aws

Panduan Pengguna

Layanan Basis Data Relasional Amazon

Layanan Basis Data Relasional Amazon: Panduan Pengguna

Table of Contents

Apa itu Amazon Relational Database Service (Amazon RDS)?

Amazon Relational Database Service (Amazon RDS) adalah layanan web yang mempermudah pengaturan, pengoperasian, dan penskalaan basis data relasional di AWS Cloud Cloud. Layanan ini menyediakan kapasitas yang hemat biaya dan dapat diubah ukurannya untuk basis data relasional standar industri serta mengelola tugas administrasi basis data umum.

> **Note**
>
> Panduan ini membahas mesin basis data Amazon RDS, bukan Amazon Aurora. Untuk informasi tentang penggunaan Amazon Aurora, lihat Panduan Pengguna Amazon Aurora.

Jika Anda baru mengenal produk dan layanan AWS, mulai pelajari selengkapnya dengan sumber daya berikut:

- Untuk gambaran umum seluruh produk AWS, lihat Apa itu komputasi cloud?
- Amazon Web Services menyediakan sejumlah layanan basis data. Untuk mempelajari selengkapnya tentang berbagai opsi basis data yang tersedia di AWS, lihat Memilih layanan basis data AWS dan Menjalankan basis data di AWS.

Gambaran Umum Amazon RDS

Mengapa Anda ingin menjalankan basis data relasional di AWS Cloud? Karena AWS mengambil alih banyak tugas yang sulit dan menjemukan dalam manajemen basis data relasional.

Topik

- Amazon EC2 dan basis data on-premise
- Amazon RDS dan Amazon EC2
- Amazon RDS Custom untuk Oracle dan Microsoft SQL Server
- Amazon RDS on AWS Outposts

Amazon EC2 dan basis data on-premise

Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di AWS Cloud. Dengan menggunakan Amazon EC2, Anda tidak perlu berinvestasi pada perangkat keras di awal, sehingga Anda bisa mengembangkan dan men-deploy aplikasi lebih cepat.

Saat Anda membeli server on-premise, Anda mendapatkan CPU, memori, penyimpanan, dan IOPS, semuanya digabungkan. Dengan Amazon EC2, sumber daya ini akan dibagi sehingga Anda dapat menskalakannya secara terpisah. Jika Anda memerlukan lebih banyak CPU, lebih sedikit IOPS, atau lebih banyak penyimpanan, Anda dapat dengan mudah mengalokasikannya.

Untuk basis data relasional di server on-premise, Anda bertanggung jawab penuh atas server, sistem operasi, dan perangkat lunak. Untuk basis data pada instans Amazon EC2, AWS mengelola lapisan di bawah sistem operasi. Dengan cara ini, Amazon EC2 menghilangkan beberapa beban dalam mengelola server basis data on-premise.

Pada tabel berikut, Anda dapat menemukan perbandingan model manajemen untuk basis data on-premise dan Amazon EC2.

Fitur	Manajemen on-premise	Manajemen Amazon EC2
Optimisasi aplikasi	Pelanggan	Pelanggan
Penskalaan	Pelanggan	Pelanggan
Ketersediaan tinggi	Pelanggan	Pelanggan
Pencadangan basis data	Pelanggan	Pelanggan
Patching perangkat lunak basis data	Pelanggan	Pelanggan
Instalasi perangkat lunak basis data	Pelanggan	Pelanggan
Patching sistem operasi (OS)	Pelanggan	Pelanggan
Instalasi OS	Pelanggan	Pelanggan
Pemeliharaan server	Pelanggan	AWS

Fitur	Manajemen on-premise	Manajemen Amazon EC2
Siklus hidup perangkat keras	Pelanggan	AWS
Daya, jaringan, dan pendinginan	Pelanggan	AWS

Amazon EC2 bukan layanan yang dikelola sepenuhnya. Jadi, ketika Anda menjalankan basis data di Amazon EC2, Anda lebih rentan terhadap kesalahan pengguna. Misalnya, ketika Anda memperbarui sistem operasi atau perangkat lunak basis data secara manual, Anda mungkin secara tidak sengaja menyebabkan waktu henti aplikasi. Anda mungkin perlu waktu berjam-jam untuk memeriksa setiap perubahan untuk mengidentifikasi dan memperbaiki masalah.

Amazon RDS dan Amazon EC2

Amazon RDS adalah layanan basis data terkelola. Layanan ini bertanggung jawab atas sebagian besar tugas manajemen. Dengan menghilangkan tugas manual yang menjemukan, Amazon RDS membebaskan Anda untuk fokus pada aplikasi dan pengguna Anda. Kami merekomendasikan Amazon RDS melalui Amazon EC2 sebagai pilihan default Anda untuk sebagian besar deployment basis data.

Pada tabel berikut, Anda dapat menemukan perbandingan model manajemen di Amazon EC2 dan Amazon RDS.

Fitur	Manajemen Amazon EC2	Manajemen Amazon RDS
Optimisasi aplikasi	Pelanggan	Pelanggan
Penskalaan	Pelanggan	AWS
Ketersediaan tinggi	Pelanggan	AWS
Pencadangan basis data	Pelanggan	AWS
Patching perangkat lunak basis data	Pelanggan	AWS

Fitur	Manajemen Amazon EC2	Manajemen Amazon RDS
Instalasi perangkat lunak basis data	Pelanggan	AWS
Patching OS	Pelanggan	AWS
Instalasi OS	Pelanggan	AWS
Pemeliharaan server	AWS	AWS
Siklus hidup perangkat keras	AWS	AWS
Daya, jaringan, dan pendingin an	AWS	AWS

Amazon RDS memberikan keuntungan spesifik berikut dibandingkan deployment basis data yang tidak sepenuhnya dikelola:

