site-to-Site VPN?](#) dan [Apa itu Direct Connect?](#)

Untuk merutekan lalu lintas antara cluster Amazon Redshift di bucket VPC dan Amazon S3 di Wilayah yang sama: AWS

- Siapkan titik akhir VPC pribadi Amazon S3 untuk mengakses data Amazon S3 secara pribadi dari pemuatan atau pembongkaran ETL. Untuk informasi selengkapnya, lihat [Titik Akhir untuk Amazon S3](#).
- Aktifkan “Perutean VPC yang Ditingkatkan” untuk klaster Amazon Redshift, yang menentukan titik akhir VPC Amazon S3 target. Lalu lintas yang dihasilkan oleh perintah Amazon Redshift COPY, UNLOAD, atau CREATE LIBRARY kemudian dirutekan melalui titik akhir pribadi. Lihat informasi yang lebih lengkap di [Mengaktifkan perutean VPC yang disempurnakan](#).

Manajemen identitas dan akses di Amazon Redshift

Akses ke Amazon Redshift memerlukan kredensial yang AWS dapat digunakan untuk mengautentikasi permintaan Anda. Kredensi tersebut harus memiliki izin untuk mengakses AWS sumber daya, seperti klaster Amazon Redshift. Bagian berikut memberikan detail tentang bagaimana Anda dapat menggunakan [AWS Identity and Access Management \(IAM\)](#) dan Amazon Redshift untuk membantu mengamankan sumber daya Anda dengan mengontrol siapa yang dapat mengaksesnya:

- [Otentikasi dengan identitas](#)
- [Kontrol akses](#)

Important

Topik ini berisi kumpulan praktik terbaik untuk mengelola izin, identitas, dan akses aman. Kami menyarankan Anda untuk membiasakan diri dengan praktik terbaik untuk menggunakan IAM dengan Amazon Redshift. Ini termasuk menggunakan peran IAM untuk

menerapkan izin. Memiliki pemahaman yang baik tentang bagian-bagian ini akan membantu Anda mempertahankan gudang data Amazon Redshift yang lebih aman.

Otentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensi identitas Anda. Anda harus diautentikasi sebagai Pengguna root akun AWS, pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk sebagai identitas federasi menggunakan kredensial dari sumber identitas seperti AWS IAM Identity Center (Pusat Identitas IAM), otentikasi masuk tunggal, atau kredensial. Google/Facebook Untuk informasi selengkapnya tentang cara masuk, lihat [Cara masuk ke Akun AWS Anda](#) dalam Panduan Pengguna AWS Sign-In .

Untuk akses terprogram, AWS sediakan SDK dan CLI untuk menandatangani permintaan secara kriptografis. Untuk informasi selengkapnya, lihat [AWS Signature Version 4 untuk permintaan API](#) dalam Panduan Pengguna IAM.

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang disebut pengguna Akun AWS root yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Untuk tugas yang memerlukan kredensial pengguna root, lihat [Tugas yang memerlukan kredensial pengguna root](#) dalam Panduan Pengguna IAM.

Pengguna dan grup IAM

[Pengguna IAM](#) adalah identitas dengan izin khusus untuk satu orang atau aplikasi. Sebaiknya gunakan kredensial sementara alih-alih pengguna IAM dengan kredensial jangka panjang. Untuk informasi selengkapnya, lihat [Mewajibkan pengguna manusia untuk menggunakan federasi dengan penyedia identitas untuk mengakses AWS menggunakan kredensi sementara](#) di Panduan Pengguna IAM.

[Grup IAM](#) menentukan kumpulan pengguna IAM dan mempermudah pengelolaan izin untuk pengguna dalam jumlah besar. Untuk mempelajari selengkapnya, lihat [Kasus penggunaan untuk pengguna IAM](#) dalam Panduan Pengguna IAM.

Peran IAM

[Peran IAM](#) adalah identitas dengan izin khusus yang menyediakan kredensial sementara. Anda dapat mengambil peran dengan [beralih dari pengguna ke peran IAM \(konsol\)](#) atau dengan memanggil operasi AWS CLI atau AWS API. Untuk informasi selengkapnya, lihat [Metode untuk mengambil peran](#) dalam Panduan Pengguna IAM.

Peran IAM berguna untuk akses pengguna terfederasi, izin pengguna IAM sementara, akses lintas akun, akses lintas layanan, dan aplikasi yang berjalan di Amazon EC2. Untuk informasi selengkapnya, lihat [Akses sumber daya lintas akun di IAM](#) dalam Panduan Pengguna IAM.

Note

Sesi akses maju (FAS) di Redshift hanya berlaku selama 12 jam. Setelah periode ini, setiap sesi koneksi menggunakan FAS untuk berintegrasi dengan layanan lain harus dibuat kembali.

Pencegahan "confused deputy" lintas layanan

Masalah "confused deputy" adalah masalah keamanan di mana entitas yang tidak memiliki izin untuk melakukan tindakan dapat memengaruhi entitas yang memiliki hak akses lebih tinggi untuk melakukan tindakan. Pada tahun AWS, peniruan lintas layanan dapat mengakibatkan masalah wakil yang membingungkan. Peniruan identitas lintas layanan dapat terjadi ketika satu layanan (layanan yang dipanggil) memanggil layanan lain (layanan yang dipanggil). Layanan pemanggilan dapat dimanipulasi menggunakan izinnya untuk bertindak pada sumber daya pelanggan lain dengan cara yang seharusnya tidak dilakukannya kecuali bila memiliki izin untuk mengakses. Untuk mencegah hal ini, AWS menyediakan alat yang membantu Anda melindungi data untuk semua layanan dengan principal layanan yang telah diberi akses ke sumber daya di akun Anda. Untuk informasi lebih lanjut, lihat [Pencegahan deputi kebingungan lintas layanan](#) di Panduan Pengguna IAM.

Sebaiknya gunakan kunci konteks kondisi [aws:SourceAccount](#) global [aws:SourceArn](#) dan global dalam kebijakan berbasis sumber daya untuk membatasi izin yang diberikan Amazon Redshift ke layanan lain ke sumber daya. Jika `aws:SourceArn` nilainya tidak berisi ID akun, seperti ARN bucket Amazon S3, Anda harus menggunakan kedua kunci konteks kondisi global untuk membatasi izin. Jika Anda menggunakan kunci konteks kondisi global dan nilai `aws:SourceArn` berisi ID akun, nilai `aws:SourceAccount` dan akun dalam nilai `aws:SourceArn` harus menggunakan ID akun yang sama saat digunakan dalam pernyataan kebijakan yang sama.

Contoh berikut menunjukkan kebijakan yang dapat Anda terapkan untuk membatasi masalah wakil yang membingungkan untuk Amazon Redshift.

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

Kontrol akses

Anda dapat memiliki kredensi yang valid untuk mengautentikasi permintaan Anda, tetapi kecuali Anda memiliki izin, Anda tidak dapat membuat atau mengakses sumber daya Amazon Redshift. Misalnya, Anda harus memiliki izin untuk membuat klaster Amazon Redshift, membuat snapshot, menambahkan langganan acara, dan sebagainya.

Bagian berikut menjelaskan cara mengelola izin untuk Amazon Redshift. Anda sebaiknya membaca gambaran umum terlebih dahulu.

- [Ikhtisar mengelola izin akses ke sumber daya Amazon Redshift](#)
- [Menggunakan kebijakan berbasis identitas \(kebijakan IAM\) untuk Amazon Redshift](#)

Ikhtisar mengelola izin akses ke sumber daya Amazon Redshift

Setiap AWS sumber daya dimiliki oleh AWS akun, dan izin untuk membuat atau mengakses sumber daya diatur oleh kebijakan izin. Administrator akun dapat melampirkan kebijakan izin ke identitas IAM

(yaitu, pengguna, grup, dan peran), dan beberapa layanan (seperti AWS Lambda) juga mendukung melampirkan kebijakan izin ke sumber daya.

Note

Administrator akun (atau pengguna administrator) adalah pengguna dengan hak akses administrator. Untuk informasi selengkapnya, lihat [Praktik terbaik IAM](#) dalam Panduan Pengguna IAM.

Saat memberikan izin, Anda memutuskan siapa yang mendapatkan izin, sumber daya mana yang mereka dapatkan izin, dan tindakan spesifik yang ingin Anda izinkan pada sumber daya tersebut.

Sumber daya dan operasi Amazon Redshift

Amazon Redshift menyediakan sumber daya, tindakan, dan kunci konteks kondisi khusus layanan untuk digunakan dalam kebijakan izin IAM.

Amazon Redshift, Amazon Redshift Tanpa Server, Amazon Redshift Data API, dan izin akses editor kueri Amazon Redshift v2

Saat menyiapkan [Kontrol akses](#), Anda menulis kebijakan izin yang dapat dilampirkan ke identitas IAM (kebijakan berbasis identitas). Untuk informasi referensi terperinci, lihat topik berikut di Referensi Otorisasi Layanan:

- Untuk Amazon Redshift, lihat [Tindakan, sumber daya, dan kunci kondisi untuk Amazon Redshift](#) yang menggunakan awalan. `redshift`:
- Untuk Amazon Redshift Tanpa Server, lihat [Tindakan, sumber daya, dan kunci kondisi untuk Amazon Redshift Tanpa Server yang menggunakan awalan](#). `redshift-serverless`:
- Untuk Amazon Redshift Data API, lihat [Tindakan, sumber daya, dan kunci kondisi untuk Amazon Redshift Data](#) API yang menggunakan awalan. `redshift-data`:
- Untuk editor kueri Amazon Redshift v2, lihat [Tindakan, sumber daya, dan kunci kondisi untuk AWS SQL Workbench \(editor kueri Amazon Redshift v2\)](#) yang menggunakan awalan. `sqlworkbench`:

Editor kueri v2 menyertakan tindakan khusus izin yang tidak secara langsung sesuai dengan operasi API. Tindakan ini ditunjukkan dalam Referensi Otorisasi Layanan dengan `[permission only]`.

Referensi Otorisasi Layanan berisi informasi tentang operasi API mana yang dapat digunakan dalam kebijakan IAM. Ini juga mencakup AWS sumber daya yang dapat Anda berikan izin, dan kunci kondisi yang dapat Anda sertakan untuk kontrol akses berbutir halus. Untuk informasi lebih lanjut tentang syarat, lihat [Menggunakan ketentuan kebijakan IAM untuk kontrol akses terperinci](#).

Anda menentukan tindakan di bidang Action kebijakan, nilai sumber daya di bidang Resource kebijakan, dan syarat di bidang Condition kebijakan. Untuk menentukan tindakan Amazon Redshift, gunakan `redshift:` awalan yang diikuti dengan nama operasi API (misalnya, `redshift:CreateCluster`).

Memahami kepemilikan sumber daya

Pemilik sumber daya adalah AWS akun yang membuat sumber daya. Artinya, pemilik sumber daya adalah AWS akun entitas utama (akun root, pengguna IAM, atau peran IAM) yang mengautentikasi permintaan yang membuat sumber daya. Contoh berikut menggambarkan cara kerjanya:

- Jika Anda menggunakan kredensi akun root AWS akun Anda untuk membuat cluster DB, AWS akun Anda adalah pemilik sumber daya Amazon Redshift.
- Jika Anda membuat peran IAM di AWS akun Anda dengan izin untuk membuat sumber daya Amazon Redshift, siapa pun yang dapat mengambil peran tersebut dapat membuat sumber daya Amazon Redshift. AWS Akun Anda, tempat perannya berada, memiliki sumber daya Amazon Redshift.
- Jika Anda membuat pengguna IAM di AWS akun Anda dan memberikan izin untuk membuat sumber daya Amazon Redshift kepada pengguna tersebut, pengguna dapat membuat sumber daya Amazon Redshift. Namun, AWS akun Anda, tempat pengguna berada, memiliki sumber daya Amazon Redshift. Dalam kebanyakan kasus, metode ini tidak disarankan. Sebaiknya buat peran IAM dan melampirkan izin ke peran tersebut, lalu menetapkan peran tersebut ke pengguna.

Mengelola akses ke sumber daya

Kebijakan izin menjelaskan siapa yang memiliki akses ke suatu objek. Bagian berikut menjelaskan opsi yang tersedia untuk membuat kebijakan izin.

Note

Bagian ini membahas penggunaan IAM dalam konteks Amazon Redshift. Bagian ini tidak memberikan informasi yang mendetail tentang layanan IAM. Untuk dokumentasi lengkap

IAM, lihat [Apa yang Dimaksud dengan IAM?](#) dalam Panduan Pengguna IAM. Untuk informasi tentang sintaksis dan penjelasan kebijakan IAM, lihat [AWS Referensi Kebijakan IAM](#) dalam Panduan Pengguna IAM.

Kebijakan yang terlampir pada identitas IAM disebut kebijakan (kebijakan IAM) berbasis identitas dan kebijakan yang dilampirkan pada sumber daya disebut kebijakan berbasis sumber daya. Amazon Redshift hanya mendukung kebijakan berbasis identitas (kebijakan IAM).

Kebijakan berbasis identitas (kebijakan IAM)

Anda dapat menetapkan izin dengan melampirkan kebijakan ke peran IAM dan kemudian menetapkan peran tersebut ke pengguna atau grup. Berikut ini adalah contoh kebijakan yang berisi izin untuk membuat, menghapus, memodifikasi, dan me-reboot klaster Amazon Redshift untuk akun Anda. AWS

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowManageClusters",
      "Effect": "Allow",
      "Action": [
        "redshift:CreateCluster",
        "redshift>DeleteCluster",
        "redshift:ModifyCluster",
        "redshift:RebootCluster"
      ],
      "Resource": "*"
    }
  ]
}
```

Untuk informasi selengkapnya tentang penggunaan kebijakan berbasis identitas dengan Amazon Redshift, lihat. [Menggunakan kebijakan berbasis identitas \(kebijakan IAM\) untuk Amazon Redshift](#) Untuk informasi lebih lanjut tentang pengguna, kelompok, peran, dan izin, lihat [Identitas \(Pengguna, Grup, dan Peran\)](#) dalam Panduan Pengguna IAM.

Kebijakan berbasis sumber daya

Layanan lain, seperti Amazon S3, juga mendukung kebijakan izin berbasis sumber daya. Misalnya, Anda dapat melampirkan kebijakan ke bucket S3 untuk mengelola izin akses ke bucket tersebut. Amazon Redshift tidak mendukung kebijakan berbasis sumber daya.

Menentukan elemen kebijakan: Tindakan, efek, sumber daya, dan principal

[Untuk setiap resource Amazon Redshift \(lihat Sumber daya dan operasi Amazon Redshift\), layanan mendefinisikan sekumpulan operasi API \(lihat Tindakan\).](#) Untuk memberikan izin untuk operasi API ini, Amazon Redshift mendefinisikan serangkaian tindakan yang dapat Anda tentukan dalam kebijakan. Melakukan operasi API akan memerlukan izin untuk lebih dari satu tindakan.

Berikut ini adalah elemen-elemen kebijakan dasar:

- Sumber daya – Dalam kebijakan, Anda menggunakan Amazon Resource Name (ARN) untuk mengidentifikasi sumber daya yang diatur kebijakan. Untuk informasi selengkapnya, lihat [Sumber daya dan operasi Amazon Redshift](#).
- Tindakan – Anda menggunakan kata kunci tindakan untuk mengidentifikasi operasi sumber daya yang ingin Anda izinkan atau tolak. Misalnya, `redshift:DescribeClusters` izin memungkinkan izin pengguna untuk melakukan operasi Amazon DescribeClusters Redshift.
- Efek – Anda menentukan efek ketika pengguna meminta tindakan tertentu—efek ini dapat berupa pemberian izin atau penolakan. Jika Anda tidak secara eksplisit memberikan akses ke (mengizinkan) sumber daya, akses akan ditolak secara implisit. Anda juga dapat secara eksplisit menolak akses ke sumber daya, yang mungkin Anda lakukan untuk memastikan bahwa pengguna tidak dapat mengaksesnya, meskipun kebijakan yang berbeda memberikan akses.
- Principal – Dalam kebijakan berbasis identitas (Kebijakan IAM), pengguna yang kebijakannya terlampir adalah principal yang implisit. Untuk kebijakan berbasis sumber daya, Anda menentukan pengguna, akun, layanan, atau entitas lain yang diinginkan untuk menerima izin (berlaku hanya untuk kebijakan berbasis sumber daya). Amazon Redshift tidak mendukung kebijakan berbasis sumber daya.

Untuk mempelajari lebih lanjut tentang sintaks dan deskripsi kebijakan IAM, lihat [referensi kebijakan AWS IAM](#) di Panduan Pengguna IAM.

Untuk tabel yang menunjukkan semua tindakan Amazon Redshift API dan resource yang diterapkan, lihat. [Amazon Redshift, Amazon Redshift Tanpa Server, Amazon Redshift Data API, dan izin akses editor kueri Amazon Redshift v2](#)

Menentukan kondisi dalam kebijakan

Ketika Anda memberikan izin, Anda dapat menggunakan bahasa kebijakan akses untuk menentukan syarat kapan kebijakan akan berlaku. Misalnya, Anda mungkin ingin kebijakan diterapkan hanya setelah tanggal tertentu. Untuk informasi selengkapnya tentang menentukan kondisi dalam bahasa kebijakan akses, lihat [elemen kebijakan IAM JSON: Kondisi](#) dalam Panduan Pengguna IAM.

Untuk mengidentifikasi kondisi di mana kebijakan izin berlaku, sertakan `Condition` elemen dalam kebijakan izin IAM Anda. Misalnya, Anda dapat membuat kebijakan yang mengizinkan pengguna membuat klaster menggunakan `redshift:CreateCluster` tindakan, dan Anda dapat menambahkan `Condition` elemen untuk membatasi pengguna tersebut agar hanya membuat klaster di wilayah tertentu. Lihat perinciannya di [Menggunakan ketentuan kebijakan IAM untuk kontrol akses terperinci](#). Untuk daftar yang menampilkan semua nilai kunci kondisi dan tindakan serta sumber daya Amazon Redshift yang diterapkan, lihat [Amazon Redshift, Amazon Redshift Tanpa Server, Amazon Redshift Data API, dan izin akses editor kueri Amazon Redshift v2](#)

Menggunakan ketentuan kebijakan IAM untuk kontrol akses terperinci

Di Amazon Redshift, Anda dapat menggunakan tombol kondisi untuk membatasi akses ke sumber daya berdasarkan tag untuk sumber daya tersebut. Berikut ini adalah kunci kondisi Amazon Redshift yang umum.

Kunci syarat	Deskripsi
<code>aws:RequestTag</code>	Mengharuskan pengguna untuk menyertakan kunci tag (nama) dan nilai setiap kali mereka membuat sumber daya. Untuk informasi selengkapnya, lihat aws: RequestTag di Panduan Pengguna IAM.
<code>aws:ResourceTag</code>	Membatasi akses pengguna ke sumber daya berdasarkan kunci dan nilai tag tertentu. Untuk informasi selengkapnya, lihat aws: ResourceTag di Panduan Pengguna IAM.
<code>aws:TagKeys</code>	Gunakan kunci ini untuk membandingkan kunci tanda dalam permintaan dengan kunci yang Anda sebutkan dalam kebijakan. Untuk informasi selengkapnya, lihat aws: TagKeys di Panduan Pengguna IAM.

Untuk informasi tentang tag, lihat [Menandai sumber daya di Amazon Redshift](#).

Untuk daftar tindakan API yang mendukung kunci `redshift:RequestTag` dan `redshift:ResourceTag` kondisi, lihat [Amazon Redshift](#), [Amazon Redshift Tanpa Server](#), [Amazon Redshift Data API](#), dan [izin akses editor kueri Amazon Redshift v2](#).

Kunci kondisi berikut dapat digunakan dengan tindakan Amazon Redshift `GetClusterCredentials`.

Kunci syarat	Deskripsi
<code>redshift:DurationSeconds</code>	Membatasi jumlah detik yang dapat ditentukan untuk durasi.
<code>redshift:DbName</code>	Membatasi nama database yang dapat ditentukan.
<code>redshift:DbUser</code>	Membatasi nama pengguna database yang dapat ditentukan.

Contoh 1: Membatasi akses dengan menggunakan `aws:ResourceTag` condition key

Gunakan kebijakan IAM berikut untuk mengizinkan pengguna memodifikasi kluster Amazon Redshift hanya untuk akun AWS tertentu di `us-west-2` wilayah dengan tag `environment` bernama dengan nilai tag sebesar `test`

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"
        }
      }
    }
  ]
}
```

Contoh 2: Membatasi akses dengan menggunakan aws: RequestTag condition key

Gunakan kebijakan IAM berikut untuk mengizinkan pengguna membuat kluster Amazon Redshift hanya jika perintah untuk membuat kluster menyertakan tag usage bernama dan nilai tag. production Kondisi dengan aws:TagKeys dan ForAllValues pengubah menentukan bahwa hanya kunci costcenter dan usage dapat ditentukan dalam permintaan.

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"
          ]
        }
      }
    }
  ]
}
```

Menggunakan kebijakan berbasis identitas (kebijakan IAM) untuk Amazon Redshift

Topik ini memberikan contoh kebijakan berbasis identitas di mana administrator akun dapat melampirkan kebijakan izin ke identitas IAM (yaitu, pengguna, grup, dan peran).

 Important

Kami menyarankan Anda terlebih dahulu meninjau topik pengantar yang menjelaskan konsep dasar dan opsi yang tersedia bagi Anda untuk mengelola akses ke sumber daya Amazon Redshift Anda. Untuk informasi selengkapnya, lihat [Ikhtisar mengelola izin akses ke sumber daya Amazon Redshift](#).

Berikut adalah contoh kebijakan izin. Kebijakan ini memungkinkan pengguna untuk membuat, menghapus, memodifikasi, dan me-reboot semua kluster, lalu menolak izin untuk menghapus atau memodifikasi kluster tempat pengenalan kluster dimulai dengan in dan. production AWS Region us-west-2 Akun AWS 123456789012

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"
    }
  ]
}
```

```
}  
]  
}
```

Kebijakan tersebut memiliki dua pernyataan:

- Pernyataan pertama memberikan izin bagi pengguna kepada pengguna untuk membuat, menghapus, memodifikasi, dan me-reboot cluster. Pernyataan tersebut menetapkan karakter wildcard (*) sebagai Resource nilai sehingga kebijakan tersebut berlaku untuk semua resource Amazon Redshift yang dimiliki oleh akun root. AWS
- Pernyataan kedua menolak izin untuk menghapus atau memodifikasi cluster. Pernyataan tersebut menentukan klaster Amazon Resource Name (ARN) untuk nilai Resource yang menyertakan karakter wildcard (*). Akibatnya, pernyataan ini berlaku untuk semua cluster Amazon Redshift yang dimiliki oleh AWS akun root tempat pengidentifikasi cluster dimulai. `production`

AWS kebijakan terkelola untuk Amazon Redshift

AWS mengatasi banyak kasus penggunaan umum dengan menyediakan kebijakan IAM mandiri yang dibuat dan dikelola oleh. AWS Kebijakan terkelola memberikan izin yang diperlukan untuk kasus penggunaan umum sehingga Anda tidak perlu menyelidiki izin apa yang diperlukan. Untuk informasi selengkapnya, lihat [Kebijakan terkelola AWS](#) dalam Panduan Pengguna IAM.

Anda juga dapat membuat kebijakan IAM kustom Anda sendiri untuk mengizinkan izin untuk operasi dan sumber daya Amazon Redshift API. Anda dapat melampirkan kebijakan kustom ini ke IAM role atau grup yang memerlukan izin tersebut.

Bagian berikut menjelaskan kebijakan AWS terkelola, yang dapat Anda lampirkan ke pengguna di akun Anda, dan khusus untuk Amazon Redshift.

Pembaruan Amazon Redshift ke kebijakan terkelola AWS

Lihat detail tentang pembaruan kebijakan AWS terkelola untuk Amazon Redshift sejak layanan ini mulai melacak perubahan ini. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan umpan RSS di halaman riwayat Dokumen Amazon Redshift.

Ubah	Deskripsi	Date
AmazonRedshiftFederatedAuthorization – Kebijakan baru	Amazon Redshift menambahkan ease-of-use kebijakan kebijakan baru untuk menjalankan kueri dengan Otorisasi Federasi Amazon Redshift.	November 21, 2025
AmazonRedshiftServiceLinkedRolePolicy – Pembaruan ke kebijakan yang ada	Izin untuk tindakan <code>lakeformation:GetDataAccess</code> ditambahkan ke kebijakan terkelola. Menambahkannya memberikan izin untuk mendapatkan informasi katalog federasi dari. AWS Lake Formation Ketentuan tambahan untuk tindakan <code>glue:GetCatalog</code> dan <code>glue:GetCatalogs</code> ditambahkan ke kebijakan yang dikelola.	Maret 13, 2025
AmazonRedshiftServiceLinkedRolePolicy – Pembaruan ke kebijakan yang ada	Izin untuk tindakan <code>glue:GetCatalog</code> dan <code>glue:GetCatalogs</code> ditambahkan ke kebijakan terkelola. Menambahkannya memberikan izin untuk mendapatkan informasi katalog. AWS Glue	Desember 3, 2024
AmazonRedshiftServiceLinkedRolePolicy – Pembaruan ke kebijakan yang ada	Izin untuk tindakan <code>servicequotas:GetServiceQuota</code> ditambahkan ke kebijakan terkelola.	8 Maret 2024

Ubah	Deskripsi	Date
	Ini memberikan izin untuk mengakses kuota atau batas.	
AmazonRedshiftQueryEditorV2 FullAccess – Pembaruan ke kebijakan yang ada	Izin untuk tindakan <code>redshift-serverless:ListNamespaces</code> dan <code>redshift-serverless:ListWorkgroups</code> ditambahkan ke kebijakan terkelola. Menambahkannya memberikan izin untuk mencantumkan ruang nama tanpa server dan grup kerja tanpa server di gudang data Amazon Redshift.	Februari 21, 2024
AmazonRedshiftQueryEditorV2 NoSharing – Pembaruan ke kebijakan yang ada	Izin untuk tindakan <code>redshift-serverless:ListNamespaces</code> dan <code>redshift-serverless:ListWorkgroups</code> ditambahkan ke kebijakan terkelola. Menambahkannya memberikan izin untuk mencantumkan ruang nama tanpa server dan grup kerja tanpa server di gudang data Amazon Redshift.	Februari 21, 2024

Ubah	Deskripsi	Date
AmazonRedshiftQueryEditorV2 ReadSharing – Pembaruan ke kebijakan yang ada	Izin untuk tindakan <code>redshift-serverless:ListNamespaces</code> dan <code>redshift-serverless:ListWorkgroups</code> ditambahkan ke kebijakan terkelola. Menambahkannya memberikan izin untuk mencantumkan ruang nama tanpa server dan grup kerja tanpa server di gudang data Amazon Redshift.	Februari 21, 2024
AmazonRedshiftQueryEditorV2 ReadWriteSharing – Pembaruan ke kebijakan yang ada	Izin untuk tindakan <code>redshift-serverless:ListNamespaces</code> dan <code>redshift-serverless:ListWorkgroups</code> ditambahkan ke kebijakan terkelola. Menambahkannya memberikan izin untuk mencantumkan ruang nama tanpa server dan grup kerja tanpa server di gudang data Amazon Redshift.	Februari 21, 2024
AmazonRedshiftReadOnlyAccess – Pembaruan ke kebijakan yang ada	Izin untuk tindakan <code>redshift:ListRecommendations</code> ditambahkan ke kebijakan terkelola. Ini memberikan izin untuk mencantumkan rekomendasi Amazon Redshift Advisor.	Februari 7, 2024

Ubah	Deskripsi	Date
AmazonRedshiftServiceLinkedRolePolicy – Pembaruan ke kebijakan yang ada	Izin untuk tindakan <code>ec2:AssignIpv6Addresses</code> dan <code>ec2:UnassignIpv6Addresses</code> ditambahkan ke kebijakan terkelola. Menambahkannya memberikan izin untuk menetapkan dan membatalkan penetapan alamat IP.	31 Oktober 2023
AmazonRedshiftQueryEditorV2 NoSharing – Pembaruan ke kebijakan yang ada	Izin untuk tindakan <code>sqlworkbench:GetAutocompletionMetadata</code> dan <code>sqlworkbench:GetAutocompletionResource</code> ditambahkan ke kebijakan terkelola. Menambahkannya memberikan izin untuk menghasilkan dan mengambil informasi database untuk penyelesaian otomatis SQL saat mengedit kueri.	16 Agustus 2023
AmazonRedshiftQueryEditorV2 ReadSharing – Pembaruan ke kebijakan yang ada	Izin untuk tindakan <code>sqlworkbench:GetAutocompletionMetadata</code> dan <code>sqlworkbench:GetAutocompletionResource</code> ditambahkan ke kebijakan terkelola. Menambahkannya memberikan izin untuk menghasilkan dan mengambil informasi database untuk penyelesaian otomatis SQL saat mengedit kueri.	16 Agustus 2023

Ubah	Deskripsi	Date
AmazonRedshiftQueryEditorV2 ReadWriteSharing – Pembaruan ke kebijakan yang ada	Izin untuk tindakan <code>sqlworkbench:GetAutocompletionMetadata</code> dan <code>sqlworkbench:GetAutocompletionResource</code> ditambahkan ke kebijakan terkelola. Menambahkannya memberikan izin untuk menghasilkan dan mengambil informasi database untuk penyelesaian otomatis SQL saat mengedit kueri.	16 Agustus 2023

Ubah	Deskripsi	Date
AmazonRedshiftServiceLinkedRolePolicy – Pembaruan ke kebijakan yang ada	Izin untuk tindakan AWS Secrets Manager untuk membuat dan mengelola rahasia ditambahkan ke kebijakan terkelola. Izin yang ditambahkan adalah sebagai berikut: • <code>secretsmanager:GetRandomPassword</code> • <code>secretsmanager:DescribeSecret</code> • <code>secretsmanager:PutSecretValue</code> • <code>secretsmanager:UpdateSecret</code> • <code>secretsmanager:UpdateSecretVersionStage</code> • <code>secretsmanager:RotateSecret</code> • <code>secretsmanager:DeleteSecret</code>	Agustus 14, 2023

Ubah	Deskripsi	Date
AmazonRedshiftServiceLinkedRolePolicy – Pembaruan ke kebijakan yang ada	Izin untuk tindakan di Amazon EC2 untuk membuat dan mengelola grup keamanan dan aturan perutean dihapus dari kebijakan terkelola. Izin ini berkaitan dengan pembuatan subnet dan VPCs Izin yang dihapus adalah sebagai berikut: • <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>	08 Mei 2023

Ubah	Deskripsi	Date
	• <code>ec2:RevokeSecurityGroupEgress</code> • <code>ec2:ModifyVpcAttribute</code> • <code>ec2:CreateSubnet</code> • <code>ec2:CreateInternetGateway</code> • <code>ec2:CreateVpc</code> Ini terkait dengan tag Purpose: <code>RedshiftMigrateToVpc</code> resource. Tag membatasi cakupan izin untuk tugas untuk Amazon EC2 Classic ke migrasi VPC Amazon EC2. Untuk informasi selengkapnya tentang tag sumber daya, lihat Mengontrol akses ke AWS sumber daya menggunakan tag.	
AmazonRedshiftDataFullAccess – Pembaruan ke kebijakan yang ada	Izin untuk tindakan <code>redshift:GetClusterCredentialsWithIAM</code> ditambahkan ke kebijakan terkelola. Menambahkannya memberikan izin untuk mendapatkan kredensial sementara yang disempurnakan untuk mengakses database Amazon Redshift dengan yang ditentukan. Akun AWS	April 7, 2023

Ubah	Deskripsi	Date
AmazonRedshiftServiceLinkedRolePolicy – Pembaruan ke kebijakan yang ada	Izin untuk tindakan di Amazon EC2 untuk pembuatan dan pengelolaan aturan grup keamanan ditambahkan ke kebijakan terkelola. Grup dan aturan keamanan ini secara khusus terkait dengan tag <code>aws:RequestTag/Redshift</code> sumber daya Amazon Redshift. Ini membatasi ruang lingkup izin untuk sumber daya Amazon Redshift tertentu.	April 06, 2023
AmazonRedshiftQueryEditorV2 NoSharing – Pembaruan ke kebijakan yang ada	Izin untuk tindakan <code>sqlworkbench:GetSchemaInference</code> ditambahkan ke kebijakan terkelola. Menambahkannya memberikan izin untuk mendapatkan kolom dan tipe data yang disimpulkan dari file.	21 Maret 2023
AmazonRedshiftQueryEditorV2 ReadSharing – Pembaruan ke kebijakan yang ada	Izin untuk tindakan <code>sqlworkbench:GetSchemaInference</code> ditambahkan ke kebijakan terkelola. Menambahkannya memberikan izin untuk mendapatkan kolom dan tipe data yang disimpulkan dari file.	21 Maret 2023

Ubah	Deskripsi	Date
AmazonRedshiftQueryEditorV2 ReadWriteSharing – Pembaruan ke kebijakan yang ada	Izin untuk tindakan <code>sqlworkbench:GetSchemaInference</code> ditambahkan ke kebijakan terkelola. Menambahkannya memberikan izin untuk mendapatkan kolom dan tipe data yang disimpulkan dari file.	21 Maret 2023
AmazonRedshiftQueryEditorV2 NoSharing – Pembaruan ke kebijakan yang ada	Izin untuk tindakan <code>sqlworkbench:AssociateNotebookWithTab</code> ditambahkan ke kebijakan terkelola. Menambahkannya memberikan izin untuk membuat dan memperbarui tab yang ditautkan ke buku catatan pengguna sendiri.	2 Februari 2023
AmazonRedshiftQueryEditorV2 ReadSharing – Pembaruan ke kebijakan yang ada	Izin untuk tindakan <code>sqlworkbench:AssociateNotebookWithTab</code> ditambahkan ke kebijakan terkelola. Menambahkannya memberikan izin untuk membuat dan memperbarui tab yang ditautkan ke buku catatan pengguna sendiri atau ke buku catatan yang dibagikan dengannya.	2 Februari 2023

Ubah	Deskripsi	Date
AmazonRedshiftQueryEditorV2 ReadWriteSharing – Pembaruan ke kebijakan yang ada	Izin untuk tindakan <code>sqlworkbench:AssociateNotebookWithTab</code> ditambahkan ke kebijakan terkelola. Menambahkannya memberikan izin untuk membuat dan memperbarui tab yang ditautkan ke buku catatan pengguna sendiri atau ke buku catatan yang dibagikan dengannya.	2 Februari 2023

Ubah	Deskripsi	Date
AmazonRedshiftQueryEditorV2 NoSharing – Pembaruan ke kebijakan yang ada	Untuk memberikan izin menggunakan buku catatan, Amazon Redshift menambahkan izin untuk tindakan berikut: • <code>sqlworkbench:ListNotebooks</code> • <code>sqlworkbench:CreateNotebook</code> • <code>sqlworkbench:DuplicateNotebook</code> • <code>sqlworkbench:CreateNotebookFromVersion</code> • <code>sqlworkbench:ImportNotebook</code> • <code>sqlworkbench:GetNotebook</code> • <code>sqlworkbench:UpdateNotebook</code> • <code>sqlworkbench>DeleteNotebook</code> • <code>sqlworkbench:CreateNotebookCell</code> • <code>sqlworkbench>DeleteNotebookCell</code> • <code>sqlworkbench:UpdateNotebookCellContent</code> • <code>sqlworkbench:UpdateNotebookCellLayout</code>	Oktober 17, 2022

Ubah	Deskripsi	Date
	• <code>sqlworkbench:BatchGetNotebookCell</code> • <code>sqlworkbench:ListNotebookVersions</code> • <code>sqlworkbench:CreateNotebookVersion</code> • <code>sqlworkbench:GetNotebookVersion</code> • <code>sqlworkbench>DeleteNotebookVersion</code> • <code>sqlworkbench:RestoreNotebookVersion</code> • <code>sqlworkbench:ExportNotebook</code>	

Ubah	Deskripsi	Date
AmazonRedshiftQueryEditorV2 ReadSharing – Pembaruan ke kebijakan yang ada	Untuk memberikan izin menggunakan buku catatan, Amazon Redshift menambahkan izin untuk tindakan berikut: • <code>sqlworkbench:ListNotebooks</code> • <code>sqlworkbench:CreateNotebook</code> • <code>sqlworkbench:DuplicateNotebook</code> • <code>sqlworkbench:CreateNotebookFromVersion</code> • <code>sqlworkbench:ImportNotebook</code> • <code>sqlworkbench:GetNotebook</code> • <code>sqlworkbench:UpdateNotebook</code> • <code>sqlworkbench>DeleteNotebook</code> • <code>sqlworkbench:CreateNotebookCell</code> • <code>sqlworkbench>DeleteNotebookCell</code> • <code>sqlworkbench:UpdateNotebookCellContent</code> • <code>sqlworkbench:UpdateNotebookCellLayout</code>	Oktober 17, 2022

Ubah	Deskripsi	Date
	• sqlworkbench:BatchGetNotebookCell • sqlworkbench:ListNotebookVersions • sqlworkbench:CreateNotebookVersion • sqlworkbench:GetNotebookVersion • sqlworkbench>DeleteNotebookVersion • sqlworkbench:RestoreNotebookVersion • sqlworkbench:ExportNotebook	

Ubah	Deskripsi	Date
AmazonRedshiftQueryEditorV2 ReadWriteSharing – Pembaruan ke kebijakan yang ada	Untuk memberikan izin menggunakan buku catatan, Amazon Redshift menambahkan izin untuk tindakan berikut: • <code>sqlworkbench:ListNotebooks</code> • <code>sqlworkbench:CreateNotebook</code> • <code>sqlworkbench:DuplicateNotebook</code> • <code>sqlworkbench:CreateNotebookFromVersion</code> • <code>sqlworkbench:ImportNotebook</code> • <code>sqlworkbench:GetNotebook</code> • <code>sqlworkbench:UpdateNotebook</code> • <code>sqlworkbench>DeleteNotebook</code> • <code>sqlworkbench:CreateNotebookCell</code> • <code>sqlworkbench>DeleteNotebookCell</code> • <code>sqlworkbench:UpdateNotebookCellContent</code> • <code>sqlworkbench:UpdateNotebookCellLayout</code>	Oktober 17, 2022

Ubah	Deskripsi	Date
	• <code>sqlworkbench:BatchGetNotebookCell</code> • <code>sqlworkbench:ListNotebookVersions</code> • <code>sqlworkbench:CreateNotebookVersion</code> • <code>sqlworkbench:GetNotebookVersion</code> • <code>sqlworkbench>DeleteNotebookVersion</code> • <code>sqlworkbench:RestoreNotebookVersion</code> • <code>sqlworkbench:ExportNotebook</code>	
AmazonRedshiftServiceLinkedRolePolicy – Pembaruan ke kebijakan yang ada	Amazon Redshift menambahkan namespace AWS/Redshift untuk memungkinkan metrik penerbitan. CloudWatch	7 September 2022
AmazonRedshiftQueryEditorV2 NoSharing – Pembaruan ke kebijakan yang ada	Amazon Redshift menambahkan izin ke tindakan <code>sqlworkbench:ListQueryExecutionHistory</code> dan <code>sqlworkbench:GetQueryExecutionHistory</code> . Ini memberikan izin untuk melihat riwayat kueri.	30 Agustus 2022

Ubah	Deskripsi	Date
AmazonRedshiftQueryEditorV2 ReadSharing – Pembaruan ke kebijakan yang ada	Amazon Redshift menambahkan izin ke tindakan <code>sqlworkbench:ListQueryExecutionHistory</code> dan <code>sqlworkbench:GetQueryExecutionHistory</code> . Ini memberikan izin untuk melihat riwayat kueri.	30 Agustus 2022
AmazonRedshiftQueryEditorV2 ReadWriteSharing – Pembaruan ke kebijakan yang ada	Amazon Redshift menambahkan izin ke tindakan <code>sqlworkbench:ListQueryExecutionHistory</code> dan <code>sqlworkbench:GetQueryExecutionHistory</code> . Ini memberikan izin untuk melihat riwayat kueri.	30 Agustus 2022
AmazonRedshiftFullAccess – Pembaruan ke kebijakan yang ada	Izin untuk Amazon Redshift Tanpa Server ditambahkan ke kebijakan terkelola yang ada. <code>AmazonRedshiftFullAccess</code>	22 Juli 2022

Ubah	Deskripsi	Date
AmazonRedshiftDataFullAccess – Pembaruan ke kebijakan yang ada	Amazon Redshift memperbarui kondisi pelingkupan <code>redshift-serverless:GetCredentials default aws:ResourceTag/RedshiftDataFullAccess</code> izin tag dari <code>StringEquals</code> untuk memberikan akses <code>StringLike</code> ke sumber daya yang ditandai dengan kunci <code>RedshiftDataFullAccess</code> tag dan nilai tag apa pun.	Juli 11, 2022
AmazonRedshiftDataFullAccess – Pembaruan ke kebijakan yang ada	Amazon Redshift menambahkan izin baru <code>redshift-serverless:GetCredentials</code> untuk memungkinkan kredensial sementara ke Amazon Redshift Tanpa Server.	8 Juli 2022
AmazonRedshiftQueryEditorV2NoSharing – Pembaruan ke kebijakan yang ada	Amazon Redshift menambahkan izin untuk tindakan tersebut. <code>sqlworkbench:GetAccountSettings</code> Ini memberikan izin untuk mendapatkan pengaturan akun.	15 Juni 2022

Ubah	Deskripsi	Date
AmazonRedshiftQueryEditorV2 ReadSharing – Pembaruan ke kebijakan yang ada	Amazon Redshift menambahkan izin untuk tindakan tersebut. <code>sqlworkbench:GetAccountSettings</code> Ini memberikan izin untuk mendapatkan pengaturan akun.	15 Juni 2022
AmazonRedshiftQueryEditorV2 ReadWriteSharing – Pembaruan ke kebijakan yang ada	Amazon Redshift menambahkan izin untuk tindakan tersebut. <code>sqlworkbench:GetAccountSettings</code> Ini memberikan izin untuk mendapatkan pengaturan akun.	15 Juni 2022
AmazonRedshiftServiceLinkedRolePolicy – Pembaruan ke kebijakan yang ada	Untuk mengaktifkan akses publik ke titik akhir Amazon Redshift Serverless baru, Amazon Redshift mengalokasikan dan mengaitkan alamat IP Elastic ke antarmuka jaringan Elastic titik akhir VPC di akun pelanggan. Ini dilakukan melalui izin yang diberikan melalui peran terkait layanan. Untuk mengaktifkan kasus penggunaan ini, tindakan untuk mengalokasikan dan melepaskan alamat IP Elastis ditambahkan ke peran terkait layanan Amazon Redshift Tanpa Server.	26 Mei 2022

Ubah	Deskripsi	Date
AmazonRedshiftQueryEditorV2 FullAccess – Pembaruan ke kebijakan yang ada	Izin untuk <code>tindakansqlworkbench:ListTaggedResources</code> . Ini dicakup secara khusus untuk sumber daya editor kueri Amazon Redshift v2. Pembaruan kebijakan ini memberikan hak untuk menelepon <code>tag:GetResources</code> hanya melalui editor kueri v2.	Februari 22, 2022
AmazonRedshiftQueryEditorV2 NoSharing – Pembaruan ke kebijakan yang ada	Izin untuk <code>tindakansqlworkbench:ListTaggedResources</code> . Ini dicakup secara khusus untuk sumber daya editor kueri Amazon Redshift v2. Pembaruan kebijakan ini memberikan hak untuk menelepon <code>tag:GetResources</code> hanya melalui editor kueri v2.	Februari 22, 2022
AmazonRedshiftQueryEditorV2 ReadSharing – Pembaruan ke kebijakan yang ada	Izin untuk <code>tindakansqlworkbench:ListTaggedResources</code> . Ini dicakup secara khusus untuk sumber daya editor kueri Amazon Redshift v2. Pembaruan kebijakan ini memberikan hak untuk menelepon <code>tag:GetResources</code> hanya melalui editor kueri v2.	Februari 22, 2022

Ubah	Deskripsi	Date
AmazonRedshiftQueryEditorV2 ReadWriteSharing – Pembaruan ke kebijakan yang ada	Izin untuk tindakan <code>sqlworkbench:ListTaggedResources</code> . Ini dicakup secara khusus untuk sumber daya editor kueri Amazon Redshift v2. Pembaruan kebijakan ini memberikan hak untuk menelepon <code>tag:GetResources</code> hanya melalui editor kueri v2.	Februari 22, 2022
AmazonRedshiftQueryEditorV2 ReadSharing – Pembaruan ke kebijakan yang ada	Izin untuk tindakan <code>sqlworkbench:AssociateQueryWithTab</code> ditambahkan ke kebijakan terkelola. Menambahkannya memungkinkan pelanggan untuk membuat tab editor yang ditautkan ke kueri yang dibagikan dengan mereka.	Februari 22, 2022
AmazonRedshiftServiceLinkedRolePolicy – Pembaruan ke kebijakan yang ada	Amazon Redshift menambahkan izin untuk tindakan baru guna memungkinkan pengelolaan jaringan Amazon Redshift dan sumber daya VPC.	22 November 2021

Ubah	Deskripsi	Date
AmazonRedshiftAllCommandsFullAccess – Kebijakan baru	Amazon Redshift menambahkan kebijakan baru untuk mengizinkan penggunaan peran IAM yang dibuat dari konsol Amazon Redshift dan mengaturnya sebagai default untuk kluster untuk menjalankan perintah COPY dari Amazon S3, UNLOAD, CREATE EXTERNAL SCHEMA, CREATE EXTERNAL FUNCTION, CREATE MODEL, atau CREATE LIBRARY.	18 November 2021
AmazonRedshiftServiceLinkedRolePolicy – Pembaruan ke kebijakan yang ada	Amazon Redshift menambahkan izin untuk tindakan baru guna memungkinkan pengelolaan grup CloudWatch log Amazon Redshift dan aliran log, termasuk ekspor log audit.	15 November 2021
AmazonRedshiftFullAccess – Pembaruan ke kebijakan yang ada	Amazon Redshift menambahkan izin baru untuk memungkinkan penjelasan model, DynamoDB, Redshift Spectrum, dan federasi Amazon RDS.	Oktober 07, 2021
AmazonRedshiftQueryEditorV2 FullAccess – Kebijakan baru	Amazon Redshift menambahkan kebijakan baru untuk mengizinkan akses penuh ke editor kueri Amazon Redshift v2.	24 September 2021

Ubah	Deskripsi	Date
AmazonRedshiftQueryEditorV2 NoSharing – Kebijakan baru	Amazon Redshift menambahkan kebijakan baru untuk mengizinkan penggunaan editor kueri Amazon Redshift v2 tanpa berbagi sumber daya.	24 September 2021
AmazonRedshiftQueryEditorV2 ReadSharing – Kebijakan baru	Amazon Redshift menambahkan kebijakan baru untuk mengizinkan berbagi baca dalam editor kueri Amazon Redshift v2.	24 September 2021
AmazonRedshiftQueryEditorV2 ReadWriteSharing – Kebijakan baru	Amazon Redshift menambahkan kebijakan baru untuk mengizinkan pembacaan dan pembaruan berbagi dalam editor kueri Amazon Redshift v2.	24 September 2021
AmazonRedshiftFullAccess – Pembaruan ke kebijakan yang ada	Amazon Redshift menambahkan izin baru untuk mengizinkan. <code>sagemaker:*Job*</code>	18 Agustus 2021
AmazonRedshiftDataFullAccess – Pembaruan ke kebijakan yang ada	Amazon Redshift menambahkan izin baru untuk mengizinkan. <code>AuthorizeDataShare</code>	Agustus 12, 2021
AmazonRedshiftDataFullAccess – Pembaruan ke kebijakan yang ada	Amazon Redshift menambahkan izin baru untuk mengizinkan. <code>BatchExecuteStatement</code>	27 Juli 2021
Amazon Redshift mulai melacak perubahan	Amazon Redshift mulai melacak perubahan untuk kebijakan AWS terkelolanya.	27 Juli 2021

AmazonRedshiftReadOnlyAccess

Memberikan akses hanya-baca ke semua sumber daya Amazon Redshift untuk sebuah akun. AWS

Anda dapat menemukan [AmazonRedshiftReadOnlyAccess](#) kebijakan di konsol IAM dan [AmazonRedshiftReadOnlyAccess](#) di Panduan Referensi Kebijakan AWS Terkelola.

AmazonRedshiftFullAccess

Memberikan akses penuh ke semua sumber daya Amazon Redshift untuk AWS sebuah akun. Selain itu, kebijakan ini memberikan akses penuh ke semua sumber daya Amazon Redshift Tanpa Server.

Anda dapat menemukan [AmazonRedshiftFullAccess](#) kebijakan di konsol IAM dan [AmazonRedshiftFullAccess](#) di Panduan Referensi Kebijakan AWS Terkelola.

AmazonRedshiftQueryEditor

Memberikan akses penuh ke editor kueri di konsol Amazon Redshift.

Anda dapat menemukan [AmazonRedshiftQueryEditor](#) kebijakan di konsol IAM dan [AmazonRedshiftQueryEditor](#) di Panduan Referensi Kebijakan AWS Terkelola.

AmazonRedshiftDataFullAccess

Memberikan akses penuh ke operasi Amazon Redshift Data API dan sumber daya untuk AWS akun.

Anda dapat menemukan [AmazonRedshiftDataFullAccess](#) kebijakan di konsol IAM dan [AmazonRedshiftDataFullAccess](#) di Panduan Referensi Kebijakan AWS Terkelola.

AmazonRedshiftQueryEditorV2 FullAccess

Memberikan akses penuh ke operasi dan sumber daya Amazon Redshift query editor v2. Kebijakan ini juga memberikan akses ke layanan lain yang diperlukan.

Anda dapat menemukan FullAccess kebijakan [AmazonRedshiftQueryEditorV2](#) di konsol IAM dan [AmazonRedshiftQueryEditorV2 FullAccess](#) di Panduan Referensi Kebijakan AWS Terkelola.

AmazonRedshiftQueryEditorV2 NoSharing

Memberikan kemampuan untuk bekerja dengan editor kueri Amazon Redshift v2 tanpa berbagi sumber daya. Kebijakan ini juga memberikan akses ke layanan lain yang diperlukan. Prinsipal yang menggunakan kebijakan ini tidak dapat menandai sumber dayanya (seperti kueri) untuk membagikannya dengan prinsipal lain dalam hal yang sama. Akun AWS

Anda dapat menemukan NoSharing kebijakan [AmazonRedshiftQueryEditorV2](#) di konsol IAM dan [AmazonRedshiftQueryEditorV2 NoSharing](#) di Panduan Referensi Kebijakan AWS Terkelola.

AmazonRedshiftQueryEditorV2 ReadSharing

Memberikan kemampuan untuk bekerja dengan editor kueri Amazon Redshift v2 dengan berbagi sumber daya yang terbatas. Kebijakan ini juga memberikan akses ke layanan lain yang diperlukan. Prinsipal yang menggunakan kebijakan ini dapat menandai sumber dayanya (seperti kueri) untuk membagikannya dengan prinsipal lain dalam hal yang sama. Akun AWS Prinsipal yang diberikan dapat membaca sumber daya yang dibagikan dengan timnya tetapi tidak dapat memperbaruinya.

Anda dapat menemukan ReadSharing kebijakan [AmazonRedshiftQueryEditorV2](#) di konsol IAM dan [AmazonRedshiftQueryEditorV2 ReadSharing](#) di Panduan Referensi Kebijakan AWS Terkelola.

AmazonRedshiftQueryEditorV2 ReadWriteSharing

Memberikan kemampuan untuk bekerja dengan editor kueri Amazon Redshift v2 dengan berbagi sumber daya. Kebijakan ini juga memberikan akses ke layanan lain yang diperlukan. Prinsipal yang menggunakan kebijakan ini dapat menandai sumber dayanya (seperti kueri) untuk membagikannya dengan prinsipal lain dalam hal yang sama. Akun AWS Kepala sekolah yang diberikan dapat membaca dan memperbarui sumber daya yang dibagikan dengan timnya.

Anda dapat menemukan ReadWriteSharing kebijakan [AmazonRedshiftQueryEditorV2](#) di konsol IAM dan [AmazonRedshiftQueryEditorV2 ReadWriteSharing](#) di Panduan Referensi Kebijakan AWS Terkelola.

AmazonRedshiftServiceLinkedRolePolicy

Anda tidak dapat melampirkan AmazonRedshiftServiceLinkedRolePolicy ke entitas IAM Anda. Kebijakan ini dilampirkan ke peran terkait layanan yang memungkinkan Amazon Redshift mengakses sumber daya akun. Untuk informasi selengkapnya, lihat [Menggunakan peran terkait layanan untuk Amazon Redshift](#).

Anda dapat menemukan [AmazonRedshiftServiceLinkedRolePolicy](#) kebijakan di konsol IAM dan [AmazonRedshiftServiceLinkedRolePolicy](#) di Panduan Referensi Kebijakan AWS Terkelola.

AmazonRedshiftAllCommandsFullAccess

Memberikan kemampuan untuk menggunakan peran IAM yang dibuat dari konsol Amazon Redshift dan mengaturnya sebagai default untuk cluster untuk menjalankan perintah COPY dari Amazon

S3, UNLOAD, CREATE EXTERNAL SCHEMA, CREATE EXTERNAL FUNCTION, dan CREATE MODEL. Kebijakan ini juga memberikan izin untuk menjalankan pernyataan SELECT untuk layanan terkait, seperti Amazon S3, CloudWatch Log, SageMaker Amazon AI, atau AWS Glue

Anda dapat menemukan [AmazonRedshiftAllCommandsFullAccess](#) kebijakan di konsol IAM dan [AmazonRedshiftAllCommandsFullAccess](#) di Panduan Referensi Kebijakan AWS Terkelola.

AmazonRedshiftFederatedAuthorization

Kebijakan ini menggabungkan tindakan IAM yang diperlukan untuk menjalankan kueri terhadap database Glue Data Catalog dengan Izin Federasi Amazon Redshift. Kueri tersebut melewati AWS Glue dan oleh karena itu membutuhkan tindakan Dapatkan pada objek katalog untuk menemukan objek, dan Buat, Perbarui, Ganti Nama, dan Hapus tindakan untuk memodifikasi objek. Perhatikan bahwa sumber daya dikelola oleh Amazon Redshift, oleh karena itu prinsipal juga memerlukan izin Redshift untuk menyelesaikan kueri. `glue:FederateAuthorizationaction` memungkinkan AWS Glue untuk mendelegasikan keputusan otorisasi pada objek katalog ke Amazon Redshift.

Kebijakan ini memungkinkan prinsipal untuk menjalankan kueri terhadap katalog dengan Izin Federasi Amazon Redshift, tetapi tidak mengizinkan Mendaftar dan Membatalkan Pendaftaran namespace Amazon Redshift ke Glue. AWS Lihat dokumentasi tentang Persyaratan Kebijakan IAM untuk Pengaturan Izin Federasi Amazon Redshift.

Anda dapat menemukan [AmazonRedshiftFederatedAuthorization](#) kebijakan di konsol IAM dan [AmazonRedshiftFederatedAuthorization](#) di Panduan Referensi Kebijakan AWS Terkelola.

Anda juga dapat membuat kebijakan IAM kustom Anda sendiri untuk mengizinkan izin untuk operasi dan sumber daya Amazon Redshift API. Anda dapat melampirkan kebijakan kustom ini ke IAM role atau grup yang memerlukan izin tersebut.

Izin diperlukan untuk menggunakan Redshift Spectrum

Amazon Redshift Spectrum memerlukan izin ke layanan AWS lain untuk mengakses sumber daya. Untuk detail tentang izin dalam kebijakan IAM untuk Redshift Spectrum, lihat kebijakan [IAM untuk Amazon Redshift Spectrum di Panduan Pengembang Database Amazon](#) Redshift.

Izin diperlukan untuk menggunakan konsol Amazon Redshift

Agar pengguna dapat bekerja dengan konsol Amazon Redshift, pengguna tersebut harus memiliki seperangkat izin minimum yang memungkinkan pengguna mendeskripsikan sumber daya Amazon

Redshift untuk akun mereka. AWS IZIN ini juga harus memungkinkan pengguna untuk menjelaskan informasi terkait lainnya, termasuk keamanan Amazon EC2, Amazon, CloudWatch Amazon SNS, dan informasi jaringan.

Jika Anda membuat kebijakan IAM yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana mestinya untuk pengguna dengan kebijakan IAM. Untuk memastikan bahwa pengguna tersebut masih dapat menggunakan konsol Amazon Redshift, lampirkan juga kebijakan `AmazonRedshiftReadOnlyAccess` terkelola ke pengguna. Cara melakukan ini dijelaskan dalam [AWS kebijakan terkelola untuk Amazon Redshift](#).

Untuk informasi yang dapat memberi pengguna akses ke editor kueri di konsol Amazon Redshift, lihat. [Izin diperlukan untuk menggunakan editor kueri konsol Amazon Redshift](#)

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau Amazon Redshift API.

Izin diperlukan untuk menggunakan editor kueri konsol Amazon Redshift

Agar pengguna dapat bekerja dengan editor kueri Amazon Redshift, pengguna tersebut harus memiliki set izin minimum untuk operasi Amazon Redshift dan Amazon Redshift Data API. Untuk terhubung ke database menggunakan rahasia, Anda juga harus memiliki izin Secrets Manager.

Untuk memberi pengguna akses ke editor kueri di konsol Amazon Redshift, lampirkan kebijakan `AmazonRedshiftQueryEditor` dan `AmazonRedshiftReadOnlyAccess` AWS terkelola. `AmazonRedshiftQueryEditor` kebijakan ini memungkinkan izin pengguna untuk mengambil hasil hanya pernyataan SQL mereka sendiri. Artinya, pernyataan yang diajukan sama `aws:userid` seperti yang ditunjukkan pada bagian kebijakan yang `AmazonRedshiftQueryEditor` AWS dikelola ini.

```
{
  "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:user-id}"
    }
}
```

Untuk memungkinkan pengguna mengambil hasil pernyataan SQL orang lain dalam peran IAM yang sama, buat kebijakan Anda sendiri tanpa syarat untuk membatasi akses ke pengguna saat ini. Batasi juga akses untuk mengubah kebijakan ke administrator.

Izin yang diperlukan untuk menggunakan editor kueri v2

Agar pengguna dapat bekerja dengan editor kueri Amazon Redshift v2, pengguna tersebut harus memiliki set izin minimum untuk Amazon Redshift, operasi editor kueri v2, dan AWS layanan lain seperti AWS Key Management Service, dan layanan penandaan. AWS Secrets Manager

Untuk memberi pengguna akses penuh ke editor kueri v2, lampirkan kebijakan `AmazonRedshiftQueryEditorV2FullAccess` AWS terkelola. `AmazonRedshiftQueryEditorV2FullAccess` kebijakan ini memungkinkan izin pengguna untuk berbagi sumber daya editor kueri v2, seperti kueri, dengan orang lain di tim yang sama. Untuk detail tentang cara akses ke resource editor kueri v2 dikendalikan, lihat definisi kebijakan terkelola khusus untuk editor kueri v2 di konsol IAM.

Beberapa kebijakan AWS terkelola Amazon Redshift query editor v2 menggunakan AWS tag dalam kondisi untuk cakupan akses ke sumber daya. Dalam editor kueri v2, kueri berbagi didasarkan pada kunci tag dan nilai `"aws:ResourceTag/sqlworkbench-team": "${aws:PrincipalTag/sqlworkbench-team}"` dalam kebijakan IAM yang dilampirkan pada prinsipal (peran IAM). Prinsipal yang sama Akun AWS dengan nilai tag yang sama (misalnya, `accounting-team`), berada di tim yang sama di editor kueri v2. Anda hanya dapat dikaitkan dengan satu tim pada satu waktu. Pengguna dengan izin administratif dapat mengatur tim di konsol IAM dengan memberi semua anggota tim nilai yang sama untuk tag tersebut `sqlworkbench-team`. Jika nilai tag diubah untuk pengguna IAM atau peran IAM, mungkin ada penundaan hingga perubahan tercermin dalam sumber daya bersama. `sqlworkbench-team` Jika nilai tag sumber daya (seperti kueri) diubah, sekali lagi mungkin ada penundaan hingga perubahan tercermin. Anggota tim juga harus memiliki `tag:GetResources` izin untuk berbagi.

Contoh: Untuk menambahkan **accounting-team** tag untuk peran IAM

1. Masuk ke Konsol Manajemen AWS dan buka konsol IAM di <https://console.aws.amazon.com/iam/>.

2. Di panel navigasi konsol, pilih Peran lalu pilih nama peran yang ingin Anda edit.
3. Pilih tab Tag dan kemudian pilih Tambahkan tag.
4. Tambahkan kunci tag `sqlworkbench-team` dan nilainya. `accounting-team`
5. Pilih Simpan perubahan.

Sekarang ketika prinsipal IAM (dengan peran IAM ini terlampir) berbagi kueri dengan tim, prinsipal lain dengan nilai `accounting-team` tag yang sama dapat melihat kueri.

Untuk informasi selengkapnya tentang cara melampirkan tag ke prinsipal, termasuk peran IAM dan pengguna IAM, lihat [Menandai sumber daya IAM](#) di Panduan Pengguna IAM.

Anda juga dapat mengatur tim di tingkat sesi menggunakan Penyedia Identitas (IdP). Hal ini memungkinkan beberapa pengguna yang menggunakan peran IAM yang sama untuk memiliki tim yang berbeda. Kebijakan kepercayaan peran IAM harus memungkinkan `sts:TagSession` operasi. Untuk informasi selengkapnya, lihat [Izin yang diperlukan untuk menambahkan tag sesi](#) di Panduan Pengguna IAM. Tambahkan atribut tag utama ke pernyataan SAMP yang disediakan oleh idP Anda.

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/PrincipalTag:sqlworkbench-team">
  <AttributeValue>accounting-team</AttributeValue>
</Attribute>
```

Ikuti petunjuk untuk penyedia Identitas (IdP) Anda untuk mengisi atribut SAMP dengan konten yang berasal dari direktori Anda. Untuk informasi selengkapnya tentang Penyedia identitas (IdPs) dan Amazon Redshift, lihat [Menggunakan otentikasi IAM untuk menghasilkan kredensi pengguna basis data](#) dan [Penyedia identitas dan federasi](#) di Panduan Pengguna IAM.

`sqlworkbench:CreateNotebookVersion` Memberikan izin untuk mendapatkan konten sel notebook saat ini dan membuat versi buku catatan di akun Anda. Artinya, pada saat pembuatan versi, konten notebook saat ini sama dengan konten versi. Kemudian, konten sel dalam versi tetap sama dengan notebook saat ini diperbarui. `sqlworkbench:GetNotebookVersion` Memberikan izin untuk mendapatkan versi notebook. Pengguna yang tidak memiliki `sqlworkbench:BatchGetNotebookCell` izin tetapi memiliki `sqlworkbench:CreateNotebookVersion` dan `sqlworkbench:GetNotebookVersion` izin pada notebook memiliki akses ke sel notebook dalam versi. Pengguna ini tanpa `sqlworkbench:BatchGetNotebookCell` izin masih dapat mengambil konten sel notebook dengan terlebih dahulu membuat versi dan kemudian mendapatkan versi yang dibuat ini.

Izin diperlukan untuk menggunakan penjadwal Amazon Redshift

Saat Anda menggunakan penjadwal Amazon Redshift, Anda menyiapkan peran IAM dengan hubungan kepercayaan dengan penjadwal Amazon Redshift (**scheduler.redshift.amazonaws.com**) untuk memungkinkan penjadwal menerima izin atas nama Anda. Anda juga melampirkan kebijakan (izin) ke peran untuk operasi Amazon Redshift API yang ingin Anda jadwalkan.

Contoh berikut menunjukkan dokumen kebijakan dalam format JSON untuk menyiapkan hubungan kepercayaan dengan penjadwal Amazon Redshift dan Amazon Redshift.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "scheduler.redshift.amazonaws.com",
          "redshift.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Untuk informasi selengkapnya tentang entitas trust, lihat [Membuat peran untuk mendelegasikan izin ke AWS layanan di Panduan Pengguna IAM](#).

Anda juga harus menambahkan izin untuk operasi Amazon Redshift yang ingin Anda jadwalkan.

Agar penjadwal dapat menggunakan `ResizeCluster` operasi, tambahkan izin yang mirip dengan kebijakan IAM Anda berikut ini. Tergantung pada lingkungan Anda, Anda mungkin ingin membuat kebijakan lebih ketat.

JSON

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

Untuk langkah-langkah membuat peran bagi penjadwal Amazon Redshift, lihat [Membuat peran untuk AWS layanan \(konsol\)](#) di Panduan Pengguna IAM. Buat pilihan ini saat Anda membuat peran di konsol IAM:

- Untuk Pilih layanan yang akan menggunakan peran ini: Pilih Redshift.
- Untuk Pilih kasus penggunaan Anda: Pilih Redshift - Scheduler.
- Buat atau lampirkan kebijakan ke peran yang memungkinkan operasi Amazon Redshift dijadwalkan. Pilih Buat kebijakan atau ubah peran untuk melampirkan kebijakan. Masukkan kebijakan JSON untuk operasi yang akan dijadwalkan.
- Setelah Anda membuat peran, edit Hubungan Kepercayaan peran IAM untuk menyertakan `layanaredshift.amazonaws.com`.

Peran IAM yang Anda buat memiliki entitas tepercaya `scheduler.redshift.amazonaws.com` dan `redshift.amazonaws.com`. Ini juga memiliki kebijakan terlampir yang memungkinkan tindakan Amazon Redshift API yang didukung, seperti, `"redshift:ResizeCluster"`

Izin diperlukan untuk menggunakan penjadwal Amazon EventBridge

Saat Anda menggunakan EventBridge penjadwal Amazon, Anda menyiapkan peran IAM dengan hubungan kepercayaan dengan EventBridge scheduler (**`events.amazonaws.com`**) untuk memungkinkan penjadwal menerima izin atas nama Anda. Anda juga melampirkan kebijakan (izin) ke peran untuk operasi Amazon Redshift Data API yang ingin Anda jadwalkan dan kebijakan untuk operasi Amazon EventBridge.

Anda menggunakan EventBridge penjadwal saat membuat kueri terjadwal dengan editor kueri Amazon Redshift di konsol.

Anda dapat membuat peran IAM untuk menjalankan kueri terjadwal di konsol IAM. Dalam peran IAM ini, lampirkan `AmazonEventBridgeFullAccess` dan `AmazonRedshiftDataFullAccess`.

Contoh berikut menunjukkan dokumen kebijakan dalam format JSON untuk mengatur hubungan kepercayaan dengan EventBridge penjadwal.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "events.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Untuk informasi selengkapnya tentang entitas trust, lihat [Membuat peran untuk mendelegasikan izin ke AWS layanan di Panduan Pengguna IAM](#).

Untuk langkah-langkah membuat peran bagi EventBridge penjadwal, lihat [Membuat peran untuk AWS layanan \(konsol\)](#) di Panduan Pengguna IAM. Buat pilihan ini saat Anda membuat peran di konsol IAM:

- Untuk Pilih layanan yang akan menggunakan peran ini: Pilih CloudWatch Acara.
- Untuk Pilih kasus penggunaan Anda: Pilih CloudWatch Acara.
- Lampirkan kebijakan izin berikut: `AmazonEventBridgeFullAccess` dan `AmazonRedshiftDataFullAccess`.

Peran IAM yang Anda buat memiliki entitas tepercaya. `events.amazonaws.com` Ini juga memiliki kebijakan terlampir yang memungkinkan tindakan API Data Amazon Redshift yang didukung, seperti, `"redshift-data:*"`

Izin yang diperlukan untuk menggunakan machine learning Amazon Redshift (ML)

Berikut ini, Anda dapat menemukan deskripsi izin yang diperlukan untuk menggunakan pembelajaran mesin Amazon Redshift (ML) untuk berbagai kasus penggunaan.

Agar pengguna Anda dapat menggunakan Amazon Redshift ML dengan Amazon SageMaker AI, buat peran IAM dengan kebijakan yang lebih ketat daripada default. Anda dapat menggunakan kebijakan berikut. Anda juga dapat mengubah kebijakan ini untuk memenuhi kebutuhan Anda.

Kebijakan berikut menunjukkan izin yang diperlukan untuk menjalankan SageMaker AI Autopilot dengan penjelasan model dari Amazon Redshift.

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"
            ]
        }
    }
}
}
]
}

```

Kebijakan berikut menunjukkan izin minimal penuh untuk mengizinkan akses ke Amazon DynamoDB, Redshift Spectrum, dan federasi Amazon RDS.

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"
            ]
        }
    }
}
]
}

```

Secara opsional, untuk menggunakan AWS KMS kunci enkripsi, tambahkan izin berikut ke kebijakan.

```

{
    "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>"
  ]
}

```

Untuk memungkinkan Amazon Redshift dan SageMaker AI untuk mengambil peran IAM sebelumnya untuk berinteraksi dengan layanan lain, tambahkan kebijakan kepercayaan berikut ke peran tersebut.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "redshift.amazonaws.com",
          "sagemaker.amazonaws.com",
          "forecast.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}

```

Di urutan sebelumnya, `redshift-downloads/redshift-ml/` bucket Amazon S3 adalah lokasi penyimpanan data sampel yang digunakan untuk langkah dan contoh lain. Anda dapat menghapus bucket ini jika Anda tidak perlu memuat data dari Amazon S3. Atau ganti dengan bucket Amazon S3 lain yang Anda gunakan untuk memuat data ke Amazon Redshift.

your-s3-bucket Nilai **your-account-id**, **your-role**, dan adalah ID akun, peran, dan bucket yang Anda tentukan dalam perintah CREATE MODEL.

Secara opsional, Anda dapat menggunakan bagian AWS KMS kunci dari kebijakan sampel jika Anda menentukan AWS KMS kunci untuk digunakan dengan Amazon Redshift ML. **your-kms-key** Nilai adalah kunci yang Anda gunakan sebagai bagian dari perintah CREATE MODEL Anda.

Saat Anda menentukan private virtual private cloud (VPC) untuk pekerjaan tuning hyperparameter, tambahkan izin berikut.

```
{
    "Effect": "Allow",
    "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:CreateNetworkInterfacePermission",
        "ec2:DeleteNetworkInterface",
        "ec2:DeleteNetworkInterfacePermission",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeVpcs",
        "ec2:DescribeDhcpOptions",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups"
    ]
}
```

Untuk bekerja dengan penjelasan model, pastikan Anda memiliki izin untuk memanggil operasi SageMaker AI API. Kami menyarankan Anda menggunakan kebijakan AmazonSageMakerFullAccess terkelola. Jika Anda ingin membuat peran IAM dengan kebijakan yang lebih ketat, gunakan kebijakan berikut ini.

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": "*"
        }
    ]
}
```

```
}
```

Untuk informasi selengkapnya tentang kebijakan `AmazonSageMakerFullAccess` terkelola, lihat [AmazonSageMakerFullAccess](#) di Panduan Pengembang Amazon SageMaker AI.

Jika Anda ingin membuat model Forecast, sebaiknya gunakan kebijakan `AmazonForecastFullAccess` terkelola. Jika Anda ingin menggunakan kebijakan yang lebih ketat, tambahkan kebijakan berikut ke peran IAM Anda.

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": "*"
    }
  ]
}
```

Jika Anda ingin membuat model Amazon Bedrock, sebaiknya gunakan kebijakan `AmazonBedrockFullAccess` terkelola. Jika Anda ingin menggunakan kebijakan yang lebih ketat, tambahkan kebijakan berikut ke peran IAM Anda.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "bedrock:InvokeModel",
      "Resource": [
        "*",
        "arn:aws:bedrock:us-east-1::foundation-model/*"
      ]
    }
  ]
}
```

Untuk informasi selengkapnya tentang Amazon Redshift ML, lihat [Menggunakan machine learning di Amazon Redshift](#), [CREATE MODEL](#), atau [CREATE EXTERNAL MODEL](#).

Izin untuk konsumsi streaming

Streaming ingestion bekerja dengan dua layanan. Ini adalah Kinesis Data Streams dan Amazon MSK.

Izin yang diperlukan untuk menggunakan konsumsi streaming dengan Kinesis Data Streams

Prosedur dengan contoh kebijakan terkelola tersedia di [Memulai dengan konsumsi streaming dari Amazon Kinesis Data Streams](#).

Izin diperlukan untuk menggunakan streaming ingestion dengan Amazon MSK

Prosedur dengan contoh kebijakan terkelola tersedia di [Memulai dengan konsumsi streaming dari Amazon Managed Streaming for Apache Kafka](#).

Izin yang diperlukan untuk menggunakan operasi API berbagi data

Untuk mengontrol akses ke operasi API berbagi data, gunakan kebijakan berbasis tindakan IAM. Untuk informasi tentang cara mengelola kebijakan IAM, lihat [Mengelola kebijakan IAM](#) di Panduan Pengguna IAM.

Secara khusus, anggaplah administrator cluster produser perlu menggunakan `AuthorizeDataShare` panggilan untuk mengotorisasi jalan keluar untuk datashare di luar file. Akun AWS Dalam hal ini, Anda menyiapkan kebijakan berbasis tindakan IAM untuk memberikan izin ini. Gunakan `DeauthorizeDataShare` panggilan untuk mencabut jalan keluar.

Saat menggunakan kebijakan berbasis tindakan IAM, Anda juga dapat menentukan sumber daya IAM dalam kebijakan, seperti. `DataShareARN` Berikut ini menunjukkan format dan contoh `untukDataShareARN`.

```
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
```

Anda dapat membatasi `AuthorizeDataShare` akses ke datashare tertentu dengan menentukan nama datashare dalam kebijakan IAM.

```
{
  "Statement": [
    {
      "Action": [
        "redshift:AuthorizeDataShare",
      ],
      "Resource": [
        "arn:aws:redshift:us-east-1:555555555555:datashare:86b5169f-01dc-4a6f-9fbb-e2e24359e9a8/SalesShare"
      ],
      "Effect": "Deny"
    }
  ]
}
```

Anda juga dapat membatasi kebijakan IAM untuk semua datashares yang dimiliki oleh cluster produser tertentu. Untuk melakukannya, ganti **`datashare-name`** nilai dalam kebijakan dengan wildcard atau tanda bintang. Pertahankan `namespace-guid` nilai cluster.

```
arn:aws:redshift:us-east-1:555555555555:datashare:86b5169f-01dc-4a6f-9fbb-e2e24359e9a8/*
```

Berikut ini adalah kebijakan IAM yang mencegah entitas memanggil `AuthorizeDataShare` datashares yang dimiliki oleh cluster produsen tertentu.

```
{
  "Statement": [
    {
      "Action": [
        "redshift:AuthorizeDataShare",
      ],
      "Resource": [
        "arn:aws:redshift:us-east-1:555555555555:datashare:86b5169f-01dc-4a6f-9fbb-e2e24359e9a8/*"
      ],
      "Effect": "Deny"
    }
  ]
}
```

`DataShareARN` membatasi akses berdasarkan nama datashare dan global unique ID (GUID) untuk namespace cluster yang memiliki. Ini dilakukan dengan menentukan nama sebagai tanda bintang.

Kebijakan sumber daya untuk `GetClusterCredentials`

Untuk terhubung ke database cluster menggunakan koneksi JDBC atau ODBC dengan kredensial database IAM, atau untuk memanggil tindakan secara terprogram, Anda memerlukan izin untuk memanggil `GetClusterCredentials` tindakan dengan akses ke sumber daya. `redshift:GetClusterCredentials dbuser`

Jika Anda menggunakan koneksi JDBC atau ODBC, bukan `server` dan `port` Anda dapat menentukan `cluster_id` dan `region`, tetapi untuk melakukannya kebijakan Anda harus mengizinkan `redshift:DescribeClusters` tindakan dengan akses ke sumber daya. `cluster`

Jika Anda memanggil `GetClusterCredentials` dengan parameter opsional `AutocreateDbGroups`, `DbName`, dan, pastikan juga mengizinkan tindakan dan mengizinkan akses ke sumber daya yang tercantum dalam tabel berikut.

GetClusterCredentials parameter	Tindakan	Sumber daya
Autocreate	redshift:dbuser CreateClusterUser	
DbGroups	redshift:dbgroup JoinGroup	
DbName	TA	dbname

Untuk informasi selengkapnya tentang sumber daya, lihat [Sumber daya dan operasi Amazon Redshift](#).

Anda juga dapat memasukkan ketentuan berikut dalam polis Anda:

- redshift:DurationSeconds
- redshift:DbName
- redshift:DbUser

Important

Untuk integrasi SAMP SSO, Anda mungkin diminta untuk menentukan Kebijakan IAM menggunakan variabel. `${redshift:DbUser}` Dalam kasus tersebut, kami sangat menyarankan penggunaan pernyataan kondisi yang memastikan pemanggil tidak dapat memperoleh kredensial untuk pengguna yang tidak cocok dengan userid mereka. AWS Misalnya. "StringEquals": {"aws:userid": "AIDIODR4TAW7CSEXAMPLE: \${redshift:DbUser}"}" Lihat [Contoh 8: Kebijakan IAM untuk menggunakan GetClusterCredentials](#). Untuk informasi lebih lanjut tentang kondisi, lihat [Menentukan kondisi dalam kebijakan](#)

Contoh kebijakan yang dikelola pelanggan

Di bagian ini, Anda dapat menemukan contoh kebijakan pengguna yang memberikan izin untuk berbagai tindakan Amazon Redshift. Kebijakan ini berfungsi saat Anda menggunakan Amazon Redshift API AWS SDKs, atau. AWS CLI

Note

Semua contoh menggunakan Wilayah Barat AS (Oregon) (us-west-2) dan berisi akun fiktif. IDs

Contoh 1: Izinkan pengguna akses penuh ke semua tindakan dan sumber daya Amazon Redshift

Kebijakan berikut memungkinkan akses ke semua tindakan Amazon Redshift di semua sumber daya.

JSON

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

Nilai `redshift:*` dalam Action elemen menunjukkan semua tindakan di Amazon Redshift.

Contoh 2: Tolak akses pengguna ke serangkaian tindakan Amazon Redshift

Secara default, semua izin ditolak. Namun, terkadang Anda perlu secara eksplisit menolak akses ke tindakan atau serangkaian tindakan tertentu. Kebijakan berikut memungkinkan akses ke semua tindakan Amazon Redshift dan secara eksplisit menolak akses ke tindakan Amazon Redshift di mana

nama dimulai. Delete Kebijakan ini berlaku untuk semua sumber daya Amazon Redshift di. us-west-2

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:*"
    }
  ]
}
```

Contoh 3: Izinkan pengguna untuk mengelola cluster

Kebijakan berikut memungkinkan pengguna untuk membuat, menghapus, memodifikasi, dan me-reboot semua cluster, lalu menolak izin untuk menghapus klaster mana pun di mana nama cluster dimulai. protected

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"
  }
]
}

```

Contoh 4: Izinkan pengguna untuk mengotorisasi dan mencabut akses snapshot

Kebijakan berikut memungkinkan pengguna, misalnya Pengguna A, untuk melakukan hal berikut:

- Otorisasi akses ke snapshot apa pun yang dibuat dari cluster bernama. `shared`
- Cabut akses snapshot untuk snapshot apa pun yang dibuat dari `shared` cluster tempat nama snapshot dimulai. `revokable`

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"
}
]
}

```

Jika Pengguna A telah mengizinkan Pengguna B mengakses snapshot, Pengguna B harus memiliki kebijakan seperti berikut ini untuk mengizinkan Pengguna B memulihkan klaster dari snapshot. Kebijakan berikut memungkinkan Pengguna B untuk mendeskripsikan dan memulihkan dari snapshot, dan membuat cluster. Nama cluster ini harus dimulai dengan `from-other-account`.

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"
  }
]
}

```

Contoh 5: Izinkan pengguna menyalin snapshot cluster dan memulihkan cluster dari snapshot

Kebijakan berikut memungkinkan pengguna untuk menyalin snapshot apa pun yang dibuat dari klaster bernama `big-cluster-1`, dan memulihkan snapshot apa pun tempat nama snapshot dimulai. `snapshot-for-restore`

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"
}
]
}

```

Contoh 6: Izinkan pengguna mengakses Amazon Redshift, serta tindakan serta sumber daya umum untuk layanan terkait AWS

Kebijakan contoh berikut memungkinkan akses ke semua tindakan dan sumber daya untuk Amazon Redshift, Amazon Simple Notification Service (Amazon SNS), dan Amazon CloudWatch. Ini juga memungkinkan tindakan yang ditentukan pada semua sumber daya Amazon EC2 terkait di bawah akun.

Note

Izin tingkat sumber daya tidak didukung untuk tindakan Amazon EC2 yang ditentukan dalam kebijakan contoh ini.

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": [
        "*"
    ]
}
]
```

Contoh 7: Izinkan pengguna menandai sumber daya dengan konsol Amazon Redshift

Kebijakan contoh berikut memungkinkan pengguna untuk menandai sumber daya dengan konsol Amazon Redshift menggunakan AWS Resource Groups Kebijakan ini dapat dilampirkan ke peran pengguna yang memanggil konsol Amazon Redshift baru atau asli. Untuk informasi lebih lanjut tentang penandaan, lihat [Menandai sumber daya di Amazon Redshift](#).

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "TaggingPermissions",
      "Effect": "Allow",
      "Action": [
        "redshift:DeleteTags",
        "redshift:CreateTags",
        "redshift:DescribeTags",
        "tag:UntagResources",
        "tag:TagResources"
      ],
      "Resource": "*"
    }
  ]
}
```

Contoh 8: Kebijakan IAM untuk menggunakan GetClusterCredentials

Kebijakan berikut menggunakan nilai parameter sampel ini:

- Wilayah: us-west-2
- AWS Akun: 123456789012
- Nama cluster: examplecluster

Kebijakan berikut memungkinkan `GetCredentials`, `CreateClusterUser`, dan `JoinGroup` tindakan. Kebijakan ini menggunakan kunci kondisi untuk mengizinkan `GetClusterCredentials` dan `CreateClusterUser` tindakan hanya jika ID AWS pengguna cocok `"AIDIO4TAW7CSEXAMPLE:${redshift:DbUser}@yourdomain.com"`. Akses IAM

diminta untuk "testdb" database saja. Kebijakan ini juga memungkinkan pengguna untuk bergabung dengan grup bernama "common_group".

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GetClusterCredsStatement",
      "Effect": "Allow",
      "Action": [
        "redshift:GetClusterCredentials"
      ],
      "Resource": [
        "arn:aws:redshift:us-west-2:123456789012:dbuser:examplecluster/
        ${redshift:DbUser}",
        "arn:aws:redshift:us-west-2:123456789012:dbname:examplecluster/testdb",
        "arn:aws:redshift:us-west-2:123456789012:dbgroup:examplecluster/
        common_group"
      ],
      "Condition": {
        "StringEquals": {
          "aws:userid": "AIDIODR4TAW7CSEXAMPLE:${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"
  ]
}
```

Contoh berikut menunjukkan kebijakan yang memungkinkan peran IAM untuk memanggil `GetClusterCredentials` operasi. Menentukan resource Amazon dbuser Redshift memberikan akses peran ke `temp_creds_user` nama pengguna database pada kluster bernama `examplecluster`

JSON

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "redshift:GetClusterCredentials",
    "Resource": "arn:aws:redshift:us-west-2:123456789012:dbuser:examplecluster/
temp_creds_user"
  }
}
```

Anda dapat menggunakan wildcard (*) untuk mengganti semua, atau sebagian, nama cluster, nama pengguna, dan nama grup database. Contoh berikut memungkinkan nama pengguna apa pun yang dimulai `temp_` dengan cluster apa pun di akun yang ditentukan.

Important

Pernyataan dalam contoh berikut menentukan karakter wildcard (*) sebagai bagian dari nilai sumber daya sehingga kebijakan mengizinkan sumber daya apa pun yang dimulai

dengan karakter yang ditentukan. Menggunakan karakter wildcard dalam kebijakan IAM Anda mungkin terlalu permisif. Sebagai praktik terbaik, kami sarankan untuk menggunakan kebijakan paling ketat yang layak untuk aplikasi bisnis Anda.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "redshift:GetClusterCredentials",
    "Resource": "arn:aws:redshift:us-west-2:123456789012:dbuser:*/temp_*"
  }
}
```

Contoh berikut menunjukkan kebijakan yang memungkinkan peran IAM untuk memanggil `GetClusterCredentials` operasi dengan opsi untuk secara otomatis membuat pengguna baru dan menentukan grup yang bergabung dengan pengguna saat login. `"Resource": "*"` Klausula memberikan akses peran ke sumber daya apa pun, termasuk cluster, pengguna database, atau grup pengguna.

Important

Pernyataan dalam contoh berikut menetapkan karakter wildcard (*) sebagai sumber daya untuk tindakan yang diberikan, sehingga kebijakan mengizinkan akses ke setiap pengguna cluster dan database, dan memungkinkan pembuatan pengguna apa pun. Menggunakan karakter wildcard dalam kebijakan IAM Anda mungkin terlalu permisif. Sebagai praktik terbaik, kami sarankan untuk menggunakan kebijakan paling ketat yang layak untuk aplikasi bisnis Anda.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": {
```

```
"Effect": "Allow",
"Action": [
    "redshift:GetClusterCredentials",
    "redshift:CreateClusterUser",
    "redshift:JoinGroup"
],
"Resource": "*"
}
```

Untuk informasi selengkapnya, lihat [sintaks Amazon Redshift ARN](#).

Federasi penyedia identitas asli (iDP) untuk Amazon Redshift

Mengelola identitas dan izin untuk Amazon Redshift menjadi lebih mudah dengan federasi penyedia identitas asli karena memanfaatkan penyedia identitas yang ada untuk menyederhanakan autentikasi dan mengelola izin. Ini dilakukan dengan memungkinkan untuk berbagi metadata identitas ke Redshift dari penyedia identitas Anda. Untuk iterasi pertama fitur ini, penyedia identitas yang didukung adalah [Microsoft Azure Active Directory \(Azure AD\)](#).

Untuk mengonfigurasi Amazon Redshift sehingga dapat mengautentikasi identitas dari penyedia identitas pihak ketiga, Anda mendaftarkan penyedia identitas dengan Amazon Redshift. Melakukan hal ini memungkinkan Redshift untuk mengautentikasi pengguna dan peran yang ditentukan oleh penyedia identitas. Dengan demikian Anda dapat menghindari keharusan melakukan manajemen identitas terperinci di penyedia identitas pihak ketiga Anda dan di Amazon Redshift, karena informasi identitas dibagikan.

Untuk informasi tentang penggunaan peran sesi yang ditransfer dari grup penyedia identitas (IDP), lihat [PG_GET_SESSION_ROLES](#) di Panduan Pengembang Database Amazon Redshift.

Federasi penyedia identitas asli (iDP)

Untuk menyelesaikan penyiapan awal antara penyedia identitas dan Amazon Redshift, Anda melakukan beberapa langkah: Pertama, Anda mendaftarkan Amazon Redshift sebagai aplikasi pihak ketiga dengan penyedia identitas Anda, meminta izin API yang diperlukan. Kemudian Anda membuat pengguna dan grup di penyedia identitas. Terakhir, Anda mendaftarkan penyedia identitas dengan Amazon Redshift, menggunakan pernyataan SQL, yang menetapkan parameter otentikasi yang unik untuk penyedia identitas. Sebagai bagian dari mendaftarkan penyedia identitas dengan Redshift,

Anda menetapkan namespace untuk memastikan pengguna dan peran dikelompokkan dengan benar.

Dengan penyedia identitas yang terdaftar di Amazon Redshift, komunikasi diatur antara Redshift dan penyedia identitas. Klien kemudian dapat meneruskan token dan mengautentikasi ke Redshift sebagai entitas penyedia identitas. Amazon Redshift menggunakan informasi keanggotaan grup IDP untuk memetakan ke peran Redshift. Jika pengguna sebelumnya tidak ada di Redshift, pengguna dibuat. Peran dibuat yang memetakan ke grup penyedia identitas, jika tidak ada. Administrator Amazon Redshift memberikan izin pada peran, dan pengguna dapat menjalankan kueri dan melakukan tugas database lainnya.

Langkah-langkah berikut menguraikan cara kerja federasi penyedia identitas asli, saat pengguna masuk:

1. Ketika pengguna log in menggunakan opsi iDP asli, dari klien, token penyedia identitas dikirim dari klien ke driver.
2. Pengguna diautentikasi. Jika pengguna belum ada di Amazon Redshift, pengguna baru akan dibuat. Redshift memetakan grup penyedia identitas pengguna ke peran Redshift.
3. Izin ditetapkan, berdasarkan peran Redshift pengguna. Ini diberikan kepada pengguna dan peran oleh administrator.
4. Pengguna dapat menanyakan Redshift.

Alat klien desktop

Untuk petunjuk tentang cara menggunakan federasi penyedia identitas asli untuk terhubung ke Amazon Redshift dengan Power BI, lihat posting blog [Mengintegrasikan federasi iDP asli Amazon Redshift dengan Microsoft Azure Active Directory](#) (AD) dan Power BI. Ini menjelaskan step-by-step implementasi penyiapan iDP asli Amazon Redshift dengan Azure AD. Ini merinci langkah-langkah untuk mengatur koneksi klien untuk Power BI Desktop atau layanan Power BI. Langkah-langkahnya meliputi pendaftaran aplikasi, mengonfigurasi izin, dan mengonfigurasi kredensial.

Untuk mempelajari cara mengintegrasikan federasi iDP asli Amazon Redshift dengan Azure AD, menggunakan Power BI Desktop dan JDBC Client-SQL Workbench/J, tonton video berikut:

Untuk petunjuk tentang cara menggunakan federasi penyedia identitas asli untuk terhubung ke Amazon Redshift dengan klien SQL, khususnya DBeaver atau SQL Workbench/J, lihat posting blog [Mengintegrasikan federasi iDP asli Amazon Redshift](#) dengan Microsoft Azure AD menggunakan klien SQL.

Batasan

Keterbatasan ini berlaku:

- Dukungan driver Amazon Redshift BrowserIdcAuthPlugin mulai dari versi berikut:
 - Driver Amazon Redshift JDBC v2.1.0.30
 - Driver Amazon Redshift ODBC v2.1.3
 - Driver Amazon Redshift Python v2.1.3
- Dukungan driver Amazon Redshift IdpTokenAuthPlugin mulai dari versi berikut:
 - Driver Amazon Redshift JDBC v2.1.0.19
 - Driver Amazon Redshift ODBC v2.0.0.9
 - Driver Amazon Redshift Python v2.0.914
- Tidak ada dukungan untuk VPC yang disempurnakan — VPC yang disempurnakan tidak didukung saat Anda mengonfigurasi propagasi identitas tepercaya Redshift dengan IAM Identity Center. AWS Untuk informasi selengkapnya tentang VPC yang disempurnakan, lihat Perutean [VPC yang Ditingkatkan di Amazon Redshift](#).
- AWS Caching Pusat Identitas IAM - Pusat Identitas AWS IAM menyimpan informasi sesi cache. Ini dapat menyebabkan masalah akses yang tidak dapat diprediksi saat Anda mencoba terhubung ke database Redshift Anda melalui editor kueri Redshift v2. Ini karena sesi Pusat AWS Identitas IAM terkait di editor kueri v2 tetap valid, bahkan dalam kasus di mana pengguna database keluar dari AWS konsol. Cache kedaluwarsa setelah satu jam, yang biasanya memperbaiki masalah apa pun.

Menyiapkan penyedia identitas di Amazon Redshift

Bagian ini menunjukkan langkah-langkah untuk mengonfigurasi penyedia identitas dan Amazon Redshift untuk menjalin komunikasi bagi federasi penyedia identitas asli. Anda memerlukan akun aktif dengan penyedia identitas Anda. Sebelum mengonfigurasi Amazon Redshift, Anda mendaftarkan Redshift sebagai aplikasi dengan penyedia identitas Anda, memberikan persetujuan administrator.

Selesaikan langkah-langkah berikut di Amazon Redshift:

1. Anda menjalankan pernyataan SQL untuk mendaftarkan penyedia identitas, termasuk deskripsi metadata aplikasi Azure. Untuk membuat penyedia identitas di Amazon Redshift, jalankan perintah berikut setelah mengganti nilai parameter issuer, client_id, client_secret, dan audience. Parameter ini khusus untuk Microsoft Azure AD. Ganti nama penyedia identitas dengan nama

yang Anda pilih, dan ganti namespace dengan nama unik untuk memuat pengguna dan peran dari direktori penyedia identitas Anda.

```
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"]
}'
```

Jenis ini azure menunjukkan bahwa penyedia secara khusus memfasilitasi komunikasi dengan Microsoft Azure AD. Ini adalah satu-satunya penyedia identitas pihak ketiga yang didukung.

- penerbit - ID penerbit untuk dipercaya ketika token diterima. Pengidentifikasi unik untuk tenant_id ditambahkan ke penerbit.
- client_id - Pengidentifikasi publik yang unik dari aplikasi yang terdaftar dengan penyedia identitas. Ini dapat disebut sebagai ID aplikasi.
- client_secret - Pengidentifikasi rahasia, atau kata sandi, yang hanya diketahui oleh penyedia identitas dan aplikasi terdaftar.
- audiens - ID Aplikasi yang ditugaskan ke aplikasi di Azure.

Alih-alih menggunakan rahasia klien bersama, Anda dapat mengatur parameter untuk menentukan sertifikat, kunci pribadi, dan kata sandi kunci pribadi saat membuat penyedia identitas.

```
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"}';
```

Kata sandi kunci pribadi, client_pk_password, adalah opsional.

- Opsional: Jalankan perintah SQL di Amazon Redshift untuk membuat pengguna dan peran sebelumnya. Ini memfasilitasi pemberian izin terlebih dahulu. Nama peran di Amazon Redshift adalah seperti berikut: <GroupName pada Azure <Namespace>AD>. Misalnya, ketika Anda membuat grup di Microsoft Azure AD dipanggil `rsgroup` dan namespace dipanggil `aad`, nama peran adalah `aad:rsgroup`. Nama pengguna dan peran di Amazon Redshift ditentukan dari nama pengguna dan keanggotaan grup ini di namespace penyedia identitas.

Pemetaan untuk peran dan pengguna termasuk memverifikasi `external_id` nilainya, untuk memastikannya mutakhir. ID eksternal memetakan ke pengenalan grup atau pengguna di penyedia identitas. Misalnya, ID eksternal peran memetakan ke ID grup Azure AD yang sesuai. Demikian pula, ID eksternal setiap pengguna memetakan ke ID mereka di penyedia identitas.

```
create role "aad:rsgroup";
```

- Berikan izin yang relevan untuk peran sesuai kebutuhan Anda. Contoh:

```
GRANT SELECT on all tables in schema public to role "aad:rsgroup";
```

- Anda juga dapat memberikan izin kepada pengguna tertentu.

```
GRANT SELECT on table foo to aad:alice@example.com
```

Perhatikan bahwa keanggotaan peran pengguna eksternal federasi hanya tersedia di sesi pengguna tersebut. Ini memiliki implikasi untuk membuat objek database. Ketika pengguna eksternal federasi membuat tampilan atau prosedur tersimpan, misalnya, pengguna yang sama tidak dapat mendelegasikan izin objek tersebut ke pengguna dan peran lain.

Penjelasan ruang nama

Namespace memetakan pengguna atau peran ke penyedia identitas tertentu. Misalnya, awalan untuk pengguna yang dibuat di AWS IAM adalah `iam`. Awalan ini mencegah tabrakan nama pengguna dan memungkinkan dukungan untuk beberapa penyimpanan identitas. Jika pengguna `alice@example.com` dari sumber identitas yang terdaftar dengan namespace `aad` masuk, pengguna `aad:alice@example.com` dibuat di Redshift jika belum ada. Perhatikan bahwa namespace pengguna dan peran memiliki fungsi yang berbeda dari namespace kluster Amazon Redshift, yang merupakan pengenalan unik yang terkait dengan cluster.

Membuat peran Amazon Redshift secara otomatis untuk penyedia identitas

Fitur ini memungkinkan Anda untuk secara otomatis membuat peran di Redshift berdasarkan keanggotaan grup dari Penyedia Identitas (iDP) Anda. Peran pembuatan otomatis mendukung Azure Active Directory dengan integrasi iDP asli.

Ada beberapa manfaat untuk membuat peran secara otomatis. Saat Anda membuat peran secara otomatis, Redshift membuat peran dengan keanggotaan grup di IDP Anda, sehingga Anda dapat menghindari pembuatan dan pemeliharaan peran manual yang membosankan. Anda juga memiliki opsi untuk memfilter grup mana yang dipetakan ke peran Redshift.

Cara kerjanya

Saat Anda, sebagai pengguna iDP, masuk ke Redshift, urutan peristiwa berikut terjadi:

1. Redshift mengambil keanggotaan grup Anda dari IDP.
2. Redshift secara otomatis membuat pemetaan peran ke grup tersebut, dengan format peran.
idp_namespace:rolename
3. Redshift memberi Anda izin dengan peran yang dipetakan.

Setelah setiap login pengguna, setiap grup yang tidak ada dalam katalog tetapi pengguna adalah bagian darinya, dibuat secara otomatis. Anda dapat mengatur filter include dan exclude secara opsional untuk mengontrol grup IDP mana yang memiliki peran Redshift yang dibuat.

Mengkonfigurasi peran buat otomatis

Gunakan ALTER IDENTITY PROVIDER perintah CREATE IDENTITY PROVIDER and untuk mengaktifkan dan mengkonfigurasi pembuatan peran otomatis.

```
-- 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;
```

Grup penyaringan

Anda dapat secara opsional memfilter grup IDP mana yang dipetakan ke peran INCLUDE Redshift menggunakan dan pola. EXCLUDE Ketika pola konflik, EXCLUDE lebih diutamakan. INCLUDE

```
-- 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%';
```

Contoh

Contoh berikut menunjukkan cara mengaktifkan peran buat otomatis tanpa pemfilteran.

```
CREATE IDENTITY PROVIDER prod_idc TYPE azure ...
  AUTO_CREATE_ROLES TRUE;
```

Contoh berikut mencakup kelompok pengembangan dan tidak termasuk kelompok uji.

```
ALTER IDENTITY PROVIDER prod_idc
  AUTO_CREATE_ROLES TRUE
  INCLUDE GROUPS LIKE '%dev%'
  EXCLUDE GROUPS LIKE '%test%';
```

Praktik terbaik

Pertimbangkan praktik terbaik berikut saat Anda mengaktifkan pembuatan otomatis untuk peran:

- Gunakan INCLUDE dan EXCLUDE filter untuk mengontrol grup mana yang mendapatkan peran.
- Secara berkala mengaudit peran dan membersihkan yang tidak terpakai.
- Manfaatkan hierarki peran Redshift untuk menyederhanakan manajemen izin.

Hubungkan Redshift dengan AWS IAM Identity Center untuk pengalaman masuk tunggal

Anda dapat mengelola akses pengguna dan grup ke gudang data Amazon Redshift melalui propagasi identitas terpercaya.

[Propagasi identitas terpercaya](#) adalah AWS IAM Identity Center fitur yang Layanan AWS dapat digunakan administrator terhubung untuk memberikan dan mengaudit akses ke data layanan. Akses ke data ini didasarkan pada atribut pengguna seperti asosiasi grup. Menyiapkan propagasi identitas terpercaya memerlukan kolaborasi antara administrator yang terhubung Layanan AWS dan administrator Pusat Identitas IAM. Untuk informasi lebih lanjut, lihat [Prasyarat](#) dan pertimbangan.