- Anda dapat menggunakan produk basis data yang Anda kenal: Db2, MariaDB, Microsoft SQL Server, MySQL, Oracle, dan PostgreSQL.
- Amazon RDS mengelola pencadangan, patching perangkat lunak, deteksi kegagalan otomatis, dan pemulihan.
- Anda dapat mengaktifkan cadangan otomatis, atau membuat snapshot cadangan Anda sendiri secara manual. Anda dapat menggunakan cadangan ini untuk memulihkan basis data. Proses pemulihan Amazon RDS berjalan dengan andal dan efisien.
- Anda dapat memperoleh ketersediaan tinggi dengan instans primer dan instans sekunder yang sinkron yang dapat menjadi target failover jika terjadi masalah. Anda juga dapat menggunakan replika baca untuk meningkatkan penskalaan proses baca.
- Selain keamanan dalam paket basis data, Anda dapat membantu mengontrol siapa yang dapat mengakses basis data RDS Anda. Untuk melakukannya, Anda dapat menggunakan AWS Identity and Access Management (IAM) untuk menentukan pengguna dan izin. Anda juga dapat membantu melindungi basis data dengan memasukkannya ke cloud privat virtual (VPC).

Amazon RDS Custom untuk Oracle dan Microsoft SQL Server

Amazon RDS Custom adalah jenis manajemen RDS yang memberi Anda akses penuh ke basis data dan sistem operasi Anda.

Anda dapat menggunakan kemampuan kontrol RDS Custom untuk mengakses dan menyesuaikan lingkungan basis data dan sistem operasi untuk aplikasi bisnis lama dan paket aplikasi bisnis. Sementara itu, Amazon RDS mengotomatiskan tugas dan operasi administrasi basis data.

Dalam model deployment ini, Anda dapat menginstal aplikasi dan mengubah pengaturan konfigurasi agar sesuai dengan aplikasi Anda. Pada saat yang sama, Anda dapat mengalihkan tugas administrasi basis data seperti penyediaan, penskalaan, upgrade, dan pencadangan ke AWS. Anda dapat memanfaatkan manfaat manajemen basis data Amazon RDS, dengan lebih banyak kontrol dan fleksibilitas.

Untuk Oracle Database dan Microsoft SQL Server, RDS Custom menggabungkan otomatisasi Amazon RDS dengan fleksibilitas Amazon EC2. Untuk informasi selengkapnya tentang RDS Custom, lihat Menggunakan Amazon RDS Custom.

Dengan model tanggung jawab bersama RDS Custom, Anda mendapatkan lebih banyak kontrol daripada di Amazon RDS, tetapi juga lebih banyak tanggung jawab. Untuk informasi selengkapnya, lihat Model tanggung jawab bersama dalam RDS Custom.

Amazon RDS on AWS Outposts

Amazon RDS on AWS Outposts memperluas basis data RDS for SQL Server, RDS for MySQL, dan RDS for PostgreSQL ke lingkungan AWS Outposts. AWS Outposts menggunakan perangkat keras yang sama dengan yang ada di Wilayah AWS publik untuk menghadirkan layanan AWS, infrastruktur, dan model operasi on-premise. Dengan RDS on Outposts, Anda dapat menyediakan instans DB terkelola di dekat aplikasi bisnis yang harus dijalankan secara on-premise. Untuk informasi selengkapnya, lihat Menggunakan Amazon RDS on AWS Outposts.

Instans DB

Instans DB adalah lingkungan basis data terisolasi di AWS Cloud. Blok bangunan dasar Amazon RDS adalah instans DB.

Instans DB Anda dapat berisi satu atau beberapa basis data yang dibuat pengguna. Anda dapat mengakses instans DB tersebut dengan menggunakan alat dan aplikasi yang sama dengan yang Anda gunakan dengan instans basis data mandiri. Anda dapat membuat dan mengubah instans

DB dengan menggunakan AWS Command Line Interface (AWS CLI), API Amazon RDS, atau AWS Management Console.

Mesin DB

Mesin DB adalah perangkat lunak basis data relasional spesifik yang berjalan pada instans DB Anda. Amazon RDS saat ini mendukung mesin berikut:

- Db2
- MariaDB
- Microsoft SQL Server
- MySQL
- Oracle
- PostgreSQL

Setiap mesin DB memiliki fitur yang didukungnya sendiri, dan setiap versi mesin DB dapat menyertakan fitur tertentu. Dukungan untuk fitur Amazon RDS bervariasi di seluruh Wilayah AWS dan versi spesifik dari setiap mesin DB. Untuk memeriksa dukungan fitur di berbagai versi mesin dan Wilayah, lihat Fitur yang didukung di Amazon RDS oleh Wilayah AWS dan mesin DB.

Selain itu, setiap mesin DB memiliki serangkaian parameter dalam grup parameter DB yang mengontrol perilaku basis data yang dikelola.

Kelas instans DB

Kelas instans DB menentukan kapasitas komputasi dan memori dari instans DB. Sebuah kelas instans DB terdiri dari jenis dan ukuran instans DB. Setiap jenis instans menawarkan kemampuan komputasi, memori, dan penyimpanan yang berbeda. Misalnya, db.m6g adalah jenis instans DB tujuan umum yang didukung oleh prosesor AWS Graviton2. Dalam jenis instans db.m6g, db.m6g.2xlarge adalah kelas instans DB.

Anda dapat memilih instans DB yang paling sesuai dengan kebutuhan Anda. Jika kebutuhan Anda berubah seiring waktu, Anda dapat mengubah instans DB. Untuk informasi, lihat Kelas instans DB .

> **Note**
>
> Untuk informasi harga kelas instans DB, lihat bagian Harga di halaman produk Amazon RDS.

Penyimpanan instans DB

EBS Amazon memberikan volume penyimpanan tingkat blok durabel yang dapat Anda lampirkan ke instans berjalan. Penyimpanan instans DB tersedia dalam jenis berikut:

- Tujuan Umum (SSD)
- IOPS yang Tersedia (PIOPS)
- Magnetik

Jenis penyimpanan ini berbeda dalam karakteristik performa dan harga. Anda dapat menyesuaikan performa dan biaya penyimpanan sesuai kebutuhan basis data Anda.

Setiap instans DB memiliki persyaratan penyimpanan minimum dan maksimum bergantung pada jenis penyimpanan dan mesin basis data yang didukungnya. Penting untuk memiliki penyimpanan yang memadai sehingga basis data Anda dapat berkembang. Selain itu, penyimpanan yang memadai memastikan bahwa fitur untuk mesin DB memiliki ruang untuk menulis entri konten atau log. Untuk informasi selengkapnya, lihat Penyimpanan instans DB Amazon RDS.

Amazon Virtual Private Cloud (Amazon VPC)

Anda dapat menjalankan instans DB pada cloud privat virtual (VPC) menggunakan layanan Amazon Virtual Private Cloud (Amazon VPC). Saat menggunakan VPC, Anda dapat mengontrol lingkungan jaringan virtual Anda. Anda dapat memilih rentang alamat IP Anda sendiri, membuat subnet, serta mengonfigurasi perutean dan daftar kontrol akses. Fungsi dasar Amazon RDS sama saat dijalankan di VPC atau cloud lain. Amazon RDS mengelola pencadangan, patching perangkat lunak, deteksi kegagalan otomatis, dan pemulihan. Tidak ada biaya tambahan untuk menjalankan instans DB Anda dalam VPC. Untuk informasi selengkapnya tentang penggunaan Amazon VPC dengan RDS, lihat Amazon VPC dan Amazon RDS.

Amazon RDS menggunakan Network Time Protocol (NTP) untuk menyinkronkan waktu pada Instans DB.

Wilayah dan Zona Ketersediaan AWS

Sumber daya komputasi cloud Amazon berlokasi di fasilitas pusat data dengan ketersediaan tinggi di berbagai wilayah di dunia (misalnya, Amerika Utara, Eropa, atau Asia). Setiap lokasi pusat data disebut Wilayah AWS.

Setiap Wilayah AWS berisi beberapa lokasi berbeda yang disebut Zona Ketersediaan, atau AZ. Setiap Zona Ketersediaan dirancang agar terisolasi dari kegagalan di Zona Ketersediaan yang lain. Setiap AZ direkayasa untuk menyediakan konektivitas jaringan latensi rendah yang hemat biaya ke Zona Ketersediaan lainnya di dalam Wilayah AWS yang sama. Dengan meluncurkan instans dalam Zona Ketersediaan yang terpisah, Anda dapat melindungi aplikasi Anda dari kegagalan di satu lokasi. Untuk informasi selengkapnya, lihat Wilayah, Zona Ketersediaan, dan Zona Lokal.

Anda dapat menjalankan instans DB Anda di beberapa Zona Ketersediaan, sebuah opsi yang disebut deployment Multi-AZ. Saat Anda memilih opsi ini, Amazon secara otomatis menyediakan dan memelihara satu atau beberapa instans DB siaga sekunder di Zona Ketersediaan yang berbeda. Instans DB primer Anda direplikasi di seluruh Zona Ketersediaan ke setiap instans DB sekunder. Pendekatan ini membantu menyediakan redundansi data dan dukungan failover, menghilangkan kemacetan I/O, dan meminimalkan lonjakan latensi selama pencadangan sistem. Dalam deployment klaster DB Multi-AZ, instans DB sekunder juga dapat melayani lalu lintas baca. Untuk informasi selengkapnya, lihat Mengonfigurasi dan mengelola deployment Multi-AZ.

Keamanan

Grup keamanan mengontrol akses ke instans DB. Ini dilakukan dengan memungkinkan akses ke rentang alamat IP atau instans Amazon EC2 yang Anda tentukan.

Untuk informasi selengkapnya tentang grup keamanan, lihat Keamanan dalam Amazon RDS.

Pemantauan Amazon RDS

Ada beberapa cara untuk melacak performa dan kondisi instans DB. Anda dapat menggunakan CloudWatch layanan Amazon untuk memantau kinerja dan kesehatan instans DB. CloudWatch grafik kinerja ditampilkan di konsol Amazon RDS. Anda juga dapat berlangganan peristiwa Amazon RDS untuk mendapatkan notifikasi tentang perubahan pada instans DB, snapshot DB, atau grup parameter DB. Untuk informasi selengkapnya, lihat Memantau metrik dalam instans Amazon RDS.

Cara menggunakan Amazon RDS

Ada beberapa cara berinteraksi dengan Amazon RDS.

AWS Management Console

AWS Management Console adalah antarmuka pengguna berbasis web yang sederhana. Anda dapat mengelola instans DB Anda dari konsol tersebut tanpa perlu pemrograman. Untuk mengakses

konsol Amazon RDS, masuk ke AWS Management Console lalu buka konsol tersebut di https://console.aws.amazon.com/rds/.

Antarmuka baris perintah

Anda dapat menggunakan AWS Command Line Interface (AWS CLI) untuk mengakses API Amazon RDS secara interaktif. Untuk menginstal,AWS CLI lihat Menginstal AWS Command Line Interface. Untuk mulai menggunakan AWS CLI untuk RDS, lihat Referensi AWS Command Line Interface untuk Amazon RDS.

API Amazon RDS

Jika Anda adalah seorang developer, Anda dapat mengakses Amazon RDS secara programatis menggunakan API. Untuk informasi selengkapnya, lihat Referensi API Amazon RDS.

Untuk pengembangan aplikasi, kami sarankan Anda menggunakan Kit AWS Pengembangan Perangkat Lunak (SDK). SDK AWS menangani detail tingkat rendah seperti autentikasi, logika percobaan ulang, dan penanganan kesalahan, sehingga Anda dapat fokus pada logika aplikasi Anda. SDK AWS tersedia dalam berbagai bahasa. Untuk informasi selengkapnya, lihat Alat untuk Amazon Web Services.

AWS juga menyediakan pustaka, kode sampel, tutorial, dan sumber daya lain untuk membantu Anda memulai dengan lebih mudah. Untuk informasi selengkapnya, lihat Kode sampel & pustaka.