Untuk mengilustrasikan satu end-to-end kasus, Anda dapat menggunakan Amazon Quick dasbor atau editor kueri Amazon Redshift v2 untuk mengakses Redshift. Akses dalam hal ini didasarkan pada grup Pusat AWS Identitas IAM. Redshift dapat menentukan siapa pengguna dan keanggotaan grup mereka. AWS IAM Identity Center juga memungkinkan untuk menghubungkan dan mengelola identitas melalui penyedia identitas pihak ketiga (iDP) seperti Okta atau PingOne

Setelah administrator Anda mengatur koneksi antara Redshift dan AWS IAM Identity Center, mereka dapat mengonfigurasi akses berbutir halus berdasarkan grup penyedia identitas untuk mengotorisasi akses pengguna ke data.

Important

Saat Anda menghapus pengguna dari Pusat Identitas AWS IAM atau direktori penyedia identitas terhubung (iDP), pengguna tidak secara otomatis dihapus dari katalog Amazon Redshift. Untuk menghapus pengguna secara manual dari katalog Amazon Redshift, jalankan `DROP USER` perintah untuk menghapus sepenuhnya pengguna yang telah dihapus dari Pusat Identitas AWS IAM atau iDP. Untuk informasi selengkapnya tentang cara menjatuhkan pengguna, lihat [DROP USER di Panduan](#) Pengembang Database Amazon Redshift.

Manfaat integrasi Redshift dengan AWS IAM Identity Center

Menggunakan AWS IAM Identity Center dengan Redshift dapat menguntungkan organisasi Anda dengan cara-cara berikut:

- Penulis dasbor Amazon Quick dapat terhubung ke sumber data Redshift tanpa harus memasukkan kembali kata sandi atau mengharuskan administrator untuk mengatur peran IAM dengan izin yang kompleks.
- AWS IAM Identity Center menyediakan lokasi pusat bagi pengguna tenaga kerja Anda. AWS Anda dapat membuat pengguna dan grup secara langsung di Pusat AWS Identitas IAM atau menghubungkan pengguna dan grup yang ada yang Anda kelola di penyedia identitas berbasis standar seperti Okta, PingOne atau Microsoft Entra ID (Azure AD). AWS IAM Identity Center mengarahkan otentikasi ke sumber kebenaran yang Anda pilih untuk pengguna dan grup, dan memelihara direktori pengguna dan grup untuk diakses oleh Redshift. Untuk informasi selengkapnya, lihat [Mengelola sumber identitas Anda](#) dan [Penyedia identitas yang didukung](#) di Panduan Pengguna Pusat AWS Identitas IAM.
- Anda dapat berbagi satu instans Pusat AWS Identitas IAM dengan beberapa cluster dan grup kerja Redshift dengan kemampuan penemuan otomatis dan koneksi yang sederhana. Ini membuatnya cepat untuk menambahkan cluster tanpa upaya ekstra untuk mengonfigurasi koneksi Pusat Identitas AWS IAM untuk masing-masing, dan memastikan bahwa semua cluster dan kelompok kerja memiliki pandangan pengguna, atribut, dan grup yang konsisten. Perhatikan bahwa instans Pusat AWS Identitas IAM organisasi Anda harus berada di wilayah yang sama dengan rangkaian data Redshift yang Anda sambungkan.
- Karena identitas pengguna diketahui dan dicatat bersama dengan akses data, lebih mudah bagi Anda untuk memenuhi peraturan kepatuhan melalui audit akses pengguna. AWS CloudTrail

Persona administrator untuk menghubungkan aplikasi

Berikut ini adalah persona yang merupakan kunci untuk menghubungkan aplikasi analitik ke aplikasi terkelola AWS IAM Identity Center untuk Redshift:

- Administrator aplikasi — Membuat aplikasi dan mengonfigurasi layanan mana yang akan memungkinkan pertukaran token identitas. Administrator ini juga menentukan pengguna atau grup mana yang memiliki akses ke aplikasi.
- Administrator data — Mengkonfigurasi akses halus ke data. Pengguna dan grup di Pusat AWS Identitas IAM dapat memetakan ke izin tertentu.

Menghubungkan ke Amazon Redshift dengan AWS IAM Identity Center melalui Amazon Quick

Berikut ini menunjukkan cara menggunakan Quick untuk mengautentikasi dengan Redshift saat terhubung dan akses dikelola AWS melalui Pusat Identitas IAM: [Mengotorisasi koneksi dari cluster Quick ke Amazon Redshift](#). Langkah-langkah ini berlaku untuk Amazon Redshift Serverless juga.

Menghubungkan ke Amazon Redshift dengan AWS IAM Identity Center melalui editor kueri Amazon Redshift v2

Setelah menyelesaikan langkah-langkah untuk mengatur koneksi AWS IAM Identity Center dengan Redshift, pengguna dapat mengakses database dan objek yang sesuai dalam database melalui identitas IAM Identity Center, AWS namespace-prefixed. Untuk informasi selengkapnya tentang menghubungkan ke database Redshift dengan login editor kueri v2, lihat. [Menanyakan database menggunakan editor kueri v2](#)

Menggunakan Pusat AWS Identitas IAM di beberapa Wilayah AWS

Amazon Redshift mendukung Pusat Identitas AWS IAM dalam beberapa. Wilayah AWS Anda dapat memperluas Pusat AWS Identitas IAM dari Wilayah utama Anda AWS Region ke Wilayah tambahan untuk meningkatkan kinerja melalui kedekatan dengan pengguna dan keandalan. Saat Wilayah baru ditambahkan di Pusat Identitas AWS IAM, Anda dapat membuat aplikasi Pusat Identitas IAM Redshift di Wilayah baru tanpa mereplikasi identitas dari Wilayah utama. Anda dapat mengatur izin gabungan Amazon Redshift menggunakan Pusat Identitas AWS IAM di Wilayah baru tempat Anda dapat mengaktifkan kontrol tingkat baris, tingkat kolom, dan masking. Untuk detail selengkapnya untuk memulai Pusat Identitas AWS IAM di beberapa Wilayah, lihat [Mengelola Pusat Identitas AWS IAM dalam beberapa Wilayah AWS di Panduan Pengguna Pusat AWS Identitas IAM](#).

Batasan untuk menghubungkan ke Amazon Redshift dengan AWS IAM Identity Center

Saat menggunakan sistem masuk tunggal AWS IAM Identity Center, pertimbangkan batasan berikut:

- Tidak ada dukungan untuk VPC yang disempurnakan — VPC yang disempurnakan tidak didukung saat Anda menggunakan sistem masuk tunggal Pusat Identitas AWS IAM untuk Amazon Redshift. Untuk informasi selengkapnya tentang VPC yang disempurnakan, lihat Perutean [VPC yang Ditingkatkan di Amazon Redshift](#).

Menyiapkan integrasi Pusat AWS Identitas IAM dengan Amazon Redshift

Administrator klaster Amazon Redshift atau administrator Amazon Redshift Tanpa Server Anda harus melakukan beberapa langkah untuk mengonfigurasi Redshift sebagai aplikasi yang diaktifkan Pusat Identitas IAM. AWS Ini membuatnya sehingga Redshift dapat menemukan dan terhubung ke AWS IAM Identity Center secara otomatis untuk menerima layanan login dan direktori pengguna. Setelah ini, ketika administrator Redshift Anda membuat cluster atau workgroup, mereka dapat mengaktifkan gudang data baru untuk menggunakan AWS IAM Identity Center untuk mengelola akses database.

Tujuan mengaktifkan Redshift sebagai aplikasi AWS yang dikelola Pusat Identitas IAM adalah agar Anda dapat mengontrol izin pengguna dan grup dari AWS dalam Pusat Identitas IAM, atau dari penyedia identitas pihak ketiga yang terintegrasi dengannya. Saat pengguna database Anda masuk ke database Redshift, misalnya analis atau ilmuwan data, database akan memeriksa grup mereka di Pusat Identitas AWS IAM dan ini cocok dengan nama peran di Redshift. Dengan cara ini, grup yang mendefinisikan nama untuk peran database Redshift dapat mengakses satu set tabel untuk analisis penjualan, misalnya. Bagian berikut menunjukkan cara mengatur ini.

Prasyarat

Ini adalah prasyarat untuk mengintegrasikan AWS IAM Identity Center dengan Amazon Redshift:

- **Konfigurasi akun** — Anda harus mengonfigurasi Pusat AWS Identitas IAM di akun manajemen AWS organisasi jika Anda berencana untuk memiliki kasus penggunaan lintas akun, atau jika Anda menggunakan kluster Redshift di akun yang berbeda dengan instans Pusat Identitas IAM yang sama AWS . Ini termasuk mengonfigurasi sumber identitas Anda. Untuk informasi selengkapnya, lihat [Memulai](#), [identitas tenaga kerja](#), dan [penyedia identitas yang didukung di Panduan](#) Pengguna Pusat AWS Identitas IAM. Anda harus memastikan bahwa Anda telah membuat pengguna atau grup di Pusat Identitas AWS IAM, atau menyinkronkan pengguna dan grup dari sumber identitas Anda sebelum Anda dapat menetapkan ke data di Redshift.

Note

Anda memiliki opsi untuk menggunakan instance akun AWS IAM Identity Center, asalkan Redshift AWS dan IAM Identity Center berada di akun yang sama. Anda dapat membuat instance ini menggunakan widget saat membuat dan mengonfigurasi cluster atau workgroup Redshift.

- **Mengonfigurasi penerbit token tepercaya** — Dalam beberapa kasus, Anda mungkin perlu menggunakan penerbit token tepercaya, yang merupakan entitas yang dapat mengeluarkan

dan memverifikasi token kepercayaan. Sebelum Anda dapat melakukannya, langkah-langkah awal diperlukan sebelum administrator Redshift yang mengonfigurasi integrasi AWS IAM Identity Center dapat memilih penerbit token terpercaya dan menambahkan atribut yang diperlukan untuk menyelesaikan konfigurasi. Ini dapat mencakup mengonfigurasi penyedia identitas eksternal untuk berfungsi sebagai penerbit token terpercaya dan menambahkan atributnya di konsol Pusat Identitas AWS IAM. Untuk menyelesaikan langkah-langkah ini, lihat [Menggunakan aplikasi dengan penerbit token terpercaya](#).

Note

Menyiapkan penerbit token terpercaya tidak diperlukan untuk semua koneksi eksternal. Menyambungkan ke database Redshift Anda dengan editor kueri Amazon Redshift v2 tidak memerlukan konfigurasi penerbit token terpercaya. Tetapi itu dapat berlaku untuk aplikasi pihak ketiga seperti dasbor atau aplikasi khusus yang mengautentikasi dengan penyedia identitas Anda.

- Mengkonfigurasi peran atau peran IAM — Bagian yang mengikuti menyebutkan izin yang harus dikonfigurasi. Anda harus menambahkan izin per praktik terbaik IAM. Izin khusus dirinci dalam prosedur berikut.

Untuk informasi selengkapnya, lihat [Memulai dengan Pusat AWS Identitas IAM](#).

Mengkonfigurasi penyedia identitas Anda untuk bekerja dengan AWS IAM Identity Center

Langkah pertama dalam mengendalikan manajemen identitas pengguna dan grup adalah terhubung ke AWS IAM Identity Center dan mengkonfigurasi penyedia identitas Anda. Anda dapat menggunakan AWS IAM Identity Center sendiri sebagai penyedia identitas Anda, atau Anda dapat menghubungkan toko identitas pihak ketiga, seperti Okta, misalnya. Untuk informasi selengkapnya tentang menyiapkan sambungan ke dan mengonfigurasi penyedia identitas Anda, lihat [Connect ke penyedia identitas eksternal di panduan](#) pengguna AWS IAM Identity Center. Pastikan pada akhir proses ini bahwa Anda memiliki koleksi kecil pengguna dan grup yang ditambahkan ke AWS IAM Identity Center, untuk tujuan pengujian.

Izin Administratif

Izin diperlukan untuk manajemen siklus hidup aplikasi AWS Redshift/IAM Identity Center

Anda harus membuat identitas IAM, yang digunakan administrator Redshift untuk mengonfigurasi Redshift untuk digunakan AWS dengan IAM Identity Center. Paling umum, Anda akan membuat

peran IAM dengan izin dan menentukannya ke identitas lain sesuai kebutuhan. Itu harus memiliki izin yang terdaftar untuk melakukan tindakan berikut.

Membuat aplikasi AWS Redshift/IAM Identity Center

- `sso:PutApplicationAssignmentConfiguration`— Untuk keamanan.
- `sso:CreateApplication`— Digunakan untuk membuat aplikasi AWS IAM Identity Center.
- `sso:PutApplicationAuthenticationMethod`— Memberikan akses otentikasi Redshift.
- `sso:PutApplicationGrant`— Digunakan untuk mengubah informasi penerbit token tepercaya.
- `sso:PutApplicationAccessScope`— Untuk pengaturan aplikasi Redshift AWS IAM Identity Center. Ini termasuk untuk AWS Lake Formation dan untuk Hibah [Akses Amazon S3](#).
- `redshift:CreateRedshiftIdcApplication`— Digunakan untuk membuat aplikasi Redshift AWS IAM Identity Center.

Menjelaskan aplikasi AWS Redshift/IAM Identity Center

- `sso:GetApplicationGrant`— Digunakan untuk mencantumkan informasi penerbit token tepercaya.
- `sso:ListApplicationAccessScopes`— Untuk pengaturan aplikasi Redshift AWS IAM Identity Center untuk mencantumkan integrasi hilir, seperti for AWS Lake Formation dan S3 Access Grants.
- `redshift:DescribeRedshiftIdcApplications`— Digunakan untuk menggambarkan aplikasi AWS IAM Identity Center yang ada.

Mengubah aplikasi AWS Redshift/IAM Identity Center

- `redshift:ModifyRedshiftIdcApplication`— Digunakan untuk mengubah aplikasi Redshift yang ada.
- `sso:UpdateApplication`— Digunakan untuk memperbarui aplikasi Pusat AWS Identitas IAM.
- `sso:GetApplicationGrant`— Mendapatkan informasi penerbit token kepercayaan.
- `sso:ListApplicationAccessScopes`— Untuk pengaturan aplikasi Redshift AWS IAM Identity Center.
- `sso>DeleteApplicationGrant`— Menghapus informasi penerbit token kepercayaan.
- `sso:PutApplicationGrant`— Digunakan untuk mengubah informasi penerbit token tepercaya.
- `sso:PutApplicationAccessScope`— Untuk pengaturan aplikasi Redshift AWS IAM Identity Center. Ini termasuk untuk AWS Lake Formation dan untuk Hibah [Akses Amazon S3](#).

- `sso:DeleteApplicationAccessScope`— Untuk menghapus pengaturan aplikasi AWS Redshift IAM Identity Center. Ini termasuk untuk AWS Lake Formation dan untuk Hibah [Akses Amazon S3](#).

Menghapus aplikasi AWS Redshift/IAM Identity Center

- `sso:DeleteApplication`— Digunakan untuk menghapus aplikasi AWS IAM Identity Center.
- `redshift:DeleteRedshiftIdcApplication`— Memberikan kemampuan untuk menghapus aplikasi Redshift AWS IAM Identity Center yang ada.

Izin diperlukan untuk manajemen Redshift/query siklus hidup aplikasi editor v2

Anda harus membuat identitas IAM, yang digunakan administrator Redshift untuk mengonfigurasi Redshift untuk digunakan AWS dengan IAM Identity Center. Paling umum, Anda akan membuat peran IAM dengan izin dan menetapkan ke identitas lain sesuai kebutuhan. Itu harus memiliki izin yang terdaftar untuk melakukan tindakan berikut.

Membuat aplikasi query editor v2

- `redshift:CreateQev2IdcApplication`— Digunakan untuk membuat QEV2 aplikasi.
- `sso:CreateApplication`— Memberikan kemampuan untuk membuat aplikasi AWS IAM Identity Center.
- `sso:PutApplicationAuthenticationMethod`— Memberikan akses otentikasi Redshift.
- `sso:PutApplicationGrant`— Digunakan untuk mengubah informasi penerbit token tepercaya.
- `sso:PutApplicationAccessScope`— Untuk pengaturan aplikasi Redshift AWS IAM Identity Center. Ini termasuk editor kueri v2.
- `sso:PutApplicationAssignmentConfiguration`— Untuk keamanan.

Jelaskan aplikasi editor kueri v2

- `redshift:DescribeQev2IdcApplications`— Digunakan untuk menggambarkan QEV2 aplikasi AWS IAM Identity Center.

Ubah aplikasi editor kueri v2

- `redshift:ModifyQev2IdcApplication`— Digunakan untuk mengubah QEV2 aplikasi AWS IAM Identity Center.
- `sso:UpdateApplication`— Digunakan untuk mengubah QEV2 aplikasi AWS IAM Identity Center.

Hapus aplikasi editor kueri v2

- `redshift>DeleteQev2IdcApplication`— Digunakan untuk menghapus QEV2 aplikasi.
- `sso>DeleteApplication`— Digunakan untuk menghapus QEV2 aplikasi.

 Note

Di Amazon Redshift SDK, berikut ini APIs tidak tersedia:

- `CreateQev2 IdcApplication`
- `DescribeQev2 IdcApplications`
- `ModifyQev2 IdcApplication`
- `DeleteQev2 IdcApplication`

Tindakan ini khusus untuk melakukan integrasi AWS IAM Identity Center dengan QEV2 Redshift di AWS konsol. Untuk informasi selengkapnya, lihat [Tindakan yang ditentukan oleh Amazon Redshift](#).

Izin yang diperlukan bagi administrator database untuk menghubungkan sumber daya baru di konsol

Izin ini diperlukan untuk menghubungkan kluster baru yang disediakan atau grup kerja Amazon Redshift Tanpa Server selama proses pembuatan. Jika Anda memiliki izin ini, pilihan muncul di konsol untuk memilih untuk terhubung ke aplikasi yang dikelola Pusat Identitas AWS IAM untuk Redshift.

- `redshift:DescribeRedshiftIdcApplications`
- `sso:ListApplicationAccessScopes`
- `sso:GetApplicationAccessScope`
- `sso:GetApplicationGrant`

Sebagai praktik terbaik, kami menyarankan untuk melampirkan kebijakan izin ke peran IAM dan kemudian menetapkan ke pengguna dan grup sesuai kebutuhan. Untuk informasi selengkapnya, lihat [Identitas dan manajemen akses di Amazon Redshift](#).

Menyiapkan Redshift sebagai aplikasi AWS terkelola dengan AWS IAM Identity Center

Sebelum Pusat AWS Identitas IAM dapat mengelola identitas untuk klaster yang disediakan Amazon Redshift atau grup kerja Tanpa Server Amazon Redshift, administrator Redshift harus menyelesaikan langkah-langkah untuk menjadikan Redshift sebagai aplikasi yang dikelola Pusat Identitas IAM: AWS

1. Pilih integrasi Pusat AWS Identitas IAM di menu konsol Amazon Redshift atau Amazon Redshift Tanpa Server, lalu pilih Connect to IAM Identity Center. AWS Dari sana Anda melangkah melalui serangkaian pilihan untuk mengisi properti untuk integrasi AWS IAM Identity Center.
2. Pilih nama tampilan dan nama unik untuk aplikasi yang dikelola AWS IAM Identity Center Redshift.
3. Tentukan namespace untuk organisasi Anda. Ini biasanya merupakan versi singkat dari nama organisasi Anda. Ini ditambahkan sebagai awalan untuk pengguna dan peran yang dikelola Pusat AWS Identitas IAM Anda dalam database Redshift.
4. Pilih peran IAM yang akan digunakan. Peran IAM ini harus terpisah dari yang lain yang digunakan untuk Redshift, dan kami menyarankan agar tidak digunakan untuk tujuan lain. Izin kebijakan khusus yang diperlukan adalah sebagai berikut:
 - `sso:DescribeApplication`— Diperlukan untuk membuat entri penyedia identitas (IDP) di katalog.
 - `sso:DescribeInstance`— Digunakan untuk membuat peran atau pengguna federasi IDP secara manual.
5. Konfigurasi koneksi klien dan penerbit token tepercaya. Mengkonfigurasi penerbit token tepercaya memfasilitasi propagasi identitas tepercaya dengan mengatur hubungan dengan penyedia identitas eksternal. Propagasi identitas memungkinkan pengguna, misalnya, untuk masuk ke satu aplikasi dan mengakses data tertentu di aplikasi lain. Hal ini memungkinkan pengguna untuk mengumpulkan data dari lokasi yang berbeda dengan lebih mulus. Pada langkah ini, di konsol, Anda menetapkan atribut untuk setiap penerbit token tepercaya. Atribut termasuk nama dan klaim audiens (atau klaim aud), yang mungkin harus Anda dapatkan dari atribut konfigurasi alat atau layanan. Anda mungkin juga perlu memberikan nama aplikasi dari JSON Web Token (JWT) alat pihak ketiga.

 Note

Yang aud claim diperlukan dari setiap alat atau layanan pihak ketiga dapat bervariasi, berdasarkan jenis token, yang dapat berupa token akses yang dikeluarkan oleh penyedia identitas, atau jenis lain, seperti token ID. Setiap vendor bisa berbeda. Saat Anda menerapkan propagasi identitas terpercaya dan berintegrasi dengan Redshift, Anda harus memberikan nilai aud yang benar untuk jenis token yang dikirimkan oleh alat pihak ketiga. AWS Periksa rekomendasi dari vendor alat atau layanan Anda.

Untuk informasi rinci mengenai propagasi identitas terpercaya, lihat [Ikhtisar propagasi identitas terpercaya di Panduan Pengguna AWS IAM Identity Center](#)

Setelah administrator Redshift menyelesaikan langkah-langkah dan menyimpan konfigurasi, properti AWS IAM Identity Center muncul di konsol Redshift. Anda juga dapat menanyakan tampilan sistem [SVV_IDENTITY_PROVIDERS](#) untuk memverifikasi properti aplikasi. Ini termasuk nama aplikasi dan namespace. Anda menggunakan namespace sebagai awalan untuk objek database Redshift yang terkait dengan aplikasi. Menyelesaikan tugas-tugas ini membuat Redshift menjadi aplikasi yang diaktifkan Pusat AWS Identitas IAM. Properti di konsol termasuk status integrasi. Dikatakan Diaktifkan ketika integrasi selesai. Setelah proses ini, integrasi AWS IAM Identity Center dapat diaktifkan pada setiap cluster baru.

Setelah konfigurasi, Anda dapat menyertakan pengguna dan grup dari Pusat AWS Identitas IAM di Redshift dengan memilih tab Pengguna atau Grup dan memilih Tetapkan.

Mengaktifkan integrasi AWS IAM Identity Center untuk cluster Amazon Redshift baru atau grup kerja Amazon Redshift Serverless

Administrator database Anda mengonfigurasi sumber daya Redshift baru agar berfungsi sejajar AWS dengan IAM Identity Center untuk mempermudah proses masuk dan akses data. Ini dilakukan sebagai bagian dari langkah-langkah untuk membuat klaster yang disediakan atau grup kerja Tanpa Server. Siapa pun yang memiliki izin untuk membuat resource Redshift dapat melakukan tugas integrasi Pusat Identitas IAM AWS ini. Saat Anda membuat klaster yang disediakan, Anda mulai dengan memilih Create Cluster di konsol Amazon Redshift. Langkah-langkah berikut menunjukkan cara mengaktifkan manajemen AWS IAM Identity Center untuk database. (Itu tidak termasuk semua langkah untuk membuat cluster.)

1. Pilih Aktifkan untuk <your cluster name> di bagian untuk integrasi IAM Identity Center dalam langkah-langkah create-cluster.
2. Ada langkah dalam proses ketika Anda mengaktifkan integrasi. Anda melakukan ini dengan memilih Aktifkan integrasi IAM Identity Center di konsol.
3. Untuk cluster atau workgroup baru, buat peran database di Redshift menggunakan perintah SQL. Berikut ini adalah perintahnya:

```
CREATE ROLE <idcnamespace:rolename>;
```

Namespace dan nama peran adalah sebagai berikut:

- Awalan namespace IAM Identity Center — Ini adalah namespace yang Anda tentukan saat Anda mengatur koneksi antara IAM AWS Identity Center dan Redshift.
- Nama peran - Peran database Redshift ini harus cocok dengan nama grup di Pusat Identitas AWS IAM.

Redshift terhubung dengan AWS IAM Identity Center dan mengambil informasi yang diperlukan untuk membuat dan memetakan peran database ke grup IAM AWS Identity Center.

Perhatikan bahwa ketika gudang data baru dibuat, peran IAM yang ditentukan untuk integrasi Pusat Identitas AWS IAM secara otomatis dilampirkan ke klaster yang disediakan atau grup kerja Amazon Redshift Serverless. Setelah Anda selesai memasukkan metadata cluster yang diperlukan dan membuat sumber daya, Anda dapat memeriksa status integrasi Pusat Identitas AWS IAM di properti. Jika nama grup Anda di AWS IAM Identity Center memiliki spasi, Anda harus menggunakan tanda kutip di SQL saat Anda membuat peran yang cocok.

Setelah Anda mengaktifkan database Redshift dan membuat peran, Anda siap untuk terhubung ke database dengan Amazon Redshift query editor v2 atau. Amazon Quick Rinciannya dijelaskan lebih lanjut di bagian berikutnya.

Menyiapkan default **RedshiftIdcApplication** menggunakan API

Pengaturan dilakukan oleh administrator identitas Anda. Menggunakan API, Anda membuat dan mengisi **RedshiftIdcApplication**, yang mewakili aplikasi Redshift AWS dalam IAM Identity Center.

1. Untuk memulai, Anda dapat membuat pengguna dan menambahkannya ke grup di Pusat AWS Identitas IAM. Anda melakukan ini di AWS konsol untuk AWS IAM Identity Center.
2. Hubungi `create-redshift-idc-application` untuk membuat aplikasi AWS IAM Identity Center dan membuatnya kompatibel dengan penggunaan Redshift. Anda membuat aplikasi dengan mengisi nilai yang diperlukan. Nama tampilan adalah nama yang akan ditampilkan di dasbor AWS IAM Identity Center. Peran IAM ARN adalah ARN yang memiliki izin AWS untuk IAM Identity Center dan juga dapat diasumsikan oleh Redshift.

```
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'
```

Contoh berikut menunjukkan `RedshiftIdcApplication` respons sampel yang dikembalikan dari panggilan `create-redshift-idc-application`.

```
"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. Anda dapat menggunakan `create-application-assignment` untuk menetapkan grup tertentu atau pengguna individu ke aplikasi terkelola di Pusat Identitas AWS IAM. Dengan melakukan ini, Anda dapat menentukan grup untuk dikelola melalui AWS IAM Identity Center.

Jika administrator database membuat peran database di Redshift, nama grup di Pusat Identitas AWS IAM memetakan ke nama peran di Redshift. Peran mengontrol izin dalam database. Untuk informasi selengkapnya, lihat [Menetapkan akses pengguna ke aplikasi di konsol Pusat AWS Identitas IAM](#).

4. Setelah Anda mengaktifkan aplikasi, panggil `create-cluster` dan sertakan ARN aplikasi terkelola Redshift dari AWS IAM Identity Center. Melakukan hal ini mengaitkan cluster dengan aplikasi terkelola di AWS IAM Identity Center.

Mengaitkan aplikasi AWS IAM Identity Center dengan cluster atau workgroup yang ada

Jika Anda memiliki cluster atau workgroup yang ingin Anda aktifkan untuk integrasi AWS IAM Identity Center, dimungkinkan untuk melakukannya, menjalankan perintah SQL. Anda juga dapat menjalankan perintah SQL untuk mengubah pengaturan untuk integrasi. Untuk informasi selengkapnya, lihat [MENGUBAH PENYEDIA IDENTITAS](#).

Dimungkinkan juga untuk menjatuhkan penyedia identitas yang ada. Contoh berikut menunjukkan bagaimana CASCADE menghapus pengguna dan peran yang dilampirkan ke penyedia identitas.

```
DROP IDENTITY PROVIDER  
<provider_name> [ CASCADE ]
```

Menyiapkan izin pengguna

Administrator mengonfigurasi izin ke berbagai sumber daya, berdasarkan atribut identitas pengguna dan keanggotaan grup, dalam penyedia identitas mereka atau di dalam Pusat Identitas AWS IAM secara langsung. Misalnya, administrator penyedia identitas dapat menambahkan insinyur basis data ke grup yang sesuai dengan peran mereka. Nama grup ini dipetakan ke nama peran database Redshift. Peran menyediakan atau membatasi akses ke tabel atau tampilan tertentu di Redshift.

Secara otomatis membuat peran Amazon Redshift untuk AWS IAM Identity Center

Fitur ini merupakan integrasi dengan AWS IAM Identity Center yang memungkinkan Anda untuk secara otomatis membuat peran di Redshift berdasarkan keanggotaan grup.

Ada beberapa manfaat untuk membuat peran secara otomatis. Saat Anda membuat peran secara otomatis, Redshift membuat peran dengan keanggotaan grup di IDP Anda, sehingga Anda dapat menghindari pembuatan dan pemeliharaan peran manual yang membosankan. Anda juga memiliki opsi untuk memfilter grup mana yang dipetakan ke peran Redshift dengan pola include dan exclude.

Cara kerjanya

Saat Anda, sebagai pengguna iDP, masuk ke Redshift, urutan peristiwa berikut terjadi:

1. Redshift mengambil keanggotaan grup Anda dari IDP.
2. Redshift secara otomatis membuat pemetaan peran ke grup tersebut, dengan format peran.
idp_namespace:rolename
3. Redshift memberi Anda izin dengan peran yang dipetakan.

Setelah setiap login pengguna, setiap grup yang tidak ada dalam katalog tetapi pengguna adalah bagian darinya, dibuat secara otomatis. Anda dapat mengatur filter include dan exclude secara opsional untuk mengontrol grup IDP mana yang memiliki peran Redshift yang dibuat.

Mengkonfigurasi peran buat otomatis

Gunakan ALTER IDENTITY PROVIDER perintah CREATE IDENTITY PROVIDER and untuk mengaktifkan dan mengkonfigurasi pembuatan peran otomatis.

```
-- 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;
```

Grup penyaringan

Anda dapat secara opsional memfilter grup IDP mana yang dipetakan ke peran INCLUDE Redshift menggunakan dan pola. EXCLUDE Ketika pola konflik, EXCLUDE lebih diutamakan. INCLUDE

```
-- 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%';
```

Contoh

Contoh berikut menunjukkan cara mengaktifkan peran buat otomatis tanpa pemfilteran.

```
CREATE IDENTITY PROVIDER prod_idc TYPE AWSIDC ...
  AUTO_CREATE_ROLES TRUE;
```

Contoh berikut mencakup kelompok pengembangan dan tidak termasuk kelompok uji.

```
ALTER IDENTITY PROVIDER prod_idc
  AUTO_CREATE_ROLES TRUE
  INCLUDE GROUPS LIKE '%dev%'
  EXCLUDE GROUPS LIKE '%test%';
```

Praktik terbaik

Pertimbangkan praktik terbaik berikut saat Anda mengaktifkan pembuatan otomatis untuk peran:

- Gunakan INCLUDE dan EXCLUDE filter untuk mengontrol grup mana yang mendapatkan peran.
- Secara berkala mengaudit peran dan membersihkan yang tidak terpakai.
- Manfaatkan hierarki peran Redshift untuk menyederhanakan manajemen izin.

Integrasi Amazon Redshift dengan Hibah Akses Amazon S3

Menggunakan integrasi dengan Amazon S3 Access Grants, Anda dapat menyebarkan identitas Pusat Identitas IAM Anda dengan mulus untuk mengontrol akses ke data Amazon S3. Integrasi ini memungkinkan Anda mengotorisasi akses data Amazon S3 berdasarkan pengguna dan grup Pusat Identitas IAM.

Untuk informasi tentang Hibah Akses Amazon S3, lihat [Mengelola akses dengan Hibah Akses S3](#).

Menggunakan Amazon S3 Access Grants memberi aplikasi Anda manfaat berikut:

- Kontrol akses berbutir halus ke data Amazon S3, berdasarkan identitas Pusat Identitas IAM.
- Manajemen terpusat identitas IAM Identity Center di Amazon Redshift dan Amazon S3.
- Anda dapat menghindari mengelola izin IAM terpisah untuk akses Amazon S3.

Cara kerjanya

Untuk mengintegrasikan aplikasi Anda dengan hibah akses Amazon S3, Anda melakukan hal berikut:

- Pertama, Anda mengonfigurasi Amazon Redshift untuk berintegrasi dengan Amazon S3 Access Grants menggunakan atau. Konsol Manajemen AWS AWS CLI
- Selanjutnya, pengguna dengan hak administrator IDC memberikan bucket Amazon S3 atau akses awalan ke pengguna/grup IDC tertentu, menggunakan layanan Amazon S3 Access Grants. Untuk informasi selengkapnya, lihat [Bekerja dengan hibah di Hibah Akses S3](#).
- Saat pengguna iDC diautentikasi ke Redshift menjalankan kueri yang mengakses S3 (seperti operasi COPY, UNLOAD, atau Spectrum), Amazon Redshift mengambil kredensial akses S3 sementara yang tercakup ke identitas idC tersebut dari layanan Amazon S3 Access Grants.
- Amazon Redshift kemudian menggunakan kredensial sementara yang diambil untuk mengakses lokasi Amazon S3 resmi untuk kueri tersebut.

Menyiapkan integrasi dengan Amazon S3 Access Grants

Untuk mengatur integrasi dengan integrasi dengan Amazon S3 Access Grants for Amazon Redshift, lakukan hal berikut:

Topik

- [Menyiapkan integrasi dengan Amazon S3 Access Grants menggunakan Konsol Manajemen AWS](#)
- [Mengaktifkan integrasi dengan Amazon S3 Access Grants menggunakan AWS CLI](#)

Menyiapkan integrasi dengan Amazon S3 Access Grants menggunakan Konsol Manajemen AWS

1. Buka konsol Amazon Redshift.
2. Pilih klaster Anda dari panel Clusters.
3. Di halaman detail klaster Anda, di bagian Integrasi penyedia identitas, aktifkan integrasi dengan layanan S3 Access Grants.

 Note

Bagian integrasi penyedia Identity tidak muncul jika Anda tidak memiliki IAM Identity Center yang dikonfigurasi. Untuk informasi selengkapnya, lihat [Mengaktifkan AWS IAM Identity Center](#).

Mengaktifkan integrasi dengan Amazon S3 Access Grants menggunakan AWS CLI

1. Untuk membuat aplikasi Amazon Redshift iDC baru dengan integrasi S3 diaktifkan, lakukan hal berikut:

```
aws redshift create-redshift-idc-application <other parameters>
  --service-integrations '[ {"S3AccessGrants": [{"ReadWriteAccess":
{"Authorization": "Enabled"}]} ]'
```

2. Untuk memodifikasi aplikasi yang ada untuk mengaktifkan integrasi S3 Access Grants, lakukan hal berikut:

```
aws redshift modify-redshift-idc-application <other parameters>
  --service-integrations '[ {"S3AccessGrants": [{"ReadWriteAccess":
{"Authorization": "Enabled"}]} ]'
```

3. Untuk memodifikasi aplikasi yang ada untuk menonaktifkan integrasi S3 Access Grants, lakukan hal berikut:

```
aws redshift modify-redshift-idc-application <other parameters>
  --service-integrations '[ {"S3AccessGrants": [{"ReadWriteAccess":
{"Authorization": "Disabled"}]} ]'
```

Menggunakan integrasi dengan S3 Access Grants

Setelah Anda mengonfigurasi integrasi S3 Access Grants, kueri yang mengakses data S3 (seperti COPY, UNLOAD, atau kueri Spectrum) menggunakan identitas IDC untuk otorisasi. Pengguna yang tidak diautentikasi menggunakan IDC juga dapat menjalankan kueri ini, tetapi akun pengguna tersebut tidak memanfaatkan administrasi terpusat yang disediakan IDC.

Contoh berikut menunjukkan kueri yang berjalan dengan integrasi S3 Access Grants:

```
COPY table FROM 's3://mybucket/data'; // -- Redshift uses IdC identity
UNLOAD ('SELECT * FROM table') TO 's3://mybucket/unloaded/' // -- Redshift uses IdC
identity
```

Memeriksa data melalui AWS Lake Formation

Menggunakan AWS Lake Formation membuatnya lebih mudah untuk mengatur dan mengamankan data lake Anda secara terpusat, dan untuk menyediakan akses data. Mengkonfigurasi propagasi identitas ke Lake Formation melalui AWS IAM Identity Center dan Redshift membuatnya sehingga administrator dapat mengizinkan akses halus ke danau data Amazon S3, berdasarkan grup penyedia identitas organisasi (iDP). Kelompok-kelompok ini dikelola melalui AWS IAM Identity Center. Bagian ini menunjukkan cara mengonfigurasi beberapa kasus penggunaan, kueri dari data lake dan kueri dari berbagi data, yang menunjukkan cara memanfaatkan Pusat Identitas AWS IAM dengan Redshift untuk terhubung ke sumber daya yang diatur oleh Lake Formation.

Menggunakan Pusat AWS Identitas IAM dan koneksi Redshift untuk menanyakan data lake

Langkah-langkah ini mencakup kasus penggunaan di mana Anda menggunakan Pusat AWS Identitas IAM yang terhubung ke Redshift untuk menanyakan data lake yang diatur oleh Lake Formation.

Prasyarat

Prosedur ini memiliki beberapa langkah prasyarat:

1. AWS Pusat Identitas IAM harus disiapkan untuk mendukung otentikasi dan manajemen identitas dengan Redshift. Anda dapat mengaktifkan Pusat AWS Identitas IAM dari konsol dan memilih sumber penyedia identitas (iDP). Setelah ini, sinkronkan satu set pengguna IDP Anda AWS dengan IAM Identity Center. Anda juga harus mengatur koneksi antara AWS IAM Identity Center dan Redshift, mengikuti langkah-langkah yang dijelaskan sebelumnya dalam dokumen ini.
2. Buat klaster Amazon Redshift baru dan aktifkan manajemen identitas melalui AWS IAM Identity Center dalam langkah-langkah konfigurasi.
3. Buat aplikasi Pusat Identitas AWS IAM terkelola untuk Lake Formation dan konfigurasikan. Ini mengikuti pengaturan koneksi antara AWS IAM Identity Center dan Redshift. Langkah-langkahnya adalah sebagai berikut:
 - a. Di AWS CLI, gunakan `modify-redshift-idc-application` perintah untuk mengaktifkan integrasi layanan Lake Formation dengan aplikasi yang dikelola Pusat AWS Identitas IAM untuk

Redshift. Panggilan ini mencakup `service-integrations` parameter, yang diatur ke nilai string konfigurasi yang memungkinkan otorisasi ke Lake Formation.

- b. Konfigurasi Lake Formation dengan menggunakan `create-lake-formation-identity-center-configuration` perintah. Ini menciptakan aplikasi Pusat AWS Identitas IAM untuk Lake Formation, yang terlihat di portal AWS IAM Identity Center. Administrator harus mengatur `--cli-input-json` argumen, yang nilainya adalah jalur ke file JSON yang menggunakan format standar untuk semua panggilan AWS CLI API. Anda harus menyertakan nilai untuk hal-hal berikut:

- `CatalogId`— ID katalog Lake Formation.
- `InstanceArn`— Pusat AWS Identitas IAM misalnya nilai ARN.

Setelah administrator menyelesaikan konfigurasi prasyarat, administrator database dapat membuat skema eksternal untuk tujuan kueri data lake.

1. Administrator membuat skema eksternal - Administrator database Redshift terhubung ke database dan membuat skema eksternal, menggunakan pernyataan SQL berikut:

```
CREATE EXTERNAL SCHEMA if not exists my_external_schema from DATA CATALOG database
'my_lf_integrated_db' catalog_id '12345678901234';
```

Perhatikan bahwa menentukan peran IAM tidak diperlukan dalam kasus ini, karena akses dikelola melalui AWS IAM Identity Center.

2. Administrator memberikan izin — Administrator memberikan penggunaan ke grup Pusat Identitas AWS IAM, yang memberikan izin pada sumber daya Redshift. Hal ini dilakukan dengan menjalankan pernyataan SQL seperti berikut:

```
GRANT USAGE ON SCHEMA "my_external_schema" to "MYC0:sales";
```

Selanjutnya, administrator memberikan izin Lake Formation pada objek, berdasarkan persyaratan untuk organisasi, menggunakan CLI AWS :

```
aws lakeformation grant-permissions ...
```

3. Pengguna menjalankan kueri — Pada titik ini, pengguna Pusat AWS Identitas IAM yang merupakan bagian dari grup penjualan, untuk tujuan ilustrasi, dapat masuk melalui editor kueri v2

ke database Redshift. Kemudian mereka dapat menjalankan kueri yang mengakses tabel dalam skema eksternal, seperti contoh berikut:

```
SELECT * from my_external_schema.table1;
```

Menggunakan Pusat AWS Identitas IAM dan koneksi Redshift untuk terhubung ke datashare

Anda dapat mengakses datashare dari gudang data Redshift yang berbeda saat akses dikelola melalui AWS IAM Identity Center. Untuk melakukan ini, Anda menjalankan kueri untuk menyiapkan database eksternal. Sebelum menyelesaikan langkah-langkah ini, diasumsikan bahwa Anda memiliki koneksi yang diatur antara Redshift dan AWS IAM Identity Center, dan Anda telah membuat AWS Lake Formation aplikasi, seperti yang dijelaskan dalam prosedur sebelumnya.

1. Membuat database eksternal — Administrator membuat database eksternal untuk berbagi data, mereferensikannya melalui ARN nya. Berikut ini adalah contoh yang menunjukkan cara melakukannya:

```
CREATE DATABASE "redshift_external_db" FROM ARN 'arn:aws:glue:us-east-1:123456789012:database/redshift_external_db-iad' WITH NO DATA CATALOG SCHEMA;
```

Dalam kasus penggunaan ini, di mana Anda menggunakan Pusat AWS Identitas IAM dengan Redshift untuk manajemen identitas, peran IAM tidak disertakan.

2. Admin menyiapkan izin — Setelah membuat database, administrator memberikan penggunaan ke grup Pusat Identitas AWS IAM. Ini memberikan izin pada sumber daya Redshift:

```
GRANT USAGE ON DATABASE "my_external_db" to "MYC0:sales";
```

Administrator juga memberikan izin Lake Formation pada objek, menggunakan CLI AWS :

```
aws lakeformation grant-permissions ...
```

3. Pengguna menjalankan kueri — Pengguna dari grup penjualan dapat melakukan kueri tabel di database, berdasarkan izin yang ditetapkan:

```
select * from redshift_external_db.public.employees;
```

Untuk informasi selengkapnya tentang pemberian izin di data lake dan pemberian izin pada pembagian data, lihat [Memberikan izin kepada pengguna dan grup](#). [Untuk informasi selengkapnya tentang pemberian penggunaan skema atau database, lihat GRANT](#).

Mengintegrasikan aplikasi atau alat Anda dengan OAuth menggunakan penerbit token tepercaya

Anda dapat menambahkan fungsionalitas ke alat klien yang Anda buat untuk terhubung ke Redshift melalui koneksi AWS IAM Identity Center. Jika Anda sudah mengonfigurasi integrasi Redshift ke AWS IAM Identity Center, gunakan properti yang dirinci di bagian ini untuk menyiapkan koneksi.

Plugin otentikasi untuk menghubungkan ke Redshift AWS menggunakan IAM Identity Center

Anda dapat menggunakan AWS IAM Identity Center untuk terhubung ke Amazon Redshift menggunakan plugin driver berikut:

- **BrowserIdcAuthPlugin**- Plugin ini memfasilitasi single-sign-on integrasi tanpa batas dengan AWS IAM Identity Center. Ini menciptakan jendela browser bagi pengguna untuk masuk dengan kredensi pengguna yang ditentukan dalam penyedia identitas perusahaan mereka.
- **IdpTokenAuthPlugin**- Plugin ini harus digunakan oleh aplikasi yang ingin mengelola aliran otentikasi sendiri, alih-alih membiarkan driver Amazon Redshift membuka jendela browser AWS untuk otentikasi IAM Identity Center. Ia menerima token Access vended AWS IAM Identity Center atau OpenID Connect (OIDC) JSON web token (JWT) dari penyedia identitas web apa pun yang terhubung AWS dengan IAM Identity Center, seperti Okta,, dan Microsoft Entra ID (Azure AD). PingOne Aplikasi klien bertanggung jawab untuk menghasilkan token akses/JWT yang diperlukan ini.

Autentikasi dengan **BrowserIdcAuthPlugin**

Gunakan nama plugin berikut untuk terhubung menggunakan **BrowserIdcAuthPlugin**, tergantung pada driver Amazon Redshift Anda.

Driver	Kunci opsi koneksi	Nilai	Catatan
JDBC	plugin_name	com.amazon.redshif t.plugin. BrowserId cAuthPlugin	Anda harus memasukkan nama kelas plugin yang sepenuhnya

Driver	Kunci opsi koneksi	Nilai	Catatan
			memenuhi syarat saat Anda terhubung.
ODBC	plugin_name	BrowserIdcAuthPlugin	
Python	credential_provider	BrowserIdcAuthPlugin	Tidak ada plugin_name pilihan yang tersedia untuk driver Python. Sebaliknya, gunakan credential_provider .

BrowserIdcAuthPlugin ini memiliki opsi koneksi tambahan berikut:

Nama opsi	Wajib?	Deskripsi	Contoh
idc_region	Diperlukan	Di AWS Region mana instans AWS IAM Identity Center berada.	us-east-1
issuer_url	Diperlukan	Titik akhir instance server AWS IAM Identity Center. Anda dapat menemukan nilai ini menggunakan konsol Pusat AWS Identitas IAM.	https://identitycenter.amazonaws.com/ssoins-g5j2k6yc5nsc
listen_port	Opsional	Port yang digunakan driver Amazon Redshift untuk menerima auth_code respons	7890

Nama opsi	Wajib?	Deskripsi	Contoh
		dari AWS IAM Identity Center melalui pengalihan browser.	
idc_client_display_name	Opsional	Nama yang digunakan klien AWS IAM Identity Center untuk aplikasi dalam popup persetujuan masuk tunggal AWS IAM Identity Center.	Pengemudi Amazon Redshift
idp_response_timeout	Opsional	Jumlah waktu, dalam hitungan detik, driver Redshift menunggu aliran autentikasi selesai.	60

Anda harus memasukkan nilai-nilai ini di properti koneksi alat yang Anda buat dan hubungkan. Untuk informasi selengkapnya, lihat dokumentasi opsi koneksi untuk setiap driver masing-masing:

- [Opsi untuk konfigurasi driver JDBC versi 2.x](#)
- [Opsi driver ODBC](#)
- [Opsi konfigurasi untuk konektor Amazon Redshift Python](#)

Autentikasi dengan **IdpTokenAuthPlugin**

Gunakan nama plugin berikut untuk terhubung menggunakan `IdpTokenAuthPlugin`, tergantung pada driver Amazon Redshift Anda.

Driver	Kunci opsi koneksi	Nilai	Catatan
JDBC	plugin_name	com.amazon.redshift.plugin.IdpTokenAuthPlugin	Anda harus memasukkan nama kelas plugin

Driver	Kunci opsi koneksi	Nilai	Catatan
			yang sepenuhnya memenuhi syarat saat Anda terhubung.
ODBC	plugin_name	IdpTokenAuthPlugin	
Python	credential_provider	IdpTokenAuthPlugin	Tidak ada plugin_name pilihan yang tersedia untuk driver Python. Sebaliknya, gunakan credential_provider .

IdpTokenAuthPlugin ini memiliki opsi koneksi tambahan berikut:

Nama opsi	Wajib?	Deskripsi
token	Diperlukan	Pusat AWS Identitas IAM menjual token akses atau OpenID Connect (OIDC) JSON Web Token (JWT) yang disediakan oleh penyedia identitas web yang terhubung dengan IAM Identity Center. AWS Aplikasi Anda harus menghasilkan token ini dengan mengautentikasi pengguna aplikasi Anda dengan AWS IAM Identity Center atau penyedia identitas yang terhubung dengan AWS IAM Identity Center.

Nama opsi	Wajib?	Deskripsi
token_type	Diperlukan	Jenis token yang digunakan untukIdpTokenAuthPlugin. Nilai yang mungkin adalah sebagai berikut: • ACCESS_TOKEN — Masukkan ini jika Anda menggunakan token akses yang disediakan Pusat AWS Identitas IAM. • EXT_JWT — Masukkan ini jika Anda menggunakan OpenID Connect (OIDC) JSON Web Token (JWT) yang disediakan oleh penyedia identitas berbasis web yang terhubung dengan IAM Identity Center. AWS

Anda harus memasukkan nilai-nilai ini di properti koneksi alat yang Anda buat dan hubungkan. Untuk informasi selengkapnya, lihat dokumentasi opsi koneksi untuk setiap driver masing-masing:

- [Opsinya untuk konfigurasi driver JDBC versi 2.x](#)
- [Opsinya driver ODBC](#)
- [Opsinya konfigurasi untuk konektor Amazon Redshift Python](#)

Memecahkan masalah koneksi dari editor kueri Amazon Redshift v2

Daftar ini merinci kesalahan yang biasanya terjadi dan dapat membantu Anda terhubung ke database Redshift Anda dengan editor kueri v2, menggunakan identitas Pusat Identitas AWS IAM.

- **Kesalahan: Masalah Koneksi:** Tidak ada informasi sesi pusat identitas yang tersedia. — Ketika kesalahan ini terjadi, periksa pengaturan keamanan dan privasi browser Anda. Pengaturan

browser ini, terutama untuk cookie aman, seperti fitur Perlindungan Cookie Total Firefox, dapat mengakibatkan upaya koneksi yang diblokir dari editor kueri Amazon Redshift v2 ke database Redshift. Ikuti langkah-langkah remediasi yang dirinci untuk browser Anda:

- Firefox — Saat ini, cookie pihak ketiga diblokir secara default. Klik perisai di bilah alamat browser dan alihkan sakelar untuk mematikan perlindungan pelacakan yang disempurnakan untuk editor kueri v2.
- Mode penyamaran Chrome — Secara default, mode Penyamaran Chrome memblokir cookie pihak ketiga. Klik ikon mata di bilah alamat untuk mengizinkan cookie pihak ketiga untuk editor kueri v2. Setelah Anda mengubah pengaturan untuk mengizinkan cookie, Anda mungkin tidak melihat ikon mata di bilah alamat.
- Safari — Di Mac, buka aplikasi Safari. Pilih Pengaturan, lalu pilih Advanced. Beralih untuk mematikan: Blokir semua cookie.
- Edge — Pilih Pengaturan, lalu pilih Cookie dan izin situs. Kemudian pilih Kelola dan hapus cookie dan data situs dan matikan Blokir cookie pihak ketiga.

Jika Anda mencoba menghubungkan setelah mengubah pengaturan dan terus menerima pesan kesalahan Masalah Koneksi: Tidak ada informasi sesi pusat Identitas yang tersedia, kami sarankan Anda menyegarkan koneksi Anda dengan AWS IAM Identity Center. Untuk melakukan ini, klik kanan instance database Redshift Anda dan pilih Refresh. Jendela baru muncul, yang dapat Anda gunakan untuk mengautentikasi.

- Kesalahan: Masalah koneksi: Sesi pusat identitas kedaluwarsa atau tidak valid. — Setelah integrasi cluster yang disediakan Redshift atau grup kerja Tanpa Server dengan AWS IAM Identity Center, pengguna mungkin menerima kesalahan ini ketika mereka mencoba terhubung ke database Redshift dari editor kueri v2. Ini dapat mengikuti upaya koneksi yang berhasil. Dalam hal ini, kami sarankan Anda mengautentikasi ulang. Untuk melakukan ini, klik kanan instance database Redshift Anda dan pilih Refresh. Jendela baru muncul, yang dapat Anda gunakan untuk mengautentikasi.
- Kesalahan: Cakupan tidak valid. Kredensi pengguna tidak diizinkan untuk terhubung ke Redshift. — Setelah integrasi cluster yang disediakan Redshift atau grup kerja Tanpa Server dengan Pusat Identitas AWS IAM untuk manajemen identitas, pengguna mungkin menerima kesalahan ini ketika mereka mencoba terhubung ke database Redshift dari editor kueri v2. Dalam hal ini, agar editor kueri v2 berhasil menghubungkan dan mengautentikasi pengguna melalui AWS IAM Identity Center untuk mengakses sumber daya yang benar, administrator harus menetapkan pengguna ke aplikasi Redshift AWS IAM Identity Center melalui konsol Redshift. Ini diselesaikan di bawah koneksi IAM Identity Center. Setelah ini, pengguna dapat membuat koneksi yang sukses setelah satu jam, yang merupakan batas caching sesi AWS IAM Identity Center.

- **Kesalahan: Database tidak dapat dicantumkan. FATAL: Kueri gagal saat cluster dijeda secara otomatis.** — Saat database Amazon Redshift Tanpa Server dalam keadaan idle, tidak memproses beban kerja apa pun, database tersebut dapat tetap dijeda saat Anda terhubung dengan identitas Pusat Identitas IAM. AWS Untuk memperbaiki ini, masuk dengan metode otentikasi lain untuk melanjutkan workgroup Tanpa Server. Kemudian sambungkan ke database dengan identitas AWS IAM Identity Center Anda.
- **Kesalahan: Terjadi kesalahan selama upaya untuk berfederasi dengan AWS IAM Identity Center.** Administrator Amazon Redshift harus menghapus dan membuat ulang QEV2 aplikasi Pusat Identitas AWS IAM, menggunakan konsol Redshift. — Kesalahan ini biasanya terjadi ketika instance aplikasi AWS IAM Identity Center yang terkait dengan editor kueri v2 dihapus. Untuk mengatasinya, administrator Amazon Redshift harus menghapus dan membuat ulang aplikasi Redshift dan query editor v2 untuk IAM Identity Center. AWS Ini dapat dilakukan pada konsol Redshift atau menggunakan perintah CLI <https://docs.aws.amazon.com/cli/latest/reference/redshift/delete-redshift-idc-application.html>.

Menggunakan peran terkait layanan untuk Amazon Redshift

Amazon Redshift menggunakan peran terkait layanan AWS Identity and Access Management (IAM).

Peran terkait layanan adalah jenis peran IAM unik yang ditautkan langsung ke Amazon Redshift. Peran terkait layanan telah ditentukan sebelumnya oleh Amazon Redshift dan menyertakan semua izin yang diperlukan layanan untuk memanggil AWS layanan atas nama klaster Amazon Redshift Anda.

Peran terkait layanan membuat pengaturan Amazon Redshift lebih mudah karena Anda tidak perlu menambahkan izin yang diperlukan secara manual. Peran tersebut ditautkan ke kasus penggunaan Amazon Redshift dan memiliki izin yang telah ditentukan sebelumnya. Hanya Amazon Redshift yang dapat mengambil peran tersebut, dan hanya peran terkait layanan yang dapat menggunakan kebijakan izin yang telah ditentukan sebelumnya. Amazon Redshift membuat peran terkait layanan di akun Anda saat pertama kali membuat klaster atau titik akhir VPC yang dikelola RedShift. Anda dapat menghapus peran terkait layanan hanya setelah menghapus semua klaster Amazon Redshift atau titik akhir VPC yang dikelola RedShift di akun Anda. Ini melindungi sumber daya Amazon Redshift karena Anda tidak dapat secara tidak sengaja menghapus izin yang diperlukan untuk mengakses sumber daya.

Amazon Redshift mendukung penggunaan peran terkait layanan di semua Wilayah tempat layanan tersedia. Untuk informasi selengkapnya, lihat [AWS Wilayah dan Titik Akhir](#).

Untuk informasi tentang layanan lain yang mendukung peran terkait layanan, lihat [Layanan AWS yang berfungsi dengan IAM](#) lalu cari layanan yang menampilkan Ya pada kolom Peran Terkait Layanan. Pilih Ya dengan sebuah tautan untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut.

Izin peran terkait layanan untuk Amazon Redshift

Amazon Redshift menggunakan peran terkait layanan bernama — Memungkinkan AWSServiceRoleForRedshiftAmazon Redshift memanggil layanan atas nama Anda. AWS Peran terkait layanan ini dilampirkan ke kebijakan terkelola berikut ini: AmazonRedshiftServiceLinkedRolePolicy. Untuk pembaruan kebijakan ini, lihat kebijakan yang [AWS dikelola \(telah ditentukan sebelumnya\) untuk Amazon Redshift](#).

Peran AWSService RoleForRedshift terkait layanan hanya percaya **redshift.amazonaws.com** untuk mengambil peran.

Kebijakan izin peran AWSService RoleForRedshift terkait layanan memungkinkan Amazon Redshift menyelesaikan hal berikut di semua sumber daya terkait:

- 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`

Izin untuk sumber daya jaringan

Izin berikut memungkinkan tindakan di Amazon EC2 untuk pembuatan dan pengelolaan aturan grup keamanan. Grup dan aturan keamanan ini secara khusus terkait dengan tag `aws:RequestTag/Redshift sumber daya Amazon Redshift`. Ini membatasi ruang lingkup izin untuk sumber daya Amazon Redshift tertentu.

- `ec2:CreateSecurityGroup`
- `ec2:AuthorizeSecurityGroupEgress`
- `ec2:AuthorizeSecurityGroupIngress`
- `ec2:RevokeSecurityGroupEgress`
- `ec2:RevokeSecurityGroupIngress`
- `ec2:ModifySecurityGroupRules`
- `ec2>DeleteSecurityGroup`

Izin untuk kuota layanan

Izin berikut memungkinkan penelepon untuk mendapatkan kuota layanan.

`servicequotas:GetServiceQuota`

Fragmen JSON berikut menunjukkan tindakan dan lingkup sumber daya untuk kuota layanan.

```
{
  "Sid": "ServiceQuotasToCheckCustomerLimits",
  "Effect": "Allow",
  "Action": [
```

```

    "servicequotas:GetServiceQuota"
  ],
  "Resource": [
    "arn:aws:servicequotas::*:ec2/L-0263D0A3",
    "arn:aws:servicequotas::*:vpc/L-29B6F2EB"
  ]
}

```

Kode kuota adalah sebagai berikut:

- L-0263D0A3 — Kode kuota untuk EC2-VPC Elastic. IPs
- L-29B6F2EB — Kode kuota untuk titik akhir VPC Antarmuka per VPC.

Untuk informasi selengkapnya, lihat [AWS service quotas](#).

Tindakan untuk pencatatan audit

Tindakan yang tercantum dengan logs awalan berkaitan dengan audit logging dan fitur terkait. Secara khusus, pembuatan dan pengelolaan grup log dan aliran log.

- logs:CreateLogGroup
- logs:PutRetentionPolicy
- logs:CreateLogStream
- logs:PutLogEvents
- logs:DescribeLogStreams
- logs:GetLogEvents

JSON berikut menunjukkan tindakan dan cakupan sumber daya, ke Amazon Redshift, untuk pencatatan audit.

```

[
  {
    "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:*"
    ]
}
]

```

Untuk informasi selengkapnya tentang peran terkait layanan dan tujuannya AWS, lihat [Menggunakan peran terkait layanan](#). Untuk informasi selengkapnya tentang tindakan tertentu dan sumber daya IAM lainnya untuk Amazon Redshift, [lihat Tindakan, sumber daya, dan kunci kondisi untuk Amazon Redshift](#).

Tindakan untuk mengelola kredensi admin dengan AWS Secrets Manager

Tindakan yang tercantum dengan `secretsmanager` awalan berkaitan dengan penggunaan Amazon Redshift untuk mengelola kredensial admin Anda. Tindakan ini memungkinkan Amazon Redshift digunakan AWS Secrets Manager untuk membuat dan mengelola rahasia kredensi admin Anda.

JSON berikut menunjukkan tindakan dan cakupan sumber daya, ke Amazon Redshift, untuk mengelola kredensi admin dengan AWS Secrets Manager

```

[
  {
    "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": "*"
}
]

```

Tindakan untuk mendaftarkan cluster dan ruang nama tanpa server ke AWS Glue Data Catalog

Tindakan yang terdaftar dengan glue awalan berkaitan dengan mengakses katalog yang AWS Glue Data Catalog dibuat dari mendaftarkan cluster yang disediakan atau ruang nama tanpa server. Untuk informasi selengkapnya, lihat [kompatibilitas Apache Iceberg untuk Amazon Redshift di Panduan Pengembang Database](#) Amazon Redshift.

JSON berikut menunjukkan tindakan dan cakupan sumber daya, ke Amazon Redshift, untuk mengakses katalog di: AWS Glue Data Catalog

```

[
  {
    "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:GetCatalogs` izin `glue:GetCatalog` dan memiliki kondisi `glue:EnabledForRedshiftAutoDiscovery:true`, yang berarti Amazon Redshift memberikan akses IAM untuk menemukan katalog secara otomatis. Untuk memilih keluar, tambahkan kebijakan sumber daya AWS Glue tingkat akun untuk secara selektif menolak akses peran terkait layanan ke katalog. Karena peran terkait layanan sudah memiliki tindakan izin eksplisit dalam kebijakan, kebijakan opt-out perlu secara eksplisit menolak tindakan tersebut. Pertimbangkan contoh berikut, di mana kebijakan tambahan menolak penemuan otomatis untuk Amazon Redshift:

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>/*"
    ]
  }
}
```

Untuk mengizinkan entitas IAM membuat peran terkait AWSService RoleForRedshift layanan

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

Untuk mengizinkan entitas IAM menghapus peran terkait AWSService RoleForRedshift layanan

Tambahkan pernyataan kebijakan berikut ini ke izin untuk entitas IAM:

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

Atau, Anda dapat menggunakan kebijakan AWS terkelola untuk [menyediakan akses penuh](#) ke Amazon Redshift.

Membuat peran terkait layanan untuk Amazon Redshift

Anda tidak perlu membuat peran AWSService RoleForRedshift terkait layanan secara manual. Amazon Redshift menciptakan peran terkait layanan untuk Anda. Jika peran AWSService RoleForRedshift terkait layanan telah dihapus dari akun Anda, Amazon Redshift akan membuat peran tersebut saat Anda meluncurkan klaster Amazon Redshift baru.

Important

Jika Anda menggunakan layanan Amazon Redshift sebelum 18 September 2017, ketika mulai mendukung peran terkait layanan, maka Amazon Redshift membuat peran di akun Anda. AWSService RoleForRedshift Untuk mempelajari lebih lanjut, lihat [Peran baru muncul di akun IAM saya](#).

Mengedit peran terkait layanan untuk Amazon Redshift

Amazon Redshift tidak memungkinkan Anda mengedit peran terkait AWSService RoleForRedshift layanan. Setelah membuat peran terkait layanan, Anda tidak dapat mengubah nama peran karena berbagai entitas mungkin mereferensikan peran tersebut. Namun, Anda dapat mengedit deskripsi peran menggunakan konsol IAM, AWS Command Line Interface (AWS CLI), atau IAM API. Untuk informasi selengkapnya, lihat [Memodifikasi peran](#) dalam Panduan Pengguna IAM.

Menghapus peran terkait layanan untuk Amazon Redshift

Jika Anda tidak perlu lagi menggunakan fitur atau layanan yang memerlukan peran terkait layanan, kami merekomendasikan Anda menghapus peran tersebut. Dengan begitu, Anda tidak memiliki entitas yang tidak digunakan yang tidak dipantau atau dipelihara secara aktif.

Sebelum Anda dapat menghapus peran terkait layanan untuk akun, Anda harus mematikan dan menghapus klaster apa pun di akun tersebut. Untuk informasi selengkapnya, lihat [Mematikan dan menghapus cluster](#).

Anda dapat menggunakan konsol IAM, API IAM AWS CLI, atau IAM untuk menghapus peran terkait layanan. Untuk informasi selengkapnya, lihat [Menghapus peran tertaut layanan](#) dalam Panduan Pengguna IAM.

Menggunakan otentikasi IAM untuk menghasilkan kredensi pengguna basis data

Anda dapat membuat kredensial database sementara berdasarkan izin yang diberikan melalui kebijakan izin AWS Identity and Access Management (IAM) untuk mengelola akses yang dimiliki pengguna ke database Amazon Redshift Anda.

Umumnya, pengguna database Amazon Redshift masuk ke database dengan memberikan nama pengguna dan kata sandi database. Namun, Anda tidak perlu mempertahankan nama pengguna dan kata sandi di database Amazon Redshift Anda. Sebagai alternatif, Anda dapat mengonfigurasi sistem Anda untuk mengizinkan pengguna membuat kredensial pengguna dan masuk ke database berdasarkan kredensial IAM mereka.

Amazon Redshift menyediakan operasi [GetClusterCredentials](#) API untuk menghasilkan kredensial pengguna database sementara. Anda dapat mengonfigurasi klien SQL Anda dengan driver Amazon Redshift JDBC atau ODBC yang mengelola proses pemanggilan operasi. `GetClusterCredentials` Mereka melakukannya dengan mengambil kredensial pengguna database, dan membuat koneksi antara klien SQL Anda dan database Amazon Redshift Anda. Anda juga dapat menggunakan aplikasi database Anda untuk memanggil `GetClusterCredentials` operasi secara terprogram, mengambil kredensial pengguna database, dan terhubung ke database.

Jika Anda sudah mengelola identitas pengguna di luar AWS, Anda dapat menggunakan penyedia identitas (IDP) yang sesuai dengan Security Assertion Markup Language (SAMP) 2.0 untuk mengelola akses ke sumber daya Amazon Redshift. Anda mengonfigurasi IDP Anda untuk mengizinkan pengguna federasi Anda mengakses peran IAM. Dengan peran IAM tersebut, Anda dapat menghasilkan kredensial database sementara dan masuk ke database Amazon Redshift.

Klien SQL Anda memerlukan izin untuk memanggil `GetClusterCredentials` operasi untuk Anda. Anda mengelola izin tersebut dengan membuat peran IAM dan melampirkan kebijakan izin IAM yang memberikan atau membatasi akses ke operasi dan tindakan terkait. `GetClusterCredentials` Sebagai praktik terbaik, kami menyarankan untuk melampirkan kebijakan izin ke peran IAM dan

kemudian menentukannya ke pengguna dan grup sesuai kebutuhan. Untuk informasi selengkapnya, lihat [Manajemen identitas dan akses di Amazon Redshift](#).

Kebijakan ini juga memberikan atau membatasi akses ke sumber daya tertentu, seperti klaster Amazon Redshift, database, nama pengguna database, dan nama grup pengguna.

Note

Sebaiknya gunakan driver Amazon Redshift JDBC atau ODBC untuk mengelola proses panggilan `GetClusterCredentials` operasi dan masuk ke database. Untuk mempermudah, kami berasumsi bahwa Anda menggunakan klien SQL dengan driver JDBC atau ODBC di seluruh topik ini.

Untuk detail spesifik dan contoh penggunaan `GetClusterCredentials` operasi atau perintah `get-cluster-credentials` CLI paralel, lihat [GetClusterCredentials](#) dan [get-cluster-credentials](#).

Untuk mengelola otentikasi dan otorisasi secara terpusat, Amazon Redshift mendukung otentikasi database dengan IAM, memungkinkan otentikasi pengguna melalui federasi perusahaan. Alih-alih membuat pengguna, Anda dapat menggunakan identitas yang ada dari AWS Directory Service, direktori pengguna perusahaan Anda, atau penyedia identitas web. Ini dikenal sebagai pengguna federasi. AWS menetapkan peran untuk pengguna federasi ketika akses diminta melalui iDP.

Untuk menyediakan akses gabungan ke aplikasi pengguna atau klien di organisasi Anda untuk memanggil operasi Amazon Redshift API, Anda juga dapat menggunakan driver JDBC atau ODBC dengan dukungan SAMP 2.0 untuk meminta autentikasi dari IDP organisasi Anda. Dalam hal ini, pengguna organisasi Anda tidak memiliki akses langsung ke Amazon Redshift.

Untuk informasi selengkapnya, lihat [Penyedia Identitas dan Federasi](#) di Panduan Pengguna IAM.

Membuat kredensi IAM sementara

Di bagian ini, Anda dapat menemukan cara mengkonfigurasi sistem Anda untuk menghasilkan kredensial pengguna basis data berbasis IAM sementara dan masuk ke database Anda menggunakan kredensial baru.

Pada tingkat tinggi, proses mengalir sebagai berikut:

1. [Langkah 1: Buat peran IAM untuk akses masuk tunggal IAM](#)

(Opsional) Anda dapat mengautentikasi pengguna untuk mengakses database Amazon Redshift dengan mengintegrasikan autentikasi IAM dan penyedia identitas pihak ketiga (iDP).

2. [Langkah 2: Konfigurasi pernyataan SAMP untuk IDP Anda](#)

(Opsional) Untuk menggunakan autentikasi IAM menggunakan iDP, Anda perlu menentukan aturan klaim dalam aplikasi iDP yang memetakan pengguna atau grup di organisasi Anda ke peran IAM. Secara opsional, Anda dapat menyertakan elemen atribut untuk mengatur `GetClusterCredentials` parameter.

3. [Langkah 3: Buat peran IAM dengan izin untuk memanggil `GetClusterCredentialsWith IAM` atau `GetClusterCredentials`](#)

Aplikasi klien SQL Anda mengasumsikan pengguna saat memanggil operasi.

`GetClusterCredentials` Jika Anda membuat peran IAM untuk akses penyedia identitas, Anda dapat menambahkan izin yang diperlukan ke peran tersebut.

4. [Langkah 4: Buat basis data pengguna dan grup basis data](#)

(Opsional) Secara default, `GetClusterCredentials` mengembalikan kredensi membuat pengguna baru jika nama pengguna tidak ada. Anda juga dapat memilih untuk menentukan grup pengguna yang bergabung dengan pengguna saat masuk. Secara default, pengguna database bergabung dengan grup PUBLIC.

5. [Langkah 5: Konfigurasi koneksi JDBC atau ODBC untuk menggunakan kredensial IAM](#)

Untuk terhubung ke database Amazon Redshift, Anda mengonfigurasi klien SQL untuk menggunakan driver Amazon Redshift JDBC atau ODBC.

Langkah 1: Buat peran IAM untuk akses masuk tunggal IAM

Jika Anda tidak menggunakan penyedia identitas untuk akses masuk tunggal, Anda dapat melewati langkah ini.

Jika Anda sudah mengelola identitas pengguna di luar AWS, Anda dapat mengautentikasi pengguna untuk mengakses database Amazon Redshift dengan mengintegrasikan autentikasi IAM dan penyedia identitas SAML-2.0 pihak ketiga (iDP).

Untuk informasi selengkapnya, lihat [Penyedia Identitas dan Federasi](#) di Panduan Pengguna IAM.

Sebelum Anda dapat menggunakan autentikasi Amazon Redshift IDP, buat penyedia identitas SAMP. AWS Anda membuat iDP di konsol IAM untuk menginformasikan tentang iDP AWS dan

konfigurasinya. Melakukan hal ini membangun kepercayaan antara AWS akun Anda dan IDP. Untuk langkah-langkah membuat peran, lihat [Membuat Peran untuk Federasi SAMP 2.0 \(Konsol\)](#) di Panduan Pengguna IAM.

Langkah 2: Konfigurasi pernyataan SAMP untuk IDP Anda

Setelah membuat peran IAM, Anda menentukan aturan klaim di aplikasi IDP untuk memetakan pengguna atau grup di organisasi Anda ke peran IAM. Untuk informasi selengkapnya, lihat [Mengonfigurasi Pernyataan SAMP untuk Respons Otentikasi](#) di Panduan Pengguna IAM.

Jika Anda memilih untuk menggunakan `GetClusterCredentials` parameter opsional `DbUser`, `AutoCreate`, dan `DbGroups`, Anda memiliki dua opsi. Anda dapat mengatur nilai untuk parameter dengan koneksi JDBC atau ODBC Anda, atau Anda dapat mengatur nilai dengan menambahkan elemen atribut SAMP ke IDP Anda. Untuk informasi tentang parameter `DbUser`, `AutoCreate` dan `DbGroups`, lihat [Langkah 5: Konfigurasi koneksi JDBC atau ODBC untuk menggunakan kredensial IAM](#).

Note

Jika Anda menggunakan variabel kebijakan `IAM${redshift:DbUser}`, seperti yang dijelaskan dalam [Kebijakan sumber daya untuk GetClusterCredentials](#) nilai for `DbUser` diganti dengan nilai yang diambil oleh konteks permintaan operasi API. Driver Amazon Redshift menggunakan nilai untuk `DbUser` variabel yang disediakan oleh URL koneksi, bukan nilai yang diberikan sebagai atribut SAMP.

Untuk membantu mengamankan konfigurasi ini, sebaiknya gunakan kondisi dalam kebijakan IAM untuk memvalidasi `DbUser` nilai dengan menggunakan `RoleSessionName` Anda dapat menemukan contoh cara menyetel kondisi menggunakan kebijakan IAM di [Contoh 8: Kebijakan IAM untuk menggunakan GetClusterCredentials](#).

Untuk mengkonfigurasi IDP Anda untuk mengatur,, dan `DbGroups` parameter `DbUserAutoCreate`, sertakan elemen-elemen berikut: `Attribute`

- `AttributeElement` dengan `Name` atribut diatur ke "Atributhttps://redshift.amazon.com/SAML//`DbUser`"

Atur `AttributeValue` elemen ke nama pengguna yang akan terhubung ke database Amazon Redshift.

Nilai dalam `AttributeValue` elemen harus huruf kecil, dimulai dengan huruf, hanya berisi karakter alfanumerik, garis bawah ('_'), tanda plus ('+'), titik ('.'), at ('@'), atau tanda hubung ('-'), dan kurang dari 128 karakter. Biasanya, nama pengguna adalah ID pengguna (misalnya, bobsmith) atau alamat email (misalnya bobsmith@example.com). Nilai tidak dapat menyertakan spasi (misalnya, nama tampilan pengguna seperti Bob Smith).

```
<Attribute Name="https://redshift.amazon.com/SAML/Attributes/DbUser">
  <AttributeValue>user-name</AttributeValue>
</Attribute>
```

- Elemen Atribut dengan atribut Name diatur ke "https://redshift.amazon.com/SAML/Attributes/AutoCreate"

Tetapkan `AttributeValue` elemen ke true untuk membuat pengguna database baru jika tidak ada. Setel `AttributeValue` ke false untuk menentukan bahwa pengguna database harus ada di database Amazon Redshift.

```
<Attribute Name="https://redshift.amazon.com/SAML/Attributes/AutoCreate">
  <AttributeValue>true</AttributeValue>
</Attribute>
```

- `AttributeElement` dengan Name atribut diatur untuk disetel ke "https://redshift.amazon.com/SAML/Attributes/DbGroups"

Elemen ini mengandung satu atau lebih `AttributeValue` elemen. Tetapkan setiap `AttributeValue` elemen ke nama grup database yang `DbUser` bergabung selama durasi sesi saat menghubungkan ke database Amazon Redshift.

```
<Attribute Name="https://redshift.amazon.com/SAML/Attributes/DbGroups">
  <AttributeValue>group1</AttributeValue>
  <AttributeValue>group2</AttributeValue>
  <AttributeValue>group3</AttributeValue>
</Attribute>
```

Langkah 3: Buat peran IAM dengan izin untuk memanggil `GetClusterCredentialsWithIAM` atau `GetClusterCredentials`

Klien SQL Anda memerlukan otorisasi untuk memanggil `GetClusterCredentialsWithIAM` atau `GetClusterCredentials` operasi atas nama Anda. Untuk memberikan otorisasi tersebut, Anda

membuat pengguna atau peran dan melampirkan kebijakan yang memberikan izin yang diperlukan. Kedua operasi tersedia untuk mendapatkan kredensi cluster, tetapi mereka berbeda dalam metode otentikasi mereka. `GetClusterCredentialsWithIAM` menggunakan peran IAM, secara otomatis membuat pengguna database yang memetakan ke peran, yang bermanfaat untuk mengelola izin di tingkat peran IAM, sementara `GetClusterCredentials` memberikan kredensial untuk nama pengguna tertentu dalam database.

Untuk membuat peran IAM dengan izin untuk memanggil `GetClusterCredentialsWithIAM`

1. Menggunakan layanan IAM, buat pengguna atau peran. Anda juga dapat menggunakan pengguna atau peran yang ada. Misalnya, jika Anda membuat peran IAM untuk akses penyedia identitas, Anda dapat melampirkan kebijakan IAM yang diperlukan ke peran tersebut.
2. Lampirkan kebijakan izin dengan izin untuk memanggil `redshift:GetClusterCredentialsWithIAM` operasi. Contoh kebijakan berikut menunjukkan opsi untuk mengizinkan operasi untuk klaster dan database tertentu, database apa pun di klaster, dan database apa pun di klaster mana pun.

```
{
  "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": "*"
    }
  ]
}
```

```

    }
  ]
}
```

Untuk membuat peran IAM dengan izin untuk memanggil `GetClusterCredentials`

1. Menggunakan layanan IAM, buat pengguna atau peran. Anda juga dapat menggunakan pengguna atau peran yang ada. Misalnya, jika Anda membuat peran IAM untuk akses penyedia identitas, Anda dapat melampirkan kebijakan IAM yang diperlukan ke peran tersebut.
2. Lampirkan kebijakan izin dengan izin untuk memanggil `redshift:GetClusterCredentials` operasi. Bergantung pada parameter opsional yang Anda tentukan, Anda juga dapat mengizinkan atau membatasi tindakan dan sumber daya tambahan dalam kebijakan Anda:
 - Untuk mengizinkan klien SQL Anda mengambil ID cluster, AWS Region, dan port, sertakan izin untuk memanggil `redshift:DescribeClusters` operasi dengan sumber daya cluster Redshift.
 - Jika Anda menggunakan `AutoCreate` opsi, sertakan izin untuk menelepon `redshift>CreateClusterUser` dengan `dbuser` sumber daya. Nama Sumber Daya Amazon (ARN) berikut menentukan Amazon Redshift. `dbuser` Ganti *regionaccount-id*, dan *cluster-name* dengan nilai untuk AWS Wilayah, akun, dan klaster Anda. Untuk *dbuser-name*, tentukan nama pengguna yang akan digunakan untuk masuk ke database cluster.

```
arn:aws:redshift:region:account-id:dbuser:cluster-name/dbuser-name
```

- (Opsional) Tambahkan ARN yang menentukan dbname resource Amazon Redshift dalam format berikut. Ganti *regionaccount-id*, dan *cluster-name* dengan nilai untuk AWS Wilayah, akun, dan klaster Anda. Untuk *database-name*, tentukan nama database tempat pengguna akan masuk.

```
arn:aws:redshift:region:account-id:dbname:cluster-name/database-name
```

- Jika Anda menggunakan `DbGroups` opsi, sertakan izin untuk memanggil `redshift:JoinGroup` operasi dengan `dbgroup` sumber daya Amazon Redshift dalam format berikut. Ganti *regionaccount-id*, dan *cluster-name* dengan nilai untuk AWS

Wilayah, akun, dan klaster Anda. Untuk *dbgroup-name*, tentukan nama grup pengguna yang bergabung dengan pengguna saat login.

```
arn:aws:redshift:region:account-id:dbgroup:cluster-name/dbgroup-name
```

Untuk informasi selengkapnya dan contoh tambahan, lihat [Kebijakan sumber daya untuk GetClusterCredentials](#).

Langkah 4: Buat basis data pengguna dan grup basis data

Secara opsional, Anda dapat membuat pengguna database yang Anda gunakan untuk masuk ke database cluster. Jika Anda membuat kredensi pengguna sementara untuk pengguna yang sudah ada, Anda dapat menonaktifkan kata sandi pengguna untuk memaksa pengguna masuk dengan kata sandi sementara. Atau, Anda dapat menggunakan opsi `GetClusterCredentials Autocreate` untuk secara otomatis membuat pengguna database baru.

Anda dapat membuat grup pengguna database dengan izin yang Anda inginkan agar pengguna database IAM bergabung saat login. Saat Anda memanggil `GetClusterCredentials` operasi, Anda dapat menentukan daftar nama grup pengguna yang bergabung dengan pengguna baru saat login. Keanggotaan grup ini hanya berlaku untuk sesi yang dibuat menggunakan kredensial yang dihasilkan dengan permintaan yang diberikan.

Untuk membuat pengguna basis data dan grup basis data

1. [Masuk ke database Amazon Redshift Anda dan buat pengguna database menggunakan `CREATE USER` atau ubah pengguna yang ada menggunakan `ALTER USER`.](#)
2. Secara opsional, tentukan opsi `DISABLE PASSWORD` untuk mencegah pengguna menggunakan kata sandi. Ketika kata sandi pengguna dinonaktifkan, pengguna dapat masuk hanya menggunakan kredensial sementara. Jika kata sandi tidak dinonaktifkan, pengguna dapat masuk dengan kata sandi atau menggunakan kredensial sementara. Anda tidak dapat menonaktifkan kata sandi untuk pengguna super.

Pengguna membutuhkan akses terprogram jika mereka ingin berinteraksi dengan AWS luar. Konsol Manajemen AWS Cara untuk memberikan akses terprogram tergantung pada jenis pengguna yang mengakses AWS.

Untuk memberi pengguna akses programatis, pilih salah satu opsi berikut.

Pengguna mana yang membutuhkan akses programatis?	Untuk	Oleh
IAM	(Disarankan) Gunakan kredensial konsol sebagai kredensial sementara untuk menandatangani permintaan terprogram ke,, atau. AWS CLI AWS SDKs AWS APIs	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan. • Untuk itu AWS CLI, lihat Login untuk pengembangan AWS lokal di Panduan AWS Command Line Interface Pengguna. • Untuk AWS SDKs, lihat Login untuk pengembangan AWS lokal di Panduan Referensi Alat AWS SDKs dan Alat.
Identitas tenaga kerja (Pengguna yang dikelola di Pusat Identitas IAM)	Gunakan kredensial sementara untuk menandatangani permintaan terprogram ke AWS CLI,, AWS SDKs atau. AWS APIs	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan. • Untuk AWS CLI, lihat Mengkonfigurasi yang akan AWS CLI digunakan AWS IAM Identity Center dalam Panduan AWS Command Line Interface Pengguna. • Untuk AWS SDKs, alat, dan AWS APIs, lihat Autentikasi Pusat Identitas IAM di Panduan Referensi Alat AWS SDKs dan Alat.

Pengguna mana yang membutuhkan akses programatis?	Untuk	Oleh
IAM	Gunakan kredensi sementara untuk menandatangani permintaan terprogram ke AWS CLI,, AWS SDKs atau. AWS APIs	Mengikuti petunjuk dalam Menggunakan kredensial sementara dengan AWS sumber daya di Panduan Pengguna IAM.
IAM	(Tidak direkomendasikan) Gunakan kredensi jangka panjang untuk menandatangani permintaan terprogram ke AWS CLI,, AWS SDKs atau. AWS APIs	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan. • Untuk mengetahui AWS CLI, lihat Mengautentikasi menggunakan kredensial pengguna IAM di Panduan Pengguna. AWS Command Line Interface • Untuk AWS SDKs dan alat, lihat Mengautentikasi menggunakan kredensial jangka panjang di Panduan Referensi Alat AWS SDKs dan Alat. • Untuk AWS APIs, lihat Mengelola kunci akses untuk pengguna IAM di Panduan Pengguna IAM.

Contoh berikut membuat pengguna dengan password dinonaktifkan.

```
create user temp_creds_user password disable;
```

Contoh berikut menonaktifkan password untuk pengguna yang sudah ada.

```
alter user temp_creds_user password disable;
```

3. Buat grup pengguna database menggunakan [CREATE GROUP](#).
4. Gunakan perintah [GRANT](#) untuk menentukan hak akses untuk grup.

Langkah 5: Konfigurasi koneksi JDBC atau ODBC untuk menggunakan kredensial IAM

Anda dapat mengonfigurasi klien SQL Anda dengan driver Amazon Redshift JDBC atau ODBC. Driver ini mengelola proses pembuatan kredensial pengguna database dan membuat koneksi antara klien SQL Anda dan database Amazon Redshift Anda.

Jika Anda menggunakan penyedia identitas untuk otentikasi, tentukan nama plugin penyedia kredensial. Driver Amazon Redshift JDBC dan ODBC menyertakan plugin untuk penyedia identitas berbasis SAMP berikut:

- Microsoft Active Directory Federation Services (AD FS)
- PingOne
- Okta
- Microsoft Azure AD

Untuk langkah-langkah mengatur Microsoft Azure AD sebagai penyedia identitas, lihat [Menyiapkan otentikasi masuk tunggal JDBC atau ODBC](#).

Untuk mengonfigurasi koneksi JDBC untuk menggunakan kredensial IAM

1. Unduh driver Amazon Redshift JDBC terbaru dari halaman. [Mengonfigurasi koneksi untuk driver JDBC versi 2.x untuk Amazon Redshift](#)
2. Buat URL JDBC dengan opsi kredensial IAM dalam salah satu format berikut. Untuk menggunakan autentikasi IAM, tambahkan `iam:` ke URL Amazon Redshift JDBC `jdbc:redshift:` berikut seperti yang ditunjukkan pada contoh berikut.

```
jdbc:redshift:iam://
```

Tambahkan `cluster-name`, `region`, dan `account-id`. Driver JDBC menggunakan informasi akun IAM dan nama cluster Anda untuk mengambil ID cluster dan Region. AWS Untuk melakukannya, pengguna atau peran Anda harus memiliki izin untuk memanggil

`redshift:DescribeClusters` operasi dengan cluster yang ditentukan. Jika pengguna atau peran Anda tidak memiliki izin untuk memanggil `redshift:DescribeClusters` operasi, sertakan ID cluster, AWS Wilayah, dan port seperti yang ditunjukkan pada contoh berikut. Nomor port adalah opsional.

```
jdbc:redshift:iam://examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com:5439/dev
```

3. Tambahkan opsi JDBC untuk memberikan kredensial IAM. Anda menggunakan kombinasi opsi JDBC yang berbeda untuk memberikan kredensial IAM. Lihat perinciannya di [Opsi JDBC dan ODBC untuk membuat kredensial pengguna basis data](#).

URL berikut menentukan AccessKey ID dan SecretAccessKey untuk pengguna.

```
jdbc:redshift:iam://examplecluster:us-west-2/dev?AccessKeyID=AKIAIOSFODNN7EXAMPLE&SecretAccessKey=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

Contoh berikut menentukan profil bernama yang berisi kredensial IAM.

```
jdbc:redshift:iam://examplecluster:us-west-2/dev?Profile=user2
```

4. Tambahkan opsi JDBC yang digunakan driver JDBC untuk memanggil operasi API. `GetClusterCredentials` Jangan sertakan opsi ini jika Anda memanggil operasi `GetClusterCredentials` API secara terprogram.

Contoh berikut mencakup opsi `JDBCGetClusterCredentials`.

```
jdbc:redshift:iam://examplecluster:us-west-2/dev?plugin_name=com.amazon.redshift.plugin.AzureCredentialsProvider&UID=user&PWD=password&idp_t
```

Untuk mengonfigurasi koneksi ODBC untuk menggunakan kredensial IAM

Dalam prosedur berikut, Anda dapat menemukan langkah-langkah hanya untuk mengkonfigurasi otentikasi IAM. Untuk langkah-langkah menggunakan otentikasi standar, menggunakan nama pengguna dan kata sandi database, lihat [Mengonfigurasi koneksi untuk driver ODBC versi 2.x untuk Amazon Redshift](#).

1. Instal dan konfigurasi driver Amazon Redshift ODBC terbaru untuk sistem operasi Anda. Untuk informasi lebih lanjut, lihat [Mengonfigurasi koneksi untuk driver ODBC versi 2.x untuk Amazon Redshift](#) halaman.

 Important

Driver Amazon Redshift ODBC harus versi 1.3.6.1000 atau yang lebih baru.

2. Ikuti langkah-langkah untuk sistem operasi Anda untuk mengonfigurasi pengaturan koneksi.
3. Pada sistem operasi Microsoft Windows, akses jendela Pengaturan DSN Driver Amazon Redshift ODBC.

a. Di bawah Pengaturan Koneksi, masukkan informasi berikut:

- Nama Sumber Data
- Server (opsional)
- Port (opsional)
- Basis Data

Jika pengguna atau peran Anda memiliki izin untuk memanggil `redshift:DescribeClusters` operasi, hanya Nama Sumber Data dan Database yang diperlukan. Amazon Redshift menggunakan `ClusterId` dan Wilayah untuk mendapatkan server dan port dengan memanggil operasi `DescribeCluster`

Jika pengguna atau peran Anda tidak memiliki izin untuk memanggil `redshift:DescribeClusters` operasi, tentukan Server dan Port.

b. Di bawah Otentikasi, pilih nilai untuk Jenis Auth.

Untuk setiap jenis otentikasi, masukkan nilai seperti yang tercantum berikut:

AWS Profil

Masukkan informasi berikut:

- ClusterID
- Wilayah
- Nama profil

Masukkan nama profil dalam file AWS konfigurasi yang berisi nilai untuk opsi koneksi ODBC. Untuk informasi selengkapnya, lihat [Menggunakan profil konfigurasi](#).

(Opsional) Berikan detail untuk opsi yang digunakan driver ODBC untuk memanggil operasi `GetClusterCredentials` API:

- DbUser
- Pengguna AutoCreate
- DbGroups

Untuk informasi selengkapnya, lihat [Opsi JDBC dan ODBC untuk membuat kredensi pengguna basis data](#).

Kredensi IAM

Masukkan informasi berikut:

- ClusterID
- Wilayah
- AccessKeyId dan SecretAccessKey

ID kunci akses dan kunci akses rahasia untuk peran IAM atau pengguna yang dikonfigurasi untuk otentikasi database IAM.

- SessionToken

SessionToken diperlukan untuk peran IAM dengan kredensi sementara. Untuk informasi selengkapnya, lihat [Kredensial Keamanan Sementara](#).

Berikan detail untuk opsi yang digunakan driver ODBC untuk memanggil operasi `GetClusterCredentials` API:

- DbUser(Diperlukan)
- Pengguna AutoCreate (opsional)
- DbGroups (opsional)

Untuk informasi selengkapnya, lihat [Opsi JDBC dan ODBC untuk membuat kredensi pengguna basis data](#).

Penyedia Identitas: AD FS

Untuk Otentikasi Terintegrasi Windows dengan AD FS, biarkan Pengguna dan Kata Sandi kosong.

Berikan detail IDP:

- IdP Host

Nama host penyedia identitas perusahaan. Nama ini tidak boleh menyertakan garis miring (/).

- IDP Port (opsional)

Port yang digunakan oleh penyedia identitas. Default-nya adalah 443.

- Peran yang Diutamakan

Nama Sumber Daya Amazon (ARN) untuk peran IAM dari `AttributeValue` elemen multi-nilai untuk `Role` atribut dalam pernyataan SAMP. Untuk menemukan nilai yang sesuai untuk peran yang diinginkan, bekerjalah dengan administrator IDP Anda. Untuk informasi selengkapnya, lihat [Langkah 2: Konfigurasi pernyataan SAMP untuk IDP Anda](#).

(Opsional) Berikan detail untuk opsi yang digunakan driver ODBC untuk memanggil operasi `GetClusterCredentials` API:

- DbUser
- Pengguna AutoCreate
- DbGroups

Untuk informasi selengkapnya, lihat [Opsi JDBC dan ODBC untuk membuat kredensi pengguna basis data](#).

Penyedia Identitas: PingFederate

Untuk Pengguna dan Kata Sandi, masukkan nama pengguna dan kata sandi IDP Anda.

Berikan detail IDP:

- IdP Host

Nama host penyedia identitas perusahaan. Nama ini tidak boleh menyertakan garis miring (/).

- IDP Port (opsional)

Port yang digunakan oleh penyedia identitas. Default-nya adalah 443.

- Peran yang Diutamakan

Nama Sumber Daya Amazon (ARN) untuk peran IAM dari `AttributeValue` elemen multi-nilai untuk `Role` atribut dalam pernyataan SAMP. Untuk menemukan nilai yang sesuai untuk peran yang diinginkan, bekerjalah dengan administrator IDP Anda. Untuk informasi selengkapnya, lihat [Langkah 2: Konfigurasi pernyataan SAMP untuk IDP Anda](#).

(Opsional) Berikan detail untuk opsi yang digunakan driver ODBC untuk memanggil operasi `GetClusterCredentials` API:

- `DbUser`
- Pengguna `AutoCreate`
- `DbGroups`

Untuk informasi selengkapnya, lihat [Opsi JDBC dan ODBC untuk membuat kredensi pengguna basis data](#).

Penyedia Identitas: Okta

Untuk Pengguna dan Kata Sandi, masukkan nama pengguna dan kata sandi IDP Anda.

Berikan detail IDP:

- IDP Host

Nama host penyedia identitas perusahaan. Nama ini tidak boleh menyertakan garis miring (/).

- Pelabuan IDP

Nilai ini tidak digunakan oleh Okta.

- Peran yang Diutamakan

Nama Sumber Daya Amazon (ARN) untuk peran IAM dari AttributeValue elemen untuk Role atribut dalam pernyataan SAMP. Untuk menemukan nilai yang sesuai untuk peran yang diinginkan, bekerjalah dengan administrator iDP Anda. Untuk informasi selengkapnya, lihat [Langkah 2: Konfigurasi pernyataan SAMP untuk iDP Anda](#).

- ID Aplikasi Okta

ID untuk aplikasi Okta. Nilai untuk ID Aplikasi mengikuti “amazon_aws” di tautan penempatan aplikasi Okta. Bekerja dengan administrator iDP Anda untuk mendapatkan nilai ini.

(Opsional) Berikan detail untuk opsi yang digunakan driver ODBC untuk memanggil operasi GetClusterCredentials API:

- DbUser
- Pengguna AutoCreate
- DbGroups

Untuk informasi selengkapnya, lihat [Opsi JDBC dan ODBC untuk membuat kredensi pengguna basis data](#).

Penyedia Identitas: Azure AD

Untuk Pengguna dan Kata Sandi, masukkan nama pengguna dan kata sandi iDP Anda.

Untuk ID Cluster dan Region, masukkan ID cluster dan AWS Region klaster Amazon Redshift Anda.

Untuk Database, masukkan database yang Anda buat untuk cluster Amazon Redshift Anda.

Berikan detail iDP:

- Penyewa iDP

Penyewa digunakan untuk Azure AD.

- Rahasia Klien Azure

Rahasia klien dari aplikasi perusahaan Amazon Redshift di Azure.

- ID Klien Azure

ID klien (ID aplikasi) dari aplikasi perusahaan Amazon Redshift di Azure.

(Opsional) Berikan detail untuk opsi yang digunakan driver ODBC untuk memanggil operasi `GetClusterCredentials` API:

- DbUser
- Pengguna AutoCreate
- DbGroups

Untuk informasi selengkapnya, lihat [Opsi JDBC dan ODBC untuk membuat kredensi pengguna basis data](#).

Opsinya untuk memberikan kredensi IAM

Untuk memberikan kredensial IAM untuk koneksi JDBC atau ODBC, pilih salah satu opsi berikut.

- AWS profil

Sebagai alternatif untuk memberikan nilai kredensial dalam bentuk pengaturan JDBC atau ODBC, Anda dapat menempatkan nilai dalam profil bernama. Untuk informasi selengkapnya, lihat [Menggunakan profil konfigurasi](#).

- Kredensi IAM

Berikan nilai untuk `AccessKey ID`, `SecretAccessKey`, dan, secara opsional, `SessionToken` dalam bentuk pengaturan JDBC atau ODBC. `SessionToken` diperlukan hanya untuk peran IAM dengan kredensial sementara. Untuk informasi selengkapnya, lihat [Opsi JDBC dan ODBC untuk menyediakan kredensial IAM](#).

- Federasi penyedia identitas

Saat Anda menggunakan federasi penyedia identitas untuk mengaktifkan pengguna dari penyedia identitas untuk mengautentikasi ke Amazon Redshift, tentukan nama plugin penyedia kredensi. Untuk informasi selengkapnya, lihat [Plugin penyedia kredensial](#).

Driver Amazon Redshift JDBC dan ODBC menyertakan plugin untuk penyedia kredensi federasi identitas berbasis SAMP berikut:

- Layanan Federasi Identitas Aktif Microsoft (AD FS)

- PingOne
- Okta
- Microsoft Azure Active Directory (Azure AD)

Anda dapat memberikan nama plugin dan nilai terkait dalam bentuk pengaturan JDBC atau ODBC atau dengan menggunakan profil. Untuk informasi selengkapnya, lihat [Opsinya untuk konfigurasi driver JDBC versi 2.x](#).

Untuk informasi selengkapnya, lihat [Langkah 5: Konfigurasi koneksi JDBC atau ODBC untuk menggunakan kredensial IAM](#).

Menggunakan profil konfigurasi

Anda dapat menyediakan opsi dan `GetClusterCredentials` opsi kredensial IAM sebagai pengaturan dalam profil bernama di file konfigurasi Anda AWS. Untuk memberikan nama profil, gunakan opsi `Profil JDBC`. Konfigurasi disimpan dalam file bernama `config` atau file bernama `credentials` dalam folder bernama `.aws` di direktori home Anda.

Untuk plugin penyedia kredensial berbasis SAMP yang disertakan dengan driver Amazon Redshift JDBC atau ODBC, Anda dapat menggunakan pengaturan yang dijelaskan sebelumnya. [Plugin penyedia kredensial](#) Jika `plugin_name` tidak digunakan, opsi lain diabaikan.

Contoh berikut menunjukkan file `~/.aws/credentials` dengan dua profil.

```
[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
QrmGdeehM4IC1NtBmUpp2wUE8phUZampKsburEDy0KPkyQDYwT7WZ0wq5VVSXDvp75YU
9HFv1Rd8Tx6q6fE8YQcHNvXAKiY9q6d+xo0rKwT38xVqr7ZD0u0iPPkUL64lIZbqBAz
+scqKmlzm8FDrypNC9Yjc8fP0Ln9FX9KSYvKTr4rvx3iSI1TJabIQwj2ICCR/oLxBA==
```

Untuk menggunakan kredensial `user2` sebagai contoh, tentukan `Profile=user2` di URL JDBC.

Untuk informasi selengkapnya tentang penggunaan profil, lihat [Pengaturan konfigurasi dan file kredensi](#) di Panduan AWS Command Line Interface Pengguna.

Untuk informasi selengkapnya tentang penggunaan profil untuk driver JDBC, lihat. [Menentukan profil](#)

Untuk informasi selengkapnya tentang penggunaan profil untuk driver ODBC, lihat [Metode autentikasi](#).

Opsi JDBC dan ODBC untuk menyediakan kredensial IAM

Tabel berikut mencantumkan opsi JDBC dan ODBC untuk menyediakan kredensial IAM.

Opsi	Deskripsi
Iam	Untuk digunakan hanya dalam string koneksi ODBC. Setel ke 1 untuk menggunakan otentikasi IAM.
AccessKey ID	ID kunci akses dan kunci akses rahasia untuk peran IAM atau pengguna yang dikonfigurasi untuk otentikasi database IAM. SessionToken diperlukan hanya untuk peran IAM dengan kredensial sementara. SessionToken tidak digunakan untuk pengguna. Untuk informasi selengkapnya, lihat Kredensial Keamanan Sementara .
SecretAccessKey	
SessionToken	
plugin_name	Nama kelas yang sepenuhnya memenuhi syarat yang mengimplementasikan penyedia kredensial. Driver Amazon Redshift JDBC menyertakan plugin penyedia kredensi berbasis SALL. Jika Anda menyediakan plugin_name, Anda juga dapat memberikan opsi terkait lainnya. Untuk informasi selengkapnya, lihat Plugin penyedia kredensial .
Profile	Nama profil dalam file AWS kredensial atau konfigurasi yang berisi nilai untuk opsi koneksi JDBC. Untuk informasi selengkapnya, lihat Menggunakan profil konfigurasi .

Opsi JDBC dan ODBC untuk membuat kredensi pengguna basis data

Untuk menggunakan driver Amazon Redshift JDBC atau ODBC untuk membuat kredensial pengguna database, berikan nama pengguna database sebagai opsi JDBC atau ODBC. Secara opsional, Anda dapat meminta driver membuat pengguna database baru jika tidak ada, dan Anda dapat menentukan daftar grup pengguna database yang bergabung dengan pengguna saat login.

Jika Anda menggunakan penyedia identitas (iDP), bekerjalah dengan administrator iDP Anda untuk menentukan nilai yang benar untuk opsi ini. Administrator iDP Anda juga dapat mengonfigurasi iDP Anda untuk menyediakan opsi ini, dalam hal ini Anda tidak perlu menyediakannya sebagai opsi JDBC atau ODBC. Untuk informasi selengkapnya, lihat [Langkah 2: Konfigurasi pernyataan SAMP untuk iDP Anda](#).

 Note

Jika Anda menggunakan variabel kebijakan IAM `IAM${redshift:DbUser}`, seperti yang dijelaskan dalam [Kebijakan sumber daya untuk GetClusterCredentials](#) nilai `for DbUser` diganti dengan nilai yang diambil oleh konteks permintaan operasi API. Driver Amazon Redshift menggunakan nilai untuk `DbUser` variabel yang disediakan oleh URL koneksi, bukan nilai yang diberikan sebagai atribut SAMP.

Untuk membantu mengamankan konfigurasi ini, kami sarankan Anda menggunakan kondisi dalam kebijakan IAM untuk memvalidasi `DbUser` nilai dengan `RoleSessionName` Anda dapat menemukan contoh cara menyetel kondisi menggunakan kebijakan IAM di [Contoh 8: Kebijakan IAM untuk menggunakan GetClusterCredentials](#).

Tabel berikut mencantumkan opsi untuk membuat kredensi pengguna database.

Opsi	Deskripsi
DbUser	Nama pengguna database. Jika pengguna bernama <code>DbUser</code> ada dalam database, kredensial pengguna sementara memiliki izin yang sama dengan pengguna yang ada. Jika <code>DbUser</code> tidak ada dalam database dan <code>AutoCreate</code> benar, nama pengguna baru <code>DbUser</code> dibuat. Secara opsional, nonaktifkan kata sandi untuk pengguna yang ada. Untuk informasi selengkapnya, lihat ALTER_USER
AutoCreate	Tentukan <code>true</code> untuk membuat pengguna database dengan nama yang ditentukan untuk <code>DbUser</code> jika tidak ada. Default-nya adalah salah.
DbGroups	Daftar yang dibatasi koma dari nama-nama dari satu atau lebih kelompok database yang ada yang bergabung dengan pengguna database untuk sesi saat ini. Secara default, pengguna baru hanya ditambahkan ke PUBLIK.

Plugin penyedia kredensial

Amazon Redshift menggunakan plugin penyedia kredensial untuk autentikasi masuk tunggal.

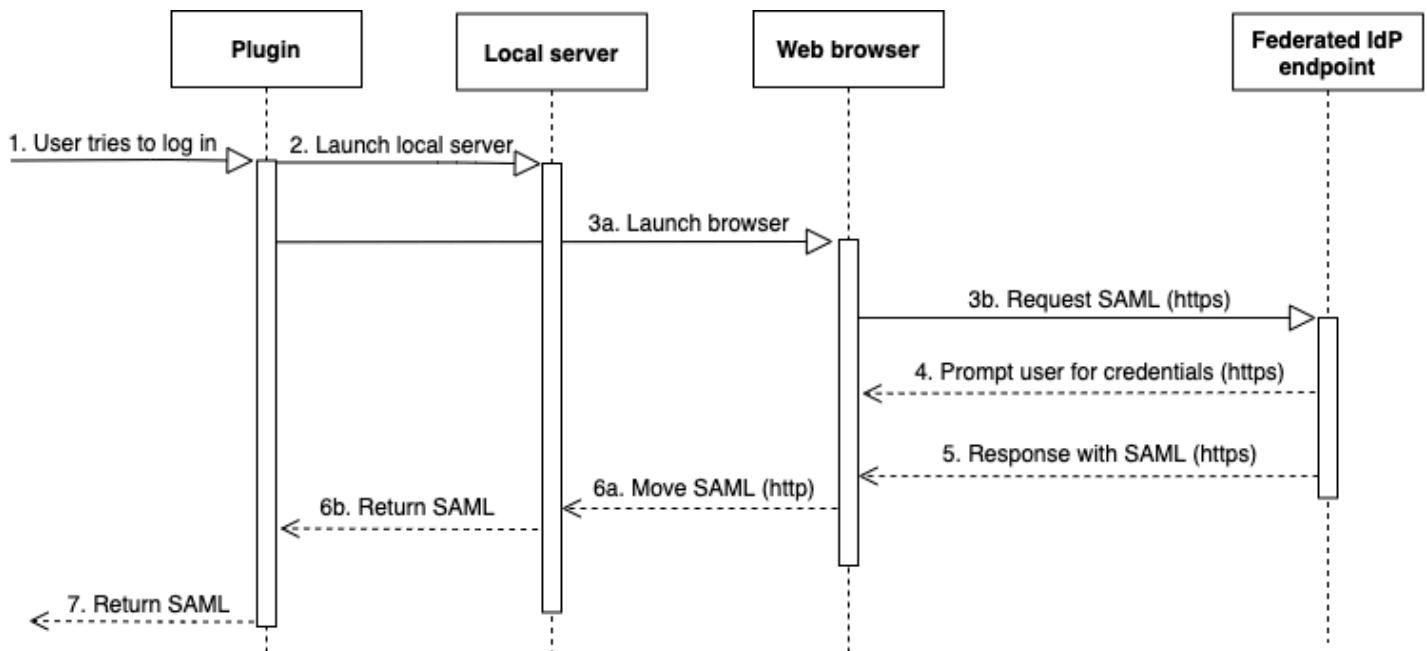
Untuk mendukung otentikasi masuk tunggal, Amazon Redshift menyediakan plugin Azure AD untuk Microsoft Azure Active Directory. Untuk informasi tentang cara mengkonfigurasi plugin ini, lihat [Menyiapkan otentikasi masuk tunggal JDBC atau ODBC](#).

Otentikasi multi-faktor

Otentikasi multi-faktor

Untuk mendukung otentikasi multi-faktor (MFA), Amazon Redshift menyediakan plugin berbasis browser. Gunakan plugin SAMP browser untuk Okta PingOne, dan plugin Azure AD browser untuk Microsoft Azure Active Directory.

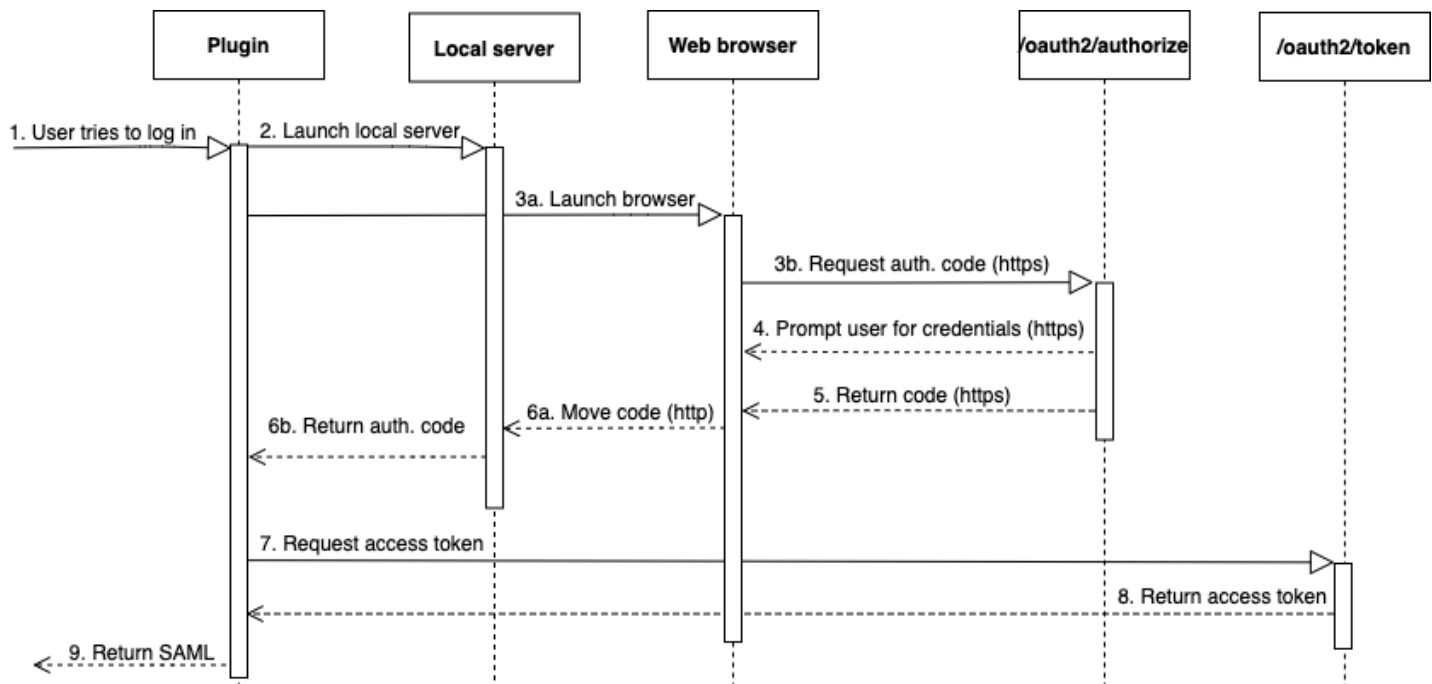
Dengan plugin SAMP browser, OAuth otentikasi mengalir seperti ini:



1. Seorang pengguna mencoba masuk.
2. Plugin meluncurkan server lokal untuk mendengarkan koneksi masuk di localhost.
3. Plugin ini meluncurkan browser web untuk meminta respons SAMP melalui HTTPS dari titik akhir penyedia identitas federasi URL login masuk tunggal yang ditentukan.
4. Browser web mengikuti tautan dan meminta pengguna untuk memasukkan kredensial.

5. Setelah pengguna mengautentikasi dan memberikan persetujuan, titik akhir penyedia identitas federasi mengembalikan respons SAMP melalui HTTPS ke URI yang ditunjukkan oleh `redirect_uri`.
6. Browser web memindahkan pesan respons dengan respons SAMP ke yang ditunjukkan `redirect_uri`.
7. Server lokal menerima koneksi masuk dan plugin mengambil respons SAMP dan meneruskannya ke Amazon Redshift.

Dengan plugin browser Azure AD, otentikasi SAMP mengalir seperti ini:



1. Seorang pengguna mencoba masuk.
2. Plugin meluncurkan server lokal untuk mendengarkan koneksi masuk di localhost.
3. Plugin meluncurkan browser web untuk meminta kode otorisasi dari titik akhir Azure AD. `oauth2/authorize`
4. Browser web mengikuti tautan yang dihasilkan melalui HTTPS dan meminta pengguna untuk memasukkan kredensial. Tautan dihasilkan menggunakan properti konfigurasi, seperti `tenant` dan `client_id`.
5. Setelah pengguna mengautentikasi dan memberikan persetujuan, `oauth2/authorize` titik akhir Azure AD mengembalikan dan mengirimkan respons melalui HTTPS dengan kode otorisasi ke yang ditunjukkan. `redirect_uri`

6. Browser web memindahkan pesan respons dengan respons SAMP ke yang ditunjukkan `redirect_uri`.
7. Server lokal menerima koneksi masuk dan permintaan plugin dan mengambil kode otorisasi dan mengirimkan permintaan POST ke titik akhir Azure AD. `oauth2/token`
8. `oauth2/token` Titik akhir Azure AD mengembalikan respons dengan token akses ke yang ditunjukkan. `redirect_uri`
9. Plugin mengambil respons SAMP dan meneruskannya ke Amazon Redshift.

Lihat bagian berikut:

- Microsoft Active Directory Federation Services (AD FS)

Untuk informasi selengkapnya, lihat [Menyiapkan otentikasi masuk tunggal JDBC atau ODBC](#).

- PingOne (Ping)

Ping hanya didukung dengan Adaptor PingOne iDP yang telah ditentukan menggunakan otentikasi Formulir.

Untuk informasi selengkapnya, lihat [Menyiapkan otentikasi masuk tunggal JDBC atau ODBC](#).

- Okta

Okta hanya didukung untuk aplikasi yang disediakan Okta yang digunakan dengan. Konsol Manajemen AWS

Untuk informasi selengkapnya, lihat [Menyiapkan otentikasi masuk tunggal JDBC atau ODBC](#).

- Direktori Aktif Microsoft Azure

Untuk informasi selengkapnya, lihat [Menyiapkan otentikasi masuk tunggal JDBC atau ODBC](#).

Opsi plugin

Opsi plugin

Untuk menggunakan plugin penyedia kredensial berbasis SAML, tentukan opsi berikut menggunakan opsi JDBC atau ODBC atau dalam profil bernama. Jika `plugin_name` tidak ditentukan, opsi lain diabaikan.

Opsi	Deskripsi
plugin_name	Untuk JDBC, nama kelas yang mengimplementasikan penyedia kredensial. Tentukan satu dari yang berikut ini: • Untuk Layanan Federasi Direktori Aktif com.amazon.redshift.plugin.AdfsCredentialsProvider • Untuk Okta com.amazon.redshift.plugin.OktaCredentialsProvider • Untuk PingFederate com.amazon.redshift.plugin.PingCredentialsProvider • Untuk Microsoft Azure Active Directory com.amazon.redshift.plugin.AzureCredentialsProvider • Untuk SAMP MFA com.amazon.redshift.plugin.BrowserSamlCredentialsProvider • Untuk Microsoft Azure Active Directory masuk tunggal dengan MFA com.amazon.redshift.plugin.BrowserAzureCredentialsProvider
	Untuk ODBC, tentukan salah satu dari berikut ini: • Untuk Layanan Federasi Direktori Aktif: <code>adfs</code> • Untuk Okta: <code>okta</code> • Untuk PingFederate: <code>ping</code> • Untuk Microsoft Azure Active Directory: <code>azure</code> • Untuk SAMP MFA: <code>browser saml</code>

Opsi	Deskripsi
	Untuk sistem masuk tunggal Microsoft Azure Active Directory dengan MFA: <code>browser azure ad</code>
<code>idp_host</code>	Nama host penyedia identitas perusahaan. Nama ini tidak boleh menyertakan garis miring ('/'). Untuk penyedia identitas Okta, nilai untuk <code>idp_host</code> harus diakhiri dengan <code>.okta.com</code> .
<code>idp_port</code>	Port yang digunakan oleh penyedia identitas. Default-nya adalah 443. Port ini diabaikan untuk Okta.
<code>preferred_role</code>	Peran Amazon Resource Name (ARN) dari <code>AttributeValue</code> elemen untuk <code>Role</code> atribut dalam pernyataan SAMP. Untuk menemukan nilai yang sesuai untuk peran yang diinginkan, bekerjalah dengan administrator iDP Anda. Untuk informasi selengkapnya, lihat Langkah 2: Konfigurasi pernyataan SAMP untuk IDP Anda .
<code>user</code>	Nama pengguna perusahaan, termasuk domain bila berlaku. Misalnya, untuk Active Directory, nama domain diperlukan dalam format <code>domain\username</code> .
<code>password</code>	Kata sandi pengguna perusahaan. Kami menyarankan untuk tidak menggunakan opsi ini. Sebagai gantinya, gunakan klien SQL Anda untuk memasok kata sandi.
<code>app_id</code>	ID untuk aplikasi Okta. Hanya digunakan dengan Okta. Nilai untuk <code>app_id</code> mengikuti <code>amazon_aws</code> di tautan embed aplikasi Okta. Untuk mendapatkan nilai ini, bekerja dengan administrator iDP Anda. Berikut ini adalah contoh link embed aplikasi: <code>https://example.okta.com/home/amazon_aws/0oa2hylwrpM8UGehd1t7/272</code>
<code>idp_tenant</code>	Penyewa yang digunakan untuk Azure AD. Digunakan hanya dengan Azure.
<code>client_id</code>	ID klien untuk aplikasi perusahaan Amazon Redshift di Azure AD. Digunakan hanya dengan Azure.

Membuat kredensial database untuk identitas IAM menggunakan Amazon Redshift CLI atau API

Untuk menghasilkan kredensial pengguna database sementara secara terprogram, Amazon Redshift menyediakan [get-cluster-credentials](#) perintah untuk AWS Command Line Interface

()AWS CLI dan operasi API. [GetClusterCredentials](#) Atau Anda dapat mengonfigurasi klien SQL Anda dengan driver Amazon Redshift JDBC atau ODBC yang mengelola proses pemanggilan `GetClusterCredentials` operasi, mengambil kredensi pengguna database, dan membuat koneksi antara klien SQL Anda dan database Amazon Redshift Anda. Untuk informasi selengkapnya, lihat [Opsis JDBC dan ODBC untuk membuat kredensi pengguna basis data](#).

 Note

Sebaiknya gunakan driver Amazon Redshift JDBC atau ODBC untuk menghasilkan kredensi pengguna database.

Di bagian ini, Anda dapat menemukan langkah-langkah untuk memanggil `GetClusterCredentials` operasi atau `get-cluster-credentials` perintah secara terprogram, mengambil kredensi pengguna database, dan terhubung ke database.

Untuk menghasilkan dan menggunakan kredensial basis data sementara

1. Membuat atau memodifikasi pengguna atau peran dengan izin yang diperlukan. Untuk informasi selengkapnya tentang izin IAM, lihat. [Langkah 3: Buat peran IAM dengan izin untuk memanggil `GetClusterCredentialsWith IAM` atau `GetClusterCredentials`](#)
2. Sebagai pengguna atau peran yang Anda otorisasi pada langkah sebelumnya, jalankan perintah `get-cluster-credentials` CLI atau panggil operasi `GetClusterCredentials` API dan berikan nilai-nilai berikut:
 - Cluster identifier — Nama cluster yang berisi database.
 - Nama pengguna database — Nama pengguna database yang sudah ada atau baru.
 - Jika pengguna tidak ada dalam database dan `AutoCreate` benar, pengguna baru dibuat dengan `PASSWORD` dinonaktifkan.
 - Jika pengguna tidak ada, dan `AutoCreate` salah, permintaan gagal.
 - Untuk contoh ini, nama pengguna database adalah `temp_creds_user`.
 - `Autocreate` — (Opsional) Buat pengguna baru jika nama pengguna database tidak ada.
 - Nama basis data — (Opsional) Nama database yang diizinkan pengguna untuk masuk. Jika nama database tidak ditentukan, pengguna dapat masuk ke database cluster apa pun.
 - Grup basis data — (Opsional) Daftar grup pengguna database yang ada. Setelah login berhasil, pengguna database ditambahkan ke grup pengguna yang ditentukan. Jika tidak

ada grup yang ditentukan, pengguna hanya memiliki izin PUBLIK. Nama grup pengguna harus cocok dengan sumber daya dbgroup yang ARNs ditentukan dalam kebijakan IAM yang dilampirkan pada pengguna atau peran.

- Waktu kedaluwarsa — (Opsional) Waktu, dalam hitungan detik, hingga kredensial sementara kedaluwarsa. Anda dapat menentukan nilai antara 900 detik (15 menit) dan 3600 detik (60 menit). Defaultnya adalah 900 detik.
3. Amazon Redshift memverifikasi bahwa pengguna memiliki izin untuk memanggil `GetClusterCredentials` operasi dengan sumber daya yang ditentukan.
 4. Amazon Redshift mengembalikan kata sandi sementara dan nama pengguna database.

Contoh berikut menggunakan Amazon Redshift CLI untuk menghasilkan kredensial database sementara untuk pengguna yang sudah ada bernama `temp_creds_user`

```
aws redshift get-cluster-credentials --cluster-identifier examplecluster --db-user temp_creds_user --db-name exampledb --duration-seconds 3600
```

Hasilnya adalah sebagai berikut.

```
{
  "DbUser": "IAM:temp_creds_user",
  "Expiration": "2016-12-08T21:12:53Z",
  "DbPassword": "EXAMPLEjArE3hcnQj8zt4XQj9Xtma8oxYEM80yxpDHwXVPyJYBDm/
gqX2Eeaq6P3DgTzgPg=="
}
```

Contoh berikut menggunakan Amazon Redshift CLI dengan `autocreate` untuk menghasilkan kredensial database sementara untuk pengguna baru dan menambahkan pengguna ke grup `example_group`

```
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
```

Hasilnya adalah sebagai berikut.

```
{
  "DbUser": "IAMA:temp_creds_user:example_group",
  "Expiration": "2016-12-08T21:12:53Z",
}
```



```
"DbPassword": "EXAMPLEjArE3hcnQj8zt4XQj9Xtma8oxYEM80yxpDHwXVPyJYBDm/  
gqX2Eeaq6P3DgTzgPg=="  
}
```

5. Buat koneksi otentikasi Secure Socket Layer (SSL) dengan cluster Amazon Redshift dan kirim permintaan login dengan nama pengguna dan kata sandi dari respons. `GetClusterCredentials` Sertakan `IAMA:` awalan `IAM:` atau dengan nama pengguna, misalnya `IAM:temp_creds_user` atau `IAMA:temp_creds_user`.

Important

Konfigurasi klien SQL Anda untuk memerlukan SSL. Jika tidak, jika klien SQL Anda secara otomatis mencoba terhubung dengan SSL, itu dapat kembali ke non-SSL jika ada jenis kegagalan. Dalam hal ini, upaya koneksi pertama mungkin gagal karena kredensialnya kedaluwarsa atau tidak valid, maka upaya koneksi kedua gagal karena koneksi tidak SSL. Jika itu terjadi, pesan kesalahan pertama mungkin terlewatkan. Untuk informasi selengkapnya tentang menghubungkan ke kluster menggunakan SSL, lihat [Mengkonfigurasi opsi keamanan untuk koneksi](#).

6. Jika koneksi tidak menggunakan SSL, upaya koneksi gagal.
7. Cluster mengirimkan authentication permintaan ke klien SQL.
8. Klien SQL kemudian mengirimkan kata sandi sementara ke cluster.
9. Jika kata sandi valid dan belum kedaluwarsa, cluster menyelesaikan koneksi.

Menyiapkan otentikasi masuk tunggal JDBC atau ODBC

Anda dapat memanfaatkan penyedia identitas eksternal (IdPs) untuk mengautentikasi dan mengotorisasi pengguna yang mengakses kluster Amazon Redshift Anda, menyederhanakan manajemen pengguna, dan meningkatkan keamanan. Hal ini memungkinkan manajemen pengguna terpusat, kontrol akses berbasis peran, dan kemampuan audit di berbagai layanan. Kasus penggunaan umum termasuk merampingkan otentikasi untuk beragam grup pengguna, menegakkan kebijakan akses yang konsisten, dan memenuhi persyaratan peraturan.

Halaman-halaman berikut memandu Anda melalui konfigurasi integrasi IDP dengan cluster Redshift Anda. Untuk informasi selengkapnya tentang mengonfigurasi AWS sebagai penyedia layanan untuk IDP, lihat [Mengonfigurasi IDP SAMP 2.0 Anda dengan Kepercayaan Pihak yang Mengandalkan dan Menambahkan](#) Klaim di Panduan Pengguna IAM.

IKLAN FS

Tutorial ini menunjukkan kepada Anda bagaimana Anda dapat menggunakan AD FS sebagai penyedia identitas (iDP) untuk mengakses klaster Amazon Redshift Anda.

Langkah 1: Siapkan AD FS dan AWS akun Anda untuk saling percaya

Prosedur berikut menjelaskan cara mengatur hubungan kepercayaan.

1. Buat atau gunakan klaster Amazon Redshift yang ada agar pengguna AD FS dapat terhubung. Untuk mengkonfigurasi koneksi, properti tertentu dari cluster ini diperlukan, seperti pengidentifikasi cluster. Untuk informasi selengkapnya, lihat [Membuat Cluster](#).
2. Siapkan AD FS untuk mengontrol akses Amazon Redshift di Konsol Manajemen Microsoft:
 1. Pilih ADFS 2.0, lalu pilih Add Relying Party Trust. Pada halaman Add Relying Party Trust Wizard, pilih Mulai.
 2. Pada halaman Pilih Sumber Data, pilih Impor data tentang pihak yang mengandalkan yang dipublikasikan secara online atau di jaringan lokal.
 3. Untuk alamat metadata Federasi (nama host atau URL), masukkan. **https://signin.aws.amazon.com/saml-metadata.xml** File XHTML metadata adalah dokumen metadata SAMP standar yang menggambarkan sebagai pihak yang mengandalkan. AWS
 4. Pada halaman Tentukan Nama Tampilan, masukkan nilai untuk Nama tampilan.
 5. Pada halaman Pilih Aturan Otorisasi Penerbitan, pilih aturan otorisasi penerbitan untuk mengizinkan atau menolak semua pengguna mengakses pihak yang bergantung ini.
 6. Pada halaman Siap Tambah Kepercayaan, tinjau pengaturan Anda.
 7. Pada halaman Selesai, pilih Buka dialog Edit Aturan Klaim untuk kepercayaan pihak yang mengandalkan ini saat wizard ditutup.
 8. Pada menu konteks (klik kanan), pilih Mengandalkan Perwalian Partai.
 9. Untuk pihak yang Anda andalkan, buka menu konteks (klik kanan) dan pilih Edit Aturan Klaim. Pada halaman Edit Aturan Klaim, pilih Tambah Aturan.
 10. Untuk templat aturan Klaim, pilih Ubah Klaim Masuk, lalu pada Nameld halaman Edit Aturan —, lakukan hal berikut:
 - Untuk nama aturan Klaim, masukkan Nameld.
 - Untuk Nama klaim masuk, pilih Nama Akun Windows.
 - Untuk Nama klaim keluar, pilih ID Nama.
 - Untuk format ID nama keluar, pilih Persistent Identifier.

- Pilih Lewati semua nilai klaim.

11 Pada halaman Edit Aturan Klaim, pilih Tambah Aturan. Pada halaman Pilih Templat Aturan, untuk templat aturan Klaim, pilih Kirim Atribut LDAP sebagai Klaim.

12 Pada halaman Configure Rule, lakukan hal berikut:

- Untuk nama aturan Klaim, masukkan RoleSessionName.
- Untuk toko Atribut, pilih Active Directory.
- Untuk Atribut LDAP, pilih Alamat Email.
- Untuk Jenis Klaim Keluar, pilih <https://aws.amazon.com/SAML/Attributes/RoleSessionName>.

13 Pada halaman Edit Aturan Klaim, pilih Tambah Aturan. Pada halaman Pilih Templat Aturan, untuk templat aturan Klaim, pilih Kirim Klaim Menggunakan Aturan Kustom.

14 Pada halaman Aturan Edit — Dapatkan Grup IKLAN, untuk nama aturan Klaim, masukkan Dapatkan Grup IKLAN.

15 Untuk aturan Kustom, masukkan yang berikut ini.

```
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 Pada halaman Edit Aturan Klaim, pilih Tambah Aturan. Pada halaman Pilih Templat Aturan, untuk templat aturan Klaim, pilih Kirim Klaim Menggunakan Aturan Kustom.

17 Pada halaman Edit Aturan — Peran, untuk nama aturan Klaim, ketik Peran.

18 Untuk aturan Kustom, masukkan yang berikut ini.

```
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-"));
```

Perhatikan penyedia SAFL dan peran yang harus diasumsikan. ARNs Dalam contoh ini, `arn:aws:iam:123456789012:saml-provider/ADFS` adalah ARN dari penyedia SAMP dan merupakan ARN `arn:aws:iam:123456789012:role/ADFS-` dari peran tersebut.

3. Pastikan Anda telah mengunduh `federationmetadata.xml` file tersebut. Periksa apakah isi dokumen tidak memiliki karakter yang tidak valid. Ini adalah file metadata yang Anda gunakan saat mengonfigurasi hubungan kepercayaan. AWS
4. Buat penyedia identitas SALL IAM di konsol IAM. Dokumen metadata. yang Anda berikan adalah file XHTML metadata federasi yang Anda simpan saat Anda menyiapkan Aplikasi Azure Enterprise. Untuk langkah-langkah mendetail, lihat [Membuat dan Mengelola Penyedia Identitas IAM \(Konsol\)](#) di Panduan Pengguna IAM.
5. Buat peran IAM untuk federasi SAMP 2.0 di konsol IAM. Untuk langkah-langkah mendetail, lihat [Membuat Peran untuk SAMP](#) di Panduan Pengguna IAM.
6. Buat kebijakan IAM yang dapat Anda lampirkan ke peran IAM yang Anda buat untuk federasi SAMP 2.0 di konsol IAM. Untuk langkah-langkah mendetail, lihat [Membuat Kebijakan IAM \(Konsol\)](#) di Panduan Pengguna IAM. Untuk contoh Azure AD, lihat [Menyiapkan otentikasi masuk tunggal JDBC atau ODBC](#).

Langkah 2: Siapkan JDBC atau ODBC untuk otentikasi ke AD FS

JDBC

Prosedur berikut menjelaskan cara mengatur hubungan JDBC ke AD FS.

- Konfigurasi klien database Anda untuk terhubung ke cluster Anda melalui JDBC menggunakan sistem masuk tunggal AD FS.

Anda dapat menggunakan klien apa pun yang menggunakan driver JDBC untuk terhubung menggunakan sistem masuk tunggal AD FS atau menggunakan bahasa seperti Java untuk terhubung menggunakan skrip. Untuk informasi instalasi dan konfigurasi, lihat [Mengonfigurasi koneksi untuk driver JDBC versi 2.x untuk Amazon Redshift](#).

Misalnya, Anda dapat menggunakan SQLWorkbench/J sebagai klien. Ketika Anda mengkonfigurasi SQLWorkbench /J, URL database Anda menggunakan format berikut.

```
jdbc:redshift:iam://cluster-identifier:us-west-1/dev
```

Jika Anda menggunakan SQLWorkbench/J sebagai klien, ambil langkah-langkah berikut:

- a. Mulai SQL Workbench/J. Di halaman Select Connection Profile, tambahkan Profil Grup, misalnya. **ADFS**

- b. Untuk Profil Koneksi, masukkan nama profil koneksi Anda, misalnya **ADFS**.
- c. Pilih Kelola Driver, dan pilih Amazon Redshift. Pilih ikon Open Folder di sebelah Library, lalu pilih file JDBC .jar yang sesuai.
- d. Pada halaman Pilih Profil Koneksi, tambahkan informasi ke profil koneksi sebagai berikut:
 - Untuk Pengguna, masukkan nama pengguna AD FS Anda. Ini adalah nama pengguna akun yang Anda gunakan untuk sistem masuk tunggal yang memiliki izin ke kluster yang Anda coba autentikasi.
 - Untuk Kata Sandi, masukkan kata sandi AD FS Anda.
 - Untuk Driver, pilih Amazon Redshift (com.amazon.redshift.jdbc.driver).
 - Untuk URL, masukkan **`jdbc:redshift:iam://your-cluster-identifier:your-cluster-region/your-database-name`**.
- e. Pilih Extended Properties. Untuk plugin_name, masukkan **`com.amazon.redshift.plugin.AdfsCredentialsProvider`**. Nilai ini menentukan driver untuk menggunakan sistem masuk tunggal AD FS sebagai metode otentikasi.

ODBC

Untuk mengatur ODBC untuk otentikasi ke AD FS

- Konfigurasi klien database Anda untuk terhubung ke kluster Anda melalui ODBC menggunakan sistem masuk tunggal AD FS.

Amazon Redshift menyediakan driver ODBC untuk sistem operasi Linux, Windows, dan macOS. Sebelum Anda instal driver ODBC, tentukan apakah alat klien SQL Anda 32-bit atau 64-bit. Instal driver ODBC yang sesuai dengan persyaratan alat klien SQL Anda.

Di Windows, di halaman Pengaturan DSN Driver Amazon Redshift ODBC, di bawah Pengaturan Koneksi, masukkan informasi berikut:

- Untuk Nama Sumber Data, masukkan **`your-DSN`**. Ini menentukan nama sumber data yang digunakan sebagai nama profil ODBC.
- Untuk jenis Auth, pilih Penyedia Identitas: SAMP. Ini adalah metode otentikasi yang digunakan driver ODBC untuk mengautentikasi menggunakan sistem masuk tunggal AD FS.

- Untuk ID Cluster, masukkan ***your-cluster-identifier***.
- Untuk Wilayah, masukkan ***your-cluster-region***.
- Untuk Database, masukkan ***your-database-name***.
- Untuk Pengguna, masukkan ***your-adfs-username***. Ini adalah nama pengguna untuk akun AD FS yang Anda gunakan untuk sistem masuk tunggal yang memiliki izin ke klaster yang Anda coba autentikasi. Gunakan ini hanya untuk jenis Auth adalah Penyedia Identitas: SAMP.
- Untuk Kata Sandi, masukkan ***your-adfs-password***. Gunakan ini hanya untuk jenis Auth adalah Penyedia Identitas: SAMP.

Di macOS dan Linux, edit `odbc.ini` file sebagai berikut:



Note

Semua entri tidak peka huruf besar/kecil.

- Untuk clusterid, masukkan. ***your-cluster-identifier*** Ini adalah nama cluster Amazon Redshift yang dibuat.
- Untuk wilayah, masukkan ***your-cluster-region***. Ini adalah AWS Wilayah cluster Amazon Redshift yang dibuat.
- Untuk database, masukkan ***your-database-name***. Ini adalah nama database yang Anda coba akses di cluster Amazon Redshift.
- Untuk lokal, masukkan ***en-us***. Ini adalah bahasa yang menampilkan pesan kesalahan.
- Untuk iam, masukkan ***1***. Nilai ini menentukan driver untuk mengautentikasi menggunakan kredensial IAM.
- Untuk plugin_name, lakukan salah satu hal berikut:
 - Untuk proses masuk tunggal AD FS dengan konfigurasi MFA, masukkan. ***BrowserSAML*** Ini adalah metode otentikasi yang digunakan driver ODBC untuk mengautentikasi ke AD FS.
 - Untuk konfigurasi masuk tunggal AD FS, masukkan. ***ADFS*** Ini adalah metode otentikasi yang digunakan driver ODBC untuk mengautentikasi menggunakan sistem masuk tunggal Azure AD.

- Untuk uid, masukkan ***your-adfs-username***. Ini adalah nama pengguna akun Microsoft Azure yang Anda gunakan untuk sistem masuk tunggal yang memiliki izin ke cluster yang Anda coba autentikasi. Gunakan ini hanya untuk plugin_name adalah ADFS.
- Untuk pwd, masukkan. ***your-adfs-password*** Gunakan ini hanya untuk plugin_name adalah ADFS.

Di macOS dan Linux, edit juga pengaturan profil untuk menambahkan ekspor berikut.

```
export ODBCINI=/opt/amazon/redshift/Setup/odbc.ini
```

```
export ODBCINSTINI=/opt/amazon/redshift/Setup/odbcinst.ini
```

Biru langit

Anda dapat menggunakan Microsoft Azure AD sebagai penyedia identitas (iDP) untuk mengakses kluster Amazon Redshift Anda. Tutorial ini menunjukkan kepada Anda bagaimana Anda dapat menggunakan Azure sebagai penyedia identitas (iDP) untuk mengakses cluster Amazon Redshift Anda.

Tonton video berikut untuk mempelajari cara menggabungkan akses Amazon Redshift dengan sistem masuk tunggal Microsoft Azure AD: [Menggabungkan akses Amazon Redshift dengan sistem masuk tunggal Microsoft Azure AD](#).

Langkah 1: Siapkan Azure dan AWS akun Anda untuk saling percaya

Prosedur berikut menjelaskan cara mengatur hubungan kepercayaan.

Untuk mengatur Azure AD dan AWS akun Anda agar saling percaya

1. Buat atau gunakan kluster Amazon Redshift yang ada agar pengguna Azure AD dapat terhubung. Untuk mengkonfigurasi koneksi, properti tertentu dari cluster ini diperlukan, seperti pengidentifikasi cluster. Untuk informasi selengkapnya, lihat [Membuat Cluster](#).
2. Siapkan Direktori Aktif Azure, grup, pengguna yang digunakan AWS di portal Microsoft Azure.
3. Tambahkan Amazon Redshift sebagai aplikasi perusahaan di portal Microsoft Azure untuk digunakan untuk masuk tunggal ke AWS Konsol dan login gabungan ke Amazon Redshift. Pilih aplikasi Enterprise.
4. Pilih +Aplikasi baru. Halaman Tambahkan aplikasi muncul.

5. Cari **AWS** di bidang pencarian.
6. Pilih Amazon Web Services (AWS) dan pilih Tambah. Ini menciptakan AWS aplikasi.
7. Di bawah Kelola, pilih Single sign-on.
8. Pilih SAML. Halaman Amazon Web Services (AWS) | Masuk berbasis SAML muncul.
9. Pilih Ya untuk melanjutkan ke halaman Mengatur Single Sign-On dengan SAFL. Halaman ini menampilkan daftar atribut terkait sistem masuk tunggal yang telah dikonfigurasi sebelumnya.
10. Untuk Konfigurasi SAMP Dasar, pilih ikon edit dan pilih Simpan.
11. Saat Anda mengonfigurasi lebih dari satu aplikasi, berikan nilai pengenalan. Misalnya, masukkan **<https://signin.aws.amazon.com/saml#2>**. Perhatikan bahwa dari aplikasi kedua dan seterusnya, gunakan format ini dengan tanda # untuk menentukan nilai SPN yang unik.
12. Di bagian Atribut Pengguna dan Klaim, pilih ikon edit.

Secara default, Unique User Identifier (UID), Peran RoleSessionName, dan SessionDuration klaim sudah dikonfigurasi sebelumnya.

13. Pilih + Tambahkan klaim baru untuk menambahkan klaim bagi pengguna database.

Untuk Nama, masukkan **DbUser**.

Untuk Namespace, masukkan **<https://redshift.amazon.com/SAML/Attributes>**.

Untuk Sumber, pilih Atribut.

Untuk atribut Source, pilih user.userprincipalname. Lalu, pilih Simpan.

14. Pilih + Tambahkan klaim baru untuk menambahkan klaim AutoCreate.

Untuk Nama, masukkan **AutoCreate**.

Untuk Namespace, masukkan **<https://redshift.amazon.com/SAML/Attributes>**.

Untuk Sumber, pilih Atribut.

Untuk atribut Source, pilih "true". Lalu, pilih Simpan.

Di sini, **123456789012** adalah AWS akun Anda, **AzureSSO** adalah peran IAM yang Anda buat, dan **AzureADProvider** merupakan penyedia IAM.

Nama klaim	Nilai
Pengenal pengguna unik (ID nama)	user.userprincipalname
https://aws.amazon.com/SAML/Attributes/SessionDuration	"900"
https://aws.amazon.com/SAML/Attributes/Role	arn:aws:iam: ::role/, arn:aws:iam: ::saml-provider/ <i>123456789012 AzureSSO123456789012 AzureADProvider</i>
https://aws.amazon.com/SAML/Attributes/RoleSessionName	user.userprincipalname
https://redshift.amazon.com/SAML/Attributes/AutoCreate	"benar"
https://redshift.amazon.com/SAML/Attributes/DbGroups	user.assignedroles
https://redshift.amazon.com/SAML/Attributes/DbUser	user.userprincipalname

15. Di bawah Pendaftaran Aplikasi > ***your-application-name*** > Otentikasi, tambahkan Aplikasi Seluler Dan Desktop. Tentukan URL sebagai http://localhost/redshift/.
16. Di bagian Sertifikat Penandatanganan SAMP, pilih Unduh untuk mengunduh dan menyimpan file XHTML metadata federasi untuk digunakan saat Anda membuat penyedia identitas SAMP IAM. File ini digunakan untuk membuat identitas federasi masuk tunggal.
17. Buat penyedia identitas SALL IAM di konsol IAM. Dokumen metadata yang Anda berikan adalah file XMLmetadata federasi yang Anda simpan saat Anda menyiapkan Aplikasi Azure Enterprise. Untuk langkah-langkah mendetail, lihat [Membuat dan Mengelola Penyedia Identitas IAM \(Konsol\)](#) di Panduan Pengguna IAM.
18. Buat peran IAM untuk federasi SAMP 2.0 di konsol IAM. Untuk langkah-langkah mendetail, lihat [Membuat Peran untuk SAMP](#) di Panduan Pengguna IAM.
19. Buat kebijakan IAM yang dapat Anda lampirkan ke peran IAM yang Anda buat untuk federasi SAMP 2.0 di konsol IAM. Untuk langkah-langkah mendetail, lihat [Membuat Kebijakan IAM \(Konsol\)](#) di Panduan Pengguna IAM.

Ubah kebijakan berikut (dalam format JSON) untuk lingkungan Anda:

- Gantikan AWS Region cluster Anda untuk *us-west-1*.
- Ganti AWS akun Anda untuk *123456789012*.
- Gantikan pengenalan klaster Anda (atau * untuk semua cluster) untuk *cluster-identifier*
- Gantikan database Anda (atau * untuk semua database) untuk *dev*.
- Gantikan pengenalan unik peran IAM Anda. *AROAJ2UCCR6DPCEXAMPLE*
- Gantikan domain email penyewa atau perusahaan Anda. *example.com*
- Gantikan grup database yang Anda rencanakan untuk menetapkan pengguna. *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": "*"
    }
]
}

```

Kebijakan ini memberikan izin sebagai berikut:

- Bagian pertama memberikan izin ke operasi `GetClusterCredentials` API untuk mendapatkan kredensial sementara untuk klaster yang ditentukan. Dalam contoh ini, sumber daya adalah `cluster-identifier` dengan `database:dev`, dalam akun `123456789012`, dan di AWS Wilayah `us-west-1`. `${redshift:DbUser}` Klausula ini hanya memungkinkan pengguna yang cocok dengan `DbUser` nilai yang ditentukan dalam Azure AD untuk terhubung.
- Klausul kondisi memberlakukan bahwa hanya pengguna tertentu yang mendapatkan kredensial sementara. Ini adalah pengguna di bawah peran yang ditentukan oleh ID unik peran `AROAJ2UCCR6DPCEXAMPLE` di akun IAM yang diidentifikasi oleh alamat email di domain email perusahaan Anda. Untuk informasi selengkapnya tentang unik IDs, lihat [Unik IDs](#) di Panduan Pengguna IAM.

Penyiapan Anda dengan iDP Anda (dalam hal ini, Azure AD) menentukan bagaimana klausa kondisi ditulis. Jika email karyawan Anda `johndoe@example.com`, atur terlebih dahulu `${redshift:DbUser}` ke bidang super yang cocok dengan nama pengguna karyawan `johndoe`. Kemudian, untuk membuat kondisi ini berfungsi, atur `RoleSessionName` bidang AWS SAMP ke bidang super yang cocok dengan email `johndoe@example.com` karyawan. Saat Anda mengambil pendekatan ini, pertimbangkan hal berikut:

- Jika Anda mengatur `${redshift:DbUser}` untuk menjadi email karyawan, maka hapus `@example.com` dalam contoh JSON untuk mencocokkan `RoleSessionName`.
- Jika Anda menetapkan `RoleSessionId` untuk menjadi hanya nama pengguna karyawan, maka hapus `@example.com` dalam contoh untuk mencocokkan `RoleSessionName`.

- Dalam contoh JSON, `${redshift:DbUser}` dan `RoleSessionName` keduanya diatur ke email karyawan. Contoh ini JSON menggunakan nama `@example.com` pengguna database Amazon Redshift untuk menandatangani pengguna untuk mengakses cluster.
- Bagian kedua memberikan izin untuk membuat `dbuser` nama di cluster yang ditentukan. Dalam contoh ini JSON, ini membatasi pembuatan untuk `${redshift:DbUser}`
- Bagian ketiga memberikan izin untuk menentukan pengguna mana `dbgroup` yang dapat bergabung. Dalam contoh ini JSON, pengguna dapat bergabung dengan `my_dbgroup` grup dalam cluster yang ditentukan.
- Bagian keempat memberikan izin untuk tindakan yang dapat dilakukan pengguna pada semua sumber daya. Dalam contoh ini JSON, memungkinkan pengguna untuk memanggil `redshift:DescribeClusters` untuk mendapatkan informasi cluster seperti titik akhir cluster, AWS Region, dan port. Ini juga memungkinkan pengguna untuk menelepon `iam:ListRoles` untuk memeriksa peran mana yang dapat diasumsikan pengguna.

Langkah 2: Siapkan JDBC atau ODBC untuk otentikasi ke Azure

JDBC

Untuk mengatur JDBC untuk otentikasi ke Microsoft Azure AD

- Konfigurasi klien database Anda untuk terhubung ke cluster Anda melalui JDBC menggunakan sistem masuk tunggal Azure AD Anda.

Anda dapat menggunakan klien apa pun yang menggunakan driver JDBC untuk terhubung menggunakan sistem masuk tunggal Azure AD atau menggunakan bahasa seperti Java untuk terhubung menggunakan skrip. Untuk informasi instalasi dan konfigurasi, lihat [Mengonfigurasi koneksi untuk driver JDBC versi 2.x untuk Amazon Redshift](#).

Misalnya, Anda dapat menggunakan SQLWorkbench/J sebagai klien. Ketika Anda mengkonfigurasi SQLWorkbench /J, URL database Anda menggunakan format berikut.

```
jdbc:redshift:iam://cluster-identifier:us-west-1/dev
```

Jika Anda menggunakan SQLWorkbench/J sebagai klien, ambil langkah-langkah berikut:

- a. Mulai SQL Workbench/J. Pada halaman Select Connection Profile, tambahkan Profil Grup yang disebut. **AzureAuth**

- b. Untuk Profil Koneksi, masukkan **Azure**.
- c. Pilih Kelola Driver, dan pilih Amazon Redshift. Pilih ikon Open Folder di sebelah Library, lalu pilih file JDBC .jar yang sesuai.
- d. Pada halaman Pilih Profil Koneksi, tambahkan informasi ke profil koneksi sebagai berikut:
 - Untuk Pengguna, masukkan nama pengguna Microsoft Azure Anda. Ini adalah nama pengguna akun Microsoft Azure yang Anda gunakan untuk sistem masuk tunggal yang memiliki izin ke cluster yang Anda coba autentikasi.
 - Untuk Kata Sandi, masukkan kata sandi Microsoft Azure Anda.
 - Untuk Driver, pilih Amazon Redshift (com.amazon.redshift.jdbc.driver).
 - Untuk URL, masukkan **jdbc:redshift:iam://your-cluster-identifier:your-cluster-region/your-database-name**.
- e. Pilih Extended Properties untuk menambahkan informasi tambahan ke properti koneksi, seperti yang dijelaskan berikut.

Untuk konfigurasi masuk tunggal Azure AD, tambahkan informasi tambahan sebagai berikut:

- Untuk plugin_name, masukkan.
com.amazon.redshift.plugin.AzureCredentialsProvider Nilai ini menentukan driver untuk menggunakan Azure AD Single Sign-On sebagai metode otentikasi.
- Untuk idp_tenant, masukkan. **your-idp-tenant** Hanya digunakan untuk Microsoft Azure AD. Ini adalah nama penyewa perusahaan Anda yang dikonfigurasi pada Azure AD. Nilai ini dapat berupa nama penyewa atau ID unik penyewa dengan tanda hubung.
- Untuk client_secret, masukkan. **your-azure-redshift-application-client-secret** Hanya digunakan untuk Microsoft Azure AD. Ini adalah rahasia klien Anda dari aplikasi Amazon Redshift yang Anda buat saat mengatur konfigurasi Azure Single Sign-On Anda. Ini hanya berlaku untuk com.amazon.redshift.plugin.AzureCredentialsProviderplugin.
- Untuk client_id, masukkan. **your-azure-redshift-application-client-id** Hanya digunakan untuk Microsoft Azure AD. Ini adalah ID klien (dengan tanda

hubung) dari aplikasi Amazon Redshift yang Anda buat saat mengatur konfigurasi Azure Single Sign-On Anda.

Untuk sistem masuk tunggal Azure AD dengan konfigurasi MFA, tambahkan informasi tambahan ke properti koneksi sebagai berikut:

- Untuk `plugin_name`, masukkan. **`com.amazon.redshift.plugin.BrowserAzureCredentialsProvider`** Nilai ini menentukan driver untuk menggunakan sistem masuk tunggal Azure AD dengan MFA sebagai metode otentikasi.
- Untuk `idp_tenant`, masukkan. ***your-idp-tenant*** Hanya digunakan untuk Microsoft Azure AD. Ini adalah nama penyewa perusahaan Anda yang dikonfigurasi pada Azure AD. Nilai ini dapat berupa nama penyewa atau ID unik penyewa dengan tanda hubung.
- Untuk `client_id`, masukkan. ***your-azure-redshift-application-client-id*** Opsi ini hanya digunakan untuk Microsoft Azure AD. Ini adalah ID klien (dengan tanda hubung) dari aplikasi Amazon Redshift yang Anda buat saat mengatur sistem masuk tunggal Azure AD Anda dengan konfigurasi MFA.
- Untuk `listen_port`, masukkan. ***your-listen-port*** Ini adalah port yang didengarkan oleh server lokal. Defaultnya adalah 7890.
- Untuk `idp_response_timeout`, masukkan. ***the-number-of-seconds*** Ini adalah jumlah detik untuk menunggu sebelum waktu habis ketika server iDP mengirim kembali respons. Jumlah minimum detik harus 10. Jika membuat koneksi membutuhkan waktu lebih lama dari ambang ini, maka koneksi dibatalkan.

ODBC

Untuk mengatur ODBC untuk otentikasi ke Microsoft Azure AD

- Konfigurasi klien database Anda untuk terhubung ke klaster Anda melalui ODBC menggunakan sistem masuk tunggal Azure AD Anda.

Amazon Redshift menyediakan driver ODBC untuk sistem operasi Linux, Windows, dan macOS. Sebelum Anda instal driver ODBC, tentukan apakah alat klien SQL Anda 32-bit atau 64-bit. Instal driver ODBC yang sesuai dengan persyaratan alat klien SQL Anda.

Di Windows, di halaman Pengaturan DSN Driver Amazon Redshift ODBC, di bawah Pengaturan Koneksi, masukkan informasi berikut:

- Untuk Nama Sumber Data, masukkan ***your-DSN***. Ini menentukan nama sumber data yang digunakan sebagai nama profil ODBC.
- Untuk tipe Auth untuk konfigurasi masuk tunggal Azure AD, pilih. **Identity Provider: Azure AD** Ini adalah metode otentikasi yang digunakan driver ODBC untuk mengautentikasi menggunakan sistem masuk tunggal Azure.
- Untuk tipe Auth untuk sistem masuk tunggal Azure AD dengan konfigurasi MFA, pilih. **Identity Provider: Browser Azure AD** Ini adalah metode otentikasi yang digunakan driver ODBC untuk mengautentikasi menggunakan Azure single sign-on dengan MFA.
- Untuk ID Cluster, masukkan ***your-cluster-identifier***.
- Untuk Wilayah, masukkan ***your-cluster-region***.
- Untuk Database, masukkan ***your-database-name***.
- Untuk Pengguna, masukkan ***your-azure-username***. Ini adalah nama pengguna untuk akun Microsoft Azure yang Anda gunakan untuk sistem masuk tunggal yang memiliki izin ke cluster yang Anda coba autentikasi. Gunakan ini hanya untuk Jenis Auth adalah Penyedia Identitas: Azure AD.
- Untuk Kata Sandi, masukkan ***your-azure-password***. Gunakan ini hanya untuk Jenis Auth adalah Penyedia Identitas: Azure AD.
- Untuk IDP Tenant, masukkan. ***your-idp-tenant*** Ini adalah nama penyewa perusahaan Anda yang dikonfigurasi di IDP (Azure) Anda. Nilai ini dapat berupa nama penyewa atau ID unik penyewa dengan tanda hubung.
- Untuk Rahasia Klien Azure, masukkan ***your-azure-redshift-application-client-secret***. Ini adalah rahasia klien dari aplikasi Amazon Redshift yang Anda buat saat mengatur konfigurasi masuk tunggal Azure Anda.
- Untuk ID Klien Azure, masukkan ***your-azure-redshift-application-client-id***. Ini adalah ID klien (dengan tanda hubung) dari aplikasi Amazon Redshift yang Anda buat saat mengatur konfigurasi masuk tunggal Azure Anda.
- Untuk Listen Port, masukkan ***your-listen-port***. Ini adalah port mendengarkan default yang didengarkan oleh server lokal. Defaultnya adalah 7890. Ini hanya berlaku untuk plugin Browser Azure AD.

- Untuk Response Timeout, masukkan ***the-number-of-seconds***. Ini adalah jumlah detik untuk menunggu sebelum waktu habis ketika server iDP mengirim kembali respons. Jumlah minimum detik harus 10. Jika membuat koneksi membutuhkan waktu lebih lama dari ambang ini, maka koneksi dibatalkan. Opsi ini hanya berlaku untuk plugin Browser Azure AD.

Di macOS dan Linux, edit `odbc.ini` file sebagai berikut:

 Note

Semua entri tidak peka huruf besar/kecil.

- Untuk clusterid, masukkan ***your-cluster-identifier***. Ini adalah nama cluster Amazon Redshift yang dibuat.
- Untuk wilayah, masukkan ***your-cluster-region***. Ini adalah AWS Wilayah cluster Amazon Redshift yang dibuat.
- Untuk database, masukkan ***your-database-name***. Ini adalah nama database yang Anda coba akses di cluster Amazon Redshift.
- Untuk lokal, masukkan ***en-us***. Ini adalah bahasa yang menampilkan pesan kesalahan.
- Untuk iam, masukkan ***1***. Nilai ini menentukan driver untuk mengautentikasi menggunakan kredensial IAM.
- Untuk plugin_name untuk konfigurasi masuk tunggal Azure AD, masukkan ***AzureAD***. Ini menentukan driver untuk menggunakan Azure Single Sign-On sebagai metode otentikasi.
- Untuk plugin_name untuk sistem masuk tunggal Azure AD dengan konfigurasi MFA, masukkan ***BrowserAzureAD***. Ini menentukan driver untuk menggunakan Azure Single Sign-On dengan MFA sebagai metode otentikasi.
- Untuk uid, masukkan ***your-azure-username***. Ini adalah nama pengguna akun Microsoft Azure yang Anda gunakan untuk sistem masuk tunggal yang memiliki izin ke cluster yang Anda coba autentikasi. Gunakan ini hanya untuk plugin_name adalah AzuRead.
- Untuk pwd, masukkan ***your-azure-password***. Gunakan ini hanya untuk plugin_name adalah AzuRead.

- Untuk `idp_tenant`, masukkan. ***your-idp-tenant*** Ini adalah nama penyewa perusahaan Anda yang dikonfigurasi di IDP (Azure) Anda. Nilai ini dapat berupa nama penyewa atau ID unik penyewa dengan tanda hubung.
- Untuk `client_secret`, masukkan. ***your-azure-redshift-application-client-secret*** Ini adalah rahasia klien dari aplikasi Amazon Redshift yang Anda buat saat mengatur konfigurasi masuk tunggal Azure Anda.
- Untuk `client_id`, masukkan. ***your-azure-redshift-application-client-id*** Ini adalah ID klien (dengan tanda hubung) dari aplikasi Amazon Redshift yang Anda buat saat mengatur konfigurasi masuk tunggal Azure Anda.
- Untuk `listen_port`, masukkan. ***your-listen-port*** Ini adalah port yang didengarkan oleh server lokal. Defaultnya adalah 7890. Ini berlaku untuk plugin Browser Azure AD.
- Untuk `idp_response_timeout`, masukkan. ***the-number-of-seconds*** Ini adalah periode waktu yang ditentukan dalam hitungan detik untuk menunggu respons dari Azure. Opsi ini berlaku untuk plugin Browser Azure AD.

Di macOS dan Linux, edit juga pengaturan profil untuk menambahkan ekspor berikut.

```
export ODBCINI=/opt/amazon/redshift/Setup/odbc.ini
```

```
export ODBCINSTINI=/opt/amazon/redshift/Setup/odbcinst.ini
```

Pemecahan masalah

Untuk memecahkan masalah dengan plugin Browser Azure AD, pertimbangkan hal berikut.

- Untuk menggunakan plugin Browser Azure AD, Anda harus menyetel URL balasan yang ditentukan dalam permintaan agar sesuai dengan URL balasan yang dikonfigurasi untuk aplikasi Anda. Arahkan ke halaman Siapkan Single Sign-On dengan SAMP di portal Microsoft Azure. Kemudian periksa URL Balas diatur ke `http://localhost/redshift/`.
- Jika Anda mendapatkan kesalahan penyewa IDP, verifikasi bahwa nama IDP Tenant cocok dengan nama domain yang awalnya Anda gunakan untuk mengatur Direktori Aktif di Microsoft Azure.

Di Windows, navigasikan ke bagian Pengaturan Koneksi di halaman Pengaturan Amazon Redshift ODBC DSN. Kemudian periksa nama penyewa perusahaan Anda yang dikonfigurasi di IDP

(Azure) cocok dengan nama domain yang awalnya Anda gunakan untuk mengatur Direktori Aktif di Microsoft Azure.

Di macOS dan Linux, temukan file `odbc.ini`. Kemudian periksa nama penyewa perusahaan Anda yang dikonfigurasi di IDP (Azure) cocok dengan nama domain yang awalnya Anda gunakan untuk mengatur Direktori Aktif di Microsoft Azure.

- Jika Anda mendapatkan kesalahan bahwa URL balasan yang ditentukan dalam permintaan tidak cocok dengan balasan yang URLs dikonfigurasi untuk aplikasi Anda, verifikasi bahwa Pengalihan URLs sama dengan URL balasan.

Arahkan ke halaman pendaftaran Aplikasi aplikasi Anda di portal Microsoft Azure. Kemudian periksa Redirect URLs cocok dengan URL balasan.

- Jika Anda mendapatkan respons tak terduga: kesalahan tidak sah, verifikasi bahwa Anda telah menyelesaikan konfigurasi aplikasi Seluler dan desktop.

Arahkan ke halaman pendaftaran Aplikasi aplikasi Anda di portal Microsoft Azure. Kemudian navigasikan ke Otentikasi dan periksa apakah Anda mengonfigurasi aplikasi Seluler dan desktop untuk menggunakan `http://localhost/redshift/` sebagai URLs pengalihan.

Identitas Ping

Anda dapat menggunakan Ping Identity sebagai penyedia identitas (iDP) untuk mengakses kluster Amazon Redshift Anda. Tutorial ini menunjukkan kepada Anda bagaimana Anda dapat menggunakan Ping Identity sebagai penyedia identitas (iDP) untuk mengakses cluster Amazon Redshift Anda.

Langkah 1: Siapkan Identitas Ping dan AWS akun Anda untuk saling percaya

Prosedur berikut menjelaskan cara mengatur hubungan kepercayaan menggunakan PingOne portal.

Untuk mengatur Identitas Ping dan AWS akun Anda agar saling percaya

1. Buat atau gunakan kluster Amazon Redshift yang ada agar pengguna Identitas Ping Anda dapat terhubung. Untuk mengkonfigurasi koneksi, properti tertentu dari cluster ini diperlukan, seperti pengidentifikasi cluster. Untuk informasi selengkapnya, lihat [Membuat Cluster](#).
2. Tambahkan Amazon Redshift sebagai aplikasi SAMP baru di portal. PingOne Untuk langkah-langkah rinci, lihat [dokumentasi Ping Identity](#).

1. Buka Aplikasi Saya.

2. Di bawah Add Application, pilih New SALL Application.
3. Untuk Nama Aplikasi, masukkan **Amazon Redshift**.
4. Untuk Protocol Version, pilih SAMP v2.0.
5. Untuk Kategori, pilih ***your-application-category***.
6. Untuk Assertion Consumer Service (ACS), ketik. ***your-redshift-local-host-url*** Ini adalah host dan port lokal yang dialihkan oleh pernyataan SAMP.
7. Untuk ID Entitas, masukkan `urn:amazon:webservices`.
8. Untuk Menandatangani, pilih Tanda Pernyataan.
9. Di bagian Pemetaan Atribut SSO, buat klaim seperti yang ditunjukkan pada tabel berikut.

Atribut aplikasi	Atribut jembatan identitas dari nilai literal
<code>https://aws.amazon.com/SAML/Attributes/Role</code>	<code>arn:aws:iam: ::role/, arn:aws:iam: ::saml-provider/ <i>123456789012 Ping 123456789012 PingProvider</i></code>
<code>https://aws.amazon.com/SAML/Attributes/RoleSessionName</code>	Email
<code>https://redshift.amazon.com/SAML/Attributes/AutoCreate</code>	"benar"
<code>https://redshift.amazon.com/SAML/Attribute/DbUser</code>	Email
<code>https://redshift.amazon.com/SAML/Attribute/DbGroups</code>	Grup dalam atribut "DbGroups" berisi awalan @directory. Untuk menghapus ini, di jembatan Identity, masukkan memberOf. Di Function, pilih ExtractByRegularExpression. Dalam Ekspresi, masukkan <code>(.*) [\ @] (? :.*).</code>

3. Untuk Akses Grup, siapkan akses grup berikut, jika diperlukan:
 - `https://aws.amazon.com/SAML/Attributes/Role`
 - `https://aws.amazon.com/SAML/Attributes/RoleSessionName`
 - `https://redshift.amazon.com/SAML/Attributes/AutoCreate`

- <https://redshift.amazonaws.com/SAML/Attributes/DbUser>
- 4. Tinjau pengaturan Anda dan buat perubahan, jika perlu.
- 5. Gunakan URL Initiate Single Sign-On (SSO) sebagai URL login untuk plugin Browser SAMP.
- 6. Buat penyedia identitas SALL IAM di konsol IAM. Dokumen metadata yang Anda berikan adalah file XMLmetadata federasi yang Anda simpan saat Anda mengatur Ping Identity. Untuk langkah-langkah mendetail, lihat [Membuat dan Mengelola Penyedia Identitas IAM \(Konsol\)](#) di Panduan Pengguna IAM.
- 7. Buat peran IAM untuk federasi SAMP 2.0 di konsol IAM. Untuk langkah-langkah mendetail, lihat [Membuat Peran untuk SAMP](#) di Panduan Pengguna IAM.
- 8. Buat kebijakan IAM yang dapat Anda lampirkan ke peran IAM yang Anda buat untuk federasi SAMP 2.0 di konsol IAM. Untuk langkah-langkah mendetail, lihat [Membuat Kebijakan IAM \(Konsol\)](#) di Panduan Pengguna IAM. Untuk contoh Azure AD, lihat [Menyiapkan otentikasi masuk tunggal JDBC atau ODBC](#).

Langkah 2: Siapkan JDBC atau ODBC untuk otentikasi ke Ping Identity

JDBC

Untuk mengatur JDBC untuk otentikasi ke Ping Identity

- Konfigurasi klien database Anda untuk terhubung ke cluster Anda melalui JDBC menggunakan sistem masuk tunggal Ping Identity.

Anda dapat menggunakan klien apa pun yang menggunakan driver JDBC untuk terhubung menggunakan sistem masuk tunggal Ping Identity atau menggunakan bahasa seperti Java untuk terhubung menggunakan skrip. Untuk informasi instalasi dan konfigurasi, lihat [Mengonfigurasi koneksi untuk driver JDBC versi 2.x untuk Amazon Redshift](#).

Misalnya, Anda dapat menggunakan SQLWorkbench/J sebagai klien. Ketika Anda mengkonfigurasi SQLWorkbench /J, URL database Anda menggunakan format berikut.

```
jdbc:redshift:iam://cluster-identifier:us-west-1/dev
```

Jika Anda menggunakan SQLWorkbench/J sebagai klien, ambil langkah-langkah berikut:

- a. Mulai SQL Workbench/J. Di halaman Select Connection Profile, tambahkan Profil Grup, misalnya. **Ping**

- b. Untuk Profil Koneksi ***your-connection-profile-name***, masukkan, misalnya **Ping**.
- c. Pilih Kelola Driver, dan pilih Amazon Redshift. Pilih ikon Open Folder di sebelah Library, lalu pilih file JDBC .jar yang sesuai.
- d. Pada halaman Pilih Profil Koneksi, tambahkan informasi ke profil koneksi sebagai berikut:
 - Untuk Pengguna, masukkan nama PingOne pengguna Anda. Ini adalah nama pengguna PingOne akun yang Anda gunakan untuk sistem masuk tunggal yang memiliki izin ke klaster yang Anda coba autentikasi.
 - Untuk Kata Sandi, masukkan PingOne kata sandi Anda.
 - Untuk Driver, pilih Amazon Redshift (com.amazon.redshift.jdbc.driver).
 - Untuk URL, masukkan ***jdbc:redshift:iam://your-cluster-identifier:your-cluster-region/your-database-name***.
- e. Pilih Extended Properties dan lakukan salah satu hal berikut:
 - Untuk login_url, masukkan. ***your-ping-sso-login-url*** Nilai ini menentukan URL untuk menggunakan sistem masuk tunggal sebagai otentikasi untuk masuk.
 - Untuk Ping Identity, untuk plugin_name, masukkan. ***com.amazon.redshift.plugin.PingCredentialsProvider*** Nilai ini menentukan driver untuk menggunakan Ping Identity single sign-on sebagai metode otentikasi.
 - Untuk Ping Identity dengan sistem masuk tunggal, untuk plugin_name, masukkan. ***com.amazon.redshift.plugin.BrowserSamlCredentialsProvider*** Nilai ini menentukan driver untuk menggunakan Ping Identity PingOne dengan single sign-on sebagai metode otentikasi.

ODBC

Untuk mengatur ODBC untuk otentikasi ke Ping Identity

- Konfigurasi klien database Anda untuk terhubung ke cluster Anda melalui ODBC menggunakan sistem masuk PingOne tunggal Ping Identity.

Amazon Redshift menyediakan driver ODBC untuk sistem operasi Linux, Windows, dan macOS. Sebelum Anda instal driver ODBC, tentukan apakah alat klien SQL Anda 32-bit atau 64-bit. Instal driver ODBC yang sesuai dengan persyaratan alat klien SQL Anda.

Di Windows, di halaman Pengaturan DSN Driver Amazon Redshift ODBC, di bawah Pengaturan Koneksi, masukkan informasi berikut:

- Untuk Nama Sumber Data, masukkan ***your-DSN***. Ini menentukan nama sumber data yang digunakan sebagai nama profil ODBC.
- Untuk jenis Auth, lakukan salah satu hal berikut:
 - Untuk konfigurasi Ping Identity, pilih Identity Provider: Ping Federate. Ini adalah metode otentikasi yang digunakan driver ODBC untuk mengautentikasi menggunakan sistem masuk tunggal Ping Identity.
 - Untuk Ping Identity dengan konfigurasi masuk tunggal, pilih Penyedia Identitas: Browser SAMP. Ini adalah metode otentikasi yang digunakan driver ODBC untuk mengautentikasi menggunakan Ping Identity dengan sistem masuk tunggal.
- Untuk ID Cluster, masukkan ***your-cluster-identifier***.
- Untuk Wilayah, masukkan ***your-cluster-region***.
- Untuk Database, masukkan ***your-database-name***.
- Untuk Pengguna, masukkan ***your-ping-username***. Ini adalah nama pengguna untuk PingOne akun yang Anda gunakan untuk sistem masuk tunggal yang memiliki izin ke klaster yang Anda coba autentikasi. Gunakan ini hanya untuk jenis Auth adalah Penyedia Identitas: PingFederate.
- Untuk Kata Sandi, masukkan ***your-ping-password***. Gunakan ini hanya untuk jenis Auth adalah Penyedia Identitas: PingFederate.
- Untuk Listen Port, masukkan ***your-listen-port***. Ini adalah port yang didengarkan oleh server lokal. Defaultnya adalah 7890. Ini hanya berlaku untuk plugin Browser SALL.
- Untuk Response Timeout, masukkan ***the-number-of-seconds***. Ini adalah jumlah detik untuk menunggu sebelum waktu habis ketika server iDP mengirim kembali respons. Jumlah minimum detik harus 10. Jika membuat koneksi membutuhkan waktu lebih lama dari ambang ini, maka koneksi dibatalkan. Ini hanya berlaku untuk plugin Browser SALL.
- Untuk URL Login, masukkan ***your-login-url***. Ini hanya berlaku untuk plugin Browser SALL.

Di macOS dan Linux, edit `odbc.ini` file sebagai berikut:

 Note

Semua entri tidak peka huruf besar/kecil.

- Untuk clusterid, masukkan. ***your-cluster-identifier*** Ini adalah nama cluster Amazon Redshift yang dibuat.
- Untuk wilayah, masukkan ***your-cluster-region***. Ini adalah AWS Wilayah cluster Amazon Redshift yang dibuat.
- Untuk database, masukkan ***your-database-name***. Ini adalah nama database yang Anda coba akses di cluster Amazon Redshift.
- Untuk lokal, masukkan ***en-us***. Ini adalah bahasa yang menampilkan pesan kesalahan.
- Untuk iam, masukkan ***1***. Nilai ini menentukan driver untuk mengautentikasi menggunakan kredensial IAM.
- Untuk plugin_name, lakukan salah satu hal berikut:
 - Untuk konfigurasi Ping Identity, masukkan ***BrowserSAML***. Ini adalah metode otentikasi yang digunakan driver ODBC untuk mengautentikasi ke Ping Identity.
 - Untuk Ping Identity dengan konfigurasi masuk tunggal, masukkan. ***Ping*** Ini adalah metode otentikasi yang digunakan driver ODBC untuk mengautentikasi menggunakan Ping Identity dengan sistem masuk tunggal.
- Untuk uid, masukkan ***your-ping-username***. Ini adalah nama pengguna akun Microsoft Azure yang Anda gunakan untuk sistem masuk tunggal yang memiliki izin ke cluster yang Anda coba autentikasi. Gunakan ini hanya untuk plugin_name adalah Ping.
- Untuk pwd, masukkan. ***your-ping-password*** Gunakan ini hanya untuk plugin_name adalah Ping.
- Untuk login_url, masukkan. ***your-login-url*** Ini adalah Initiate single sign-on URL yang mengembalikan SAML Response. Ini hanya berlaku untuk plugin Browser SALL.
- Untuk idp_response_timeout, masukkan. ***the-number-of-seconds*** Ini adalah periode waktu yang ditentukan dalam hitungan detik untuk menunggu respons dari PingOne Identity. Ini hanya berlaku untuk plugin Browser SALL.
- Untuk listen_port, masukkan. ***your-listen-port*** Ini adalah port yang didengarkan oleh server lokal. Defaultnya adalah 7890. Ini hanya berlaku untuk plugin Browser SALL.

Di macOS dan Linux, edit juga pengaturan profil untuk menambahkan ekspor berikut.

```
export ODBCINI=/opt/amazon/redshift/Setup/odbc.ini
```

```
export ODBCINSTINI=/opt/amazon/redshift/Setup/odbcinst.ini
```

Okta

Anda dapat menggunakan Okta sebagai penyedia identitas (iDP) untuk mengakses kluster Amazon Redshift Anda. Tutorial ini menunjukkan kepada Anda bagaimana Anda dapat menggunakan Okta sebagai penyedia identitas (iDP) untuk mengakses cluster Amazon Redshift Anda.

Langkah 1: Siapkan Okta dan AWS akun Anda untuk saling percaya

Prosedur berikut menjelaskan cara mengatur hubungan kepercayaan.

Untuk mengatur Okta dan AWS akun Anda agar saling percaya

1. Buat atau gunakan kluster Amazon Redshift yang ada agar pengguna Okta dapat terhubung. Untuk mengkonfigurasi koneksi, properti tertentu dari cluster ini diperlukan, seperti pengidentifikasi cluster. Untuk informasi selengkapnya, lihat [Membuat Cluster](#).
2. Tambahkan Amazon Redshift sebagai aplikasi baru di portal Okta. Untuk langkah-langkah rinci, lihat [dokumentasi Okta](#).
 - Pilih Tambahkan Aplikasi.
 - Di bawah Tambahkan Aplikasi, pilih Buat Aplikasi Baru.
 - Pada halaman Create a New Add Application Integration, untuk Platform, pilih Web.
 - Untuk metode Sign on, pilih SAMP v2.0.
 - Pada halaman Pengaturan Umum, untuk nama Aplikasi, masukkan ***your-redshift-saml-sso-name***. Ini adalah nama aplikasi Anda.
 - Pada halaman Pengaturan SAMP, untuk Single sign on URL, masukkan ***your-redshift-local-host-url***. Ini adalah host dan port lokal yang dialihkan oleh pernyataan SAMP, misalnya. `http://localhost:7890/redshift/`
3. Gunakan tanda tunggal pada nilai URL sebagai URL Penerima dan URL Tujuan.
4. Untuk Menandatangani, pilih Tanda Pernyataan.

5. Untuk URI Audiens (SP Entity ID), masukkan **urn:amazon:webservices** klaim, seperti yang ditunjukkan pada tabel berikut.
6. Di bagian Pengaturan Lanjut, untuk ID Penerbit SAMP, masukkan ***your-Identity-Provider-Issuer-ID***, yang dapat Anda temukan di bagian Lihat Petunjuk Pengaturan.
7. Di bagian Pernyataan Atribut, buat klaim seperti yang ditunjukkan pada tabel berikut.

Nama klaim	Nilai
https://aws.amazon.com/SAML/Attributes/Role	arn:aws:iam: ::role/, arn:aws:iam: ::saml-provider/ <i>123456789012 Okta 123456789012 Okta</i>
https://aws.amazon.com/SAML/Attributes/RoleSessionName	user.email
https://redshift.amazon.com/SAML/Attributes/AutoCreate	“benar”
https://redshift.amazon.com/SAML/Attributes/DbUser	user.email

8. Di bagian App Embed Link, temukan URL yang dapat Anda gunakan sebagai URL login untuk plugin Browser SAMP.
9. Buat penyedia identitas SALL IAM di konsol IAM. Dokumen metadata yang Anda berikan adalah file XMLmetadata federasi yang Anda simpan saat Anda mengatur Okta. Untuk langkah-langkah mendetail, lihat [Membuat dan Mengelola Penyedia Identitas IAM \(Konsol\)](#) di Panduan Pengguna IAM.
10. Buat peran IAM untuk federasi SAMP 2.0 di konsol IAM. Untuk langkah-langkah mendetail, lihat [Membuat Peran untuk SAMP](#) di Panduan Pengguna IAM.
11. Buat kebijakan IAM yang dapat Anda lampirkan ke peran IAM yang Anda buat untuk federasi SAMP 2.0 di konsol IAM. Untuk langkah-langkah mendetail, lihat [Membuat Kebijakan IAM \(Konsol\)](#) di Panduan Pengguna IAM. Untuk contoh Azure AD, lihat [Menyiapkan otentikasi masuk tunggal JDBC atau ODBC](#).

Langkah 2: Siapkan JDBC atau ODBC untuk otentikasi ke Okta

JDBC

Untuk mengatur JDBC untuk otentikasi ke Okta

- Konfigurasi klien database Anda untuk terhubung ke cluster Anda melalui JDBC menggunakan sistem masuk tunggal Okta.

Anda dapat menggunakan klien apa pun yang menggunakan driver JDBC untuk terhubung menggunakan sistem masuk tunggal Okta atau menggunakan bahasa seperti Java untuk terhubung menggunakan skrip. Untuk informasi instalasi dan konfigurasi, lihat [Mengonfigurasi koneksi untuk driver JDBC versi 2.x untuk Amazon Redshift](#).

Misalnya, Anda dapat menggunakan SQLWorkbench/J sebagai klien. Ketika Anda mengkonfigurasi SQLWorkbench /J, URL database Anda menggunakan format berikut.

```
jdbc:redshift:iam://cluster-identifier:us-west-1/dev
```

Jika Anda menggunakan SQLWorkbench/J sebagai klien, ambil langkah-langkah berikut:

- a. Mulai SQL Workbench/J. Di halaman Select Connection Profile, tambahkan Profil Grup, misalnya. **Okta**
- b. Untuk Profil Koneksi ***your-connection-profile-name***, masukkan, misalnya **Okta**.
- c. Pilih Kelola Driver, dan pilih Amazon Redshift. Pilih ikon Open Folder di sebelah Library, lalu pilih file JDBC .jar yang sesuai.
- d. Pada halaman Pilih Profil Koneksi, tambahkan informasi ke profil koneksi sebagai berikut:
 - Untuk Pengguna, masukkan nama pengguna Okta Anda. Ini adalah nama pengguna akun Okta yang Anda gunakan untuk sistem masuk tunggal yang memiliki izin ke klaster yang Anda coba autentikasi.
 - Untuk Kata Sandi, masukkan kata sandi Okta Anda.
 - Untuk Driver, pilih Amazon Redshift (com.amazon.redshift.jdbc.driver).
 - Untuk URL, masukkan ***jdbc:redshift:iam://your-cluster-identifier:your-cluster-region/your-database-name***.
- e. Pilih Extended Properties dan lakukan salah satu hal berikut:

- Untuk `login_url`, masukkan. ***your-okta-ssso-login-url*** Nilai ini menentukan URL untuk menggunakan sistem masuk tunggal sebagai otentikasi untuk masuk ke Okta.
- Untuk sistem masuk tunggal Okta, untuk `plugin_name`, masukkan. **`com.amazon.redshift.plugin.OktaCredentialsProvider`** Nilai ini menentukan driver untuk menggunakan Okta single sign-on sebagai metode otentikasi.
- Untuk sistem masuk tunggal Okta dengan MFA, untuk `plugin_name`, masukkan. **`com.amazon.redshift.plugin.BrowserSamlCredentialsProvider`** Nilai ini menentukan driver untuk menggunakan Okta single sign-on dengan MFA sebagai metode otentikasi.

ODBC

Untuk mengatur ODBC untuk otentikasi ke Okta

- Konfigurasi klien database Anda untuk terhubung ke cluster Anda melalui ODBC menggunakan sistem masuk tunggal Okta.

Amazon Redshift menyediakan driver ODBC untuk sistem operasi Linux, Windows, dan macOS. Sebelum Anda menginstal driver ODBC, tentukan apakah alat klien SQL Anda 32-bit atau 64-bit. Instal driver ODBC yang sesuai dengan persyaratan alat klien SQL Anda.

Di Windows, di halaman Pengaturan DSN Driver Amazon Redshift ODBC, di bawah Pengaturan Koneksi, masukkan informasi berikut:

- Untuk Nama Sumber Data, masukkan ***your-DSN***. Ini menentukan nama sumber data yang digunakan sebagai nama profil ODBC.
- Untuk jenis Auth, lakukan salah satu hal berikut:
 - Untuk konfigurasi masuk tunggal Okta, pilih. **Identity Provider: Okta** Ini adalah metode otentikasi yang digunakan driver ODBC untuk mengautentikasi menggunakan Okta single sign-on.
 - Untuk sistem masuk tunggal Okta dengan konfigurasi MFA, pilih. **Identity Provider: Browser SAML** Ini adalah metode otentikasi yang digunakan driver ODBC untuk mengautentikasi menggunakan Okta single sign-on dengan MFA.
- Untuk ID Cluster, masukkan ***your-cluster-identifier***.
- Untuk Wilayah, masukkan ***your-cluster-region***.

- Untuk Database, masukkan ***your-database-name***.
- Untuk Pengguna, masukkan ***your-okta-username***. Ini adalah nama pengguna untuk akun Okta yang Anda gunakan untuk sistem masuk tunggal yang memiliki izin ke cluster yang Anda coba autentikasi. Gunakan ini hanya untuk jenis Auth adalah Penyedia Identitas: Okta.
- Untuk Kata Sandi, masukkan ***your-okta-password***. Gunakan ini hanya untuk jenis Auth adalah Penyedia Identitas: Okta.

Di macOS dan Linux, edit `odbc.ini` file sebagai berikut:

 Note

Semua entri tidak peka huruf besar/kecil.

- Untuk clusterid, masukkan. ***your-cluster-identifier*** Ini adalah nama cluster Amazon Redshift yang dibuat.
- Untuk wilayah, masukkan ***your-cluster-region***. Ini adalah AWS Wilayah cluster Amazon Redshift yang dibuat.
- Untuk database, masukkan ***your-database-name***. Ini adalah nama database yang Anda coba akses di cluster Amazon Redshift.
- Untuk lokal, masukkan ***en-us***. Ini adalah bahasa yang menampilkan pesan kesalahan.
- Untuk iam, masukkan ***1***. Nilai ini menentukan driver untuk mengautentikasi menggunakan kredensial IAM.
- Untuk plugin_name, lakukan salah satu hal berikut:
 - Untuk sistem masuk tunggal Okta dengan konfigurasi MFA, masukkan. ***BrowserSAML*** Ini adalah metode otentikasi yang digunakan driver ODBC untuk mengautentikasi ke Okta single sign-on dengan MFA.
 - Untuk konfigurasi masuk tunggal Okta, masukkan. ***Okta*** Ini adalah metode otentikasi yang digunakan driver ODBC untuk mengautentikasi menggunakan Okta single sign-on.
- Untuk uid, masukkan ***your-okta-username***. Ini adalah nama pengguna akun Okta yang Anda gunakan untuk sistem masuk tunggal yang memiliki izin ke cluster yang Anda coba autentikasi. Gunakan ini hanya untuk plugin_name adalah Okta.

- Untuk pwd, masukkan. ***your-okta-password*** Gunakan ini hanya untuk plugin_name adalah Okta.
- Untuk login_url, masukkan. ***your-login-url*** Ini adalah Initiate single sign-on URL yang mengembalikan SAML Response. Ini hanya berlaku untuk plugin Browser SAMP.
- Untuk idp_response_timeout, masukkan. ***the-number-of-seconds*** Ini adalah periode waktu yang ditentukan dalam hitungan detik untuk menunggu respons dari PingOne. Ini hanya berlaku untuk plugin Browser SAMP.
- Untuk listen_port, masukkan. ***your-listen-port*** Ini adalah port yang didengarkan oleh server lokal. Defaultnya adalah 7890. Ini hanya berlaku untuk plugin Browser SAMP.

Di macOS dan Linux, edit juga pengaturan profil untuk menambahkan ekspor berikut.

```
export ODBCINI=/opt/amazon/redshift/Setup/odbc.ini
```

```
export ODBCINSTINI=/opt/amazon/redshift/Setup/odbcinst.ini
```

Mengotorisasi Amazon Redshift untuk AWS mengakses layanan atas nama Anda

Beberapa fitur Amazon Redshift memerlukan Amazon Redshift untuk mengakses layanan AWS lain atas nama Anda. Misalnya, perintah [COPY](#) dan [UNLOAD](#) dapat memuat atau membongkar data ke cluster Amazon Redshift menggunakan bucket Amazon S3. Perintah [CREATE EXTERNAL FUNCTION](#) dapat memanggil fungsi AWS Lambda menggunakan fungsi yang ditentukan pengguna Lambda skalar (UDF). Amazon Redshift Spectrum dapat menggunakan katalog data di Amazon AWS Glue Athena atau. Agar kluster Amazon Redshift bertindak atas nama Anda, Anda memberikan kredensi keamanan ke kluster Anda. Metode yang lebih disukai untuk menyediakan kredensial keamanan adalah dengan menentukan peran AWS Identity and Access Management (IAM). Untuk COPY dan UNLOAD, Anda dapat memberikan kredensi sementara.

Pengguna membutuhkan akses terprogram jika mereka ingin berinteraksi dengan AWS luar. Konsol Manajemen AWS Cara untuk memberikan akses terprogram tergantung pada jenis pengguna yang mengakses AWS.

Untuk memberi pengguna akses programatis, pilih salah satu opsi berikut.

Pengguna mana yang membutuhkan akses programatis?	Untuk	Oleh
IAM	(Disarankan) Gunakan kredensial konsol sebagai kredensial sementara untuk menandatangani permintaan terprogram ke,, atau. AWS CLI AWS SDKs AWS APIs	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan. • Untuk itu AWS CLI, lihat Login untuk pengembangan AWS lokal di Panduan AWS Command Line Interface Pengguna. • Untuk AWS SDKs, lihat Login untuk pengembangan AWS lokal di Panduan Referensi Alat AWS SDKs dan Alat.
Identitas tenaga kerja (Pengguna yang dikelola di Pusat Identitas IAM)	Gunakan kredensi sementara untuk menandatangani permintaan terprogram ke AWS CLI,, AWS SDKs atau. AWS APIs	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan. • Untuk AWS CLI, lihat Mengkonfigurasi yang akan AWS CLI digunakan AWS IAM Identity Center dalam Panduan AWS Command Line Interface Pengguna. • Untuk AWS SDKs, alat, dan AWS APIs, lihat Autentikasi Pusat Identitas IAM di Panduan Referensi Alat AWS SDKs dan Alat.
IAM	Gunakan kredensi sementara untuk menandatangani	Mengikuti petunjuk dalam Menggunakan kredensial

Pengguna mana yang membutuhkan akses programatis?	Untuk	Oleh
	permintaan terprogram ke AWS CLI,, AWS SDKs atau. AWS APIs	sementara dengan AWS sumber daya di Panduan Pengguna IAM.
IAM	(Tidak direkomendasikan) Gunakan kredensi jangka panjang untuk menandatangani permintaan terprogram ke AWS CLI,, AWS SDKs atau. AWS APIs	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan. • Untuk mengetahui AWS CLI, lihat Mengautentikasi menggunakan kredensial pengguna IAM di Panduan Pengguna. AWS Command Line Interface • Untuk AWS SDKs dan alat, lihat Mengautentikasi menggunakan kredensial jangka panjang di Panduan Referensi Alat AWS SDKs dan Alat. • Untuk AWS APIs, lihat Mengelola kunci akses untuk pengguna IAM di Panduan Pengguna IAM.

Berikut ini, cari tahu cara membuat peran IAM dengan izin yang sesuai untuk mengakses layanan lain AWS . Anda juga perlu mengaitkan peran dengan klaster Anda dan menentukan Nama Sumber Daya Amazon (ARN) peran saat menjalankan perintah Amazon Redshift. Untuk informasi selengkapnya, lihat [Mengotorisasi operasi COPY, UNLOAD, CREATE EXTERNAL FUNCTION, dan CREATE EXTERNAL SCHEMA menggunakan peran IAM](#).

Selain itu, pengguna super dapat memberikan hak istimewa ASSUMEROLE kepada pengguna dan grup tertentu untuk menyediakan akses ke peran untuk operasi COPY dan UNLOAD. Untuk selengkapnya, lihat [GRANT](#) di Panduan Pengembang Database Amazon Redshift.

Membuat peran IAM untuk memungkinkan klaster Amazon Redshift Anda mengakses layanan AWS

Membuat peran IAM untuk memungkinkan klaster Amazon Redshift Anda mengakses layanan AWS

Untuk membuat peran IAM untuk mengizinkan klaster Amazon Redshift Anda berkomunikasi dengan layanan AWS lain atas nama Anda, lakukan langkah-langkah berikut. Nilai yang digunakan di bagian ini adalah contoh, Anda dapat memilih nilai berdasarkan kebutuhan Anda.

Untuk membuat peran IAM untuk memungkinkan Amazon Redshift mengakses layanan AWS

1. Buka [konsol IAM](#).
2. Di panel navigasi, pilih Peran.
3. Pilih Buat peran.
4. Pilih AWS layanan, lalu pilih Redshift.
5. Di bawah Pilih kasus penggunaan Anda, pilih Redshift - Customizable dan kemudian pilih Next: Permissions. Halaman kebijakan izin lampirkan akan muncul.
6. Untuk akses ke Amazon S3 menggunakan COPY, sebagai contoh, Anda dapat menggunakan **AmazonS3ReadOnlyAccess** dan menambahkan. Untuk akses ke Amazon S3 menggunakan COPY atau UNLOAD, sebaiknya Anda membuat kebijakan terkelola yang membatasi akses ke bucket dan awalan yang diinginkan. Untuk operasi baca dan tulis, kami sarankan untuk menerapkan hak istimewa paling sedikit dan membatasi hanya bucket Amazon S3 dan awalan kunci yang diperlukan Amazon Redshift.

Untuk akses untuk memanggil fungsi Lambda untuk perintah CREATE EXTERNAL FUNCTION, tambahkan. **AWSLambdaRole**

Untuk Redshift Spectrum, selain akses Amazon S3, tambahkan atau.

AWSGlueConsoleFullAccess AmazonAthenaFullAccess

Pilih Berikutnya: Penandaan.

7. Halaman Tambahkan penandaan akan muncul. Anda dapat menambahkan tanda secara opsional. Pilih Berikutnya: Tinjauan.

8. Untuk nama Peran, ketikkan nama untuk peran Anda, misalnya **RedshiftCopyUnload**. Pilih **Buat peran**.
9. Peran baru tersedia untuk semua pengguna di cluster yang menggunakan peran tersebut. Untuk membatasi akses hanya ke pengguna tertentu pada kluster tertentu, atau ke cluster di wilayah tertentu, edit hubungan kepercayaan untuk peran tersebut. Untuk informasi selengkapnya, lihat [Membatasi akses ke peran IAM](#).
10. Kaitkan peran dengan cluster Anda. Anda dapat mengaitkan peran IAM dengan kluster saat membuat kluster, atau menambahkan peran tersebut ke kluster yang ada. Untuk informasi selengkapnya, lihat [Mengaitkan peran IAM dengan cluster](#).

 Note

Untuk membatasi akses ke data tertentu, gunakan peran IAM yang memberikan hak istimewa paling sedikit yang diperlukan.

Membatasi akses ke peran IAM

Secara default, peran IAM yang tersedia untuk kluster Amazon Redshift tersedia untuk semua pengguna di cluster tersebut. Anda dapat memilih untuk membatasi peran IAM ke pengguna database Amazon Redshift tertentu pada kluster tertentu atau ke wilayah tertentu.

Untuk mengizinkan hanya pengguna database tertentu untuk menggunakan peran IAM, lakukan langkah-langkah berikut.

Untuk mengidentifikasi pengguna database tertentu dengan akses ke peran IAM

1. Identifikasi Nama Sumber Daya Amazon (ARN) untuk pengguna database di kluster Amazon Redshift Anda. ARN untuk pengguna database dalam format:
`arn:aws:redshift:region:account-id:dbuser:cluster-name/user-name`

Untuk Amazon Redshift Tanpa Server gunakan format ARN berikut.

`arn:aws:redshift:region:account-id:dbuser:serverless-account-id-workgroup-id/user-name`

2. Buka [konsol IAM](#).
3. Di panel navigasi, pilih **Peran**.
4. Pilih peran IAM yang ingin Anda batasi untuk pengguna database Amazon Redshift tertentu.

- Pilih tab Trust Relationships, lalu pilih Edit Trust Relationship. Peran IAM baru yang memungkinkan Amazon Redshift mengakses layanan AWS lain atas nama Anda memiliki hubungan kepercayaan sebagai berikut:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "redshift.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- Tambahkan kondisi ke bagian `sts:AssumeRole` tindakan dari hubungan kepercayaan yang membatasi `sts:ExternalId` bidang ke nilai yang Anda tentukan. Sertakan ARN untuk setiap pengguna database yang ingin Anda berikan akses ke peran tersebut. ID eksternal dapat berupa string unik apa pun.

Misalnya, hubungan kepercayaan berikut menetapkan bahwa hanya pengguna database `user1` dan `user2` di kluster `my-cluster` di wilayah yang `us-west-2` memiliki izin untuk menggunakan peran IAM ini.

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. Pilih Perbarui Kebijakan Kepercayaan.

Membatasi peran IAM ke Wilayah AWS

Anda dapat membatasi peran IAM agar hanya dapat diakses di Wilayah tertentu AWS . Secara default, peran IAM untuk Amazon Redshift tidak dibatasi untuk wilayah tunggal mana pun.

Untuk membatasi penggunaan peran IAM berdasarkan wilayah, lakukan langkah-langkah berikut.

Untuk mengidentifikasi wilayah yang diizinkan untuk peran IAM

1. Buka [konsol IAM](https://console.aws.amazon.com/) di <https://console.aws.amazon.com/>.
2. Di panel navigasi, pilih Peran.
3. Pilih peran yang ingin Anda ubah dengan wilayah tertentu.
4. Pilih tab Trust Relationships dan kemudian pilih Edit Trust Relationship. Peran IAM baru yang memungkinkan Amazon Redshift mengakses layanan AWS lain atas nama Anda memiliki hubungan kepercayaan sebagai berikut:

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "redshift.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}

```

```
]
}
```

5. Ubah Service daftar untuk daftar wilayah tertentu yang ingin Anda izinkan penggunaan peran tersebut. Principal Setiap wilayah dalam Service daftar harus dalam format berikut: `redshift.region.amazonaws.com`.

Misalnya, hubungan kepercayaan yang diedit berikut mengizinkan penggunaan peran IAM di `us-east-1` dan `us-west-2` wilayah saja.

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. Pilih Perbarui Kebijakan Kepercayaan

Merantai peran IAM di Amazon Redshift

Saat Anda melampirkan peran ke klaster, klaster Anda dapat mengambil peran tersebut untuk mengakses Amazon S3, Amazon Athena AWS Glue, AWS Lambda dan atas nama Anda. Jika peran yang dilampirkan ke klaster Anda tidak memiliki akses ke sumber daya yang diperlukan, Anda dapat merantai peran lain, mungkin milik akun lain. Cluster Anda kemudian sementara mengasumsikan peran berantai untuk mengakses data. Anda juga dapat memberikan akses lintas akun dengan merantai peran. Setiap peran dalam rantai mengasumsikan peran berikutnya dalam rantai, sampai cluster mengambil peran di akhir rantai. Jumlah maksimum peran IAM yang dapat Anda kaitkan

tunduk pada kuota. Untuk informasi selengkapnya, lihat kuota “Peran IAM klaster untuk Amazon Redshift untuk mengakses layanan AWS lain” di [Kuota untuk objek Amazon Redshift](#)

Note

Anda harus menentukan peran IAM agar rantai berfungsi dengan benar.

Misalnya, Perusahaan A ingin mengakses data dalam bucket Amazon S3 milik Perusahaan B. Perusahaan A menciptakan peran AWS layanan untuk Amazon Redshift RoleA bernama dan menempelkannya ke klaster mereka. Perusahaan B membuat peran bernama RoleB yang diizinkan untuk mengakses data di bucket Perusahaan B. Untuk mengakses data dalam bucket B Perusahaan, Perusahaan A menjalankan perintah COPY menggunakan iam_role parameter yang berantai RoleA dan RoleB. Selama operasi COPY, untuk RoleA sementara mengasumsikan RoleB untuk mengakses bucket Amazon S3.

Untuk peran rantai, Anda membangun hubungan kepercayaan antara peran. Peran yang mengasumsikan peran lain (misalnya, RoleA) harus memiliki kebijakan izin yang memungkinkannya mengambil peran berantai berikutnya (misalnya, RoleB). Pada gilirannya, peran yang meneruskan izin (RoleB) harus memiliki kebijakan kepercayaan yang memungkinkannya meneruskan izinnya ke peran berantai sebelumnya (). RoleA Untuk informasi selengkapnya, lihat [Menggunakan peran IAM](#) di Panduan Pengguna IAM.

Peran pertama dalam rantai harus menjadi peran yang melekat pada cluster. Peran pertama, dan setiap peran berikutnya yang mengambil peran berikutnya dalam rantai, harus memiliki kebijakan yang mencakup pernyataan tertentu. Pernyataan ini memiliki Allow efek pada sts:AssumeRole tindakan dan Nama Sumber Daya Amazon (ARN) dari peran berikutnya dalam suatu Resource elemen. Dalam contoh kami, RoleA memiliki kebijakan izin berikut yang memungkinkannya untuk berasumsi RoleB, dimiliki oleh AWS akun 210987654321.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1487639602000",
      "Effect": "Allow",
```

```

        "Action": [
            "sts:AssumeRole"
        ],
        "Resource": "arn:aws:iam::111122223333:role/RoleB"
    }
]
}

```

Peran yang beralih ke peran lain harus membangun hubungan kepercayaan dengan peran yang mengambil peran atau dengan AWS akun yang memiliki peran tersebut. Dalam contoh kami, RoleB memiliki kebijakan kepercayaan berikut untuk membangun hubungan kepercayaan dengan RoleA.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/RoleA"
      }
    }
  ]
}

```

Kebijakan kepercayaan berikut menetapkan hubungan kepercayaan dengan pemilikRoleA, AWS akun123456789012.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Principal": {

```

```

    "AWS": "arn:aws:iam::123456789012:root"
  }
}
]
}

```

Note

Untuk membatasi otorisasi rantai peran untuk pengguna tertentu, tentukan kondisi. Untuk informasi selengkapnya, lihat [Membatasi akses ke peran IAM](#).

Saat menjalankan perintah UNLOAD, COPY, CREATE EXTERNAL FUNCTION, atau CREATE EXTERNAL SCHEMA, Anda merantai peran dengan menyertakan daftar peran yang dipisahkan koma dalam ARNs parameter. `iam_role` Berikut ini menunjukkan sintaks untuk peran rantai dalam parameter. `iam_role`

```

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

Seluruh rantai peran tertutup dalam tanda kutip tunggal dan tidak boleh berisi spasi.

Dalam contoh berikut, `RoleA` dilampirkan ke cluster milik AWS akun123456789012. `RoleB`, yang termasuk dalam akun210987654321, memiliki izin untuk mengakses bucket bernama `s3://companyb/redshift/`. Contoh rantai berikut `RoleA` dan `RoleB` untuk MEMBONGKAR data ke bucket `s3://companyb/redshift/`.

```

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';

```

Contoh berikut menggunakan perintah COPY untuk memuat data yang diturunkan dalam contoh sebelumnya.

```
copy venue
from 's3://companyb/redshift/venue_pipe_'
iam_role 'arn:aws:iam::123456789012:role/RoleA,arn:aws:iam::210987654321:role/RoleB';
```

Dalam contoh berikut, CREATE EXTERNAL SCHEMA menggunakan peran berantai untuk mengambil peran. RoleB

```
create external schema spectrumexample from data catalog
database 'exampledb' region 'us-west-2'
iam_role 'arn:aws:iam::123456789012:role/RoleA,arn:aws:iam::210987654321:role/RoleB';
```

Dalam contoh berikut, CREATE EXTERNAL FUNCTION menggunakan peran berantai untuk mengambil peranRoleB.