Bagaimana Anda dikenai biaya untuk Amazon RDS

Saat Anda menggunakan Amazon RDS, Anda dapat memilih untuk menggunakan instans DB sesuai permintaan atau instans DB terpesan. Untuk informasi selengkapnya, lihat Penagihan instans DB untuk Amazon RDS.

Untuk informasi harga Amazon RDS, lihat Halaman produk Amazon RDS.

Apa selanjutnya?

Bagian sebelumnya memperkenalkan komponen infrastruktur dasar yang ditawarkan RDS. Apa yang harus Anda lakukan selanjutnya?

Memulai

Buat instans DB sesuai petunjuk dalam Mulai menggunakan Amazon RDS.

Topik khusus untuk mesin basis data

Anda dapat meninjau informasi khusus tentang mesin DB tertentu di bagian berikut:

- Amazon RDS for Db2
- Amazon RDS for MariaDB
- Amazon RDS for Microsoft SQL Server
- Amazon RDS for MySQL
- Amazon RDS for Oracle
- Amazon RDS for PostgreSQL

Model tanggung jawab bersama Amazon RDS

Amazon RDS bertanggung jawab untuk meng-host komponen perangkat lunak dan infrastruktur instans DB dan klaster DB. Anda bertanggung jawab untuk penyetelan kueri, yang merupakan proses penyetelan kueri SQL untuk meningkatkan performa. Performa kueri sangat bergantung pada desain basis data, ukuran data, distribusi data, beban kerja aplikasi, dan pola kueri, yang dapat sangat bervariasi. Pemantauan dan penyetelan adalah proses sangat khusus yang Anda miliki untuk basis data RDS Anda. Anda dapat menggunakan Wawasan Performa Amazon RDS dan alat lainnya untuk mengidentifikasi kueri yang bermasalah.

Instans DB Amazon RDS

Instans DB adalah lingkungan basis data terisolasi yang berjalan di cloud. Ini adalah blok bangunan dasar Amazon RDS. Instans DB dapat berisi beberapa basis data buatan pengguna, dan dapat diakses menggunakan alat dan aplikasi klien yang sama yang dapat Anda gunakan untuk mengakses instans basis data mandiri. Pembuatan dan modifikasi instans DB mudah dilakukan menggunakan alat baris perintah AWS, operasi Amazon RDS API, atau AWS Management Console.

> **Note**
>
> Amazon RDS mendukung akses ke basis data menggunakan aplikasi klien SQL standar. Amazon RDS tidak mengizinkan akses host langsung.

Anda dapat memiliki hingga 40 instans DB Amazon RDS, dengan batasan berikut:

- 10 untuk setiap edisi SQL Server (Enterprise, Standard, Web, dan Express) di dalam model "license-included"
- 10 untuk Oracle dalam model "license-included"
- 40 untuk Db2 di bawah model lisensi "bring-your-own-license" (BYOL)
- 40 untuk MySQL, MariaDB, atau PostgreSQL
- 40 untuk Oracle di bawah model lisensi bring-your-own-license "" (BYOL)

> **Note**
>
> Jika aplikasi Anda memerlukan lebih banyak instans DB, Anda dapat meminta instans DB tambahan menggunakan formulir ini.

Setiap instans DB memiliki pengidentifikasi instans DB. Nama yang diberikan pelanggan ini secara unik mengidentifikasi instans DB saat berinteraksi dengan Amazon RDS API dan perintah AWS CLI. Pengidentifikasi instans DB harus unik untuk pelanggan di Wilayah AWS.

Pengidentifikasi instans DB merupakan bagian dari nama host DNS yang dialokasikan ke instans Anda oleh RDS. Misalnya, jika Anda menetapkan db1 sebagai pengidentifikasi instans DB, RDS akan secara otomatis mengalokasikan titik akhir DNS untuk instans Anda. Contoh titik akhir adalah*db1*.*abcdefghijkl*.*us-east-1*.rds.amazonaws.com, dengan instans ID *db1*.

Dalam contoh titik akhir *db1*.*abcdefghijkl*.*us-east-1*.rds.amazonaws.com, string *abcdefghijkl* adalah pengidentifikasi unik untuk kombinasi spesifik Wilayah AWS dan Akun AWS. Pengidentifikasi *abcdefghijkl* dalam contoh dihasilkan secara internal oleh RDS dan tidak berubah untuk kombinasi Wilayah dan akun yang ditentukan. Dengan demikian, semua instans DB Anda di Wilayah ini memiliki pengidentifikasi tetap yang sama. Pertimbangkan fitur pengenal tetap berikut:

- Saat nama instans DB diubah, titik akhir akan berbeda, tetapi pengidentifikasi tetapnya masih sama. Misalnya, jika Anda mengganti nama *db1* menjadi *renamed-db1*, titik akhir instans yang baru adalah *renamed-db1*.*abcdefghijkl*.*us-east-1*.rds.amazonaws.com.
- Jika Anda menghapus dan membuat ulang instans DB dengan pengidentifikasi instans DB yang sama, titik akhirnya akan sama.
- Jika Anda menggunakan akun yang sama untuk membuat instans DB di Wilayah yang berbeda, pengidentifikasi yang dibuat secara internal akan berbeda karena Wilayahnya berbeda, seperti dalam *db2*.*mnopqrstuvwx*.*us-west-1*.rds.amazonaws.com.

Setiap instans DB mendukung sebuah mesin basis data. Amazon RDS saat ini mendukung mesin basis data Db2, MySQL, MariaDB, PostgreSQL, Oracle, Microsoft SQL Server, dan Amazon Aurora.

Saat membuat instans DB, beberapa mesin basis data mengharuskan penentuan nama basis data. Instans DB dapat menjadi host untuk banyak basis data, satu basis data Db2, atau satu basis data Oracle dengan beberapa skema. Nilai nama basis data bergantung pada mesin basis data:

- Untuk mesin basis data Db2, nama basis datanya adalah nama basis data yang di-hosting di instans DB Anda. Jika Anda ingin menggunakan prosedur tersimpan Amazon RDS untuk membuat atau membatalkan basis data, jangan masukkan nama basis data saat Anda membuat instans DB.
- Untuk mesin basis data MySQL dan MariaDB, nama basis datanya adalah nama basis data yang di-hosting di instans DB Anda. Basis data yang di-hosting oleh instans DB yang sama harus memiliki nama unik dalam instans tersebut.
- Untuk mesin basis data Oracle, nama basis data digunakan untuk menetapkan nilai ORACLE_SID, yang harus disediakan saat terhubung ke instans RDS Oracle.
- Mesin basis data Microsoft SQL Server tidak mendukung parameter nama basis data.
- Untuk mesin basis data PostgreSQL, nama basis datanya adalah nama basis data yang di-hosting di instans DB Anda. Nama basis data tidak diperlukan saat membuat instans DB. Basis data yang di-hosting oleh instans DB yang sama harus memiliki nama unik dalam instans tersebut.

Amazon RDS membuat akun pengguna utama untuk instans DB Anda sebagai bagian dari proses pembuatan. Pengguna utama ini memiliki izin untuk membuat basis data dan untuk menjalankan operasi pembuatan, penghapusan, pemilihan, pembaruan, dan penyisipan pada tabel yang dibuat oleh pengguna utama. Anda harus menetapkan kata sandi pengguna utama saat membuat instans DB, yang dapat diubah kapan saja menggunakan perintah AWS CLI, operasi Amazon RDS API, atau AWS Management Console. Anda juga dapat mengubah kata sandi pengguna utama dan mengelola pengguna menggunakan perintah SQL standar.

> **Note**
>
> Panduan ini mencakup mesin basis data Amazon RDS non-Aurora. Untuk informasi tentang penggunaan Amazon Aurora, lihat Panduan Pengguna Amazon Aurora.

Kelas instans DB

Kelas instans DB menentukan komputasi dan kapasitas memori instans DB Amazon RDS . Kelas instans DB yang Anda butuhkan tergantung pada kebutuhan daya dan memori pemrosesan Anda.

Sebuah kelas instans DB terdiri dari jenis dan ukuran kelas instans DB. Misalnya, db.r6g adalah tipe kelas instans DB yang dioptimalkan memori yang didukung oleh prosesor Graviton2. AWS Dalam jenis kelas instans db.r6g, db.r6g.2xlarge adalah kelas instans DB. Ukuran kelas ini adalah 2xlarge.

Untuk informasi selengkapnya tentang harga kelas instans, lihat Harga Amazon RDS.

Topik

Jenis kelas instans DB

Amazon RDS mendukung kelas instans DB untuk kasus penggunaan berikut:

- Tujuan umum
- Memori yang dioptimalkan
- Komputasi dioptimalkan
- Performa yang dapat melonjak
- Optimized Reads

Untuk informasi selengkapnya tentang jenis instans Amazon EC2, lihat Jenis instans di dokumentasi Amazon EC2.

Jenis kelas instans tujuan umum

Kelas instans DB tujuan umum berikut tersedia:

- db.m7g - Kelas instans DB tujuan umum yang didukung oleh prosesor Graviton3. AWS Kelas instans ini memberikan komputasi, memori, dan jaringan yang seimbang untuk berbagai beban kerja tujuan umum.

  Anda dapat memodifikasi instans DB untuk menggunakan salah satu kelas instans DB yang didukung oleh prosesor AWS Graviton3. Untuk melakukannya, selesaikan langkah yang sama seperti modifikasi instans DB lainnya.
- db.m6g - Kelas instans DB tujuan umum yang didukung oleh prosesor Graviton2. AWS Instans ini memberikan komputasi, memori, dan jaringan yang seimbang untuk berbagai beban kerja tujuan umum. Kelas instans db.m6gd memiliki penyimpanan tingkat blok SSD berbasis NVMe lokal untuk aplikasi yang membutuhkan penyimpanan lokal berkecepatan tinggi dan latensi rendah.

  Anda dapat memodifikasi instans DB untuk menggunakan salah satu kelas instans DB yang didukung oleh prosesor AWS Graviton2. Untuk melakukannya, selesaikan langkah yang sama seperti modifikasi instans DB lainnya.
- db.m6i – Kelas instans DB tujuan umum yang didukung oleh prosesor Intel Xeon Scalable Generasi ke-3. Instans ini bersertifikat SAP dan ideal untuk beban kerja seperti server backend yang mendukung aplikasi perusahaan, server game, armada cache, dan lingkungan pengembangan aplikasi. Kelas instans db.m6id dan db.m6idn menawarkan hingga 7,6 TB penyimpanan SSD berbasis NVMe lokal, sedangkan db.m6in menawarkan penyimpanan khusus EBS. Kelas db.m6in dan db.m6idn menawarkan bandwidth jaringan hingga 200 Gbps.
- db.m5 – Kelas instans DB tujuan umum yang menghadirkan keseimbangan sumber daya komputasi, memori, dan jaringan, dan merupakan pilihan tepat untuk banyak aplikasi. Kelas instans db.m5d menawarkan penyimpanan SSD berbasis NVMe yang terhubung secara fisik ke server host. Kelas instans db.m5 menyediakan kapasitas komputasi yang lebih besar dari kelas instans db.m4 sebelumnya. Produk ini didukung oleh AWS Nitro System, kombinasi perangkat keras khusus dan hipervisor ringan.
- db.m4 – Kelas instans DB tujuan umum yang menyediakan kapasitas komputasi lebih besar daripada kelas instans db.m3 sebelumnya.

  Untuk mesin DB RDS for Oracle, Amazon RDS tidak lagi mendukung kelas instans DB db.m4. Jika sebelumnya Anda telah membuat instans DB RDS for Oracle db.m4, Amazon RDS secara otomatis meningkatkan instans DB tersebut ke kelas instans DB db.m5 yang setara.
- db.m3 – Kelas instans DB tujuan umum yang menyediakan kapasitas komputasi lebih besar daripada kelas instans db.m1 sebelumnya.