```
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';
```

Mengotorisasi operasi COPY, UNLOAD, CREATE EXTERNAL FUNCTION, dan CREATE EXTERNAL SCHEMA menggunakan peran IAM

Anda dapat menggunakan perintah [COPY](#) untuk memuat (atau mengimpor) data ke Amazon Redshift dan perintah [UNLOAD](#) untuk membongkar (atau mengeksport) data dari Amazon Redshift. Anda dapat menggunakan perintah CREATE EXTERNAL FUNCTION untuk membuat fungsi yang ditentukan pengguna yang memanggil fungsi dari AWS Lambda

Saat menggunakan Amazon Redshift Spectrum, Anda menggunakan perintah [CREATE EXTERNAL SCHEMA](#) untuk menentukan lokasi bucket Amazon S3 yang berisi data Anda. Saat menjalankan perintah COPY, UNLOAD, atau CREATE EXTERNAL SCHEMA, Anda memberikan kredensi keamanan. Kredensi ini mengizinkan kluster Amazon Redshift Anda untuk membaca atau menulis data ke dan dari tujuan target Anda, seperti bucket Amazon S3.

Saat menjalankan CREATE EXTERNAL FUNCTION, Anda memberikan kredensi keamanan menggunakan parameter peran IAM. Kredensi ini mengizinkan kluster Amazon Redshift Anda untuk menjalankan fungsi Lambda. AWS Lambda Metode yang lebih disukai untuk menyediakan kredensial keamanan adalah dengan menentukan peran AWS Identity and Access Management (IAM). Untuk COPY dan UNLOAD, Anda dapat memberikan kredensi sementara. Untuk informasi

tentang membuat peran IAM, lihat [Mengotorisasi Amazon Redshift untuk AWS mengakses layanan atas nama Anda](#).

Pengguna membutuhkan akses terprogram jika mereka ingin berinteraksi dengan AWS luar. Konsol Manajemen AWS Cara untuk memberikan akses terprogram tergantung pada jenis pengguna yang mengakses AWS.

Untuk memberi pengguna akses programatis, pilih salah satu opsi berikut.

Pengguna mana yang membutuhkan akses programatis?	Untuk	Oleh
IAM	(Disarankan) Gunakan kredensyal konsol sebagai kredensyal sementara untuk menandatangani permintaan terprogram ke,, atau. AWS CLI AWS SDKs AWS APIs	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan. • Untuk itu AWS CLI, lihat Login untuk pengembangan AWS lokal di Panduan AWS Command Line Interface Pengguna. • Untuk AWS SDKs, lihat Login untuk pengembangan AWS lokal di Panduan Referensi Alat AWS SDKs dan Alat.
Identitas tenaga kerja (Pengguna yang dikelola di Pusat Identitas IAM)	Gunakan kredensi sementara untuk menandatangani permintaan terprogram ke AWS CLI,, AWS SDKs atau. AWS APIs	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan. • Untuk AWS CLI, lihat Mengkonfigurasi yang akan AWS CLI digunakan AWS IAM Identity Center dalam Panduan AWS Command Line Interface Pengguna.

Pengguna mana yang membutuhkan akses programatis?	Untuk	Oleh
		• Untuk AWS SDKs, alat, dan AWS APIs, lihat Autentikasi Pusat Identitas IAM di Panduan Referensi Alat AWS SDKs dan Alat.
IAM	Gunakan kredensi sementara untuk menandatangani permintaan terprogram ke AWS CLI,, AWS SDKs atau. AWS APIs	Mengikuti petunjuk dalam Menggunakan kredensi sementara dengan AWS sumber daya di Panduan Pengguna IAM.
IAM	(Tidak direkomendasikan) Gunakan kredensi jangka panjang untuk menandatangani permintaan terprogram ke AWS CLI,, AWS SDKs atau. AWS APIs	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan. • Untuk mengetahui AWS CLI, lihat Mengautentikasi menggunakan kredensi pengguna IAM di Panduan Pengguna. AWS Command Line Interface • Untuk AWS SDKs dan alat, lihat Mengautentikasi menggunakan kredensial jangka panjang di Panduan Referensi Alat AWS SDKs dan Alat. • Untuk AWS APIs, lihat Mengelola kunci akses untuk pengguna IAM di Panduan Pengguna IAM.

Langkah-langkah untuk menggunakan peran IAM adalah sebagai berikut:

- Buat peran IAM untuk digunakan dengan cluster Amazon Redshift Anda.
- Kaitkan peran IAM dengan cluster.
- Sertakan ARN peran IAM saat Anda memanggil perintah COPY, UNLOAD, CREATE EXTERNAL SCHEMA, atau CREATE EXTERNAL FUNCTION.

Mengaitkan peran IAM dengan cluster

Setelah Anda membuat peran IAM yang mengizinkan Amazon Redshift untuk mengakses layanan AWS lain untuk Anda, Anda harus mengaitkan peran tersebut dengan kluster Amazon Redshift. Anda harus melakukan ini sebelum Anda dapat menggunakan peran untuk memuat atau membongkar data.

Izin diperlukan untuk mengaitkan peran IAM dengan kluster

Untuk mengaitkan peran IAM dengan kluster, pengguna harus memiliki `iam:PassRole` izin untuk peran IAM tersebut. Izin ini memungkinkan administrator untuk membatasi peran IAM mana yang dapat dikaitkan pengguna dengan kluster Amazon Redshift. Sebagai praktik terbaik, kami merekomendasikan untuk melampirkan kebijakan izin ke peran IAM dan kemudian menetapkan ke pengguna dan grup sesuai kebutuhan. Untuk informasi selengkapnya, lihat [Identitas dan manajemen akses di Amazon Redshift](#).

Contoh berikut menunjukkan kebijakan IAM yang dapat dilampirkan ke pengguna yang memungkinkan pengguna untuk mengambil tindakan ini:

- Dapatkan detail untuk semua cluster Amazon Redshift yang dimiliki oleh akun pengguna tersebut.
- Kaitkan salah satu dari tiga peran IAM dengan salah satu dari dua cluster Amazon Redshift.

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"
      ]
    }
  ]
}

```

Setelah pengguna memiliki izin yang sesuai, pengguna tersebut dapat mengaitkan peran IAM dengan kluster Amazon Redshift. Peran IAM kemudian siap digunakan dengan perintah COPY atau UNLOAD atau perintah Amazon Redshift lainnya.

Untuk informasi selengkapnya tentang kebijakan IAM, lihat [Ikhtisar kebijakan IAM](#) di Panduan Pengguna IAM.

Mengelola asosiasi peran IAM dengan cluster

Anda dapat mengaitkan peran IAM dengan klaster Amazon Redshift saat membuat klaster. Atau Anda dapat memodifikasi klaster yang ada dan menambahkan atau menghapus satu atau beberapa asosiasi peran IAM.

Ketahui hal-hal berikut:

- Jumlah maksimum peran IAM yang dapat Anda kaitkan tunduk pada kuota.

- Peran IAM dapat dikaitkan dengan beberapa cluster Amazon Redshift.
- Peran IAM dapat dikaitkan dengan cluster Amazon Redshift hanya jika peran IAM dan cluster dimiliki oleh akun yang sama. AWS

Anda dapat mengelola asosiasi peran IAM untuk klaster dengan konsol menggunakan prosedur berikut.

Mengelola asosiasi peran IAM

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster, lalu pilih cluster yang ingin Anda perbarui.
3. Untuk Tindakan, pilih Kelola peran IAM untuk menampilkan daftar peran IAM saat ini yang terkait dengan cluster.
4. Pada halaman Kelola peran IAM, pilih peran IAM yang tersedia untuk ditambahkan, lalu pilih Tambahkan peran IAM.
5. Pilih Selesai untuk menyimpan perubahan Anda.

Anda dapat mengelola asosiasi peran IAM untuk klaster AWS CLI dengan menggunakan pendekatan berikut.

Untuk mengaitkan peran IAM dengan cluster saat cluster dibuat, tentukan Amazon Resource Name (ARN) peran IAM untuk parameter `--iam-role-arns` perintah `create-cluster`. Jumlah maksimum peran IAM yang dapat Anda tambahkan saat memanggil `create-cluster` perintah tunduk pada kuota.

Mengaitkan dan memisahkan peran IAM dengan kluster Amazon Redshift adalah proses asinkron. Anda bisa mendapatkan status semua asosiasi klaster peran IAM dengan memanggil `describe-clusters` perintah.

Contoh berikut mengaitkan dua peran IAM dengan cluster yang baru dibuat bernama `my-redshift-cluster`

```
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/MyRedshiftRole1,arn:aws:iam::123456789012:role/MyRedshiftRole2"
```

```
--iam-role-arns "arn:aws:iam::123456789012:role/RedshiftCopyUnload" \
                "arn:aws:iam::123456789012:role/SecondRedshiftRole"
```

Untuk mengaitkan peran IAM dengan klaster Amazon Redshift yang ada, tentukan Nama Sumber Daya Amazon (ARN) peran IAM untuk `--add-iam-roles` parameter perintah. `modify-cluster-iam-roles` Jumlah maksimum peran IAM yang dapat Anda tambahkan saat memanggil `modify-cluster-iam-roles` perintah tunduk pada kuota.

Contoh berikut mengaitkan peran IAM dengan cluster yang ada bernama `my-redshift-cluster`

```
aws redshift modify-cluster-iam-roles \
    --cluster-identifier "my-redshift-cluster" \
    --add-iam-roles "arn:aws:iam::123456789012:role/RedshiftCopyUnload"
```

Untuk memisahkan peran IAM dari cluster, tentukan ARN dari peran IAM untuk parameter perintah. `--remove-iam-roles` `modify-cluster-iam-roles` `modify-cluster-iam-roles` Jumlah maksimum peran IAM yang dapat Anda hapus saat memanggil `modify-cluster-iam-roles` perintah tunduk pada kuota.

Contoh berikut menghapus asosiasi untuk peran IAM untuk 123456789012 AWS akun dari cluster bernama `my-redshift-cluster`.

```
aws redshift modify-cluster-iam-roles \
    --cluster-identifier "my-redshift-cluster" \
    --remove-iam-roles "arn:aws:iam::123456789012:role/RedshiftCopyUnload"
```

Daftar asosiasi peran IAM untuk cluster menggunakan AWS CLI

Untuk mencantumkan semua peran IAM yang terkait dengan kluster Amazon Redshift, dan status asosiasi peran IAM, panggil perintah tersebut. `describe-clusters` ARN untuk setiap peran IAM yang terkait dengan cluster dikembalikan dalam `IamRoles` daftar seperti yang ditunjukkan pada contoh output berikut.

Peran yang telah dikaitkan dengan cluster menunjukkan status `in-sync`. Peran yang sedang dalam proses dikaitkan dengan cluster menunjukkan status `adding`. Peran yang sedang dipisah dari cluster menunjukkan status `removing`

```
{
```

```
"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"
      }
    ],
    ...
  }
]
```

Untuk informasi selengkapnya tentang penggunaan AWS CLI, lihat [Panduan AWS CLI Pengguna](#).

Membuat peran IAM sebagai default untuk Amazon Redshift

Saat Anda membuat peran IAM melalui konsol Redshift, Amazon Redshift secara terprogram membuat peran dalam peran Akun AWS Anda dan secara otomatis melampirkan kebijakan terkelola yang ada padanya. AWS Pendekatan ini berarti Anda dapat tetap berada di dalam konsol Redshift dan tidak perlu beralih ke konsol IAM untuk pembuatan peran. Untuk kontrol izin yang lebih terperinci untuk peran IAM yang ada yang dibuat di konsol Amazon Redshift, Anda dapat melampirkan kebijakan terkelola yang disesuaikan ke peran IAM.

Peran IAM dibuat di konsol

Saat Anda menggunakan konsol Amazon Redshift untuk membuat peran IAM, Amazon Redshift melacak semua peran IAM yang dibuat melalui konsol. Amazon Redshift memilih sebelumnya peran IAM default terbaru untuk membuat semua cluster baru dan memulihkan cluster dari snapshot.

Anda dapat membuat peran IAM melalui konsol yang memiliki kebijakan dengan izin untuk menjalankan perintah SQL. Perintah-perintah ini termasuk COPY, UNLOAD, CREATE EXTERNAL FUNCTION, CREATE EXTERNAL TABLE, CREATE EXTERNAL SCHEMA, CREATE MODEL, atau CREATE LIBRARY. Secara opsional, Anda bisa mendapatkan kontrol yang lebih terperinci atas akses pengguna ke AWS sumber daya Anda dengan membuat dan melampirkan kebijakan khusus ke peran IAM.

Saat Anda membuat peran IAM dan mengaturnya sebagai default untuk klaster menggunakan konsol, Anda tidak perlu memberikan Nama Sumber Daya Amazon (ARN) peran IAM untuk melakukan autentikasi dan otorisasi.

Peran IAM yang Anda buat melalui konsol untuk klaster Anda memiliki kebijakan `AmazonRedshiftAllCommandsFullAccess` terkelola yang dilampirkan secara otomatis. Peran IAM ini memungkinkan Amazon Redshift untuk menyalin, membongkar, menanyakan, dan menganalisis data AWS untuk sumber daya di akun IAM Anda. Kebijakan terkelola menyediakan akses ke operasi [COPY](#), [UNLOAD](#), [CREATE EXTERNAL FUNCTION](#), [CREATE EXTERNAL SCHEMA](#), [CREATE MODEL](#), dan [CREATE LIBRARY](#). Kebijakan ini juga memberikan izin untuk menjalankan pernyataan SELECT untuk AWS layanan terkait, seperti Amazon S3, Amazon Logs, CloudWatch Amazon AI, SageMaker dan. AWS Glue

Perintah CREATE EXTERNAL FUNCTION, CREATE EXTERNAL SCHEMA, CREATE MODEL, dan CREATE LIBRARY memiliki default kata kunci. Untuk kata kunci ini untuk perintah ini, Amazon Redshift menggunakan peran IAM yang ditetapkan sebagai default dan terkait dengan cluster saat perintah berjalan. Anda dapat menjalankan perintah [DEFAULT_IAM_ROLE untuk memeriksa peran IAM](#) default saat ini yang dilampirkan ke cluster.

Untuk mengontrol hak akses peran IAM yang dibuat dan ditetapkan sebagai default untuk kluster Redshift Anda, gunakan hak istimewa ASSUMEROLE. Kontrol akses ini berlaku untuk pengguna database dan grup ketika mereka menjalankan perintah seperti yang tercantum sebelumnya. Setelah Anda memberikan hak istimewa ASSUMEROLE kepada pengguna atau grup untuk peran IAM, pengguna atau grup dapat mengambil peran tersebut saat menjalankan perintah ini. Dengan menggunakan hak istimewa ASSUMEROLE, Anda dapat memberikan akses ke perintah yang sesuai sesuai kebutuhan.

Menggunakan konsol Amazon Redshift, Anda dapat melakukan hal berikut:

- [Membuat peran IAM sebagai default](#)
- [Menghapus peran IAM dari kluster Anda](#)
- [Mengaitkan peran IAM dengan kluster Anda](#)
- [Menyetel peran IAM sebagai default](#)
- [Membuat peran IAM tidak lagi default untuk kluster Anda](#)

Izin kebijakan AmazonRedshiftAllCommandsFullAccess terkelola

Contoh berikut menunjukkan izin dalam kebijakan AmazonRedshiftAllCommandsFullAccess terkelola yang mengizinkan tindakan tertentu untuk peran IAM yang ditetapkan sebagai default untuk kluster Anda. Peran IAM dengan kebijakan izin yang dilampirkan mengotorisasi apa yang dapat dan tidak dapat dilakukan oleh pengguna atau grup. Dengan izin ini, Anda dapat menjalankan perintah COPY dari Amazon S3, menjalankan UNLOAD, dan menggunakan perintah CREATE MODEL.

```
{
    "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*/*"
    ]
}

```

Contoh berikut menunjukkan izin dalam kebijakan `AmazonRedshiftAllCommandsFullAccess` terkelola yang mengizinkan tindakan tertentu untuk peran IAM yang ditetapkan sebagai default untuk kluster. Peran IAM dengan kebijakan izin yang dilampirkan mengotorisasi apa yang dapat dan tidak dapat dilakukan oleh pengguna atau grup. Dengan izin berikut, Anda dapat menjalankan perintah `CREATE EXTERNAL FUNCTION`.

```

{
    "Action": [
        "lambda:InvokeFunction"
    ],
    "Resource": "arn:aws:lambda:*:*:function:*redshift*"
}

```

Contoh berikut menunjukkan izin dalam kebijakan `AmazonRedshiftAllCommandsFullAccess` terkelola yang mengizinkan tindakan tertentu untuk peran IAM yang ditetapkan sebagai default untuk kluster. Peran IAM dengan kebijakan izin yang dilampirkan mengotorisasi apa yang dapat dan tidak dapat dilakukan oleh pengguna atau grup. Dengan izin berikut, Anda dapat menjalankan perintah `CREATE EXTERNAL SCHEMA` dan `CREATE EXTERNAL TABLE` yang diperlukan untuk Amazon Redshift Spectrum.

```

{
    "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*"
    ]
}

```

Contoh berikut menunjukkan izin dalam kebijakan AmazonRedshiftAllCommandsFullAccess terkelola yang mengizinkan tindakan tertentu untuk peran IAM ditetapkan sebagai default untuk kluster. Peran IAM dengan kebijakan izin yang dilampirkan mengotorisasi apa yang dapat dan tidak dapat dilakukan oleh pengguna atau grup. Dengan izin berikut, Anda dapat menjalankan perintah CREATE EXTERNAL SCHEMA menggunakan kueri federasi.

```

{
    "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"
        }
    }
},
```

Mengelola peran IAM yang dibuat untuk cluster menggunakan konsol

Untuk membuat, memodifikasi, dan menghapus peran IAM yang dibuat dari konsol Amazon Redshift, gunakan bagian Clusters di konsol.

Membuat peran IAM sebagai default

Di konsol, Anda dapat membuat peran IAM untuk klaster yang memiliki `AmazonRedshiftAllCommandsFullAccess` kebijakan yang dilampirkan secara otomatis. Peran IAM baru yang Anda buat memungkinkan Amazon Redshift menyalin, memuat, menanyakan, dan menganalisis data dari sumber daya Amazon di akun IAM Anda.

Hanya ada satu set peran IAM sebagai default untuk cluster. Jika Anda membuat peran IAM lain sebagai default cluster saat peran IAM yang ada saat ini ditetapkan sebagai default, peran IAM baru menggantikan peran IAM lainnya sebagai default.

Untuk membuat klaster baru dan peran IAM ditetapkan sebagai default untuk cluster baru

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster. Cluster untuk akun Anda saat ini AWS Region terdaftar. Subset properti dari setiap cluster ditampilkan dalam kolom dalam daftar.
3. Pilih Buat cluster untuk membuat cluster.
4. Ikuti petunjuk di halaman konsol untuk memasukkan properti untuk konfigurasi Cluster. Untuk informasi selengkapnya tentang langkah ini, lihat [Membuat kluster](#).
5. (Opsional) Pilih Muat data sampel untuk memuat kumpulan data sampel ke cluster Amazon Redshift Anda untuk mulai menggunakan editor kueri untuk menanyakan data.

Jika Anda berada di belakang firewall, port database harus berupa port terbuka yang menerima koneksi masuk.

6. Ikuti petunjuk di halaman konsol untuk memasukkan properti untuk konfigurasi Database.
7. Di bawah Izin klaster, dari Kelola peran IAM, pilih Buat peran IAM.
8. Tentukan bucket Amazon S3 agar peran IAM dapat diakses dengan memilih salah satu metode berikut:
 - Pilih Tidak ada bucket Amazon S3 tambahan untuk membuat peran IAM tanpa menentukan bucket Amazon S3 tertentu.
 - Pilih bucket Amazon S3 apa pun untuk memungkinkan pengguna yang memiliki akses ke klaster Amazon Redshift Anda untuk juga mengakses bucket Amazon S3 apa pun dan isinya di dalam Anda. Akun AWS
 - Pilih bucket Amazon S3 Spesifik untuk menentukan satu atau beberapa bucket Amazon S3 yang memiliki izin untuk diakses oleh peran IAM yang dibuat. Kemudian pilih satu atau lebih ember Amazon S3 dari tabel.
9. Pilih Buat peran IAM sebagai default. Amazon Redshift secara otomatis membuat dan menetapkan peran IAM sebagai default untuk klaster Anda.
10. Pilih Buat cluster untuk membuat cluster. Cluster mungkin membutuhkan waktu beberapa menit untuk siap digunakan.

Menghapus peran IAM dari klaster Anda

Anda dapat menghapus satu atau beberapa peran IAM dari klaster Anda.

Untuk menghapus peran IAM dari klaster

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster. Cluster untuk akun Anda saat ini AWS Region terdaftar. Subset properti dari setiap cluster ditampilkan dalam kolom dalam daftar.
3. Pilih klaster tempat Anda ingin menghapus peran IAM.
4. Di bawah Izin cluster, pilih satu atau beberapa peran IAM yang ingin Anda hapus dari cluster.
5. Dari Kelola peran IAM, pilih Hapus peran IAM.

Mengaitkan peran IAM dengan klaster Anda

Anda dapat mengaitkan satu atau beberapa peran IAM dengan klaster Anda.

Untuk mengaitkan peran IAM dengan klaster Anda

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster. Cluster untuk akun Anda saat ini AWS Region terdaftar. Subset properti dari setiap cluster ditampilkan dalam kolom dalam daftar.
3. Pilih cluster yang ingin Anda kaitkan dengan peran IAM.
4. Di bawah Izin cluster, pilih satu atau beberapa peran IAM yang ingin Anda kaitkan dengan klaster.
5. Dari Kelola peran IAM, pilih peran IAM Associate.
6. Pilih satu biji lagi peran IAM untuk diasosiasikan dengan cluster Anda.
7. Pilih peran IAM Associate.

Menyetel peran IAM sebagai default

Anda dapat mengatur peran IAM sebagai default untuk cluster Anda.

Untuk menjadikan peran IAM sebagai default untuk klaster Anda

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster. Cluster untuk akun Anda saat ini AWS Region terdaftar. Subset properti dari setiap cluster ditampilkan dalam kolom dalam daftar.
3. Pilih klaster yang ingin Anda tetapkan peran IAM default.
4. Di bawah Izin klaster, dari peran IAM Terkait, pilih peran IAM yang ingin Anda jadikan sebagai default untuk klaster.
5. Di bawah Set default, pilih Make default.
6. Saat diminta, pilih Setel default untuk mengonfirmasi membuat peran IAM yang ditentukan sebagai default.

Membuat peran IAM tidak lagi default untuk klaster Anda

Anda dapat membuat peran IAM bukan lagi default untuk klaster Anda.

Untuk menghapus peran IAM sebagai default untuk klaster Anda

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster. Cluster untuk akun Anda saat ini AWS Region terdaftar. Subset properti dari setiap cluster ditampilkan dalam kolom dalam daftar.
3. Pilih cluster yang ingin Anda kaitkan dengan peran IAM.
4. Di bawah Izin cluster, dari peran IAM Terkait, pilih peran IAM default.
5. Di bawah Setel default, pilih Hapus default.
6. Saat diminta, pilih Hapus default untuk mengonfirmasi pembersihan peran IAM yang ditentukan sebagai default.

Mengelola peran IAM yang dibuat di cluster menggunakan AWS CLI

Anda dapat mengelola peran IAM yang dibuat di cluster menggunakan. AWS CLI

Untuk membuat klaster Amazon Redshift dengan peran IAM ditetapkan sebagai default

Untuk membuat klaster Amazon Redshift dengan peran IAM, tetapkan sebagai default untuk cluster, gunakan perintah. `aws redshift create-cluster` AWS CLI

AWS CLI Perintah berikut membuat cluster Amazon Redshift dan peran IAM bernama myrole1. AWS CLI Perintah ini juga menetapkan myrole1 sebagai default untuk cluster.

```
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'
```

Cuplikan berikut adalah contoh responsnya.

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

```

Untuk menambahkan satu atau beberapa peran IAM ke cluster Amazon Redshift

Untuk menambahkan satu atau beberapa peran IAM yang terkait dengan cluster, gunakan `aws redshift modify-cluster-iam-roles` AWS CLI perintah.

AWS CLI Perintah berikut menambahkan `myrole3` dan `myrole4` ke cluster.

```

aws redshift modify-cluster-iam-roles \
  --cluster-identifier mycluster \
  --add-iam-roles 'arn:aws:iam::012345678910:role/myrole3'
  'arn:aws:iam::012345678910:role/myrole4'

```

Cuplikan berikut adalah contoh responsnya.

```

{
  "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"
    }
],
...
}
}

```

Untuk menghapus satu atau beberapa peran IAM dari klaster Amazon Redshift

Untuk menghapus satu atau beberapa peran IAM yang terkait dengan cluster, gunakan `aws redshift modify-cluster-iam-roles` AWS CLI perintah.

AWS CLI Perintah berikut menghapus `myrole3` dan `myrole4` dari cluster.

```

aws redshift modify-cluster-iam-roles \
  --cluster-identifier mycluster \
  --remove-iam-roles 'arn:aws:iam::012345678910:role/myrole3'
  'arn:aws:iam::012345678910:role/myrole4'

```

Cuplikan berikut adalah contoh responsnya.

```

{
  "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"
    }
],
...
}
}

```

Untuk menetapkan peran IAM terkait sebagai default untuk cluster

Untuk menetapkan peran IAM terkait sebagai default untuk cluster, gunakan `aws redshift modify-cluster-iam-roles` AWS CLI perintah.

AWS CLI Perintah berikut ditetapkan `myrole2` sebagai default untuk cluster.

```

aws redshift modify-cluster-iam-roles \
  --cluster-identifier mycluster \
  --default-iam-role-arn 'arn:aws:iam::012345678910:role/myrole2'

```

Cuplikan berikut adalah contoh responsnya.

```

{
  "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"
      }
    ]
  }
}

```

```
    ],
    ...
  }
}
```

Untuk menetapkan peran IAM yang tidak terkait sebagai default untuk cluster

Untuk menetapkan peran IAM yang tidak terkait sebagai default untuk cluster, gunakan perintah. `aws redshift modify-cluster-iam-roles` AWS CLI

AWS CLI Perintah berikut ditambahkan `myrole2` ke cluster Amazon Redshift dan menentukannya sebagai default untuk cluster.

```
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'
```

Cuplikan berikut adalah contoh responsnya.

```
{
  "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"
      }
    ],
    ...
  }
}
```

```
}
```

Untuk memulihkan cluster dari snapshot dan menetapkan peran IAM sebagai default untuk itu

Saat memulihkan klaster dari snapshot, Anda dapat mengaitkan peran IAM yang ada atau membuat yang baru dan mengaturnya sebagai default untuk cluster.

Untuk memulihkan klaster Amazon Redshift dari snapshot dan menetapkan peran IAM sebagai default cluster, gunakan perintah `aws redshift restore-from-cluster-snapshot` AWS CLI

AWS CLI Perintah berikut mengembalikan cluster dari snapshot dan menetapkan `myrole2` sebagai default untuk cluster.

```
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'
```

Cuplikan berikut adalah contoh responsnya.

```
{
  "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"
      }
    ],
    ...
  }
}
```

Menggunakan identitas federasi untuk mengelola akses Amazon Redshift ke sumber daya lokal dan tabel eksternal Amazon Redshift Spectrum

Menggunakan federasi identitas AWS dengan kredensial yang disediakan dari `GetDatabaseCredentials` dapat menyederhanakan otorisasi dan akses ke data lokal dan ke data eksternal. Dalam tutorial ini, kami menunjukkan kepada Anda bagaimana menyediakan akses ke sumber daya dengan federasi AWS identitas, alih-alih menggunakan peran IAM tertentu.

Saat ini, untuk memberi pengguna akses ke data eksternal yang berada di Amazon S3, Anda membuat peran IAM dengan izin yang ditentukan dalam kebijakan izin. Kemudian, pengguna dengan peran terlampir dapat mengakses data eksternal. Ini berfungsi, tetapi jika Anda ingin memberikan aturan granular, seperti membuat kolom tertentu tidak tersedia untuk pengguna tertentu, Anda mungkin harus melakukan konfigurasi tambahan pada skema eksternal.

Federasi identitas, dengan kredensial yang disediakan dari `GetDatabaseCredentials`, dapat menyediakan akses ke dan sumber daya AWS Glue Redshift Spectrum dengan aturan IAM granular yang lebih mudah ditentukan dan diubah. Ini membuatnya lebih mudah untuk menerapkan akses yang sesuai dengan aturan bisnis Anda.

Manfaat menggunakan kredensial federasi adalah sebagai berikut:

- Anda tidak perlu mengelola peran IAM terlampir cluster untuk Redshift Spectrum.
- Administrator cluster dapat membuat skema eksternal yang dapat diakses oleh konsumen dengan konteks IAM yang berbeda. Ini berguna, misalnya, untuk melakukan pemfilteran kolom pada tabel, di mana konsumen yang berbeda menanyakan skema eksternal yang sama dan mendapatkan berbagai bidang dalam catatan yang dikembalikan.
- Anda dapat menanyakan Amazon Redshift menggunakan pengguna dengan izin IAM, bukan hanya dengan peran.

Mempersiapkan identitas untuk masuk dengan identitas federasi

Sebelum masuk dengan identitas federasi, Anda harus melakukan beberapa langkah awal. Instruksi ini mengasumsikan Anda memiliki skema eksternal Redshift Spectrum yang sudah ada yang mereferensikan file data yang disimpan di bucket Amazon S3, dan bucket berada di akun yang sama dengan cluster Amazon Redshift atau gudang data Amazon Redshift Tanpa Server.

1. Buat identitas IAM. Ini bisa menjadi pengguna atau peran IAM. Gunakan nama apa pun yang didukung oleh IAM.

2. Lampirkan kebijakan izin ke identitas. Tentukan salah satu dari berikut ini:

- `redshift:GetClusterCredentialsWithIAM`(untuk cluster yang disediakan Amazon Redshift)
- `redshift-serverless:GetCredentials`(untuk Amazon Redshift Tanpa Server)

Anda dapat menambahkan izin dengan editor kebijakan, menggunakan konsol IAM.

Identitas IAM juga memerlukan izin untuk mengakses data eksternal. Berikan akses ke Amazon S3 dengan menambahkan kebijakan AWS terkelola berikut secara langsung:

- `AmazonS3ReadOnlyAccess`
- `AWSGlueConsoleFullAccess`

Kebijakan terkelola terakhir diperlukan jika Anda menggunakan AWS Glue untuk menyiapkan data eksternal Anda. Untuk informasi selengkapnya tentang langkah-langkah pemberian akses ke Amazon Redshift Spectrum, lihat [Membuat peran IAM untuk Amazon Redshift, yang merupakan bagian dari panduan memulai Amazon Redshift](#) dan Redshift Spectrum. Ini menunjukkan langkah-langkah untuk menambahkan kebijakan IAM untuk mengakses Redshift Spectrum.

3. Siapkan klien SQL Anda untuk terhubung ke Amazon Redshift. Gunakan driver Amazon Redshift JDBC, dan tambahkan kredensial pengguna Anda ke properti kredensi alat. Klien seperti SQL Workbench/J bekerja dengan baik untuk ini. Mengatur properti `extended client-connection` berikut:

- `AccessKeyID` — Pengidentifikasi kunci akses Anda.
- `SecretAccessKey`— Kunci akses rahasia Anda. (Perhatikan risiko keamanan mentransmisikan kunci rahasia jika Anda tidak menggunakan enkripsi.)
- `SessionToken`— Satu set kredensial sementara untuk peran IAM.
- `GroupFederation` - Setel ke `true` jika Anda mengonfigurasi identitas federasi untuk kluster yang disediakan. Jangan setel parameter ini jika Anda menggunakan Amazon Redshift Serverless.
- `LogLevel`— Nilai tingkat log integer. Ini opsional.

4. Setel URL ke titik akhir JDBC yang ditemukan di konsol Amazon Redshift atau Amazon Redshift Tanpa Server. Ganti skema URL Anda dengan `jdbc:redshift:iam:` dan gunakan pemformatan ini:

- Format untuk kluster yang disediakan Amazon Redshift: `jdbc:redshift:iam://<cluster_id>.<unique_suffix>.<region>.redshift.amazonaws.com:<port>/<database_name>`

Contoh: `jdbc:redshift:iam://test1.12345abcdefg.us-east-1.redshift.amazonaws.com:5439/dev`

- Format untuk Amazon Redshift Tanpa Server: `jdbc:redshift:iam://<workgroup-name>.<account-number>.<aws-region>.redshift-serverless.amazonaws.com:5439:<port>/<database_name>`

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

Setelah Anda terhubung ke database untuk pertama kalinya, menggunakan identitas IAM, Amazon Redshift secara otomatis membuat identitas Amazon Redshift dengan nama yang sama, diawali IAM: dengan untuk pengguna IAMR: atau untuk peran IAM. Langkah-langkah yang tersisa dalam topik ini menunjukkan contoh untuk pengguna.

Jika pengguna Redshift tidak dibuat secara otomatis, Anda dapat membuatnya dengan menjalankan `CREATE USER` pernyataan, menggunakan akun admin, yang menentukan nama pengguna dalam format. `IAM:<user name>`

5. Sebagai administrator kluster Amazon Redshift Anda, berikan pengguna Redshift izin yang diperlukan untuk mengakses skema eksternal.

```
GRANT ALL ON SCHEMA my_schema to "IAM:my_user";
```

Untuk memberikan kemampuan kepada pengguna Redshift Anda untuk membuat tabel dalam skema eksternal, mereka harus menjadi pemilik skema. Contoh:

```
ALTER SCHEMA my_schema owner to "IAM:my_user";
```

6. Untuk memverifikasi konfigurasi, jalankan kueri sebagai pengguna, menggunakan klien SQL, setelah izin diberikan. Sampel kueri ini mengambil data dari tabel eksternal.

```
SELECT * FROM my_schema.my_table;
```

Memulai dengan propagasi identitas dan otorisasi ke Redshift Spectrum

Untuk meneruskan identitas federasi ke tabel eksternal kueri, Anda menetapkan `SESSION` sebagai nilai untuk parameter `IAM_ROLE` kueri. `CREATE EXTERNAL SCHEMA` Langkah-langkah berikut menunjukkan cara mengatur dan memanfaatkan `SESSION` untuk mengotorisasi kueri pada skema eksternal.

1. Buat tabel lokal dan tabel eksternal. Tabel eksternal dikatalogkan dengan AWS Glue pekerjaan untuk ini.
2. Connect ke Amazon Redshift dengan identitas IAM Anda. Seperti disebutkan di bagian sebelumnya, ketika identitas terhubung ke Amazon Redshift, pengguna database Redshift dibuat. Pengguna dibuat jika sebelumnya tidak ada. Jika pengguna baru, administrator harus memberi mereka izin untuk melakukan tugas di Amazon Redshift, seperti menanyakan dan membuat tabel.
3. Connect ke Redshift dengan akun admin Anda. Jalankan perintah untuk membuat skema eksternal, menggunakan `SESSION` nilai.

```
create external schema spectrum_schema from data catalog
database '<my_external_database>'
region '<my_region>'
iam_role 'SESSION'
catalog_id '<my_catalog_id>;'
```

Perhatikan bahwa `catalog_id` diatur dalam kasus ini. Ini adalah pengaturan baru yang ditambahkan dengan fitur, karena `SESSION` menggantikan peran tertentu.

Dalam contoh ini, nilai dalam kueri meniru bagaimana nilai nyata muncul.

```
create external schema spectrum_schema from data catalog
database 'spectrum_db'
region 'us-east-1'
iam_role 'SESSION'
catalog_id '123456789012'
```

`catalog_id` Nilai dalam hal ini adalah ID AWS akun Anda.

4. Jalankan kueri untuk mengakses data eksternal Anda, menggunakan identitas IAM yang terhubung dengan Anda pada langkah 2. Contoh:


```
select * from spectrum_schema.table1;
```

Dalam hal ini, `table1` dapat berupa, misalnya, data berformat JSON dalam file, di bucket Amazon S3.

5. Jika Anda sudah memiliki skema eksternal yang menggunakan peran IAM terlampir cluster, menunjuk ke database atau skema eksternal Anda, Anda dapat mengganti skema yang ada dan menggunakan identitas federasi seperti yang dirinci dalam langkah-langkah ini, atau membuat yang baru.

`SESSION` menunjukkan bahwa kredensial identitas federasi digunakan untuk menanyakan skema eksternal. Saat Anda menggunakan parameter `SESSION` kueri, pastikan Anda mengatur `catalog_id`. Ini diperlukan karena menunjuk ke katalog data yang digunakan untuk skema. Sebelumnya, `catalog_id` diambil dari nilai yang ditetapkan untuk `iam_role`. Saat Anda mengatur propagasi identitas dan otorisasi dengan cara ini, misalnya, ke Redshift Spectrum, dengan menggunakan kredensial federasi untuk menanyakan skema eksternal, otorisasi melalui peran IAM tidak diperlukan.

Catatan penggunaan

Kesalahan koneksi umum adalah sebagai berikut: Kesalahan IAM mengambil kredensial sementara: Tidak dapat membatalkan respons pengecualian dengan unmarshallers yang disediakan. Kesalahan ini adalah hasil dari memiliki driver JDBC lama. Versi driver minimum yang diperlukan untuk identitas federasi adalah 2.1.0.9. Anda bisa mendapatkan driver JDBC dari [Unduh driver Amazon Redshift JDBC](#), versi 2.x.

Sumber daya tambahan

Tautan ini memberikan informasi tambahan untuk mengelola akses ke data eksternal.

- Anda masih dapat mengakses data Redshift Spectrum menggunakan peran IAM. Untuk informasi selengkapnya, lihat [Mengotorisasi Amazon Redshift untuk AWS mengakses layanan atas nama Anda](#).
- Saat Anda mengelola akses ke tabel eksternal AWS Lake Formation, Anda dapat menanyakannya menggunakan Redshift Spectrum dengan identitas IAM federasi. Anda tidak lagi harus mengelola peran IAM terlampir cluster untuk Redshift Spectrum untuk menanyakan data yang terdaftar. AWS Lake Formation Untuk informasi selengkapnya, lihat [Menggunakan AWS Lake Formation Amazon Redshift Spectrum](#).

Mengelola kata sandi admin Amazon Redshift menggunakan AWS Secrets Manager

Amazon Redshift dapat diintegrasikan dengan AWS Secrets Manager untuk menghasilkan dan mengelola kredensi admin Anda di dalam rahasia terenkripsi. Dengan AWS Secrets Manager, Anda dapat mengganti kata sandi admin Anda dengan panggilan API untuk mengambil rahasia secara terprogram saat diperlukan. Menggunakan rahasia alih-alih kredensial hard-code mengurangi risiko kredensial tersebut terpapar atau dikompromikan. Untuk informasi selengkapnya AWS Secrets Manager, lihat [Panduan AWS Secrets Manager Pengguna](#).

Anda dapat menentukan bahwa Amazon Redshift mengelola kata sandi admin Anda menggunakan AWS Secrets Manager saat Anda melakukan salah satu operasi berikut:

- Buat klaster yang disediakan atau namespace tanpa server
- Mengedit, memperbarui, atau memodifikasi kredensi admin dari klaster yang disediakan atau namespace tanpa server
- Mengembalikan cluster atau namespace tanpa server dari snapshot

Saat Anda menentukan bahwa Amazon Redshift mengelola kata sandi admin di AWS Secrets Manager, Amazon Redshift membuat kata sandi dan menyimpannya di Secrets Manager. Anda dapat mengakses rahasia secara langsung AWS Secrets Manager untuk mengambil kredensi untuk pengguna admin. Secara opsional, Anda dapat menentukan kunci yang dikelola pelanggan untuk mengenkripsi rahasia jika Anda perlu mengakses rahasia dari akun lain AWS. Anda juga dapat menggunakan kunci KMS yang AWS Secrets Manager menyediakan.

Amazon Redshift mengelola pengaturan untuk rahasia dan memutar rahasia setiap 30 hari secara default. Anda dapat memutar rahasia secara manual kapan saja. Jika Anda menghapus klaster yang disediakan atau namespace tanpa server yang mengelola rahasia AWS Secrets Manager, rahasia dan metadata terkait juga akan dihapus.

Untuk menyambung ke klaster atau namespace tanpa server dengan kredensial yang dikelola rahasia, Anda dapat mengambil rahasia dari menggunakan AWS Secrets Manager konsol Secrets Manager atau panggilan Secrets Manager API. `GetSecretValue` Untuk informasi selengkapnya, lihat [Mengambil rahasia dari AWS Secrets Manager](#) dan [Connect ke database SQL dengan kredensi dalam AWS Secrets Manager rahasia di Panduan Pengguna](#).AWS Secrets Manager

Izin diperlukan untuk integrasi AWS Secrets Manager

Pengguna harus memiliki izin yang diperlukan untuk melakukan operasi yang terkait dengan AWS Secrets Manager integrasi. Buat kebijakan IAM yang memberikan izin untuk melakukan operasi API tertentu pada sumber daya tertentu yang mereka butuhkan. Kemudian lampirkan kebijakan tersebut ke set izin IAM atau peran yang memerlukan izin tersebut. Untuk informasi selengkapnya, lihat [Manajemen identitas dan akses di Amazon Redshift](#).

Pengguna yang menentukan bahwa Amazon Redshift mengelola kata sandi AWS Secrets Manager admin harus memiliki izin untuk melakukan operasi berikut:

- `secretsmanager:CreateSecret`
- `secretsmanager:RotateSecret`
- `secretsmanager:DescribeSecret`
- `secretsmanager:UpdateSecret`
- `secretsmanager>DeleteSecret`
- `secretsmanager:GetRandomPassword`
- `secretsmanager:TagResource`

Jika pengguna ingin meneruskan kunci KMS dalam `MasterPasswordSecretKmsKeyId` parameter untuk kluster yang disediakan, atau `AdminPasswordSecretKmsKeyId` parameter untuk ruang nama tanpa server, mereka memerlukan izin berikut selain izin yang tercantum di atas.

- `kms:Decrypt`
- `kms:GenerateDataKey`
- `kms>CreateGrant`
- `kms:RetireGrant`

Rotasi rahasia kata sandi admin

Secara default, Amazon Redshift secara otomatis memutar rahasia Anda setiap 30 hari untuk memastikan kredensial Anda tidak tetap sama untuk waktu yang lama. Saat Amazon Redshift memutar rahasia kata sandi admin, AWS Secrets Manager memperbarui rahasia yang ada untuk memuat kata sandi admin baru. Amazon Redshift mengubah kata sandi admin untuk cluster agar sesuai dengan kata sandi dalam rahasia yang diperbarui.

Anda dapat memutar rahasia segera alih-alih menunggu rotasi terjadwal dengan menggunakan AWS Secrets Manager. Untuk informasi selengkapnya tentang memutar rahasia, lihat [Memutar AWS Secrets Manager rahasia](#) di Panduan AWS Secrets Manager Pengguna.

Pertimbangan menggunakan Amazon AWS Secrets Manager Redshift

Saat menggunakan AWS Secrets Manager untuk mengelola kredensi admin klaster atau namespace tanpa server yang disediakan, pertimbangkan hal berikut:

- Saat Anda menjeda klaster yang kredensialnya dikelola oleh admin AWS Secrets Manager, rahasia klaster Anda tidak akan dihapus dan Anda akan terus ditagih untuk rahasia tersebut. Rahasia hanya dihapus ketika Anda menghapus cluster.
- Jika klaster Anda dijeda saat Amazon Redshift mencoba memutar rahasia terlampirnya, rotasi akan gagal. Dalam hal ini, Amazon Redshift menghentikan rotasi otomatis dan tidak akan mencoba memutarnya lagi, bahkan setelah Anda melanjutkan cluster. Anda harus memulai ulang jadwal rotasi otomatis menggunakan panggilan `secretsmanager:RotateSecret` API untuk terus memutar rahasia Anda AWS Secrets Manager secara otomatis.
- Jika namespace tanpa server Anda tidak memiliki grup kerja yang terkait saat Amazon Redshift mencoba memutar rahasia terlampirnya, rotasi akan gagal dan tidak akan mencoba memutarnya lagi, bahkan setelah Anda melampirkan grup kerja. Anda harus memulai ulang jadwal rotasi otomatis menggunakan panggilan `secretsmanager:RotateSecret` API untuk terus memutar rahasia Anda AWS Secrets Manager secara otomatis.

Mengambil Nama Sumber Daya Amazon (ARN) dari rahasia di Amazon Redshift

Anda dapat melihat Nama Sumber Daya Amazon (ARN) untuk mengetahui rahasia apa pun yang dikelola dengan AWS Secrets Manager menggunakan konsol Amazon Redshift. Setelah Anda memiliki ARN rahasia, Anda dapat melihat detail tentang rahasia Anda dan data terenkripsi dalam rahasia Anda menggunakan AWS Secrets Manager. Untuk informasi selengkapnya tentang mengambil rahasia menggunakan ARN, lihat [Mengambil](#) rahasia di AWS Secrets Manager Panduan Pengguna.

Melihat detail tentang rahasia untuk cluster yang disediakan Amazon Redshift

Lihat Nama Sumber Daya Amazon (ARN) untuk rahasia klaster Anda menggunakan konsol Amazon Redshift dengan prosedur berikut:

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift.
2. Di panel ikhtisar Cluster, pilih cluster yang rahasianya ingin Anda lihat.
3. Pilih tab Properti.
4. Lihat ARN rahasia di bawah kredensi Admin ARN. ARN ini adalah pengidentifikasi rahasia, yang dapat Anda gunakan AWS Secrets Manager untuk melihat detail rahasia.

Melihat detail tentang rahasia untuk namespace Amazon Redshift Tanpa Server

Lihat Amazon Resource Name (ARN) untuk rahasia namespace tanpa server Anda menggunakan konsol Amazon Redshift dengan prosedur berikut:

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift.
2. Dari dasbor klaster yang disediakan, pilih Buka Tanpa Server di kanan atas halaman.
3. Dari dasbor Tanpa Server, gulir ke panel Namespace/ Workgroups dan pilih namespace yang rahasianya ingin Anda lihat.
4. Di panel Informasi umum, lihat ARN rahasia di bawah kredensi Admin ARN. ARN ini adalah pengidentifikasi rahasia, yang dapat Anda gunakan AWS Secrets Manager untuk melihat detail rahasia.

Membuat rahasia untuk kredensi koneksi database

Anda dapat membuat rahasia Secrets Manager untuk menyimpan kredensial yang digunakan untuk menyambung ke klaster yang disediakan Amazon Redshift atau namespace dan workgroup Redshift Tanpa Server. Anda juga dapat menggunakan rahasia ini saat menjadwalkan kueri di editor kueri Amazon Redshift v2.

Untuk membuat rahasia untuk database di klaster yang disediakan Amazon Redshift menggunakan konsol Secrets Manager

1. Buka konsol Secrets Manager (<https://console.aws.amazon.com/secretsmanager/>).
2. Arahkan ke daftar Rahasia dan pilih Simpan rahasia baru.
3. Pilih Kredensial untuk gudang data Amazon Redshift. Masukkan informasi Anda dalam langkah-langkah untuk membuat rahasia sebagai berikut:
 - Di Kredensial untuk Nama Pengguna, masukkan nama pengguna administratif gudang data.
 - Di Kredensial untuk Kata Sandi, masukkan kata sandi untuk nama Pengguna.
 - Untuk kunci Enkripsi, pilih kunci enkripsi Anda.

- Untuk gudang Data, pilih klaster yang disediakan Amazon Redshift yang berisi data Anda.
 - Untuk nama Rahasia, masukkan nama untuk rahasianya.
 - Untuk Deskripsi, masukkan deskripsi rahasia.
 - Untuk Tag, masukkan kunci Tag dengan kata **Redshift**. Kunci tag ini diperlukan untuk mencantumkan rahasia saat Anda mencoba menyambung ke gudang data menggunakan editor kueri Amazon Redshift v2. Rahasia harus memiliki kunci tag yang dimulai dengan string **Redshift** untuk rahasia yang akan terdaftar di bawah AWS Secrets Manager pada konsol manajemen.
4. Lanjutkan memasukkan informasi tentang rahasia Anda melalui beberapa langkah hingga Anda Menyimpan perubahan pada langkah Tinjauan.

Nilai spesifik kredensial, engine, host, port, dan pengidentifikasi cluster Anda disimpan dalam rahasia. Juga, rahasianya ditandai dengan kunci Redshift tag.

Untuk membuat rahasia untuk database di namespace Redshift Serverless menggunakan konsol Redshift Serverless

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pilih Redshift tanpa server dan arahkan ke konfigurasi Namespace.
3. Pilih namespace untuk membuat kredensial rahasia.
4. Buka Tindakan, Edit kredensial admin.
5. Untuk kata sandi Admin, pilih Kelola kredensial admin di. AWS Secrets Manager
6. Pilih Simpan perubahan untuk menyimpan perubahan Anda.

Konfirmasikan bahwa muncul pesan bahwa kata sandi berhasil diubah. Anda juga dapat melihat rahasia di konsol Secrets Manager. Anda dapat menggunakan rahasia ini untuk menyambung ke database di workgroup di konsol Redshift Serverless dan editor kueri Amazon Redshift v2, menggunakan metode koneksi. AWS Secrets Manager Rahasiannya harus memiliki kunci tag yang dimulai dengan string "Redshift" agar rahasianya terdaftar di aplikasi web editor kueri v2. Rahasia harus memiliki kunci tag yang dimulai dengan string **Redshift** untuk rahasia yang akan terdaftar di bawah AWS Secrets Manager pada konsol manajemen.

Untuk membuat rahasia untuk database di namespace Redshift Serverless menggunakan konsol Secrets Manager

1. Buka konsol Secrets Manager (<https://console.aws.amazon.com/secretsmanager/>).
2. Arahkan ke daftar Rahasia dan pilih Simpan rahasia baru.
3. Pilih Kredensial untuk gudang data Amazon Redshift. Masukkan informasi Anda dalam langkah-langkah untuk membuat rahasia sebagai berikut:
 - Di Kredensial untuk Nama Pengguna, masukkan nama pengguna administratif gudang data.
 - Di Kredensial untuk Kata Sandi, masukkan kata sandi untuk nama Pengguna.
 - Untuk kunci Enkripsi, pilih kunci enkripsi Anda.
 - Untuk gudang Data, pilih namespace Redshift Tanpa Server yang berisi data Anda.
 - Untuk nama Rahasia, masukkan nama untuk rahasianya.
 - Untuk Deskripsi, masukkan deskripsi rahasia.
 - Untuk Tag, masukkan kunci Tag dengan kata **Redshift**. Kunci tag ini diperlukan untuk mencantumkan rahasia saat Anda mencoba menyambung ke gudang data menggunakan editor kueri Amazon Redshift v2. Rahasia harus memiliki kunci tag yang dimulai dengan string **Redshift** untuk rahasia yang akan terdaftar di bawah AWS Secrets Manager pada konsol manajemen.
4. Lanjutkan memasukkan informasi tentang rahasia Anda melalui beberapa langkah hingga Anda Menyimpan perubahan pada langkah Tinjauan.

Nilai spesifik kredensial, nama database, host, port, namespace, dan mesin Anda disimpan dalam rahasia. Juga, rahasianya ditandai dengan kunci Redshift tag.

Untuk membuat rahasia untuk database di namespace Redshift Tanpa Server menggunakan AWS CLI

Anda dapat menggunakan AWS CLI untuk membuat rahasia. Salah satu metode adalah dengan menggunakan AWS CloudShell untuk menjalankan AWS CLI perintah Secrets Manager sebagai berikut. Anda harus memiliki izin yang tepat untuk menjalankan AWS CLI perintah yang ditunjukkan dalam prosedur berikut.

1. Di AWS konsol, buka prompt AWS CloudShell perintah. Untuk informasi selengkapnya AWS CloudShell, lihat [Apa yang ada AWS CloudShell](#) di Panduan AWS CloudShell Pengguna.

2. Misalnya, untuk rahasia MyTestSecret masukkan perintah Secrets Manager untuk menyimpan rahasia yang digunakan untuk menyambung ke database atau menjadwalkan kueri Amazon Redshift query editor v2. Ganti nilai berikut dalam perintah dengan nilai untuk lingkungan Anda:

- *admin* adalah nama pengguna administrator untuk gudang data.
- *passwd* adalah kata sandi administrator.
- *dev* adalah nama database awal di gudang data.
- *region* adalah AWS Region yang berisi gudang data. Sebagai contoh, *us-east-1*.
- *123456789012* adalah Akun AWS.
- *namespace-id* adalah pengidentifikasi namespace yang mirip dengan *c3928f0e-c889-4d2b-97a5-5738324d5d3e* Anda dapat menemukan pengenalan ini di halaman detail konsol Amazon Redshift untuk namespace tanpa server.

```
aws secretsmanager create-secret \
--name MyTestSecret \
--description "My test secret created with the CLI." \
--secret-string "{\"username\":\"admin\",\"password\":\"passwd\",\"dbname\":\
\"dev\",\"engine\":\"redshift\"}" \
--tags "[{\"Key\":\"redshift-serverless:namespaceArn\",\"Value\":\
\"arn:aws:redshift-serverless:region:123456789012:namespace/namespace-id\"}]"
```

Pencatatan dan pemantauan di Amazon Redshift

Pemantauan adalah bagian penting dalam menjaga keandalan, ketersediaan, dan kinerja Amazon Redshift dan solusi Anda AWS . Anda dapat mengumpulkan data pemantauan dari semua bagian AWS solusi Anda sehingga Anda dapat lebih mudah men-debug kegagalan multi-titik jika terjadi. AWS menyediakan beberapa alat untuk memantau sumber daya Amazon Redshift Anda dan menanggapi potensi insiden:

CloudWatch Alarm Amazon

Menggunakan CloudWatch alarm Amazon, Anda menonton satu metrik selama periode waktu yang Anda tentukan. Jika metrik melebihi ambang batas tertentu, pemberitahuan akan dikirim ke topik atau AWS Auto Scaling kebijakan Amazon SNS. CloudWatch alarm tidak memanggil tindakan karena mereka berada dalam keadaan tertentu. Sebaliknya, kondisi tersebut harus

diubah dan dipertahankan selama periode tertentu. Untuk informasi selengkapnya, lihat [Membuat alarm](#). Untuk daftar metrik, lihat [Data kinerja di Amazon Redshift](#).

AWS CloudTrail Log

CloudTrail menyediakan catatan operasi API yang diambil oleh pengguna, peran IAM, atau AWS layanan di Amazon Redshift. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat untuk Amazon Redshift, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan. Lihat informasi yang lebih lengkap di [Logging dengan CloudTrail](#).

Pencatatan audit basis data

Amazon Redshift mencatat informasi tentang koneksi dan aktivitas pengguna di database Anda. Log ini membantu Anda memantau database untuk tujuan keamanan dan pemecahan masalah, proses yang disebut audit database. Log dapat disimpan di:

- Bucket Amazon S3 - Ini menyediakan akses dengan fitur keamanan data untuk pengguna yang bertanggung jawab untuk memantau aktivitas dalam database.
- Amazon CloudWatch - Anda dapat melihat data pencatatan audit menggunakan fitur bawaan CloudWatch, seperti fitur visualisasi dan tindakan pengaturan.

Note

[SYS_CONNECTION_LOG mengumpulkan data log](#) koneksi untuk Amazon Redshift Tanpa Server. Perhatikan bahwa saat Anda mengumpulkan data pencatatan audit untuk Amazon Redshift Tanpa Server, data tersebut tidak dapat dikirim ke file log, hanya ke file log. CloudWatch

Topik

- [Log Amazon Redshift](#)
- [Log audit dan Amazon CloudWatch](#)
- [Mengaktifkan pencatatan audit](#)
- [Pencatatan aman](#)

Log Amazon Redshift

Amazon Redshift mencatat informasi dalam file log berikut:

- Log koneksi - Upaya otentikasi log, koneksi, dan pemutusan.
- Log pengguna — Log informasi tentang perubahan definisi pengguna database.
- Log aktivitas pengguna — Log setiap kueri sebelum dijalankan pada database.

Koneksi dan log pengguna berguna terutama untuk tujuan keamanan. Anda dapat menggunakan log koneksi untuk memantau informasi tentang pengguna yang terhubung ke database dan informasi koneksi terkait. Informasi ini mungkin alamat IP mereka, ketika mereka membuat permintaan, jenis otentikasi apa yang mereka gunakan, dan sebagainya. Anda dapat menggunakan log pengguna untuk memantau perubahan definisi pengguna database.

Log aktivitas pengguna berguna terutama untuk tujuan pemecahan masalah. Ini melacak informasi tentang jenis kueri yang dilakukan pengguna dan sistem dalam database.

Log koneksi dan log pengguna keduanya sesuai dengan informasi yang disimpan dalam tabel sistem di database Anda. Anda dapat menggunakan tabel sistem untuk mendapatkan informasi yang sama, tetapi file log menyediakan mekanisme yang lebih sederhana untuk pengambilan dan peninjauan. File log mengandalkan izin Amazon S3 daripada izin database untuk melakukan kueri terhadap tabel. Selain itu, dengan melihat informasi dalam file log daripada menanyakan tabel sistem, Anda mengurangi dampak berinteraksi dengan database.

Note

File log tidak terkini seperti tabel log sistem yang [STL_USERLOG](#) dan [STL_CONNECTION_LOG](#). Catatan yang lebih tua dari, tetapi tidak termasuk, catatan terbaru disalin ke file log.

Note

Untuk Amazon Redshift Tanpa Server, [SYS_CONNECTION_LOG](#) mengumpulkan data log koneksi. Saat Anda mengumpulkan data pencatatan audit untuk Amazon Redshift Tanpa Server, data tersebut tidak dapat dikirim ke file log, hanya ke file log. CloudWatch

Log koneksi

Log upaya otentikasi, dan koneksi dan pemutusan. Tabel berikut menjelaskan informasi dalam log koneksi. Untuk informasi selengkapnya tentang bidang ini, lihat [STL_CONNECTION_LOG di Panduan Pengembang Database](#) Amazon Redshift. [Untuk informasi selengkapnya tentang data log koneksi yang dikumpulkan untuk Amazon Redshift Serverless, lihat SYS_CONNECTION_LOG.](#)

Nama kolom	Deskripsi
kejadian	Koneksi atau acara otentikasi.
rekor waktu	Waktu peristiwa itu terjadi.
remotehost	Nama atau alamat IP host jarak jauh.
remoteport	Nomor port untuk host jarak jauh.
pid	ID proses yang terkait dengan pernyataan.
dbname	Nama basis data.
nama pengguna	Nama pengguna.
authmethod	Metode otentikasi.
durasi	Durasi koneksi dalam mikrodetik.
sslversion	Versi Secure Sockets Layer (SSL).
sslcipher	Sandi SSL.
mtu	Unit transmisi maksimum (MTU).
sslkompresi	Jenis kompresi SSL.
sslexpansion	Jenis ekspansi SSL.
iamauthguid	ID otentikasi AWS Identity and Access Management (IAM) untuk permintaan tersebut. AWS CloudTrail Ini adalah pengenalan untuk panggilan GetClusterCredentials API untuk membuat kredensial yang digunakan untuk koneksi tertentu.

Nama kolom	Deskripsi
application_name	Nama awal atau yang diperbarui dari aplikasi untuk sesi.
os_version	Versi sistem operasi yang ada di mesin klien yang terhubung ke cluster Amazon Redshift Anda.
driver_version	Versi driver ODBC atau JDBC yang terhubung ke cluster Amazon Redshift Anda dari alat klien SQL pihak ketiga Anda.
plugin_name	Nama plugin yang digunakan untuk terhubung ke cluster Amazon Redshift Anda.
protocol_version	Versi protokol internal yang digunakan driver Amazon Redshift saat membuat koneksi dengan server.
sessionid	Pengidentifikasi unik global untuk sesi saat ini.
Kompresi	Algoritma kompresi yang digunakan untuk koneksi.

Log pengguna

Merekam detail untuk perubahan berikut pada pengguna database:

- Buat pengguna
- Jatuhkan pengguna
- Ubah pengguna (ganti nama)
- Mengubah pengguna (mengubah properti)

Nama kolom	Deskripsi
userid	ID pengguna yang terpengaruh oleh perubahan.
nama pengguna	Nama pengguna pengguna yang terpengaruh oleh perubahan.

Nama kolom	Deskripsi
nama pengguna lama	Untuk tindakan ganti nama, nama pengguna asli. Untuk tindakan lain, bidang ini kosong.
tindakan	Tindakan yang terjadi. Nilai valid: • Mengubah • Buat • Jatuhkan • Ubah Nama
digunakan createdb	Jika benar (1), menunjukkan bahwa pengguna telah membuat izin database.
menggunakan ansuper	Jika benar (1), menunjukkan bahwa pengguna adalah superuser.
menggunakan ancatupd	Jika benar (1), menunjukkan bahwa pengguna dapat memperbarui katalog sistem.
valuntil	Tanggal kedaluwarsa kata sandi.
pid	ID Proses.
xid	ID Transaksi.
rekor waktu	Waktu di UTC kueri dimulai.

Kueri tampilan sistem [SYS_USERLOG](#) untuk menemukan informasi tambahan tentang perubahan pada pengguna. Tampilan ini mencakup data log dari Amazon Redshift Tanpa Server.

Log aktivitas pengguna

Log setiap query sebelum dijalankan pada database.

Nama kolom	Deskripsi
rekor waktu	Waktu peristiwa itu terjadi.

Nama kolom	Deskripsi
db	Nama basis data.
user	Nama pengguna.
pid	ID proses yang terkait dengan pernyataan.
userid	ID pengguna.
xid	ID Transaksi.
kueri	Awalan LOG: diikuti oleh teks kueri, termasuk baris baru.

Log audit dan Amazon CloudWatch

Pencatatan audit tidak diaktifkan secara default di Amazon Redshift. Saat Anda mengaktifkan logging di klaster, Amazon Redshift mengeksport log ke Amazon CloudWatch, atau membuat dan mengunggah log ke Amazon S3, yang menangkap data dari saat pencatatan audit diaktifkan hingga saat ini. Setiap pembaruan logging merupakan kelanjutan dari log sebelumnya.

Audit logging ke CloudWatch atau ke Amazon S3 adalah proses opsional. Logging ke tabel sistem tidak opsional dan terjadi secara otomatis. Untuk informasi selengkapnya tentang pencatatan ke tabel sistem, lihat [Referensi Tabel Sistem](#) di Panduan Pengembang Database Amazon Redshift.

Log koneksi, log pengguna, dan log aktivitas pengguna diaktifkan bersama-sama dengan menggunakan Referensi API Amazon Redshift, atau AWS Command Line Interface (AWS CLI). Konsol Manajemen AWS Untuk log aktivitas pengguna, Anda juga harus mengaktifkan parameter `enable_user_activity_logging` database. Jika Anda hanya mengaktifkan fitur audit logging, tetapi bukan parameter terkait, audit database mencatat informasi log hanya untuk log koneksi dan log pengguna, tetapi tidak untuk log aktivitas pengguna. `enable_user_activity_logging` Parameter tidak diaktifkan (`false`) secara default. Anda dapat mengaturnya `true` untuk mengaktifkan log aktivitas pengguna. Untuk informasi selengkapnya, lihat [Grup parameter Amazon Redshift](#).

Saat Anda mengaktifkan login ke CloudWatch, Amazon Redshift mengeksport data log koneksi klaster, pengguna, dan aktivitas pengguna ke grup log Amazon Logs. CloudWatch Data log tidak berubah, dalam hal skema. CloudWatch dibangun untuk memantau aplikasi, dan Anda dapat menggunakannya untuk melakukan analisis real-time atau mengaturnya untuk mengambil tindakan.

Anda juga dapat menggunakan Amazon CloudWatch Logs untuk menyimpan catatan log Anda dalam penyimpanan yang tahan lama.

Menggunakan CloudWatch untuk melihat log adalah alternatif yang disarankan untuk menyimpan file log di Amazon S3. Ini tidak memerlukan banyak konfigurasi, dan mungkin sesuai dengan persyaratan pemantauan Anda, terutama jika Anda sudah menggunakannya untuk memantau layanan dan aplikasi lain.

Grup log dan peristiwa log di Amazon CloudWatch

Setelah memilih log Amazon Redshift mana yang akan diekspor, Anda dapat memantau peristiwa log di Amazon CloudWatch Logs. Grup log baru secara otomatis dibuat untuk Amazon Redshift Tanpa Server, di bawah awalan berikut, yang `log_type` mewakili jenis log.

```
/aws/redshift/cluster/<cluster_name>/<log_type>
```

Misalnya, jika Anda memilih untuk mengekspor log koneksi, data log disimpan dalam grup log berikut.

```
/aws/redshift/cluster/cluster1/connectionlog
```

Peristiwa log diekspor ke grup log menggunakan aliran log. Untuk mencari informasi dalam peristiwa log untuk titik akhir tanpa server Anda, gunakan konsol Amazon CloudWatch Logs, API, atau Amazon CloudWatch Logs. AWS CLI Untuk informasi tentang mencari dan memfilter data log, lihat [Membuat metrik dari peristiwa log menggunakan filter](#).

Di CloudWatch, Anda dapat mencari data log Anda dengan sintaks kueri yang menyediakan perincian dan fleksibilitas. Untuk informasi selengkapnya, lihat [sintaks kueri Wawasan CloudWatch Log](#).

Migrasi ke pencatatan CloudWatch audit Amazon

Dalam kasus apa pun di mana Anda mengirim log ke Amazon S3 dan Anda mengubah konfigurasi, misalnya untuk mengirim log ke CloudWatch, log yang tetap di Amazon S3 tidak terpengaruh. Anda masih dapat menanyakan data log di ember Amazon S3 di mana ia berada.

Log file di Amazon S3

Jumlah dan ukuran file log Amazon Redshift di Amazon S3 sangat bergantung pada aktivitas di cluster Anda. Jika Anda memiliki cluster aktif yang menghasilkan sejumlah besar log, Amazon

Redshift mungkin menghasilkan file log lebih sering. Anda mungkin memiliki serangkaian file log untuk jenis aktivitas yang sama, seperti memiliki beberapa log koneksi dalam jam yang sama.

Saat Amazon Redshift menggunakan Amazon S3 untuk menyimpan log, Anda dikenakan biaya untuk penyimpanan yang Anda gunakan di Amazon S3. Sebelum Anda mengonfigurasi logging ke Amazon S3, rencanakan berapa lama Anda perlu menyimpan file log. Sebagai bagian dari ini, tentukan kapan file log dapat dihapus atau diarsipkan, berdasarkan kebutuhan audit Anda. Paket yang Anda buat sangat bergantung pada jenis data yang Anda simpan, seperti data yang tunduk pada kepatuhan atau persyaratan peraturan. Untuk informasi selengkapnya tentang harga Amazon S3, buka [Harga Amazon Simple Storage Service \(S3\)](#).

Batasan saat Anda mengaktifkan logging ke Amazon S3

Audit logging memiliki kendala sebagai berikut:

- Anda hanya dapat menggunakan enkripsi kunci terkelola Amazon S3 (SSE-S3) (AES-256).
- Bucket Amazon S3 harus mematikan fitur Kunci Objek S3.

Izin bucket untuk pencatatan audit Amazon Redshift

Saat Anda mengaktifkan logging ke Amazon S3, Amazon Redshift mengumpulkan informasi pencatatan dan mengunggahnya ke file log yang disimpan di Amazon S3. Anda dapat menggunakan ember yang sudah ada atau ember baru. Amazon Redshift memerlukan izin IAM berikut ke bucket:

- `s3:GetBucketACL` Layanan ini memerlukan izin baca ke bucket Amazon S3 sehingga dapat mengidentifikasi pemilik bucket.
- `s3:PutObject` Layanan ini memerlukan izin objek put untuk mengunggah log. Selain itu, peran pengguna atau IAM yang mengaktifkan logging harus memiliki `s3:PutObject` izin ke bucket Amazon S3. Setiap kali log diunggah, layanan menentukan apakah pemilik bucket saat ini cocok dengan pemilik bucket pada saat logging diaktifkan. Jika pemilik ini tidak cocok, Anda menerima kesalahan.

Jika, saat mengaktifkan pencatatan audit, Anda memilih opsi untuk membuat bucket baru, izin yang benar akan diterapkan padanya. Namun, jika Anda membuat bucket sendiri di Amazon S3, atau menggunakan bucket yang sudah ada, pastikan untuk menambahkan kebijakan bucket yang menyertakan nama bucket. Log dikirimkan menggunakan kredensi prinsipal layanan. Untuk sebagian besar Wilayah AWS, Anda menambahkan nama utama layanan Redshift,.

redshift.amazonaws.com

Kebijakan bucket menggunakan format berikut. *ServiceName* dan *BucketName* merupakan placeholder untuk nilai-nilai Anda sendiri. Tentukan juga tindakan dan sumber daya terkait dalam kebijakan bucket.

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/*"
      ]
    }
  ]
}
```

Contoh berikut adalah kebijakan ember untuk Wilayah AS Timur (Virginia N.) dan ember bernama `AuditLogs`.

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/*"
    ]
}
]
}

```

Wilayah yang tidak diaktifkan secara default, juga dikenal sebagai Wilayah “keikutsertaan”, memerlukan nama utama layanan khusus Wilayah. Untuk ini, nama prinsipal layanan mencakup wilayah, dalam format `redshift.region.amazonaws.com`. Misalnya, `redshift.ap-east-1.amazonaws.com` untuk Wilayah Asia Pasifik (Hong Kong). Untuk daftar Wilayah yang tidak diaktifkan secara default, lihat [Mengelola Wilayah AWS](#) di Referensi Umum AWS.

Note

Nama utama layanan khusus Wilayah sesuai dengan Wilayah tempat cluster berada.

Praktik terbaik untuk file log

Saat Redshift mengunggah file log ke Amazon S3, file besar dapat diunggah menjadi beberapa bagian. Jika unggahan multibagian tidak berhasil, sebagian file dapat tetap berada di bucket Amazon S3. Hal ini dapat mengakibatkan biaya penyimpanan tambahan, jadi penting untuk memahami apa yang terjadi ketika unggahan multipart gagal. Untuk penjelasan rinci tentang unggahan multibagian untuk log audit, lihat [Mengunggah dan menyalin objek menggunakan unggahan multibagian dan membatalkan unggahan multibagian](#).

Untuk informasi selengkapnya tentang membuat bucket S3 dan menambahkan kebijakan bucket, lihat [Membuat bucket tujuan umum](#) dan [kebijakan Bucket untuk Amazon S3](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Struktur bucket untuk pencatatan audit Amazon Redshift

Secara default, Amazon Redshift mengatur file log di bucket Amazon S3 dengan menggunakan bucket dan struktur objek berikut:

`AWSLogs/AccountID/ServiceName/Region/Year/Month/Day/AccountID_ServiceName_Region`

Contohnya adalah: `AWSLogs/123456789012/redshift/us-east-1/2013/10/29/123456789012_redshift_us-east-1_mycluster_userlog_2013-10-29T18:01.gz`

Jika Anda memberikan awalan kunci Amazon S3, letakkan awalan di awal kunci.

Misalnya, jika Anda menentukan awalan myprefix: `myprefix/AWSLogs/123456789012/redshift/us-east-1/2013/10/29/123456789012_redshift_us-east-1_mycluster_userlog_2013-10-29T18:01.gz`

Awalan key Amazon S3 tidak dapat melebihi 512 karakter. Itu tidak dapat berisi spasi (), tanda kutip ganda ("), tanda kutip tunggal ('), garis miring terbalik (\). Ada juga sejumlah karakter khusus dan karakter kontrol yang tidak diizinkan. Kode heksadesimal untuk karakter ini adalah sebagai berikut:

- x00 ke x20
- x22
- x27
- x5c
- x7f atau lebih besar

Audit logging di pertimbangan Amazon S3

Pencatatan audit Amazon Redshift dapat terganggu karena alasan berikut:

- Amazon Redshift tidak memiliki izin untuk mengunggah log ke bucket Amazon S3. Verifikasi bahwa bucket dikonfigurasi dengan kebijakan IAM yang benar. Untuk informasi selengkapnya, lihat [izin bucket untuk pencatatan audit Amazon Redshift](#).
- Pemilik ember berubah. Saat Amazon Redshift mengunggah log, Amazon Redshift memverifikasi bahwa pemilik bucket sama seperti saat logging diaktifkan. Jika pemilik bucket telah berubah, Amazon Redshift tidak dapat mengunggah log hingga Anda mengonfigurasi bucket lain yang akan digunakan untuk pencatatan audit.

- Ember tidak dapat ditemukan. Jika bucket dihapus di Amazon S3, Amazon Redshift tidak dapat mengunggah log. Anda harus membuat ulang bucket atau mengonfigurasi Amazon Redshift untuk mengunggah log ke bucket lain.

Panggilan API dengan AWS CloudTrail

Amazon Redshift terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di Amazon Redshift. CloudTrail menangkap semua panggilan API untuk Amazon Redshift sebagai peristiwa. Untuk informasi selengkapnya tentang integrasi Amazon Redshift dengan AWS CloudTrail, lihat [Logging with CloudTrail](#).

Anda dapat menggunakan CloudTrail secara independen dari atau selain pencatatan audit database Amazon Redshift.

Untuk mempelajari selengkapnya CloudTrail, lihat [Panduan AWS CloudTrail Pengguna](#).

Mengaktifkan pencatatan audit

Konfigurasi Amazon Redshift untuk mengekspor data log audit. Log dapat diekspor ke CloudWatch, atau sebagai file ke ember Amazon S3.

Mengaktifkan pencatatan audit menggunakan konsol

Langkah-langkah konsol

Untuk mengaktifkan pencatatan audit untuk kluster

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster, lalu pilih cluster yang ingin Anda perbarui.
3. Pilih tab Properti. Pada panel konfigurasi Database, pilih Edit, lalu Edit pencatatan audit.
4. Pada halaman Edit pencatatan audit, pilih Aktifkan dan pilih bucket S3 atau CloudWatch. Kami merekomendasikan penggunaan CloudWatch karena administrasi mudah dan memiliki fitur yang berguna untuk visualisasi data.
5. Pilih log mana yang akan diekspor.
6. Untuk menyimpan pilihan Anda, pilih Simpan perubahan.

Pencatatan aman

Saat Amazon Redshift mencatat kueri yang mereferensikan satu atau beberapa AWS Glue Data Catalog tampilan, Amazon Redshift secara otomatis menutupi bidang dalam tabel sistem tertentu dan kolom tampilan saat mencatat metadata tentang kueri tersebut.

Masking log aman berlaku untuk semua tabel sistem dan entri tampilan yang dihasilkan Amazon Redshift saat menjalankan kueri yang sesuai dengan kondisi masking. Tabel berikut mencantumkan tampilan sistem dan kolom yang memiliki logging aman diterapkan, menutupi teks dengan ********* dan angka dengan -1. Jumlah tanda bintang yang digunakan untuk menutupi teks cocok dengan jumlah karakter dalam teks asli, hingga 6 karakter. String yang lebih panjang dari 6 karakter masih muncul sebagai 6 tanda bintang.

Tabel sistem	Kolom sensitif
<u>SYS_EXTENSIONAL_QUERY_DETAIL</u>	Kolom: source_type, total_partitions, qualified_partitions, scanned_files, returned_rows, returned_bytes, file_format, file_location, external_query_text, warning_message.
<u>SYS_EXTENSIONAL_QUERY_ERROR</u>	Kolom: file_location, rowid, column_name, original_value, modified_value, trigger, action, action_value, error_code.
<u>SYS_QUERY_DETAIL</u>	Kolom: step_id, step_name, table_id, table_name, input_bytes, input_rows, output_byte, output_rows, blocks_read, blocks_write, local_read_io, remote_read_io, spilled_block_local_disk, spilled_block_remote_disk, step_attribute.
<u>SYS_QUERY_HISTORY</u>	Kolom: returned_rows, returned_bytes.
<u>STL_AGGR</u>	Kolom: baris, byte, tbl, jenis.
<u>STL_BCAST</u>	Kolom: baris, byte, paket.
<u>STL_DDLTEXT</u>	Kolom: label, teks.
<u>STL_DELETE</u>	Kolom: baris, tbl.
<u>STL_DIST</u>	Kolom: baris, byte, paket.

Tabel sistem	Kolom sensitif
<u>STL_ERROR</u>	Kolom: file, linenum, konteks, kesalahan.
<u>STL_JELASKAN</u>	Kolom: plannode, info.
<u>STL_FILE_SCAN</u>	Kolom: nama, baris, byte.
<u>STL_HASH</u>	Kolom: baris, byte, tbl, est_rows.
<u>STL_HASHJOIN</u>	Kolom: baris, tbl, num_parts, join_type.
<u>STL_INSERT</u>	Kolom: baris, tbl.
<u>STL_LIMIT</u>	Kolom: baris.
<u>STL_MERGE</u>	Kolom: baris.
<u>STL_MERGEJOIN</u>	Kolom: baris, tbl.
<u>STL_NESTLOOP</u>	Kolom: baris, tbl.
<u>STL_PARSE</u>	Kolom: baris.
<u>STL_PLAN_INFO</u>	Kolom: startupcost, totalcost, baris, byte.
<u>STL_PROJECT</u>	Kolom: baris, tbl.
<u>STL_QUERY</u>	Kolom: querytxt.
<u>STL_QUERY_METRICS</u>	Kolom: max_rows, baris, max_blocks_read, max_blocks_to_disk, blocks_to_disk, max_query_scan_size, query_scan_size.
<u>STL_QUERYTEXT</u>	Kolom: teks.
<u>STL_KEMBALI</u>	Kolom: baris, byte.
<u>STL_S3KLIEN</u>	Kolom: ember, kunci, transfer_size, data_size.
<u>STL_S3CLIENT_ERROR</u>	Kolom: ember, kunci, kesalahan, transfer_size.

Tabel sistem	Kolom sensitif
<u>STL_SIMPAN</u>	Kolom: baris, byte, tbl.
<u>STL_SCAN</u>	Kolom: baris, byte, pengambilan, jenis, tbl, rows_pre_filter, rows_pre_user_filter, perm_table_name, scanned_mega_value.
<u>STL_SORT</u>	Kolom: baris, byte, tbl.
<u>STL_SSHCL IENT_ERROR</u>	Kolom: ssh_username, titik akhir, perintah, kesalahan.
<u>STL_TR_CONFLICT</u>	Kolom: table_id.
<u>STL_UNDONE</u>	Kolom: table_id.
<u>STL_UNIK</u>	Kolom: baris, jenis, byte.
<u>STL_UTILITYTEXT</u>	Kolom: label, teks.
<u>STL_WINDOW</u>	Kolom: baris.
<u>STV_BLOCKLIST</u>	Kolom: col, tbl, num_values, minvalue, maxvalue.
<u>STV_EXEC_STATE</u>	Kolom: baris, byte, label.
<u>STV_DALAM PESAWAT</u>	Kolom: label, teks.
<u>STV_LOCKS</u>	Kolom: table_id.
<u>STV_QUERY _METRICS</u>	Kolom: baris, max_rows, blocks_read, max_blocks_read, max_block s_to_disk, blocks_to_disk, max_query_scan_size, query_scan_size.
<u>STV_START UP_RECOVE RY_STATE</u>	Kolom: table_id, table_name.
<u>STV_TBL_PERM</u>	Kolom: id, nama, baris, sorted_rows, temp, block_count, query_sca n_size.

Tabel sistem	Kolom sensitif
<u>STV_TBL_TRANS</u>	Kolom: id, baris, ukuran.
<u>SVCS_JELASKAN</u>	Kolom: plannode, info.
<u>SVCS_PLAN_INFO</u>	Kolom: baris, byte.
<u>SVCS_QUER Y_RINGKASAN</u>	Kolom: langkah, baris, byte, rate_row, rate_byte, label, rows_pre_filter.
<u>SVCS_S3LIST</u>	Kolom: ember, awalan, retrieved_files, max_file_size, avg_file_size.
<u>SVCS_S3LOG</u>	Kolom: pesan.
<u>SVCS_S3PA RTITION_RINGKASAN</u>	Kolom: total_partitions, qualified_partitions, min_assigned_partitions, max_assigned_partitions, avg_assigned_partitions.
<u>SVCS_S3QU ERY_RINGKASAN</u>	Kolom: external_table_name, file_format, s3_scanned_rows, s3_scanned_bytes, s3query_returned_rows, s3query_returned_bytes.
<u>SVL_QUERY_METRICS</u>	Kolom: step_label, scan_row_count, join_row_count, nested_lo op_join_row_count, return_row_count, spectrum_scan_row_count, spectrum_scan_size_mb.
<u>SVL_QUERY _METRICS_SUMMARY</u>	Kolom: step_label, scan_row_count, join_row_count, nested_lo op_join_row_count, return_row_count, spectrum_scan_row_count, spectrum_scan_size_mb.
<u>SVL_QUERY _LAPORAN</u>	Kolom: baris, byte, label, rows_pre_filter.
<u>SVL_QUERY _RINGKASAN</u>	Kolom: baris, byte, rows_pre_filter.
<u>SVL_S3LIST</u>	Kolom: ember, awalan, retrieved_files, max_file_size, avg_file_size.
<u>SVL_S3LOG</u>	Kolom: pesan.
<u>SVL_S3PARTISI</u>	Kolom: baris, byte, label, rows_pre_filter.

Tabel sistem	Kolom sensitif
SVL_S3PARTITION_RINGKASAN	Kolom: total_partitions, qualified_partitions, min_assigned_partitions, max_assigned_partitions, avg_assigned_partitions.
SVL_S3QUERY	Kolom: external_table_name, file_format, s3_scanned_rows, s3_scanned_bytes, s3query_returned_rows, s3query_returned_bytes, file.
SVL_S3QUERY_RINGKASAN	Kolom: external_table_name, file_format, s3_scanned_rows, s3_scanned_bytes, s3query_returned_rows, s3query_returned_bytes.
SVL_S3REMEMCOBA	Kolom: file_size, lokasi, pesan.
SVL_SPECTRUM_SCAN_ERROR	Kolom: lokasi, rowid, colname, original_value, modified_value, trigger, action, action_value, error_code.
SVL_STATEMENTTEXT	Kolom: jenis, teks.
SVL_STORED_PROC_PANGGILAN	Kolom: querytxt.
SVL_STORED_PROC_MESSAGES	Kolom: pesan, linenum, querytext.
SVL_UDF_LOG	Kolom: pesan, funcname.
SVV_DISKUSAGE	Kolom: nama, col, tbl, blocknum, num_values, minvalue, maxvalue.
SVV_QUERY_STATE	Kolom: baris, byte, label.
SVV_TABLE_INFO	Kolom: table_id, tabel.
SVV_TRANSAKSI	Kolom: relasi.

Untuk informasi selengkapnya tentang tabel dan tampilan sistem, lihat [Referensi tabel dan tampilan sistem](#) di Panduan Pengembang Database Amazon Redshift. Untuk informasi tentang kemampuan Amazon Redshift untuk secara dinamis menutupi hasil kueri, lihat Penyembunyian [data dinamis](#) di

Panduan Pengembang Database Amazon Redshift. Untuk informasi tentang cara membuat tampilan dalam AWS Glue Data Catalog menggunakan Amazon Redshift, lihat [AWS Glue Data Catalog tampilan](#) di Panduan Pengembang Database Amazon Redshift.

Logging dengan CloudTrail

Amazon Redshift, berbagi data, Amazon Redshift Serverless, Amazon Redshift Data API, dan editor kueri v2 semuanya terintegrasi dengannya. AWS CloudTrail CloudTrail adalah layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di Amazon Redshift. CloudTrail menangkap semua panggilan API untuk Amazon Redshift sebagai peristiwa. Panggilan yang diambil termasuk panggilan dari konsol Redshift dan panggilan kode ke operasi Redshift.

Jika Anda membuat CloudTrail jejak, Anda dapat mengirimkan CloudTrail acara secara berkelanjutan ke bucket Amazon S3, termasuk acara untuk Redshift. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol dalam Riwayat acara. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan hal-hal tertentu. Ini termasuk permintaan yang dibuat untuk Redshift, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Anda dapat menggunakan CloudTrail secara independen dari atau selain pencatatan audit database Amazon Redshift.

Untuk mempelajari selengkapnya CloudTrail, lihat [Panduan AWS CloudTrail Pengguna](#).

Informasi di CloudTrail

CloudTrail diaktifkan di AWS akun Anda saat Anda membuat akun. Ketika aktivitas terjadi, aktivitas tersebut dicatat dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh acara terbaru di AWS akun Anda. Untuk informasi selengkapnya, lihat [Melihat CloudTrail Acara dengan Riwayat Acara](#) di Panduan AWS CloudTrail Pengguna.

Untuk catatan peristiwa yang sedang berlangsung di AWS akun Anda, termasuk acara untuk Redshift, buat jejak. CloudTrail menggunakan jejak untuk mengirimkan file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol tersebut, jejak diterapkan ke semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah di AWS partisi dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lain untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam

CloudTrail log. Untuk informasi selengkapnya, pelajari topik berikut di Panduan Pengguna AWS CloudTrail :

- [Gambaran Umum untuk Membuat Jejak](#)
- [CloudTrail Layanan dan Integrasi yang Didukung](#)
- [Mengkonfigurasi Notifikasi Amazon SNS untuk CloudTrail](#)
- [Menerima File CloudTrail Log dari Beberapa Wilayah](#) dan [Menerima File CloudTrail Log dari Beberapa Akun](#)

Semua tindakan Amazon Redshift, Amazon Redshift Serverless, Data API, berbagi data, dan editor kueri v2 dicatat oleh CloudTrail. Misalnya, panggilan ke `AuthorizeDataShare`, `CreateNamespaceExecuteStatement`, dan `CreateConnection` tindakan menghasilkan entri dalam file CloudTrail log.

Setiap entri peristiwa atau log berisi informasi tentang entitas yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan hal berikut:

- Baik permintaan tersebut dibuat dengan kredensial pengguna atau root.
- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna gabungan.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi selengkapnya, lihat [Elemen CloudTrail UserIdentity](#) di AWS CloudTrail Panduan Pengguna.

Entri file log

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa mewakili permintaan tunggal dari sumber manapun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, sehingga file tersebut tidak muncul dalam urutan tertentu.

Contoh Data Pergeseran Merah Amazon

Contoh berikut menunjukkan entri CloudTrail log yang menggambarkan AuthorizeDataShare operasi.

```
{
  "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 botocore/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"
}

```

Contoh Tanpa Server Amazon Redshift

Amazon Redshift Serverless terintegrasi AWS CloudTrail untuk memberikan catatan tindakan yang diambil di Amazon Redshift Tanpa Server. CloudTrail menangkap semua panggilan API untuk Amazon Redshift Serverless sebagai peristiwa. [Untuk informasi selengkapnya tentang fitur Amazon Redshift Tanpa Server, lihat ikhtisar fitur Amazon Redshift Tanpa Server.](#)

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan CreateNamespace tindakan.

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AAKEOFPINEXAMPLE:admin",
    "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",
  }

```

Contoh API Data Amazon Redshift

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan ExecuteStatement tindakan.

```

{
  "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"
}

```

}

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan ExecuteStatement tindakan yang menunjukkan clientToken digunakan untuk idempotensi.

```
{
  "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"
}
```


Contoh editor kueri Amazon Redshift v2

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan CreateConnection tindakan.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AAKEOFPINEXAMPLE:session",
    "arn": "arn:aws:sts::123456789012:assumed-role/MyRole/session",
    "accountId": "123456789012",
    "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AAKEOFPINEXAMPLE",
        "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": "AAKE0FPINEXAMPLE: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"
}

```

Akun IDs Amazon Redshift di log AWS CloudTrail

Saat Amazon Redshift memanggil AWS layanan lain untuk Anda, panggilan tersebut dicatat dengan ID akun milik Amazon Redshift. Itu tidak login dengan ID akun Anda. Misalnya, Amazon Redshift memanggil AWS Key Management Service (AWS KMS) operasi seperti `CreateGrant`, `DecryptEncrypt`, dan `RetireGrant` mengelola enkripsi di klaster Anda. Dalam hal ini, panggilan dicatat dengan AWS CloudTrail menggunakan ID akun Amazon Redshift.

Amazon Redshift menggunakan akun IDs di tabel berikut saat memanggil layanan lain AWS .

Wilayah	Wilayah	account-id
Wilayah US East (N. Virginia)	us-east-1	368064434614
Wilayah AS Timur (Ohio)	us-east-2	790247189693
Wilayah AS Barat (California Utara)	us-west-1	703715109447
Wilayah AS Barat (Oregon)	us-west-2	473191095985
Wilayah Afrika (Cape Town)	af-south-1	420376844563
Wilayah Asia Pasifik (Hong Kong)	ap-east-1	651179539253
Wilayah Asia Pasifik (Hyderabad)	ap-south-2	297058826802
Wilayah Asia Pasifik (Jakarta)	ap-southeast-3	623197973179
Wilayah Asia Pasifik (Malaysia)	ap-southeast-5	590184011157
Wilayah Asia Pasifik (Melbourne)	ap-southeast-4	945512339897
Wilayah Asia Pasifik (Mumbai)	ap-south-1	408097707231
Asia Pasifik (Selandia Baru)	ap-tenggara 6	730335362089
Wilayah Asia Pasifik (Osaka)	ap-northeast-3	398671365691
Wilayah Asia Pasifik (Seoul)	ap-northeast-2	713597048934
Wilayah Asia Pacific (Singapore)	ap-southeast-1	960118270566
Wilayah Asia Pasifik (Sydney)	ap-southeast-2	485979073181
Asia Pasifik (Taipei)	ap-timur-2	211125526206

Wilayah	Wilayah	account-id
Wilayah Asia Pasifik (Thailand)	ap-tenggara 7	767397930036
Wilayah Asia Pasifik (Tokyo)	ap-northeast-1	615915377779
Wilayah Kanada (Pusat)	ca-central-1	764870610256
Wilayah Kanada Barat (Calgary)	ca-west-1	830903446466
Wilayah Eropa (Frankfurt)	eu-central-1	434091160558
Wilayah Eropa (Irlandia)	eu-west-1	246478207311
Wilayah Eropa (London)	eu-west-2	885798887673
Wilayah Eropa (Milan)	eu-south-1	041313461515
Wilayah Eropa (Paris)	eu-west-3	694668203235
Wilayah Eropa (Spanyol)	eu-south-2	028811157404
Wilayah Eropa (Stockholm)	eu-north-1	553461782468
Wilayah Eropa (Zürich)	eu-central-2	668912161003
Wilayah Israel (Tel Aviv)	il-central-1	901883065212
Wilayah Meksiko (Tengah)	mx-pusat-1	058264411980
Wilayah Timur Tengah (Bahrain)	me-south-1	051362938876
Wilayah Timur Tengah (UEA)	me-central-1	595013617770
Wilayah Amerika Selatan (Sao Paulo)	sa-east-1	392442076723

Contoh berikut menunjukkan entri CloudTrail log untuk operasi AWS KMS Dekripsi yang dipanggil oleh Amazon Redshift.

```
{
```

```

    "eventVersion": "1.05",
    "userIdentity": {
      "type": "AssumedRole",
      "principalId": "AROAI5QPCMKLTL4VHFCYY: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": "AROAI5QPCMKLTL4VHFCYY",
          "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"
}
```

Validasi kepatuhan untuk Amazon Redshift

Auditor pihak ketiga menilai keamanan dan kepatuhan Amazon Redshift sebagai bagian dari AWS beberapa program kepatuhan. Program ini mencakup SOC, PCI, FedRAMP, HIPAA, dan lainnya.

Untuk daftar AWS layanan dalam lingkup program kepatuhan tertentu, lihat [AWSlayanan dalam lingkup oleh program kepatuhan](#). Untuk informasi umum, lihat [Program kepatuhan AWS](#).

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat [Mengunduh laporan di AWS Artifak](#).

Tanggung jawab kepatuhan Anda saat menggunakan Amazon Redshift ditentukan oleh sensitivitas data Anda, tujuan kepatuhan organisasi Anda, serta hukum dan peraturan yang berlaku. Jika penggunaan Amazon Redshift Anda tunduk pada kepatuhan terhadap standar seperti HIPAA, PCI, atau FedRAMP, menyediakan sumber daya untuk membantu: AWS

- [Panduan memulai cepat keamanan dan kepatuhan](#) yang membahas pertimbangan arsitektur dan langkah-langkah untuk menerapkan lingkungan dasar yang berfokus pada keamanan dan kepatuhan. AWS
- [Arsitektur untuk whitepaper keamanan dan kepatuhan HIPAA](#), yang menjelaskan bagaimana perusahaan dapat menggunakan AWS untuk membuat aplikasi yang sesuai dengan HIPAA.
- [AWSsumber daya kepatuhan](#), buku kerja, dan panduan yang mungkin berlaku untuk industri dan lokasi Anda.
- [AWS Config](#) AWSLayanan, dapat menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- [AWS Security Hub CSPM](#), AWS layanan, memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS yang membantu Anda memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik. Security Hub CSPM menggunakan kontrol

keamanan untuk mengevaluasi konfigurasi sumber daya dan standar keamanan untuk membantu Anda mematuhi berbagai kerangka kerja kepatuhan. Untuk informasi selengkapnya tentang menggunakan Security Hub CSPM guna mengevaluasi resource Amazon Redshift, lihat kontrol Amazon [Redshift di Panduan Pengguna](#). AWS Security Hub

Dokumen kepatuhan dan keamanan berikut mencakup Amazon Redshift dan tersedia sesuai permintaan melalui. AWS Artifact Untuk informasi selengkapnya, lihat [AWS Artifact](#).

- Katalog Kontrol Kepatuhan Komputasi Cloud (C5)
- ISO 27001:2013 Pernyataan Penerapan (SoA)
- Sertifikasi ISO 27001:2013
- ISO 27017:2015 Pernyataan Penerapan (SoA)
- Sertifikasi ISO 27017:2015
- ISO 27018:2015 Pernyataan Penerapan (SoA)
- Sertifikasi ISO 27018:2014
- Sertifikasi ISO 9001:2015
- Pengesahan Kepatuhan (AOC) PCI DSS dan Ringkasan Tanggung Jawab
- Laporan Kontrol Organisasi Layanan (SOC) 1
- Laporan Kontrol Organisasi Layanan (SOC) 2
- Laporan Kontrol Organisasi Layanan (SOC) 2 untuk Kerahasiaan

Ketahanan dalam Pergeseran Merah Amazon

Infrastruktur AWS global dibangun di sekitar AWS Wilayah dan Zona Ketersediaan (AZs). AWS Wilayah menyediakan beberapa Zona Ketersediaan yang terpisah secara fisik dan terisolasi yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan Zona Ketersediaan, Anda dapat merancang dan mengoperasikan aplikasi dan basis data yang secara otomatis melakukan failover di antara Zona Ketersediaan tanpa gangguan. Availability Zone lebih tersedia, toleran terhadap kesalahan, dan skalabel daripada infrastruktur pusat data tunggal tradisional atau beberapa infrastruktur pusat data.

Hampir semua AWS Wilayah memiliki beberapa Availability Zone dan pusat data. Anda dapat menerapkan aplikasi Anda di beberapa Zona Ketersediaan di Wilayah yang sama untuk toleransi kesalahan dan latensi rendah.

Untuk memindahkan kluster ke Availability Zone lain tanpa kehilangan data atau perubahan pada aplikasi Anda, Anda dapat mengatur relokasi untuk kluster Anda. Dengan relokasi, Anda dapat melanjutkan operasi ketika ada gangguan layanan pada cluster Anda dengan dampak minimal. Saat relokasi cluster diaktifkan, Amazon Redshift mungkin memilih untuk merelokasi cluster dalam beberapa situasi. Untuk informasi selengkapnya tentang relokasi di Amazon Redshift, lihat [Merelokasi cluster](#).

[Merelokasi cluster](#)

Dalam skenario kegagalan di mana peristiwa tak terduga terjadi di Availability Zone, Anda dapat mengatur penerapan beberapa Availability Zones (Multi-AZ) untuk memastikan bahwa gudang data Amazon Redshift Anda dapat terus beroperasi. Amazon Redshift menerapkan sumber daya komputasi yang sama di dua Availability Zone yang dapat diakses melalui satu titik akhir. Jika terjadi kegagalan seluruh Availability Zone, sumber daya komputasi yang tersisa di Availability Zone kedua akan tersedia untuk melanjutkan pemrosesan beban kerja. Untuk informasi selengkapnya tentang penerapan Multi-AZ, lihat [Deployment multi-AZ](#).

Untuk informasi selengkapnya tentang AWS Wilayah dan Availability Zone, lihat [infrastruktur AWS global](#). Untuk informasi selengkapnya tentang penggunaan Amazon Redshift untuk pemulihan bencana, lihat [Menerapkan pemulihan bencana dengan Amazon Redshift](#).

Keamanan infrastruktur di Amazon Redshift

Sebagai layanan terkelola, Amazon Redshift dilindungi oleh keamanan jaringan AWS global. Untuk informasi tentang layanan AWS keamanan dan cara AWS melindungi infrastruktur, lihat [Keamanan AWS Cloud](#). Untuk mendesain AWS lingkungan Anda menggunakan praktik terbaik untuk keamanan infrastruktur, lihat [Perlindungan Infrastruktur dalam Kerangka Kerja](#) yang AWS Diarsiteksikan dengan Baik Pilar Keamanan.

Anda menggunakan panggilan API yang AWS dipublikasikan untuk mengakses Amazon Redshift melalui jaringan. Klien harus mendukung hal-hal berikut:

- Keamanan Lapisan Pengangkutan (TLS). Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Sandi cocok dengan sistem kerahasiaan maju sempurna (perfect forward secrecy, PFS) seperti DHE (Ephemeral Diffie-Hellman) atau ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

Isolasi jaringan

Virtual Private Cloud (VPC) berbasis layanan Amazon VPC adalah jaringan pribadi Anda yang terisolasi secara logis di Cloud. AWS Anda dapat menerapkan cluster Amazon Redshift atau workgroup Redshift Tanpa Server dalam VPC dengan mengambil langkah-langkah berikut:

- Buat VPC di Wilayah. AWS Untuk informasi lebih lanjut, lihat [Apa itu Amazon VPC?](#) di Panduan Pengguna Amazon VPC.
- Buat dua atau lebih subnet VPC pribadi. Untuk informasi lebih lanjut, lihat [VPCs dan subnet](#) di Panduan Pengguna Amazon VPC.
- Menerapkan cluster Amazon Redshift atau grup kerja Redshift Serverless. Untuk informasi selengkapnya, lihat [Subnet untuk sumber daya Redshift](#) atau [Kelompok kerja dan ruang nama](#).

Cluster Amazon Redshift dikunci secara default saat penyediaan. Untuk mengizinkan lalu lintas jaringan masuk dari klien Amazon Redshift, kaitkan grup keamanan VPC dengan cluster Amazon Redshift. Untuk informasi selengkapnya, lihat [Subnet untuk sumber daya Redshift](#).

Untuk mengizinkan lalu lintas hanya ke atau dari rentang alamat IP tertentu, perbarui grup keamanan dengan VPC Anda. Contohnya adalah mengizinkan lalu lintas hanya dari atau ke jaringan perusahaan Anda.

Saat mengonfigurasi daftar kontrol akses jaringan yang terkait dengan subnet yang ditandai dengan klaster Amazon Redshift Anda, pastikan rentang CIDR S3 Wilayah AWS masing-masing ditambahkan ke daftar yang diizinkan untuk aturan masuk dan keluar. Melakukannya memungkinkan Anda menjalankan operasi berbasis S3 seperti Redshift Spectrum, COPY, dan UNLOAD tanpa gangguan apa pun.

Contoh perintah berikut mem-parsing respons JSON untuk semua IPv4 alamat yang digunakan di Amazon S3 di Wilayah us-east-1.