Untuk mesin RDS untuk MariaDB, RDS untuk MySQL, dan RDS untuk PostgreSQL DB, Amazon RDS telah end-of-life memulai proses untuk kelas instans db.m3 DB menggunakan jadwal berikut, yang mencakup rekomendasi peningkatan. Untuk semua instans DB RDS yang menggunakan kelas instans DB db.m3, sebaiknya ditingkatkan ke kelas instans DB generasi yang lebih tinggi sesegera mungkin.

Tindakan atau rekomendasi	Tanggal
Anda tidak dapat lagi membuat instans DB RDS yang menggunakan kelas instans DB db.m3.	Sekarang
Amazon RDS memulai peningkatan otomatis instans DB RDS yang menggunakan kelas instans DB db.m3 ke kelas instans DB db.m5 yang setara.	1 Februari 2023

Jenis kelas instans memori yang dioptimalkan

Rangkaian Z dengan memori yang dioptimalkan mendukung kelas instans berikut:

- db.z1d – Kelas instans yang dioptimalkan untuk aplikasi yang memakan banyak memori. Kelas instans ini menawarkan kapasitas komputasi tinggi dan jejak memori yang tinggi. Instans z1d frekuensi tinggi menghadirkan frekuensi all-core berkelanjutan hingga 4,0 GHz.

Rangkaian X dengan memori yang dioptimalkan mendukung kelas instans berikut:

- db.x2g - Kelas instans yang dioptimalkan untuk aplikasi intensif memori dan didukung oleh prosesor Graviton2. AWS Kelas instans ini menawarkan biaya rendah per GiB memori.

  Anda dapat memodifikasi instans DB untuk menggunakan salah satu kelas instans DB yang didukung oleh prosesor AWS Graviton2. Untuk melakukannya, selesaikan langkah yang sama seperti modifikasi instans DB lainnya.
- db.x2i – Kelas instans yang dioptimalkan untuk aplikasi yang memakan banyak memori. Jenis kelas instans db.x2iedn dan db.x2idn didukung oleh prosesor Intel Xeon Scalable generasi ke-3 (Ice Lake). Kelas instans tersebut mencakup hingga 3,8 TB penyimpanan SSD NVMe lokal, bandwidth jaringan hingga 100 Gbps, dan memori hingga 4 TiB (db.x2iden) atau 2 TiB (db.x2idn).

Jenis db.x2iezn didukung oleh prosesor Intel Xeon Scalable generasi ke-2 (Cascade Lake) dengan frekuensi turbo all-core hingga 4,5 GHz dan memori hingga 1,5 TiB.

- db.x1 – Kelas instans yang dioptimalkan untuk aplikasi yang memakan banyak memori. Kelas instans ini menawarkan salah satu harga terendah per GiB RAM di antara kelas instans DB dan memori instans berbasis DRAM hingga 1.952 GiB. Jenis kelas instans db.x1e menawarkan memori instans berbasis DRAM hingga 3.904 GiB.

Rangkaian R dengan memori yang dioptimalkan mendukung jenis kelas instans berikut:

- db.r7g - Kelas instans yang didukung oleh prosesor Graviton3. AWS Kelas instans ini ideal untuk menjalankan beban kerja yang memerlukan banyak memori dalam basis data sumber terbuka, seperti MySQL dan PostgreSQL.

  Anda dapat memodifikasi instans DB untuk menggunakan salah satu kelas instans DB yang didukung oleh prosesor AWS Graviton3. Untuk melakukannya, selesaikan langkah yang sama seperti modifikasi instans DB lainnya.
- db.r6g - Kelas instans yang didukung oleh prosesor Graviton2. AWS Kelas instans ini ideal untuk menjalankan beban kerja yang memerlukan banyak memori dalam basis data sumber terbuka, seperti MySQL dan PostgreSQL. Jenis db.r6gd menawarkan penyimpanan tingkat blok SSD berbasis NVMe lokal untuk aplikasi yang memerlukan penyimpanan lokal berkecepatan tinggi dan latensi rendah.

  Anda dapat memodifikasi instans DB untuk menggunakan salah satu kelas instans DB yang didukung oleh prosesor AWS Graviton2. Untuk melakukannya, selesaikan langkah yang sama seperti modifikasi instans DB lainnya.
- db.r6i – Kelas instans yang didukung oleh prosesor Intel Xeon Scalable Generasi ke-3. Kelas instans ini bersertifikat SAP dan ideal untuk beban kerja yang memerlukan banyak memori dalam basis data sumber terbuka, seperti MySQL dan PostgreSQL. Kelas instans db.r6id, db.r6in, dan db.r6idn memiliki rasio CPU 8:1 dan memori maksimum 1 TiB. memory-to-v Kelas db.r6id dan db.r6idn menawarkan hingga 7,6 TB penyimpanan SSD berbasis NVMe yang terpasang langsung, sedangkan db.r6in menawarkan penyimpanan khusus EBS. Kelas db.r6idn dan db.r6in menawarkan bandwidth jaringan hingga 200 Gbps.
- db.r5b – Kelas instans yang memorinya dioptimalkan untuk aplikasi intensif throughput. Didukung oleh Sistem AWS Nitro, instans db.r5b memberikan bandwidth hingga 60 Gbps dan kinerja EBS 260.000 IOPS. Ini adalah performa penyimpanan blok tercepat di EC2.

- db.r5d – Kelas instans yang dioptimalkan untuk latensi rendah, performa I/O acak yang sangat tinggi, dan throughput baca berurutan tinggi.
- db.r5 – Kelas instans yang dioptimalkan untuk aplikasi yang memakan banyak memori. Kelas instans ini menawarkan peningkatan performa jaringan). Mereka didukung oleh Sistem AWS Nitro, kombinasi perangkat keras khusus dan hypervisor ringan.
- db.r4 – Kelas instans yang menyediakan peningkatan jaringan dibandingkan kelas instans db.r3 sebelumnya.