```
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
```

Untuk petunjuk tentang cara mendapatkan rentang IP S3 untuk wilayah tertentu, lihat [rentang alamat AWS IP](#).

Amazon Redshift mendukung penerapan cluster ke dalam penyewaan khusus. VPCs Untuk informasi selengkapnya, lihat [Instans khusus](#) di Panduan Pengguna Amazon EC2.

Grup keamanan Amazon Redshift

Saat Anda menyediakan klaster Amazon Redshift, klaster tersebut dikunci secara default sehingga tidak ada yang memiliki akses ke sana. Untuk memberi pengguna lain akses masuk ke klaster Amazon Redshift, Anda mengaitkan klaster dengan grup keamanan. Jika Anda menggunakan platform EC2-VPC, Anda dapat menggunakan grup keamanan Amazon VPC yang ada atau menentukan yang baru dan kemudian mengaitkannya dengan cluster. Untuk informasi lebih lanjut tentang mengelola cluster pada platform EC2-VPC, lihat [Sumber daya Redshift dalam VPC](#)

Titik akhir VPC antarmuka

Anda dapat terhubung langsung ke layanan Amazon Redshift dan Amazon Redshift Serverless API menggunakan antarmuka VPC endpoint (AWS PrivateLink) di cloud pribadi virtual (VPC) Anda alih-alih terhubung melalui internet. Untuk informasi tentang tindakan Amazon Redshift API, lihat [Tindakan](#) di Referensi API Amazon Redshift. Untuk informasi tentang tindakan Redshift API Tanpa Server, lihat [Tindakan](#) di Referensi API Tanpa Server Amazon Redshift. Untuk informasi selengkapnya AWS PrivateLink, lihat [Titik akhir VPC Antarmuka \(AWS PrivateLink\) di Panduan Pengguna Amazon VPC](#). Perhatikan bahwa JDBC/ODBC koneksi ke cluster atau ruang kerja bukan bagian dari layanan Amazon Redshift API.

Saat Anda menggunakan titik akhir VPC antarmuka, komunikasi antara VPC dan Amazon Redshift atau Redshift Serverless dilakukan sepenuhnya di dalam jaringan, yang dapat memberikan keamanan yang lebih besar. AWS Setiap titik akhir VPC diwakili oleh satu atau lebih antarmuka jaringan elastis dengan alamat IP pribadi di subnet VPC Anda. Untuk informasi selengkapnya tentang antarmuka jaringan elastis, lihat [Antarmuka jaringan elastis](#) dalam Panduan Pengguna Amazon EC2.

Titik akhir VPC antarmuka menghubungkan VPC Anda langsung ke Amazon Redshift. Itu tidak menggunakan gateway internet, perangkat terjemahan alamat jaringan (NAT), koneksi jaringan pribadi virtual (VPN), atau Direct Connect koneksi. Instans di VPC Anda tidak memerlukan alamat IP publik untuk berkomunikasi dengan Amazon Redshift API.

Untuk menggunakan Amazon Redshift atau Redshift Serverless melalui VPC Anda, Anda memiliki dua opsi. Salah satunya adalah terhubung dari instance yang ada di dalam VPC Anda. Yang lainnya

adalah menghubungkan jaringan pribadi Anda ke VPC Anda dengan menggunakan Site-to-Site VPN opsi atau. Direct Connect Untuk informasi selengkapnya tentang Site-to-Site VPN opsi, lihat [Koneksi VPN](#) di Panduan Pengguna Amazon VPC. Untuk informasi tentang Direct Connect, lihat [Membuat hubungan](#) di Panduan Pengguna Direct Connect .

Anda dapat membuat titik akhir VPC antarmuka untuk terhubung ke Amazon Redshift menggunakan perintah `or` (). Konsol Manajemen AWS AWS Command Line Interface AWS CLI Untuk informasi selengkapnya, lihat [Membuat Titik Akhir Antarmuka](#).

Setelah Anda membuat antarmuka VPC endpoint, Anda dapat mengaktifkan nama host DNS pribadi untuk titik akhir. Ketika Anda melakukannya, titik akhir default adalah sebagai berikut:

- Amazon Redshift disediakan: `https://redshift.Region.amazonaws.com`
- Amazon Redshift Tanpa Server: `https://redshift-serverless.Region.amazonaws.com`

Jika Anda tidak mengaktifkan nama host DNS pribadi, Amazon VPC menyediakan nama endpoint DNS yang dapat Anda gunakan dalam format berikut.

- Amazon Redshift disediakan: `VPC_endpoint_ID.redshift.Region.vpce.amazonaws.com`
- Amazon Redshift Tanpa Server: `VPC_endpoint_ID.redshift-serverless.Region.vpce.amazonaws.com`

Untuk informasi selengkapnya, lihat [Antarmuka VPC endpoint \(AWS PrivateLink\)](#) dalam Panduan Pengguna Amazon VPC.

[Dukungan Amazon Redshift dan Redshift Serverless melakukan panggilan ke semua operasi Amazon Redshift API dan operasi Redshift Serverless API di dalam VPC Anda.](#)

Anda dapat melampirkan kebijakan titik akhir VPC ke titik akhir VPC untuk mengontrol akses bagi prinsipal (IAM). AWS Identity and Access Management Anda juga dapat mengaitkan grup keamanan dengan titik akhir VPC untuk mengontrol akses masuk dan keluar berdasarkan asal dan tujuan lalu lintas jaringan. Contohnya adalah berbagai alamat IP. Untuk informasi selengkapnya, silakan lihat [Mengendalikan Akses ke Layanan dengan titik akhir VPC](#) dalam Panduan Pengguna Amazon VPC.

Kebijakan titik akhir VPC untuk Amazon Redshift

Anda dapat membuat kebijakan untuk titik akhir VPC untuk Amazon Redshift untuk menentukan hal berikut:

- Kepala sekolah yang bisa atau tidak bisa melakukan tindakan
- Tindakan-tindakan yang dapat dilakukan
- Sumber daya yang dapat digunakan untuk mengambil tindakan

Untuk informasi selengkapnya, lihat [Mengendalikan akses ke layanan dengan VPC endpoint](#) di Panduan Pengguna Amazon VPC.

Berikut ini, Anda dapat menemukan contoh kebijakan titik akhir VPC.

Contoh Kebijakan Titik Akhir yang Disediakan Amazon Redshift

Berikut ini, Anda dapat menemukan contoh kebijakan titik akhir VPC untuk Amazon Redshift Provisioned.

Contoh: Kebijakan titik akhir VPC untuk menolak semua akses dari akun tertentu AWS

Kebijakan titik akhir VPC berikut menyangkal **123456789012** semua akses AWS akun ke sumber daya menggunakan titik akhir ini.

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": "*"
    },
    {
      "Action": "*",
      "Effect": "Deny",
      "Resource": "*",
      "Principal": {
        "AWS": [
          "123456789012"
        ]
      }
    }
  ]
}
```

Contoh: Kebijakan titik akhir VPC untuk mengizinkan akses VPC hanya ke peran IAM tertentu

Kebijakan titik akhir VPC berikut memungkinkan akses penuh hanya ke peran IAM di akun.

redshiftrole AWS *123456789012* Semua prinsipal IAM lain ditolak aksesnya menggunakan titik akhir.

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:role/redshiftrole"
        ]
      }
    }
  ]
}
```

Ini hanya sampel. Dalam sebagian besar kasus penggunaan, kami menyarankan untuk melampirkan izin untuk tindakan tertentu untuk mempersempit ruang lingkup izin.

Contoh: Kebijakan titik akhir VPC untuk mengizinkan akses VPC hanya ke prinsipal IAM tertentu (pengguna)

Kebijakan titik akhir VPC berikut memungkinkan akses penuh hanya ke akun pengguna IAM.

redshiftadmin AWS *123456789012* Semua prinsipal IAM lain ditolak aksesnya menggunakan titik akhir.

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:user/redshiftadmin"
        ]
      }
    }
  ]
}
```

```

    }
  }]
}
```

Ini hanya sampel. Dalam sebagian besar kasus penggunaan, kami menyarankan untuk melampirkan izin ke peran sebelum menetapkan ke pengguna. Selain itu, sebaiknya gunakan tindakan khusus untuk mempersempit ruang lingkup izin.

Contoh: Kebijakan titik akhir VPC untuk mengizinkan operasi Amazon Redshift hanya-baca

Kebijakan titik akhir VPC berikut hanya mengizinkan AWS akun **123456789012** untuk melakukan tindakan Amazon Redshift yang ditentukan.

Tindakan yang ditentukan menyediakan akses hanya-baca yang setara untuk Amazon Redshift. Semua tindakan lain pada VPC ditolak untuk akun yang ditentukan. Juga, semua akun lain ditolak aksesnya. Untuk daftar tindakan Amazon Redshift, lihat [Tindakan, Sumber Daya, dan Kunci Kondisi untuk Amazon Redshift](#) di Panduan Pengguna IAM.

```

{
  "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"
        ]
    }
}
]
}

```

Contoh: Kebijakan titik akhir VPC yang menolak akses ke cluster tertentu

Kebijakan titik akhir VPC berikut memungkinkan akses penuh untuk semua akun dan prinsipal. Pada saat yang sama, ia menolak akses apa pun untuk AWS akun **123456789012** ke tindakan yang dilakukan di cluster Amazon Redshift dengan ID cluster **my-redshift-cluster**. Tindakan Amazon Redshift lainnya yang tidak mendukung izin tingkat sumber daya untuk kluster masih diizinkan. Untuk daftar tindakan Amazon Redshift dan jenis sumber daya yang sesuai, lihat [Tindakan, Sumber Daya, dan Kunci Kondisi untuk Amazon Redshift](#) di Panduan Pengguna IAM.

```

{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": "*"
    },
    {
      "Action": "*",
      "Effect": "Deny",

```

```

        "Resource": "arn:aws:redshift:us-east-1:123456789012:cluster:my-redshift-
cluster",
        "Principal": {
            "AWS": [
                "123456789012"
            ]
        }
    ]
}

```

Contoh Kebijakan Titik Akhir Tanpa Server Amazon Redshift

Berikut ini, Anda dapat menemukan contoh kebijakan titik akhir VPC untuk Redshift Serverless.

Contoh: Kebijakan titik akhir VPC untuk mengizinkan operasi Redshift Tanpa Server hanya-baca

Kebijakan titik akhir VPC berikut hanya mengizinkan AWS akun **123456789012** untuk melakukan tindakan Redshift Tanpa Server yang ditentukan.

Tindakan yang ditentukan menyediakan akses hanya-baca yang setara untuk Redshift Tanpa Server. Semua tindakan lain pada VPC ditolak untuk akun yang ditentukan. Juga, semua akun lain ditolak aksesnya. Untuk daftar tindakan Redshift Tanpa Server, lihat [Tindakan, Sumber Daya, dan Kunci Kondisi untuk Redshift Tanpa Server](#) di Panduan Pengguna IAM.

```

{
  "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"
      ]
    }
  }
]
}

```

Contoh: Kebijakan titik akhir VPC yang menolak akses ke workgroup tertentu

Kebijakan titik akhir VPC berikut memungkinkan akses penuh untuk semua akun dan prinsipal. Pada saat yang sama, ia menolak akses apa pun untuk AWS akun **123456789012** ke tindakan yang dilakukan pada workgroup Amazon Redshift dengan ID workgroup **my-redshift-workgroup**. Tindakan Amazon Redshift lainnya yang tidak mendukung izin tingkat sumber daya untuk grup kerja masih diizinkan. Untuk daftar tindakan Redshift Tanpa Server dan jenis sumber daya yang sesuai, lihat [Tindakan, Sumber Daya, dan Kunci Kondisi untuk Redshift Tanpa Server](#) di Panduan Pengguna IAM.

```

{
  "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"
        ]
      }
    }
  ]
}

```

```
}  
  ]  
}
```

Analisis konfigurasi dan kerentanan di Amazon Redshift

AWS menangani tugas-tugas keamanan dasar seperti sistem operasi tamu (OS) dan patching database, konfigurasi firewall, dan pemulihan bencana (DR). Prosedur ini telah ditinjau oleh pihak ketiga bersertifikat. Untuk informasi selengkapnya [Validasi kepatuhan untuk Amazon Redshift](#), lihat [model tanggung jawab bersama](#), dan [Praktik Terbaik untuk Keamanan, Identitas, dan Kepatuhan](#).

Amazon Redshift secara otomatis menerapkan peningkatan dan tambalan gudang data Anda sehingga Anda dapat fokus pada aplikasi Anda dan bukan pada administrasinya. Patch dan upgrade diterapkan selama jendela pemeliharaan yang dapat dikonfigurasi. Untuk informasi selengkapnya, lihat [Jendela pemeliharaan](#).

Editor kueri Amazon Redshift v2 adalah aplikasi yang AWS dikelola. Semua tambalan dan pembaruan diterapkan sesuai AWS kebutuhan.

Connect ke Redshift dengan sesi peran IAM yang disempurnakan Identitas

Anda dapat menggunakan Pusat Identitas IAM untuk menyediakan akses gabungan ke kluster Amazon Redshift dan grup kerja tanpa server. Pendekatan ini memungkinkan pengguna untuk mengautentikasi menggunakan kredensial Pusat Identitas mereka.

Amazon Redshift menyediakan operasi `GetIdentityCenterAuthToken` API untuk menghasilkan token resmi yang berisi informasi identitas pengguna. Ini APIs tersedia untuk cluster yang disediakan dan grup kerja tanpa server. Token memungkinkan akses masuk tunggal yang mulus ke database Amazon Redshift menggunakan penyiapan Pusat Identitas yang ada.

Prasyarat

Sebelum menggunakan otentikasi Pusat Identitas dengan Amazon Redshift, pastikan Anda memiliki yang berikut:

- Pengaturan Pusat Identitas: Akun Anda harus memiliki Pusat Identitas IAM yang dikonfigurasi dengan identitas pengguna dan tugas aplikasi yang sesuai. Untuk petunjuk penyiapan, lihat [Menyiapkan Pusat Identitas IAM](#).

Important

Jika Anda ingin terhubung ke Redshift, Anda harus menggunakan `redshift:connect` scope.

- Kredensial yang disempurnakan identitas: Aplikasi Anda harus menggunakan kredensial yang disempurnakan identitas yang berisi informasi identitas pengguna yang disematkan. Untuk informasi selengkapnya, lihat [Menggunakan sesi peran IAM yang disempurnakan identitas](#).
- Izin IAM: Peran atau pengguna IAM Anda harus memiliki izin untuk memanggil `GetIdentityCenterAuthToken` API dan mengakses cluster atau grup kerja yang ditentukan. Izin yang diperlukan:
 - Untuk cluster yang disediakan: `redshift:GetIdentityCenterAuthToken` di cluster ARNs (format:) `arn:aws:redshift:region:account:cluster:cluster-name`
 - Untuk grup kerja tanpa server: `redshift-serverless:GetIdentityCenterAuthToken` pada workgroup ARNs (format:) `arn:aws:redshift-serverless:region:account:workgroup/workgroup-name`

- Driver yang kompatibel: Gunakan driver Amazon Redshift JDBC atau ODBC yang mendukung token resmi Identity Center:
 - Driver JDBC: Lihat [Menginstal dan mengonfigurasi driver Amazon Redshift](#) JDBC versi 2.0
 - Driver ODBC: Lihat [Menginstal dan mengonfigurasi driver Amazon Redshift](#) ODBC versi 2.0

Cara kerja otentikasi Pusat Identitas

Autentikasi Pusat Identitas untuk Amazon Redshift menggunakan alur kerja berikut:

1. Aplikasi Anda memanggil `GetIdentityCenterAuthToken` API menggunakan kredensial yang disempurnakan identitas yang berisi informasi identitas pengguna yang disematkan.
2. Amazon Redshift memvalidasi identitas Pusat Identitas dan menghasilkan token resmi terenkripsi yang dicakup ke cluster atau grup kerja tertentu. Lihat contoh kebijakan IAM.
3. Aplikasi Anda menggunakan token ini untuk terhubung ke cluster atau workgroup Amazon Redshift yang ditentukan.
4. Bidang data Amazon Redshift memvalidasi token dan memberikan akses berdasarkan izin pengguna Pusat Identitas dalam aplikasi Pusat Identitas.

Important

API ini membutuhkan kredensi yang ditingkatkan identitas. Untuk informasi selengkapnya, lihat [Menggunakan sesi peran IAM yang disempurnakan identitas](#).

Jika Anda memanggil API tanpa kredensi yang disempurnakan identitas, Anda akan menerima kesalahan. `UnsupportedOperationFault`

GetIdentityCenterAuthToken Operasi API

Amazon Redshift menyediakan dua operasi `GetIdentityCenterAuthToken` API terpisah: satu untuk cluster yang disediakan dan satu untuk grup kerja tanpa server. Kedua operasi memiliki nama yang sama tetapi menerima parameter yang berbeda tergantung pada jenis sumber daya target.

GetIdentityCenterAuthToken untuk cluster yang disediakan

Untuk klaster Amazon Redshift yang disediakan, gunakan API `GetIdentityCenterAuthToken` di layanan Amazon Redshift untuk menghasilkan token resmi.

Sintaksis Permintaan

```
{  
  "ClusterIds": [ "string" ]  
}
```

Permintaan parameter

ClusterIds

Daftar pengidentifikasi klaster Amazon Redshift yang token akan diizinkan untuk diakses. Token hanya dapat digunakan untuk mengautentikasi dengan cluster yang ditentukan dalam daftar ini.

Tipe: Array string

Kendala panjang: Minimal 1 item. Maksimal 20 item.

Wajib: Ya

Contoh-contoh CLI

Contoh: Dapatkan token resmi untuk satu cluster

```
aws redshift get-identity-center-auth-token \  
  --cluster-ids my-redshift-cluster
```

Contoh: Dapatkan token resmi untuk beberapa cluster

```
aws redshift get-identity-center-auth-token \  
  --cluster-ids my-cluster-1 my-cluster-2
```

GetIdentityCenterAuthToken untuk kelompok kerja tanpa server

Untuk workgroup Amazon Redshift Tanpa Server, gunakan API di `GetIdentityCenterAuthToken` layanan Amazon Redshift Serverless untuk menghasilkan token resmi.

Sintaksis Permintaan

```
{
  "WorkgroupNames": [ "string" ]
}
```

Permintaan parameter

WorkgroupNames

Daftar nama workgroup Amazon Redshift Tanpa Server yang akan diizinkan untuk diakses oleh token. Token hanya dapat digunakan untuk mengautentikasi dengan kelompok kerja yang ditentukan dalam daftar ini.

Tipe: Array string

Kendala panjang: Minimal 1 item. Maksimal 20 item.

Wajib: Ya

Contoh-contoh CLI

Contoh: Dapatkan token resmi untuk satu grup kerja

```
aws redshift-serverless get-identity-center-auth-token \
  --workgroup-names my-workgroup
```

Contoh: Dapatkan token resmi untuk beberapa grup kerja

```
aws redshift-serverless get-identity-center-auth-token \
  --workgroup-names workgroup-1 workgroup-2
```

Sintaksis Respons

Keduanya APIs mengembalikan struktur respons yang sama:

```
{
  "AuthorizedToken": "string",
  "ExpirationTime": "timestamp"
}
```

Parameter respons

AuthorizedToken

Token resmi terenkripsi yang berisi informasi identitas pengguna dan daftar cluster atau kelompok kerja resmi. Token ini harus diperlakukan sebagai data sensitif.

Tipe: String

ExpirationTime

Tanggal dan waktu ketika token kedaluwarsa, di UTC. Token berlaku selama 1 jam dari waktu pembuatan.

Tipe: Timestamp

Contoh tanggapan

```
{
  "AuthorizedToken":
  "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJ1c2VyQGV4YW1wbGUuY29tIiwiaWF0IjoxNjQwOTk1MjAw
  "ExpirationTime": "2024-01-01T12:00:00Z"
}
```

Integrasi driver

Driver Amazon Redshift mendukung otentikasi Pusat Identitas melalui penggunaan token langsung:

Penggunaan token langsung

Setelah memanggil `GetIdentityCenterAuthToken` API untuk mendapatkan token, gunakan `IdpTokenAuthPlugin` dengan jenis `SUBJECT_TOKEN` token.

Konfigurasi Koneksi:

```
plugin_name = com.amazon.redshift.plugin.IdpTokenAuthPlugin
token_type = SUBJECT_TOKEN
token = {encrypted_token_from_api_response}
```

Untuk informasi mendetail tentang plugin autentikasi Pusat Identitas dan konfigurasi driver, lihat [Menyambung ke kluster Amazon Redshift](#).

Contoh kode Java

Contoh kode Java untuk terhubung menggunakan otentikasi Pusat Identitas:

```
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();
}
```

Persyaratan kebijakan IAM

Untuk menggunakan autentikasi Pusat Identitas dengan Amazon Redshift, izin IAM tertentu diperlukan di luar izin standar yang diperlukan untuk menghubungkan ke kluster dan grup kerja Amazon Redshift.

Izin API:

Untuk klaster yang disediakan, sesi peran IAM Anda yang disempurnakan harus memiliki:

- `redshift:GetIdentityCenterAuthToken` pada cluster ARNs
(format: `arn:aws:redshift:region:account:cluster:cluster-name`)

Untuk grup kerja tanpa server, sesi peran IAM Anda yang disempurnakan harus memiliki:

- `redshift-serverless:GetIdentityCenterAuthToken` pada workgroup ARNs
(format: `arn:aws:redshift-serverless:region:account:workgroup/workgroup-name`)

Contoh kebijakan IAM

Contoh kebijakan untuk klaster yang disediakan:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "redshift:GetIdentityCenterAuthToken"
      ],
      "Resource": [
        "arn:aws:redshift:us-east-1:123456789012:cluster:my-cluster"
      ]
    }
  ]
}
```

Contoh kebijakan untuk grup kerja tanpa server:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "redshift-serverless:GetIdentityCenterAuthToken"
      ]
    }
  ]
}
```

```

    ],
    "Resource": [
        "arn:aws:redshift-serverless:us-east-1:123456789012:workgroup/my-
workgroup"
    ]
}
]
}

```

Contoh kebijakan untuk beberapa sumber daya:

```

{
  "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/*"
      ]
    }
  ]
}

```

Ketersediaan wilayah

Otentikasi Pusat Identitas tersedia di AWS wilayah berikut:

- Wilayah komersial: Semua wilayah Amazon Redshift yang didukung
- AWS GovCloud: Tersedia dalam us-gov-east -1 dan us-gov-west -1

- Wilayah Tiongkok: Tersedia dalam cn-north-1 dan cn-northwest-1

Note

Ketersediaan fitur dapat bervariasi selama peluncuran awal.

Pemecahan masalah otentikasi Pusat Identitas

Gunakan bagian berikut untuk membantu memecahkan masalah yang Anda miliki dengan otentikasi Pusat Identitas.

Topik

- [Kesalahan API](#)
- [Kesalahan koneksi](#)

Kesalahan API

`GetIdentityCenterAuthTokenAPI` dapat mengembalikan kesalahan berikut:

- `UnsupportedOperationFault`— Permintaan dibuat tanpa kredensial yang ditingkatkan identitas atau propagasi identitas Pusat Identitas tidak didukung dalam konfigurasi saat ini.
- `ClusterNotFoundFault` (`redshift:GetIdentityCenterAuthToken`) — Satu atau lebih pengidentifikasi cluster yang ditentukan tidak ada atau Anda tidak memiliki izin untuk mengaksesnya.
- `WorkgroupNotFoundFault` (`redshift-serverless:GetIdentityCenterAuthToken`) - Satu atau beberapa nama workgroup yang ditentukan tidak ada atau Anda tidak memiliki izin untuk mengaksesnya.
- `InvalidParameterValueException`— Parameter permintaan tidak valid, seperti daftar kosong cluster atau workgroup, atau lebih dari 20 item dalam daftar.

Kesalahan koneksi

Saat menghubungkan ke Amazon Redshift menggunakan token Pusat Identitas, Anda mungkin mengalami kesalahan berikut:

- Kredensial token kedaluwarsa atau tidak valid — Buat token baru menggunakan `GetIdentityCenterAuthToken` API dan perbarui konfigurasi koneksi Anda.

- token tidak valid — Verifikasi format token dan pastikan itu dibuat untuk cluster atau workgroup yang benar.
- TYPE_INVALID_TOKEN — Pastikan Anda menggunakan parameter koneksi Anda. `token_type = SUBJECT_TOKEN`
- tipe penyedia identitas “awsidc” tidak didukung — Verifikasi klaster atau grup kerja Amazon Redshift Anda mendukung autentikasi Pusat Identitas dan perbarui ke versi yang kompatibel.
- AWS Penyedia identitas IDC tidak ada — Konfigurasi integrasi Pusat Identitas untuk sumber daya Amazon Redshift Anda melalui AWS konsol.
- Cakupan tidak valid. Kredensial pengguna tidak diizinkan untuk terhubung ke Amazon Redshift — Verifikasi pengguna memiliki izin yang sesuai di Pusat Identitas dan token dibuat untuk sumber daya yang benar.

Tugas jaringan

Anda dapat melakukan tugas-tugas jaringan seperti menyesuaikan koneksi Anda ke database Redshift. Anda mungkin ingin melakukan ini untuk mengontrol lalu lintas untuk keamanan atau tujuan lain. Anda juga dapat melakukan tugas terkait DNS, seperti menyiapkan nama domain khusus untuk sumber daya Redshift Anda. Tugas konfigurasi ini tersedia untuk Anda jika Anda memiliki kluster yang disediakan Amazon Redshift atau dengan grup kerja Amazon Redshift Tanpa Server.

Topik

- [Nama domain khusus untuk koneksi klien](#)
- [Titik akhir VPC yang dikelola Redshift](#)
- [Sumber daya Redshift dalam VPC](#)
- [Mengontrol lalu lintas jaringan dengan perutean VPC Redshift yang ditingkatkan](#)

Nama domain khusus untuk koneksi klien

Anda dapat membuat nama domain khusus, juga dikenal sebagai URL khusus, untuk kluster Amazon Redshift dan grup kerja Amazon Redshift Tanpa Server. Ini adalah catatan easy-to-read DNS yang merutekan koneksi klien SQL ke titik akhir Anda. Anda dapat mengonfigurasinya untuk cluster atau workgroup yang ada kapan saja. Ini memberikan beberapa manfaat:

- Nama domain kustom adalah string yang lebih sederhana daripada URL default, yang biasanya mencakup nama cluster atau nama workgroup dan wilayah. Lebih mudah untuk mengingat dan menggunakan.
- Anda dapat dengan cepat merutekan lalu lintas ke kluster atau grup kerja baru dalam kasus fail-over, misalnya. Ini membuatnya sehingga klien tidak perlu membuat perubahan konfigurasi ketika mereka terhubung kembali. Koneksi dapat dialihkan kembali secara terpusat, dengan gangguan minimal.
- Anda dapat menghindari berbagi informasi pribadi seperti nama server di URL koneksi. Anda dapat menyembunyikannya di URL khusus.

Saat Anda menyiapkan nama domain khusus menggunakan CNAME, tidak ada biaya tambahan dari Amazon Redshift. Anda mungkin ditagih dari penyedia DNS Anda untuk nama domain, jika Anda membuat yang baru, tetapi biaya ini biasanya kecil.

Mendaftarkan nama domain

Menyiapkan nama domain khusus terdiri dari beberapa tugas: Ini termasuk mendaftarkan nama domain dengan penyedia DNS Anda dan membuat sertifikat. Setelah Anda melakukan pekerjaan ini, Anda mengonfigurasi nama domain khusus di konsol Amazon Redshift, atau di konsol Amazon Redshift Tanpa Server, atau mengonfigurasinya dengan perintah. AWS CLI

Anda harus memiliki nama domain internet terdaftar untuk mengonfigurasi nama domain khusus di Amazon Redshift. Anda dapat mendaftarkan domain internet menggunakan Route 53, atau menggunakan penyedia pendaftaran domain pihak ketiga. Anda menyelesaikan tugas-tugas ini di luar konsol Amazon Redshift. Domain terdaftar adalah prasyarat untuk menyelesaikan prosedur yang tersisa untuk membuat domain khusus.

Note

Jika Anda menggunakan klaster yang disediakan, sebelum melakukan langkah-langkah untuk mengonfigurasi nama domain kustom, klaster tersebut harus diaktifkan relokasi. Untuk informasi selengkapnya, lihat [Merelokasi cluster](#). Langkah ini tidak diperlukan untuk Amazon Redshift Serverless.

Nama domain kustom biasanya mencakup domain root dan subdomain, seperti `mycluster.example.com`. Untuk mengkonfigurasinya, lakukan langkah-langkah berikut:

Buat entri DNS CNAME untuk nama domain kustom Anda

1. Daftarkan domain root, misalnya `example.com`. Secara opsional, Anda dapat menggunakan domain yang ada. Nama kustom Anda dapat dibatasi oleh pembatasan karakter tertentu, atau validasi penamaan lainnya. Untuk informasi selengkapnya tentang mendaftarkan domain dengan Route 53, lihat [Mendaftarkan domain baru](#).
2. Tambahkan data CNAME DNS yang mengarahkan nama domain kustom Anda ke titik akhir Redshift untuk klaster atau grup kerja Anda. Anda dapat menemukan titik akhir di properti untuk klaster atau grup kerja, di konsol Redshift atau di konsol Amazon Redshift Tanpa Server. Salin URL JDBC yang tersedia di properti klaster atau grup kerja, di bawah Informasi umum. URLs Tampak seperti berikut ini:
 - Untuk cluster Amazon Redshift: `redshift-cluster-sample.abc123456.us-east-1.redshift.amazonaws.com`

- Untuk grup kerja Amazon Redshift Tanpa Server: `endpoint-name.012345678901.us-east-1-dev.redshift-serverless-dev.amazonaws.com`

Jika URL memiliki awalan JDBC, hapus.

 Note

Catatan DNS tergantung pada ketersediaan, karena setiap nama harus unik dan tersedia untuk digunakan dalam organisasi Anda.

Batasan

Ada beberapa batasan terkait pembuatan catatan CNAME untuk domain khusus:

- Membuat beberapa nama domain khusus untuk klaster yang disediakan atau grup kerja Amazon Redshift Tanpa Server yang sama tidak didukung. Anda hanya dapat mengaitkan satu catatan CNAME.
- Mengaitkan catatan CNAME dengan lebih dari satu klaster atau grup kerja tidak didukung. CNAME untuk setiap sumber daya Redshift harus unik.

Setelah mendaftarkan domain dan membuat catatan CNAME, Anda memilih sertifikat baru atau yang sudah ada. Anda melakukan langkah ini menggunakan AWS Certificate Manager:

Kami menyarankan Anda membuat [sertifikat tervalidasi DNS](#) yang memenuhi kelayakan untuk perpanjangan terkelola, yang tersedia dengan AWS Certificate Manager Perpanjangan terkelola berarti ACM memperbarui sertifikat Anda secara otomatis atau mengirimkan Anda pemberitahuan email saat kedaluwarsa mendekati. Untuk informasi selengkapnya, lihat [Perpanjangan terkelola untuk sertifikat ACM](#).

Meminta sertifikat untuk nama domain

Amazon Redshift atau Amazon Redshift Serverless memerlukan sertifikat Secure Sockets Layer (SSL) yang divalidasi untuk titik akhir kustom agar komunikasi tetap aman dan memverifikasi kepemilikan nama domain. Anda dapat menggunakan AWS Certificate Manager akun Anda dengan manajemen sertifikat AWS KMS key untuk aman. Validasi keamanan mencakup verifikasi nama host lengkap (`sslmode=verify-full`).

Perpanjangan sertifikat dikelola oleh Amazon Redshift hanya jika Anda memilih validasi DNS, bukan validasi email. Jika Anda menggunakan validasi email, Anda dapat menggunakan sertifikat, tetapi Anda harus melakukan pembaruan sendiri, sebelum kedaluwarsa. Kami menyarankan Anda memilih validasi DNS untuk sertifikat Anda. Anda dapat memantau tanggal kedaluwarsa sertifikat yang diimpor di. AWS Certificate Manager

Meminta sertifikat dari ACM untuk nama domain

1. Masuk ke Konsol Manajemen AWS dan buka konsol ACM di <https://console.aws.amazon.com/acm/>.
2. Pilih Minta sertifikat.
3. Masukkan nama domain kustom Anda di bidang Nama domain.

 Note

Anda dapat menentukan banyak awalan, selain domain sertifikat, untuk menggunakan satu sertifikat untuk beberapa catatan domain khusus. Untuk mengilustrasikan, Anda dapat menggunakan catatan tambahan seperti `one.example.com`, `two.example.com`, atau catatan DNS wildcard seperti `*.example.com` dengan sertifikat yang sama.

4. Pilih Tinjau dan minta.
5. Pilih Konfirmasi dan minta.
6. Untuk permintaan yang valid, pemilik domain internet yang terdaftar harus menyetujui permintaan tersebut sebelum ACM mengeluarkan sertifikat. Pastikan status muncul sebagai Diterbitkan di konsol ACM, setelah Anda selesai dengan langkah-langkahnya.

Mengkonfigurasi domain kustom

Anda dapat menggunakan konsol Amazon Redshift atau Amazon Redshift Tanpa Server untuk membuat URL domain khusus Anda. Jika Anda belum mengonfigurasinya, properti nama domain kustom akan muncul sebagai tanda hubung (-) di bawah Informasi umum. Setelah membuat catatan CNAME dan sertifikat, Anda mengaitkan nama domain khusus untuk klaster atau grup kerja.

Untuk membuat asosiasi domain kustom, izin IAM berikut diperlukan:

- `redshift:CreateCustomDomainAssociation`— Anda dapat membatasi izin ke cluster tertentu dengan menambahkan ARN-nya.

- `redshiftServerless:CreateCustomDomainAssociation`— Anda dapat membatasi izin ke workgroup tertentu dengan menambahkan ARN nya.
- `acm:DescribeCertificate`

Sebagai praktik terbaik, sebaiknya lampirkan kebijakan izin ke peran IAM dan kemudian menetapkan ke pengguna dan grup sesuai kebutuhan. Untuk informasi selengkapnya, lihat [Manajemen identitas dan akses di Amazon Redshift](#).

Anda menetapkan nama domain kustom dengan melakukan langkah-langkah berikut.

1. Pilih klaster di konsol Redshift, atau workgroup di konsol Amazon Redshift Tanpa Server, dan pilih Buat nama domain khusus di bawah menu Tindakan. Dialog muncul.
2. Masukkan nama domain kustom.
3. Pilih ARN dari AWS Certificate Manager untuk Sertifikat ACM. Konfirmasikan perubahan Anda. Sesuai panduan dalam langkah-langkah yang Anda ambil untuk membuat sertifikat, sebaiknya Anda memilih sertifikat tervalidasi DNS yang memenuhi syarat untuk perpanjangan terkelola AWS Certificate Manager
4. Verifikasi di properti cluster bahwa nama domain kustom dan sertifikat domain kustom ARN diisi dengan entri Anda. Tanggal kedaluwarsa sertifikat domain kustom juga terdaftar.

Setelah domain kustom dikonfigurasi, menggunakan hanya `sslmode=verify-full` berfungsi untuk domain kustom baru. Itu tidak berfungsi untuk titik akhir default. Tetapi Anda masih dapat terhubung ke titik akhir default dengan menggunakan mode ssl lainnya, seperti `sslmode=verify-ca`

Note

Sebagai pengingat, [relokasi cluster](#) bukanlah prasyarat untuk mengonfigurasi fitur jaringan Redshift tambahan. Anda tidak perlu menyalakannya untuk mengaktifkan yang berikut:

- Menghubungkan dari VPC lintas akun atau lintas wilayah ke Redshift — Anda dapat terhubung dari satu AWS virtual private cloud (VPC) ke yang lain yang berisi database Redshift. Ini membuatnya lebih mudah untuk mengelola, misalnya, akses klien dari akun yang berbeda atau VPCs, tanpa harus menyediakan akses VPC lokal ke identitas yang terhubung ke database. Untuk informasi selengkapnya, lihat [Menghubungkan ke Amazon Redshift Tanpa Server dari titik akhir VPC Redshift](#) di akun atau wilayah lain.

- Menyiapkan nama domain khusus — Anda dapat membuat nama domain khusus, seperti yang dijelaskan dalam topik ini, untuk membuat nama titik akhir lebih relevan dan sederhana.

Menyambung ke klaster yang disediakan Amazon Redshift atau grup kerja Amazon Redshift Tanpa Server

Untuk terhubung dengan nama domain kustom, izin IAM berikut diperlukan untuk klaster yang disediakan: `redshift:DescribeCustomDomainAssociations` Untuk Amazon Redshift Tanpa Server, Anda tidak perlu menambahkan izin.

Sebagai praktik terbaik, sebaiknya lampirkan kebijakan izin ke peran IAM dan kemudian menentukannya ke pengguna dan grup sesuai kebutuhan. Untuk informasi selengkapnya, lihat [Manajemen identitas dan akses di Amazon Redshift](#).

Setelah menyelesaikan langkah-langkah untuk membuat CNAME dan menentukannya ke klaster atau grup kerja di konsol, Anda dapat memberikan URL kustom di properti koneksi klien SQL Anda. Perhatikan bahwa mungkin ada penundaan dari propagasi DNS segera setelah pembuatan catatan CNAME.

1. Buka klien SQL. Misalnya, Anda dapat menggunakan SQL/Workbench J. Buka properti untuk koneksi, dan tambahkan nama domain khusus untuk string koneksi. Misalnya, `jdbc:redshift://mycluster.example.com:5439/dev?sslmode=verify-full`. Dalam contoh ini, `dev` menentukan database default.
2. Tambahkan Username dan Password untuk pengguna database Anda.
3. Uji koneksi. Kemampuan Anda untuk menanyakan sumber daya database seperti tabel tertentu dapat bervariasi, berdasarkan izin yang diberikan kepada pengguna database atau diberikan ke peran database Amazon Redshift yang ditetapkan.

Perhatikan bahwa Anda mungkin harus menyetel klaster atau grup kerja agar dapat diakses publik untuk terhubung dengannya jika ada di VPC. Anda dapat mengubah pengaturan ini di properti jaringan.

 Note

Koneksi ke nama domain khusus didukung dengan driver JDBC, ODBC, dan Python.

Mengganti nama cluster yang memiliki domain kustom yang ditetapkan

 Note

Rangkaian langkah ini tidak berlaku untuk workgroup Amazon Redshift Serverless. Anda tidak dapat mengubah nama workgroup.

Untuk mengganti nama cluster yang memiliki nama domain khusus, izin `acm:DescribeCertificate` IAM diperlukan.

1. Buka konsol Amazon Redshift dan pilih cluster yang namanya ingin Anda ubah. Pilih Edit untuk mengedit properti cluster.
2. Edit pengenalan Cluster. Anda juga dapat mengubah properti lain untuk cluster. Lalu pilih Simpan Perubahan.
3. Setelah klaster diganti namanya, Anda harus memperbarui catatan DNS untuk mengubah entri CNAME untuk domain kustom untuk menunjuk ke titik akhir Amazon Redshift yang diperbarui.

Menjelaskan asosiasi domain kustom

Gunakan perintah di bagian ini untuk mendapatkan daftar nama domain kustom yang terkait dengan klaster tertentu yang disediakan atau dengan workgroup Amazon Redshift Tanpa Server.

Anda memerlukan izin berikut:

- Untuk klaster yang disediakan: `redshift:DescribeCustomDomainAssociations`
- Untuk grup kerja Amazon Redshift Tanpa Server:
`redshiftServerless:ListCnameAssociations`

Sebagai praktik terbaik, sebaiknya lampirkan kebijakan izin ke peran IAM dan kemudian menetakannya ke pengguna dan grup sesuai kebutuhan. Untuk informasi selengkapnya, lihat [Manajemen identitas dan akses di Amazon Redshift](#).

Berikut ini menunjukkan contoh perintah untuk mencantumkan nama domain kustom untuk klaster Amazon Redshift tertentu:

```
aws redshift describe-custom-domain-associations --custom-domain-name customdomainname
```

Anda dapat menjalankan perintah ini ketika Anda memiliki nama domain kustom diaktifkan untuk menentukan nama domain kustom yang terkait dengan cluster. Untuk informasi selengkapnya tentang perintah CLI untuk menjelaskan asosiasi domain kustom, lihat. [describe-custom-domain-associations](#)

Demikian pula, berikut ini menunjukkan contoh perintah untuk mencantumkan nama domain kustom untuk workgroup Amazon Redshift Serverless yang diberikan. Ada beberapa cara berbeda untuk melakukan ini. Anda hanya dapat memberikan nama domain khusus:

```
aws redshift-serverless list-custom-domain-associations --custom-domain-name customdomainname
```

Anda juga bisa mendapatkan asosiasi dengan hanya memberikan sertifikat ARN:

```
aws redshift-serverless list-custom-domain-associations --custom-domain-certificate-arn certificatearn
```

Anda dapat menjalankan perintah ini ketika Anda memiliki nama domain kustom diaktifkan untuk menentukan nama domain kustom yang terkait dengan workgroup. Anda juga dapat menjalankan perintah untuk mendapatkan properti dari asosiasi domain kustom. Untuk melakukan ini, Anda harus memberikan nama domain kustom dan nama workgroup sebagai parameter. Ini mengembalikan sertifikat ARN, nama workgroup, dan waktu kedaluwarsa sertifikat domain kustom:

```
aws redshift-serverless get-custom-domain-association --workgroup-name workgroupname --custom-domain-name customdomainname
```

[Untuk informasi selengkapnya tentang perintah referensi CLI yang tersedia untuk Amazon Redshift Tanpa Server, lihat `redshift-serverless`.](#)

Mengaitkan domain kustom dengan sertifikat yang berbeda

Untuk mengubah asosiasi sertifikat untuk nama domain kustom, izin IAM berikut diperlukan:

- `redshift:ModifyCustomDomainAssociation`

- `acm:DescribeCertificate`

Sebagai praktik terbaik, sebaiknya lampirkan kebijakan izin ke peran IAM dan kemudian menentukannya ke pengguna dan grup sesuai kebutuhan. Untuk informasi selengkapnya, lihat [Manajemen identitas dan akses di Amazon Redshift](#).

Gunakan perintah berikut untuk mengaitkan domain kustom dengan sertifikat yang berbeda. `custom-domain-certificate-arn` Argumen `--custom-domain-name` dan adalah wajib. ARN untuk sertifikat baru harus berbeda dari ARN yang ada.

```
aws redshift modify-custom-domain-association --cluster-id redshiftcluster --custom-domain-name customdomainname --custom-domain-certificate-arn certificatearn
```

Contoh berikut menunjukkan cara mengaitkan domain kustom dengan sertifikat yang berbeda untuk workgroup Amazon Redshift Tanpa Server.

```
aws redshift-serverless modify-custom-domain-association --workgroup-name redshiftworkgroup --custom-domain-name customdomainname --custom-domain-certificate-arn certificatearn
```

Ada penundaan maksimum 30 detik sebelum Anda dapat terhubung ke cluster. Bagian dari penundaan terjadi saat cluster Amazon Redshift memperbarui propertinya, dan ada beberapa penundaan tambahan saat DNS diperbarui. Untuk informasi selengkapnya tentang API dan setiap setelan properti, lihat [ModifyCustomDomainAssociation](#).

Menghapus domain kustom

Untuk menghapus nama domain kustom, pengguna harus memiliki izin untuk tindakan berikut:

- Untuk klaster yang disediakan: `redshift:DeleteCustomDomainAssociation`
- Untuk grup kerja Amazon Redshift Tanpa Server:
`redshiftServerless:DeleteCustomDomainAssociation`

Di konsol

Anda dapat menghapus nama domain kustom dengan memilih tombol Tindakan dan memilih Hapus nama domain kustom. Setelah Anda melakukan ini, Anda masih dapat terhubung ke server dengan memperbarui alat Anda untuk menggunakan titik akhir yang tercantum di konsol.

Menggunakan perintah CLI

Contoh berikut menunjukkan cara menghapus nama domain kustom. Operasi hapus mengharuskan Anda memberikan nama domain kustom yang ada untuk kluster.

```
aws redshift delete-custom-domain-association --cluster-id redshiftcluster --custom-domain-name customdomainname
```

Contoh berikut menunjukkan cara menghapus nama domain kustom untuk workgroup Amazon Redshift Tanpa Server. Nama domain kustom adalah parameter yang diperlukan.

```
aws redshift-serverless delete-custom-domain-association --workgroup-name workgroupname --custom-domain-name customdomainname
```

Lihat informasi yang lebih lengkap di [DeleteCustomDomainAssociation](#).

Titik akhir VPC yang dikelola Redshift

Secara default, kluster Amazon Redshift atau grup kerja Amazon Redshift Tanpa Server disediakan di cloud pribadi virtual (VPC). VPC dapat diakses dari VPC atau subnet lain ketika Anda mengizinkan akses publik atau mengatur gateway internet, perangkat NAT, atau AWS Direct Connect koneksi untuk mengarahkan lalu lintas ke sana. Anda juga dapat mengakses kluster atau workgroup dengan menyiapkan endpoint VPC yang dikelola RedShift (didukung oleh). AWS PrivateLink

Anda dapat mengatur endpoint VPC yang dikelola RedShift sebagai koneksi pribadi antara VPC yang berisi cluster atau workgroup dan VPC tempat alat klien berjalan. Jika cluster atau workgroup berada di akun lain, pemilik akun (pemberi) harus memberikan akses ke akun penghubung (penerima hibah). Dengan pendekatan ini, Anda dapat mengakses gudang data tanpa menggunakan alamat IP publik atau merutekan lalu lintas melalui internet.

Ini adalah alasan umum untuk mengizinkan akses menggunakan titik akhir VPC yang dikelola RedShift:

- AWS akun A ingin mengizinkan VPC di AWS akun B memiliki akses ke cluster atau workgroup.
- AWS akun A ingin mengizinkan VPC yang juga ada di AWS akun A untuk memiliki akses ke cluster atau workgroup.
- AWS akun A ingin mengizinkan subnet yang berbeda di VPC AWS dalam akun A untuk memiliki akses ke cluster atau workgroup.

Alur kerja untuk menyiapkan titik akhir VPC yang dikelola RedShift untuk mengakses klaster atau grup kerja di akun lain adalah sebagai berikut:

1. Akun pemilik memberikan otorisasi akses ke akun lain dan menentukan ID akun dan AWS pengidentifikasi VPC (atau semua) penerima hibah. VPCs
2. Akun penerima hibah diberi tahu bahwa mereka memiliki izin untuk membuat titik akhir VPC yang dikelola RedShift.
3. Akun penerima hibah membuat titik akhir VPC yang dikelola RedShift.
4. Akun penerima hibah mengakses klaster atau grup kerja akun pemilik menggunakan titik akhir VPC yang dikelola RedShift.

Anda dapat melakukan ini menggunakan konsol Amazon Redshift, the AWS CLI, atau Amazon Redshift API.

Pertimbangan saat menggunakan titik akhir VPC yang dikelola RedShift

Note

Untuk membuat atau mengubah titik akhir VPC yang dikelola RedShift, Anda memerlukan `ec2:CreateVpcEndpoint` izin `ec2:ModifyVpcEndpoint` atau kebijakan IAM, selain izin lain yang ditentukan dalam kebijakan terkelola. `AWS AmazonRedshiftFullAccess`

Saat menggunakan titik akhir VPC yang dikelola RedShift, ingatlah hal berikut:

- Jika Anda menggunakan cluster yang disediakan, itu harus memiliki tipe RA3 node. Workgroup Amazon Redshift Tanpa Server juga berfungsi untuk menyiapkan titik akhir VPC.
- Untuk kluster yang disediakan, pastikan klaster diaktifkan untuk relokasi cluster atau multi-AZ. Untuk informasi tentang persyaratan untuk mengaktifkan relokasi klaster, lihat [Merelokasi cluster](#). Untuk informasi tentang mengaktifkan Multi-AZ, lihat. [Menyiapkan Multi-AZ saat membuat cluster baru](#)
- Pastikan bahwa cluster atau workgroup untuk mengakses melalui grup keamanannya tersedia dalam rentang port yang valid 5431-5455 dan 8191-8215. Defaultnya adalah 5439.
- Anda dapat memodifikasi grup keamanan VPC yang terkait dengan titik akhir VPC yang dikelola RedShift yang ada. Untuk mengubah pengaturan lain, hapus titik akhir VPC yang dikelola RedShift saat ini dan buat yang baru.

- Jumlah endpoint VPC yang dikelola RedShift yang dapat Anda buat terbatas pada kuota endpoint VPC Anda.
- Titik akhir VPC yang dikelola RedShift tidak dapat diakses dari internet. Titik akhir VPC yang dikelola RedShift hanya dapat diakses dalam VPC di mana titik akhir disediakan atau dari titik akhir mana pun yang VPCs diintip dengan VPC di mana titik akhir disediakan sebagaimana diizinkan oleh tabel rute dan grup keamanan.
- Anda tidak dapat menggunakan konsol VPC Amazon untuk mengelola titik akhir VPC yang dikelola RedShift.
- Saat Anda membuat endpoint VPC yang dikelola RedShift untuk kluster yang disediakan, VPC yang Anda pilih harus memiliki grup subnet. Untuk membuat grup subnet, lihat [Membuat grup subnet cluster](#).
- Jika Availability Zone sedang down, Amazon Redshift tidak membuat elastic network interface baru di Availability Zone lainnya. Anda mungkin perlu membuat titik akhir baru dalam kasus ini.

Untuk informasi tentang kuota dan batasan penamaan, lihat. [Kuota dan batas di Amazon Redshift](#)

Untuk informasi lebih lanjut mengenai harga, lihat [harga AWS PrivateLink](#).

Memberikan akses ke VPC

Jika VPC yang ingin Anda akses ke kluster atau grup kerja Anda ada di AWS akun lain, pastikan untuk mengotorisasinya dari akun pemilik (pemberi).

Untuk mengizinkan VPC di AWS akun lain memiliki akses ke kluster atau grup kerja Anda

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Cluster. Untuk Amazon Redshift Tanpa Server, pilih dasbor Tanpa Server.
3. Untuk kluster yang ingin Anda izinkan aksesnya, lihat detailnya dengan memilih nama cluster. Pilih tab Properties dari cluster.

Bagian Akun yang Diberikan menampilkan akun dan terkait VPCs yang memiliki akses ke kluster Anda. Untuk workgroup Amazon Redshift Tanpa Server, pilih workgroup. Akun yang diberikan tersedia di bawah tab Akses data.

4. Pilih Berikan akses untuk menampilkan formulir untuk memasukkan informasi Penerima Hibah untuk menambahkan akun.

5. Untuk ID AWS akun, masukkan ID akun yang Anda berikan akses. Anda dapat memberikan akses ke spesifik VPCs atau semua VPCs di akun yang ditentukan.
6. Pilih Berikan akses untuk memberikan akses.

Membuat titik akhir VPC yang dikelola Redshift

Jika Anda memiliki klaster atau workgroup, atau Anda telah diberikan akses untuk mengelolanya, Anda dapat membuat endpoint VPC yang dikelola RedShift untuknya.

Untuk membuat titik akhir VPC yang dikelola RedShift

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Konfigurasi.

Halaman Konfigurasi menampilkan titik akhir VPC yang dikelola RedShift yang telah dibuat. Untuk melihat detail titik akhir, pilih namanya. Untuk Amazon Redshift Tanpa Server, titik akhir VPC berada di bawah tab Akses data, saat Anda memilih grup kerja.

3. Pilih Buat titik akhir untuk menampilkan formulir untuk memasukkan informasi tentang titik akhir yang akan ditambahkan.
4. Masukkan nilai untuk nama Endpoint, ID AWS akun 12 digit, Virtual Private Cloud (VPC) tempat titik akhir berada, Subnet dan grup keamanan VPC.

Subnet di Subnet mendefinisikan subnet dan alamat IP tempat Amazon Redshift menyebarkan titik akhir. Amazon Redshift memilih subnet yang memiliki alamat IP yang tersedia untuk antarmuka jaringan yang terkait dengan titik akhir.

Aturan grup keamanan dalam grup keamanan VPC menentukan port, protokol, dan sumber untuk lalu lintas masuk yang Anda otorisasi untuk titik akhir Anda. Anda mengizinkan akses ke port yang dipilih melalui grup keamanan atau rentang CIDR tempat beban kerja Anda berjalan.

5. Pilih Buat titik akhir untuk membuat titik akhir.

Setelah titik akhir dibuat, Anda dapat mengakses klaster atau grup kerja melalui URL yang ditampilkan di URL Endpoint dalam pengaturan konfigurasi untuk titik akhir VPC yang dikelola RedShift.

Sumber daya Redshift dalam VPC

Anda dapat meluncurkan cluster Amazon Redshift atau workgroup Amazon Redshift Tanpa Server di VPC pada platform EC2-VPC berdasarkan layanan Amazon VPC. Untuk informasi selengkapnya, lihat [Gunakan EC2 untuk membuat cluster Anda](#).

Note

Meluncurkan cluster dan grup kerja Tanpa Server ke dalam VPCs penyewaan khusus tidak didukung. Untuk informasi selengkapnya, lihat [Instans khusus](#) di Panduan Pengguna Amazon VPC.

Saat menyediakan sumber daya dalam VPC, Anda harus melakukan hal berikut:

- Berikan informasi VPC.

Saat membuat klaster yang disediakan di VPC, Anda harus memberikan informasi VPC Anda dengan membuat grup subnet klaster. Informasi ini mencakup ID VPC dan daftar subnet di VPC Anda. Saat Anda meluncurkan cluster, Anda menyediakan grup subnet sehingga Redshift dapat menyediakannya di salah satu subnet di VPC. Dengan Amazon Redshift Serverless, prosesnya serupa. Anda menetapkan subnet langsung ke grup kerja Tanpa Server Anda. Tetapi dalam kasus Tanpa Server Anda tidak membuat grup subnet. Untuk informasi selengkapnya tentang membuat grup subnet di Amazon Redshift, lihat [Subnet untuk sumber daya Redshift](#). Untuk informasi selengkapnya tentang pengaturan VPC, lihat [Memulai Amazon VPC di](#) Panduan Memulai VPC Amazon.

- Secara opsional, konfigurasi opsi aksesibilitas.

Cluster yang disediakan dan grup kerja tanpa server di Amazon Redshift bersifat pribadi secara default. Jika Anda mengonfigurasi klaster yang disediakan atau grup kerja tanpa server agar dapat diakses publik, Amazon Redshift menggunakan alamat IP elastis untuk alamat IP eksternal. Alamat IP elastis adalah alamat IP statis. Dengan itu, Anda dapat mengubah konfigurasi dasar Anda tanpa mempengaruhi alamat IP yang digunakan klien untuk terhubung. Pendekatan ini dapat membantu untuk situasi seperti pemulihan setelah kegagalan. Apakah Anda membuat alamat IP elastis tergantung pada pengaturan relokasi zona ketersediaan Anda. Ada dua opsi:

1. Jika relokasi zona ketersediaan diaktifkan dan ingin mengaktifkan akses publik, Anda tidak menentukan alamat IP elastis. Alamat IP elastis yang dikelola oleh Amazon Redshift ditetapkan. Ini terkait dengan AWS akun Anda.

2. Jika relokasi zona ketersediaan dimatikan dan ingin mengaktifkan akses publik, Anda dapat memilih untuk membuat alamat IP elastis untuk VPC di Amazon EC2, sebelum meluncurkan kluster atau grup kerja Amazon Redshift Anda. Jika Anda tidak membuat alamat IP, Amazon Redshift menyediakan alamat IP elastis yang dikonfigurasi untuk digunakan untuk VPC. Alamat IP elastis ini dikelola oleh Amazon Redshift dan tidak terkait dengan akun Anda AWS .

Untuk informasi selengkapnya, lihat [Alamat IP Elastis](#) di Panduan Pengguna Amazon EC2.

Dalam beberapa kasus, Anda mungkin memiliki cluster yang dapat diakses publik di VPC dan Anda ingin menghubungkannya dengan menggunakan alamat IP pribadi dari dalam VPC. Jika demikian, atur parameter VPC berikut ke: `true`

- `DNS resolution`
- `DNS hostnames`

Perhatikan bahwa dengan Amazon Redshift Serverless, Anda tidak dapat terhubung dengan cara ini.

Misalkan Anda memiliki kluster penyediaan yang dapat diakses publik di VPC tetapi tidak menyetel parameter tersebut ke dalam VPC. `true` Dalam kasus ini, koneksi yang dibuat dari dalam VPC menyelesaikan ke alamat IP elastis sumber daya alih-alih alamat IP pribadi. Kami menyarankan Anda mengatur parameter ini `true` dan menggunakan alamat IP pribadi untuk cluster yang dapat diakses publik saat menghubungkan dari dalam VPC. Untuk informasi selengkapnya, lihat [Menggunakan DNS dengan VPC Anda](#) dalam Panduan Pengguna Amazon VPC.

Note

Jika Anda memiliki kluster yang dapat diakses publik di VPC, koneksi dari dalam VPC terus menggunakan alamat IP elastis untuk menghubungkannya, hingga Anda mengubah ukurannya, jika itu adalah kluster yang disediakan. Hal ini terjadi bahkan dengan parameter sebelumnya ditetapkan. Setiap cluster baru yang dibuat mengikuti perilaku baru menggunakan alamat IP pribadi saat menghubungkan ke cluster yang dapat diakses publik dari dalam VPC yang sama.

Alamat IP elastis adalah alamat IP eksternal untuk mengakses sumber daya di luar VPC. Untuk kluster yang disediakan, ini tidak terkait dengan alamat IP Publik dan alamat IP Pribadi yang ditampilkan di konsol Amazon Redshift di bawah alamat IP Node. Alamat IP node cluster publik

dan pribadi muncul terlepas dari apakah cluster dapat diakses publik atau tidak. Mereka hanya digunakan dalam keadaan tertentu untuk mengonfigurasi aturan masuk pada host jarak jauh. Keadaan ini terjadi saat Anda memuat data dari instans Amazon EC2 atau host jarak jauh lainnya menggunakan koneksi Secure Shell (SSH). Untuk informasi selengkapnya, lihat [Langkah 1: Ambil kunci publik klaster dan alamat IP node cluster](#) di Panduan Pengembang Database Amazon Redshift.

 Note

Alamat IP node tidak berlaku untuk grup kerja Redshift Serverless.

Opsi untuk mengaitkan cluster yang disediakan dengan alamat IP elastis tersedia saat Anda membuat cluster atau memulihkan cluster dari snapshot. Dalam beberapa kasus, Anda mungkin ingin mengaitkan cluster dengan alamat IP elastis atau mengubah alamat IP elastis yang terkait dengan cluster. Untuk melampirkan alamat IP elastis setelah cluster dibuat, pertama-tama perbarui cluster sehingga tidak dapat diakses publik, kemudian membuatnya dapat diakses publik dan menambahkan alamat IP Elastis dalam operasi yang sama.

Untuk informasi selengkapnya tentang cara membuat klaster yang disediakan atau grup kerja Amazon Redshift Tanpa Server dapat diakses publik, dan memiliki alamat IP Elastis yang ditetapkan, [lihat Aksesibilitas publik](#) dengan konfigurasi grup keamanan default atau kustom.

- Kaitkan grup keamanan VPC.

Anda memberikan akses masuk menggunakan grup keamanan VPC. Untuk informasi selengkapnya, lihat [Mengonfigurasi setelan komunikasi grup keamanan untuk klaster Amazon Redshift](#), yang memberikan panduan tentang mengonfigurasi aturan masuk dan keluar antara klien dan klaster yang disediakan atau grup kerja Amazon Redshift Tanpa Server. Sumber daya lain yang membantu Anda memahami grup keamanan adalah [Keamanan di VPC Anda di Panduan Pengguna Amazon VPC](#)

Memulihkan snapshot dari klaster yang disediakan atau grup kerja Tanpa Server di VPC

Cuplikan cluster atau grup kerja Tanpa Server di VPC hanya dapat dipulihkan di VPC, bukan di luar VPC. Anda dapat mengembalikannya di VPC yang sama atau VPC lain di akun Anda. Untuk informasi selengkapnya tentang snapshot, lihat [Cuplikan dan cadangan Amazon Redshift](#).

Membuat klaster yang disediakan Redshift atau workgroup Amazon Redshift Serverless di VPC

Berikut ini adalah langkah-langkah umum bagaimana Anda dapat menyebarkan cluster atau workgroup di virtual private cloud (VPC) Anda.

Untuk membuat cluster atau grup kerja Tanpa Server di VPC

1. Konfigurasi VPC — Anda dapat membuat sumber daya Redshift baik di VPC default untuk akun Anda, jika akun Anda memilikinya, atau di VPC yang Anda buat. Untuk informasi selengkapnya, lihat [Gunakan EC2 untuk membuat cluster Anda](#). Untuk membuat VPC, lihat [Subnet untuk VPC Anda di Panduan Pengguna Amazon VPC](#). Catat pengenal VPC, subnet, dan Availability Zone subnet. Anda memerlukan informasi ini saat meluncurkan klaster atau grup kerja Anda.

Note

Anda harus memiliki setidaknya satu subnet yang ditentukan dalam VPC Anda, sehingga Anda dapat menambahkannya ke grup subnet di langkah berikutnya. Untuk informasi selengkapnya tentang menambahkan subnet ke VPC Anda, [lihat Menambahkan subnet ke VPC Anda di Panduan Pengguna Amazon VPC](#).

2. Buat grup subnet klaster Amazon Redshift untuk menentukan subnet mana yang dapat digunakan cluster Amazon Redshift Anda di VPC. Untuk Redshift Serverless, Anda tidak membuat grup subnet, melainkan menetapkan kumpulan subnet ke workgroup Anda saat Anda membuatnya. Anda dapat melakukan ini di dasbor Tanpa Server saat membuat grup kerja.

Anda dapat membuat grup subnet menggunakan konsol Amazon Redshift atau secara terprogram. Untuk informasi selengkapnya, lihat [Subnet untuk sumber daya Redshift](#).

3. Otorisasi akses untuk koneksi masuk dalam grup keamanan VPC yang Anda kaitkan dengan klaster atau grup kerja. Anda dapat mengaktifkan klien di luar VPC (di internet publik) untuk terhubung ke cluster. Untuk melakukan ini, Anda mengaitkan cluster dengan grup keamanan VPC yang memberikan akses masuk. Untuk informasi selengkapnya, lihat [Mengonfigurasi setelan komunikasi grup keamanan untuk klaster Amazon Redshift atau grup kerja Amazon Redshift Tanpa Server](#).
4. Ikuti langkah-langkah untuk membuat klaster di konsol yang disediakan Redshift atau grup kerja atau di konsol Amazon Redshift Tanpa Server. Di Jaringan dan keamanan, tentukan grup

keamanan Virtual Private Cloud (VPC), Cluster subnet, dan grup keamanan VPC yang Anda atur.

Untuk panduan yang menunjukkan langkah-langkah lebih mendetail untuk membuat klaster gudang data yang disediakan, lihat Memulai gudang data yang [disediakan Amazon Redshift di Panduan Memulai Amazon Redshift](#). Untuk informasi selengkapnya tentang membuat workgroup Amazon Redshift Tanpa Server, lihat Memulai [gudang data Amazon Redshift Tanpa Server di Panduan Memulai Amazon Redshift](#).

Anda dapat mengikuti langkah-langkah Memulai untuk menguji klaster atau workgroup dengan mengunggah data sampel dan mencoba contoh kueri. Untuk informasi selengkapnya, lihat [Memulai gudang data Amazon Redshift Tanpa Server di Panduan Memulai Pergeseran](#) Merah Amazon.

Grup keamanan VPC

Saat Anda menyediakan klaster Amazon Redshift atau workgroup Amazon Redshift Tanpa Server, akses dibatasi secara default sehingga tidak ada yang memiliki akses ke sana. Untuk memberikan akses masuk kepada pengguna lain, Anda mengaitkannya dengan grup keamanan. Jika Anda menggunakan platform EC2-VPC, Anda dapat menggunakan grup keamanan Amazon VPC yang ada atau menentukan yang baru. Anda kemudian mengaitkannya dengan cluster atau workgroup seperti yang dijelaskan berikut. Jika Anda menggunakan platform EC2-Classic, Anda menentukan grup keamanan dan mengaitkannya dengan cluster atau workgroup Anda. Untuk informasi selengkapnya tentang penggunaan grup keamanan di platform EC2-Classic, lihat. [Grup keamanan Amazon Redshift](#)

Grup keamanan VPC terdiri dari seperangkat aturan yang mengontrol akses ke instance di VPC, seperti cluster Anda. Aturan individu menetapkan akses berdasarkan rentang alamat IP atau pada grup keamanan VPC lainnya. Saat Anda mengaitkan grup keamanan VPC dengan klaster atau grup kerja, aturan yang ditentukan dalam akses kontrol grup keamanan VPC.

Setiap cluster yang Anda sediakan di platform EC2-VPC memiliki satu atau lebih grup keamanan Amazon VPC yang terkait dengannya. Amazon VPC menyediakan grup keamanan VPC yang disebut default, yang dibuat secara otomatis saat Anda membuat VPC. Setiap cluster yang Anda luncurkan di VPC secara otomatis dikaitkan dengan grup keamanan VPC default jika Anda tidak menentukan grup keamanan VPC yang berbeda saat sumber daya Redshift Anda. Anda dapat mengaitkan grup keamanan VPC dengan klaster saat membuat klaster, atau Anda dapat mengaitkan grup keamanan VPC nanti dengan memodifikasi klaster.

Tangkapan layar berikut menunjukkan aturan default untuk grup keamanan VPC default.

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

Anda dapat mengubah aturan untuk grup keamanan VPC default sesuai kebutuhan.

Jika grup keamanan VPC default sudah cukup untuk Anda, Anda tidak perlu membuat lebih banyak. Namun, Anda dapat secara opsional membuat grup keamanan VPC tambahan untuk mengelola akses masuk dengan lebih baik. Misalnya, Anda menjalankan layanan di klaster Amazon Redshift atau grup kerja Tanpa Server, dan Anda memiliki beberapa tingkat layanan berbeda yang Anda berikan kepada pelanggan Anda. Jika Anda tidak ingin menyediakan akses yang sama di semua tingkat layanan, Anda mungkin ingin membuat grup keamanan VPC terpisah, satu untuk setiap tingkat layanan. Anda kemudian dapat mengaitkan grup keamanan VPC ini dengan klaster atau grup kerja Anda.

Anda dapat membuat hingga 100 grup keamanan VPC untuk VPC dan mengaitkan grup keamanan VPC dengan beberapa cluster dan grup kerja. Namun, perhatikan bahwa ada batasan jumlah grup keamanan VPC yang dapat Anda kaitkan dengan cluster atau workgroup.

Amazon Redshift segera menerapkan perubahan ke grup keamanan VPC. Jadi, jika Anda telah mengaitkan grup keamanan VPC dengan klaster, aturan akses cluster masuk dalam grup keamanan VPC yang diperbarui segera berlaku.

Anda dapat membuat dan memodifikasi grup keamanan VPC di <https://console.aws.amazon.com/vpc/> Anda juga dapat mengelola grup keamanan VPC secara terprogram dengan menggunakan, AWS CLI Amazon EC2 CLI, dan file. AWS Tools for Windows PowerShell Untuk informasi selengkapnya tentang bekerja dengan grup keamanan VPC, lihat [Grup keamanan untuk VPC Anda di Panduan Pengguna](#) Amazon VPC.

Mengonfigurasi setelan komunikasi grup keamanan untuk klaster Amazon Redshift atau grup kerja Amazon Redshift Tanpa Server

Topik ini membantu Anda mengonfigurasi grup keamanan untuk merutekan dan menerima lalu lintas jaringan dengan tepat. Berikut ini adalah beberapa kasus penggunaan umum:

- Anda mengaktifkan aksesibilitas publik untuk klaster Amazon Redshift atau grup kerja Amazon Redshift Tanpa Server, tetapi tidak menerima lalu lintas. Untuk ini, Anda harus mengonfigurasi aturan masuk untuk memungkinkan lalu lintas mencapainya dari internet.
- Cluster atau grup kerja Anda tidak dapat diakses publik, dan Anda menggunakan grup keamanan VPC default Redshift yang telah dikonfigurasi sebelumnya untuk mengizinkan lalu lintas masuk. Tetapi Anda memiliki persyaratan untuk menggunakan grup keamanan selain default, dan grup keamanan khusus ini tidak mengizinkan lalu lintas masuk. Anda harus mengkonfigurasinya untuk memungkinkan komunikasi.

Bagian berikut membantu Anda memilih respons yang benar untuk setiap kasus penggunaan dan menunjukkan cara mengonfigurasi lalu lintas jaringan sesuai kebutuhan Anda. Anda dapat secara opsional menggunakan langkah-langkah untuk mengatur komunikasi dari grup keamanan swasta lainnya.

Note

Pengaturan lalu lintas jaringan dalam banyak kasus tidak dikonfigurasi secara otomatis di Amazon Redshift. Ini karena mereka dapat bervariasi pada tingkat granular, tergantung pada apakah sumber lalu lintas adalah internet atau grup keamanan swasta, dan karena persyaratan keamanan bervariasi.

Aksesibilitas publik dengan konfigurasi grup keamanan default atau kustom

Jika Anda membuat atau Anda sudah memiliki cluster atau workgroup, lakukan langkah-langkah konfigurasi berikut untuk membuatnya dapat diakses publik. Ini berlaku baik ketika Anda memilih grup keamanan default atau grup keamanan khusus:

1. Temukan pengaturan jaringan:
 - Untuk klaster Amazon Redshift yang disediakan, pilih tab Properties, lalu di bawah Pengaturan jaringan dan keamanan, pilih VPC untuk klaster Anda.
 - Untuk grup kerja Amazon Redshift Tanpa Server, pilih konfigurasi Workgroup. Pilih workgroup dari daftar. Kemudian, di bawah Akses data, di panel Jaringan dan keamanan, pilih edit.
2. Konfigurasi gateway Internet dan tabel rute untuk VPC Anda. Anda memulai konfigurasi dengan memilih VPC berdasarkan nama. Ini membuka dasbor VPC. Untuk terhubung ke cluster atau workgroup yang dapat diakses publik dari internet, gateway internet harus dilampirkan

ke tabel rute. Anda dapat mengonfigurasinya dengan memilih tabel Route di dasbor VPC. Konfirmasikan bahwa target gateway internet ditetapkan dengan sumber 0.0.0.0/0 atau IP CIDR publik. Tabel rute harus dikaitkan dengan VPC tempat klaster Anda berada. Untuk informasi selengkapnya mengenai pengaturan akses internet untuk VPC, seperti yang dijelaskan di sini, lihat [Mengaktifkan akses internet di dokumentasi](#) Amazon VPC. Untuk informasi selengkapnya tentang mengonfigurasi tabel rute, lihat [Mengonfigurasi tabel rute](#).

3. Setelah Anda mengkonfigurasi gateway internet dan tabel rute, kembali ke pengaturan jaringan untuk Redshift. Buka akses masuk dengan memilih grup keamanan dan kemudian memilih aturan Inbound. Pilih Edit aturan masuk.
4. Pilih Protokol dan Port untuk aturan masuk, atau aturan, sesuai kebutuhan Anda, untuk mengizinkan lalu lintas dari klien. Untuk RA3 cluster, pilih port dalam rentang 5431-5455 atau 8191-8215. Setelah selesai, simpan setiap aturan.
5. Edit pengaturan yang dapat diakses publik untuk mengaktifkannya. Anda dapat melakukan ini dari menu Tindakan klaster atau grup kerja Anda.

Saat Anda mengaktifkan pengaturan yang dapat diakses publik, Redshift membuat alamat IP Elastis. Ini adalah alamat IP statis yang terkait dengan AWS akun Anda. Klien di luar VPC dapat menggunakannya untuk terhubung.

Untuk informasi selengkapnya tentang mengonfigurasi grup keamanan Anda, lihat [Grup keamanan Amazon Redshift](#).

Anda dapat menguji aturan Anda dengan menghubungkan dengan klien, lakukan hal berikut jika Anda terhubung ke Amazon Redshift Tanpa Server. Setelah Anda menyelesaikan konfigurasi jaringan, sambungkan dengan alat klien Anda, seperti [Amazon Redshift](#) RSQL. Menggunakan domain Amazon Redshift Tanpa Server sebagai host, masukkan yang berikut ini:

```
rsql -h workgroup-name.account-id.region.amazonaws.com -U admin -d dev -p 5439
```

Aksesibilitas pribadi dengan konfigurasi grup keamanan default atau kustom

Ketika Anda tidak berkomunikasi melalui internet ke cluster atau kelompok kerja Anda, itu disebut sebagai dapat diakses secara pribadi. Jika Anda memilih grup keamanan default saat membuatnya, grup keamanan menyertakan aturan komunikasi default berikut:

- Aturan masuk yang memungkinkan lalu lintas dari semua sumber daya yang ditetapkan ke grup keamanan.

- Aturan keluar yang memungkinkan semua lalu lintas keluar. Tujuan untuk aturan ini adalah 0.0.0.0/0. Dalam notasi routing antar domain tanpa kelas (CIDR), ini mewakili semua alamat IP yang mungkin.

Anda dapat melihat aturan di konsol dengan memilih grup keamanan untuk klaster atau grup kerja Anda.

Jika klaster atau grup kerja dan klien Anda menggunakan grup keamanan default, tidak ada konfigurasi tambahan yang diperlukan untuk mengizinkan lalu lintas jaringan. Tetapi jika Anda menghapus atau mengubah aturan apa pun di grup keamanan default untuk Redshift atau klien, ini tidak lagi berlaku. Dalam hal ini, Anda harus mengonfigurasi aturan untuk memungkinkan komunikasi masuk dan keluar. Konfigurasi grup keamanan umum adalah sebagai berikut:

- Untuk instans Amazon EC2 klien:
 - Aturan inbound yang memungkinkan alamat IP klien.
 - Aturan keluar yang memungkinkan rentang alamat IP (blok CIDR) dari semua subnet yang disediakan untuk penggunaan Redshift. Atau Anda dapat menentukan 0.0.0.0/0, yang merupakan semua rentang alamat IP.
- Untuk klaster atau grup kerja Redshift Anda:
 - Aturan masuk yang memungkinkan grup keamanan klien.
 - Aturan keluar yang memungkinkan lalu lintas ke 0.0.0.0/0. Biasanya, aturan keluar memungkinkan semua lalu lintas keluar. Secara opsional, Anda dapat menambahkan aturan keluar untuk mengizinkan lalu lintas ke grup keamanan klien. Dalam kasus opsional ini, aturan keluar tidak selalu diperlukan, karena lalu lintas respons untuk setiap permintaan diizinkan untuk mencapai instance. Untuk detail selengkapnya mengenai perilaku permintaan dan respons, lihat [Grup keamanan](#) di panduan pengguna Amazon VPC.

Jika Anda mengubah konfigurasi untuk subnet atau grup keamanan yang ditentukan untuk penggunaan Redshift, Anda mungkin perlu mengubah aturan lalu lintas yang sesuai untuk menjaga komunikasi tetap terbuka. Untuk informasi selengkapnya tentang membuat aturan masuk dan keluar, lihat [pemblokiran CIDR VPC di](#) panduan pengguna Amazon VPC. Untuk informasi selengkapnya tentang menghubungkan ke Amazon Redshift dari klien, lihat [Mengonfigurasi koneksi di Amazon Redshift](#).

Subnet untuk sumber daya Redshift

Anda membuat grup subnet jika Anda membuat klaster yang disediakan di cloud pribadi virtual (VPC). Setiap VPC dapat memiliki satu atau lebih subnet, yang merupakan himpunan bagian dari alamat IP dalam VPC yang memungkinkan Anda mengelompokkan sumber daya berdasarkan kebutuhan keamanan dan operasi Anda. Anda membuat grup subnet untuk menentukan satu set subnet di VPC Anda saat Anda membuat klaster yang disediakan. Di dasbor klaster yang disediakan, Anda dapat menemukan dan mengedit grup subnet cluster di bawah Konfigurasi. Selama konfigurasi awal untuk klaster yang disediakan, Anda menentukan grup subnet dan Amazon Redshift membuat cluster di salah satu subnetnya. Untuk informasi selengkapnya tentang layanan VPC, lihat halaman detail produk Amazon [VPC](#).

Konfigurasi subnet untuk workgroup Amazon Redshift Serverless mirip dengan cluster yang disediakan, tetapi langkah-langkahnya sedikit berbeda. Saat Anda membuat dan menyiapkan grup kerja Tanpa Server, Anda menentukan subnet untuk grup kerja, dan mereka ditambahkan ke daftar. Anda dapat melihat subnet untuk workgroup yang ada dengan memilih properti workgroup, di dasbor Tanpa Server. Mereka tersedia di Jaringan dan properti keamanan. Untuk informasi selengkapnya, lihat [Membuat workgroup dengan namespace](#).

Untuk informasi selengkapnya tentang membuat VPC, buka dokumentasi Panduan Pengguna Amazon [VPC](#).

Setelah membuat grup subnet untuk klaster yang disediakan, atau memilih subnet untuk grup kerja Tanpa Server, dimungkinkan untuk menghapus subnet yang sebelumnya ditambahkan atau menambahkan lebih banyak. Anda dapat membuat perubahan ini menggunakan konsol, atau menggunakan operasi API. Untuk informasi selengkapnya mengenai operasi API untuk klaster yang disediakan, lihat [ModifyClusterSubnetGroup](#) Untuk operasi API untuk workgroup Tanpa Server, lihat [UpdateWorkgroup](#)

Anda dapat menyediakan cluster di salah satu subnet di grup subnet. Grup subnet cluster memungkinkan Anda menentukan satu set subnet di virtual private cloud (VPC) Anda.

Warning

Selama operasi pemeliharaan klaster seperti pengubahan ukuran klasik, jeda dan lanjutkan, failover multi-AZ, atau peristiwa lainnya, node komputasi yang disediakan dapat dipindahkan ke subnet lain dalam grup subnet klaster Amazon Redshift Anda. Perhatikan bahwa semua subnet dalam grup subnet harus memiliki aturan masuk dan keluar ACL Jaringan yang

sama dan rute tabel rute yang sama. Ini memastikan konektivitas ke dan dari sumber daya komputasi Amazon Redshift, sehingga mereka dapat berkomunikasi dan berfungsi secara optimal setelah peristiwa pemeliharaan tersebut. Hindari menambahkan subnet dengan berbagai konfigurasi ACL jaringan atau tabel rute ke grup subnet klaster Amazon Redshift yang sama.

Untuk informasi selengkapnya tentang mengonfigurasi subnet, lihat [Subnet untuk VPC Anda di panduan pengguna Amazon VPC](#). Untuk informasi selengkapnya tentang penerapan Redshift Multi-AZ, lihat di panduan manajemen Redshift [Deployment multi-AZ](#). [Mengubah ukuran cluster](#) juga tercakup dalam panduan manajemen Redshift.

Membuat grup subnet cluster

Prosedur berikut memandu Anda melalui cara membuat grup subnet untuk klaster yang disediakan. Anda harus memiliki setidaknya satu grup subnet cluster yang ditentukan untuk menyediakan klaster di VPC.

Untuk membuat grup subnet cluster untuk klaster yang disediakan

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Konfigurasi, lalu pilih Grup subnet. Daftar grup subnet ditampilkan.
3. Pilih Buat grup subnet cluster untuk menampilkan halaman buat.
4. Masukkan informasi untuk grup subnet, termasuk subnet mana yang akan ditambahkan.
5. Pilih Buat grup subnet cluster untuk membuat grup dengan subnet yang Anda pilih.

Note

Untuk informasi tentang cara membuat workgroup Amazon Redshift Tanpa Server dengan kumpulan subnet, lihat [Membuat grup kerja dengan namespace atau Membuat subnet di Panduan Pengguna Amazon VPC](#).

Memodifikasi grup subnet cluster

Setelah membuat grup subnet, Anda dapat memodifikasi informasinya di konsol Amazon Redshift. Prosedur berikut memandu Anda melalui cara memodifikasi grup subnet untuk klaster yang disediakan.

Untuk memodifikasi grup subnet klaster untuk klaster yang disediakan

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Konfigurasi, lalu pilih Grup subnet. Daftar grup subnet ditampilkan.
3. Pilih grup subnet untuk dimodifikasi.
4. Untuk Tindakan, pilih Ubah untuk menampilkan detail grup subnet.
5. Perbarui informasi untuk grup subnet.
6. Pilih Simpan untuk memodifikasi grup.

Untuk mengubah atau menghapus subnet dalam beberapa kasus memerlukan langkah tambahan. Misalnya, artikel Pusat AWS Pengetahuan ini, [Bagaimana cara memindahkan cluster Amazon Redshift saya yang disediakan ke subnet yang berbeda?](#), menjelaskan kasus penggunaan yang mencakup pemindahan cluster.

Menghapus grup subnet cluster untuk klaster yang disediakan

Setelah selesai menggunakan grup subnet cluster, Anda harus membersihkan dengan menghapus grup. Prosedur berikut memandu Anda melalui langkah-langkah untuk menghapus grup subnet untuk klaster yang disediakan.

Untuk menghapus grup subnet cluster

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Konfigurasi, lalu pilih Grup subnet. Daftar grup subnet ditampilkan.
3. Pilih grup subnet yang akan dihapus, lalu pilih Hapus.

 Note

Anda tidak dapat menghapus grup subnet cluster yang digunakan oleh cluster.

Memblokir akses publik ke VPCs dan subnet

VPC Block Public Access (BPA) adalah fitur keamanan terpusat yang dapat Anda gunakan untuk memblokir sumber daya di VPCs dan subnet yang Anda miliki di internet agar tidak menjangkau internet atau dijangkau AWS Region dari internet melalui gateway internet dan gateway internet khusus egres. Jika Anda mengaktifkan fitur ini secara default Akun AWS, fitur ini akan berdampak pada VPC atau subnet apa pun yang digunakan Amazon Redshift. Ini berarti Amazon Redshift memblokir semua operasi ke publik.

Ketika Anda mengaktifkan VPC BPA dan ingin menggunakan Amazon Redshift API melalui internet publik, Anda harus menambahkan pengecualian untuk menggunakan Amazon APIs EC2 untuk VPC atau subnet Anda. Pengecualian dapat memiliki salah satu dari mode berikut:

- Dua arah: Semua lalu lintas internet ke dan dari yang dikecualikan VPCs dan subnet diperbolehkan.
- Egress-only: Lalu lintas internet keluar dari yang dikecualikan VPCs dan subnet diizinkan. Lalu lintas internet masuk ke yang dikecualikan VPCs dan subnet diblokir. Ini hanya berlaku ketika BPA diatur ke dua arah.

Pengecualian VPC BPA menetapkan seluruh VPC atau subnet tertentu dalam VPC sebagai kemampuan akses publik. Antarmuka jaringan dalam batas tersebut menghormati kontrol jaringan VPC reguler, seperti grup keamanan, tabel rute, dan jaringan ACLs, sehubungan dengan apakah antarmuka tersebut memiliki rute dan akses ke internet publik. Untuk informasi selengkapnya tentang menambahkan pengecualian, lihat [Membuat dan menghapus pengecualian di Panduan Pengguna Amazon VPC](#).

Cluster yang disediakan

Grup subnet adalah kombinasi subnet dari VPC yang sama. Jika grup subnet untuk klaster yang disediakan berada di akun dengan VPC BPA diaktifkan, kemampuan berikut akan diblokir:

- Membuat klaster publik
- Memulihkan klaster publik

- Memodifikasi klaster pribadi menjadi publik
- Menambahkan subnet dengan VPC BPA diaktifkan ke grup subnet ketika setidaknya ada satu cluster publik dalam grup

Cluster tanpa server

Redshift Serverless tidak menggunakan grup subnet. Sebaliknya, setiap cluster memiliki set subnet sendiri. Jika grup kerja ada di akun dengan VPC BPA dihidupkan, kemampuan berikut diblokir:

- Membuat workgroup akses publik
- Memodifikasi grup kerja pribadi menjadi publik
- Menambahkan subnet dengan VPC BPA dihidupkan ke workgroup saat workgroup bersifat publik

Mengontrol lalu lintas jaringan dengan perutean VPC Redshift yang ditingkatkan

Saat Anda menggunakan perutean VPC Amazon Redshift yang disempurnakan, Amazon Redshift memaksa [semua](#) lalu lintas COPY [dan](#) UNLOAD antara cluster dan repositori data Anda melalui virtual private cloud (VPC) berdasarkan layanan Amazon VPC. Dengan menggunakan perutean VPC yang disempurnakan, Anda dapat menggunakan fitur VPC standar, seperti [grup keamanan VPC](#), [daftar kontrol akses jaringan \(\)](#), [titik akhir VPC](#), [kebijakan titik akhir ACLs VPC](#), [gateway internet](#), dan [server Sistem Nama Domain \(DNS\)](#), seperti yang dijelaskan dalam Panduan Pengguna Amazon VPC. Anda menggunakan fitur ini untuk mengontrol aliran data antara cluster Amazon Redshift dan sumber daya lainnya. Saat Anda menggunakan perutean VPC yang disempurnakan untuk merutekan lalu lintas melalui VPC Anda, Anda juga dapat menggunakan [log aliran VPC](#) untuk memantau lalu lintas COPY dan UNLOAD.

Cluster Amazon Redshift dan workgroup Amazon Redshift Serverless keduanya mendukung peningkatan perutean VPC. Anda tidak dapat menggunakan perutean VPC yang disempurnakan dengan Redshift Spectrum. Untuk informasi selengkapnya, lihat [Mengakses bucket Amazon S3 dengan Redshift Spectrum](#).

Jika perutean VPC yang disempurnakan tidak diaktifkan, Amazon Redshift merutekan lalu lintas melalui internet, termasuk lalu lintas ke layanan lain di dalam jaringan. AWS

⚠ Important

Karena perutean VPC yang disempurnakan memengaruhi cara Amazon Redshift mengakses sumber daya lain, perintah COPY dan UNLOAD mungkin gagal kecuali Anda mengonfigurasi VPC dengan benar. Anda harus secara khusus membuat jalur jaringan antara VPC klaster Anda dan sumber daya data Anda, seperti yang dijelaskan berikut.

Saat Anda menjalankan perintah COPY atau UNLOAD di klaster dengan perutean VPC yang ditingkatkan diaktifkan, VPC Anda merutekan lalu lintas ke sumber daya yang ditentukan menggunakan jalur jaringan paling ketat, atau paling spesifik, yang tersedia.

Misalnya, Anda dapat mengonfigurasi jalur berikut di VPC Anda:

- Titik akhir VPC — Untuk lalu lintas ke bucket Amazon S3 di AWS Wilayah yang sama dengan klaster atau grup kerja, Anda dapat membuat titik akhir VPC untuk mengarahkan lalu lintas langsung ke bucket. Saat menggunakan titik akhir VPC, Anda dapat melampirkan kebijakan titik akhir untuk mengelola akses ke Amazon S3. Untuk informasi selengkapnya tentang penggunaan titik akhir dengan Redshift, lihat [Mengontrol lalu lintas database dengan titik akhir VPC](#). Jika Anda menggunakan Lake Formation, Anda dapat menemukan informasi lebih lanjut tentang membuat koneksi pribadi antara VPC dan AWS Lake Formation di [AWS Lake Formation dan antarmuka VPC endpoint](#) ().AWS PrivateLink

i Note

Saat Anda menggunakan titik akhir VPC Redshift dengan titik akhir Amazon S3 VPC Gateway, Anda harus mengaktifkan perutean VPC yang disempurnakan di Redshift. Untuk informasi selengkapnya, lihat [Titik akhir gateway untuk Amazon S3](#).

- Gateway NAT - Anda dapat terhubung ke bucket Amazon S3 di Wilayah AWS lain, dan Anda dapat terhubung ke layanan lain dalam AWS jaringan. Anda juga dapat mengakses instance host di luar AWS jaringan. Untuk melakukannya, konfigurasi [gateway terjemahan alamat jaringan \(NAT\)](#), seperti yang dijelaskan dalam Panduan Pengguna Amazon VPC.
- Internet gateway — Untuk terhubung ke AWS layanan di luar VPC Anda, Anda dapat melampirkan [gateway internet](#) ke subnet VPC Anda, seperti yang dijelaskan dalam Panduan Pengguna Amazon VPC. Untuk menggunakan gateway internet, cluster atau workgroup Anda harus dapat diakses publik untuk memungkinkan layanan lain mengkomunikasikannya.

Untuk informasi selengkapnya, lihat [Titik Akhir VPC di Panduan Pengguna Amazon VPC](#).

Tidak ada biaya tambahan untuk menggunakan perutean VPC yang disempurnakan. Anda mungkin dikenakan biaya transfer data tambahan untuk operasi tertentu. Ini termasuk operasi seperti UNLOAD ke Amazon S3 di Wilayah yang AWS berbeda. SALIN dari Amazon EMR, atau Secure Shell (SSH) dengan alamat IP publik. Untuk informasi selengkapnya tentang harga, lihat [Harga Amazon EC2](#).

Topik

- [Mengontrol lalu lintas database dengan titik akhir VPC](#)
- [Mengaktifkan perutean VPC yang disempurnakan](#)
- [Mengakses bucket Amazon S3 dengan Redshift Spectrum](#)

Mengontrol lalu lintas database dengan titik akhir VPC

Anda dapat menggunakan titik akhir VPC untuk membuat koneksi terkelola antara cluster Amazon Redshift atau workgroup Tanpa Server di VPC dan Amazon Simple Storage Service (Amazon S3). Ketika Anda melakukannya, COPY dan BONGKAR lalu lintas antara database Anda dan data Anda di Amazon S3 tetap berada di Amazon VPC Anda. Anda dapat melampirkan kebijakan titik akhir ke titik akhir Anda untuk mengelola akses ke data Anda dengan lebih dekat. Misalnya, Anda dapat menambahkan kebijakan ke titik akhir VPC yang mengizinkan pembongkaran data hanya ke bucket Amazon S3 tertentu di akun Anda.

Untuk menggunakan titik akhir VPC, buat titik akhir VPC untuk VPC tempat gudang data Anda berada, lalu aktifkan perutean VPC yang disempurnakan. Anda dapat mengaktifkan perutean VPC yang disempurnakan saat membuat klaster atau grup kerja, atau Anda dapat memodifikasi klaster atau grup kerja di VPC untuk menggunakan perutean VPC yang disempurnakan.

Titik akhir VPC menggunakan tabel rute untuk mengontrol perutean lalu lintas antara cluster atau workgroup di VPC dan Amazon S3. Semua cluster dan workgroup dalam subnet yang terkait dengan tabel rute yang ditentukan secara otomatis menggunakan endpoint tersebut untuk mengakses layanan.

VPC Anda menggunakan rute paling spesifik, atau paling ketat, yang cocok dengan lalu lintas Anda untuk menentukan cara merutekan lalu lintas. Misalnya, Anda memiliki rute di tabel rute Anda untuk semua lalu lintas internet (0.0.0.0/0) yang menunjuk ke gateway internet dan titik akhir Amazon S3. Dalam hal ini, rute titik akhir diutamakan untuk semua lalu lintas yang ditujukan untuk Amazon S3. Ini karena rentang alamat IP untuk layanan Amazon S3 lebih spesifik daripada 0.0.0.0/0. Dalam contoh

ini, semua lalu lintas internet lainnya masuk ke gateway internet Anda, termasuk lalu lintas yang ditujukan untuk ember Amazon S3 di tempat lain. Wilayah AWS

Untuk informasi selengkapnya tentang membuat titik akhir, lihat [Membuat titik akhir VPC](#) di Panduan Pengguna Amazon VPC.

Anda menggunakan kebijakan titik akhir untuk mengontrol akses dari kluster atau grup kerja ke bucket Amazon S3 yang menyimpan file data Anda. Untuk kontrol yang lebih spesifik, Anda dapat melampirkan kebijakan endpoint kustom secara opsional. Untuk informasi selengkapnya, lihat [Mengontrol akses ke layanan menggunakan kebijakan titik akhir](#) di Panduan AWS PrivateLink .

Note

AWS Database Migration Service (AWS DMS) adalah layanan cloud yang memungkinkan untuk memigrasikan database relasional, gudang data, dan jenis penyimpanan data lainnya. Ini dapat terhubung ke AWS sumber atau basis data target apa pun, termasuk database Amazon Redshift yang diaktifkan VPC, dengan beberapa batasan konfigurasi. Mendukung titik akhir Amazon VPC memudahkan untuk menjaga keamanan end-to-end jaringan AWS DMS untuk tugas replikasi. Untuk informasi selengkapnya tentang penggunaan Redshift with AWS DMS, lihat [Mengonfigurasi titik akhir VPC AWS DMS sebagai sumber dan titik akhir target](#) di Panduan Pengguna AWS Database Migration Service

Tidak ada biaya tambahan untuk menggunakan titik akhir. Biaya standar untuk transfer data dan penggunaan sumber daya berlaku. Untuk informasi selengkapnya tentang harga, lihat Harga [Amazon EC2](#).

Mengaktifkan perutean VPC yang disempurnakan

Anda dapat mengaktifkan perutean VPC yang disempurnakan saat membuat atau memodifikasi kluster, dan saat Anda membuat atau memodifikasi grup kerja Amazon Redshift Tanpa Server.

Untuk bekerja dengan perutean VPC yang disempurnakan, kluster atau grup kerja Tanpa Server Anda harus memenuhi persyaratan dan kendala berikut:

- Cluster Anda harus dalam VPC.

Jika Anda melampirkan titik akhir VPC Amazon S3, titik akhir VPC hanya digunakan untuk akses ke bucket Amazon S3 di Wilayah yang sama. AWS [Untuk mengakses bucket di AWS Wilayah lain](#)

(tidak menggunakan titik akhir VPC) atau untuk mengakses layanan AWS lain, buat klaster atau grup kerja Tanpa Server Anda dapat diakses publik atau gunakan gateway terjemahan alamat jaringan (NAT). Untuk informasi selengkapnya, lihat [Membuat klaster yang disediakan Redshift atau workgroup Amazon Redshift Serverless di VPC](#).

- Anda harus mengaktifkan resolusi Domain Name Service (DNS) di VPC Anda. Atau, jika Anda menggunakan server DNS Anda sendiri, pastikan bahwa permintaan DNS ke Amazon S3 diselesaikan dengan benar ke alamat IP yang dikelola oleh AWS. Untuk informasi lebih lanjut, lihat [Menggunakan DNS dengan VPC](#) di Panduan Pengguna Amazon VPC.
- Nama host DNS harus diaktifkan di VPC Anda. Nama host DNS diaktifkan secara default.
- Kebijakan titik akhir VPC Anda harus mengizinkan akses ke bucket Amazon S3 apa pun yang digunakan dengan panggilan COPY, UNLOAD, atau CREATE LIBRARY di Amazon Redshift, termasuk akses ke file manifes apa pun yang terlibat. Untuk COPY dari host jarak jauh, kebijakan endpoint Anda harus mengizinkan akses ke setiap mesin host. Untuk informasi selengkapnya, lihat [Izin IAM untuk COPY, UNLOAD, dan CREATE LIBRARY di Panduan Pengembang Database Amazon Redshift](#).

Untuk mengaktifkan perutean VPC yang disempurnakan untuk klaster yang disediakan

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Dasbor cluster yang disediakan, lalu pilih Buat klaster dan masukkan properti detail Cluster.
3. Untuk menampilkan bagian Konfigurasi tambahan, pilih untuk menonaktifkan Gunakan default.
4. Arahkan ke bagian Jaringan dan keamanan.
5. Untuk mengaktifkan perutean VPC yang Ditingkatkan, pilih Aktifkan untuk memaksa lalu lintas cluster melalui VPC.
6. Pilih Buat cluster untuk membuat cluster. Cluster mungkin membutuhkan waktu beberapa menit untuk siap digunakan.

Untuk mengaktifkan perutean VPC yang disempurnakan untuk Amazon Redshift Tanpa Server

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Dasbor tanpa server, lalu pilih Buat grup kerja dan masukkan properti untuk grup kerja Anda.

3. Arahkan ke bagian Jaringan dan keamanan.
4. Pilih Aktifkan perutean VPC yang disempurnakan untuk merutekan lalu lintas jaringan melalui VPC.
5. Pilih Berikutnya dan selesai memasukkan properti workgroup Anda sampai Anda membuat workgroup.

Mengakses bucket Amazon S3 dengan Redshift Spectrum

Secara umum, Amazon Redshift Spectrum tidak mendukung perutean VPC yang disempurnakan dengan kluster yang disediakan, meskipun kluster yang disediakan dapat menanyakan tabel eksternal dari Amazon S3 saat perutean VPC yang disempurnakan diaktifkan.

Perutean VPC Amazon Redshift yang disempurnakan mengirimkan lalu lintas tertentu melalui VPC Anda, yang berarti bahwa semua lalu lintas antara cluster Anda dan bucket Amazon S3 Anda terpaksa melewati VPC Amazon Anda. Karena Redshift Spectrum berjalan pada sumber daya AWS terkelola yang dimiliki oleh Amazon Redshift tetapi berada di luar VPC Anda, Redshift Spectrum tidak menggunakan perutean VPC yang disempurnakan.

Lalu lintas antara Redshift Spectrum dan Amazon S3 dirutekan dengan aman melalui jaringan pribadi, AWS di luar VPC Anda. Lalu lintas dalam penerbangan ditandatangani menggunakan protokol Amazon Signature Version 4 (SIGv4) dan dienkripsi menggunakan HTTPS. Lalu lintas ini diotorisasi berdasarkan peran IAM yang dilampirkan ke cluster Amazon Redshift Anda. Untuk mengelola lalu lintas Redshift Spectrum lebih lanjut, Anda dapat memodifikasi peran IAM klaster dan kebijakan yang dilampirkan ke bucket Amazon S3. Anda mungkin juga perlu mengonfigurasi VPC Anda untuk memungkinkan klaster Anda mengakses atau AWS Glue Athena, seperti yang dijelaskan berikut ini.

Perhatikan bahwa karena perutean VPC yang disempurnakan memengaruhi cara Amazon Redshift mengakses sumber daya lain, kueri mungkin gagal kecuali Anda mengonfigurasi VPC dengan benar. Untuk informasi selengkapnya, lihat [Mengontrol lalu lintas jaringan dengan perutean VPC Redshift yang ditingkatkan](#), yang membahas secara lebih rinci membuat titik akhir VPC, gateway NAT, dan sumber daya jaringan lainnya untuk mengarahkan lalu lintas ke bucket Amazon S3 Anda.

 Note

Amazon Redshift Serverless mendukung perutean VPC yang disempurnakan untuk kueri ke tabel eksternal di Amazon S3. Untuk informasi selengkapnya tentang konfigurasi, lihat [Memuat data dari Amazon S3 di Panduan Memulai Tanpa Server Amazon Redshift](#).

Konfigurasi kebijakan izin saat menggunakan Amazon Redshift Spectrum

Pertimbangkan hal berikut saat menggunakan Redshift Spectrum:

- [Kebijakan akses bucket Amazon S3 dan peran IAM](#)
- [Izin untuk mengasumsikan peran IAM](#)
- [Pencatatan dan audit akses Amazon S3](#)
- [Akses ke AWS Glue atau Amazon Athena](#)

Kebijakan akses bucket Amazon S3 dan peran IAM

Anda dapat mengontrol akses ke data di bucket Amazon S3 dengan menggunakan kebijakan bucket yang dilampirkan ke bucket dan dengan menggunakan peran IAM yang dilampirkan ke kluster yang disediakan.

Redshift Spectrum pada kluster yang disediakan tidak dapat mengakses data yang disimpan di bucket Amazon S3 yang menggunakan kebijakan bucket yang membatasi akses hanya ke titik akhir VPC yang ditentukan. Sebagai gantinya, gunakan kebijakan bucket yang membatasi akses hanya ke prinsipal tertentu, seperti AWS akun tertentu atau pengguna tertentu.

Untuk peran IAM yang diberikan akses ke bucket, gunakan hubungan kepercayaan yang memungkinkan peran hanya diasumsikan oleh prinsipal layanan Amazon Redshift. Saat dilampirkan ke cluster Anda, peran hanya dapat digunakan dalam konteks Amazon Redshift dan tidak dapat dibagikan di luar cluster. Untuk informasi selengkapnya, lihat [Membatasi akses ke peran IAM](#). Kebijakan kontrol layanan (SCP) juga dapat digunakan untuk membatasi peran lebih lanjut, lihat [Mencegah pengguna IAM dan peran membuat perubahan tertentu, dengan pengecualian untuk peran admin tertentu dalam Panduan Pengguna AWS Organizations](#)

Note

Untuk menggunakan Redshift Spectrum, tidak ada kebijakan IAM yang memblokir penggunaan Amazon S3 URLs presigned yang dapat diterapkan. Presigned URLs yang dihasilkan oleh Amazon Redshift Spectrum berlaku selama 1 jam sehingga Amazon Redshift memiliki cukup waktu untuk memuat semua file dari bucket Amazon S3. URL presigned unik dibuat untuk setiap file yang dipindai oleh Redshift Spectrum. Untuk kebijakan bucket yang menyertakan `s3:signatureAge` tindakan, pastikan untuk menetapkan nilainya setidaknya 3.600.000 milidetik.

Contoh kebijakan bucket berikut mengizinkan akses ke bucket tertentu yang dimiliki oleh AWS akun123456789012.