  Untuk mesin RDS untuk Oracle DB, Amazon RDS telah memulai end-of-life proses untuk kelas instans db.r4 DB menggunakan jadwal berikut, yang mencakup rekomendasi peningkatan. Untuk instans DB RDS for Oracle yang menggunakan kelas instans db.r4, sebaiknya Anda meningkatkan ke kelas instans generasi yang lebih tinggi sesegera mungkin.

Tindakan atau rekomendasi	Tanggal
Anda tidak dapat lagi membuat instans DB RDS for Oracle yang menggunakan kelas instans DB db.r4.	Sekarang
Amazon RDS memulai peningkatan otomatis untuk instans DB RDS for Oracle yang menggunakan kelas instans DB db.r4 ke kelas instans DB db.r5 yang setara.	17 April 2023

- db.r3 – Kelas instans yang menyediakan optimasi memori.

  Untuk mesin RDS untuk MariaDB, RDS untuk MySQL, dan RDS untuk PostgreSQL DB, Amazon RDS telah end-of-life memulai proses untuk kelas instans db.r3 DB menggunakan jadwal berikut, yang mencakup rekomendasi peningkatan. Untuk semua instans DB RDS yang menggunakan kelas instans DB db.r3, sebaiknya ditingkatkan ke kelas instans DB generasi yang lebih tinggi sesegera mungkin.

Tindakan atau rekomendasi	Tanggal
Anda tidak dapat lagi membuat instans DB RDS yang menggunakan kelas instans DB db.r3.	Sekarang

Tindakan atau rekomendasi	Tanggal
Amazon RDS memulai peningkatan otomatis untuk instans DB RDS yang menggunakan kelas instans DB db.r3 ke kelas instans DB db.r5 yang setara.	1 Februari 2023

Jenis kelas instance yang dioptimalkan untuk komputasi

Jenis kelas instance yang dioptimalkan komputasi berikut tersedia:

- db.c6gd — Kelas instans yang ideal untuk menjalankan beban kerja intensif komputasi tingkat lanjut. Didukung oleh prosesor AWS Graviton2, kelas instans ini menawarkan penyimpanan tingkat blok SSD berbasis NVME lokal untuk aplikasi yang membutuhkan penyimpanan lokal berkecepatan tinggi dan latensi rendah.

> **Note**
>
> Kelas instans c6gd hanya didukung untuk penerapan klaster DB multi-AZ. Mereka adalah satu-satunya kelas instance yang didukung untuk cluster DB multi-AZ yang menawarkan ukuran `medium` instans. Untuk informasi selengkapnya, lihat the section called “Deployment klaster basis data Multi-AZ”.

Jenis kelas instans performa yang dapat melonjak

Jenis kelas instans DB performa yang dapat melonjak berikut tersedia:

- db.t4g - Kelas instance tujuan umum yang didukung oleh prosesor Graviton2 berbasis ARM. AWS Kelas instans ini memberikan performa harga yang lebih baik daripada kelas instans DB performa yang dapat melonjak sebelumnya untuk serangkaian beban kerja tujuan umum yang dapat melonjak. Instans Amazon RDS db.t4g dikonfigurasi untuk mode Tidak Terbatas. Artinya, instans tersebut dapat melampaui garis dasar dalam jangka waktu 24 jam dengan biaya tambahan.

  Anda dapat memodifikasi instans DB untuk menggunakan salah satu kelas instans DB yang didukung oleh prosesor AWS Graviton2. Untuk melakukannya, selesaikan langkah yang sama seperti modifikasi instans DB lainnya.

- db.t3 – Kelas instans yang memberikan tingkat performa dasar, dengan kemampuan untuk melonjak hingga penggunaan CPU penuh. Instans db.t3 dikonfigurasi untuk mode Tidak Terbatas. Kelas instans ini memberikan kapasitas komputasi yang lebih besar dibandingkan kelas instans db.t2 sebelumnya. Produk ini didukung oleh Nitro System AWS , kombinasi perangkat keras khusus dan hypervisor ringan.
- db.t2 – Kelas instans yang memberikan tingkat performa dasar, dengan kemampuan untuk melonjak hingga penggunaan CPU penuh. Instans db.t2 dikonfigurasi untuk mode Tidak Terbatas. Sebaiknya gunakan kelas instans ini hanya untuk server pengujian dan pengembangan, atau server non-produksi lainnya.

> **Note**
>
> Kelas instans DB yang menggunakan Sistem AWS Nitro (db.m5, db.r5, db.t3) dibatasi pada beban kerja gabungan baca plus tulis.

Untuk spesifikasi perangkat keras kelas instans DB, lihat Spesifikasi perangkat keras kelas instans DB .

Jenis kelas instans Optimized Reads

Jenis kelas instans Optimized Reads berikut tersedia:

- db.r6gd - Kelas instans yang didukung oleh prosesor Graviton2. AWS Kelas instans ini ideal untuk menjalankan beban kerja intensif memori dan menawarkan penyimpanan tingkat blok SSD berbasis NVME lokal untuk aplikasi yang membutuhkan penyimpanan lokal berkecepatan tinggi dan latensi rendah.
- db.r6id – Kelas instans yang didukung prosesor Intel Xeon Scalable Generasi ke-3. Kelas instans ini bersertifikat SAP dan ideal untuk beban kerja yang menggunakan banyak memori. Kelas instans tersebut menawarkan memori maksimum 1 TiB dan hingga 7,6 TB penyimpanan SSD berbasis NVMe default.

Mesin DB yang didukung untuk kelas instans DB

Berikut ini adalah pertimbangan khusus mesin DB untuk kelas instans DB:

Db2

Dukungan kelas instans DB bervariasi menurut versi dan edisi Db2. Untuk dukungan kelas instans berdasarkan versi dan edisi, lihat Kelas-kelas instans RDS for Db2.

Microsoft SQL Server

Dukungan kelas instans DB bervariasi menurut versi dan edisi SQL Server. Untuk dukungan kelas instans berdasarkan versi dan edisi, lihat Dukungan kelas instans DB untuk Microsoft SQL Server.