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/*"
      ]
    }
  ]
}
```

Izin untuk mengasumsikan peran IAM

Peran yang dilampirkan ke klaster Anda harus memiliki hubungan kepercayaan yang memungkinkannya diasumsikan hanya oleh layanan Amazon Redshift, seperti yang ditunjukkan berikut.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "redshift.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Untuk informasi selengkapnya, lihat [Kebijakan IAM untuk Spektrum Pergeseran Merah](#) di Panduan Pengembang Database Amazon Redshift.

Pencatatan dan audit akses Amazon S3

Salah satu manfaat menggunakan perutean VPC Amazon Redshift yang disempurnakan adalah bahwa semua lalu lintas COPY dan UNLOAD dicatat di log aliran VPC. Lalu lintas yang berasal dari Redshift Spectrum ke Amazon S3 tidak melewati VPC Anda, jadi tidak masuk ke log aliran VPC. Saat Redshift Spectrum mengakses data di Amazon S3, ia melakukan operasi ini dalam konteks AWS akun dan hak istimewa peran masing-masing. Anda dapat mencatat dan mengaudit akses Amazon S3 menggunakan akses server masuk AWS CloudTrail dan Amazon S3.

Pastikan rentang IP S3 ditambahkan ke daftar izin Anda. Untuk mempelajari lebih lanjut tentang rentang IP S3 yang diperlukan, lihat [Isolasi jaringan](#).

AWS CloudTrail Log

Untuk melacak semua akses ke objek di Amazon S3, termasuk akses Redshift Spectrum, aktifkan pencatatan untuk objek CloudTrail Amazon S3.

Anda dapat menggunakan CloudTrail untuk melihat, mencari, mengunduh, mengarsipkan, menganalisis, dan menanggapi aktivitas akun di seluruh AWS infrastruktur Anda. Untuk informasi selengkapnya, lihat [Memulai dengan CloudTrail](#).

Secara default, hanya CloudTrail melacak tindakan tingkat ember. Untuk melacak tindakan tingkat objek (seperti `GetObject`), aktifkan data dan peristiwa manajemen untuk setiap bucket yang dicatat.

Pencatatan Log Akses Server Amazon S3

Pencatatan akses server menyediakan catatan terperinci untuk permintaan yang dilakukan ke bucket. Informasi log akses dapat berguna dalam audit keamanan dan akses. Untuk informasi selengkapnya, lihat [Cara Mengaktifkan Pencatatan Akses Server](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Untuk informasi selengkapnya, lihat posting blog AWS Keamanan [Cara Menggunakan Kebijakan Bucket dan Terapkan Defense-in-Depth untuk Membantu Mengamankan Data Amazon S3 Anda](#).

Akses ke AWS Glue atau Amazon Athena

Redshift Spectrum mengakses katalog data Anda di atau AWS Glue Athena. Pilihan lain adalah menggunakan metastore Hive khusus untuk katalog data Anda.

Untuk mengaktifkan akses ke AWS Glue atau Athena, konfigurasi VPC Anda dengan gateway internet atau gateway NAT. Konfigurasi grup keamanan VPC Anda untuk mengizinkan lalu lintas keluar ke titik akhir publik untuk dan Athena. AWS Glue Atau, Anda dapat mengonfigurasi titik akhir VPC antarmuka AWS Glue untuk mengakses. AWS Glue Data Catalog Saat Anda menggunakan titik akhir antarmuka VPC, komunikasi antara VPC Anda dan AWS Glue dilakukan di dalam jaringan. AWS Untuk informasi selengkapnya, lihat [Membuat Titik Akhir Antarmuka](#).

Anda dapat mengonfigurasi jalur berikut di VPC Anda:

- Internet gateway —Untuk terhubung ke AWS layanan di luar VPC Anda, Anda dapat melampirkan gateway [internet](#) ke subnet VPC Anda, seperti yang dijelaskan dalam Panduan Pengguna Amazon VPC. Untuk menggunakan gateway internet, cluster yang disediakan harus memiliki alamat IP publik untuk memungkinkan layanan lain berkomunikasi dengannya.
- Gateway NAT —Untuk menyambung ke bucket Amazon S3 di Wilayah AWS lain atau ke layanan lain dalam AWS jaringan, konfigurasi gateway [terjemahan alamat jaringan \(NAT\)](#), seperti yang dijelaskan dalam Panduan Pengguna Amazon VPC. Gunakan konfigurasi ini juga untuk mengakses instance host di luar AWS jaringan.

Lihat informasi yang lebih lengkap di [Mengontrol lalu lintas jaringan dengan perutean VPC Redshift yang ditingkatkan](#).

Acara Amazon Redshift

Amazon Redshift melacak peristiwa cluster dan menyimpan informasi tentangnya selama beberapa minggu di akun Anda. AWS Untuk setiap peristiwa, Amazon Redshift melaporkan informasi seperti tanggal peristiwa terjadi, deskripsi, sumber peristiwa (misalnya, cluster, grup parameter, atau snapshot), dan ID sumber.

Amazon Redshift memberikan pemberitahuan sebelumnya untuk beberapa acara. Peristiwa ini memiliki kategori *acarapending*. Misalnya, kami mengirim pemberitahuan terlebih dahulu jika pembaruan perangkat keras diperlukan untuk salah satu node di cluster Anda. Anda dapat berlangganan acara yang tertunda sama seperti acara Amazon Redshift lainnya. Untuk informasi selengkapnya, lihat [Langganan pemberitahuan acara klaster Amazon Redshift](#).

Anda dapat menggunakan Amazon Redshift Management Console, Amazon Redshift API, atau untuk mendapatkan informasi AWS SDKs peristiwa. Anda dapat memperoleh daftar semua acara, atau Anda dapat menerapkan filter, seperti durasi acara atau tanggal mulai dan berakhir, untuk mendapatkan informasi acara untuk periode tertentu.

Anda juga dapat memperoleh peristiwa yang dihasilkan oleh jenis sumber tertentu, seperti peristiwa cluster atau peristiwa grup parameter. Kolom Sumber menunjukkan nama sumber daya dan jenis sumber daya yang memicu tindakan yang diberikan.

Anda dapat membuat langganan notifikasi acara Amazon Redshift yang menentukan satu set filter peristiwa. Saat terjadi peristiwa yang sesuai dengan kriteria filter, Amazon Redshift menggunakan Layanan Pemberitahuan Sederhana Amazon untuk secara aktif memberi tahu Anda bahwa peristiwa tersebut telah terjadi.

Untuk daftar peristiwa Amazon Redshift menurut jenis dan kategori sumber, lihat [the section called “Pemberitahuan acara cluster yang disediakan”](#)

Langganan pemberitahuan acara klaster Amazon Redshift

Amazon Redshift menggunakan Amazon Simple Notification Service (Amazon SNS) untuk mengkomunikasikan notifikasi peristiwa Amazon Redshift. Anda mengaktifkan notifikasi dengan membuat langganan acara Amazon Redshift. Anda dapat diberi tahu saat peristiwa terjadi untuk klaster, snapshot, grup keamanan, atau grup parameter tertentu. Cara termudah untuk membuat langganan adalah dengan konsol Amazon SNS. Untuk informasi tentang membuat topik Amazon SNS dan berlangganan itu, lihat [Mulai dengan Amazon SNS](#).

Di langganan Amazon Redshift, Anda menentukan satu set filter untuk acara Amazon Redshift dan topik Amazon SNS. Setiap kali terjadi peristiwa yang cocok dengan kriteria filter, Amazon Redshift menerbitkan pesan notifikasi ke topik Amazon SNS.

Amazon SNS kemudian mengirimkan pesan ke konsumen Amazon SNS yang memiliki langganan Amazon SNS untuk topik tersebut. Pesan yang dikirim ke konsumen Amazon SNS dapat dalam bentuk apa pun yang didukung oleh Amazon SNS untuk AWS suatu Wilayah, seperti email, pesan teks, atau panggilan ke titik akhir HTTP. Misalnya, semua Wilayah mendukung pemberitahuan email, tetapi pemberitahuan SMS hanya dapat dibuat di Wilayah AS Timur (Virginia N.).

 Note

Saat ini, Anda hanya dapat membuat langganan acara ke topik standar Amazon SNS (bukan ke topik FIFO Amazon SNS). Untuk informasi selengkapnya, lihat [Sumber peristiwa Amazon SNS di Panduan](#) Pengembang Layanan Pemberitahuan Sederhana Amazon.

Saat Anda membuat langganan pemberitahuan acara, Anda menentukan satu atau beberapa filter acara. Amazon Redshift mengirimkan pemberitahuan melalui langganan setiap kali terjadi peristiwa yang cocok dengan semua kriteria filter. Kriteria filter mencakup jenis sumber (seperti cluster atau snapshot), ID sumber (seperti nama cluster atau snapshot), kategori peristiwa (seperti Monitoring atau Security), dan tingkat keparahan peristiwa (seperti INFO atau ERROR).

Jika Anda membuat langganan notifikasi acara menggunakan CLI atau API, Anda harus membuat topik Layanan Pemberitahuan Sederhana Amazon dan berlangganan topik tersebut dengan konsol Amazon SNS atau Amazon SNS API. Anda juga perlu mempertahankan Nama Sumber Daya Amazon (ARN) topik karena digunakan saat mengirimkan perintah CLI atau tindakan API.

Anda dapat dengan mudah mematikan notifikasi tanpa menghapus langganan dengan menyetel tombol radio yang diaktifkan ke No dalam Konsol Manajemen AWS atau dengan mengatur Enabled parameter untuk false menggunakan Amazon Redshift CLI atau API.

Langganan acara Amazon Redshift dapat menentukan kriteria acara ini:

- Jenis sumber, nilainya adalah cluster, snapshot, parameter-grup, dan grup keamanan.
- ID sumber daya, seperti `my-cluster-1` atau `my-snapshot-20130823`. ID harus untuk sumber daya di AWS Wilayah yang sama dengan langganan acara.
- Kategori acara, nilainya adalah Konfigurasi, Manajemen, Pemantauan, Keamanan, dan Pending

- Tingkat keparahan peristiwa, nilainya INFO atau ERROR.

Kriteria acara dapat ditentukan secara independen, kecuali bahwa Anda harus menentukan jenis sumber sebelum Anda dapat menentukan sumber IDs di konsol. Misalnya, Anda dapat menentukan kategori acara tanpa harus menentukan jenis sumber, ID sumber, atau tingkat keparahan. Meskipun Anda dapat menentukan sumber IDs untuk sumber daya yang bukan dari jenis yang ditentukan dalam tipe sumber, tidak ada pemberitahuan yang akan dikirim untuk acara dari sumber daya tersebut. Misalnya, jika Anda menentukan jenis sumber klaster dan ID grup keamanan, tidak ada peristiwa yang dimunculkan oleh grup keamanan tersebut yang akan cocok dengan kriteria filter tipe sumber, sehingga tidak ada pemberitahuan yang akan dikirim untuk peristiwa tersebut.

Amazon Redshift mengirimkan pemberitahuan untuk acara apa pun yang cocok dengan semua kriteria yang ditentukan dalam langganan. Beberapa contoh rangkaian acara yang dikembalikan:

- Langganan menentukan jenis sumber cluster, ID sumber my-cluster-1, kategori Monitoring, dan tingkat keparahan ERROR. Langganan akan mengirimkan pemberitahuan hanya untuk memantau peristiwa dengan tingkat keparahan ERROR dari my-cluster-1.
- Langganan menentukan jenis sumber cluster, kategori Konfigurasi, dan tingkat keparahan INFO. Langganan akan mengirimkan pemberitahuan untuk peristiwa konfigurasi dengan tingkat keparahan INFO dari klaster Amazon Redshift mana pun di AWS akun.
- Langganan menentukan kategori Konfigurasi, dan tingkat keparahan INFO. Langganan akan mengirimkan pemberitahuan untuk peristiwa konfigurasi dengan tingkat keparahan INFO dari sumber daya Amazon Redshift di AWS akun.
- Langganan menentukan tingkat keparahan ERROR. Langganan akan mengirim pemberitahuan untuk semua acara dengan tingkat keparahan ERROR dari sumber daya Amazon Redshift di AWS akun.

Jika Anda menghapus atau mengganti nama objek yang namanya direferensikan sebagai ID sumber dalam langganan yang ada, langganan akan tetap aktif, tetapi tidak akan memiliki acara untuk diteruskan dari objek tersebut. Jika nanti Anda membuat objek baru dengan nama yang sama seperti yang direferensikan di ID sumber langganan, langganan akan mulai mengirim pemberitahuan untuk acara dari objek baru.

Amazon Redshift menerbitkan pemberitahuan acara ke topik Amazon SNS, yang diidentifikasi oleh Nama Sumber Daya Amazon (ARN). Saat membuat langganan acara menggunakan konsol Amazon

Redshift, Anda dapat menentukan topik Amazon SNS yang ada, atau meminta konsol membuat topik saat membuat langganan.

Semua pemberitahuan acara Amazon Redshift yang dikirim ke topik Amazon SNS pada gilirannya dikirimkan ke semua konsumen Amazon SNS yang berlangganan topik itu. Gunakan konsol Amazon SNS untuk membuat perubahan pada topik Amazon SNS, seperti menambahkan atau menghapus langganan konsumen ke topik tersebut.

Bagian berikut mencantumkan semua kategori dan acara yang dapat Anda beri tahu. Ini juga memberikan informasi tentang berlangganan dan bekerja dengan langganan acara Amazon Redshift.

Membuat langganan pemberitahuan acara

Anda dapat membuat langganan notifikasi acara Amazon Simple Notification Service (Amazon SNS) untuk mengirim notifikasi saat peristiwa terjadi untuk cluster Amazon Redshift, snapshot, grup keamanan, atau grup parameter tertentu. Pemberitahuan ini dikirim ke topik SNS, yang pada gilirannya mengirimkan pesan ke konsumen SNS yang berlangganan topik tersebut.

Pesan SNS kepada konsumen dapat dalam bentuk pemberitahuan apa pun yang didukung oleh Amazon SNS untuk AWS suatu Wilayah, seperti email, pesan teks, atau panggilan ke titik akhir HTTP. Misalnya, semua wilayah mendukung pemberitahuan email, tetapi pemberitahuan SMS hanya dapat dibuat di Wilayah AS Timur (Virginia N.). Untuk informasi selengkapnya, lihat [Pemberitahuan acara klaster yang disediakan Amazon Redshift](#).

Untuk membuat langganan acara

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Acara.
3. Pilih tab Berlangganan acara, lalu pilih Buat langganan acara.
4. Masukkan properti langganan acara Anda, seperti nama, jenis sumber, kategori, dan tingkat keparahan. Anda juga dapat mengaktifkan topik Amazon SNS untuk mendapatkan pemberitahuan tentang acara.
5. Pilih Buat langganan acara untuk membuat langganan Anda.

Pemberitahuan acara klaster yang disediakan Amazon Redshift

Halaman ini menampilkan acara IDs dan kategori untuk setiap jenis sumber Amazon Redshift.

Kategori dan peristiwa untuk jenis sumber cluster

Tabel berikut menunjukkan kategori acara dan daftar peristiwa ketika cluster adalah jenis sumber.

Kategori Amazon Redshift	ID peristiwa	Tingkat keparahan peristiwa	Deskripsi
Konfigurasi	REDSFT-EV ENT-1000	INFO	Grup parameter [nama grup parameter] diperbarui pada [waktu]. Jika Anda hanya mengubah parameter dinamis, cluster terkait sedang dimodifikasi sekarang. Jika Anda mengubah parameter statis, semua pembaruan, termasuk parameter dinamis, akan diterapkan saat Anda me-reboot cluster terkait.
Konfigurasi	REDSHIFT- EVENT-1001	INFO	Cluster Amazon Redshift [nama cluster] Anda telah dimodifikasi untuk menggunakan grup parameter [nama grup parameter] pada [waktu].
Konfigurasi	ACARA PERGESERAN MERAH-1500	ERROR	VPC Amazon [nama VPC] tidak ada. Perubahan konfigurasi Anda untuk klaster [nama klaster] tidak diterapkan. Silakan kunjungi Konsol Manajemen AWS untuk memperbaiki masalah.
Konfigurasi	ACARA PERGESERAN MERAH-1501	ERROR	Subnet pelanggan [nama subnet] yang Anda tentukan untuk Amazon VPC [nama VPC] tidak ada atau tidak valid. Perubahan konfigurasi Anda untuk klaster [nama klaster] tidak diterapkan. Silakan kunjungi Konsol Manajemen AWS untuk memperbaiki masalah.

Kategori Amazon Redshift	ID peristiwa	Tingkat keparahan peristiwa	Deskripsi
Konfigurasi	ACARA PERGESERAN MERAH-1502	ERROR	Subnet dalam grup subnet cluster [nama grup subnet] tidak memiliki alamat IP yang tersedia. Cluster [nama cluster] tidak dapat dibuat.
Konfigurasi	ACARA PERGESERAN MERAH-1503	ERROR	VPC Amazon [nama VPC] tidak memiliki gateway internet yang melekat padanya. Perubahan konfigurasi Anda untuk klaster [nama klaster] tidak diterapkan. Silakan kunjungi Konsol Manajemen AWS untuk memperbaiki masalah.
Konfigurasi	REDSFT-EV ENT-1504	ERROR	HSM untuk klaster [nama cluster] tidak dapat dijangkau.
Konfigurasi	ACARA PERGESERAN MERAH-1505	ERROR	HSM untuk klaster [nama klaster] tidak dapat didaftarkan. Coba konfigurasi yang berbeda.
Konfigurasi	REDSFT-EV ENT-1506	ERROR	Amazon Redshift melampaui batas elastis network interface akun Anda. Hapus hingga [jumlah maksimum antarmuka jaringan elastis] antarmuka jaringan elastis atau minta peningkatan batas jumlah antarmuka jaringan per AWS Wilayah dengan EC2.
Konfigurasi	REDSFT-EV ENT-1509	ERROR	Cluster Amazon Redshift [nama cluster] tidak dapat dibuat karena batas titik akhir VPC akun Anda telah tercapai. Hapus titik akhir VPC yang tidak digunakan atau minta peningkatan batas titik akhir VPC. Untuk informasi selengkapnya, lihat Titik akhir VPC di Panduan Pengguna Amazon VPC.

Kategori Amazon Redshift	ID peristiwa	Tingkat keparahan peristiwa	Deskripsi
Konfigurasi	ACARA PERGESERAN MERAH-1510	ERROR	Kami telah mendeteksi bahwa upaya untuk memuat data sampel di klaster Amazon Redshift [nama cluster] Anda tidak berhasil. Untuk memuat data sampel, pertama-tama konfigurasikan VPC Anda agar memiliki akses ke bucket Amazon S3, lalu buat cluster baru dan muat data sampel. Untuk informasi selengkapnya, lihat Mengaktifkan perutean VPC yang disempurnakan di Panduan Manajemen Pergeseran Merah Amazon.
Konfigurasi	ACARA PERGESERAN MERAH-1511	ERROR	Cluster Amazon Redshift [nama klaster] tidak dapat dibuat karena Anda melebihi batas alamat IP Elastis akun Anda. Hapus alamat IP Elastis yang tidak digunakan atau minta peningkatan batas dengan Amazon EC2.
Manajemen	ACARA PERGESERAN MERAH-2000	INFO	Cluster Amazon Redshift Anda: [nama cluster] telah dibuat dan siap digunakan.
Manajemen	ACARA PERGESERAN MERAH-2001	INFO	Cluster Amazon Redshift [nama cluster] Anda telah dihapus pada [waktu]. Cuplikan terakhir [adalah/tidak] disimpan.
Manajemen	ACARA PERGESERAN MERAH-2002	INFO	Grup keamanan VPC untuk klaster [nama cluster] diperbarui pada [waktu di UTC].
Manajemen	ACARA PERGESERAN MERAH-2003	INFO	Pemeliharaan dimulai pada cluster [nama cluster] pada [waktu di UTC].

Kategori Amazon Redshift	ID peristiwa	Tingkat keparahan peristiwa	Deskripsi
Manajemen	ACARA PERGESERAN MERAH-2004	INFO	Pemeliharaan pada cluster [nama cluster] selesai pada [waktu di UTC].
Manajemen	ACARA PERGESERAN MERAH-2006	INFO	Pengubahan ukuran cluster [nama cluster] dimulai pada [waktu di UTC]. Cluster dalam mode read-only.
Manajemen	ACARA PERGESERAN MERAH-2007	INFO	Permintaan pengubahan ukuran untuk klaster [nama cluster] telah diakui.
Manajemen	ACARA PERGESERAN MERAH-2008	INFO	Operasi pemulihan Anda untuk membuat snapshot [nama klaster] klaster Amazon Redshift baru [nama snapshot] dimulai pada [waktu]. Untuk memantau kemajuan pemulihan, silakan kunjungi Konsol Manajemen AWS.
Manajemen	ACARA PERGESERAN MERAH-2013	INFO	Cluster Amazon Redshift Anda [nama cluster] diganti namanya pada [waktu].
Manajemen	ACARA PERGESERAN MERAH-2014	INFO	Permintaan pemulihan tabel untuk klaster Amazon Redshift [nama cluster] telah diterima.
Manajemen	REDSHIFT-EVENT-2015	INFO	Pemulihan tabel dibatalkan untuk klaster Amazon Redshift [nama cluster] pada [waktu].
Manajemen	ACARA PERGESERAN MERAH-2016	INFO	Penggantian klaster Amazon Redshift Anda [nama cluster] dimulai pada [waktu].

Kategori Amazon Redshift	ID peristiwa	Tingkat keparahan peristiwa	Deskripsi
Manajemen	ACARA PERGESERAN MERAH-2017	INFO	Pemeliharaan yang dimulai oleh pelanggan dimulai di klaster Amazon Redshift [nama klaster] Anda pada [waktu]. Cluster mungkin tidak tersedia selama pemeliharaan.
Manajemen	ACARA PERGESERAN MERAH-2018	INFO	Pemeliharaan yang dimulai oleh pelanggan diselesaikan di klaster Amazon Redshift [nama klaster] Anda pada [waktu].
Manajemen	ACARA PERGESERAN MERAH-2019	ERROR	Pemeliharaan yang dimulai oleh pelanggan gagal di klaster Amazon Redshift [nama klaster] Anda pada [waktu]. Mengembalikan cluster kembali ke keadaan semula.
Manajemen	ACARA PERGESERAN MERAH-2020	INFO	Lagu [nama klaster] klaster Amazon Redshift Anda telah dimodifikasi dari [dari trek] menjadi [untuk melacak].
Manajemen	ACARA PERGESERAN MERAH-2021	ERROR	[Operasi] klaster Amazon Redshift [nama cluster] tidak berhasil saat memperoleh kapasitas dari kumpulan kapasitas kami. Kami bekerja untuk memperoleh kapasitas tetapi untuk saat ini, kami telah membatalkan permintaan Anda. Hapus cluster ini dan coba lagi nanti.
Manajemen	ACARA PERGESERAN MERAH-2022	ERROR	[Operasi] klaster Amazon Redshift [nama cluster] tidak berhasil saat memperoleh kapasitas dari kumpulan kapasitas kami. Kami bekerja untuk memperoleh kapasitas tetapi untuk saat ini, kami telah membatalkan permintaan Anda. Kapasitas tersedia di [Availability Zone alternatif]. Hapus klaster ini dan coba lagi di Availability Zone alternatif.

Kategori Amazon Redshift	ID peristiwa	Tingkat keparahan peristiwa	Deskripsi
Manajemen	ACARA PERGESERAN MERAH-2023	ERROR	Kami telah mendeteksi kegagalan perangkat keras pada cluster Amazon Redshift tunggal Anda [nama klaster], yang mungkin mengakibatkan kueri gagal atau ketersediaan klaster yang terputus-putus. Mengganti cluster tidak berhasil saat memperoleh kapasitas dari kumpulan kapasitas kami. Anda perlu memulihkan cluster baru dari snapshot. Hapus klaster ini, pilih snapshot terbaru yang tersedia, dan pulihkan cluster baru dari snapshot itu. Ini akan secara otomatis memberi Anda perangkat keras yang sehat.
Manajemen	ACARA PERGESERAN MERAH-2024	ERROR	Kami telah mendeteksi kegagalan perangkat keras pada cluster Amazon Redshift tunggal Anda [nama klaster], yang mungkin mengakibatkan kueri gagal atau ketersediaan klaster yang terputus-putus. Mengganti cluster tidak berhasil saat memperoleh kapasitas dari kumpulan kapasitas kami. Kapasitas tersedia di Availability Zone: [alternatif Availability Zones]. Hapus klaster ini, pilih snapshot terbaru yang tersedia, dan pulihkan cluster baru dari snapshot itu. Ini akan secara otomatis memberi Anda perangkat keras yang sehat.
Manajemen	ACARA PERGESERAN MERAH-3011	INFO	Pengubahan ukuran elastis untuk cluster Amazon Redshift '[nama cluster]' dimulai pada [waktu]. Kami akan menahan koneksi database selama mengubah ukuran. Beberapa kueri dan koneksi dapat dihentikan atau habis waktu selama operasi ini.

Kategori Amazon Redshift	ID peristiwa	Tingkat keparahan peristiwa	Deskripsi
Manajemen	ACARA PERGESERAN MERAH-3012	INFO	Kami telah menerima permintaan pengubahan ukuran elastis untuk cluster '[nama cluster]' yang dimulai pada [waktu]. Kami akan memberikan pemberitahuan acara saat mengubah ukuran dimulai.
Tertunda	ACARA PERGESERAN MERAH-2025	INFO	Database Anda untuk klaster [nama klaster] akan diperbarui antara [waktu mulai] dan [waktu akhir]. Cluster Anda tidak akan dapat diakses. Rencanakan sesuai.
Tertunda	ACARA PERGESERAN MERAH-2026	INFO	Cluster [nama cluster] Anda akan diperbarui antara [waktu mulai] dan [waktu akhir]. Cluster Anda tidak akan dapat diakses. Rencanakan sesuai.
Memantau	ACARA PERGESERAN MERAH-2050	INFO	Masalah perangkat keras terdeteksi di klaster Amazon Redshift [nama cluster]. Permintaan penggantian dimulai pada [waktu].
Memantau	REDSFT-EV ENT-3000	INFO	Cluster Amazon Redshift [nama cluster] Anda di-boot ulang pada [waktu].
Memantau	REDSFT-EV ENT-3001	INFO	Node di klaster Amazon Redshift Anda: [nama cluster] secara otomatis diganti pada [waktu], dan klaster Anda beroperasi secara normal.
Memantau	REDSFT-EV ENT-3002	INFO	Pengubahan ukuran untuk klaster Amazon Redshift [nama klaster] Anda selesai dan klaster Anda tersedia untuk dibaca dan ditulis. Pengubahan ukuran dimulai pada [waktu] dan membutuhkan [jam] jam untuk menyelesaikannya.

Kategori Amazon Redshift	ID peristiwa	Tingkat keparahan peristiwa	Deskripsi
Memantau	REDSFT-EV ENT-3003	INFO	Cluster Amazon Redshift [nama cluster] berhasil dibuat dari snapshot [nama snapshot] dan tersedia untuk digunakan.
Memantau	REDSFT-EV ENT-3007	INFO	Snapshot Amazon Redshift Anda [nama snapshot] berhasil disalin dari [AWS Wilayah sumber] ke [AWS Wilayah tujuan] pada [waktu].
Memantau	REDSFT-EV ENT-3008	INFO	Pemulihan tabel dimulai untuk klaster Amazon Redshift [nama cluster] pada [waktu].
Memantau	REDSFT-EV ENT-3009	INFO	Pemulihan tabel berhasil diselesaikan untuk klaster Amazon Redshift [nama cluster] pada [waktu].
Memantau	REDSFT-EV ENT-3010	ERROR	Pemulihan tabel gagal untuk klaster Amazon Redshift [nama klaster] pada [waktu].
Memantau	ACARA PERGESERAN MERAH-3013	ERROR	Operasi pengubahan ukuran elastis yang diminta untuk klaster Amazon Redshift [nama cluster] gagal pada [waktu] karena [alasan].
Memantau	ACARA PERGESERAN MERAH-3014	INFO	Amazon Redshift me-reboot cluster [nama cluster] pada [waktu].
Tertunda	REDSFT-EV ENT-3015	INFO	Kami telah menjadwalkan reboot untuk cluster Anda [nama cluster] untuk pemeliharaan antara [waktu mulai di UTC] dan [waktu akhir di UTC]. Cluster Anda tidak akan dapat diakses selama waktu ini. Kami menyarankan Anda merencanakan pemadaman untuk menghindari kemunduran.

Kategori Amazon Redshift	ID peristiwa	Tingkat keparahan peristiwa	Deskripsi
Memantau	ACARA PERGESERAN MERAH-3016	INFO	Cluster akan di-boot ulang.
Memantau	REDSFT-EV ENT-3500	ERROR	Pengubahan ukuran untuk klaster Amazon Redshift [nama cluster] gagal. Pengubahan ukuran akan dicoba ulang secara otomatis dalam beberapa menit.
Memantau	REDSFT-EV ENT-3501	ERROR	Operasi pemulihan Anda untuk membuat klaster Amazon Redshift [nama klaster] dari snapshot [nama snapshot] gagal pada [waktu]. Silakan coba lagi operasi Anda.
Memantau	REDSFT-EV ENT-3504	ERROR	Bucket Amazon S3 [nama bucket] tidak valid untuk pencatatan klaster [nama cluster].
Memantau	REDSFT-EV ENT-3505	ERROR	Bucket Amazon S3 [nama bucket] tidak memiliki kebijakan IAM yang benar untuk klaster [nama cluster].
Memantau	REDSFT-EV ENT-3506	ERROR	Bucket Amazon S3 [nama ember] tidak ada. Pencatatan tidak dapat dilanjutkan untuk klaster [nama klaster].
Memantau	REDSFT-EV ENT-3507	ERROR	Cluster Amazon Redshift [nama cluster] tidak dapat dibuat menggunakan EIP [alamat IP]. EIP ini sudah digunakan.
Memantau	REDSFT-EV ENT-3508	ERROR	Cluster Amazon Redshift [nama cluster] tidak dapat dibuat menggunakan EIP [alamat IP]. EIP tidak dapat ditemukan.
Memantau	REDSFT-EV ENT-3509	ERROR	Salinan snapshot lintas wilayah tidak diaktifkan untuk klaster [nama cluster].

Kategori Amazon Redshift	ID peristiwa	Tingkat keparahan peristiwa	Deskripsi
Memantau	ACARA PERGESERAN MERAH-3510	ERROR	Pemulihan tabel gagal dimulai untuk klaster Amazon Redshift [nama cluster] pada [waktu]. Alasan: [alasan].
Memantau	ACARA PERGESERAN MERAH-3511	ERROR	Pemulihan tabel gagal untuk klaster Amazon Redshift [nama klaster] pada [waktu].
Memantau	ACARA PERGESERAN MERAH-3512	ERROR	Cluster Amazon Redshift [nama cluster] gagal karena masalah perangkat keras. Cluster sedang dipulihkan secara otomatis dari snapshot terbaru [nama snapshot] yang dibuat pada [waktu].
Memantau	ACARA PERGESERAN MERAH-3513	ERROR	Cluster Amazon Redshift [nama cluster] gagal karena masalah perangkat keras. Cluster sedang dipulihkan secara otomatis dari snapshot terbaru [nama snapshot] yang dibuat pada [waktu]. Setiap perubahan database yang dilakukan setelah waktu ini perlu dikirimkan kembali.
Memantau	ACARA PERGESERAN MERAH-3514	ERROR	Cluster Amazon Redshift [nama cluster] gagal karena masalah perangkat keras. Cluster ditempatkan dalam status kegagalan perangkat keras. Harap hapus cluster dan pulihkan dari snapshot terbaru [nama snapshot] yang dibuat pada [waktu].

Kategori Amazon Redshift	ID peristiwa	Tingkat keparahan peristiwa	Deskripsi
Memantau	REDSFT-EV ENT-3515	ERROR	Cluster Amazon Redshift [nama cluster] gagal karena masalah perangkat keras. Cluster ditempatkan dalam status kegagalan perangkat keras. Harap hapus cluster dan pulihkan dari snapshot terbaru [nama snapshot] yang dibuat pada [waktu]. Setiap perubahan database yang dilakukan setelah waktu ini perlu dikirimkan kembali.
Memantau	ACARA PERGESERAN MERAH-3516	ERROR	Cluster Amazon Redshift [nama cluster] gagal karena masalah perangkat keras dan tidak ada cadangan untuk cluster. Cluster ditempatkan dalam status kegagalan perangkat keras dan dapat dihapus.
Memantau	PERGESERAN MERAH-PERISTIWA 3519	INFO	Cluster [nama cluster] mulai dimulai ulang pada [waktu].
Memantau	REDSFT-EV ENT-3520	INFO	Cluster [nama cluster] selesai restart pada [waktu].
Memantau	ACARA PERGESERAN MERAH-3521	INFO	Kami mendeteksi masalah konektivitas pada cluster '[nama cluster]'. Pemeriksaan diagnostik otomatis telah dimulai pada [waktu].
Memantau	ACARA PERGESERAN MERAH-3522	INFO	Tindakan pemulihan pada klaster '[nama cluster]' gagal pada [waktu]. Tim Amazon Redshift sedang mengerjakan solusi.
Memantau	ACARA PERGESERAN MERAH-3533	ERROR	Pengubahan ukuran cluster pada '[nama cluster]' dibatalkan pada [waktu]. Operasi dibatalkan karena [alasan]. [tindakan yang diperlukan].

Kategori Amazon Redshift	ID peristiwa	Tingkat keparahan peristiwa	Deskripsi
Memantau	ACARA PERGESERAN MERAH-3534	INFO	Pengubahan ukuran elastis untuk cluster Amazon Redshift '[nama cluster]' selesai pada [waktu]. Cluster sekarang tersedia untuk operasi baca dan tulis saat kami mentransfer data. Beberapa pertanyaan mungkin membutuhkan waktu lebih lama untuk diselesaikan hingga transfer data selesai.
Memantau	ACARA PERGESERAN MERAH-3537	INFO	Transfer data cluster '[nama cluster]' selesai pada [waktu di UTC].
Memantau	ACARA PERGESERAN MERAH-3600	INFO	Operasi pengubahan ukuran yang diminta untuk cluster Amazon Redshift '[nama cluster]' dibatalkan di masa lalu. Rollback selesai pada [waktu].
Tertunda	ACARA PERGESERAN MERAH-3601	INFO	Sebuah node pada klaster Anda [nama cluster] akan diganti antara [waktu mulai] dan [waktu akhir]. Anda tidak dapat menunda pemeliharaan ini. Rencanakan sesuai.
Tertunda	REDSFT-EV ENT-3602	INFO	Sebuah node pada klaster Anda [nama cluster] dijadwalkan untuk diganti antara [waktu mulai] dan [waktu akhir]. Cluster Anda tidak akan dapat diakses. Rencanakan sesuai.
Manajemen	ACARA PERGESERAN MERAH-3603	INFO	Operasi pemulihan untuk membuat klaster [nama cluster] dari snapshot [nama snapshot] gagal karena kesalahan internal. Cluster ditempatkan dalam status pemulihan yang tidak kompatibel dan dapat dihapus. Cobalah untuk mengembalikan snapshot ke dalam cluster dengan konfigurasi yang berbeda.

Kategori Amazon Redshift	ID peristiwa	Tingkat keparahan peristiwa	Deskripsi
Manajemen	ACARA PERGESERAN MERAH-3614	INFO	Tindakan terjadwal [nama tindakan terjadwal] dibuat pada [waktu di UTC]. Doa pertama dijadwalkan pada [waktu di UTC].
Manajemen	REDSFT-EV ENT-3615	INFO	Tindakan terjadwal [nama tindakan terjadwal] dijadwalkan pada [waktu di UTC].
Memantau	ACARA PERGESERAN MERAH-3616	INFO	Tindakan terjadwal [nama tindakan terjadwal] pada [waktu di UTC] selesai dengan status 'SUCCEEDED'.
Memantau	ACARA PERGESERAN MERAH-3617	ERROR	Tindakan terjadwal [nama tindakan terjadwal] dilewati pada [waktu di UTC] karena penundaan.
Memantau	ACARA PERGESERAN MERAH-3618	INFO	Operasi jeda cluster [nama cluster] dimulai pada [waktu UTC]. Jeda Dimulai
Memantau	REDSFT-EV ENT-3619	INFO	Cluster Amazon Redshift [nama cluster] berhasil dijeda pada [waktu UTC].
Manajemen	REDSFT-EV ENT-3626	INFO	Tindakan terjadwal [nama tindakan terjadwal] dimodifikasi pada [waktu di UTC]. Doa pertama dijadwalkan pada [waktu di UTC].
Manajemen	REDSFT-EV ENT-3627	INFO	Tindakan terjadwal [nama tindakan terjadwal] telah dihapus pada [waktu di UTC].
Memantau	REDSFT-EV ENT-3628	ERROR	Tindakan terjadwal [nama tindakan terjadwal] pada [waktu di UTC] selesai dengan status 'GAGAL'.
Memantau	REDSFT-EV ENT-3629	INFO	Permintaan failover internal untuk klaster [nama cluster] telah dimulai.

Kategori Amazon Redshift	ID peristiwa	Tingkat keparahan peristiwa	Deskripsi
Memantau	REDSFT-EV ENT-3630	INFO	Amazon Redshift berhasil memindahkan kluster Amazon Redshift [nama kluster] Anda dari [zona ketersediaan] ke [zona ketersediaan] untuk pemulihan.
Manajemen	REDSFT-EV ENT-3631	INFO	Amazon Redshift [nama cluster] menerima permintaan relokasi Anda. Saat relokasi Availability Zone selesai, Amazon Redshift mengirimkan pemberitahuan acara.
Memantau	REDSFT-EV ENT-3632	INFO	Cluster Amazon Redshift [nama cluster] berhasil dipindahkan dari [zona ketersediaan] ke [zona ketersediaan]. Anda dapat menggunakan cluster sekarang.
Memantau	REDSFT-EV ENT-3658	ERROR	Migrasi EC2-klasik ke EC2-VPC gagal untuk cluster Redshift [id cluster].
Memantau	ACARA PERGESERAN MERAH-3659	INFO	Migrasi EC2-klasik ke EC2-VPC berhasil untuk cluster Redshift [id cluster].
Memantau	REDSFT-EV ENT-3660	INFO	Cluster ditempatkan dalam status kegagalan perangkat keras. Harap hapus kluster EC2-Classic dan pulihkan ke cluster EC2-VPC dari snapshot terbaru [nama snapshot] yang dibuat pada [waktu di UTC].
Manajemen	REDSFT-EV ENT-3666	INFO	Amazon Redshift Multi-AZ cluster [nama cluster] telah mendeteksi kegagalan pada [waktu di UTC] dan memicu pemulihan otomatis.

Kategori Amazon Redshift	ID peristiwa	Tingkat keparahan peristiwa	Deskripsi
Manajemen	ACARA PERGESERAN MERAH-3667	INFO	Amazon Redshift Multi-AZ cluster [nama cluster] berhasil dipulihkan pada [waktu di UTC] dan tersedia untuk digunakan di [zona ketersediaan pertama]. Komputasi sekunder di AZ lain akan segera tersedia.
Memantau	ACARA PERGESERAN MERAH-3668	ERROR	Cluster Amazon Redshift Multi-AZ [nama cluster] gagal pulih pada [waktu di UTC].
Manajemen	REDSFT-EV ENT-3669	INFO	Amazon Redshift Multi-AZ cluster [nama cluster] berhasil dipulihkan pada [waktu di UTC] dan tersedia untuk digunakan dengan sumber daya komputasi dari [zona ketersediaan pertama] dan [zona ketersediaan kedua].
Manajemen	REDSFT-EV ENT-3670	INFO	Pemeliharaan di klaster Amazon Redshift [nama klaster] selesai pada [waktu di UTC] dan tersedia untuk digunakan dengan sumber daya komputasi di [zona ketersediaan pertama]. Komputasi sekunder di AZ lain akan segera tersedia.
Manajemen	REDSFT-EV ENT-3671	INFO	Ubah ukuran pada klaster Amazon Redshift [nama cluster] selesai pada [waktu di UTC] dan tersedia untuk digunakan di [zona ketersediaan pertama]. Komputasi sekunder di AZ lain akan segera tersedia.
Manajemen	REDSFT-EV ENT-3672	INFO	Amazon Redshift Multi-AZ cluster [nama cluster] telah mendeteksi kegagalan di [zona ketersediaan kedua] pada [waktu di UTC] dan memicu pemulihan otomatis.

Kategori Amazon Redshift	ID peristiwa	Tingkat keparahan peristiwa	Deskripsi
Manajemen	REDSFT-EV ENT-3673	INFO	Operasi untuk mengaktifkan Multi-AZ untuk klaster Amazon Redshift [nama cluster] telah dimulai pada [waktu di UTC].
Manajemen	REDSFT-EV ENT-3674	INFO	Operasi untuk mengaktifkan Multi-AZ untuk klaster Amazon Redshift [nama cluster] telah berhasil diselesaikan pada [waktu di UTC].
Memantau	ACARA PERGESERAN MERAH-3675	ERROR	Operasi untuk mengaktifkan Multi-AZ untuk klaster Amazon Redshift [nama cluster] telah gagal pada [waktu di UTC].
Manajemen	ACARA PERGESERAN MERAH-3676	INFO	Operasi untuk menonaktifkan Multi-AZ untuk klaster Amazon Redshift Multi-AZ [nama cluster] Anda telah dimulai pada [waktu di UTC].
Manajemen	REDSFT-EV ENT-3677	INFO	Operasi untuk menonaktifkan Multi-AZ untuk klaster Amazon Redshift [nama cluster] Anda telah berhasil diselesaikan pada [waktu di UTC].
Memantau	REDSFT-EV ENT-3678	ERROR	Operasi untuk menonaktifkan Multi-AZ untuk klaster Amazon Redshift [nama cluster] Anda telah gagal pada [waktu di UTC].
Konfigurasi	ACARA PERGESERAN MERAH-3679	INFO	Port cluster Amazon Redshift [nama cluster] berhasil dimodifikasi.

Kategori Amazon Redshift	ID peristiwa	Tingkat keparahan peristiwa	Deskripsi
Konfigurasi	REDSFT-EV ENT-3680	ERROR	Amazon Redshift tidak dapat membuat klaster [nama klaster] karena Service Linked Role (SLR) yang diperlukan untuk operasi ini tidak dapat diakses. Coba buat lagi dari konsol Amazon Redshift. Amazon Redshift akan membuat SLR secara otomatis.
Memantau	REDSFT-EV ENT-3684	ERROR	Bucket Amazon S3 [nama bucket] Anda telah dienkripsi dengan kunci yang tidak dikenal atau tidak dapat diakses. AWS KMS Ubah enkripsi bucket Amazon S3 Anda.
Manajemen	REDSFT-EV ENT-3685	ERROR	Operasi pemulihan pada cluster [nama cluster] gagal karena tidak memiliki cukup ruang disk yang tersedia. Operasi sedang diputar kembali. Coba pulihkan ke cluster dengan konfigurasi yang berbeda.
Manajemen	ACARA PERGESERAN MERAH-3686	ERROR	Operasi pengubahan ukuran pada klaster [nama cluster] gagal karena tidak memiliki cukup ruang disk yang tersedia. Operasi sedang diputar kembali. Coba ubah ukuran ke cluster dengan konfigurasi yang berbeda.
Manajemen	ACARA PERGESERAN MERAH-3687	INFO	Operasi pengubahan ukuran yang diminta untuk klaster Amazon Redshift [nama cluster] selesai pada [waktu di UTC].
Keamanan	REDSFT-EV ENT-3688	ERROR	Rotasi kunci enkripsi untuk klaster Amazon Redshift [nama klaster] Anda tidak dapat diselesaikan.

Kategori Amazon Redshift	ID peristiwa	Tingkat keparahan peristiwa	Deskripsi
DataSharing	REDSFT-EV ENT-3689	INFO	<time in UTC>Operasi untuk mendaftarkan namespace Amazon Redshift Anda <cluster name>ke <account id>akun Glue Data Catalog dimulai pada.
DataSharing	REDSFT-EV ENT-3690	INFO	<time in UTC>Operasi untuk mendaftarkan namespace Amazon Redshift Anda <cluster name>ke <account id>akun Glue Data Catalog telah berhasil diselesaikan di.
DataSharing	REDSFT-EV ENT-3691	INFO	<time in UTC>Operasi untuk mendaftarkan namespace Amazon Redshift Anda <cluster name>ke <account id>akun Glue Data Catalog telah gagal di.
DataSharing	ACARA PERGESERAN MERAH-3692	INFO	<account id><time in UTC>Operasi untuk membatalkan pendaftaran namespace Amazon Redshift Anda <cluster name>dari akun Glue Data Catalog dimulai pada.
DataSharing	ACARA PERGESERAN MERAH-3693	INFO	<account id><time in UTC>Operasi untuk membatalkan pendaftaran namespace Amazon Redshift Anda <cluster name>dari akun Glue Data Catalog telah berhasil diselesaikan di.
DataSharing	ACARA PERGESERAN MERAH-3694	INFO	<account id><time in UTC>Operasi untuk membatalkan pendaftaran namespace Amazon Redshift Anda <cluster name>dari akun Glue Data Catalog telah gagal di.
Keamanan	REDSFT-EV ENT-4000	INFO	Kredensi admin Anda untuk klaster Amazon Redshift Anda: [nama klaster] diperbarui pada [waktu].

Kategori Amazon Redshift	ID peristiwa	Tingkat keparahan peristiwa	Deskripsi
Keamanan	REDSFT-EV ENT-4001	INFO	Grup keamanan [nama grup keamanan] dimodifikasi pada [waktu]. Perubahan akan terjadi untuk semua cluster terkait secara otomatis.
Keamanan	REDSFT-EV ENT-4005	ERROR	Cluster [nama cluster] memiliki izin yang tidak valid AWS KMS key . Cluster Anda akan memasukkan status “INACCESSIBLE_KMS_KEY” dan Anda tidak lagi dapat mengakses nya.
Keamanan	REDSFT-EV ENT-4006	INFO	Cluster [nama cluster] kini telah dikembalikan ke keadaan sebelumnya karena AWS KMS key izin telah dipulihkan.
Keamanan	REDSFT-EV ENT-4500	ERROR	Satu atau beberapa grup keamanan yang Anda berikan untuk klaster Amazon Redshift [nama cluster] tidak dapat ditemukan. Jika ada grup keamanan yang valid, mereka diterapkan ke cluster. Setiap kelompok keamanan yang tidak valid tidak diterapkan. Jika Anda memodifikasi klaster yang ada dan tidak ada grup keamanan yang valid, klaster menyimpan grup keamanan aslinya. Jika Anda membuat klaster baru dan tidak ada grup keamanan yang valid, pembuatan klaster gagal. Dalam kedua kasus tersebut, silakan coba lagi permintaan dengan grup keamanan yang valid. Untuk informasi lebih lanjut tentang mengelola grup keamanan di Amazon Redshift, buka Grup keamanan VPC di Panduan Manajemen Pergeseran Merah Amazon.

Kategori Amazon Redshift	ID peristiwa	Tingkat keparahan peristiwa	Deskripsi
Keamanan	REDSFT-EV ENT-4501	ERROR	Grup keamanan [nama grup keamanan] yang ditentukan dalam Grup Keamanan Cluster [nama grup keamanan kluster] tidak dapat ditemukan. Otorisasi tidak dapat diselesaikan.
Keamanan	REDSFT-EV ENT-4502	ERROR	Kredensi admin untuk kluster Amazon Redshift [nama kluster] gagal diperbarui pada [waktu] karena aktivitas bersamaan. Biarkan beban kerja saat ini menyelesaikan atau mengurangi beban kerja aktif dan kemudian coba lagi operasi.
Keamanan	PERUBAHAN-PERISTIWA-4503	ERROR	Amazon Redshift tidak dapat mengakses rahasia untuk kluster [nama cluster] Anda.
Keamanan	PERUBAHAN-PERISTIWA-4504	ERROR	Amazon Redshift tidak dapat mengakses kunci KMS [kunci KMS] yang digunakan untuk mengenkripsi rahasia kredensi admin untuk kluster [nama cluster] Anda.
Keamanan	PERUBAHAN-PERISTIWA-4505	ERROR	Amazon Redshift tidak dapat memutar rahasia untuk kluster [nama cluster] karena ada operasi yang sedang berlangsung di cluster.
Keamanan	PERUBAHAN-PERISTIWA-4506	ERROR	Cluster Amazon Redshift [nama cluster] Anda dijeda. Amazon Redshift tidak dapat memutar rahasia cluster yang dijeda.

Kategori dan peristiwa untuk tipe sumber grup parameter

Tabel berikut menunjukkan kategori kejadian dan daftar kejadian saat grup parameter merupakan jenis sumber.

Kategori Amazon Redshift	ID peristiwa	Tingkat keparahan peristiwa	Deskripsi
Konfigurasi	REDSHIFT-EVENT-1002	INFO	Parameter [nama parameter] diperbarui dari [value] ke [value] pada [time].
Konfigurasi	REDSHIFT-EVENT-1003	INFO	Grup parameter cluster [nama grup] telah dibuat.
Konfigurasi	REDSHIFT-EVENT-1004	INFO	Grup parameter cluster [nama grup] telah dihapus.
Konfigurasi	REDSHIFT-EVENT-1005	INFO	Grup parameter cluster [name] diperbarui pada [waktu]. Jika Anda hanya mengubah parameter dinamis, cluster terkait sedang dimodifikasi sekarang. Jika Anda mengubah parameter statis, semua pembaruan, termasuk parameter dinamis, akan diterapkan saat Anda me-reboot cluster terkait.

Kategori dan peristiwa untuk jenis sumber grup keamanan

Tabel berikut menunjukkan kategori acara dan daftar peristiwa ketika grup keamanan adalah tipe sumber.

Kategori Amazon Redshift	ID peristiwa	Tingkat keparahan peristiwa	Deskripsi
Keamanan	REDSFT-EVENT-4002	INFO	Grup keamanan klaster [nama grup] telah dibuat.
Keamanan	REDSFT-EVENT-4003	INFO	Grup keamanan klaster [nama grup] telah dihapus.

Kategori Amazon Redshift	ID peristiwa	Tingkat keparahan peristiwa	Deskripsi
Keamanan	REDSFT-EV ENT-4004	INFO	Grup keamanan klaster [nama grup] diubah pada [waktu]. Perubahan akan diterapkan secara otomatis ke semua cluster terkait.

Kategori dan acara untuk jenis sumber snapshot

Tabel berikut menunjukkan kategori acara dan daftar peristiwa ketika snapshot adalah jenis sumber.

Kategori Amazon Redshift	ID peristiwa	Tingkat keparahan peristiwa	Deskripsi
Manajemen	ACARA PERGESERAN MERAH-2009	INFO	Snapshot pengguna [nama snapshot] untuk Amazon Redshift Cluster [nama cluster] dimulai pada [waktu]. Untuk memantau kemajuan snapshot, silakan kunjungi. Konsol Manajemen AWS
Manajemen	REDSHIFT-EVENT-2010	INFO	Snapshot pengguna [nama snapshot] untuk klaster Amazon Redshift [nama klaster] dibatalkan pada [waktu].
Manajemen	ACARA PERGESERAN MERAH-2011	INFO	Snapshot pengguna [nama snapshot] untuk klaster Amazon Redshift [nama cluster] telah dihapus pada [waktu].
Manajemen	ACARA PERGESERAN MERAH-2012	INFO	Snapshot terakhir [nama snapshot] untuk cluster Amazon Redshift [nama cluster] dimulai pada [waktu].
Memantau	REDSFT-EV ENT-3004	INFO	Snapshot pengguna [nama snapshot] untuk klaster Amazon Redshift [nama

Kategori Amazon Redshift	ID peristiwa	Tingkat keparahan peristiwa	Deskripsi
			klaster] berhasil diselesaikan pada [waktu].
Memantau	ACARA PERGESERAN MERAH-3005	INFO	Cuplikan terakhir [nama] untuk klaster Amazon Redshift [nama] berhasil diselesaikan pada [waktu].
Memantau	ACARA PERGESERAN MERAH-3006	INFO	Snapshot terakhir [nama snapshot] untuk cluster Amazon Redshift [nama cluster] dibatalkan pada [waktu].
Memantau	REDSFT-EV ENT-3502	ERROR	Snapshot terakhir [nama snapshot] untuk klaster Amazon Redshift [nama cluster] gagal pada [waktu]. Tim sedang menyelidiki masalah ini. Silakan kunjungi Konsol Manajemen AWS untuk mencoba kembali operasi.
Memantau	REDSFT-EV ENT-3503	ERROR	Snapshot pengguna [nama snapshot] untuk klaster Amazon Redshift [nama klaster] gagal pada [waktu]. Tim sedang menyelidiki masalah ini. Silakan kunjungi Konsol Manajemen AWS untuk mencoba kembali operasi.

Pemberitahuan acara Amazon Redshift Tanpa Server dengan Amazon EventBridge

Amazon Redshift Serverless menggunakan EventBridge Amazon untuk mengelola pemberitahuan acara agar up-to-date Anda tetap memperhatikan perubahan di gudang data Anda. Amazon EventBridge adalah layanan bus acara tanpa server yang dapat Anda gunakan untuk menghubungkan aplikasi Anda dengan data dari berbagai sumber. Dalam hal ini, sumber acara adalah Amazon Redshift. Peristiwa, yang merupakan perubahan yang dipantau di lingkungan, dikirim

ke gudang EventBridge data Amazon Redshift Anda secara otomatis. Acara disampaikan dalam waktu nyaris nyata.

Kemampuan EventBridge termasuk menyediakan lingkungan bagi Anda untuk menulis aturan acara, yang dapat menentukan tindakan yang harus diambil untuk acara tertentu. Anda juga dapat mengatur target, yang merupakan sumber daya yang EventBridge dapat mengirim acara ke. Target dapat mencakup tujuan API, grup CloudWatch log Amazon, dan lainnya. Untuk informasi selengkapnya tentang aturan, lihat [EventBridge Aturan Amazon](#). Untuk informasi selengkapnya tentang target, lihat [EventBridge target Amazon](#).

Acara dapat diklasifikasikan ke dalam tingkat keparahan dan kategori. Filter berikut tersedia:

- Pemfilteran sumber daya - Menerima pesan berdasarkan sumber daya yang terkait dengan peristiwa. Sumber daya termasuk workgroup, snapshot, dan sebagainya.
- Pemfilteran jendela waktu - Cakupan peristiwa dalam periode waktu tertentu.
- Pemfilteran kategori - Menerima pemberitahuan acara untuk semua acara dalam kategori tertentu.

Tabel berikut mencakup peristiwa Amazon Redshift Tanpa Server, dengan metadata tambahan:

Kategori Amazon Redshift	ID Peristiwa Eksternal	Tingkat Keparahan Acara	Deskripsi Pesan
RateChange	REDSHIFT-SERVERLESS-ACARA-1001	INFO	<time in UTC>Perubahan RPU basis kelompok kerja berhasil diselesaikan di.
RateChange	REDSHIFT-SERVERLESS-ACARA-1002	ERROR	<time in UTC>Perubahan RPU basis grup kerja gagal diselesaikan di.
Memantau	REDSHIFT-SERVERLESS-ACARA-1003	INFO	<endpoint name><time in UTC>Perangkat lunak ini diperbaru

Kategori Amazon Redshift	ID Peristiwa Eksternal	Tingkat Keparahan Acara	Deskripsi Pesan
			i di Gudang Data Amazon Redshift Anda di.
Konfigurasi	REDSHIFT-SERVERLESS-ACARA-1011	ERROR	Amazon Redshift Serverless tidak dapat membuat grup kerja [nama grup kerja] karena Service Linked Role (SLR) yang diperlukan untuk operasi ini tidak dapat diakses. Coba buat lagi di konsol Amazon Redshift. Amazon Redshift akan membuat SLR secara otomatis.
Memantau	REDSHIFT-SERVERLESS-ACARA-1029	ERROR	Perubahan RPU basis grup kerja gagal diselesaikan pada [waktu di UTC] karena tidak memiliki cukup ruang disk yang tersedia. Coba lagi dengan konfigurasi yang berbeda.

Kategori Amazon Redshift	ID Peristiwa Eksternal	Tingkat Keparahan Acara	Deskripsi Pesan
Memantau	REDSHIFT-SERVERLESS-ACARA-1500	ERROR	Workgroup <workgroup name>tidak dapat dibuat atau diperbarui karena Anda melebihi batas alamat IP Elastis akun Anda. Hapus alamat IP Elastis yang tidak digunakan atau minta peningkatan batas dengan Amazon EC2.
Memantau	REDSHIFT-SERVERLESS-ACARA-1501	ERROR	Subnet <subnet id>tidak memiliki alamat IP yang tersedia. Ini akan mencegah jenis kueri berikut berjalan dengan sukses di workgroup<workgroup name>: EMR, kueri federasi, dari COPY/UNLOAD Amazon EC2. Untuk memperbaiki masalah, bebaskan IPs di subnet Anda dengan menghapus ENIs.

Kategori Amazon Redshift	ID Peristiwa Eksternal	Tingkat Keparahan Acara	Deskripsi Pesan
Memantau	REDSHIFT-SERVERLESS-ACARA-1502	ERROR	Subnet <subnet id>tidak memiliki alamat IP yang tersedia. <workgroup name>Ini akan mencegah EMR Amazon, kueri federasi Redshift, Redshift COPY/ UNLOAD, jenis kueri Redshift ML agar tidak berhasil berjalan di workgroup . Untuk memperbaiki masalah ini, kosongkan IPs subnet Anda dengan menghapus antarmuka jaringan elastis yang tidak digunakan (). ENIs
Manajemen	REDSHIFT-SERVERLESS-ACARA-1008	INFO	Workgroup Amazon Redshift Anda <workgroup name>telah dibuat dan siap digunakan.

Kategori Amazon Redshift	ID Peristiwa Eksternal	Tingkat Keparahan Acara	Deskripsi Pesan
Manajemen	REDSHIFT-SERVERLESS-ACARA-1009	INFO	<time in UTC>Workgroup Amazon Redshift Anda <workgroup name>telah dihapus di.
Memantau	REDSHIFT-SERVERLESS-ACARA-1000	INFO	<time in UTC>Snapshot <snapshot name>berhasil diselesaikan di.
Manajemen	REDSHIFT-SERVERLESS-ACARA-1004	INFO	<time in UTC>Pulihkan dari snapshot di namespace <namespace name>berhasil diselesaikan di.
Manajemen	REDSHIFT-SERVERLESS-ACARA-1005	ERROR	<namespace name><time in UTC>Pulihkan dari snapshot di namespace gagal di.
Manajemen	REDSHIFT-SERVERLESS-ACARA-1006	INFO	<time in UTC>Pulihkan dari titik pemulihan pada namespace yang <namespace name>berhasil diselesaikan di.

Kategori Amazon Redshift	ID Peristiwa Eksternal	Tingkat Keparahan Acara	Deskripsi Pesan
Manajemen	REDSHIFT-SERVERLESS-ACARA-1007	INFO	<time in UTC>Pemulihan dari titik pemulihan pada namespace <namespace name>gagal di.
Keamanan	REDSHIFT-SERVERLESS-ACARA-1012	ERROR	<namespace name>Amazon Redshift tidak dapat mengakses rahasia untuk namespace Anda.
Keamanan	REDSHIFT-SERVERLESS-ACARA-1013	ERROR	<namespace name>Amazon Redshift tidak dapat mengakses kunci KMS yang digunakan untuk mengenkripsi rahasia kredensial admin untuk namespace Anda.
Keamanan	REDSHIFT-SERVERLESS-ACARA-1014	ERROR	Amazon Redshift tidak dapat memutar rahasia untuk namespace Anda <namespace name>karena ada operasi yang sedang berlangsung di workgroup.

Kategori Amazon Redshift	ID Peristiwa Eksternal	Tingkat Keparahannya Acara	Deskripsi Pesan
Keamanan	REDSHIFT-SERVERLESS-ACARA-1015	ERROR	Namespace Anda <namespace name>tidak memiliki workgroup yang melekat padanya. Amazon Redshift hanya dapat memutar rahasia untuk ruang nama dengan grup kerja yang melekat padanya.
Keamanan	REDSHIFT-SERVERLESS-ACARA-1016	INFO	<namespace name><time in UTC>Kredensi admin diperbarui untuk namespace Anda di.
Keamanan	REDSHIFT-SERVERLESS-ACARA-1030	INFO	<time in UTC>Operasi untuk mendaftarkan namespace Amazon Redshift Anda <namespace name>ke <account id>akun Glue Data Catalog dimulai pada.

Kategori Amazon Redshift	ID Peristiwa Eksternal	Tingkat Keparahan Acara	Deskripsi Pesan
Keamanan	REDSHIFT-SERVERLESS-ACARA-1031	INFO	Operasi untuk mendaftarkan namespace Amazon Redshift Anda <namespace name>ke <account id>akun Glue Data Catalog telah berhasil diselesaikan pada <waktu di UTC>.
Keamanan	REDSHIFT-SERVERLESS-ACARA-1032	INFO	<time in UTC>Operasi untuk mendaftarkan namespace Amazon Redshift Anda <namespace name>ke <account id>akun Glue Data Catalog telah gagal di..
Keamanan	REDSHIFT-SERVERLESS-ACARA-1033	INFO	<account id><time in UTC>Operasi untuk membatalkan pendaftaran namespace Amazon Redshift Anda <namespace name>dari akun Glue Data Catalog dimulai pada.

Kategori Amazon Redshift	ID Peristiwa Eksternal	Tingkat Keparahan Acara	Deskripsi Pesan
Keamanan	REDSHIFT-SERVERLESS-ACARA-1034	INFO	<account id><time in UTC>Operasi untuk membatalkan pendaftaran namespace Amazon Redshift Anda <namespace name>dari akun Glue Data Catalog telah berhasil diselesaikan di.
Keamanan	REDSHIFT-SERVERLESS-ACARA-1035	INFO	<account id><time in UTC>Operasi untuk membatalkan pendaftaran namespace Amazon Redshift Anda <namespace name>dari akun Glue Data Catalog telah gagal di.
Manajemen	REDSHIFT-SERVERLESS-ACARA-1036	ERROR	<workgroup name>Pembaruan trek yang diprakarsai pelanggan gagal di grup kerja Redshift Serverless Anda. Mengembalikan workgroup kembali ke trek aslinya.

Kategori Amazon Redshift	ID Peristiwa Eksternal	Tingkat Keparahannya Acara	Deskripsi Pesan
Manajemen	REDSHIFT-SERVERLESS-ACARA-1037	ERROR	<workgroup name>Pembaruan trek gagal di grup kerja Redshift Serverless Anda. Pembaruan trek gagal untuk grup kerja Amazon Redshift Serverless karena workgroup sibuk pada saat itu. Akibatnya, workgroup telah dikembalikan ke trek aslinya. Untuk mencoba lagi pembaruan trek, harap tunggu waktu aktivitas yang lebih rendah di grup kerja, lalu coba perbarui trek lagi.
Manajemen	REDSHIFT-SERVERLESS-ACARA-1038	INFO	<workgroup name>Trek workgroup Amazon Redshift Anda telah dimodifikasi. Perubahan trek selesai.

Kategori Amazon Redshift	ID Peristiwa Eksternal	Tingkat Keparahan Acara	Deskripsi Pesan
Namespace	REDSHIFT-SERVERLESS-ACARA-1039	ERROR	Namespace <namespace name>memiliki izin yang tidak valid AWS KMS key . Namespace Anda akan memasukkan status “INACCESSIBLE_KMS_KEY” dan Anda tidak akan lagi dapat mengakses nya.
Namespace	REDSHIFT-SERVERLESS-ACARA-1040	INFO	Namespace <namespace name>kini telah dikembalikan ke keadaan sebelumnya karena AWS KMS key izin telah dipulihkan.
Namespace	REDSHIFT-SERVERLESS-ACARA-1045	INFO	<namespace name>Datashare <datashare name>di namespace Redshift Tanpa Server Anda dijatuhkan karena pemulihan tanpa server.

Kategori Amazon Redshift	ID Peristiwa Eksternal	Tingkat Keparahan Acara	Deskripsi Pesan
Namespace	REDSHIFT-SERVERLESS-ACARA-1046	INFO	<namespace name>Konsumen <consumer principle id>untuk perawatan data <datashare name>di namespace Redshift Tanpa Server Anda dicabut karena pemulihan tanpa server.
Namespace	REDSHIFT-SERVERLESS-ACARA-1047	INFO	<namespace name>Aksesibilitas publik diubah untuk datashare di <datashare name>namespace Redshift Tanpa Server Anda karena pemulihan tanpa server.

Pemberitahuan acara integrasi nol-ETL dengan Amazon EventBridge

Integrasi nol-ETL menggunakan Amazon EventBridge untuk mengelola pemberitahuan acara agar Anda tetap memperhatikan perubahan dalam integrasi up-to-date Anda. Amazon EventBridge adalah layanan bus acara tanpa server yang dapat Anda gunakan untuk menghubungkan aplikasi Anda dengan data dari berbagai sumber. Dalam hal ini, sumber acara adalah Amazon Redshift. Peristiwa, yang merupakan perubahan yang dipantau di lingkungan, dikirim ke gudang EventBridge data Amazon Redshift Anda secara otomatis. Acara disampaikan dalam waktu dekat.

EventBridge menyediakan lingkungan bagi Anda untuk menulis aturan acara, yang dapat menentukan tindakan yang akan diambil untuk acara tertentu. Anda juga dapat mengatur target, yang merupakan sumber daya yang EventBridge dapat mengirim acara ke. Target dapat mencakup tujuan API, grup CloudWatch log Amazon, dan lainnya. Untuk informasi selengkapnya tentang aturan, lihat [EventBridge Aturan Amazon](#). Untuk informasi selengkapnya tentang target, lihat [EventBridge target Amazon](#).

Acara dapat diklasifikasikan ke dalam tingkat keparahan dan kategori. Filter berikut tersedia:

- Pemfilteran sumber daya — Menerima pesan berdasarkan sumber daya yang terkait dengan peristiwa tersebut. Sumber daya termasuk workgroup atau snapshot.
- Pemfilteran jendela waktu - Cakupan peristiwa dalam periode waktu tertentu.
- Pemfilteran kategori - Menerima pemberitahuan acara untuk semua acara dalam kategori tertentu.

Tabel berikut mencakup peristiwa integrasi nol-ETL, dengan metadata tambahan:

Kategori Amazon Redshift	ID Peristiwa Eksternal	Tingkat Keparahan Acara	Deskripsi Pesan
Memantau	PERUBAHAN-INTEGRASI-PERISTIWA-0000	INFO	<time in UTC>Integrasi Nol-ETL Anda <integration name>telah dibuat dan menjadi AKTIF di.
Memantau	PERUBAHAN-INTEGRASI-PERISTIWA-0001	INFO	<time in UTC>Integrasi Nol-ETL Anda <integration name>telah dihapus di.
Memantau	PERUBAHAN-INTEGRASI-PERISTIWA-0002	INFO	<time in UTC>Integrasi Nol-ETL Anda <integration

Kategori Amazon Redshift	ID Peristiwa Eksternal	Tingkat Keparahan Acara	Deskripsi Pesan
			name>telah memulai penghapusan di.
Memantau	PERUBAHAN-INTEGRASI-PERISTIWA-0003	INFO	Integrasi Nol-ETL Anda <integration name>menyinkronkan data transaksional ke gudang data Amazon Redshift.
Memantau	PERUBAHAN-INTEGRASI-PERISTIWA-0004	WARNING	Amazon Redshift tidak dapat mereplikasi tabel karena tabel tidak memiliki kunci utama. Tambahkan kunci primer ke tabel sumber dan tabel akan melakukan sinkronisasi ulang secara otomatis.

Kategori Amazon Redshift	ID Peristiwa Eksternal	Tingkat Keparahan Acara	Deskripsi Pesan
Memantau	PERUBAHAN-INTEGRASI-PERISTIWA-0005	WARNING	Amazon Redshift tidak dapat mereplikasi tabel karena satu atau beberapa kolom menggunakan tipe data yang tidak didukung. <schema name>Ubah integrasi Anda untuk mengecualikan tabel ini di filter, atau jatuhkan kolom dari tabel sumber dan jalankan 'ALTER DATABASE <Redshift database name>INTEGRATION REFRESH TABLE. <table name>' untuk menyinkronkan tabel ini.
Memantau	PERUBAHAN-INTEGRASI-PERISTIWA-0006	ERROR	Tidak dapat membuat integrasi. Hapus dan buat ulang integrasi . Jika kesalahan berlanjut, hubungi AWS Support.

Kategori Amazon Redshift	ID Peristiwa Eksternal	Tingkat Keparahan Acara	Deskripsi Pesan
Memantau	PERUBAHAN-INTEGRASI-PERISTIWA-0007	ERROR	Tidak dapat memuat data karena kesalahan internal. Hapus dan buat ulang integrasi. Jika kesalahan berlanjut, hubungi AWS Support.
Memantau	PERUBAHAN-INTEGRASI-PERISTIWA-0008	ERROR	Otorisasi gagal karena izin telah dicabut dari cluster DB sumber. Periksa izin kunci KMS jika Anda menggunakan kunci yang dikelola pelanggan (CMK) untuk mengenkripsi integrasi. Kemudian, hapus dan buat ulang integrasi.
Memantau	PERUBAHAN-INTEGRASI-PERISTIWA-0009	ERROR	Tidak dapat mengirim data ke Amazon Redshift karena jumlah tabel dan skema melebihi batas Amazon Redshift. Hapus dan buat ulang integrasi.

Kategori Amazon Redshift	ID Peristiwa Eksternal	Tingkat Keparahan Acara	Deskripsi Pesan
Memantau	PERUBAHAN-INTEGRASI-PERISTIWA-0012	ERROR	Pemulihan dari titik pemulihan dipanggil pada namespace tanpa server tujuan. Hapus dan buat ulang integrasi.
Memantau	PERUBAHAN-INTEGRASI-PERISTIWA-0013	INFO	<integration name>Integrasi Nol-ETL Anda sekarang AKTIF.
Memantau	PERUBAHAN-INTEGRASI-PERISTIWA-0014	ERROR	<integration name>Integrasi Anda dalam status GAGAL karena kami tidak dapat memodifikasinya karena kesalahan internal. Hapus dan buat ulang integrasi . Jika kesalahan berlanjut, hubungi AWS Support.
Operasi	PERUBAHAN-INTEGRASI-PERISTIWA-0016	INFO	Integrasi nol-ETL Anda <integration name>sedang memproses permintaan modifikasi.

Kategori Amazon Redshift	ID Peristiwa Eksternal	Tingkat Keparahan Acara	Deskripsi Pesan
Operasi	PERUBAHAN-INTEGRASI-PERISTIWA-0017	INFO	<integration name>Modifikasi Anda untuk integrasi nol-ETL telah diterapkan.
Operasi	PERUBAHAN-INTEGRASI-PERISTIWA-0018	WARNING	Target cluster Amazon Redshift sedang dijeda. Tunggu cluster dijeda dan kemudian lanjutkan cluster untuk melanjutkan streaming data.
Operasi	PERUBAHAN-INTEGRASI-PERISTIWA-0019	WARNING	Target kluster Amazon Redshift dijeda. Cluster harus dilanjutkan untuk melanjutkan streaming data.
Operasi	PERUBAHAN-INTEGRASI-PERISTIWA-0020	WARNING	Target cluster Amazon Redshift sedang dilanjutkan. Tunggu cluster aktif untuk melanjutkan streaming data.

Kategori Amazon Redshift	ID Peristiwa Eksternal	Tingkat Keparahan Acara	Deskripsi Pesan
Konfigurasi	PERUBAHAN-INTEGRASI-PERISTIWA-1000	ERROR	Satu atau lebih parameter pada cluster Aurora DB sumber salah dikonfigurasi. Perbaiki grup parameter dan reboot cluster untuk menerapkan perubahan, lalu buat ulang integrasi.
Konfigurasi	PERGESERAN MERAH-INTEGRASI-PERISTIWA-1001	ERROR	Integrasi gagal karena nilai parameter <code>enable_case_sensitive_identifier</code> salah. Tetapkan nilai ke <code>true</code> untuk cluster Aurora DB sumber, lalu hapus dan buat ulang integrasi.
Konfigurasi	PERGESERAN MERAH-INTEGRASI-PERISTIWA-1002	ERROR	Integrasi gagal karena nilai parameter <code>cdc_insert_enabled</code> salah. Tetapkan nilai ke <code>true</code> untuk cluster Aurora DB sumber, lalu hapus dan buat ulang integrasi.

Kategori Amazon Redshift	ID Peristiwa Eksternal	Tingkat Keparahan Acara	Deskripsi Pesan
Konfigurasi	PERGESERAN MERAH-INTEGRASI-PERISTIWA-1003	ERROR	Parameter binlog_format dalam kelompok parameter cluster DB sumber harus disetel ke ROW. Perbaiki grup parameter dan reboot cluster untuk menerapkan perubahan, lalu buat ulang integrasi.
Konfigurasi	PERGESERAN MERAH-INTEGRASI-PERISTIWA-1004	ERROR	Tidak dapat memuat data karena parameter cluster binlog_transaction_compression diaktifkan. Setel nilai parameter ke OFF dan reboot instance writer untuk menerapkan perubahan, lalu buat ulang integrasi.

Kategori Amazon Redshift	ID Peristiwa Eksternal	Tingkat Keparahan Acara	Deskripsi Pesan
Konfigurasi	PERGESERAN MERAH-INTEGRASI-PERISTIWA-1005	ERROR	Tidak dapat memuat data karena parameter cluster <code>binlog_row_value_options</code> disetel ke <code>PARTIAL_JSON</code> , yang tidak didukung. Perbaiki grup parameter dan reboot instance penulis untuk menerapkan perubahan, lalu buat ulang integrasi.
Konfigurasi	PERGESERAN MERAH-INTEGRASI-PERISTIWA-1006	WARNING	Tidak dapat mengurai filter integrasi. Perbaiki sintaks filter.

Kuota dan batas di Amazon Redshift

Amazon Redshift memiliki kuota yang membatasi penggunaan beberapa sumber daya di AWS akun Anda per Wilayah. AWS Ada nilai default untuk setiap kuota dan beberapa kuota dapat disesuaikan.

Kuota untuk objek Amazon Redshift

Amazon Redshift memiliki kuota yang membatasi penggunaan beberapa jenis objek. Ada nilai default untuk masing-masing.

Nama kuota	AWS nilai default	Dapat disesuaikan	Deskripsi
AWS akun yang dapat Anda otorisasi untuk memulihkan snapshot per snapshot	20	Tidak	Jumlah maksimum AWS akun yang dapat Anda otorisasi untuk memulihkan snapshot, per snapshot.
AWS akun yang dapat Anda otorisasi untuk memulihkan snapshot per AWS KMS key	100	Tidak	Jumlah maksimum AWS akun yang dapat Anda otorisasi untuk memulihkan snapshot, per kunci KMS. Artinya, jika Anda memiliki 10 snapshot yang dienkripsi dengan satu kunci KMS, maka Anda dapat mengotorisasi 10 AWS akun untuk memulihkan setiap snapshot, atau kombinasi lain yang menambahkan hingga 100 akun dan tidak melebihi 20 akun untuk setiap snapshot.
Peran IAM cluster	50 ¹	Tidak	Jumlah maksimum peran IAM yang dapat Anda kaitkan dengan kluster untuk mengotorisasi Amazon

Nama kuota	AWS nilai default	Dapat disesuaikan	Deskripsi
untuk Amazon Redshift untuk mengakses layanan lain AWS			Redshift untuk mengakses layanan AWS lain bagi pengguna yang memiliki peran klaster dan IAM. ¹ Kuota adalah 10 sebagai berikut Wilayah AWS: us-iso-east -1, -1, us-iso-west us-isob-east -1.
Tingkat konkurensi (slot kueri) untuk semua antrian WLM manual yang ditentukan pengguna	50	Tidak	Slot kueri maksimum untuk semua antrian yang ditentukan pengguna yang ditentukan oleh manajemen beban kerja manual.
Cluster penskalaan konkurensi	10	Ya	Jumlah maksimum cluster penskalaan konkurensi.
DC2 node dalam sebuah cluster	128	Ya	Jumlah maksimum DC2 node yang dapat Anda alokasikan ke cluster. Untuk informasi selengkapnya tentang batas node untuk setiap jenis node, lihat Cluster dan node di Amazon Redshift .
Langganan kejadian	20	Ya	Jumlah maksimum langganan acara untuk akun ini di AWS Wilayah saat ini.
Node	200	Ya	Jumlah maksimum node di semua instance database untuk akun ini di AWS Wilayah saat ini.

Nama kuota	AWS nilai default	Dapat disesuaikan	Deskripsi
Grup parameter	20	Tidak	Jumlah maksimum grup parameter untuk akun ini di AWS Wilayah saat ini.
RA3 node dalam sebuah cluster	128	Ya	Jumlah maksimum RA3 node yang dapat Anda alokasikan ke cluster. Untuk informasi selengkapnya tentang batas node untuk setiap jenis node, lihat Cluster dan node di Amazon Redshift .
Titik akhir VPC yang dikelola Redshift terhubung ke klaster	30	Ya	Jumlah maksimum titik akhir VPC yang dikelola RedShift yang dapat Anda sambungkan ke cluster. Untuk informasi selengkapnya tentang titik akhir VPC yang dikelola RedShift, lihat. Titik akhir VPC yang dikelola Redshift
Penerima hibah untuk cluster yang diakses melalui titik akhir VPC yang dikelola RedShift	10	Ya	Jumlah maksimum penerima hibah yang dapat diotorisasi oleh pemilik klaster untuk membuat titik akhir VPC yang dikelola RedShift untuk sebuah cluster. Untuk informasi selengkapnya tentang titik akhir VPC yang dikelola RedShift, lihat. Titik akhir VPC yang dikelola Redshift
Titik akhir VPC yang dikelola Redshift per otorisasi	10	Ya	Jumlah maksimum titik akhir VPC yang dikelola RedShift yang dapat Anda buat per otorisasi. Untuk informasi selengkapnya tentang titik akhir VPC yang dikelola RedShift, lihat. Titik akhir VPC yang dikelola Redshift

Nama kuota	AWS nilai default	Dapat disesuaikan	Deskripsi
Simpul terpesan	200	Ya	Jumlah maksimum node cadangan untuk akun ini di AWS Wilayah saat ini.
Skema di setiap database per cluster	9,900	Tidak	Jumlah maksimum skema yang dapat Anda buat di setiap database, per cluster. Namun, pg_temp_* skema tidak diperhitungkan dalam kuota ini.
Grup keamanan	20	Ya	Jumlah maksimum grup keamanan untuk akun ini di AWS Wilayah saat ini.
Ukuran baris tunggal saat memuat dengan COPY	4	Tidak	Ukuran maksimum (dalam MB) dari satu baris saat memuat dengan menggunakan perintah COPY.
Snapshot	700	Ya	Jumlah maksimum snapshot pengguna untuk akun ini di AWS Wilayah saat ini.
Grup Subnet	50	Ya	Jumlah maksimum grup subnet untuk akun ini di AWS Wilayah saat ini.
Subnet dalam grup subnet	20	Ya	Jumlah maksimum subnet untuk grup subnet.

Nama kuota	AWS nilai default	Dapat disesuaikan	Deskripsi
Tabel untuk tipe node large cluster	9,900	Tidak	Jumlah maksimum tabel untuk tipe node cluster besar. Batas ini mencakup tabel permanen, tabel sementara, tabel datashare, dan tampilan terwujud. Tabel eksternal dihitung sebagai tabel sementara. Tabel sementara mencakup tabel sementara yang ditentukan pengguna dan tabel sementara yang dibuat oleh Amazon Redshift selama pemrosesan kueri atau pemeliharaan sistem. Tampilan dan tabel sistem tidak termasuk dalam batas ini.
Tabel untuk tipe node xlarge cluster	9,900	Tidak	Jumlah maksimum tabel untuk tipe node xlarge cluster. Batas ini mencakup tabel permanen, tabel sementara, tabel datashare, dan tampilan terwujud. Tabel eksternal dihitung sebagai tabel sementara. Tabel sementara mencakup tabel sementara yang ditentukan pengguna dan tabel sementara yang dibuat oleh Amazon Redshift selama pemrosesan kueri atau pemeliharaan sistem. Tampilan dan tabel sistem tidak termasuk dalam batas ini.
Tabel untuk tipe node x1plus cluster dengan cluster single-node.	9,900	Tidak	Jumlah maksimum tabel untuk tipe node x1plus cluster dengan cluster single-node. Batas ini mencakup tabel permanen, tabel sementara, tabel datashare, dan tampilan terwujud. Tabel eksternal dihitung sebagai tabel sementara. Tabel sementara mencakup tabel sementara yang ditentukan pengguna dan tabel sementara yang dibuat oleh Amazon Redshift selama pemrosesan kueri atau pemeliharaan sistem. Tampilan dan tabel sistem tidak termasuk dalam batas ini.

Nama kuota	AWS nilai default	Dapat disesuaikan	Deskripsi
Tabel untuk tipe node x1plus cluster dengan cluster multiple-node.	20.000	Tidak	Jumlah maksimum tabel untuk tipe node x1plus cluster dengan cluster multiple-node. Batas ini mencakup tabel permanen, tabel sementara, tabel datashare, dan tampilan terwujud. Tabel eksternal dihitung sebagai tabel sementara. Tabel sementara mencakup tabel sementara yang ditentukan pengguna dan tabel sementara yang dibuat oleh Amazon Redshift selama pemrosesan kueri atau pemeliharaan sistem. Tampilan dan tabel sistem tidak termasuk dalam batas ini.
Tabel untuk tipe node 4xlarge cluster	200.000	Tidak	Jumlah maksimum tabel untuk tipe node 4xlarge cluster. Batas ini mencakup tabel permanen, tabel sementara, tabel datashare, dan tampilan terwujud. Tabel eksternal dihitung sebagai tabel sementara. Tabel sementara mencakup tabel sementara yang ditentukan pengguna dan tabel sementara yang dibuat oleh Amazon Redshift selama pemrosesan kueri atau pemeliharaan sistem. Tampilan dan tabel sistem tidak termasuk dalam batas ini.
Tabel untuk tipe node 8xlarge cluster	200.000	Tidak	Jumlah maksimum tabel untuk tipe node 8xlarge cluster. Batas ini mencakup tabel permanen, tabel sementara, tabel datashare, dan tampilan terwujud. Tabel eksternal dihitung sebagai tabel sementara. Tabel sementara mencakup tabel sementara yang ditentukan pengguna dan tabel sementara yang dibuat oleh Amazon Redshift selama pemrosesan kueri atau pemeliharaan sistem. Tampilan dan tabel sistem tidak termasuk dalam batas ini.

Nama kuota	AWS nilai default	Dapat disesuaikan	Deskripsi
Tabel untuk tipe node 16xlarge cluster	200.000	Tidak	Jumlah maksimum tabel untuk tipe node 16xlarge cluster. Batas ini mencakup tabel permanen, tabel sementara, tabel datashare, dan tampilan terwujud. Tabel eksternal dihitung sebagai tabel sementara. Tabel sementara mencakup tabel sementara yang ditentukan pengguna dan tabel sementara yang dibuat oleh Amazon Redshift selama pemrosesan kueri atau pemeliharaan sistem. Tampilan dan tabel sistem tidak termasuk dalam batas ini.
Jumlah database	60	Tidak	Jumlah maksimum database yang diizinkan dalam cluster Amazon Redshift. Ini tidak termasuk database yang dibuat dari datashares.
Batas waktu untuk sesi idle atau tidak aktif	4 jam	Tidak	Pengaturan ini berlaku untuk cluster. Untuk informasi tentang menyetel nilai batas waktu sesi siaga bagi pengguna, lihat MENGUBAH PENGGUNA di Panduan Pengembang Database Amazon Redshift. Pengaturan pengguna lebih diutamakan daripada pengaturan cluster.
Timeout untuk transaksi idle	6 jam	Tidak	Periode maksimum tidak aktif untuk transaksi terbuka sebelum Amazon Redshift mengakhiri sesi yang terkait dengan transaksi. Pengaturan ini lebih diutamakan daripada setelah batas waktu idle yang ditentukan pengguna. Ini berlaku untuk cluster.
Prosedur tersimpan dalam database	10.000	Tidak	Jumlah maksimum prosedur yang disimpan. Lihat Batas dan perbedaan untuk dukungan prosedur tersimpan untuk batasan lainnya.

Nama kuota	AWSnilai default	Dapat disesuaikan	Deskripsi
Jumlah maksimum koneksi untuk RA3 node	2.000	Tidak	Jumlah maksimum koneksi ke RA3 cluster. Koneksi maksimum yang diizinkan bervariasi menurut jenis node.
Jumlah maksimum koneksi untuk DC2 node	Bervariasi	Tidak	Jumlah maksimum koneksi ke cluster dc2.large adalah 500. Jumlah maksimum koneksi ke cluster dc2.8xlarge adalah 2000.
Jumlah peran Amazon Redshift dalam sebuah cluster	1.000	Ya	Jumlah maksimum peran Amazon Redshift yang dapat Anda buat per cluster. Untuk informasi selengkapnya tentang peran kontrol akses berbasis peran (RBAC), lihat Kontrol akses berbasis peran (RBAC) di Panduan Pengembang Database Amazon Redshift

Kuota untuk objek Amazon Redshift Tanpa Server

Amazon Redshift memiliki kuota yang membatasi penggunaan beberapa jenis objek di namespace Amazon Redshift Tanpa Server Anda. Ada nilai default untuk masing-masing.

Nama kuota	AWSnilai default	Dapat disesuaikan	Deskripsi
Jumlah database	100	Tidak	Jumlah maksimum database yang diizinkan di namespace Amazon Redshift Tanpa Server. Ini tidak termasuk database yang dibuat dari datashares.

Nama kuota	AWS nilai default	Dapat disesuaikan	Deskripsi
Jumlah skema	9,900	Tidak	Jumlah maksimum skema yang diizinkan di namespace Tanpa Server Amazon Redshift.
Jumlah tabel	200.000	Tidak	Jumlah maksimum tabel yang diizinkan di namespace Amazon Redshift Tanpa Server.
Unit penyimpanan Redshift Managed untuk gudang data menggunakan 4 Unit Pemrosesan Redshift () RPU	hingga 32 TB	Tidak	Penyimpanan RMS maksimum yang didukung untuk gudang data dengan 4 RPU. Dalam batas 32 TB, Amazon Redshift menggunakan beberapa ratus penyimpanan MBs internal.

Nama kuota	AWS nilai default	Dapat disesuaikan	Deskripsi
Unit penyimpanan terkelola Redshift untuk gudang data menggunakan an 8, 16, atau 24 Unit Pemrosesan Redshift () RPU	hingga 128 TB	Tidak	Penyimpanan RMS maksimum yang didukung untuk gudang data dengan 8, 16, atau 24 RPU. Dalam batas 128 TB, Amazon Redshift menggunakan beberapa ratus penyimpanan MBs internal.
Batas waktu untuk sesi idle atau tidak aktif	1 jam	Tidak	Untuk informasi tentang menyetel nilai batas waktu sesi siaga bagi pengguna, lihat MENGUBAH PENGGUNA di Panduan Pengembang Database Amazon Redshift. Pengaturan pengguna diutamakan.
Batas waktu untuk kueri yang sedang berjalan	86,399 detik (24 jam)	Tidak	Waktu maksimum untuk kueri yang berjalan sebelum Amazon Redshift mengakhirinya.

Nama kuota	AWS nilai default	Dapat disesuaikan	Deskripsi
Timeout untuk transaksi idle	6 jam	Tidak	Periode maksimum tidak aktif untuk transaksi terbuka sebelum Amazon Redshift Serverless mengakhiri sesi yang terkait dengan transaksi. Pengaturan ini lebih diutamakan daripada setelan batas waktu idle yang ditentukan pengguna.
Jumlah koneksi maksimum	2000	Tidak	Jumlah maksimum koneksi yang diizinkan untuk terhubung ke workgroup.
Jumlah kelompok kerja	25	Ya	Jumlah kelompok kerja yang didukung.
Jumlah ruang nama	25	Ya	Jumlah ruang nama yang didukung.
Jumlah peran Amazon Redshift dalam workgroup	1.000	Ya	Jumlah maksimum peran Amazon Redshift yang dapat Anda buat per workgroup. Untuk informasi selengkapnya tentang peran kontrol akses berbasis peran (RBAC), lihat Kontrol akses berbasis peran (RBAC) di Panduan Pengembang Database Amazon Redshift

Nama kuota	AWS nilai default	Dapat disesuaikan	Deskripsi
Unit Pemrosesan Pergeseran Merah Dasar () per RPU Akun AWS	Lebih besar dari 3.200 RPU atau 1,5 kali basis agregat maksimum Anda RPU dari enam bulan sebelumnya.	Ya	Akun AWS Kuota untuk basis Redshift Tanpa Server lebih besar dari RPU 3.200 atau 1,5 kali RPU basis agregat RPU maksimum Anda dari enam bulan sebelumnya.

Untuk informasi selengkapnya tentang bagaimana penagihan Amazon Redshift Tanpa Server dipengaruhi oleh konfigurasi batas waktu, lihat. [Penagihan untuk Amazon Redshift Tanpa Server](#)

Kuota untuk Amazon Redshift Data API

Amazon Redshift memiliki kuota yang membatasi penggunaan Redshift Data API. Ada nilai default untuk masing-masing. Untuk informasi selengkapnya tentang Amazon Redshift Data API, lihat.

[Menggunakan API Data Amazon Redshift](#)

Nama kuota	AWS nilai default	Dapat disesuaikan	Deskripsi
Transaksi per detik (TPS) untuk API	20	Tidak	Jumlah maksimal permintaan operasi yang dapat Anda lakukan per detik tanpa mengalami throttling.

Nama kuota	AWS nilai default	Dapat disesuaikan	Deskripsi
BatchExecuteStatement			
Transaksi per detik (TPS) untuk API CancelStatement	3	Tidak	Jumlah maksimal permintaan operasi yang dapat Anda lakukan per detik tanpa mengalami throttling.
Transaksi per detik (TPS) untuk API DescribeStatement	100	Tidak	Jumlah maksimal permintaan operasi yang dapat Anda lakukan per detik tanpa mengalami throttling.
Transaksi per detik (TPS) untuk API DescribeTable	3	Tidak	Jumlah maksimal permintaan operasi yang dapat Anda lakukan per detik tanpa mengalami throttling.
Transaksi per detik (TPS) untuk API ExecuteStatement	30	Tidak	Jumlah maksimal permintaan operasi yang dapat Anda lakukan per detik tanpa mengalami throttling.

Nama kuota	AWS nilai default	Dapat disesuaikan	Deskripsi
Transaksi per detik (TPS) untuk API <code>GetStatementResult</code>	20	Tidak	Jumlah maksimal permintaan operasi yang dapat Anda lakukan per detik tanpa mengalami throttling.
Transaksi per detik (TPS) untuk API <code>ListDatabases</code>	3	Tidak	Jumlah maksimal permintaan operasi yang dapat Anda lakukan per detik tanpa mengalami throttling.
Transaksi per detik (TPS) untuk API <code>ListSchemas</code>	3	Tidak	Jumlah maksimal permintaan operasi yang dapat Anda lakukan per detik tanpa mengalami throttling.
Transaksi per detik (TPS) untuk API <code>ListStatements</code>	3	Tidak	Jumlah maksimal permintaan operasi yang dapat Anda lakukan per detik tanpa mengalami throttling.

Nama kuota	AWSnilai default	Dapat disesuaikan	Deskripsi
Transaksi per detik (TPS) untuk API ListTables	3	Tidak	Jumlah maksimal permintaan operasi yang dapat Anda lakukan per detik tanpa mengalami throttling.

Kuota untuk objek editor kueri v2

Amazon Redshift memiliki kuota yang membatasi penggunaan beberapa jenis objek di editor kueri Amazon Redshift v2 Anda. Ada nilai default untuk masing-masing.

Nama kuota	AWSnilai default	Dapat disesuaikan	Deskripsi
Koneksi	500	Ya	Jumlah maksimum koneksi yang dapat Anda buat menggunakan editor kueri v2 di akun ini di Wilayah saat ini.
Prinsipal aktif per akun	50	Ya	Jumlah maksimum prinsipal simultan yang dapat menggunakan editor kueri v2 di akun ini di Wilayah saat ini.
Kueri tersimpan	2.500	Ya	Jumlah maksimum kueri tersimpan yang dapat Anda buat menggunakan editor kueri v2 di akun ini di Wilayah saat ini.
Versi kueri	20	Ya	Jumlah maksimum versi per kueri yang dapat Anda buat menggunakan editor kueri v2 di akun ini di Wilayah saat ini.

Nama kuota	AWS nilai default	Dapat disesuaikan	Deskripsi
Bagan tersimpan	500	Ya	Jumlah maksimum bagan tersimpan yang dapat Anda buat menggunakan editor kueri v2 di akun ini di Wilayah saat ini.
Ukuran data diambil per kueri	100	Tidak	Ukuran maksimum, dalam megabyte, dari data yang diambil per kueri oleh editor kueri v2 di akun ini di Wilayah saat ini.
Data diambil ukuran per 100 baris	5	Tidak	Ukuran maksimum, dalam megabyte, dari data yang diambil per satu halaman hasil kueri (100 baris) oleh editor kueri v2 di akun ini dan Wilayah saat ini.
Koneksi bersamaan maksimum	3	Tidak	Koneksi database maksimum per pengguna (termasuk sesi terisolasi). Nilai ini dapat diatur dari 1-10 oleh administrator editor kueri v2 di Pengaturan akun. Jika Anda mencapai batas yang ditetapkan oleh administrator Anda, pertimbangkan untuk menggunakan sesi bersama alih-alih sesi terisolasi saat menjalankan SQL Anda. Untuk informasi selengkapnya tentang koneksi, lihat Membuka editor kueri v2 . Untuk informasi selengkapnya tentang menetapkan batas, lihat Pengaturan akun .

Kuota dan batas untuk objek Amazon Redshift Spectrum

Amazon Redshift Spectrum memiliki kuota dan batasan berikut:

- Jumlah maksimum database per AWS akun saat menggunakan file AWS Glue Data Catalog. Untuk nilai ini, lihat [kuota AWS Glue layanan](#) di. Referensi Umum Amazon Web Services
- Jumlah maksimum tabel per database saat menggunakan file AWS Glue Data Catalog. Untuk nilai ini, lihat [kuota AWS Glue layanan](#) di. Referensi Umum Amazon Web Services

- Jumlah maksimum partisi per tabel saat menggunakan file AWS Glue Data Catalog. Untuk nilai ini, lihat [kuota AWS Glue layanan](#) di. Referensi Umum Amazon Web Services
- Jumlah maksimum partisi per AWS akun saat menggunakan file AWS Glue Data Catalog. Untuk nilai ini, lihat [kuota AWS Glue layanan](#) di. Referensi Umum Amazon Web Services
- Jumlah maksimum kolom untuk tabel eksternal saat menggunakan AWS Glue Data Catalog, 1.597 saat pseudocolumns diaktifkan, dan 1.600 saat pseudocolumns tidak diaktifkan.
- Ukuran maksimum nilai string dalam file ION atau JSON saat menggunakan AWS Glue Data Catalog adalah 16 KB. String dapat dipotong jika Anda mencapai batas ini.
- Anda dapat menambahkan maksimal 100 partisi menggunakan pernyataan ALTER TABLE tunggal.
- Semua data S3 harus berada di AWS Wilayah yang sama dengan cluster Amazon Redshift.
- [Stempel waktu dalam ION dan JSON harus menggunakan ISO86 format 01.](#)
- Kompresi eksternal file ORC tidak didukung.
- Teks, OpenCSV, dan SERDEs Regex tidak mendukung pembatas oktal yang lebih besar dari '\177'.
- Anda harus menentukan predikat pada kolom partisi untuk menghindari pembacaan dari semua partisi.

Misalnya, filter predikat berikut pada kolom `ship_dtm`, tetapi tidak menerapkan filter ke kolom `ship_yyyymm` partisi:

```
WHERE ship_dtm > '2018-04-01'.
```

Untuk melewati partisi yang tidak dibutuhkan, Anda perlu menambahkan predikat. `WHERE ship_yyyymm = '201804'` Predikat ini membatasi operasi baca ke `\ship_yyyymm=201804\` partisi.

Batasan ini tidak berlaku untuk metastore Apache Hive.

Kendala penamaan

Tabel berikut menjelaskan batasan penamaan dalam Amazon Redshift.

Pengidentifikasi kluster

- Sebuah identifier cluster harus berisi hanya karakter huruf kecil.

	• Pengidentifikasi ini harus berisi 1-63 karakter alfanumerik atau tanda hubung. • Karakter pertamanya harus berupa huruf. • Pengidentifikasi ini tidak boleh diakhiri dengan tanda hubung atau mengandung dua tanda hubung berturut-turut. • Itu harus unik untuk semua cluster dalam AWS akun.
Nama basis data	• Nama database harus berisi 1-64 karakter alfanumerik. • Itu harus berisi hanya huruf kecil. • Itu tidak bisa menjadi kata yang dicadangkan. Untuk daftar kata yang dicadangkan, lihat Kata-kata yang dicadangkan di Panduan Pengembang Database Amazon Redshift.
Nama titik akhir dari titik akhir VPC yang dikelola RedShift	• Nama endpoint harus berisi 1—30 karakter. • Karakter yang valid adalah A-Z, a-z, 0-9, dan tanda hubung (-). • Karakter pertama harus berupa surat. • Nama tidak dapat berisi dua tanda hubung berturut-turut atau diakhiri dengan tanda hubung.

Nama pengguna admin	• Nama pengguna admin harus hanya berisi karakter huruf kecil. • Itu harus berisi 1-128 karakter alfanumerik. • Karakter pertamanya harus berupa huruf. • Itu tidak bisa menjadi kata yang dicadangkan. Untuk daftar kata yang dicadangkan, lihat Kata-kata yang dicadangkan di Panduan Pengembang Database Amazon Redshift.
Kata sandi admin	• Kata sandi admin harus berisi 8-64 karakter. • Itu harus berisi setidaknya satu huruf besar. • Itu harus berisi setidaknya satu huruf kecil. • Itu harus berisi satu nomor. • Ini dapat menggunakan karakter ASCII apa pun dengan kode ASCII 33—126, kecuali '(kutipan tunggal), "(kutipan ganda), \, /, atau @.
Nama grup parameter	• Nama grup parameter harus berisi 1-255 karakter alfanumerik atau tanda hubung. • Itu harus berisi hanya karakter huruf kecil. • Karakter pertama harus berupa huruf. • Pengidentifikasi ini tidak boleh diakhiri dengan tanda hubung atau mengandung dua tanda hubung berturut-turut.

Nama grup keamanan klaster	• Nama grup keamanan klaster harus berisi tidak lebih dari 255 karakter alfanumerik atau tanda hubung. • Itu harus berisi hanya karakter huruf kecil. • Itu tidak harus Default. • Ini harus unik untuk semua grup keamanan yang dibuat oleh AWS akun Anda.
Nama grup subnet	• Nama grup subnet harus berisi tidak lebih dari 255 karakter alfanumerik atau tanda hubung. • Itu harus berisi hanya karakter huruf kecil. • Itu tidak harus Default. • Ini harus unik untuk semua grup subnet yang dibuat oleh AWS akun Anda.
Pengidentifikasi snapshot cluster	• Pengidentifikasi snapshot cluster harus berisi tidak lebih dari 255 karakter alfanumerik atau tanda hubung. • Itu harus berisi hanya karakter huruf kecil. • Itu tidak harus Default. • Itu harus unik untuk semua pengidentifikasi snapshot yang dibuat oleh akun Anda AWS.

Menandai sumber daya di Amazon Redshift

Dalam AWS, tag adalah label yang ditentukan pengguna yang terdiri dari pasangan nilai kunci. Amazon Redshift mendukung penandaan untuk memberikan metadata tentang sumber daya secara sekilas, dan untuk mengkategorikan laporan penagihan berdasarkan alokasi biaya. Untuk menggunakan tag untuk alokasi biaya, Anda harus terlebih dahulu mengaktifkan tag tersebut di AWS Manajemen Penagihan dan Biaya layanan. Untuk informasi selengkapnya tentang menyiapkan dan menggunakan tag untuk tujuan penagihan, lihat [Menggunakan tag alokasi biaya untuk laporan penagihan khusus dan Menyiapkan laporan alokasi biaya bulanan Anda](#).

Tag tidak diperlukan untuk sumber daya di Amazon Redshift, tetapi tag membantu memberikan konteks. Anda mungkin ingin menandai sumber daya dengan metadata tentang pusat biaya, nama proyek, dan informasi terkait lainnya yang terkait dengan sumber daya. Misalnya, Anda ingin melacak sumber daya mana yang termasuk dalam lingkungan pengujian dan lingkungan produksi. Anda dapat membuat kunci bernama `environment` dan memberikan nilai `test` atau `production` untuk mengidentifikasi sumber daya yang digunakan di setiap lingkungan. Jika Anda menggunakan penandaan di AWS layanan lain atau memiliki kategori standar untuk bisnis Anda, kami sarankan Anda membuat pasangan nilai kunci yang sama untuk sumber daya di Amazon Redshift untuk konsistensi.

Tag disimpan untuk sumber daya setelah Anda mengubah ukuran klaster, dan setelah Anda memulihkan snapshot klaster dalam wilayah yang sama. Namun, tag tidak dipertahankan jika Anda menyalin snapshot ke wilayah lain, jadi Anda harus membuat ulang tag di wilayah baru. Jika Anda menghapus sumber daya, tag terkait akan dihapus.

Setiap sumber daya memiliki satu set tag, yang merupakan kumpulan dari satu atau beberapa tag yang ditetapkan ke sumber daya. Setiap sumber daya dapat memiliki hingga 50 tag per set tag. Anda dapat menambahkan tag saat membuat sumber daya dan setelah sumber daya dibuat. Anda dapat menambahkan tag ke jenis sumber daya berikut di Amazon Redshift:

- CIDR/IP
- Klaster
- Grup keamanan klaster
- Aturan masuknya grup keamanan klaster
- Grup EC2 keamanan Amazon
- Koneksi modul keamanan perangkat keras (HSM)

- Sertifikat klien HSM
- Grup parameter
- Snapshot
- Grup subnet
- Integrasi (integrasi nol-ETL)

Untuk menggunakan penandaan dari konsol Amazon Redshift, pengguna Anda dapat melampirkan kebijakan terkelola AWS. `AmazonRedshiftFullAccess` Untuk contoh kebijakan IAM dengan izin penandaan terbatas yang dapat Anda lampirkan ke pengguna konsol Amazon Redshift, lihat. [Contoh 7: Izinkan pengguna menandai sumber daya dengan konsol Amazon Redshift](#) Untuk informasi selengkapnya tentang penandaan, lihat [Apa itu AWS Resource Groups?](#) .

Persyaratan penandaan

Tag memiliki persyaratan sebagai berikut:

- Kunci tidak dapat diawali dengan `aws :`.
- Kunci harus unik per set tag.
- Kunci harus antara 1 dan 128 karakter yang diizinkan.
- Nilai harus antara 0 dan 256 karakter yang diizinkan.
- Nilai tidak harus unik per set tag.
- Karakter yang diizinkan untuk kunci dan nilai adalah huruf Unicode, digit, spasi putih, dan salah satu simbol berikut: `_ . : / = + - @`.
- Kunci dan nilai peka huruf besar dan kecil.

Mengelola tag sumber daya

Prosedur berikut memandu Anda melalui cara bekerja dengan tag sumber daya Anda.

Untuk mengelola tag di sumber daya Amazon Redshift Anda

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon Redshift di. <https://console.aws.amazon.com/redshiftv2/>
2. Pada menu navigasi, pilih Konfigurasi, lalu pilih Kelola tag.

3. Masukkan pilihan Anda untuk sumber daya dan pilih tag mana yang akan ditambahkan, dimodifikasi, atau dihapus. Kemudian pilih Kelola tag sumber daya yang Anda pilih.

Sumber daya yang dapat Anda tag termasuk cluster, grup parameter, grup subnet, sertifikat klien HSM, koneksi HSM, dan snapshot.

4. Pada halaman navigasi Kelola tag, pilih Tinjau dan terapkan perubahan tag, lalu pilih Terapkan untuk menyimpan perubahan Anda.

AWS Backup integrasi dengan Amazon Redshift

AWS Backup adalah layanan yang dikelola sepenuhnya yang membantu Anda memusatkan dan mengotomatiskan perlindungan data di seluruh AWS layanan, di cloud, dan di tempat.

Menggunakan AWS Backup Amazon Redshift, Anda dapat mengonfigurasi kebijakan perlindungan data dan memantau aktivitas untuk berbagai sumber daya Amazon Redshift di satu tempat. Anda juga dapat membuat dan menyimpan snapshot di kluster yang disediakan Amazon Redshift dan ruang nama tanpa server. Ini memungkinkan Anda mengotomatiskan dan mengkonsolidasikan tugas pencadangan yang harus Anda lakukan secara terpisah sebelumnya, tanpa proses manual apa pun.

Note

Tabel tanpa cadangan tidak didukung untuk kluster yang RA3 disediakan dan grup kerja Amazon Redshift Tanpa Server. Tabel yang ditandai sebagai no-backup di RA3 cluster atau grup kerja tanpa server diperlakukan sebagai tabel permanen yang akan selalu dicadangkan saat mengambil snapshot, dan selalu dipulihkan saat memulihkan dari snapshot. Untuk menghindari biaya snapshot untuk tabel tanpa cadangan, potong sebelum mengambil snapshot.

Pencadangan, atau titik pemulihan, mewakili konten sumber daya, seperti kluster Amazon Redshift, pada waktu tertentu. AWS Backup menyimpan cadangan di brankas cadangan, yang dapat Anda atur sesuai dengan kebutuhan bisnis Anda. Istilah titik pemulihan dan cadangan digunakan secara bergantian. Untuk informasi selengkapnya AWS Backup, lihat [Pembuatan cadangan, pemeliharaan, dan pemulihan](#) di Panduan AWS Backup Pengembang.

Amazon Redshift terintegrasi secara native dengan AWS Backup. Itu memungkinkan Anda menentukan paket cadangan dan menetapkan sumber daya Amazon Redshift ke paket cadangan. AWS Backup mengotomatiskan pembuatan snapshot manual Amazon Redshift, dan menyimpan snapshot ini dengan aman di brankas cadangan yang Anda tentukan dalam paket cadangan Anda. Untuk informasi tentang vault, lihat [Backup vault](#) di Panduan Pengembang. AWS Backup Dalam paket cadangan, Anda dapat menentukan frekuensi cadangan, jendela cadangan, siklus hidup, atau brankas cadangan. Untuk informasi tentang paket cadangan, lihat [Paket cadangan](#) di Panduan AWS Backup Pengembang.

Untuk informasi tentang membuat dan memulihkan snapshot Amazon Redshift Tanpa Server tanpa menggunakan, lihat [AWS Backup Snapshot dan titik pemulihan](#) Untuk informasi tentang membuat

dan memulihkan snapshot klaster yang disediakan Amazon Redshift tanpa menggunakan `awscli`. AWS Backup [Cuplikan dan cadangan Amazon Redshift](#)

Topik

- [Pertimbangan untuk digunakan AWS Backup dengan Amazon Redshift](#)
- [Batasan untuk digunakan AWS Backup dengan Amazon Redshift](#)
- [Mengelola AWS Backup dengan Amazon Redshift](#)

Pertimbangan untuk digunakan AWS Backup dengan Amazon Redshift

Berikut ini adalah pertimbangan untuk digunakan AWS Backup dengan Amazon Redshift:

- AWS Backup untuk Amazon Redshift tersedia di mana keduanya AWS Backup dan Amazon Redshift tersedia dalam hal yang sama. Wilayah AWS Untuk informasi tentang tempat AWS Backup tersedia, lihat [titik akhir Amazon Redshift dan](#) kuota di. Referensi Umum AWS
- Untuk mulai menggunakan AWS Backup, verifikasi bahwa Anda telah memenuhi semua prasyarat. Untuk informasi selengkapnya, lihat [Prasyarat](#) di Panduan Pengembang AWS Backup
- Secara afirmatif memilih untuk AWS Backup layanan. Pilihan keikutsertaan berlaku untuk akun tertentu dan AWS Region. Jika Anda ingin menggunakan cadangan di beberapa Wilayah dengan akun tertentu, Anda harus memilih masing-masing Wilayah dengan akun tersebut. Untuk informasi selengkapnya, lihat [Memilih untuk mengelola layanan AWS Backup](#) di Panduan AWS Backup Pengembang.
- AWS Backup integrasi untuk Amazon Redshift hanya mendukung snapshot manual untuk cluster yang disediakan dan ruang nama tanpa server.
- Setelah Anda menggunakan AWS Backup untuk mengelola pengaturan snapshot, Anda tidak dapat melanjutkan untuk mengelola pengaturan snapshot manual menggunakan Amazon Redshift. Sebagai gantinya, Anda dapat terus mengelola pengaturan menggunakan AWS Backup paket. Untuk informasi selengkapnya, lihat [Cadangan paket](#) di Panduan AWS Backup Pengembang.
- Memulihkan seluruh snapshot gudang data ke namespace tanpa server adalah perubahan yang merusak. Semua data yang ada sebelumnya di namespace target hilang saat Anda mengembalikan snapshot gudang data ke namespace tersebut. Ini hanya berlaku untuk memulihkan snapshot gudang data. Memulihkan snapshot tabel tunggal ke namespace tidak menghapus data yang ada.

- Untuk memulihkan snapshot ke kluster yang disediakan, Anda harus memiliki kebijakan IAM dengan izin. `RestoreFromClusterSnapshot` Untuk memulihkan snapshot ke namespace tanpa server, Anda harus memiliki kebijakan IAM dengan izin. `RestoreFromSnapshot` Izin ini berlaku untuk tipe gudang data target, bukan untuk jenis snapshot sumber. Misalnya, untuk mengembalikan snapshot cluster ke namespace, Anda memerlukan `RestoreFromSnapshot` izin, bukan. `RestoreFromClusterSnapshot` Untuk informasi selengkapnya tentang mengelola kebijakan IAM, lihat [Manajemen identitas dan akses di Amazon Redshift](#).

Batasan untuk digunakan AWS Backup dengan Amazon Redshift

Berikut ini adalah batasan untuk digunakan AWS Backup dengan Amazon Redshift:

- Anda tidak dapat menggunakannya AWS Backup untuk mengelola snapshot otomatis Amazon Redshift. Untuk mengelola snapshot otomatis, gunakan tag. Untuk informasi tentang penandaan sumber daya, lihat [Menandai sumber daya di Amazon Redshift](#).
- Saat memulihkan tabel tunggal dari snapshot, Anda tidak dapat memulihkan dari snapshot kluster yang disediakan ke namespace tanpa server atau sebaliknya. Anda dapat memulihkan seluruh snapshot dalam konfigurasi apa pun. Misalnya, Anda dapat memulihkan semua database dalam snapshot kluster yang disediakan ke namespace tanpa server, tetapi Anda tidak dapat memulihkan satu tabel dari snapshot yang sama ke namespace yang sama.

Mengelola AWS Backup dengan Amazon Redshift

Untuk melindungi sumber daya di gudang data Amazon Redshift, Anda dapat menggunakan AWS Backup konsol, atau menggunakan API atau () secara terprogram. AWS Backup AWS Command Line Interface AWS CLI Ketika Anda perlu memulihkan sumber daya, Anda dapat menggunakan AWS Backup konsol atau AWS CLI untuk menemukan dan memulihkan sumber daya yang Anda butuhkan. Untuk informasi selengkapnya, lihat [AWS Command Line Interface](#).

Saat menggunakan AWS Backup Amazon Redshift, Anda dapat melakukan tindakan berikut:

- Buat cadangan berkala yang secara otomatis memulai snapshot Amazon Redshift. Pencadangan berkala berguna untuk memenuhi kebutuhan retensi data jangka panjang Anda. Untuk informasi selengkapnya, lihat [cadangan Amazon Redshift di Panduan Pengembang](#).AWS Backup
- Otomatiskan penjadwalan dan retensi cadangan dengan mengonfigurasi rencana cadangan secara terpusat.

- Pulihkan klaster yang disediakan atau namespace tanpa server ke cadangan tersimpan yang Anda pilih. Anda dapat memilih untuk mengembalikan semua data dalam snapshot atau satu tabel darinya. Anda mengatur seberapa sering untuk membuat cadangan sumber daya Anda. Untuk informasi tentang memulihkan snapshot klaster yang disediakan, lihat [Memulihkan klaster Amazon Redshift di Panduan Pengembang.AWS Backup](#) Untuk informasi tentang memulihkan snapshot namespace tanpa server, lihat Amazon [Redshift](#) Pemulihan Tanpa Server di Panduan Pengembang.AWS Backup

Versi cluster untuk Amazon Redshift

Amazon Redshift secara teratur merilis versi cluster. Cluster Amazon Redshift Anda ditambah selama jendela pemeliharaan sistem Anda. Waktu patch tergantung pada pengaturan jendela Anda AWS Region dan pemeliharaan. Anda dapat melihat atau mengubah pengaturan jendela pemeliharaan dari konsol Amazon Redshift. Untuk informasi lebih lanjut tentang pemeliharaan, lihat [Pemeliharaan cluster](#).

Anda dapat melihat versi cluster cluster Anda di konsol Amazon Redshift pada tab Maintenance pada detail cluster. Atau Anda dapat melihat versi cluster di output dari perintah SQL:

```
SELECT version();
```

Note

Pembaruan penting yang memengaruhi perilaku Amazon Redshift diperkenalkan saat Amazon Redshift berkembang. Untuk mengikuti perubahan ini, ambil tindakan, dan hindari potensi gangguan pada beban kerja Anda, lihat. [Perubahan perilaku di Amazon Redshift](#)

Topik

- [Patch Pergeseran Merah Amazon 199](#)
- [Patch Pergeseran Merah Amazon 198](#)
- [Patch Pergeseran Merah Amazon 197](#)
- [Patch Pergeseran Merah Amazon 196](#)
- [Patch Pergeseran Merah Amazon 195](#)
- [Patch Pergeseran Merah Amazon 194](#)
- [Patch Pergeseran Merah Amazon 193](#)
- [Patch Pergeseran Merah Amazon 192](#)
- [Patch Pergeseran Merah Amazon 191](#)
- [Patch Pergeseran Merah Amazon 190](#)
- [Patch Pergeseran Merah Amazon 189](#)
- [Patch Pergeseran Merah Amazon 188](#)
- [Patch Pergeseran Merah Amazon 187](#)

- [Patch Pergeseran Merah Amazon 186](#)
- [Patch Pergeseran Merah Amazon 185](#)
- [Patch Pergeseran Merah Amazon 184](#)
- [Patch Pergeseran Merah Amazon 183](#)
- [Patch Pergeseran Merah Amazon 182](#)
- [Patch Pergeseran Merah Amazon 181](#)
- [Patch Pergeseran Merah Amazon 180](#)
- [Patch Pergeseran Merah Amazon 179](#)
- [Patch Pergeseran Merah Amazon 178](#)
- [Patch Pergeseran Merah Amazon 177](#)
- [Patch Pergeseran Merah Amazon 176](#)
- [Patch Pergeseran Merah Amazon 175](#)
- [Patch Pergeseran Merah Amazon 174](#)
- [Patch Pergeseran Merah Amazon 173](#)
- [Patch Pergeseran Merah Amazon 172](#)
- [Patch Pergeseran Merah Amazon 171](#)
- [Patch Pergeseran Merah Amazon 170](#)
- [Patch Pergeseran Merah Amazon 169](#)
- [Patch Pergeseran Merah Amazon 168](#)

Patch Pergeseran Merah Amazon 199

Versi cluster di tambalan ini:

- 1.0.232773 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 4 Maret 2026

Fitur dan peningkatan baru di tambalan ini

- Merilis [peningkatan kompilasi kueri baru yang meningkatkan](#) kinerja untuk kueri baru (menjalankan kueri pertama) di Redshift CURRENT Maintenance Track. Track TRAILING akan mengikuti rilis patch yang akan datang.

- Dukungan yang diaktifkan untuk hibah tingkat Kolom dari konsumen yang menggunakan izin federasi Redshift untuk penerima hibah baru pada produsen.
- Dukungan yang diaktifkan untuk pemeriksaan izin terbatas di fitur Metadata Security (MDS).
- Izin gabungan Amazon Redshift mendukung kueri tabel eksternal melalui Late Binding Views (). LBVs
- ALTER EXTERNAL SCHEMA sekarang dapat digunakan dengan tipe skema eksternal KINESIS untuk mengubah IAM_ROLE dan REGION dari skema eksternal yang ada.
- Menambahkan dukungan pemulihan untuk tabel NO-BACKUP di gudang data RA3 Redshift dan Tanpa Server. Karena jenis instance ini tidak mendukung fitur NO-BACKUP, tabel yang ditandai sebagai NO-BACKUP selalu disertakan dalam snapshot dan sekarang dipulihkan dengan benar sebagai tabel permanen
- Amazon Redshift sekarang mendukung penulisan ke tabel eksternal saat terhubung menggunakan peran IAM atau pengguna IAM. Sebelumnya, hanya operasi baca pada tabel eksternal yang didukung dengan autentikasi berbasis peran IAM. Dengan pembaruan ini, Redshift menggunakan kredensial sesi IAM Anda dengan benar saat menulis data ke Amazon S3 untuk tabel eksternal, memungkinkan akses baca dan tulis lengkap.
- Untuk aliran Kafka yang menyegarkan secara otomatis MVs, kami sekarang menggunakan heuristik yang lebih efisien untuk menentukan apakah ada catatan baru untuk dicerna. Ini akan mengurangi jumlah data yang ditransfer dari broker Kafka untuk pemeriksaan ini, dan mengurangi latensi saat memulai penyegaran MV berikutnya.
- Operasi pencadangan dan pemulihan yang ditingkatkan untuk menggunakan enkripsi yang sesuai dengan FIPS SHA384 , menggantikan hashing lama MD5
- Peningkatan kinerja untuk kueri yang menggabungkan Fungsi yang Ditetapkan Pengguna Lambda dan pernyataan UNNEST
- Peningkatan kinerja untuk kueri menggunakan Lambda User-Defined Functions terhadap tabel dan tampilan sistem
- Peningkatan kinerja eksekusi kueri saat menggunakan klausa ORDER BY dan LIMIT dengan menghilangkan pemindaian data yang tidak perlu
- Kueri REFRESH otomatis untuk tampilan terwujud Amazon Redshift MVs () sekarang diperlakukan sebagai kueri pengguna, bukan proses otonom latar belakang. Kueri REFRESH otomatis sekarang berjalan dengan prioritas yang sama dengan kueri pengguna lain dan tidak lagi ditangguhkan oleh proses latar belakang otonom. Fitur perubahan perilaku MV Auto REFRESH hanya diaktifkan untuk kluster Amazon Redshift Provisioned pada Track CURRENT dari rilis patch P198 dan yang lebih baru. Saat ini dinonaktifkan di Tanpa Server.

- Penjadwalan yang ditingkatkan untuk tugas otomatis untuk memprioritaskan tugas-tugas penting pengguna dengan lebih baik seperti konsumsi streaming, penyalinan otomatis (COPY JOB), dan penyegaran otomatis tampilan terwujud.
- Memperbaiki masalah saat `pg_last_query_id ()` salah mengembalikan ID kueri yang ditulis ulang alih-alih ID kueri pengguna untuk klaster multi-AZ
- Memperbaiki masalah di mana SHOW GRANTS akan gagal memproses referensi objek satu bagian.
- Memperbaiki masalah saat DROP TABLE bertentangan dengan pekerja otomatis stempel waktu pekerja pembuatan ulang katalog karena pembaruan simultan pada `pg_class_extended`, dengan melewati penghapusan selama DROP TABLE dan memungkinkan pekerja stempel waktu untuk membersihkan pada proses berikutnya.
- Memperbaiki masalah penguraian filter saluran nol-ETL saat database, skema, atau nama tabel berisi kata kunci include/exclude
- Memperbaiki kondisi balapan yang dapat terjadi ketika operasi vakum berjalan bersamaan dengan operasi tabel nol-ETL
- Memperbaiki masalah saat integrasi nol-ETL salah mencoba operasi sinkronisasi ulang alih-alih beralih ke status kesalahan saat konteks enkripsi KMS berubah
- Memperbaiki kondisi balapan dalam integrasi nol-ETL yang dapat terjadi selama restart cluster saat replikasi data aktif
- Perbaikan TAMPILKAN TABEL

Patch Pergeseran Merah Amazon 198

Versi cluster di tambalan ini:

- 1.0.239402 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 12 Maret 2026
- 1.0.227967 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 24 Februari 2026
- 1.0.224252 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 19 Februari 2026
- 1.0.219674 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 11 Februari 2026

- 1.0.212832 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 09 Februari 2026
- 1.0.204436 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 03 Februari 2026

Fitur dan peningkatan baru di tambalan ini

- Merilis [peningkatan kompilasi kueri baru yang meningkatkan](#) kinerja untuk kueri baru (menjalankan kueri pertama) di Redshift CURRENT Maintenance Track. Track TRAILING akan mengikuti rilis patch yang akan datang.
- Meningkatkan pertanyaan INTERSECT dan EXCEPT dengan menerapkan aturan pemeriksaan yang tepat saat membandingkan kolom dengan nilai konstan.
- Mengaktifkan penyegaran otomatis tampilan terwujud yang dibuat secara manual untuk dijalankan pada kluster penskalaan konkurensi.
- Meningkatkan kinerja koneksi dengan mengaktifkan koneksi langsung antara cluster utama dan cluster penskalaan konkurensi, mengurangi latensi dan menghilangkan batasan throughput koneksi.
- Meningkatkan klasifikasi kesalahan untuk integrasi nol-ETL, sehingga lebih mudah untuk mengidentifikasi akar penyebab dan masalah debug.
- Menghapus pesan kesalahan dari SYS_QUERY_HISTORY untuk kueri yang berhasil dieksekusi, menyediakan catatan riwayat kueri yang lebih bersih.
- Meningkatkan akurasi pencatatan waktu kompilasi untuk kueri bursted di SYS_QUERY_HISTORY, memberikan wawasan kinerja yang lebih baik.
- Menambahkan dukungan untuk fungsi array: ARRAY_CONTAINS, ARRAY_POSITION, ARRAYS_OVERLAP, ARRAY_INTERSECTION, ARRAY_SORT, ARRAY_UNION, ARRAY_DISTINCT, ARRAY_EXCEPLE, dan ARRAY_POSITIONS.
- Menambahkan dukungan untuk fungsi GET_NUMBER_ATTRIBUTES yang mengembalikan jumlah atribut dalam objek SUPER.
- Mengaktifkan batas waktu transaksi idle untuk transaksi yang gagal untuk meningkatkan manajemen sumber daya.
- Menambahkan dukungan untuk jenis objek TEMPLATE yang memungkinkan Anda menentukan parameter pemformatan yang dapat digunakan kembali, menghilangkan kebutuhan untuk menentukannya secara manual untuk setiap operasi.

- Meningkatkan kinerja untuk kueri yang memanggil Lambda User-Defined Functions (LUDFs) pada Common Table Expressions (). CTEs
- Meningkatkan kinerja untuk kueri yang memanggil Lambda User-Defined Functions (LUDFs) dalam klausa THEN dari ekspresi CASE.
- Pembuatan Fungsi yang Ditetapkan Pengguna Python baru tidak digunakan lagi. Python yang ada UDFs akan didukung hingga 30 Juni 2026. Migrasi ke Lambda User-Defined Functions LUDFs () untuk dukungan lanjutan.
- Meningkatkan observabilitas untuk Lambda User-Defined Functions LUDFs () dengan tabel sistem baru, SYS_LUDF_DETAIL. Tabel ini mencatat informasi dan metrik untuk LUDFs digunakan dalam kueri.
- Kueri REFRESH otomatis untuk tampilan terwujud Amazon Redshift MVs () sekarang diperlakukan sebagai kueri pengguna, bukan proses otonom latar belakang. Kueri REFRESH otomatis sekarang berjalan dengan prioritas yang sama dengan kueri pengguna lain dan tidak lagi ditangguhkan oleh proses latar belakang otonom. Fitur perubahan perilaku MV Auto REFRESH hanya diaktifkan untuk kluster Amazon Redshift Provisioned pada Track CURRENT dari rilis patch P198 dan yang lebih baru. Saat ini dinonaktifkan di Tanpa Server.

Patch Pergeseran Merah Amazon 197

Versi cluster di tambalan ini:

- 1.0.237947 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 10 Maret 2026
- 1.0.198462 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 21 Januari 2026
- 1.0.194394 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 13 Januari 2026
- 1.0.187928 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 08 Januari 2026
- 1.0.179517 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 15 Desember 2025
- 1.0.176066 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 08 Desember 2025

- 1.0.166219 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 21 November 2025

Fitur dan peningkatan baru di tambalan ini

- Otonomi global untuk cluster di mesh, Autonomics sekarang mempertimbangkan beban kerja seluruh mesh untuk mengoptimalkan Vakum, Analisis dan ATO.
- Membuat dan menyegarkan Tampilan Terwujud dari beberapa gudang data Amazon Redshift.
- Menambahkan dukungan untuk menyimpan string literal hingga 16.000.000 byte dalam tipe data SUPER
- Ditambahkan GET_NUMBER_ATTRIBUTES (super_object) fungsi yang mengembalikan jumlah total atribut yang terkandung dalam objek SUPER.
- Pembatasan kompatibilitas patch rilis diberlakukan untuk operasi pemulihan tabel. Saat mencoba memulihkan tabel, cadangan hanya dapat digunakan pada cluster yang menjalankan versi tambalan yang sama seperti saat cadangan dibuat, satu versi patch lebih rendah dari cadangan, atau versi tambalan apa pun yang lebih tinggi dari cadangan.
- Meningkatkan operator LIKE untuk menangani spasi belakang dengan benar dalam pola saat menggunakan tipe data CHAR.
- Memperkenalkan 4 fungsi H3 spasial baru yang beroperasi pada hierarki sel: H3_Resolution, H3_, H3_ dan ToParent H3_. ToChildren IsValid
- Mengaktifkan Izin Federasi Amazon Redshift yang menyederhanakan pengelolaan izin di beberapa gudang data Redshift dengan memungkinkan Anda menentukan izin data sekali dan secara otomatis menerapkannya di semua gudang di Anda. Akun AWS
- Mengaktifkan hibah tingkat objek, kontrol akses tingkat kolom berbutir halus, hibah izin cakupan, dan hibah tingkat database pada objek Katalog Izin Federasi Amazon Redshift.
- Mengaktifkan operasi kebijakan CREATE, ALTER, ATTACH, DETACH, dan DROP RLS di Katalog Izin Federasi Amazon Redshift.
- Mengaktifkan kemampuan untuk mengaktifkan atau menonaktifkan keamanan tingkat baris untuk relasi di Katalog Izin Federasi Amazon Redshift.
- Aktifkan operasi kebijakan penyembunyian CREATE, ALTER, ATTACH, DETACH, dan DROP pada objek Katalog Izin Federasi Amazon Redshift.
- Mengaktifkan KEBIJAKAN TAMPILKAN untuk menampilkan kebijakan RLS dan Masking pada database yang terhubung dan lampiran pada relasi di objek Katalog Izin Federasi Amazon Redshift.

- Memungkinkan pengguna database untuk mengambil peran IAM untuk melakukan kueri pada objek Katalog Izin Federasi Amazon Redshift.
- Perintah SHOW GRANTS yang disempurnakan untuk mendukung visibilitas hibah lintas basis data dalam cluster Redshift yang sama.
- Menambahkan dukungan untuk perintah SHOW COLUMN GRANTS untuk menyederhanakan penemuan kontrol akses tingkat kolom.
- Menambahkan dukungan untuk perintah SHOW PROCEDURES untuk menyederhanakan penemuan penemuan metadata prosedur tersimpan.
- Menambahkan dukungan untuk perintah SHOW FUNCTIONS untuk menyederhanakan penemuan metadata fungsi yang ditentukan pengguna.
- Memperkenalkan perintah SHOW PARAMETERS untuk menampilkan metadata parameter untuk fungsi dan prosedur yang disimpan.
- Memperkenalkan perintah SHOW CONSTRAINTS untuk menampilkan metadata kendala untuk fungsi dan prosedur yang disimpan.
- Menambahkan dukungan untuk operasi CREATE/CREATE OR REPLACE/DROP VIEW di Amazon Redshift Federated Permissions Catalog dan pada cross-database dalam cluster Redshift yang sama.
- Peningkatan penanganan pemeriksaan string konstan di beberapa pertanyaan INTERSECT atau KECUALI.
- TAMPILKAN TABEL yang disempurnakan untuk menampilkan pesan peringatan alih-alih gagal saat kredensial FAS hilang untuk akses. AWS Glue Data Catalog
- Perintah SHOW COLUMNS yang disempurnakan dengan kolom metadata tambahan untuk kunci pengurutan, gaya distribusi, pengkodean, dan informasi pemeriksaan.
- Menambahkan kolom grantor_name ke output SHOW GRANTS dan penyertaan database_name standar di semua set hasil SHOW GRANTS.
- Perintah SHOW TABLES yang disempurnakan dengan kolom tambahan: table_owner, last_altered_time, last_modified_time, dist_style, dan table_sub_type.
- Dukungan Otentikasi Pusat Identitas: Amazon Redshift sekarang mendukung autentikasi IAM Identity Center melalui operasi API baru. GetIdentityCenterAuthToken Fitur ini memungkinkan akses federasi ke cluster Amazon Redshift dan grup kerja tanpa server menggunakan identitas Pusat Identitas yang ada. Manfaat utama termasuk akses masuk tunggal menggunakan kredensial Pusat Identitas, keamanan yang ditingkatkan dengan token otentikasi terenkripsi, pencatatan audit komprehensif untuk kepatuhan, dan integrasi tanpa batas dengan

pengaturan Pusat Identitas yang ada. Operasi API menghasilkan token otentikasi yang dicakup ke pengidentifikasi kluster atau nama grup kerja tertentu. Persyaratan termasuk kredensial yang disempurnakan identitas dengan informasi identitas Pusat Identitas yang disematkan dan izin IAM yang sesuai untuk akses API.

Patch Pergeseran Merah Amazon 196

Versi cluster di tambalan ini:

- 1.0.179872 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 16 Desember 2025
- 1.0.171165 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 08 Desember 2025
- 1.0.163480 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 19 November 2025
- 1.0.160807 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 13 November 2025
- 1.0.155905 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 10 November 2025

Fitur dan peningkatan baru di tambalan ini

- Mengoptimalkan rencana untuk subquery IN dan EXISTS yang berkorelasi dengan menghiasnya menjadi semi gabungan.
- Memperluas perintah SHOW TABLE untuk menyertakan informasi pemeriksaan untuk setiap kolom.

Patch Pergeseran Merah Amazon 195

Versi cluster di tambalan ini:

- 1.0.172541 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 08 Desember 2025
- 1.0.162991 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 17 November 2025

- 1.0.160706 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 13 November 2025
- 1.0.151179 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 4 November 2025
- 1.0.148180 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 3 November 2025
- 1.0.142459 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 18 Oktober 2025

Fitur dan peningkatan baru di tambalan ini

- Menambahkan dukungan untuk menulis ke tabel Iceberg menggunakan CREATE TABLE, CREATE TABLE AS SELECT, INSERT INTO, perintah SQL.
- Menambahkan dukungan untuk tipe data SUPER dalam database dengan pemeriksaan case-insensitive.
- Memperbarui versi database IANA Time Zone Database ke 2025b.
- Menambahkan dukungan untuk fungsi TZDB_VERSION. Fungsi ini menampilkan versi Database Zona Waktu IANA saat ini yang digunakan.
- Amazon Redshift sekarang memberlakukan pembatasan kompatibilitas tambalan. Cadangan yang dibuat pada versi patch yang lebih dari satu versi patch di depan cluster saat ini tidak lagi diizinkan untuk melakukan operasi pemulihan tabel.
- Memperluas fungsionalitas UNNEST untuk mendukung ekspresi JOIN. Sebelumnya, UNNEST terbatas pada referensi tabel tunggal.
- Menambahkan dukungan untuk fitur ANALISIS JIT (Just In Time). JIT ANALYZE secara otomatis, cerdas, dan efisien mengumpulkan kumpulan statistik minimal yang dibutuhkan pengoptimal kueri Amazon Redshift untuk menghasilkan rencana eksekusi kueri yang efisien, memungkinkan pelanggan mencapai kinerja harga terbaik untuk kueri dan beban kerja Apache Iceberg (OOB). out-of-box

Patch Pergeseran Merah Amazon 194

Versi cluster di tambalan ini:

- 1.0.148169 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 29 Oktober 2025
- 1.0.138443 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 17 Oktober 2025
- 1.0.129791 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 1 Oktober 2025

Fitur dan peningkatan baru di tambalan ini

- Amazon Redshift tidak akan lagi gagal dalam kueri yang mengakses relasi yang AWS Lake Formation diatur di mana bidang tersebut `AuthorizedColumns` menyertakan garis miring terbalik.
- Menambahkan dukungan untuk opsi `NULL AS` dalam perintah `COPY` untuk kolom `SUPER`, memungkinkan Anda menentukan string kustom untuk mewakili nilai `NULL`.
- Menambahkan dukungan untuk bagian tanggal `ISOWEEK` dan `ISOYEAR` dalam fungsi `EXTRACT` dan `DATE_PART` saat digunakan dengan tanggal dan stempel waktu.
- Navigasi PartiQL tetap pada variabel yang dikelompokkan untuk memastikan penguraian dan evaluasi akses bidang bersarang yang benar dalam kueri `GROUP BY`.
- Menambahkan dukungan untuk melihat kunci primer dan kendala kunci asing menggunakan perintah `SHOW CONSTRAINTS`.
- Menambahkan dukungan untuk tampilan terwujud lokal yang dibuat dari berbagi data tampilan terwujud.
- Meningkatkan kinerja kueri yang mengambil data dari tabel yang berbeda dan memanggil Lambda User Defined Functions dalam kondisi gabungan.
- Tambahkan dukungan untuk MVs di Datasharing MVs
- Perintah `DROP DATABASE` sekarang menggunakan pekerja latar belakang untuk menghapus direktori katalog node pemimpin setelah komit. Menggunakan pekerja menghindari bug langka yang terkait dengan pembersihan direktori katalog.

Patch Pergeseran Merah Amazon 193

Versi cluster di tambalan ini:

- 1.0.136890 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 13 Oktober 2025
- 1.0.127211 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 29 September 2025
- 1.0.125329 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 24 September 2025
- 1.0.121010 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 26 Agustus 2025

Fitur dan peningkatan baru di tambalan ini

- Menambahkan dukungan untuk membuat tabel menggunakan CREATE TABLE LIKE untuk mengkloning satu tabel induk dari database jarak jauh (baik dalam cluster yang sama maupun di antara cluster yang berbeda).
- Menambahkan ST_GeomFromGeoJSON overload yang menciptakan geometri dari tipe data SUPER dalam format GeoJSON, dengan dukungan untuk geometri hingga 1 MB.
- Mengubah opsi fallback default jika terjadi kesalahan konfigurasi dalam manajemen beban kerja otomatis (WLM). Opsi fallback default sekarang WLM otomatis, bukan WLM manual.
- Perhitungan zona waktu yang diperbarui dengan mengadopsi patch Database Zona Waktu IANA terbaru. Perubahan ini mengubah cara konversi tanggal dan waktu berfungsi untuk zona waktu dan periode waktu tertentu. Untuk informasi selengkapnya, lihat [Amazon Redshift menggunakan Database Zona Waktu up-to-date IANA](#).
- Menambahkan dukungan untuk penyortiran tata letak data multidimensi (MDDL). Kunci sortir MDDL secara dinamis mengurutkan data berdasarkan filter kueri aktual, mempercepat kinerja kueri. Untuk informasi selengkapnya, lihat [Penyimpanan tata letak data multidimensi](#) di Panduan Pengembang Database Amazon Redshift.

Patch Pergeseran Merah Amazon 192

Versi cluster di tambalan ini:

- 1.0.124422 - Versi cluster yang disediakan Amazon Redshift saat ini dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 22 September 2025
- 1.0.122914 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 16 September 2025

- 1.0.121035 - Versi cluster yang disediakan Amazon Redshift saat ini dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 22 Agustus 2025
- 1.0.119650 - Versi cluster yang disediakan Amazon Redshift saat ini dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 7 Agustus 2025
- 1.0.118447 - Versi cluster yang disediakan Amazon Redshift saat ini dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 24 Juli 2025

Fitur dan peningkatan baru di tambalan ini

- Memperbaiki masalah saat BERGABUNG pada kolom case-insensitive yang disusun dan kolom string konstan memilih fungsi hash yang tidak cocok.

Patch Pergeseran Merah Amazon 191

Versi cluster di tambalan ini:

- 1.0.117891 - Versi cluster yang disediakan Amazon Redshift Trailing dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 29 Juli 2025
- 1.0.117891 - Versi cluster yang disediakan Amazon Redshift saat ini dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 22 Juli 2025
- 1.0.117458 - Versi cluster yang disediakan Amazon Redshift saat ini dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 15 Juli 2025
- 1.0.116415 - Versi cluster yang disediakan Amazon Redshift saat ini dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 26 Juni 2025
- 1.0.116263 - Versi cluster yang disediakan Amazon Redshift saat ini dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 25 Juni 2025
- 1.0.115865 - Versi cluster yang disediakan Amazon Redshift saat ini dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 18 Juni 2025

Fitur dan peningkatan baru di tambalan ini

- Menambahkan dukungan untuk operasi UNNEST dalam klausa FROM pernyataan SELECT, memungkinkan pelanggan untuk mengubah elemen array menjadi baris individual.

- Meningkatkan penanganan kesalahan untuk sintaks unnesting PartiQL saat menggunakan tipe data SUPER dalam Common Table Expressions CTEs () dan CTEs rekursif, terutama saat menyelesaikan alias tabel dengan nama yang identik.
- Meningkatkan kinerja untuk analisis otomatis dengan menambahkan dukungan untuk analisis otomatis inkremental untuk tabel besar.
- Amazon Redshift sekarang mendukung Cascading Refresh dari Materialized Views () bersarang. MVs
- Amazon Redshift sekarang mendukung Auto Refresh dari Materialized Views (MV) yang ditentukan pada tabel Apache Iceberg.
- Menambahkan dukungan untuk integrasi Amazon RDS for Oracle Zero-ETL dengan Amazon Redshift.
- Menambahkan dukungan untuk Amazon RDS untuk integrasi PostgreSQL Zero-ETL dengan Amazon Redshift.

Patch Pergeseran Merah Amazon 190

Versi cluster di tambalan ini:

- 1.0.115140 - Versi cluster yang disediakan Amazon Redshift Trailing dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 16 Juni 2025
- 1.0.115564 - Versi cluster yang disediakan Amazon Redshift saat ini dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 14 Juni 2025
- 1.0.113436 - Versi cluster yang disediakan Amazon Redshift saat ini dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 15 Mei 2025
- 1.0.112895 - Versi cluster yang disediakan Amazon Redshift saat ini dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 7 Mei 2025

Fitur dan peningkatan baru di tambalan ini

- Menambahkan model yang disempurnakan untuk penskalaan dan pengoptimalan berbasis AI Amazon Redshift Serverless.
- Menambahkan dukungan untuk penyalinan otomatis saat menggunakan Amazon Redshift dengan perutean VPC yang disempurnakan diaktifkan.

- Menambahkan dukungan untuk beberapa operasi pengoptimalan tabel otomatis untuk berjalan secara bersamaan di berbagai tabel.
- Memperbaiki kondisi langka dalam konsumsi streaming di mana kueri autorefresh yang antri memicu kueri UNDO yang juga mengantri, yang mengarah ke pertikaian kunci.
- Menambahkan dukungan untuk integrasi Oracle Database@AWS nol-ETL dengan Amazon Redshift.

Patch Pergeseran Merah Amazon 189

Versi cluster di tambalan ini:

- 1.0.112790 - Versi cluster yang disediakan Amazon Redshift dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 14 Mei 2025
- 1.0.112790 - Versi cluster yang disediakan Amazon Redshift saat ini dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 6 Mei 2025
- 1.0.112086 - Versi cluster yang disediakan Amazon Redshift saat ini dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 29 April 2025
- 1.0.111040 - Versi cluster yang disediakan Amazon Redshift saat ini dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 16 April 2025
- 1.0.109768 - Versi cluster yang disediakan Amazon Redshift saat ini dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 22 Maret 2025
- 1.0.109284 - Versi cluster yang disediakan Amazon Redshift saat ini dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 13 Maret 2025
- 1.0.108971 - Versi cluster yang disediakan Amazon Redshift saat ini dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 7 Maret 2025

Fitur dan peningkatan baru di tambalan ini

- Menambahkan dukungan untuk penskalaan konkurensi untuk kueri tulis menggunakan tipe data SUPER, GEOMETRI, dan GEOGRAFI.
- Memperbaiki masalah yang memungkinkan kolom SUPER, GEOMETRI, atau GEOGRAFI ditetapkan sebagai kolom DISTKEY atau SORTKEY selama pembuatan tabel dengan CREATE TABLE AS.
- Menambahkan dukungan untuk fungsi TRY_CAST.

- Mengaktifkan dukungan untuk menjalankan beberapa perintah vakum secara bersamaan di berbagai tabel.

Patch Pergeseran Merah Amazon 188

Versi cluster di tambalan ini:

- 1.0.109616 - Trailing Amazon Redshift Versi workgroup Tanpa Server - Dirilis pada 27 Maret 2025
- 1.0.109616 - Versi cluster yang disediakan Amazon Redshift saat ini dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 24 Maret 2025
- 1.0.108470 - Trailing Amazon Redshift Versi workgroup Tanpa Server - Dirilis pada 13 Maret 2025
- 1.0.108470 - Versi workgroup Tanpa Server Amazon Redshift Saat Ini - Dirilis pada 11 Maret 2025
- 1.0.108950 - Versi cluster yang disediakan Amazon Redshift saat ini dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 10 Maret 2025
- 1.0.108790 - Versi cluster yang disediakan Amazon Redshift saat ini dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 9 Maret 2025
- 1.0.108470 - Versi cluster yang disediakan Amazon Redshift saat ini dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 4 Maret 2025
- 1.0.107910 - Versi cluster yang disediakan Amazon Redshift saat ini dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 20 Februari 2025
- 1.0.107360 - Versi cluster yang disediakan Amazon Redshift saat ini dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 13 Februari 2025
- 1.0.106767 - Versi cluster yang disediakan Amazon Redshift saat ini dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 5 Februari 2025

Fitur dan peningkatan baru di tambalan ini

- Menambahkan dukungan untuk pemasangan otomatis katalog Tabel Amazon S3, sehingga memudahkan Anda menjalankan kueri pada tabel Apache Iceberg yang dikelola di Tabel Amazon S3. Lihat [Prasyarat untuk mengelola ruang nama Amazon Redshift di Panduan Pengembang untuk izin yang AWS Glue Data Catalog](#) diperlukan.AWS Lake Formation
- Anda sekarang dapat melewati pemeriksaan masking data dan menambahkan tabel dengan lampiran masking data dinamis (DDM) ke datashare.

- Melarang penggunaan referensi korelasi yang melewati blok kueri, juga dikenal sebagai “referensi korelasi tingkat lewati”. Untuk informasi selengkapnya, lihat [Pola subkueri berkorelasi yang tidak didukung dalam Panduan Pengembang](#) Database Amazon Redshift.
- Menambahkan dukungan untuk kueri tipe data SUPER yang berkorelasi dengan berbagi data dan notasi 3 bagian.
- Menambahkan dukungan untuk kata kunci EXCLUDE SQL.
- Menambahkan dukungan untuk kata kunci GROUP BY ALL SQL baru.
- Meningkatkan kinerja untuk antrian manajemen beban kerja (WLM) yang dikonfigurasi dengan peran, menghasilkan pertenggaran kunci yang lebih rendah.
- Menambahkan dukungan untuk tampilan terwujud yang diperbarui secara otomatis pada tabel nol-ETL dengan mode riwayat diaktifkan.
- Menambahkan dukungan untuk karakter nol yang disematkan dalam string selama replikasi nol-ETL.
- Meningkatkan model untuk penskalaan dan pengoptimalan berbasis AI.
- Menambahkan username kolom ke sys_query_history tampilan.

Patch Pergeseran Merah Amazon 187

Versi cluster di tambalan ini:

- 1.0.107351 - Amazon Redshift menyediakan versi trailing track cluster dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 13 Februari 2025
- 1.0.106452 - Amazon Redshift menyediakan versi trailing track cluster - Dirilis pada 5 Februari 2025
- 1.0.106980 - Versi cluster yang disediakan Amazon Redshift saat ini dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 3 Februari 2025
- 1.0.106452 - Versi cluster yang disediakan Amazon Redshift saat ini dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 24 Januari 2025
- 1.0.106073 - Versi cluster yang disediakan Amazon Redshift saat ini dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 21 Januari 2025
- 1.0.105722 - Versi cluster yang disediakan Amazon Redshift saat ini dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 10 Januari 2025
- 1.0.105373 - Versi cluster yang disediakan Amazon Redshift saat ini dan versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 8 Januari 2025

- 1.0.104930 - Versi workgroup Tanpa Server Amazon Redshift - Dirilis pada 20 Desember 2024

Fitur dan peningkatan baru di tambalan ini

- Memperkenalkan dua fungsi H3 spasial baru (H3_center dan H3_Boundary).
- Mengaktifkan cache disk dinamis pada cluster penskalaan konkurensi.
- Memperbaiki inefisiensi dalam penggunaan memori selama konsumsi tabel yang diurutkan yang dapat memengaruhi kinerja.
- Memperbaiki masalah di mana kueri tertentu dengan subquery di dalam klausa SELECT, seperti `SELECT ARRAY_FLATTEN((SELECT FROM ...))`, akan salah memicu kesalahan `size >= min_partiql_size XCHECK` dalam skenario tertentu yang melibatkan nilai SUPER tunggal.
- Memperbaiki masalah di mana ekspresi SUPER yang tidak dapat dibatalkan (`json_serialize()`, `json_size()`, `json_typeof()`, `is_object()`, dan beberapa lainnya) terkadang akan menghasilkan hasil yang salah bila dikombinasikan dengan ekspresi argumen seperti `atau. CASE ... END COALESCE()`
- Memperbaiki masalah di mana SELECT dari tampilan pengikatan terlambat akan memunculkan ERROR setelah mengaktifkan keamanan tingkat baris pada tampilan.
- Memperbaiki masalah di mana CDC pada integrasi nol-ETL menyebabkan konsumsi CPU node pemimpin yang tinggi.
- Menambahkan kemampuan integrasi nol-ETL ke tabel kueri di semua negara bagian, termasuk selama pembaruan.
- Menambahkan kemampuan ke integrasi nol-ETL untuk memotong besar texts/strings untuk pemetaan ke ukuran varchar maksimal di Amazon Redshift.
- Menambahkan kemampuan ke integrasi nol-ETL untuk mengganti karakter UTF-8 yang tidak valid dengan karakter tertentu pilihan Anda.
- Memperbaiki masalah pada 'interval penyegaran' untuk integrasi nol-ETL dengan Amazon RDS for MySQL.
- Menambahkan dukungan untuk tipe JSON/JSONB data untuk integrasi nol-ETL dengan Aurora PostgreSQL.
- Memperbaiki masalah di mana kueri yang berisi ekspresi dengan `IS [NOT] {TRUE|FALSE|UNKNOWN}` akan mengalami kesalahan pernyataan.
- Memperbaiki masalah di mana `json_extract_array_element_text()` fungsi `json_extract_path_text()` dan akan menghasilkan string kosong ' ' alih-alih NULL atau

sebaliknya dalam kasus seperti mengakses elemen array atau atribut yang tidak ada, 'null' nilai JSON, atau string JSON kosong.

- Meningkatkan kinerja INSERT/COPY pernyataan dalam transaksi bersamaan menulis ke tabel yang sama dengan memungkinkan mereka untuk berbagi kunci dan membuat kemajuan sampai mereka perlu menulis data ke tabel.

Patch Pergeseran Merah Amazon 186

Versi cluster di tambalan ini:

- 1.0.82096 - Versi trailing track - Dirilis pada 10 Januari 2025
- 1.0.82000 - Amazon Redshift Versi Tanpa Server - Dirilis pada 10 Januari 2025
- 1.0.81981 - Versi trek saat ini - Dirilis pada 10 Januari 2025
- 1.0.81475 - Versi trailing track - Dirilis pada 6 Januari 2025
- 1.0.81473 - Amazon Redshift Versi Tanpa Server - Dirilis pada 6 Januari 2025
- 1.0.81462 - Versi trek saat ini - Dirilis pada 6 Januari 2025
- 1.0.80643 - Versi trailing track - Dirilis pada 17 Desember 2024
- 1.0.80583 - Amazon Redshift Versi Tanpa Server - Dirilis pada 17 Desember 2024
- 1.0.80560 - Versi trek saat ini - Dirilis pada 17 Desember 2024
- 1.0.80498 - Amazon Redshift Versi Tanpa Server - Dirilis pada 13 Desember 2024
- 1.0.80491 - Versi trek saat ini - Dirilis pada 13 Desember 2024
- 1.0.80036 - Amazon Redshift Versi Tanpa Server - Dirilis pada 6 Desember 2024
- 1.0.80009 - Versi trek saat ini - Dirilis pada 6 Desember 2024
- 1.0.79372 - Amazon Redshift Versi Tanpa Server - Dirilis pada 26 November 2024
- 1.0.79237 - Amazon Redshift Versi Tanpa Server - Dirilis pada 24 November 2024
- 1.0.79229 - Versi trek saat ini - Dirilis pada 24 November 2024
- 1.0.79003 - Amazon Redshift Versi Tanpa Server - Dirilis pada 19 November 2024
- 1.0.78987 - Versi trek saat ini - Dirilis pada 19 November 2024
- 1.0.78890 - Amazon Redshift Versi Tanpa Server - Dirilis pada 18 November 2024
- 1.0.78881 - Versi trek saat ini - Dirilis pada 18 November 2024

- 1.0.78646 - Amazon Redshift Versi Tanpa Server - Dirilis pada 14 November 2024
- 1.0.78641 - Versi trek saat ini - Dirilis pada 14 November 2024
- 1.0.78178 - Amazon Redshift Versi Tanpa Server - Dirilis pada 12 November 2024
- 1.0.78160 - Versi trek saat ini - Dirilis pada 12 November 2024
- 1.0.77809 - Amazon Redshift Versi Tanpa Server - Dirilis pada 31 Oktober 2024
- 1.0.77777 - Versi trek saat ini - Dirilis pada 31 Oktober 2024
- 1.0.77292 - Amazon Redshift Versi Tanpa Server - Dirilis pada 24 Oktober 2024
- 1.0.77272 - Versi trek saat ini - Dirilis pada 24 Oktober 2024
- 1.0.77040 - Amazon Redshift Versi Tanpa Server - Dirilis pada 22 Oktober 2024
- 1.0.77028 - Versi trek saat ini - Dirilis pada 22 Oktober 2024

Fitur dan peningkatan baru di tambalan ini

- Menambahkan dukungan untuk penyegaran otomatis dan inkremental tampilan terwujud pada tabel dari integrasi nol-ETL dengan Amazon Aurora MySQL, Amazon Aurora PostgreSQL, Amazon RDS for MySQL, dan Amazon DynamoDB.
- Meningkatkan cara Amazon Redshift menulis ulang kueri dengan subkueri baris tunggal yang berkorelasi, seperti. `SELECT ... WHERE key = (SELECT ... correlated subquery)`
Perhatikan bahwa kueri tersebut hanya valid jika subquery menghasilkan satu baris. Karena penulisan ulang yang ditingkatkan, kueri tertentu yang melanggar kondisi ini mungkin sekarang gagal dengan `ERROR: subquery baris tunggal mengembalikan lebih dari satu baris` jika sebelumnya diizinkan. Untuk menghindari hal ini, subkueri semacam itu mungkin perlu diperbaiki sehingga dijamin mengembalikan satu baris, misalnya, dengan menambahkan agregat `MIN ()` atau `MAX ()`.
- Menambahkan SQL Identifier 'KAFKA' di Amazon Redshift untuk mendukung streaming dari sumber Kafka eksternal ke Amazon Redshift dengan konsumsi streaming langsung. Sumber-sumber ini termasuk sumber Kafka eksternal seperti Confluent Managed Cloud dan Apache Kafka.
- Menambahkan dukungan untuk penulisan multi-gudang menggunakan berbagi data sehingga Anda dapat menskalakan beban kerja tulis Anda dan mencapai kinerja yang lebih baik untuk mengekstrak, mengubah, dan memuat (ETL) beban kerja dengan menggunakan gudang yang berbeda dari berbagai jenis dan ukuran, berdasarkan kebutuhan beban kerja Anda.
- Sekarang Anda dapat menjalankan `SELECT` kueri di tabel Apache Iceberg yang dikelola di Tabel Amazon S3 menggunakan Amazon Redshift Untuk informasi selengkapnya, lihat Mengakses [tabel](#)

[Amazon S3 dengan Amazon Redshift di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.](#)

- Menambahkan dukungan untuk kueri penulisan lintas basis data di Amazon Redshift, sehingga Anda dapat menulis di beberapa database dalam klaster Amazon Redshift. Untuk informasi selengkapnya, lihat [Kueri lintas basis data](#).

Patch Pergeseran Merah Amazon 185

Versi cluster di tambalan ini:

- 1.0.79878 - Versi trailing track - Dirilis pada 3 Desember 2024
- 1.0.79868 - Amazon Redshift Versi Tanpa Server - Dirilis pada 3 Desember 2024
- 1.0.79845 - Versi trek saat ini - Dirilis pada 3 Desember 2024
- 1.0.78946 - Versi trailing track - Dirilis pada 19 November 2024
- 1.0.78354 - Versi trailing track - Dirilis pada 12 November 2024
- 1.0.78130 - Amazon Redshift Versi Tanpa Server - Dirilis pada 12 November 2024
- 1.0.78125 - Versi trek saat ini - Dirilis pada 12 November 2024
- 1.0.78016 - Amazon Redshift Versi Tanpa Server - Dirilis pada 5 November 2024
- 1.0.78014 - Versi trek saat ini - Dirilis pada 5 November 2024
- 1.0.77707 - Amazon Redshift Versi Tanpa Server - Dirilis pada 4 November 2024
- 1.0.77687 - Versi trek saat ini - Dirilis pada 4 November 2024
- 1.0.77467 - Amazon Redshift Versi Tanpa Server - Dirilis pada 4 November 2024
- 1.0.77433 - Versi trek saat ini - Dirilis pada 4 November 2024
- 1.0.77467 - Amazon Redshift Versi Tanpa Server - Dirilis pada 29 Oktober 2024
- 1.0.77433 - Versi trek saat ini - Dirilis pada 29 Oktober 2024
- 1.0.76991 - Amazon Redshift Versi Tanpa Server - Dirilis pada 21 Oktober 2024
- 1.0.76913 - Versi trek saat ini - Dirilis pada 21 Oktober 2024
- 1.0.76645 - Amazon Redshift Versi Tanpa Server - Dirilis pada 14 Oktober 2024
- 1.0.76642 - Versi trek saat ini - Dirilis pada 14 Oktober 2024
- 1.0.76242 - Amazon Redshift Versi Tanpa Server - Dirilis pada 10 Oktober 2024

- 1.0.76230 - Versi trek saat ini - Dirilis pada 10 Oktober 2024

Fitur dan peningkatan baru di tambalan ini

- Menambahkan dukungan untuk penyegaran inkremental tampilan terwujud (MVs) yang dibuat pada tabel data lake.
- Memperkenalkan dukungan untuk mengubah kunci distribusi dan kunci pengurutan untuk tampilan terwujud.
- Menambahkan dukungan untuk integrasi pembelajaran mesin Amazon Redshift (ML) dengan Amazon Bedrock untuk memanfaatkan model bahasa besar (LLMs) dari perintah SQL sederhana bersama dengan data di Amazon Redshift.
- Memperbaiki bug yang memungkinkan pemrosesan shapefile kosong di data spasial Amazon Redshift tanpa menimbulkan kesalahan.
- Memperbaiki bug yang memungkinkan tipe data varbit dan varbinary untuk menelan nilai string kosong sebagai "", bukan NULL, dalam integrasi nol-ETL.
- Memperbaiki kondisi balapan dengan cache hasil, yang menyebabkan kueri lintas basis data mengembalikan hasil basi dalam integrasi nol-ETL.
- Mengoptimalkan proses sinkronisasi ulang integrasi nol-ETL, menghasilkan waktu sinkronisasi ulang yang lebih singkat.
- Meningkatkan waktu bootstrap nol-ETL setelah pemulihan dan operasi pemeliharaan sistem. Sekarang sistem dapat memulihkan CDC setelah operasi kueri terakhir, alih-alih dapat memulihkan hingga titik CDC terbaru yang tersedia.
- Meningkatkan observabilitas untuk integrasi nol-ETL dengan tabel sistem baru, `sys_integration_table_activity`. Tabel ini melacak sisipan, pembaruan, dan penghapusan tabel integrasi nol-ETL.
- Meningkatkan pembuatan otomatis pengalaman peran federasi. Administrator Amazon Redshift sekarang memiliki kontrol lebih besar atas pembuatan otomatis peran federasi selama login pengguna federasi. Anda sekarang dapat mengaktifkan, menonaktifkan, menerapkan filter ke, dan mengonfigurasi pengaturan pembuatan otomatis untuk setiap penyedia identitas.
- Pengguna IAM Identity Center dapat menjalankan COPY, UNLOAD, dan CREATE LIBRARY dengan kredensial IAM Identity Center menggunakan hibah akses S3.
- Salin otomatis (COPY JOB) yang memungkinkan konsumsi file secara terus menerus dari Amazon S3, sekarang tersedia secara umum.

Patch Pergeseran Merah Amazon 184

Versi cluster di tambalan ini:

- 1.0.77706 - Versi trailing track - Dirilis pada 12 November 2024
- 1.0.76832 - Versi trailing track - Dirilis pada 17 Oktober 2024
- 1.0.76169 - Amazon Redshift Versi Tanpa Server - Dirilis pada 10 Oktober 2024
- 1.0.76142 - Versi trek saat ini - Dirilis pada 10 Oktober 2024
- 1.0.75677 - Amazon Redshift Versi Tanpa Server - Dirilis pada 27 September 2024
- 1.0.75672 - Versi trek saat ini - Dirilis pada 27 September 2024
- 1.0.75504 - Amazon Redshift Versi Tanpa Server - Dirilis pada 23 September 2024
- 1.0.75449 - Versi trek saat ini - Dirilis pada 24 September 2024
- 1.0.74765 - Amazon Redshift Versi Tanpa Server - Dirilis pada 12 September 2024
- 1.0.74754 - Versi trek saat ini - Dirilis pada 12 September 2024

Fitur dan peningkatan baru di tambalan ini

- Tampilan terwujud streaming Amazon Redshift (MV's) untuk streaming konsumsi data telah meningkatkan ukuran kolom VARBYTE dari 1.024.000 byte. Amazon Redshift sekarang dapat menelan rekaman dari Amazon Kinesis Data Streams hingga 1.048.576 byte, atau 1MiB. Redshift dapat menyerap rekaman dari Amazon Managed Streaming untuk Apache Kafka hingga 16.777.216 byte, atau 16 MiB. Jika Anda menggunakan perintah `ATA SQLALTER TABLE <target_tbl> APPEND FROM <streaming_mv>`, lepaskan dan buat ulang <target_tbl> untuk memiliki ukuran kolom VARBYTE yang lebih besar.
- Datashares Amazon Redshift sekarang dapat menyertakan tabel dan tampilan data Amazon S3 yang mereferensikan termasuk tabel AWS Glue Data Catalog yang diatur oleh Lake Formation.
- Menambahkan dukungan untuk penyegaran otomatis dan inkremental tampilan terwujud pada tabel dari integrasi nol-ETL dengan DynamoDB.
- Menambahkan dukungan untuk mengonfigurasi 'interval penyegaran' pada tabel integrasi nol-ETL untuk menentukan kecepatan refresh replikasi di Amazon Redshift. Anda dapat mengaturnya pada saat membuat database untuk integrasi baru atau mengubah database integrasi yang ada.
- Menambahkan dukungan untuk autentikasi Mutual Transport Layer Security (mTLS) di konsumsi streaming Amazon Redshift untuk Amazon Managed Streaming for Apache Kafka.

- Menambahkan dukungan untuk kueri hash, pengidentifikasi unik untuk kueri SQL berdasarkan representasi tekstual dari kueri dan nilai-nilai parameternya. Ini dapat digunakan untuk mengidentifikasi, mengelompokkan, dan menganalisis pertanyaan serupa. Query hash sekarang dapat ditemukan dalam tampilan SYS_QUERY_HISTORY, dengan penambahan dua kolom baru:
 - `user_query_hash`— Hash seperti yang dikirimkan oleh pengguna termasuk literal kueri.
 - `generic_query_hash`— Hash seperti yang dikirimkan oleh pengguna tanpa literal query.
- Memperbaiki kebuntuan sistem dalam kasus yang jarang terjadi di mana Zero-ETL menjalankan replikasi dan pemindaian CDC, melakukan kueri tabel dengan cache hasil.
- Mengatasi masalah dalam Manajemen Beban Kerja (WLM) di mana kueri Python User Defined Function (UDF) akan mendahului kueri lain ketika sumber daya WLM untuk kueri UDF Python tidak tersedia.
- Mengatasi masalah di mana Manajemen Beban Kerja (WLM) akan gagal merutekan kueri ke antrian yang dipetakan ke peran pengguna yang baru dibuat.
- Meningkatkan pemanfaatan disk pada konsumen berbagi data yang lebih kecil yang menanyakan tabel produsen besar.
- Meningkatkan kinerja pernyataan INSERT untuk klaster yang disediakan diubah ukurannya secara elastis ke ukuran yang lebih tinggi.

Patch Pergeseran Merah Amazon 183

Versi cluster di tambalan ini:

- 1.0.75655 - Versi trailing track - Dirilis pada 30 September 2024
- 1.0.75388 - Amazon Redshift Versi Tanpa Server - Dirilis pada 25 September 2024
- 1.0.75379 - Versi trek saat ini - Dirilis pada 25 September 2024
- 1.0.74967 - Amazon Redshift Versi Tanpa Server - Dirilis pada 17 September 2024
- 1.0.74927 - Versi trek saat ini - Dirilis pada 17 September 2024
- 1.0.74518 - Amazon Redshift Versi Tanpa Server - Dirilis pada 11 September 2024
- 1.0.74503 - Versi trek saat ini - Dirilis pada 11 September 2024
- 1.0.74223 - Amazon Redshift Versi Tanpa Server - Dirilis pada 5 September 2024
- 1.0.74159 - Versi trek saat ini - Dirilis pada 5 September 2024
- 1.0.74126 - Amazon Redshift Versi Tanpa Server - Dirilis pada 30 Agustus 2024

- 1.0.74097 - Versi trek saat ini - Dirilis pada 30 Agustus 2024
- 1.0.73016 - Amazon Redshift Versi Tanpa Server - Dirilis pada 8 Agustus 2024
- 1.0.72982 - Versi trek saat ini - Dirilis pada 8 Agustus 2024

Fitur dan peningkatan baru di tambalan ini

- Menambahkan dukungan untuk penemuan Scoped Permissions melalui `SVV_DATABASE_PRIVILEGES` dan `SVV_SCHEMA_PRIVILEGES`. Juga memperkenalkan kolom `privilege_scope` ke `SVV_DATABASE_PRIVILEGES` dan `SVV_SCHEMA_PRIVILEGES`.
- Meningkatkan kinerja kueri yang menjalankan operasi agregasi yang berbeda ketika pengelompokan kolom memiliki jumlah nilai berbeda (NDV) yang rendah.
- Meningkatkan kinerja `INSERT/COPY` pernyataan untuk gudang data yang disediakan secara elastis diubah ukurannya sebesar 2x atau lebih tinggi.
- Mendukung variabel konteks sesi di dalam kebijakan Dynamic Data Masking.
- Menambahkan dukungan untuk subkueri dan tampilan sebagai sumber untuk pernyataan `MERGE`.
- Mendukung prosedur tersimpan yang berisi pernyataan `MERGE` pada penskalaan konkurensi yang disediakan dan komputasi penskalaan otomatis tanpa server.
- Meningkatkan kinerja kueri dengan prediksi sumber daya yang lebih baik dalam manajemen beban kerja untuk perintah `COPY` dan untuk gudang yang mengalami perubahan ukuran.
- Meningkatkan ketahanan terhadap kesalahan memori di cluster dengan memori terbatas yang tersedia
- Menambahkan dukungan untuk karakter non-ASCII sebagai pembatas bidang ke perintah `COPY`.
- Menambahkan dukungan untuk menelan data yang dikodekan dalam set karakter ISO-8859-1 menggunakan perintah `COPY`.
- Menghapus persyaratan untuk menentukan `CLUSTER_ARN` dalam definisi skema eksternal MSK jika menentukan URI.
- Mendukung penerapan filter pemindaian rentang selama pemindaian pada tabel integrasi nol-ETL.
- Mendukung penambahan kunci pengurutan ke tabel integrasi nol-ETL.
- Mendukung opsi database seperti `serializable` dan `collation` akan ditentukan dengan pernyataan `CREATE DATABASE` saat membuat database integrasi nol-ETL.
- Memperbaiki masalah yang menyebabkan cluster dimulai ulang saat filter data melebihi 2 KB dalam integrasi nol-ETL.

Patch Pergeseran Merah Amazon 182

Versi cluster di tambalan ini:

- 1.0.73589 - Versi trailing track - Dirilis pada 22 Agustus 2024
- 1.0.73359 - Amazon Redshift Versi Tanpa Server - Dirilis pada 15 Agustus 2024
- 1.0.73348 - Versi trek saat ini - Dirilis pada 15 Agustus 2024
- 1.0.72917 - Amazon Redshift Versi Tanpa Server - Dirilis pada 12 Agustus 2024
- 1.0.72899 - Versi trek saat ini - Dirilis pada 12 Agustus 2024
- 1.0.72528 - Amazon Redshift Versi Tanpa Server - Dirilis pada 7 Agustus 2024
- 1.0.72503 - Versi trek saat ini - Dirilis pada 8 Agustus 2024
- 1.0.72239 - Amazon Redshift Versi Tanpa Server - Dirilis pada 1 Agustus 2024
- 1.0.71714 - Amazon Redshift Versi Tanpa Server - Dirilis pada 24 Juli 2024
- 1.0.71629 - Versi trek saat ini - Dirilis pada 24 Juli 2024
- 1.0.70953 - Amazon Redshift Versi Tanpa Server - Dirilis pada 11 Juli 2024
- 1.0.70890 - Versi trek saat ini - Dirilis pada 11 Juli 2024
- 1.0.70716 - Amazon Redshift Versi Tanpa Server - Dirilis pada 8 Juli 2024
- 1.0.70695 - Versi trek saat ini - Dirilis pada 8 Juli 2024
- 1.0.69945 - Amazon Redshift Versi Tanpa Server - Dirilis pada 27 Juni 2024
- 1.0.69938 - Versi trek saat ini - Dirilis pada 27 Juni 2024

Fitur dan peningkatan baru di tambalan ini

- Memperbarui LISTAGG, MEDIAN, PERCENTILE_CONT dan PERCENTILE_DISC agar tidak lagi memerlukan tabel yang ditentukan pengguna. Kueri yang mereferensikan tabel katalog atau yang tidak mereferensikan tabel apa pun juga dapat menggunakan fungsi ini.
- Mengurangi waktu perencanaan kueri untuk pembagian data kueri baca dengan mengkonsolidasikan tabel temp di beberapa kueri dalam satu sesi, untuk beban kerja konkurensi tinggi.
- Menyediakan ketersediaan umum integrasi pembelajaran mesin Redshift (ML) dengan Amazon SageMaker AI Jumpstart, untuk menghadirkan model bahasa besar Anda sendiri.
- Memperkenalkan dukungan untuk tipe data input dan output SUPER di Redshift ML.

- Mengaktifkan dukungan untuk pernyataan UPDATE dengan klausa JOIN ketika tabel target dilindungi oleh kebijakan masking data dinamis dan direferensikan dalam klausa JOIN.
- Memungkinkan kueri database nol-ETL pada Redshift, bahkan setelah integrasi dihapus dari sumber.
- Memperbaiki kesalahan replikasi yang dapat menyebabkan integrasi nol-ETL gagal. Ini membuat integrasi lebih tangguh.
- Memungkinkan pengguna selain pengguna yang membuat integrasi nol-ETL untuk menanyakan data, setelah izin GRANT.
- Memperbaiki out-of-memory masalah yang dapat menyebabkan cluster restart di kluster yang disediakan Amazon Redshift dengan integrasi nol-ETL diaktifkan.
- Memungkinkan pembuatan RDS untuk integrasi MySQL Zero-ETL dengan Redshift, dari cluster DB multi-AZ RDS sumber. Cluster DB multi-AZ adalah mode penyebaran Amazon RDS semisinkron dan ketersediaan tinggi dengan dua instans replika DB yang dapat dibaca.
- Memungkinkan pengguna untuk terhubung ke kluster MSK Amazon dari klien konsumen streaming Amazon Redshift dengan menentukan URI broker kluster MSK Amazon dalam definisi skema eksternal yang diperlukan untuk mengaitkan tampilan terwujud streaming Amazon Redshift dengan topik MSK Amazon. Fitur ini menghilangkan kebutuhan untuk mendapatkan nama node broker bootstrap Amazon MSK dengan memanggil GetBootstrapBroker API di cluster MSK Amazon melalui gateway internet.
- Menyelesaikan masalah dengan melanjutkan instans Amazon Redshift Tanpa Server, memungkinkan pengguna database yang ada untuk melanjutkan instance Tanpa Server saat menyambungkan ke database di editor kueri Amazon Redshift v2, menggunakan metode autentikasi IAM Identity Center.
- Mengoptimalkan replikasi CDC dan mengurangi pemanfaatan sumber daya pada komputasi Redshift dengan pindah ke sharding berbasis tabel.
- Meningkatkan kinerja kueri menggunakan prediksi sumber daya yang disempurnakan dalam manajemen beban kerja (WLM).
- Memperbaiki kueri yang gagal pada cluster penskalaan konkurensi dengan pesan: kehabisan antrian WLM untuk memulai ulang.
- Memperbaiki masalah manajemen beban kerja (WLM) di mana Amazon Redshift kembali ke WLM manual saat pelanggan mencoba menerapkan konfigurasi WLM yang tidak valid.
- Memungkinkan konsumen berbagi data untuk menjalankan kueri baca bahkan ketika produsen sedang down karena pemeliharaan yang direncanakan atau pemadaman yang tidak direncanakan.

- Memperbaiki masalah restart cluster langka yang terjadi ketika fungsi ANY_VALUE digunakan dalam kueri yang mengumpulkan data, misalnya, fungsi agregasi COUNT (DISTINCT).

Patch Pergeseran Merah Amazon 181

Versi cluster di tambalan ini:

- 1.0.72031 - Versi trek saat ini - Dirilis pada 1 Agustus 2024
- 1.0.71912 - Versi trailing track - Dirilis pada 1 Agustus 2024
- 1.0.70665 - Amazon Redshift Versi Tanpa Server - Dirilis pada 8 Juli 2024
- 1.0.70634 - Versi trek saat ini - Dirilis pada 8 Juli 2024
- 1.0.69954 - Amazon Redshift Versi Tanpa Server - Dirilis pada 26 Juni 2024
- 1.0.69952 - Versi trek saat ini - Dirilis pada 26 Juni 2024
- 1.0.69497 - Amazon Redshift Versi Tanpa Server - Dirilis pada 18 Juni 2024
- 1.0.69451 - Versi trek saat ini - Dirilis pada 18 Juni 2024
- 1.0.69076 - Amazon Redshift Versi Tanpa Server - Dirilis pada 14 Juni 2024
- 1.0.69065 - Versi trek saat ini - Dirilis pada 14 Juni 2024
- 1.0.68555 - Amazon Redshift Versi Tanpa Server - Dirilis pada 31 Mei 2024
- 1.0.68540 - Versi trek saat ini - Dirilis pada 31 Mei 2024
- 1.0.68328 - Amazon Redshift Versi Tanpa Server - Dirilis pada 23 Mei 2024
- 1.0.68205 - Versi trek saat ini - Dirilis pada 23 Mei 2024
- 1.0.67796 - Amazon Redshift Versi Tanpa Server - Dirilis pada 15 Mei 2024
- 1.0.67788 - Versi trek saat ini - Dirilis pada 15 Mei 2024
- 1.0.67308 - Amazon Redshift Versi Tanpa Server - Dirilis pada 1 Mei 2024
- 1.0.67305 - Versi trek saat ini - Dirilis pada 1 Mei 2024

Fitur dan peningkatan baru di tambalan ini

- Memperkenalkan dukungan untuk fungsi 'lower_attribute_names ()' dan 'upper_attribute_names ()' yang memodifikasi kasus nama atribut untuk nilai objek SUPER.
- Memperbaiki masalah di CREATE TABLE LIKE saat menggunakan kolom identitas. Sebelumnya, tabel baru akan mewarisi pengenalan dari tabel sumber. Ini menyebabkan masalah jika tabel sumber kemudian dijatuhkan, karena pengenalan akan menjadi tidak valid di tabel baru.

- Memperbaiki masalah yang mencegah beberapa tabel eksternal ditampilkan di `SVV_ALL_TABLES`.
- Meningkatkan waktu bootstrap cluster, dan mempercepat inisialisasi kueri untuk beban kerja bersamaan yang tinggi.
- Memperbaiki masalah dengan kueri federasi yang menyebabkan kesalahan saat meneruskan fungsi `split_part ()` ke sumber federasi ke RDS dan Aurora MySQL
- Mendukung perubahan yang dimulai pengguna pada kunci distribusi melalui `ALTER TABLE... ALTER DISTYLE KEY DISTYLE` perintah pada cluster penskalaan konkurensi yang disediakan dan komputasi penskalaan otomatis tanpa server.
- Mendukung tampilan terwujud yang disegarkan secara manual yang melibatkan agregasi pada penskalaan konkurensi yang disediakan dan komputasi penskalaan otomatis tanpa server.
- Menambahkan dukungan untuk Zero-ETL untuk menangani catatan hingga 16 MB dalam ukuran dan untuk mendukung nilai SUPER hingga 16 MB.
- Meningkatkan pesan kesalahan selama sinkronisasi awal di nol-ETL dari Aurora MySQL dengan memberikan rincian tambahan seperti skema dan nama tabel.
- Memperkenalkan dukungan untuk penandaan dengan Amazon Redshift `MLCREATE MODEL`. Dengan peningkatan ini, Anda sekarang dapat menandai sumber daya Amazon SageMaker AI yang digunakan oleh Amazon Redshift ML. Penandaan membantu Anda mengelola, mengidentifikasi, mengatur, mencari, dan memfilter sumber daya.
- Meningkatkan kinerja kueri yang melibatkan fungsi yang ditentukan pengguna Lambda UDFs () dengan mengoptimalkan pemrosesan data dengan file. AWS Lambda
- Mengurangi pemanfaatan memori selama konsumsi data dalam tabel yang diurutkan dari cluster yang diubah ukurannya secara elastis dan tanpa server.
- Menambahkan dukungan untuk baris baru (`\n`) di kolom dalam tampilan `SYS_QUERY_HISTORY` dan untuk kolom `query_text` dalam tampilan `SYS_QUERY_TEXT`. `text`

Patch Pergeseran Merah Amazon 180

Versi cluster di tambalan ini:

- 1.0.68520 - Versi trailing track - Dirilis pada 28 Mei 2024
- 1.0.67699 - Versi trailing track - Dirilis pada 15 Mei 2024
- 1.0.66960 - Versi trailing track - Dirilis pada 21 April 2024
- 1.0.66954 - Versi trek saat ini - Dirilis pada 21 April 2024
- 1.0.66276 - Versi trek saat ini - Dirilis pada 12 April 2024

- 1.0.66290 - Amazon Redshift Versi Tanpa Server - Dirilis pada 10 April 2024
- 1.0.63590 - Versi trek saat ini - Dirilis pada 19 Februari 2024
- 1.0.63567 - Amazon Redshift Versi Tanpa Server - Dirilis pada 16 Februari 2024
- 1.0.63282 - Amazon Redshift Versi Tanpa Server - Dirilis pada 13 Februari 2024
- 1.0.63269 - Versi trek saat ini - Dirilis pada 13 Februari 2024
- 1.0.63215 - Amazon Redshift Versi Tanpa Server - Dirilis pada 12 Februari 2024
- 1.0.63205 - Versi trek saat ini - Dirilis pada 12 Februari 2024
- 1.0.63030 - Amazon Redshift Versi Tanpa Server - Dirilis pada 7 Februari 2024
- 1.0.62913 - Versi trek saat ini - Dirilis pada 7 Februari 2024
- 1.0.62922 - Amazon Redshift Versi Tanpa Server - Dirilis pada 5 Februari 2024
- 1.0.62878 - Versi trek saat ini - Dirilis pada 5 Februari 2024
- 1.0.62698 - Amazon Redshift Versi Tanpa Server - Dirilis pada 31 Januari 2024
- 1.0.62614 - Versi trek saat ini - Dirilis pada 31 Januari 2024
- 1.0.61687 - Amazon Redshift Versi Tanpa Server - Dirilis pada 5 Januari 2024
- 1.0.61678 - Versi trek saat ini - Dirilis pada 5 Januari 2024
- 1.0.61567 - Amazon Redshift Versi Tanpa Server - Dirilis pada 31 Desember 2023
- 1.0.61559 - Versi trek saat ini - Dirilis pada 31 Desember 2023
- 1.0.61430 - Amazon Redshift Versi Tanpa Server - Dirilis pada 29 Desember 2023
- 1.0.61395 - Versi trek saat ini - Dirilis pada 29 Desember 2023

Fitur dan peningkatan baru di tambalan ini

- Mengubah `CURRENT_USER` agar tidak lagi memotong nama pengguna yang dikembalikan menjadi 64 karakter.
- Menambahkan kemampuan untuk menerapkan kebijakan penyembunyian data pada tampilan standar dan tampilan pengikatan akhir.
- Menambahkan kemampuan untuk menerapkan masking data dinamis (DDM) ke atribut skalar di kolom tipe data `SUPER`.
- Menambahkan fungsi `SQL_OBJECT_TRANSFORM`. Untuk informasi selengkapnya, lihat [fungsi `OBJECT\_TRANSFORM` di Panduan Pengembang Database Amazon Redshift](#).
- Menambahkan kemampuan untuk menerapkan kontrol akses AWS Lake Formation berbutir halus ke data bersarang Anda, dan kueri dengan analitik data lake Amazon Redshift.

- Menambahkan tipe data INTERVAL.
- Menambahkan CONTINUE_HANDLER, yang merupakan jenis handler pengecualian yang mengontrol aliran prosedur yang disimpan. Dengan menggunakannya, Anda dapat menangkap dan menangani pengecualian tanpa mengakhiri blok pernyataan yang ada.
- Menambahkan kemampuan untuk menentukan izin pada lingkup (skema atau database) selain objek individu. Hal ini memungkinkan pengguna dan peran untuk diberikan izin pada semua objek saat ini dan masa depan dalam lingkup.
- Menambahkan kemampuan untuk membuat database dari datashare dengan izin yang memungkinkan administrator sisi konsumen memberikan izin individu pada objek database bersama kepada pengguna dan peran sisi konsumen.
- Menambahkan dukungan untuk tipe data pengembalian SUPER dari model BYOM jarak jauh. Ini memperluas jangkauan model SageMaker AI yang diterima untuk memasukkan model dengan format pengembalian yang lebih kompleks.
- Mengubah fungsi eksternal untuk sekarang secara implisit melemparkan angka dengan atau tanpa bagian pecahan ke tipe data numerik kolom. Untuk kolom int2, int4, dan int8, angka dengan digit pecahan diterima dengan memotong kecuali jumlahnya di luar jangkauan. Untuk kolom float4 dan float8, angka diterima tanpa digit fraksional.
- Menambahkan tiga fungsi spasial yang bekerja dengan sistem grid pengindeksan geospasial hierarkis H3: H3_, H3_, dan FromLongLat H3_Polyfill. FromPoint

Patch Pergeseran Merah Amazon 179

Versi cluster di tambalan ini:

- 1.0.62317 - Amazon Redshift Versi Tanpa Server - Dirilis pada 29 Januari 2024
- 1.0.62312 - Versi trailing track - Dirilis pada 29 Januari 2024
- 1.0.61631 - Amazon Redshift Versi Tanpa Server - Dirilis pada 5 Januari 2024
- 1.0.61626 - Versi trek saat ini - Dirilis pada 5 Januari 2024
- 1.0.61191 - Versi trek saat ini - Dirilis pada 16 Desember 2023
- 1.0.61150 - Amazon Redshift Versi Tanpa Server - Dirilis pada 16 Desember 2023
- 1.0.60982 - Amazon Redshift Versi Tanpa Server - Dirilis pada 13 Desember 2023
- 1.0.60854 - Versi trek saat ini - Dirilis pada 10 Desember 2023
- 1.0.60354 - Amazon Redshift Versi Tanpa Server - Dirilis pada 22 November 2023

- 1.0.60353 - Versi trek saat ini - Dirilis pada 21 November 2023
- 1.0.60293 - Amazon Redshift Versi Tanpa Server - Dirilis pada 21 November 2023
- 1.0.60292 - Versi trek saat ini - Dirilis pada 22 November 2023
- 1.0.60161 - Amazon Redshift Versi Tanpa Server - Dirilis pada 18 November 2023
- 1.0.60140 - Versi trek saat ini - Dirilis pada 18 November 2023
- 1.0.60139 - Amazon Redshift Versi Tanpa Server - Dirilis pada 18 November 2023
- 1.0.59947 - Amazon Redshift Versi Tanpa Server - Dirilis pada 16 November 2023
- 1.0.59945 - Versi trek saat ini - Dirilis pada 16 November 2023
- 1.0.59118 - Amazon Redshift Versi Tanpa Server - Dirilis pada 9 November 2023
- 1.0.59117 - Versi trek saat ini - Dirilis pada 9 November 2023

Fitur dan peningkatan baru di tambalan ini

- Menambahkan dukungan sehingga pengguna federasi dengan izin yang sesuai dapat melihat keamanan tingkat baris dan tampilan sistem masking data dinamis, termasuk:
 - SVV_ATTACHED_MASKING_POLICY
 - SVV_MASKING_POLICY
 - SVV_RLS_ATTACHED_POLICY
 - SVV_RLS_POLICY
 - SVV_RLS_RELASI
- Menambahkan fungsionalitas sedemikian rupa sehingga kueri yang hanya berisi fungsi skalar di klausa FROM sekarang menghasilkan kesalahan.
- Menambahkan pernyataan CREATE TABLE AS (CTAS) dengan fungsionalitas tabel target permanen ke cluster penskalaan konkurensi. Cluster penskalaan konkurensi sekarang mendukung lebih banyak kueri.
- Menambahkan tabel sistem berikut untuk melacak status redistribusi tabel setelah menjalankan pengubahan ukuran klasik pada RA3 cluster:
 - Tabel sistem SYS_RESTORE_STATE menunjukkan kemajuan redistribusi tingkat tabel.
 - Tabel sistem SYS_RESTORE_LOG menunjukkan throughput historis redistribusi data.
- Meningkatkan kecondongan irisan meminimalkan pada tabel EVEN setelah menjalankan pengubahan ukuran klasik pada tipe node. RA3 Ini juga berlaku untuk menambal 178 cluster yang menjalankan pengubahan ukuran klasik.

- Menambahkan dukungan untuk UNLOAD dengan EXTENSION pada cluster penskalaan konkurensi.
- Meningkatkan kinerja untuk kueri yang berisi Λ UDFs in HashJoins dan NestLoop bergabung.
- Meningkatkan kinerja Elastic Resize pada tipe RA3 node.
- Meningkatkan kinerja untuk kueri berbagi data.
- Meningkatkan kinerja kueri analisis yang dimulai secara manual dalam klaster penyediaan yang diubah ukurannya elastis dan grup kerja tanpa server.
- Meningkatkan kinerja kueri WLM auto dengan prediksi sumber daya yang lebih baik dalam manajemen beban kerja.
- Menghapus fungsionalitas peluncuran cluster ke VPC penyewaan khusus. Perubahan ini tidak memengaruhi penyewaan instans EC2 apa pun di VPC. Anda dapat memodifikasi penyewaan VPC Anda ke default dengan `modify-vpc-tenancy` AWS CLI perintah.
- Penyegaran manual tampilan terwujud sekarang didukung pada kluster penskalaan konkurensi yang disediakan dan komputasi penskalaan otomatis tanpa server.
- Menambahkan dukungan untuk literal INTERVAL ke fungsi EXTRACT. Misalnya, `EXTRACT('hours' from Interval '50 hours')` pengembalian 2 karena 50 jam ditafsirkan sebagai 2 hari dan 2 jam, dan komponen jam 2 diekstraksi.

Patch Pergeseran Merah Amazon 178

Versi cluster di tambalan ini:

- 1.0.63327 - Versi trek saat ini - Dirilis pada 9 Februari 2024
- 1.0.63313 - Versi trailing track - Dirilis pada 9 Februari 2024
- 1.0.60977 - Versi trailing track - Dirilis pada 15 Desember 2023
- 1.0.59596 - Versi trek saat ini - Dirilis pada 9 November 2023
- 1.0.58593 - Amazon Redshift Versi Tanpa Server - Dirilis pada 23 Oktober 2023
- 1.0.58558 - Versi trek saat ini - Dirilis pada 23 Oktober 2023
- 1.0.57864 - Versi trek saat ini - Dirilis pada 12 Oktober 2023
- 1.0.57850 - Amazon Redshift Versi Tanpa Server - Dirilis pada 12 Oktober 2023
- 1.0.56952 - Versi trek saat ini - Dirilis pada 25 September 2023
- 1.0.56970 - Amazon Redshift Versi Tanpa Server - Dirilis pada 25 September 2023

Fitur dan peningkatan baru di tambalan ini

- Amazon Redshift sekarang telah meningkatkan kinerja kueri berbagi data dengan mempercepat penyegaran metadata pada instance konsumen sementara perubahan data bersamaan terjadi pada instance produsen.
- Menambahkan dukungan untuk penyegaran otomatis dan inkremental tampilan terwujud di instans konsumen berbagi data Amazon Redshift saat tabel dasar tampilan terwujud merujuk ke data bersama.
- Menambahkan dukungan untuk menyimpan objek besar hingga 16 MB dalam tipe data SUPER. Saat menelan dari file sumber JSON, PARQUET, TEXT, dan CSV, Anda dapat memuat data atau dokumen semi-terstruktur sebagai nilai dalam tipe data SUPER, hingga 16 MB.
- Menambahkan dukungan untuk pengubahan ukuran elastis untuk penskalaan ke dan dari cluster Amazon Redshift simpul tunggal. RA3
- Cluster Amazon RA3 Redshift simpul tunggal sekarang dapat memperoleh manfaat dari peningkatan enkripsi, mengurangi waktu enkripsi secara keseluruhan dan meningkatkan ketersediaan gudang data selama proses enkripsi.
- Meningkatkan dukungan untuk kueri saat membuka sarang dan melepaskan data yang disimpan dalam tipe data SUPER.
- Meningkatkan kinerja menyegarkan tampilan terwujud dengan tipe data SUPER.
- Menambahkan dukungan untuk menggabungkan literal INTERVAL dengan fungsi ANY_VALUE.
- Streaming ingestion sekarang mendukung perintah SQL baru berikut untuk membersihkan data streaming: `DELETE FROM streaming_materialized_views WHERE <where filter clause>`
- Fungsi DECODE menggantikan nilai tertentu dengan nilai spesifik lain atau nilai default, tergantung pada hasil dari kondisi kesetaraan. DECODE sekarang membutuhkan tiga parameter berikut:
 - ekspresi
 - pencarian
 - result
- Menambahkan fungsionalitas ke prosedur tersimpan untuk memungkinkan penangkapan kesalahan konversi tipe data overflow data, dan penanganan di dalam blok penanganan pengecualian.
- Anda sekarang akan menerima kesalahan saat menanyakan keamanan tingkat baris atau hubungan yang dilindungi masking data dinamis jika Anda mengubah

`enable_case_sensitive_identifier` menjadi berbeda dari pengaturan default sesi. Selain itu, konfigurasi berikut akan diblokir ketika kebijakan keamanan tingkat baris atau masking data dinamis diterapkan di kluster yang disediakan atau namespace tanpa server:

```
ALTER USER <current_user> SET case-sensitive identifier.
```

- Perintah `MERGE` sekarang mendukung sintaks yang disederhanakan yang hanya membutuhkan target dan tabel sumber. Untuk informasi selengkapnya, lihat [MENGGABUNGKAN di Panduan Pengembang Database Amazon Redshift](#).
- Menambahkan dukungan untuk melampirkan kebijakan masking data dinamis yang identik ke beberapa pengguna atau peran dengan prioritas yang sama, atau tanpa menentukan prioritas.
- Anda sekarang dapat menentukan `COLLATION` saat menambahkan kolom baru melalui `ALTER TABLE ADD COLUMN`.
- Memperbaiki masalah yang menunda penegakan aturan QMR pada kluster penskalaan konkurensi dan Amazon Redshift Tanpa Server.
- Amazon Redshift Federated Query telah memperluas dukungan pushdown untuk zona waktu dengan stempel waktu di Amazon RDS untuk PostgreSQL dan Amazon Aurora PostgreSQL.
- Anda sekarang dapat menggunakan Amazon RDS untuk MySQL dan Aurora MySQL nama database MySQL dimulai dengan digit dengan kueri federasi.
- Menambahkan tampilan `SYS_ANALYZE_HISTORY`, yang berisi rincian catatan untuk operasi ANALISIS.
- Menambahkan tampilan `SYS_ANALYZE_COMPRESSION_HISTORY`, yang berisi detail catatan untuk operasi analisis kompresi selama perintah `COPY` atau `ANALYZE COMPRESSION`.
- Menambahkan tampilan `SYS_SESSION_HISTORY`, yang berisi detail rekaman yang terkait dengan sesi aktif, historis, dan dimulai ulang.
- Menambahkan tampilan `SYS_TRANSACTION_HISTORY`, yang berisi rincian catatan yang terkait dengan analisis tingkat transaksi yang menyediakan waktu yang dihabiskan untuk komit, jumlah datasha blok yang berkomitmen, dan tingkat isolasi.
- Menambahkan tampilan `SVV_REDSHIFT_SCHEMA_QUOTA`, yang berisi catatan yang terkait dengan kuota dan penggunaan disk saat ini untuk setiap skema dalam database.
- Menambahkan tampilan `SYS_PROCEDURE_CALL`, yang berisi catatan yang terkait dengan panggilan prosedur tersimpan, termasuk waktu mulai, waktu akhir, status panggilan prosedur tersimpan, dan hierarki panggilan untuk panggilan prosedur tersimpan bersarang.
- Menambahkan tampilan `SYS_CROSS_REGION_DATASHARING_USAGE`, yang berisi catatan yang terkait dengan pelacakan penggunaan berbagi data lintas wilayah.

- Menambahkan tampilan SYS_PROCEDURE_MESSAGES, yang berisi catatan yang terkait dengan informasi pelacakan tentang pesan prosedur tersimpan yang dicatat.
- Menambahkan tampilan SYS_UDF_LOG, yang berisi catatan yang terkait dengan melacak pesan log sistem dari panggilan fungsi, kesalahan, peringatan, atau jejak yang ditentukan pengguna bila berlaku.
- Menambahkan kolom baru IS_RECURSIVE, IS_NESTED, S3LIST_TIME, dan GET_PARTITION_TIME ke SYS_EXTERNAL_QUERY_DETAIL.
- Menambahkan MaxRPU, pengaturan kontrol biaya komputasi baru untuk Redshift Serverless. Dengan MaxRPU, Anda dapat secara opsional menentukan ambang batas komputasi atas untuk mengontrol biaya gudang data pada titik waktu yang berbeda dengan memilih tingkat komputasi maksimum yang dapat diskalakan Redshift Serverless per grup kerja.
- Mengoreksi output dari INTERVAL literal dengan string interval numerik. Misalnya, interval yang menentukan sebagai INTERVAL '1' YEAR sekarang kembali 1 YEAR bukan. "00:00:00 Selain itu, output dari literal INTERVAL terpotong ke komponen INTERVAL terkecil yang ditentukan. Misalnya, INTERVAL '1 day 1 hour 1 minute 1.123 seconds' HOUR TO MINUTE terpotong menjadi. 1 day 01:01:00

Patch Pergeseran Merah Amazon 177

Versi cluster di tambalan ini:

- 1.0.57922 - Versi trailing track - Dirilis pada 12 Oktober 2023
- 1.0.57799 - Amazon Redshift Versi Tanpa Server - Dirilis pada 10 Oktober 2023
- 1.0.57798 - Versi trek saat ini - Dirilis pada 10 Oktober 2023
- 1.0.57085 - Versi trailing track - Dirilis pada 26 September 2023
- 1.0.56899 - Amazon Redshift Versi Tanpa Server - Dirilis pada 21 September 2023
- 1.0.56754 - Versi trek saat ini - Dirilis pada 21 September 2023
- 1.0.56242 - Versi trek saat ini - Dirilis pada 11 September 2023
- 1.0.55539 - Amazon Redshift Versi Tanpa Server - Dirilis pada 28 Agustus 2023
- 1.0.55524 - Versi trek saat ini - Dirilis pada 28 Agustus 2023
- 1.0.54899 - Versi trek saat ini - Dirilis pada 15 Agustus 2023
- 1.0.54899 - Versi trek saat ini - Dirilis pada 14 Agustus 2023
- 1.0.54899 - Versi trek saat ini - Dirilis pada 15 Agustus 2023

- 1.0.54239 - Versi trek saat ini - Dirilis pada 3 Agustus 2023
- 1.0.54321 - Amazon Redshift Versi Tanpa Server - Dirilis pada 3 Agustus 2023

Fitur dan peningkatan baru di tambalan ini

- Menambahkan tampilan `SYS_MV_STATE`, yang berisi baris untuk setiap transisi status dari tampilan terwujud. `SYS_MV_STATE` dapat digunakan untuk pemantauan penyegaran MV untuk instans yang disediakan Amazon Redshift Serverless dan Amazon Redshift.
- Menambahkan tampilan `SYS_USERLOG`, yang mencatat detail untuk perubahan pada pengguna database untuk Buat pengguna, Jatuhkan pengguna, Ubah pengguna (ganti nama), Ubah pengguna (ubah properti).
- Menambahkan tampilan `SYS_COPY_REPLACEMENTS`, yang menampilkan log yang mencatat ketika karakter UTF-8 yang tidak valid digantikan oleh perintah `COPY` dengan opsi `ACCEPTINVCHARS`.
- Menambahkan tampilan `SYS_SPATIAL_SIMPLIFY`, yang berisi informasi tentang objek geometri spasial yang disederhanakan menggunakan perintah `COPY`.
- Menambahkan tampilan `SYS_VACUUM_HISTORY`, yang dapat Anda gunakan untuk melihat detail dan hasil operasi `VACUUM`.
- Menambahkan tampilan `SYS_SCHEMA_QUOTA_VIOLATIONS` untuk merekam kejadian, stempel waktu, XID, dan informasi berguna lainnya saat kuota skema terlampaui.
- Menambahkan tampilan `SYS_RESTORE_STATE`, yang dapat Anda gunakan memantau kemajuan redistribusi setiap tabel di cluster selama pengubahan ukuran klasik asinkron.
- Menambahkan tampilan `SYS_EXTERNAL_QUERY_ERROR` yang mengembalikan informasi tentang kesalahan pemindaian Redshift Spectrum.
- Menambahkan parameter tag ke perintah `CREATE MODEL`, sehingga Anda sekarang dapat melacak biaya pelatihan dengan pekerjaan pelatihan autopilot.
- Menambahkan nama domain khusus (CNAME) untuk klaster Amazon Redshift.
- Menambahkan dukungan pratinjau untuk Apache Iceberg, memungkinkan pelanggan menjalankan kueri analitik pada tabel Apache Iceberg dalam Amazon Redshift.
- Menambahkan dukungan untuk menggunakan peran pengguna dengan grup parameter dalam manajemen beban kerja (WLM).
- Menambahkan dukungan untuk pemasangan otomatis AWS Glue Data Catalog, sehingga memudahkan pelanggan untuk menjalankan kueri di danau data mereka.

- Menambahkan fungsionalitas sedemikian rupa sehingga menggunakan fungsi pengelompokan tanpa klausa GROUP BY atau menggunakan operasi pengelompokan dalam klausa WHERE menghasilkan kesalahan.
- Menambahkan fungsionalitas ke prosedur tersimpan untuk memungkinkan penangkapan pembagian dengan nol kesalahan dan penanganan di dalam blok penanganan pengecualian.
- Memperbaiki bug yang mencegah kueri menggunakan penskalaan konkurensi untuk menulis data ke tabel saat tabel sumber adalah tabel berbagi data.
- Memperbaiki identifier case-sensitive yang didokumentasikan di `enable_case_sensitive_identifier` untuk sekarang bekerja dengan pernyataan MERGE.
- Memperbaiki bug yang kueri pada fungsi `pg_get_late_binding_view_cols()` mungkin diabaikan sesekali. Anda sekarang selalu dapat membatalkan pertanyaan tersebut.
- Meningkatkan kinerja untuk kueri berbagi data yang berjalan pada konsumen saat menjalankan pekerjaan vakum pada produsen.
- Meningkatkan kinerja untuk berbagi data dan kueri penskalaan konkurensi, terutama dengan perubahan data bersamaan pada produsen atau saat pembongkaran ke instance penskalaan konkurensi yang dilampirkan ke konsumen.

Patch Pergeseran Merah Amazon 176

Versi cluster di tambalan ini:

- 1.0.56738 - Versi trailing track - dirilis pada 21 September 2023
- 1.0.55837 - Versi trailing track - dirilis pada 11 September 2023
- 1.0.54776 - Versi trek saat ini - dirilis pada 15 Agustus 2023
- 1.0.54052 - Versi trek saat ini - Dirilis pada 26 Juli 2023
- 1.0.53642 - Amazon Redshift Versi Tanpa Server - Dirilis pada 20 Juli 2023
- 1.0.53301 - Versi trek saat ini - Dirilis pada 20 Juli 2023
- 1.0.52943 - Amazon Redshift Versi Tanpa Server - Dirilis pada 7 Juli 2023
- 1.0.52931 - Versi trek saat ini - Dirilis pada 7 Juli 2023
- 1.0.52194 - Amazon Redshift Versi Tanpa Server - Dirilis pada 21 Juni 2023
- 1.0.51986 - Versi trek saat ini - Dirilis pada 16 Juni 2023
- 1.0.51594 - Versi trek saat ini - Dirilis pada 9 Juni 2023

Fitur dan peningkatan baru di tambalan ini

- Peningkatan penanganan kesalahan saat menulis GROUP BY () untuk kumpulan pengelompokan kosong. Ini diabaikan sebelumnya dan sekarang mengembalikan kesalahan parser.
- Peningkatan kinerja untuk menyegarkan tampilan terwujud secara bertahap dengan kolom SUPER.
- ALTER TABLE <target_tbl>APPEND FROM <streaming_mv>— (ATA) Perintah SQL sekarang mendukung pemindahan semua catatan dari streaming materialized view (MV) sebagai sumber, selain tabel sebagai sumber, ke tabel target. Dukungan untuk ATA pada streaming MVs memungkinkan pengguna untuk dengan cepat membersihkan semua catatan dalam MV streaming dengan memindahkannya ke tabel lain untuk mengelola pertumbuhan data.
- TRUNCATE <streaming_mv>— Perintah SQL sekarang mendukung pemotongan semua catatan dalam tampilan terwujud streaming (MV), selain tabel. TRUNCATE menghapus semua catatan di MV streaming, sambil membiarkan struktur MV streaming tetap utuh. Menjalankan TRUNCATE pada streaming MVs memungkinkan pelanggan untuk dengan cepat membersihkan semua catatan dalam streaming MV untuk mengelola pertumbuhan data.
- Menambahkan fungsionalitas untuk klausa QUALIFY ke perintah SELECT.
- Dukungan pembelajaran mesin Redshift untuk peramalan deret waktu dengan mengintegrasikan dengan Amazon Forecast.
- AWS Glue Data Catalog pemasangan otomatis didukung untuk menyederhanakan kueri data lake tanpa langkah tambahan untuk membuat referensi skema eksternal.
- Mengubah kebijakan RLS sekarang didukung. Lihat dokumentasi untuk detail selengkapnya di [KEBIJAKAN ALTER RLS](#).
- Lambda UDFs sekarang mendukung parameter volatilitas fungsi STABLE dalam pernyataan CREATE FUNCTION. Ketika parameter STABLE digunakan dalam pernyataan CREATE FUNCTION dan Lambda UDF dipanggil beberapa kali, dengan argumen yang sama, jumlah yang diharapkan dari pemanggilan fungsi Lambda UDF berkurang. Kategori volatilitas fungsi STABLE dijelaskan secara lebih rinci dalam [parameter CREATE FUNCTION](#).
- Beberapa peningkatan kinerja Lambda UDF. Secara khusus, peningkatan dukungan batch rekaman saat menanyakan tabel yang dilindungi oleh kebijakan keamanan tingkat baris (RLS).
- Pengurangan waktu enkripsi keseluruhan untuk RA3 cluster Amazon Redshift dan peningkatan ketersediaan gudang data selama enkripsi. Untuk informasi selengkapnya, lihat [Enkripsi basis data Amazon Redshift](#).
- Tampilan sistem baru SYS_MV_REFRESH_HISTORY telah ditambahkan ke Redshift. Tampilan SYS_MV_REFRESH_HISTORY berisi baris untuk aktivitas penyegaran tampilan terwujud.

Menggunakan SYS_MV_REFRESH_HISTORY, Anda dapat memeriksa riwayat penyegaran tampilan terwujud. SYS_MV_REFRESH_HISTORY dapat dilihat oleh semua pengguna. Pengguna super dapat melihat semua baris; pengguna biasa hanya dapat melihat data mereka sendiri.

Sebuah kolom baru SPILLED_BLOCK_LOCAL_DISK telah ditambahkan ke tampilan sistem SYS_QUERY_DETAIL. Kolom baru SPILLED_BLOCK_LOCAL_DISK membantu pelanggan untuk menentukan blok tumpah ke disk lokal. Anda dapat menggunakan SYS_QUERY_DETAIL untuk melihat detail kueri pada tingkat langkah. SYS_QUERY_DETAIL terlihat oleh semua pengguna. Pengguna super dapat melihat semua baris; pengguna biasa hanya dapat melihat metadata yang dapat mereka akses.

- Tampilan sistem baru, SYS_QUERY_TEXT, telah ditambahkan ke Amazon Redshift Tanpa Server dan Amazon Redshift yang disediakan. [Tampilan SYS_QUERY_TEXT mirip dengan SVL_STATEMENTTEXT untuk cluster yang disediakan.](#) Gunakan sequence kolom dalam tampilan SYS_QUERY_TEXT untuk mendapatkan teks pernyataan SQL lengkap.

Patch Pergeseran Merah Amazon 175

Versi cluster di tambalan ini:

- 1.0.53064 - Versi trek saat ini - Dirilis pada 7 Juli 2023
- 1.0.51973 - Versi trek saat ini - Dirilis pada 16 Juni 2023
- 1.0.51781 - Versi trek saat ini - Dirilis pada 10 Juni 2023
- 1.0.51314 - Amazon Redshift Versi Tanpa Server - Dirilis pada 3 Juni 2023
- 1.0.51304 - Versi trek saat ini - Dirilis pada 2 Juni 2023
- 1.0.50708 - Versi Track saat ini - Dirilis pada 19 Mei 2023
- 1.0.50300 - Versi trek saat ini - Dirilis pada 8 Mei 2023
- 1.0.49710 - Amazon Redshift Versi Tanpa Server - Dirilis pada 28 April 2023
- 1.0.49676 - Versi trek saat ini - Dirilis pada 28 April 2023

Fitur dan peningkatan baru di tambalan ini

- Perbaikan bug minor.
- Penyerapan streaming Amazon Redshift sekarang mendukung konsumsi streaming lintas wilayah di mana topik sumber Anda (Amazon Kinesis Data Streams KDS) atau Amazon Managed

Streaming for Apache Kafka (MSK) dapat ditemukan di wilayah yang AWS berbeda dari wilayah tempat gudang data Amazon Redshift Anda berada. AWS Dokumentasi di [Memulai dengan konsumsi streaming dari Amazon Kinesis Data Streams](#) telah direvisi dan menjelaskan bagaimana kata kunci REGION digunakan.

- Penyesuaian musim panas Mesir.
- Peningkatan waktu keseluruhan untuk enkripsi RA3 cluster.

Patch Pergeseran Merah Amazon 174

1.0.51296 - Dirilis pada 2 Juni 2023

Lepaskan ke trailing track. Tidak ada catatan rilis.

1.0.50468 - Dirilis pada 12 Mei 2023

Rilis pemeliharaan. Tidak ada catatan rilis.

1.0.49780, 1.0.49868, dan 1.0.49997 - Dirilis pada 28 April 2023

Catatan rilis untuk versi ini:

- Peningkatan dukungan batching untuk Lambda UDF.
- Batching tambahan untuk Lambda UDF.
- Perintah MERGE SQL baru untuk menerapkan perubahan data sumber ke tabel Amazon Redshift.
- Kemampuan masking data dinamis baru untuk menyederhanakan proses perlindungan data sensitif di gudang data Amazon Redshift.
- Kontrol akses terpusat baru untuk berbagi data dengan Lake Formation yang memungkinkan mengelola hibah izin, melihat kontrol akses, dan mengaudit izin pada tabel dan tampilan di database Amazon Redshift menggunakan Lake Formation dan Console. APIs AWS
- Penyesuaian musim panas Mesir.

1.0.49087 - Dirilis pada 12 April 2023

Rilis pemeliharaan. Tidak ada catatan rilis.

1.0.48805 - Dirilis pada 5 April 2023

Catatan rilis untuk versi ini:

- Amazon Redshift memperkenalkan peningkatan kinerja tambahan untuk kueri string-heavy menggunakan BYTEDICT, pengkodean kompresi baru di Amazon Redshift yang mempercepat pemrosesan data berbasis string antara 5x hingga 63x dibandingkan dengan pengkodean kompresi alternatif seperti LZO atau ZSTD. Untuk informasi selengkapnya tentang fitur ini, lihat [Pengkodean kompresi di Panduan Pengembang](#) Database Amazon Redshift.

1.0.48004 - Dirilis pada 17 Maret 2023

Rilis pemeliharaan. Tidak ada catatan rilis.

1.0.47470 - Dirilis pada 11 Maret 2023

Catatan rilis untuk versi ini:

- Meningkatkan kinerja kueri `padapg_catalog.svv_table_info`. Juga menambahkan kolom baru `create_time`. Saat membuat tabel, kolom ini menyimpan date/time stempel di UTC.
- Menambahkan dukungan untuk menentukan batas waktu tingkat sesi pada kueri federasi.

Patch Pergeseran Merah Amazon 173

1.0.49788 - Dirilis pada 28 April 2023

Catatan rilis untuk versi ini:

- Penyesuaian musim panas Mesir.

1.0.49074 - Dirilis pada 12 April 2023

Catatan rilis untuk versi ini:

- Konfigurasi zona waktu diperbarui ke rilis perpustakaan IANA 2022g.

1.0.48766 - Dirilis pada 5 April 2023

Rilis pemeliharaan. Tidak ada catatan rilis.

1.0.48714 - Dirilis pada 5 April 2023

Rilis pemeliharaan. Tidak ada catatan rilis.

1.0.48022 - Dirilis pada 17 Maret 2023

Rilis pemeliharaan. Tidak ada catatan rilis.

1.0.47357 - Dirilis pada 7 Maret 2023

Rilis pemeliharaan. Tidak ada catatan rilis.

1.0.46987 - Dirilis pada 24 Februari 2023

Rilis pemeliharaan. Tidak ada catatan rilis.

1.0.46806 - Dirilis pada 18 Februari 2023

Rilis pemeliharaan. Tidak ada catatan rilis.

1.0.46607 - Dirilis pada 13 Februari 2023

Catatan rilis untuk versi ini:

- Kami sekarang secara otomatis mengonversi tabel dengan secara manual mengatur kunci sortir interleaved ke kunci sortir majemuk jika gaya distribusinya telah disetel ke `DISTSTYLE KEY`, untuk meningkatkan kinerja tabel ini. Ini dilakukan pada saat memulihkan snapshot ke Amazon Redshift Serverless.

1.0.45698 - Dirilis pada 20 Januari 2023

Catatan rilis untuk versi ini:

- Menambahkan parameter ekstensi file ke perintah `UNLOAD`, sehingga ekstensi file secara otomatis ditambahkan ke nama file.

- Mendukung perlindungan objek yang dilindungi RLS secara default saat menambahkannya ke datashare atau jika mereka sudah menjadi bagian dari datashare. Administrator sekarang dapat mematikan RLS untuk datashares untuk memungkinkan konsumen mengakses objek yang dilindungi.
- Menambahkan tabel sistem baru untuk pemantauan:SVV_ML_MODEL_INFO,SVV_MV_DEPENDENCY, danSYS_LOAD_DETAIL. Juga menambahkan kolom data_skewness dan time_skewness ke tabel sistemSYS_QUERY_DETAIL.

Patch Pergeseran Merah Amazon 172

Versi cluster di tambalan ini:

- 1.0.46534 - Dirilis pada 18 Februari 2023
- 1.0.46523 - Dirilis pada 13 Februari 2023
- 1.0.46206 - Dirilis pada 1 Februari 2023
- 1.0.45603 - Dirilis pada 20 Januari 2023
- 1.0.44924 — Dirilis pada 19 Desember 2022
- 1.0.44903 — Dirilis pada 18 Desember 2022
- 1.0.44540 — Dirilis pada 13 Desember 2022
- 1.0.44126 — Dirilis pada 23 November 2022
- 1.0.43980 — Dirilis pada 17 November 2022

Fitur dan peningkatan baru di tambalan ini

- Tabel yang dibuat oleh CTAS adalah AUTO secara default.
- Menambahkan dukungan untuk keamanan tingkat baris (RLS) pada tampilan terwujud.
- Meningkatkan batas waktu S3 untuk meningkatkan berbagi data lintas wilayah.
- Menambahkan fungsi spasial baruST_GeomFromGeohash.
- Meningkatkan pemilihan otomatis kunci distribusi dari kunci primer komposit untuk meningkatkan out-of-the-box kinerja.
- Menambahkan kunci primer otomatis ke kunci distribusi untuk tabel dengan kunci primer komposit, meningkatkan out-of-the-box kinerja.

- Meningkatkan penskalaan konkurensi untuk memungkinkan lebih banyak kueri untuk diskalakan bahkan saat data berubah.
- Meningkatkan kinerja kueri berbagi data.
- Menambahkan metrik probabilitas Machine Learning untuk model klasifikasi.
- Menambahkan tabel sistem baru untuk pemantauan:SVV_USER_INFO,SVV_MV_INFO,SYS_CONNECTION_LOG,SYS_DATASHARE_USAGE_PRODUCED danSYS_DATASHARE_CHANGE_LOG.
- Menambahkan dukungan untuk menanyakan kolom VARBYTE di tabel eksternal untuk jenis file Parquet dan ORC.

Patch Pergeseran Merah Amazon 171

Versi cluster di tambalan ini:

- 1.0.43931 — Dirilis pada 16 November 2022
- 1.0.43551 — Dirilis pada 5 November 2022
- 1.0.43331 — Dirilis pada 29 September 2022
- 1.0.43029 — Dirilis pada 26 September 2022

Fitur dan peningkatan baru di tambalan ini

- Dukungan CONNECT BY: Menambahkan dukungan untuk konstruksi CONNECT BY SQL, memungkinkan Anda secara rekursif menanyakan data hierarkis di gudang data Anda berdasarkan hubungan induk-anak dalam kumpulan data tersebut.

Patch Pergeseran Merah Amazon 170

Versi cluster di tambalan ini:

- 1.0.43922 — Dirilis pada 21 November 2022
- 1.0.43573 — Dirilis pada 7 November 2022
- 1.0.41881 — Dirilis pada 20 September 2022
- 1.0.41465 — Dirilis pada 7 September 2022
- 1.0.40325 — Dirilis pada 27 Juli 2022

Fitur dan peningkatan baru di tambalan ini

- `ST_GeomfromGeoJSON`: Membangun objek geometri spasial Amazon Redshift dari `VARCHAR` dalam representasi `GeoJSON`.

Patch Pergeseran Merah Amazon 169

Versi cluster di tambalan ini:

- 1.0.41050 — Dirilis pada 7 September 2022
- 1.0.40083 — Dirilis pada 16 Juli 2022
- 1.0.39734 — Dirilis pada 7 Juli 2022
- 1.0.39380 — Dirilis pada 23 Juni 2022
- 1.0.39251 — Dirilis pada 15 Juni 2022
- 1.0.39009 — Dirilis pada 8 Juni 2022

Fitur dan peningkatan baru di tambalan ini

- Menambahkan peran sebagai parameter untuk perintah `Alter Default Privileges` untuk mendukung Kontrol Akses berbasis peran.
- Menambahkan parameter `ACCEPTINVCHARS` untuk mendukung penggantian karakter UTF-8 yang tidak valid saat menyalin dari file `Parquet` dan `ORC`.
- Menambahkan fungsi `OBJECT (k, v)` untuk membangun objek `SUPER` dari pasangan kunci dan nilai.

Patch Pergeseran Merah Amazon 168

Versi cluster di tambalan ini:

- 1.0.38698 — Dirilis pada 25 Mei 2022
- 1.0.38551 — Dirilis pada 20 Mei 2022
- 1.0.38463 — Dirilis pada 18 Mei 2022
- 1.0.38361 — Dirilis pada 13 Mei 2022
- 1.0.38199 — Dirilis pada 9 Mei 2022

- 1.0.38112 — Dirilis pada 6 Mei 2022
- 1.0.37684 — Dirilis pada 20 April 2022

Fitur dan peningkatan baru di tambalan ini

- Menambahkan dukungan untuk tipe model Linear Learner di Amazon Redshift ML.
- Menambahkan opsi SNAPSHOT untuk tingkat isolasi transaksi SQL.
- Menambahkan `farmhashFingerprint64` sebagai algoritma hashing baru untuk VARBYTE dan VARCHAR data.
- Mendukung fungsi AVG dalam penyegaran inkremental tampilan terwujud.
- Mendukung sub-kueri berkorelasi pada tabel eksternal di Redshift Spectrum.
- Untuk meningkatkan kinerja out-of-the-box kueri, Amazon Redshift secara otomatis memilih kunci utama kolom tunggal untuk tabel tertentu sebagai kunci distribusi.

Perubahan perilaku di Amazon Redshift

Seiring Amazon Redshift terus berkembang dan meningkat, perubahan perilaku tertentu diperkenalkan untuk meningkatkan kinerja, keamanan, dan pengalaman pengguna. Halaman ini berfungsi sebagai sumber daya komprehensif bagi Anda untuk tetap mendapat informasi tentang pembaruan penting ini, mengambil tindakan, dan menghindari potensi gangguan pada beban kerja Anda.

Perubahan perilaku yang akan datang

Berikut ini menjelaskan perubahan perilaku yang akan datang.

Topik

- [Scalar UDFs Python akan mencapai akhir dukungan setelah 30 Juni 2026](#)
- [Perubahan Perilaku Penyegaran Otomatis Tampilan Terwujud \(MV\) setelah 27 Februari 2026](#)
- [Amazon Redshift tidak akan mendukung fungsi yang mengakses informasi konsumen melalui datasharing setelah 16 Februari 2026](#)
- [Perubahan versi Minimum Transport Layer Security \(TLS\) efektif mulai 31 Januari 2026](#)
- [Amazon Redshift tidak akan mendukung pembuatan UDFs Python skalar baru setelah 30 Oktober 2025](#)

Scalar UDFs Python akan mencapai akhir dukungan setelah 30 Juni 2026

Amazon Redshift akan mengakhiri dukungan untuk Python UDFs setelah 30 Juni 2026. Sebagai alternatif, kami sarankan Anda menggunakan Lambda UDFs.

Lambda UDFs memiliki keunggulan sebagai berikut dibandingkan Python: UDFs

- Lambda UDFs dapat terhubung ke layanan eksternal dan APIs dari dalam logika UDF.
- Lambda menggunakan sumber daya UDFs komputasi Lambda. UDFs Lambda komputasi berat atau intensif memori tidak memengaruhi kinerja kueri atau konkurensi sumber daya Amazon Redshift.
- UDFs Dukungan Lambda menjalankan kode Python. Lambda UDFs mendukung beberapa runtime Python tergantung pada kasus penggunaan tertentu. Untuk informasi selengkapnya, lihat [Membangun dengan Python](#) di Panduan AWS Lambda Pengembang.

- Anda dapat mengisolasi eksekusi kode kustom dalam batas layanan terpisah. Ini menyederhanakan pemeliharaan, pemantauan, penganggaran, dan manajemen izin.

Untuk informasi tentang cara membuat dan menggunakan Lambda UDFs, lihat [Skalar UDFs Lambda](#) di Panduan Pengembang Database Amazon Redshift. [Untuk informasi tentang mengonversi UDFs Python yang ada ke UDFs Lambda, lihat posting blog.](#)

Perubahan Perilaku Penyegaran Otomatis Tampilan Terwujud (MV) setelah 27 Februari 2026

Mulai 27 Februari 2026, kueri REFRESH Otomatis untuk tampilan terwujud Amazon Redshift dijalankan sebagai kueri pengguna daripada proses otonom latar belakang. Akibatnya, kueri Auto REFRESH sekarang berjalan dengan prioritas yang sama dengan kueri pengguna lainnya.

Perubahan ini meningkatkan kesegaran tampilan terwujud dengan Auto REFRESH diaktifkan, membantu mereka tetap up to date dengan perubahan terbaru pada tabel dasar mereka dibandingkan dengan perilaku sebelumnya.

Catatan: Fitur perubahan perilaku MV Auto REFRESH hanya diaktifkan untuk klaster Amazon Redshift Provisioned pada Track CURRENT dari rilis patch P198 dan yang lebih baru. Saat ini dinonaktifkan di Tanpa Server.

Amazon Redshift tidak akan mendukung fungsi yang mengakses informasi konsumen melalui datasharing setelah 16 Februari 2026

Mulai 16 Februari 2026, Amazon Redshift tidak akan lagi mendukung penggunaan `user_is_member_of` dan fungsi terkait yang mengakses informasi pengguna, peran, atau grup konsumen melalui datasharing.

Perubahan versi Minimum Transport Layer Security (TLS) efektif mulai 31 Januari 2026

Mulai 31 Januari 2026, Amazon Redshift akan memberlakukan versi Transport Layer Security (TLS) minimum 1.2. Koneksi masuk yang menggunakan TLS versi 1.0 atau 1.1 akan ditolak. Ini berlaku untuk cluster yang disediakan Amazon Redshift dan grup kerja tanpa server. Gudang data Amazon Redshift yang tidak menggunakan TLS tidak akan terpengaruh oleh perubahan ini.

Pembaruan ini mungkin memengaruhi Anda jika Anda menggunakan TLS versi 1.0 atau 1.1 untuk terhubung ke Amazon Redshift.

Untuk memverifikasi versi TLS yang sedang Anda gunakan, Anda dapat:

Untuk Amazon Redshift Provisioned: Periksa kolom `sslversion` di tabel sistem `STL_CONNECTION_LOG` [1].

Untuk Amazon Redshift Serverless Workgroup: Periksa kolom `ssl_version` di tabel sistem `SYS_CONNECTION_LOG` [2].

Untuk mempertahankan akses tanpa gangguan ke gudang data Amazon Redshift Anda setelah perubahan ini, ikuti langkah-langkah yang tercantum di bawah ini:

1. Perbarui klien Anda untuk mendukung TLS 1.2 atau lebih tinggi
2. Instal versi driver terbaru dengan dukungan TLS 1.2+

Sebaiknya gunakan driver Amazon Redshift versi terbaru [3] jika memungkinkan.

[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 Redshift tidak akan mendukung pembuatan UDFs Python skalar baru setelah 30 Oktober 2025

Amazon Redshift tidak akan lagi mendukung pembuatan Python baru UDFs setelah 30 Oktober 2025. Python yang ada UDFs akan terus berfungsi secara normal. Kami sangat menyarankan agar Anda memigrasikan UDFs Python yang ada ke UDFs Lambda sebelum tanggal ini.

Lambda UDFs memiliki keunggulan sebagai berikut dibandingkan Python: UDFs

- Lambda UDFs dapat terhubung ke layanan eksternal dan APIs dari dalam logika UDF.
- Lambda menggunakan sumber daya UDFs komputasi Lambda. UDFs Lambda komputasi berat atau intensif memori tidak memengaruhi kinerja kueri atau konkurensi sumber daya Amazon Redshift.

- UDFs Dukungan Lambda menjalankan kode Python. Lambda UDFs mendukung beberapa runtime Python tergantung pada kasus penggunaan tertentu. Untuk informasi selengkapnya, lihat [Membangun dengan Python](#) di Panduan AWS Lambda Pengembang.
- Anda dapat mengisolasi eksekusi kode kustom dalam batas layanan terpisah. Ini menyederhanakan pemeliharaan, pemantauan, penganggaran, dan manajemen izin.

Untuk informasi tentang cara membuat dan menggunakan Lambda UDFs, lihat [Skalar UDFs Lambda](#) di Panduan Pengembang Database Amazon Redshift. [Untuk informasi tentang mengonversi UDFs Python yang ada ke UDFs Lambda, lihat posting blog.](#)

Perubahan perilaku terbaru

Topik

- [Amazon Redshift menggunakan Database Zona Waktu up-to-date IANA setelah 26 Agustus 2025](#)
- [Amazon Redshift Serverless RPU berubah efektif setelah 15 Agustus 2025](#)
- [Perubahan pencatatan audit basis data efektif setelah 10 Agustus 2025](#)
- [Perubahan Titik Akhir Cloud Pribadi Virtual untuk grup kerja tanpa server efektif setelah 27 Juni 2025](#)
- [Perubahan pemantauan kueri efektif setelah 2 Mei 2025](#)
- [Perubahan keamanan efektif setelah 10 Januari 2025](#)

Amazon Redshift menggunakan Database Zona Waktu up-to-date IANA setelah 26 Agustus 2025

Mulai 26 Agustus 2025, Amazon Redshift menghitung zona waktu dengan mengadopsi patch Database Zona Waktu IANA terbaru. Perubahan ini mengubah cara konversi tanggal dan waktu berfungsi untuk zona waktu dan periode waktu tertentu. [Pembaruan ini memengaruhi konversi zona waktu eksplisit, seperti yang dilakukan dengan fungsi CONVERT_TIMEZONE atau perintah TIMEZONE dan AT TIME ZONE, serta konversi implisit yang terjadi selama operasi casting tipe, terutama antara format TIMESTAMP dan TIMESTAMPTZ.](#)

Berikut ini adalah daftar pembaruan untuk kombinasi zona waktu dan periode waktu:

- Zona waktu sekarang dengan benar mengamati waktu musim panas (DST) setelah tahun 2038. Sebelumnya, tidak ada zona waktu yang diamati DST setelah 2038.

- Zona America/Toronto waktu, dan zona waktu yang terhubung dengannya, memiliki sakelar DST pada tahun 1947-1950 pukul 2 pagi waktu setempat, bukan pada tengah malam.
- Amazon Redshift sekarang mencerminkan waktu rata-rata lokal (LMT) dengan benar untuk periode sebelum standardisasi untuk semua zona waktu. Periode ini khusus untuk zona waktu, dengan sebagian besar zona waktu beralih ke standardisasi sebelum pertengahan abad ke-19.
- EET,CET, WET dan sekarang MET diperlakukan sebagai zona waktu normal, bukan singkatan.
- Nama zona waktu berikut tidak lagi ada di Amazon Redshift:
 - Asia/Riyadh87
 - Asia/Riyadh88
 - Asia/Riyadh89
 - Mideast/Riyadh87
 - Mideast/Riyadh88
 - Mideast/Riyadh89
 - US/Pacific-New

Untuk informasi selengkapnya tentang Database Zona Waktu IANA, lihat Database [Zona Waktu](#) di situs web IANA Time Zone Database.

Amazon Redshift Serverless RPU berubah efektif setelah 15 Agustus 2025

Mulai 15 Agustus 2025, kuota AWS akun untuk Amazon Redshift Serverless basis Redshift Processing Units RPUs () lebih besar dari RPUs 3.200 atau 1,5 kali basis agregat maksimum Anda dari enam bulan sebelumnya. RPUs

Perubahan pencatatan audit basis data efektif setelah 10 Agustus 2025

Mulai 10 Agustus 2025, Amazon Redshift membuat perubahan pada Database Audit Logging, yang memerlukan tindakan Anda. Amazon Redshift mencatat informasi tentang koneksi dan aktivitas pengguna di database Anda ke bucket Amazon S3 dan. CloudWatch Setelah 10 Agustus 2025, Amazon Redshift akan menghentikan pencatatan audit database ke bucket Amazon S3 Anda yang memiliki kebijakan bucket yang menentukan PENGGUNA IAM Redshift. Sebaiknya perbarui kebijakan Anda untuk menggunakan Redshift SERVICE-PRINCIPAL sebagai gantinya, dalam kebijakan bucket S3 untuk pencatatan audit. Untuk informasi tentang pencatatan audit, lihat [izin bucket untuk pencatatan audit Amazon Redshift](#).

Untuk menghindari gangguan logging, tinjau dan perbarui kebijakan bucket S3 Anda untuk memberikan akses ke prinsipal layanan Redshift di wilayah terkait sebelum 10 Agustus 2025. Untuk informasi tentang pencatatan audit database, lihat [Log file di Amazon S3](#)

Untuk pertanyaan atau masalah, hubungi AWS dukungan di tautan berikut: [AWS Support](#).

Perubahan Titik Akhir Cloud Pribadi Virtual untuk grup kerja tanpa server efektif setelah 27 Juni 2025

Mulai 27 Juni 2025, Amazon Redshift membuat perubahan pada dukungan Virtual Private Cloud Endpoint (VPCE) untuk grup kerja tanpa server. Sebelum tanggal ini, Amazon Redshift menerapkan dengan titik akhir ke dalam Availability Zone (AZ) tunggal selama pembuatan workgroup, dan memperluas dukungan VPCE hingga tiga dari waktu ke waktu. AZs Setelah tanggal ini, Amazon Redshift menerapkan VPCEs hingga tiga Availability Zone yang ditentukan selama pembuatan workgroup.

Untuk informasi selengkapnya, lihat [Pertimbangan saat menggunakan Amazon Redshift Serverless](#).

Untuk pertanyaan atau masalah, hubungi AWS dukungan di tautan berikut: [AWS Support](#).

Topik

- [Perubahan pemantauan kueri efektif setelah 2 Mei 2025](#)
- [Perubahan keamanan efektif setelah 10 Januari 2025](#)

Perubahan pemantauan kueri efektif setelah 2 Mei 2025

Efektif 2 Mei 2025, kami tidak akan lagi menawarkan metrik waktu CPU Kueri (`max_query_cpu_time`) dan penggunaan CPU Kueri (`max_query_cpu_percentage`) dari tab Batas Kueri untuk grup kerja Redshift Tanpa Server yang ada dan yang baru dibuat. Setelah tanggal ini, kami akan secara otomatis menghapus semua batas kueri berdasarkan metrik ini di semua grup kerja Redshift Tanpa Server.

Batas kueri dirancang untuk menangkap kueri runaway. Namun, Query CPU time (`max_query_cpu_time`) dan Query CPU usage (`max_query_cpu_percentage`) dapat bervariasi selama masa kueri, dan dengan demikian bukan metode yang efektif secara konsisten untuk menangkap kueri runaway. Untuk menangkap kueri runaway, sebaiknya Anda memanfaatkan metrik pemantauan kueri yang memberikan informasi yang konsisten dan dapat ditindaklanjuti. Beberapa contoh termasuk:

- Waktu eksekusi kueri (`max_query_execution_time`): Untuk memastikan kueri selesai dalam jangka waktu yang diharapkan.
- Mengembalikan jumlah baris (`max_scan_row_count`): Untuk memantau skala data yang sedang diproses.
- Query queue time (`max_query_queue_time`): Untuk mengidentifikasi kueri yang menghabiskan waktu antrian.

Untuk mengetahui daftar lengkap metrik yang didukung, lihat Metrik [pemantauan kueri untuk Amazon Redshift Tanpa Server](#).

Perubahan keamanan efektif setelah 10 Januari 2025

Keamanan adalah prioritas utama kami di Amazon Web Services (AWS). Untuk itu, kami semakin memperkuat postur keamanan lingkungan Amazon Redshift dengan memperkenalkan default keamanan yang ditingkatkan yang membantu Anda mematuhi praktik terbaik dalam keamanan data tanpa memerlukan pengaturan tambahan dan mengurangi risiko potensi kesalahan konfigurasi. Untuk menghindari potensi gangguan, tinjau konfigurasi, skrip, dan alat pembuatan klaster dan grup kerja tanpa server yang disediakan untuk membuat perubahan yang diperlukan agar selaras dengan pengaturan default baru sebelum tanggal efektif.

Akses publik dinonaktifkan secara default

Setelah 10 Januari 2025, [aksesibilitas publik](#) akan dinonaktifkan secara default untuk semua cluster yang baru dibuat, dan untuk cluster yang dipulihkan dari snapshot. Dengan rilis ini, secara default, koneksi ke cluster hanya akan diizinkan dari aplikasi klien dalam Virtual Private Cloud (VPC) yang sama. Untuk mengakses gudang data Anda dari aplikasi di VPC lain, konfigurasi akses [lintas-VPC](#). Perubahan ini akan tercermin dalam operasi `CreateCluster` dan `RestoreFromClusterSnapshot` API, serta SDK dan AWS CLI perintah yang sesuai. Jika Anda membuat klaster yang disediakan dari konsol Amazon Redshift, maka klaster memiliki akses publik yang dinonaktifkan secara default.

Jika Anda masih memerlukan akses publik, Anda harus mengganti default dan menyetel `PubliclyAccessible` parameter ke `true` saat Anda menjalankan `CreateCluster` atau operasi `RestoreFromClusterSnapshot` API. Dengan klaster yang dapat diakses publik, sebaiknya Anda menggunakan grup keamanan atau daftar kontrol akses jaringan (ACLs) untuk membatasi akses. Untuk informasi selengkapnya, lihat [Grup keamanan VPC](#) dan [Mengonfigurasi setelan komunikasi grup keamanan untuk klaster Amazon Redshift atau grup kerja Amazon Redshift Tanpa Server](#).

Enkripsi secara default

Setelah 10 Januari 2025, Amazon Redshift akan lebih meningkatkan keamanan data dan cluster dengan mengaktifkan enkripsi sebagai pengaturan default untuk semua cluster yang disediakan Amazon Redshift yang baru dibuat. Ini tidak berlaku untuk cluster yang dipulihkan dari snapshot.

Dengan perubahan ini, kemampuan untuk mendekripsi cluster tidak akan lagi tersedia saat menggunakan Konsol Manajemen AWS, AWS CLI, atau API untuk membuat kluster yang disediakan tanpa menentukan kunci KMS. Cluster akan secara otomatis dienkripsi dengan file. Kunci milik AWS

Pembaruan ini dapat memengaruhi Anda jika Anda membuat kluster yang tidak terenkripsi menggunakan skrip otomatis atau memanfaatkan berbagi data dengan kluster yang tidak terenkripsi. Untuk memastikan transisi yang mulus, perbarui skrip Anda yang membuat cluster yang tidak terenkripsi. Selain itu, jika Anda secara teratur membuat kluster konsumen baru yang tidak terenkripsi dan menggunakannya untuk berbagi data, tinjau konfigurasi Anda untuk memastikan produsen dan kluster konsumen dienkripsi, mencegah gangguan pada aktivitas berbagi data Anda. Untuk informasi selengkapnya, lihat [Enkripsi basis data Amazon Redshift](#).

Menegakkan koneksi SSL

Setelah 10 Januari 2025, Amazon Redshift akan menerapkan koneksi SSL secara default untuk klien yang terhubung ke cluster yang disediakan dan dipulihkan yang baru dibuat. Perubahan default ini juga akan berlaku untuk grup kerja tanpa server.

Dengan perubahan ini, grup parameter default baru bernama `default.redshift-2.0` akan diperkenalkan untuk semua cluster yang baru dibuat atau dipulihkan, dengan `require_ssl` parameter diatur ke secara `true` default. Setiap cluster baru yang dibuat tanpa grup parameter tertentu akan secara otomatis menggunakan grup `default.redshift-2.0` parameter. Saat membuat cluster melalui konsol Amazon Redshift, grup `default.redshift-2.0` parameter baru akan dipilih secara otomatis. Perubahan ini juga akan tercermin dalam operasi `CreateCluster` dan `RestoreFromClusterSnapshot` API, dan SDK dan AWS CLI perintah yang sesuai. Jika Anda menggunakan grup parameter yang ada atau kustom, Amazon Redshift akan terus menghormati `require_ssl` nilai yang ditentukan dalam grup parameter Anda. Anda terus memiliki opsi untuk mengubah `require_ssl` nilai dalam grup parameter kustom Anda sesuai kebutuhan.

Untuk pengguna Amazon Redshift Tanpa Server, nilai default di `require_ssl` dalam `config-parameters` akan diubah menjadi `true`. Setiap permintaan untuk membuat grup kerja baru dengan `require_ssl` set to `false` akan ditolak. Anda dapat mengubah `require_ssl` nilainya `false`

setelah workgroup dibuat. Untuk informasi selengkapnya, lihat [Mengkonfigurasi opsi keamanan untuk koneksi](#).

Perhatikan bahwa Anda masih memiliki kemampuan untuk memodifikasi pengaturan cluster atau workgroup untuk mengubah perilaku default, jika diperlukan untuk kasus penggunaan spesifik Anda.

Contoh kode untuk Amazon Redshift menggunakan AWS SDKs

Contoh kode berikut menunjukkan cara menggunakan Amazon Redshift dengan AWS perangkat pengembangan perangkat lunak (SDK).

Dasar-dasar adalah contoh kode yang menunjukkan kepada Anda bagaimana melakukan operasi penting dalam suatu layanan.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Skenario adalah contoh kode yang menunjukkan kepada Anda bagaimana menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan atau dikombinasikan dengan yang lain Layanan AWS.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan layanan ini dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Contoh kode

- [Contoh dasar untuk Amazon Redshift menggunakan AWS SDKs](#)
 - [Halo Amazon Redshift](#)
 - [Pelajari dasar-dasar Amazon Redshift dengan SDK AWS](#)
 - [Tindakan untuk Amazon Redshift menggunakan AWS SDKs](#)
 - [Gunakan CreateCluster dengan AWS SDK atau CLI](#)
 - [Gunakan DeleteCluster dengan AWS SDK atau CLI](#)
 - [Gunakan DescribeClusters dengan AWS SDK atau CLI](#)
 - [Gunakan DescribeStatement dengan AWS SDK](#)
 - [Gunakan ExecuteStatement dengan AWS SDK](#)
 - [Gunakan GetStatementResult dengan AWS SDK](#)
 - [Gunakan ListDatabases dengan AWS SDK](#)
 - [Gunakan ModifyCluster dengan AWS SDK atau CLI](#)
- [Skenario untuk Amazon Redshift menggunakan AWS SDKs](#)

- [Buat pelacak item Amazon Redshift](#)

Contoh dasar untuk Amazon Redshift menggunakan AWS SDKs

Contoh kode berikut menunjukkan cara menggunakan dasar-dasar Amazon Redshift dengan AWS SDKs

Contoh

- [Halo Amazon Redshift](#)
- [Pelajari dasar-dasar Amazon Redshift dengan SDK AWS](#)
- [Tindakan untuk Amazon Redshift menggunakan AWS SDKs](#)
 - [Gunakan CreateCluster dengan AWS SDK atau CLI](#)
 - [Gunakan DeleteCluster dengan AWS SDK atau CLI](#)
 - [Gunakan DescribeClusters dengan AWS SDK atau CLI](#)
 - [Gunakan DescribeStatement dengan AWS SDK](#)
 - [Gunakan ExecuteStatement dengan AWS SDK](#)
 - [Gunakan GetStatementResult dengan AWS SDK](#)
 - [Gunakan ListDatabases dengan AWS SDK](#)
 - [Gunakan ModifyCluster dengan AWS SDK atau CLI](#)

Halo Amazon Redshift

Contoh kode berikut menunjukkan cara memulai menggunakan Amazon Redshift.

.NET

SDK for .NET (v4)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
/// <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}");
    }
}
```

- Untuk detail API, lihat [DescribeClusters](#) di Referensi AWS SDK for .NET API.

Go

SDK untuk Go V2

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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\t%v : %v\n", *cluster.ClusterIdentifier, *cluster.ClusterStatus)
  }
}

```

- Untuk detail API, lihat [DescribeClusters](#) di Referensi AWS SDK for Go API.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

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()));
    }
}

```

- Untuk detail API, lihat [DescribeClusters](#) di Referensi AWS SDK for Java 2.x API.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

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"))
```

- Untuk detail API, lihat [DescribeClusters](#) di AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan layanan ini dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Pelajari dasar-dasar Amazon Redshift dengan SDK AWS

Contoh-contoh kode berikut menunjukkan cara:

- Buat cluster Redshift.
- Daftar database di cluster.
- Buat tabel bernama Movies.
- Isi tabel Film.
- Kueri tabel Film berdasarkan tahun.
- Ubah cluster Redshift.
- Hapus cluster Amazon Redshift.

.NET

SDK for .NET (v4)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Buat kelas pembungkus Redshift untuk mengelola operasi.

```
/// <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}");
}
}

```

Jalankan skenario interaktif yang menunjukkan dasar-dasar Redshift.

```

/// <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("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; }
}
}

```

- Untuk detail API, lihat topik berikut di Referensi API AWS SDK for .NET .
 - [CreateCluster](#)
 - [DescribeClusters](#)
 - [DescribeStatement](#)
 - [ExecuteStatement](#)
 - [GetStatementResult](#)
 - [ListDatabasesPaginator](#)
 - [ModifyCluster](#)

Go

SDK untuk Go V2

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkap dan pelajari cara menyiapkan dan menjalankan di [Repositori Contoh Kode AWS](#).

```
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)
        }
    }
}
```

```
}  
}  
}  
}
```

- Untuk detail API, lihat topik berikut di Referensi API AWS SDK for Go .
 - [CreateCluster](#)
 - [DescribeClusters](#)
 - [DescribeStatement](#)
 - [ExecuteStatement](#)
 - [GetStatementResult](#)
 - [ListDatabasesPaginator](#)
 - [ModifyCluster](#)

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Jalankan skenario interaktif yang mendemonstrasikan fitur Amazon Redshift.

```
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();
}
}

```

Kelas pembungkus untuk metode Amazon Redshift SDK.

```

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());
        }
    });
}
```

- Untuk detail API, lihat topik berikut di Referensi API AWS SDK for Java 2.x .

- [CreateCluster](#)
- [DescribeClusters](#)
- [DescribeStatement](#)
- [ExecuteStatement](#)
- [GetStatementResult](#)
- [ListDatabasesPaginator](#)
- [ModifyCluster](#)

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).


```

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

```

Fungsi utama yang menunjukkan implementasi skenario.

```

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"
)

```

Fungsi pembungkus yang digunakan dalam skenario.

```

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
```

- Untuk detail API, lihat topik berikut di Referensi API AWS SDK untuk Python (Boto3).
 - [CreateCluster](#)
 - [DescribeClusters](#)
 - [DescribeStatement](#)
 - [ExecuteStatement](#)
 - [GetStatementResult](#)
 - [ListDatabasesPaginator](#)
 - [ModifyCluster](#)

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan layanan ini dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Tindakan untuk Amazon Redshift menggunakan AWS SDKs

Contoh kode berikut menunjukkan cara melakukan tindakan Amazon Redshift individual dengan AWS SDKs. Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan instruksi untuk mengatur dan menjalankan kode.

Kutipan ini menyebut Amazon Redshift API dan merupakan kutipan kode dari program yang lebih besar yang harus dijalankan dalam konteks. Anda dapat melihat tindakan dalam konteks di [Skenario untuk Amazon Redshift menggunakan AWS SDKs](#).

Contoh berikut hanya mencakup tindakan yang paling umum digunakan. Untuk daftar lengkapnya, lihat Referensi [API Amazon Redshift](#).

Contoh

- [Gunakan CreateCluster dengan AWS SDK atau CLI](#)
- [Gunakan DeleteCluster dengan AWS SDK atau CLI](#)
- [Gunakan DescribeClusters dengan AWS SDK atau CLI](#)
- [Gunakan DescribeStatement dengan AWS SDK](#)
- [Gunakan ExecuteStatement dengan AWS SDK](#)
- [Gunakan GetStatementResult dengan AWS SDK](#)
- [Gunakan ListDatabases dengan AWS SDK](#)
- [Gunakan ModifyCluster dengan AWS SDK atau CLI](#)

Gunakan **CreateCluster** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateCluster`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Pelajari dasar-dasarnya](#)

.NET

SDK for .NET (v4)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
/// <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;
    }
}
```

- Untuk detail API, lihat [CreateCluster](#) di Referensi AWS SDK for .NET API.

CLI

AWS CLI

Buat Cluster dengan ParametersThis contoh Minimal membuat cluster dengan set minimal parameter. Secara default, output dalam format JSON. Command:

```
aws redshift create-cluster --node-type dw.hs1.xlarge --number-of-nodes 2 --
master-username adminuser --master-user-password TopSecret1 --cluster-identifier
mycluster
```

Hasil:

```
{
  "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"
```



```
}  
}
```

- Untuk detail API, lihat [CreateCluster](#) di Referensi AWS CLI Perintah.

Go

SDK untuk Go V2

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
}

```

- Untuk detail API, lihat [CreateCluster](#) di Referensi AWS SDK for Go API.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Buat cluster.

```

/**
 * 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);
                }
            });
    }
}

```

- Untuk detail API, lihat [CreateCluster](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Buat klien.

```
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 };
```

Buat cluster.

```
// 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();
```

- Untuk detail API, lihat [CreateCluster](#) di Referensi AWS SDK for JavaScript API.

Kotlin

SDK untuk Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Buat cluster.

```
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}")  
    }  
}
```

- Untuk detail API, lihat [CreateCluster](#) di AWS SDK untuk referensi API Kotlin.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
```

Kode berikut membuat instance objek. RedshiftWrapper

```
client = boto3.client("redshift")
redshift_wrapper = RedshiftWrapper(client)
```

- Untuk detail API, lihat [CreateCluster](#) di AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK for SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Buat cluster.

```

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.

```

- Untuk detail API, lihat [CreateCluster](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan layanan ini dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DeleteCluster** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DeleteCluster`.

.NET

SDK for .NET (v4)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).


```

/// <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;
    }
}

```

- Untuk detail API, lihat [DeleteCluster](#) di Referensi AWS SDK for .NET API.

CLI

AWS CLI

Hapus Cluster tanpa Snapshot This contoh Final Cluster menghapus cluster, memaksa penghapusan data sehingga tidak ada snapshot cluster akhir yang dibuat. Command:

```
aws redshift delete-cluster --cluster-identifier mycluster --skip-final-cluster-snapshot
```

Menghapus Cluster, Mengizinkan SnapshotThis contoh Final Cluster menghapus klaster, tetapi menentukan snapshot.Command cluster akhir:

```
aws redshift delete-cluster --cluster-identifier mycluster --final-cluster-snapshot-identifier myfinalsnapshot
```

- Untuk detail API, lihat [DeleteCluster](#) di Referensi AWS CLI Perintah.

Go

SDK untuk Go V2

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
}
```

- Untuk detail API, lihat [DeleteCluster](#) di Referensi AWS SDK for Go API.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Hapus klaster .

```

/**
 * 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());
            }
        });
}

```

- Untuk detail API, lihat [DeleteCluster](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Buat klien.

```
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 };
```

Buat cluster.

```
// 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();
```

- Untuk detail API, lihat [DeleteCluster](#) di Referensi AWS SDK for JavaScript API.

Kotlin

SDK untuk Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Hapus klaster .

```
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}")
    }
}
```

- Untuk detail API, lihat [DeleteCluster](#) di AWS SDK untuk referensi API Kotlin.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
```

Kode berikut membuat instance objek. RedshiftWrapper

```
client = boto3.client("redshift")
redshift_wrapper = RedshiftWrapper(client)
```

- Untuk detail API, lihat [DeleteCluster](#) di AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK for SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Hapus kluster .

```
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.
```

- Untuk detail API, lihat [DeleteCluster](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan layanan ini dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeClusters** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeClusters`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Pelajari dasar-dasarnya](#)

.NET

SDK for .NET (v4)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
/// <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;
}
}
}

```

- Untuk detail API, lihat [DescribeClusters](#) di Referensi AWS SDK for .NET API.

CLI

AWS CLI

Dapatkan Deskripsi Semua Clusters This contoh mengembalikan deskripsi semua cluster untuk akun. Secara default, output dalam format JSON. Command:

```
aws redshift describe-clusters
```

Hasil:

```
{
```

```

"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"
}
}

```

Anda juga dapat memperoleh informasi yang sama dalam format teks menggunakan `--output text` option.Command:

`--output textOption.command:`

`Option.command:`

```
aws redshift describe-clusters --output text
```

Hasil:

```
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
```

- Untuk detail API, lihat [DescribeClusters](#) di Referensi AWS CLI Perintah.

Go

SDK untuk Go V2

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
}

```

- Untuk detail API, lihat [DescribeClusters](#) di Referensi AWS SDK for Go API.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Jelaskan cluster.

```

/**
 * 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);
    });
}

```

- Untuk detail API, lihat [DescribeClusters](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Buat klien.

```

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 };

```

Jelaskan cluster Anda.

```
// 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();
```

- Untuk detail API, lihat [DescribeClusters](#) di Referensi AWS SDK for JavaScript API.

Kotlin

SDK untuk Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Jelaskan cluster.

```
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}")
            }
        }
    }
}

```

- Untuk detail API, lihat [DescribeClusters](#) di AWS SDK untuk referensi API Kotlin.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

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

```

Kode berikut membuat instance objek. RedshiftWrapper

```

client = boto3.client("redshift")
redshift_wrapper = RedshiftWrapper(client)

```

- Untuk detail API, lihat [DescribeClusters](#) di AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK for SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Jelaskan cluster.

```

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.

```

- Untuk detail API, lihat [DescribeClusters](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan layanan ini dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeStatement** dengan AWS SDK

Contoh kode berikut menunjukkan cara menggunakan `DescribeStatement`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Pelajari dasar-dasarnya](#)

.NET

SDK for .NET (v4)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
/// <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;
    }
}
```

- Untuk detail API, lihat [DescribeStatement](#) di Referensi AWS SDK for .NET API.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

/**
 * 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!");
            }
        })
}

```

```
    });
}
```

- Untuk detail API, lihat [DescribeStatement](#) di Referensi AWS SDK for Java 2.x API.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
```

Kode berikut membuat instance objek. RedshiftDataWrapper

```
client = boto3.client("redshift-data")
redshift_data_wrapper = RedshiftDataWrapper(client)
```

- Untuk detail API, lihat [DescribeStatement](#) di AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK for SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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.
```

- Untuk detail API, lihat [DescribeStatement](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan layanan ini dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ExecuteStatement** dengan AWS SDK

Contoh kode berikut menunjukkan cara menggunakan `ExecuteStatement`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Pelajari dasar-dasarnya](#)

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Mengeksekusi pernyataan SQL untuk membuat tabel database.

```
/**
 * 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");
            }
        });
}

```

Mengeksekusi pernyataan SQL untuk memasukkan data ke dalam tabel database.

```

/**
 * 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;
});
}

```

Mengeksekusi pernyataan SQL untuk query tabel database.

```

/**
 * 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 "";
    });
}

```

- Untuk detail API, lihat [ExecuteStatement](#) di Referensi AWS SDK for Java 2.x API.

SAP ABAP

SDK for SAP ABAP

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

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.
```

- Untuk detail API, lihat [ExecuteStatement](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan layanan ini dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **GetStatementResult** dengan AWS SDK

Contoh kode berikut menunjukkan cara menggunakan `GetStatementResult`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Pelajari dasar-dasarnya](#)

.NET

SDK for .NET (v4)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
/// <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;
}
}

```

- Untuk detail API, lihat [GetStatementResult](#) di Referensi AWS SDK for .NET API.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Periksa hasil pernyataan.

```

/**
 * 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
            });
    }

```

- Untuk detail API, lihat [GetStatementResult](#) di Referensi AWS SDK for Java 2.x API.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
```

Kode berikut membuat instance objek. `RedshiftDataWrapper`

```
client = boto3.client("redshift-data")
redshift_data_wrapper = RedshiftDataWrapper(client)
```

- Untuk detail API, lihat [GetStatementResult](#) di AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK for SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Periksa hasil pernyataan.

```
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.

```

- Untuk detail API, lihat [GetStatementResult](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan layanan ini dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ListDatabases** dengan AWS SDK

Contoh kode berikut menunjukkan cara menggunakan `ListDatabases`.

.NET

SDK for .NET (v4)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

/// <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;
        }
    }
}

```

- Untuk detail API, lihat [ListDatabases](#) di Referensi AWS SDK for .NET API.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
/**
 * 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);
    });
}
```

```
}
```

- Untuk detail API, lihat [ListDatabases](#) di Referensi AWS SDK for Java 2.x API.

SAP ABAP

SDK for SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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.
```

- Untuk detail API, lihat [ListDatabases](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan layanan ini dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ModifyCluster** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ModifyCluster`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Pelajari dasar-dasarnya](#)

.NET

SDK for .NET (v4)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
/// <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;  
    }  
}
```

- Untuk detail API, lihat [ModifyCluster](#) di Referensi AWS SDK for .NET API.

CLI

AWS CLI

Mengaitkan Grup Keamanan dengan ClusterThis contoh menunjukkan cara mengaitkan grup keamanan klaster dengan Cluster.command yang ditentukan:

```
aws redshift modify-cluster --cluster-identifier mycluster --cluster-security-groups mysecuritygroup
```

Ubah Jendela Pemeliharaan untuk ClusterThis menunjukkan cara mengubah jendela pemeliharaan pilihan mingguan untuk cluster menjadi jendela minimum empat jam mulai hari Minggu pukul 11:15, dan berakhir Senin pukul 3:15 pagi. Perintah:

```
aws redshift modify-cluster --cluster-identifier mycluster --preferred-maintenance-window Sun:23:15-Mon:03:15
```

Ubah Kata Sandi Master untuk ClusterThis contoh menunjukkan cara mengubah kata sandi master untuk cluster.Command:

```
aws redshift modify-cluster --cluster-identifier mycluster --master-user-password A1b2c3d4
```


- Untuk detail API, lihat [ModifyCluster](#) di Referensi AWS CLI Perintah.

Go

SDK untuk Go V2

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
}

```

- Untuk detail API, lihat [ModifyCluster](#) di Referensi AWS SDK for Go API.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Memodifikasi cluster.

```

/**
 * 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");
        }
    });
}

```

- Untuk detail API, lihat [ModifyCluster](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Buat klien.

```

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 };
```

Memodifikasi cluster.

```
// 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();
```

- Untuk detail API, lihat [ModifyCluster](#) di Referensi AWS SDK for JavaScript API.

Kotlin

SDK untuk Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Memodifikasi cluster.

```
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",
        )
    }
}
```

- Untuk detail API, lihat [ModifyCluster](#) di AWS SDK untuk referensi API Kotlin.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
```

Kode berikut membuat instance objek. `RedshiftWrapper`

```
client = boto3.client("redshift")
redshift_wrapper = RedshiftWrapper(client)
```

- Untuk detail API, lihat [ModifyCluster](#) di AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK for SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Memodifikasi cluster.

```

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.

```

- Untuk detail API, lihat [ModifyCluster](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan layanan ini dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Skenario untuk Amazon Redshift menggunakan AWS SDKs

Contoh kode berikut menunjukkan cara menerapkan skenario umum di Amazon Redshift dengan AWS SDKs. Skenario ini menunjukkan kepada Anda cara menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam Amazon Redshift atau digabungkan dengan yang lain. Layanan AWS Setiap skenario menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode.

Skenario menargetkan tingkat pengalaman menengah untuk membantu Anda memahami tindakan layanan dalam konteks.

Contoh

- [Buat pelacak item Amazon Redshift](#)

Buat pelacak item Amazon Redshift

Contoh kode berikut menunjukkan cara membuat aplikasi web yang melacak dan melaporkan item pekerjaan menggunakan database Amazon Redshift.

Java

SDK untuk Java 2.x

Menunjukkan cara membuat aplikasi web yang melacak dan melaporkan item pekerjaan yang disimpan dalam database Amazon Redshift.

Untuk kode sumber lengkap dan petunjuk tentang cara menyiapkan Spring REST API yang menanyakan data Amazon Redshift dan untuk digunakan oleh aplikasi React, lihat contoh lengkapnya di. [GitHub](#)

Layanan yang digunakan dalam contoh ini

- Amazon Redshift
- Amazon SES

Kotlin

SDK untuk Kotlin

Menunjukkan cara membuat aplikasi web yang melacak dan melaporkan item pekerjaan yang disimpan dalam database Amazon Redshift.

Untuk kode sumber lengkap dan petunjuk tentang cara menyiapkan Spring REST API yang menanyakan data Amazon Redshift dan untuk digunakan oleh aplikasi React, lihat contoh lengkapnya di. [GitHub](#)

Layanan yang digunakan dalam contoh ini

- Amazon Redshift
- Amazon SES

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan layanan ini dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Riwayat dokumen

Note

Untuk deskripsi fitur baru di Amazon Redshift, lihat [Apa yang baru](#).

Tabel berikut menjelaskan perubahan dokumentasi penting pada Panduan Manajemen Pergeseran Merah Amazon setelah Juni 2018. Untuk notifikasi tentang pembaruan dokumentasi ini, Anda dapat berlangganan ke umpan RSS.

Versi API: 2012-12-01

Untuk daftar perubahan pada Panduan Pengembang Database Amazon Redshift, lihat riwayat dokumen Panduan Pengembang Database [Amazon Redshift](#).

Perubahan	Deskripsi	Tanggal
Patch Amazon Redshift 199 dirilis.	Patch Amazon Redshift baru sedang digunakan. Dibutuhkan beberapa minggu agar versi baru tersedia di semua Amazon Redshift yang didukung. Wilayah AWS Untuk informasi selengkapnya tentang versi ini, lihat Amazon Redshift patch 199.	Maret 4, 2026
Patch Amazon Redshift 198 dirilis.	Patch Amazon Redshift baru sedang digunakan. Dibutuhkan beberapa minggu agar versi baru tersedia di semua Amazon Redshift yang didukung. Wilayah AWS Untuk informasi selengkapnya tentang versi ini, lihat Amazon Redshift patch 198.	Februari 3, 2026

Perbarui kebijakan terkelola Amazon Redshift	Menambahkan kebijakan AmazonRedshiftFederatedAuthorization terkelola baru.	November 21, 2025
Patch Amazon Redshift 197 dirilis.	Patch Amazon Redshift baru sedang digunakan. Dibutuhkan beberapa minggu agar versi baru tersedia di semua Amazon Redshift yang didukung. Wilayah AWS Untuk informasi selengkapnya tentang versi ini, lihat Amazon Redshift patch 197.	November 20, 2025
Patch Amazon Redshift 196 dirilis.	Patch Amazon Redshift baru sedang digunakan. Dibutuhkan beberapa minggu agar versi baru tersedia di semua Amazon Redshift yang didukung. Wilayah AWS Untuk informasi selengkapnya tentang versi ini, lihat Amazon Redshift patch 196.	November 10, 2025
Patch Amazon Redshift 195 dirilis.	Patch Amazon Redshift baru sedang digunakan. Dibutuhkan beberapa minggu agar versi baru tersedia di semua Amazon Redshift yang didukung. Wilayah AWS Untuk informasi selengkapnya tentang versi ini, lihat Amazon Redshift patch 195.	Oktober 24, 2025

[Patch Amazon Redshift 194
dirilis.](#)

Patch Amazon Redshift baru sedang digunakan. Dibutuhkan beberapa minggu agar versi baru tersedia di semua Amazon Redshift yang didukung. Wilayah AWS Untuk informasi selengkapnya tentang versi ini, lihat [Amazon Redshift patch](#) 194.

Oktober 1, 2025

[Patch Amazon Redshift 193
dirilis.](#)

Patch Amazon Redshift baru sedang digunakan. Dibutuhkan beberapa minggu agar versi baru tersedia di semua Amazon Redshift yang didukung. Wilayah AWS Untuk informasi selengkapnya tentang versi ini, lihat [Amazon Redshift patch](#) 193.

Agustus 26, 2025

[Amazon Redshift patch 192
dirilis.](#)

Patch Amazon Redshift baru sedang digunakan. Dibutuhkan beberapa minggu agar versi baru tersedia di semua Amazon Redshift yang didukung. Wilayah AWS Untuk informasi selengkapnya tentang versi ini, lihat [Amazon Redshift patch](#) 192.

Juli 30, 2025

[Patch Amazon Redshift 191
dirilis.](#)

Patch Amazon Redshift baru sedang digunakan. Dibutuhkan beberapa minggu agar versi baru tersedia di semua Amazon Redshift yang didukung. Wilayah AWS Untuk informasi selengkapnya tentang versi ini, lihat [Amazon Redshift patch 191](#).

Juni 24, 2025

[Amazon Redshift patch 190
dirilis.](#)

Patch Amazon Redshift baru sedang digunakan. Dibutuhkan beberapa minggu agar versi baru tersedia di semua Amazon Redshift yang didukung. Wilayah AWS Untuk informasi selengkapnya tentang versi ini, lihat [Amazon Redshift patch 190](#).

7 Mei 2025

[Perbarui kebijakan terkelola
Amazon Redshift](#)

Pembaruan kebijakan AmazonRedshiftServiceLinkedRolePolicy terkelola dengan izinlakeformation:GetDataAccess .

Maret 13, 2025

[Patch Amazon Redshift 189
dirilis.](#)

Patch Amazon Redshift baru sedang digunakan. Dibutuhkan beberapa minggu agar versi baru tersedia di semua Amazon Redshift yang didukung. Wilayah AWS Untuk informasi selengkapnya tentang versi ini, lihat [Amazon Redshift patch 189](#).

7 Maret 2025

[Patch Amazon Redshift 188
dirilis.](#)

Patch Amazon Redshift baru sedang digunakan. Dibutuhkan beberapa minggu agar versi baru tersedia di semua Amazon Redshift yang didukung. Wilayah AWS Untuk informasi selengkapnya tentang versi ini, lihat [Amazon Redshift patch](#) 188.

Februari 5, 2025

[Patch Amazon Redshift 187
dirilis.](#)

Patch Amazon Redshift baru sedang digunakan. Dibutuhkan beberapa minggu agar versi baru tersedia di semua Amazon Redshift yang didukung. Wilayah AWS Untuk informasi selengkapnya tentang versi ini, lihat [Amazon Redshift patch](#) 187.

Desember 20, 2024

[Perbarui kebijakan terkelola
Amazon Redshift](#)

Pembaruan kebijakan AmazonRedshiftServiceLinkedRolePolicy terkelola dengan izin glue:GetCatalog dan glue:GetCatalogs .

Desember 3, 2024

[Patch Amazon Redshift 186
dirilis.](#)

Patch Amazon Redshift baru sedang digunakan. Dibutuhkan beberapa minggu agar versi baru tersedia di semua Amazon Redshift yang didukung. Wilayah AWS Untuk informasi selengkapnya tentang versi ini, lihat [Amazon Redshift patch](#) 186.

Oktober 22, 2024

[Patch Amazon Redshift 185
dirilis.](#)

Patch Amazon Redshift baru sedang digunakan. Dibutuhkan beberapa minggu agar versi baru tersedia di semua Amazon Redshift yang didukung. Wilayah AWS Untuk informasi selengkapnya tentang versi ini, lihat [Amazon Redshift patch](#) 185.

Oktober 9, 2024

[Patch Amazon Redshift 184
dirilis.](#)

Patch Amazon Redshift baru sedang digunakan. Dibutuhkan beberapa minggu agar versi baru tersedia di semua Amazon Redshift yang didukung. Wilayah AWS Untuk informasi selengkapnya tentang versi ini, lihat [Amazon Redshift patch](#) 184.

September 12, 2024

[Patch Amazon Redshift 183
dirilis.](#)

Patch Amazon Redshift baru sedang digunakan. Dibutuhkan beberapa minggu agar versi baru tersedia di semua Amazon Redshift yang didukung. Wilayah AWS Untuk informasi selengkapnya tentang versi ini, lihat [Amazon Redshift patch](#) 183.

Agustus 8, 2024

[Patch Amazon Redshift 182
dirilis.](#)

Patch Amazon Redshift baru sedang digunakan. Dibutuhkan beberapa minggu agar versi baru tersedia di semua Amazon Redshift yang didukung. Wilayah AWS Untuk informasi selengkapnya tentang versi ini, lihat [Amazon Redshift patch](#) 182.

Juni 26, 2024

[Patch Amazon Redshift 181
dirilis.](#)

Patch Amazon Redshift baru sedang digunakan. Dibutuhkan beberapa minggu agar versi baru tersedia di semua Amazon Redshift yang didukung. Wilayah AWS Untuk informasi selengkapnya tentang versi ini, lihat [Amazon Redshift patch](#) 181.

1 Mei 2024

[Perbarui kebijakan terkelola
Amazon Redshift](#)

Perbarui ke kebijakan AmazonRedshiftServiceLinkedRolePolicy terkelola dengan izin servicequotas:GetServiceQuota untuk mengakses AWS kuota atau batasan.

8 Maret 2024

[Perbarui kebijakan terkelola editor kueri v2](#)

Pembaruan untuk AmazonRedshiftQueryEditorV2 FullAccess, AmazonRedshiftQueryEditorV2 NoSharing, AmazonRedshiftQueryEditorV2 ReadSharing, dan kebijakan AmazonRedshiftQueryEditorV2 ReadWriteSharing terkelola dengan izin redshift-serverless::ListNamespaces dan redshift-serverless::ListWorkgroups.

Februari 21, 2024

[Perbarui kebijakan terkelola akses hanya-baca Amazon Redshift](#)

Pembaruan kebijakan AmazonRedshiftReadOnlyAccess terkelola dengan izin redshift::ListRecommendations untuk mencantumkan rekomendasi Amazon Redshift Advisor.

Februari 7, 2024

[Amazon Redshift patch 180 dirilis.](#)

Patch Amazon Redshift baru sedang digunakan. Dibutuhkan beberapa minggu agar versi baru tersedia di semua Amazon Redshift yang didukung. Wilayah AWS Untuk informasi selengkapnya tentang versi ini, lihat [Amazon Redshift patch 180](#).

Desember 29, 2023

[Patch Amazon Redshift 179
dirilis.](#)

Patch Amazon Redshift baru sedang digunakan. Dibutuhkan beberapa minggu agar versi baru tersedia di semua Amazon Redshift yang didukung. Wilayah AWS Untuk informasi selengkapnya tentang versi ini, lihat [Amazon Redshift patch](#) 179.

9 November 2023

[Perbarui kebijakan terkelola
Amazon Redshift](#)

Pembaruan kebijakan AmazonRedshiftServiceLinkedRolePolicy terkelola dengan izin ec2:AssignIpv6Addresses dan ec2:UnassignIpv6Addresses .

31 Oktober 2023

[Patch Amazon Redshift 178
dirilis.](#)

Patch Amazon Redshift baru sedang digunakan. Dibutuhkan beberapa minggu agar versi baru tersedia di semua Amazon Redshift yang didukung. Wilayah AWS Untuk informasi selengkapnya tentang versi ini, lihat [Amazon Redshift patch](#) 178.

25 September 2023

[Perbarui kebijakan terkelola editor kueri v2](#)

Pembaruan Amazon Redshift Query Editor V2 NoSharing, Amazon Redshift Query Editor V2 ReadSharing, dan kebijakan Amazon Redshift Query Editor V2 ReadWriteSharing terkelola dengan izin sqlworkbench:GetAutocompletionMetadata dan sqlworkbench:GetAutocompletionResource.

16 Agustus 2023

[Perbarui kebijakan terkelola Amazon Redshift](#)

Pembaruan kebijakan Amazon Redshift ServiceLinkedRolePolicy terkelola untuk memberikan izin AWS Secrets Manager untuk membuat dan mengelola rahasia kredensi admin.

Agustus 14, 2023

[Amazon Redshift patch 177 dirilis.](#)

Patch Amazon Redshift baru sedang digunakan. Dibutuhkan beberapa minggu agar versi baru tersedia di semua Amazon Redshift yang didukung. Wilayah AWS Untuk informasi selengkapnya tentang versi ini, lihat [Amazon Redshift patch 177](#).

3 Agustus 2023

[Patch Amazon Redshift 176
dirilis.](#)

Patch Amazon Redshift baru sedang digunakan. Dibutuhkan beberapa minggu agar versi baru tersedia di semua Amazon Redshift yang didukung. Wilayah AWS Untuk informasi selengkapnya tentang versi ini, lihat [Amazon Redshift patch 176](#).

8 Juni 2023

[Amazon Redshift patch 175
dirilis.](#)

Patch Amazon Redshift baru sedang digunakan. Dibutuhkan beberapa minggu agar versi baru tersedia di semua Amazon Redshift yang didukung. Wilayah AWS Untuk informasi selengkapnya tentang versi ini, lihat [Amazon Redshift patch 175](#).

28 April 2023

[Perbarui kebijakan terkelola
Amazon Redshift](#)

Pembaruan kebijakan AmazonRedshiftServiceLinkedRolePolicy terkelola untuk menghapus izin untuk tindakan terkait jaringan ec2. Ini secara khusus terkait dengan tag Purpose: RedshiftMigrateToVpc resource.

27 April 2023

[Perbarui kebijakan terkelola
Amazon Redshift Data API](#)

Pembaruan kebijakan AmazonRedshiftDataFullAccess terkelola dengan izinredshift:GetClusterCredentialsWithIAM .

April 7, 2023

[Perbarui kebijakan terkelola editor kueri v2](#)

PembaruanAmazonRedshiftQueryEditorV2NoSharing ,AmazonRedshiftQueryEditorV2ReadSharing , dan kebijakan AmazonRedshiftQueryEditorV2ReadWriteSharing terkelola dengan izinsqlworkbench:GetSchemaInference .

21 Maret 2023

[Patch Amazon Redshift 174 dirilis.](#)

Patch Amazon Redshift baru sedang digunakan. Dibutuhkan beberapa minggu agar versi baru tersedia di semua Amazon Redshift yang didukung. Wilayah AWS Untuk informasi selengkapnya tentang versi ini, lihat [Amazon Redshift patch](#) 174.

Maret 11, 2023

[Perbarui kebijakan terkelola editor kueri v2](#)

PembaruanAmazonRedshiftQueryEditorV2NoSharing ,AmazonRedshiftQueryEditorV2ReadSharing , dan kebijakan AmazonRedshiftQueryEditorV2ReadWriteSharing terkelola dengan izinsqlworkbench:AssociateNotebookWithTab .

2 Februari 2023

[Patch Amazon Redshift 173
dirilis.](#)

Patch Amazon Redshift baru sedang digunakan. Dibutuhkan beberapa minggu agar versi baru tersedia di semua Amazon Redshift yang didukung. Wilayah AWS Untuk informasi selengkapnya tentang versi ini, lihat [Amazon Redshift patch](#) 173.

20 Januari 2023

[Amazon Redshift patch 172
dirilis.](#)

Patch Amazon Redshift baru sedang digunakan. Dibutuhkan beberapa minggu agar versi baru tersedia di semua Amazon Redshift yang didukung. Wilayah AWS Untuk informasi selengkapnya tentang versi ini, lihat [Amazon Redshift patch](#) 172.

17 November 2022

[Patch Amazon Redshift 171
dirilis.](#)

Patch Amazon Redshift baru sedang digunakan. Dibutuhkan beberapa minggu agar versi baru tersedia di semua Amazon Redshift yang didukung. Wilayah AWS Untuk informasi selengkapnya tentang versi ini, lihat [Amazon Redshift patch](#) 171.

9 November 2022

[Amazon Redshift patch 170
dirilis.](#)

Patch Amazon Redshift baru sedang digunakan. Dibutuhkan beberapa minggu agar versi baru tersedia di semua Amazon Redshift yang didukung. Wilayah AWS Untuk informasi selengkapnya tentang versi ini, lihat [Amazon Redshift patch 170](#).

20 Juli 2022

[Patch Amazon Redshift 169
dirilis.](#)

Patch Amazon Redshift baru sedang digunakan. Dibutuhkan beberapa minggu agar versi baru tersedia di semua Amazon Redshift yang didukung. Wilayah AWS Untuk informasi selengkapnya tentang versi ini, lihat [Amazon Redshift patch 169](#).

Juni 8, 2022

[Amazon Redshift patch 168
dirilis.](#)

Patch Amazon Redshift baru sedang digunakan. Dibutuhkan beberapa minggu agar versi baru tersedia di semua Amazon Redshift yang didukung. Wilayah AWS Untuk informasi selengkapnya tentang versi ini, lihat [Amazon Redshift patch 168](#).

19 April 2022

[Support untuk profil otentikasi
dengan driver Amazon
Redshift](#)

Anda sekarang dapat terhubung ke Amazon Redshift dengan profil otentikasi.

2 Agustus 2021

Dukungan untuk titik akhir lintas-VPC untuk Amazon Redshift didukung oleh AWS PrivateLink	Anda sekarang dapat menggunakan titik akhir VPC yang dikelola RedShift dengan Amazon Redshift.	1 April 2021
Dukungan untuk penyempurnaan editor kueri Amazon Redshift	Anda sekarang dapat menggunakan editor kueri dengan perutean VPC yang disempurnakan, waktu menjalankan kueri yang lebih lama, dan lebih banyak jenis node cluster.	17 Februari 2021
Support untuk integrasi konsol dengan mitra	Anda dapat berintegrasi dengan mitra menggunakan konsol Amazon Redshift.	9 Desember 2020
Support untuk kemampuan memindahkan cluster antar Availability Zones	Anda sekarang dapat memindahkan RA3 cluster antara Availability Zones.	9 Desember 2020
Support untuk tipe node ra3.xlplus	Anda sekarang dapat membuat tipe node ra3.xlplus.	9 Desember 2020
Support untuk driver JDBC versi 2.0	Anda sekarang dapat mengkonfigurasi driver JDBC versi 2.0.	5 November 2020
Support untuk Lambda UDFs dan tokenisasi	Anda sekarang dapat menulis Lambda UDFs untuk mengaktifkan tokenisasi data eksternal.	26 Oktober 2020
Support untuk menjadwalkan menjalankan pernyataan SQL	Anda sekarang dapat menjadwalkan kueri di konsol Amazon Redshift.	22 Oktober 2020

Support untuk API Data untuk Amazon Redshift	Amazon Redshift sekarang dapat diakses menggunakan API Data bawaan. Pembaruan dokumentasi mencakup Referensi API Data Amazon Redshift.	10 September 2020
Dukungan untuk pemantauan kueri konsol Amazon Redshift	Memperbarui panduan untuk menjelaskan grafik pemantauan kueri baru.	7 Mei 2020
Support untuk batas penggunaan	Memperbarui panduan untuk menjelaskan batas penggunaan.	23 April 2020
Otentikasi multi-faktor	Memperbarui panduan untuk menjelaskan dukungan otentikasi multi-faktor.	20 April 2020
Ubah ukuran elastis sekarang mendukung perubahan tipe simpul	Deskripsi pengubahan ukuran elastis yang diperbarui.	6 April 2020
Support untuk tipe node ra3.4xlarge dengan penyimpanan terkelola	Memperbarui panduan untuk menyertakan tipe node ra3.4xlarge.	2 April 2020
Support untuk jeda dan lanjutkan	Memperbarui panduan untuk menjelaskan jeda dan melanjutkan operasi cluster.	11 Maret 2020
Support untuk Microsoft Azure AD sebagai penyedia identitas	Memperbarui panduan untuk menjelaskan langkah-langkah untuk menggunakan Microsoft Azure AD sebagai penyedia identitas.	10 Februari 2020

Support untuk tipe RA3 node	Memperbarui panduan untuk menggambarkan jenis RA3 node baru.	3 Desember 2019
Support untuk konsol baru	Memperbarui panduan untuk menggambarkan konsol Amazon Redshift baru.	11 November 2019
Pembaruan informasi keamanan	Pembaruan dokumentasi informasi keamanan.	24 Juni 2019
Penyempurnaan snapshot	Amazon Redshift sekarang mendukung beberapa penyempurnaan untuk mengelola dan menjadwalkan snapshot.	4 April 2019
Penskalaan konkurensi	Anda dapat mengonfigurasi manajemen beban kerja (WLM) untuk mengaktifkan mode penskalaan konkurensi. Untuk informasi selengkapnya, lihat Mengonfigurasi manajemen beban kerja .	21 Maret 2019
Driver JDBC dan ODBC yang diperbarui	Amazon Redshift sekarang mendukung versi baru driver JDBC dan ODBC. Untuk informasi selengkapnya, lihat Mengkonfigurasi koneksi JDBC .	4 Februari 2019

Pemeliharaan yang ditangguhkan

Jika Anda perlu menjadwalkan ulang jendela pemeliharaan cluster Anda, Anda memiliki opsi untuk menunda pemeliharaan hingga 14 hari. Jika kami perlu memperbarui perangkat keras atau melakukan pembaruan wajib lainnya selama periode penundaan Anda, kami memberi tahu Anda dan membuat perubahan yang diperlukan. Cluster Anda tidak tersedia selama pembaruan ini. Untuk informasi lebih lanjut, lihat [Menunda pemeliharaan](#).

20 November 2018

Pemberitahuan di muka

Amazon Redshift memberikan pemberitahuan sebelumnya untuk beberapa acara. Peristiwa ini memiliki kategori acara pending. Misalnya, kami mengirim pemberitahuan terlebih dahulu jika pembaruan perangkat keras diperlukan untuk salah satu node di cluster Anda. Anda dapat berlangganan acara yang tertunda sama seperti acara Amazon Redshift lainnya. Untuk informasi selengkapnya, lihat [Berlangganan notifikasi acara Amazon Redshift](#).

20 November 2018

Ubah ukuran elastis

Pengubahan ukuran elastis adalah metode tercepat untuk mengubah ukuran cluster. Elastic resize menambahkan atau menghapus node pada cluster yang ada, kemudian secara otomatis mendistribusikan kembali data ke node baru. Karena tidak membuat cluster baru, operasi pengubahan ukuran elastis selesai dengan cepat, biasanya dalam beberapa menit. Untuk informasi selengkapnya, lihat [Mengubah ukuran cluster](#).

15 November 2018

Batalkan operasi pengubahan ukuran

Anda sekarang dapat membatalkan operasi pengubahan ukuran saat sedang berlangsung. Untuk informasi selengkapnya, lihat [Mengubah ukuran ikhtisar operasi](#).

2 November 2018

[Ubah cluster untuk mengubah enkripsi](#)

Anda dapat memodifikasi kluster yang tidak terenkripsi untuk menggunakan enkripsi AWS Key Management Service (AWS KMS), baik menggunakan kunci AWS-managed atau kunci yang dikelola pelanggan. Saat Anda memodifikasi kluster untuk mengaktifkan enkripsi KMS, Amazon Redshift secara otomatis memigrasikan data Anda ke kluster terenkripsi baru. Anda juga dapat memigrasikan kluster yang tidak terenkripsi ke kluster terenkripsi dengan memodifikasi kluster.

16 Oktober 2018

[Spektrum Amazon Redshift mendukung peningkatan perutean VPC](#)

Anda sekarang dapat menggunakan Redshift Spectrum dengan peningkatan perutean VPC yang diaktifkan untuk cluster Anda. Anda mungkin perlu melakukan langkah-langkah konfigurasi tambahan. Untuk informasi selengkapnya, lihat [Menggunakan spektrum Amazon Redshift dengan perutean VPC](#) yang disempurnakan.

10 Oktober 2018

Editor kueri	Sekarang Anda dapat menjalankan kueri SQL dari Amazon Redshift Management Console.	4 Oktober 2018
Bagan rincian eksekusi beban kerja	Sekarang Anda bisa mendapatkan tampilan terperinci tentang kinerja beban kerja Anda dengan melihat bagan Workload Execution Breakdown di konsol. Untuk informasi selengkapnya, lihat Menganalisis kinerja beban kerja .	30 Juli 2018
Trek pemeliharaan	Anda sekarang dapat menentukan apakah cluster Anda akan selalu diperbarui ke versi terbaru Amazon Redshift atau ke versi sebelumnya dengan memilih trek pemeliharaan. Untuk informasi selengkapnya, lihat Memilih trek pemeliharaan klaster .	26 Juli 2018

Tabel berikut menjelaskan perubahan penting pada Panduan Manajemen Pergeseran Merah Amazon sebelum Juli 2018.

Ubah	Deskripsi	Tanggal rilis
CloudWatch Metrik baru	CloudWatch Metrik baru ditambahkan untuk memantau kinerja kueri. Untuk informasi selengkapnya, lihat Data kinerja di Amazon Redshift	17 Mei 2018
Enkripsi HSM	Amazon Redshift hanya mendukung manajemen kunci AWS CloudHSM modul keamanan perangkat keras	6 Maret 2018

Ubah	Deskripsi	Tanggal rilis
	(HSM). Untuk informasi selengkapnya, lihat Enkripsi basis data Amazon Redshift .	
Rantai Peran IAM	Jika peran IAM yang dilampirkan ke klaster Anda tidak memiliki akses ke sumber daya yang diperlukan, Anda dapat merantai peran lain, mungkin milik akun lain. Cluster Anda kemudian sementara mengasumsikan peran berantai untuk mengakses data. Anda juga dapat memberikan akses lintas akun dengan merantai peran. Setiap peran dalam rantai mengasumsikan peran berikutnya dalam rantai, sampai cluster mengambil peran di akhir rantai. Anda dapat merantai maksimal 10 peran. Untuk informasi selengkapnya, lihat Merantai peran IAM di Amazon Redshift .	Februari 23, 2018
Jenis DC2 node baru	Generasi baru tipe node compact compute (DC) menawarkan kinerja yang jauh lebih baik dengan harga yang sama. DC1 Untuk memanfaatkan peningkatan kinerja, Anda dapat memigrasikan DC1 cluster Anda ke tipe DC2 node yang lebih baru. Untuk informasi selengkapnya, lihat Cluster dan node di Amazon Redshift .	17 Oktober 2017
Sertifikat ACM	Amazon Redshift mengganti sertifikat SSL di klaster Anda dengan sertifikat yang dikeluarkan AWS Certificate Manager (ACM). ACM adalah otoritas sertifikat publik terpercaya (CA) yang dipercaya oleh sebagian besar sistem saat ini. Anda mungkin perlu memperbarui sertifikat CA root kepercayaan Anda saat ini untuk terus terhubung ke cluster Anda menggunakan SSL. Untuk informasi selengkapnya, lihat Transisi ke sertifikat ACM untuk koneksi SSL .	18 September 2017

Ubah	Deskripsi	Tanggal rilis
Peran terkait layanan	Peran terkait layanan adalah jenis peran IAM unik yang ditautkan langsung ke Amazon Redshift. Peran terkait layanan telah ditentukan sebelumnya oleh Amazon Redshift dan menyertakan semua izin yang diperlukan layanan untuk memanggil AWS layanan atas nama klaster Amazon Redshift Anda. Untuk informasi selengkapnya, lihat Menggunakan peran terkait layanan untuk Amazon Redshift .	18 September 2017
Autentikasi pengguna database IAM	Anda dapat mengonfigurasi sistem Anda untuk mengizinkan pengguna membuat kredensial pengguna dan masuk ke database berdasarkan kredensi IAM mereka. Anda juga dapat mengonfigurasi sistem Anda agar pengguna dapat masuk menggunakan sistem masuk tunggal gabungan melalui penyedia identitas yang sesuai dengan SAMP 2.0. Untuk informasi selengkapnya, lihat Menggunakan otentikasi IAM untuk menghasilkan kredensi pengguna basis data .	11 Agustus 2017
Pemulihan tingkat tabel mendukung peningkatan perutean VPC	Pemulihan tingkat tabel sekarang didukung pada cluster yang menggunakan Mengontrol lalu lintas jaringan dengan perutean VPC yang ditingkatkan . Untuk informasi selengkapnya, lihat Memulihkan tabel dari snapshot .	19 Juli 2017
Aturan pemantauan kueri	Menggunakan aturan pemantauan kueri WLM, Anda dapat menentukan batas kinerja berbasis metrik untuk antrian WLM dan menentukan tindakan apa yang harus diambil ketika kueri melampaui batas-batas tersebut—log, hop, atau abort. Anda menentukan aturan pemantauan kueri sebagai bagian dari konfigurasi manajemen beban kerja (WLM) Anda. Untuk informasi selengkapnya, lihat Manajemen beban kerja .	April 21, 2017

Ubah	Deskripsi	Tanggal rilis
Perutean VPC yang disempurnakan	Saat Anda menggunakan perutean VPC Amazon Redshift yang disempurnakan, Amazon Redshift memaksa semua lalu lintas COPY dan UNLOAD antara cluster dan repositori data Anda melalui Amazon VPC Anda. Untuk informasi selengkapnya, lihat Mengontrol lalu lintas jaringan dengan perutean VPC Redshift yang ditingkatkan .	15 September 2016
Bidang log koneksi baru	Log Log koneksi audit memiliki dua bidang baru untuk melacak koneksi SSL. Jika Anda secara rutin memuat log audit ke tabel Amazon Redshift, Anda perlu menambahkan kolom baru berikut ke tabel target: sslcompression dan sslexpansion.	Mei 5, 2016
Peran IAM untuk COPY dan UNLOAD	Sekarang Anda dapat menentukan satu atau beberapa peran AWS Identity and Access Management (IAM) yang dapat digunakan klaster Anda untuk otentikasi untuk mengakses layanan lain AWS . Peran IAM menyediakan alternatif yang lebih aman untuk memberikan otentikasi dengan perintah COPY, UNLOAD, atau CREATE LIBRARY. Untuk informasi selengkapnya, lihat Mengotorisasi Amazon Redshift untuk AWS mengakses layanan atas nama Anda dan Mengotorisasi operasi COPY, UNLOAD, CREATE EXTERNAL FUNCTION, dan CREATE EXTERNAL SCHEMA menggunakan peran IAM .	29 Maret 2016
Kembalikan dari tabel	Anda dapat memulihkan tabel dari snapshot cluster ke tabel baru di cluster aktif. Untuk informasi selengkapnya, lihat Memulihkan tabel dari snapshot .	10 Maret 2016

Ubah	Deskripsi	Tanggal rilis
Menggunakan Kondisi IAM dalam kebijakan	Anda selanjutnya dapat membatasi akses ke sumber daya dengan menggunakan elemen Kondisi dalam kebijakan IAM. Untuk informasi selengkapnya, lihat Menggunakan ketentuan kebijakan IAM untuk kontrol akses terperinci .	Desember 10, 2015
Ubah yang dapat diakses publik	Anda dapat memodifikasi klaster yang ada di VPC untuk mengubah apakah klaster tersebut dapat diakses publik. Untuk informasi selengkapnya, lihat Mengubah klaster .	November 20, 2015
Perbaikan dokumentasi	Menerbitkan berbagai perbaikan dokumentasi.	Agustus 28, 2015
Pembaruan dokumentasi	Panduan pemecahan masalah yang diperbarui tentang konfigurasi pengaturan jaringan untuk memastikan bahwa host dengan ukuran unit transmisi maksimum (MTU) yang berbeda dapat menentukan ukuran paket untuk koneksi. Untuk informasi selengkapnya, lihat Kueri tampaknya hang dan terkadang gagal mencapai cluster .	Agustus 25, 2015
Pembaruan dokumentasi	Merevisi seluruh bagian tentang kelompok parameter untuk organisasi dan kejelasan yang lebih baik. Untuk informasi selengkapnya, lihat Grup parameter Amazon Redshift .	17 Agustus 2015
Properti dinamis WLM	Parameter konfigurasi WLM sekarang mendukung penerapan beberapa properti secara dinamis. Properti lain tetap berubah statis dan mengharuskan cluster terkait di-boot ulang sehingga perubahan konfigurasi dapat diterapkan. Untuk informasi selengkapnya, lihat Sifat dinamis dan statis WLM dan Grup parameter Amazon Redshift .	3 Agustus 2015

Ubah	Deskripsi	Tanggal rilis
Salin kluster terenkripsi KMS ke Wilayah lain AWS	Menambahkan konten tentang mengonfigurasi hibah salinan snapshot untuk mengaktifkan penyalinan cluster AWS KMS-enkripsi ke Wilayah lain. AWS Untuk informasi selengkapnya, lihat Menyalin AWS KMS—snapshot terenkripsi ke yang lain AWS Region .	28 Juli 2015
Pembaruan dokumentasi	Memperbarui bagian enkripsi basis data untuk menjelaskan dengan lebih baik bagaimana Amazon Redshift menggunakan AWS KMS atau HSMs untuk mengelola kunci, dan bagaimana proses enkripsi bekerja dengan masing-masing opsi ini. Untuk informasi selengkapnya, lihat Enkripsi basis data Amazon Redshift .	28 Juli 2015
Tipe node baru	Amazon Redshift sekarang menawarkan tipe node baru,. DS2 Referensi dokumentasi yang diperbarui ke tipe node yang ada untuk menggunakan nama baru yang diperkenalkan dalam rilis ini. Juga merevisi bagian untuk menjelaskan kombinasi tipe node dengan lebih baik dan memperjelas batas kuota default. Untuk informasi selengkapnya, lihat Cluster dan node di Amazon Redshift .	Juni 9, 2015
Penawaran node cadangan	Menambahkan konten tentang penawaran node cadangan baru. Juga merevisi bagian untuk menjelaskan dan membandingkan penawaran yang tersedia dengan lebih baik, dan memberikan contoh untuk menunjukkan bagaimana harga node sesuai permintaan dan cadangan memengaruhi penagihan. Untuk informasi selengkapnya, lihat Simpul terpesan .	Juni 9, 2015
Perbaikan dokumentasi	Menerbitkan berbagai perbaikan dokumentasi.	30 April 2015

Ubah	Deskripsi	Tanggal rilis
Fitur baru	Rilis Amazon Redshift ini memperkenalkan driver ODBC dan JDBC baru yang dioptimalkan untuk digunakan dengan Amazon Redshift. Untuk informasi selengkapnya, lihat Menyambung ke gudang data Amazon Redshift menggunakan alat klien SQL .	26 Februari 2015
Fitur baru	Rilis Amazon Redshift ini memperkenalkan metrik kinerja kluster yang memungkinkan Anda melihat dan menganalisis detail eksekusi kueri. Untuk informasi selengkapnya, lihat Melihat kueri dan beban .	26 Februari 2015
Pembaruan dokumentasi	Menambahkan contoh kebijakan baru yang menunjukkan pemberian izin untuk tindakan AWS layanan umum dan sumber daya yang menjadi sandaran Amazon Redshift. Untuk informasi selengkapnya, lihat Contoh kebijakan yang dikelola pelanggan .	Januari 16, 2015
Pembaruan dokumentasi	Panduan terbaru tentang pengaturan unit transmisi maksimum (MTU) untuk menonaktifkan bingkai TCP/IP jumbo. Untuk informasi selengkapnya, lihat Gunakan EC2 untuk membuat cluster Anda dan Kueri tampaknya hang dan terkadang gagal mencapai cluster .	Januari 16, 2015
Pembaruan dokumentasi	Merevisi konten tentang <code>wlm_json_configuration</code> parameter, dan memberikan contoh sintaks untuk mengkonfigurasi parameter ini dengan menggunakan sistem operasi Linux, Mac OS X, dan Microsoft Windows. AWS CLI Untuk informasi selengkapnya, lihat Manajemen beban kerja .	Januari 13, 2015
Pembaruan dokumentasi	Menambahkan pemberitahuan dan deskripsi acara yang hilang. Untuk informasi selengkapnya, lihat Pemberitahuan acara kluster yang disediakan Amazon Redshift .	8 Januari 2015

Ubah	Deskripsi	Tanggal rilis
Pembaruan dokumentasi	Panduan terbaru tentang kebijakan IAM untuk tindakan dan sumber daya Amazon Redshift. Merevisi bagian untuk meningkatkan organisasi dan kejelasan . Untuk informasi selengkapnya, lihat Keamanan di Amazon Redshift .	November 21, 2014
Fitur baru	Rilis Amazon Redshift ini memperkenalkan kemampuan untuk mengenkripsi cluster menggunakan kunci enkripsi from (). AWS Key Management Service AWS KMS AWS KMS menggabungkan perangkat keras dan perangkat lunak yang aman dan sangat tersedia untuk menyediakan sistem manajemen kunci yang diskalakan untuk cloud. Untuk informasi selengkapnya tentang AWS KMS dan opsi enkripsi untuk Amazon Redshift, lihat Enkripsi basis data Amazon Redshift dan. Operasi klaster	12 November 2014
Fitur baru	Rilis Amazon Redshift ini memperkenalkan kemampuan untuk menandai sumber daya, seperti cluster dan snapshot. Tag memungkinkan Anda menyediakan metadata yang ditentukan pengguna untuk mengkategorikan laporan penagihan berdasarkan alokasi biaya, dan untuk membantu Anda mengidentifikasi sumber daya dengan lebih baik dalam sekejap. Untuk informasi selengkapnya, lihat Menandai sumber daya di Amazon Redshift .	November 4, 2014
Fitur baru	Meningkatkan batas node maksimum menjadi 128 node untuk ukuran node dw1.8xlarge dan dw2.8xlarge. Untuk informasi selengkapnya, lihat Cluster dan node di Amazon Redshift .	30 Oktober 2014

Ubah	Deskripsi	Tanggal rilis
Fitur baru	Menambahkan kemampuan untuk menghentikan kueri dan pemuatan dari konsol Amazon Redshift. Untuk informasi selengkapnya, lihat Melihat kueri dan beban dan Melihat metrik cluster selama operasi pemuatan .	28 Oktober 2014
Perbaikan dokumentasi	Menerbitkan berbagai perbaikan dokumentasi.	Oktober 17, 2014
Konten baru	Menambahkan konten tentang mematikan cluster dan menghapus cluster. Untuk informasi selengkapnya, lihat Mematikan dan menghapus cluster .	Agustus 14, 2014
Pembaruan dokumentasi	Mengklarifikasi perilaku setelah Izinkan Peningkatan Versi untuk klaster. Untuk informasi selengkapnya, lihat Cluster yang disediakan Amazon Redshift .	Agustus 14, 2014
Pembaruan dokumentasi	Prosedur, tangkapan layar, dan pengaturan topik yang direvisi tentang bekerja dengan cluster di konsol Amazon Redshift. Untuk informasi selengkapnya, lihat Operasi klaster .	Juli 11, 2014
Konten baru	Menambahkan tutorial baru tentang mengubah ukuran cluster Amazon Redshift, termasuk cara mengubah ukuran cluster sambil meminimalkan jumlah waktu cluster dalam mode hanya-baca. Untuk informasi selengkapnya, lihat Mengubah ukuran cluster .	27 Juni 2014
Fitur baru	Menambahkan kemampuan untuk mengganti nama cluster. Untuk informasi selengkapnya, lihat Mengganti nama cluster dan Mengubah klaster .	Juni 2, 2014
Fitur baru	Menambahkan opsi untuk memilih grup parameter dan grup keamanan yang berbeda saat Anda memulihkan cluster dari snapshot. Untuk informasi selengkapnya, lihat Memulihkan cluster dari snapshot .	12 Mei 2014

Ubah	Deskripsi	Tanggal rilis
Fitur baru	Menambahkan bagian baru untuk menjelaskan cara mengonfigurasi CloudWatch alarm Amazon default untuk memantau persentase ruang disk yang digunakan dalam cluster Amazon Redshift. Alarm ini adalah opsi baru dalam proses pembuatan cluster. Untuk informasi selengkapnya, lihat Alarm ruang disk default .	28 April 2014
Pembaruan dokumentasi	Informasi yang diklarifikasi tentang dukungan Elliptic curve Diffie—Hellman Exchange (ECDHE) di Amazon Redshift. Untuk informasi selengkapnya, lihat SSL .	April 22, 2014
Fitur baru	Menambahkan pernyataan tentang dukungan Amazon Redshift untuk protokol perjanjian kunci kurva Elliptic Diffie—Hellman (ECDH). Untuk informasi selengkapnya, lihat SSL .	18 April 2014
Pembaruan dokumentasi	Merevisi dan mengatur ulang topik di bagian ini Menyambung ke gudang data Amazon Redshift menggunakan alat klien SQL . Menambahkan informasi selengkapnya tentang koneksi JDBC dan ODBC, dan bagian pemecahan masalah baru untuk masalah koneksi.	15 April 2014
Pembaruan dokumentasi	Ditambahkan versi dalam contoh kebijakan IAM di seluruh panduan.	3 April 2014
Pembaruan dokumentasi	Menambahkan informasi tentang cara kerja penetapan harga saat Anda mengubah ukuran klaster. Untuk informasi selengkapnya, lihat Simpul terpesan .	2 April 2014

Ubah	Deskripsi	Tanggal rilis
Fitur baru	Menambahkan bagian tentang parameter baru, <code>max_cursor_result_set_size</code> , yang menetapkan ukuran set hasil maksimum, dalam megabyte, yang dapat disimpan per kursor individu. Nilai parameter ini juga mempengaruhi jumlah kursor yang aktif secara bersamaan untuk cluster. Untuk informasi selengkapnya, lihat Grup parameter Amazon Redshift .	Maret 28, 2014
Fitur baru	Menambahkan penjelasan tentang bidang Versi Cluster sekarang termasuk versi mesin cluster dan nomor revisi database. Untuk informasi selengkapnya, lihat Cluster yang disediakan Amazon Redshift .	Maret 21, 2014
Fitur baru	Memperbarui prosedur pengubahan ukuran untuk menampilkan informasi kemajuan pengubahan ukuran baru pada tab Status klaster. Untuk informasi selengkapnya, lihat Mengubah ukuran cluster .	Maret 21, 2014
Pembaruan dokumentasi	Diatur ulang dan diperbarui Apa itu Amazon Redshift? dan Ikhtisar cluster yang disediakan Amazon Redshift direvisi. Menerbitkan berbagai perbaikan dokumentasi.	21 Februari 2014
Fitur baru	Menambahkan jenis dan ukuran node baru untuk klaster Amazon Redshift, dan menulis ulang topik ikhtisar klaster terkait untuk organisasi dan kejelasan yang lebih baik berdasarkan umpan balik. Untuk informasi selengkapnya, lihat Cluster yang disediakan Amazon Redshift .	Januari 23, 2014

Ubah	Deskripsi	Tanggal rilis
Fitur baru	Menambahkan informasi tentang penggunaan alamat IP elastis (EIP) untuk cluster Amazon Redshift yang dapat diakses publik di cloud pribadi virtual. Untuk informasi selengkapnya tentang EIP di Amazon Redshift, Sumber daya Redshift dalam VPC lihat dan. Membuat klaster yang disediakan Redshift atau workgroup Amazon Redshift Serverless di VPC	Desember 20, 2013
Fitur baru	Menambahkan informasi tentang AWS CloudTrail log untuk Amazon Redshift. Untuk informasi selengkapnya tentang dukungan Amazon Redshift CloudTrail, lihat. Logging dengan CloudTrail	13 Desember 2013
Fitur baru	Menambahkan informasi tentang log aktivitas pengguna baru dan parameter <code>enable_user_activity_logging</code> database untuk fitur pencatatan audit database di Amazon Redshift. Untuk informasi selengkapnya tentang pencatatan audit database, lihat Pencatatan audit basis data . Untuk informasi selengkapnya tentang parameter basis data, lihat Grup parameter Amazon Redshift .	Desember 6, 2013
Fitur baru	Diperbarui untuk menjelaskan konfigurasi Amazon Redshift untuk secara otomatis menyalin snapshot otomatis dan manual ke Wilayah sekunder. AWS Untuk informasi selengkapnya tentang mengonfigurasi salinan snapshot lintas wilayah, lihat. Menyalin snapshot ke Wilayah lain AWS	14 November 2013
Fitur baru	Menambahkan bagian untuk menjelaskan pencatatan audit Amazon Redshift untuk koneksi dan aktivitas pengguna, dan menyimpan log ini di Amazon S3. Untuk informasi selengkapnya tentang pencatatan audit database, lihat Pencatatan audit basis data .	November 11, 2013

Ubah	Deskripsi	Tanggal rilis
Fitur baru	Menambahkan bagian untuk menjelaskan enkripsi Amazon Redshift dengan fitur baru untuk mengelola kunci enkripsi dalam modul keamanan perangkat keras (HSM) dan memutar kunci enkripsi. Untuk informasi selengkapnya tentang enkripsi, HSM, dan rotasi kunci, lihat Enkripsi basis data Amazon Redshift Enkripsi menggunakan modul keamanan perangkat keras , dan Rotasi kunci enkripsi .	November 11, 2013
Fitur baru	Diperbarui untuk menjelaskan pemberitahuan penerbitan peristiwa Amazon Redshift dengan menggunakan Amazon SNS. Untuk informasi tentang pemberitahuan acara Amazon Redshift, lihat. Pemberitahuan acara klaster yang disediakan Amazon Redshift	November 11, 2013
Fitur baru	Diperbarui untuk menjelaskan izin tingkat sumber daya IAM. Untuk informasi tentang izin Amazon Redshift IAM, lihat. Keamanan di Amazon Redshift	Agustus 9, 2013
Fitur baru	Diperbarui untuk menjelaskan metrik kemajuan pemulihan. Untuk informasi selengkapnya, lihat Memulihkan cluster dari snapshot .	Agustus 9, 2013
Fitur baru	Diperbarui untuk menjelaskan berbagi snapshot cluster dan membuat metrik kemajuan snapshot. Untuk informasi selengkapnya, lihat Berbagi snapshot .	Juli 17, 2013
Perbaikan dokumentasi	Menerbitkan berbagai perbaikan dokumentasi.	8 Juli 2013
Layar konsol baru	Memperbarui Panduan Manajemen Amazon Redshift untuk mencocokkan perubahan di konsol Amazon Redshift.	April 22, 2013

Ubah	Deskripsi	Tanggal rilis
Panduan baru	Ini adalah rilis pertama dari Amazon Redshift Management Guide.	Februari 14, 2013