Oracle

Dukungan kelas instans DB bervariasi menurut versi dan edisi Oracle Database. RDS for Oracle mendukung kelas instans tambahan dengan memori yang dioptimalkan. Kelas ini memiliki nama berupa db.r5.*instance_size*.tpc*threads_per_core*.mem*ratio*. Untuk jumlah vCPU dan alokasi memori untuk setiap kelas yang dioptimalkan, lihat Kelas instans RDS for Oracle yang didukung.

RDS Custom

Untuk informasi tentang kelas instans DB yang didukung di RDS Custom, lihat Dukungan kelas instans DB untuk RDS Custom for Oracle dan Dukungan kelas instans DB untuk RDS Custom for SQL Server.

Pada tabel berikut, Anda dapat menemukan detail tentang kelas instans DB Amazon RDS yang didukung untuk setiap mesin DB Amazon RDS. Sel untuk setiap mesin berisi salah satu nilai berikut:

Ya

Kelas instans didukung untuk semua versi mesin DB.

Tidak

Kelas instans tidak didukung untuk mesin DB.

versi spesifik

Kelas instans hanya didukung untuk versi basis data mesin DB tertentu.

Amazon RDS secara berkala tidak menggunakan versi mesin DB mayor dan minor. Tidak semua Wilayah AWS mungkin memiliki dukungan untuk versi mesin sebelumnya. Untuk informasi tentang

versi yang didukung saat ini, lihat topik untuk masing-masing mesin DB: versi MariaDB, versi Microsoft SQL Server, versi MySQL, versi Oracle, dan versi PostgreSQL.

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m7g — kelas instance tujuan umum yang didukung oleh prosesor Graviton3 AWS						
db.m7g.16xlarge	Tidak	Versi MariaDB 10.11, 10.6.10 dan versi 10.6 yang lebih tinggi, 10.5.17 dan versi 10.5 yang lebih tinggi, dan 10.4.26 dan versi 10.4 yang lebih tinggi	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.4 dan 13 versi yang lebih tinggi
db.m7g.12xlarge	Tidak	Versi MariaDB 10.11, 10.6.10 dan versi 10.6 yang lebih tinggi, 10.5.17 dan versi 10.5 yang lebih tinggi, dan 10.4.26 dan versi 10.4 yang lebih tinggi	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.4 dan 13 versi yang lebih tinggi
db.m7g.8xlarge	Tidak	Versi MariaDB 10.11, 10.6.10 dan versi 10.6 yang lebih	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
		tinggi, 10.5.17 dan versi 10.5 yang lebih tinggi, dan 10.4.26 dan versi 10.4 yang lebih tinggi				dan 14 versi yang lebih tinggi, dan 13.4 dan 13 versi yang lebih tinggi
db.m7g.4xlarge	Tidak	Versi MariaDB 10.11, 10.6.10 dan versi 10.6 yang lebih tinggi, 10.5.17 dan versi 10.5 yang lebih tinggi, dan 10.4.26 dan versi 10.4 yang lebih tinggi	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.4 dan 13 versi yang lebih tinggi
db.m7g.2xlarge	Tidak	Versi MariaDB 10.11, 10.6.10 dan versi 10.6 yang lebih tinggi, 10.5.17 dan versi 10.5 yang lebih tinggi, dan 10.4.26 dan versi 10.4 yang lebih tinggi	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.4 dan 13 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m7g.xlarge	Tidak	Versi MariaDB 10.11, 10.6.10 dan versi 10.6 yang lebih tinggi, 10.5.17 dan versi 10.5 yang lebih tinggi, dan 10.4.26 dan versi 10.4 yang lebih tinggi	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.4 dan 13 versi yang lebih tinggi
db.m7g.large	Tidak	Versi MariaDB 10.11, 10.6.10 dan versi 10.6 yang lebih tinggi, 10.5.17 dan versi 10.5 yang lebih tinggi, dan 10.4.26 dan versi 10.4 yang lebih tinggi	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.4 dan 13 versi yang lebih tinggi
db.m6g — kelas instance tujuan umum yang didukung oleh prosesor Graviton2 AWS						
db.m6g.16xlarge	Tidak	Semua versi MariaDB 10.11, 10.6, 10.5, dan 10.4	Tidak	MySQL 8.0.23 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6g.12xlarge	Tidak	Semua versi MariaDB 10.11, 10.6, 10.5, dan 10.4	Tidak	MySQL 8.0.23 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi
db.m6g.8xlarge	Tidak	Semua versi MariaDB 10.11, 10.6, 10.5, dan 10.4	Tidak	MySQL 8.0.23 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi
db.m6g.4xlarge	Tidak	Semua versi MariaDB 10.11, 10.6, 10.5, dan 10.4	Tidak	MySQL 8.0.23 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi
db.m6g.2xlarge	Tidak	Semua versi MariaDB 10.11, 10.6, 10.5, dan 10.4	Tidak	MySQL 8.0.23 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6g.xlarge	Tidak	Semua versi MariaDB 10.11, 10.6, 10.5, dan 10.4	Tidak	MySQL 8.0.23 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi
db.m6g.large	Tidak	Semua versi MariaDB 10.11, 10.6, 10.5, dan 10.4	Tidak	MySQL 8.0.23 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi
db.m6gd — kelas instance tujuan umum yang didukung oleh prosesor Graviton2 AWS						
db.m6gd.16xlarge	Tidak	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua versi PostgreSQL 16, 15, dan 14; 13.7 dan 13 versi yang lebih tinggi; dan 13.4

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6gd.12xlarge	Tidak	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua versi PostgreSQL 16, 15, dan 14; 13.7 dan 13 versi yang lebih tinggi; dan 13.4
db.m6gd.8xlarge	Tidak	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua versi PostgreSQL 16, 15, dan 14; 13.7 dan 13 versi yang lebih tinggi; dan 13.4

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6gd.4xlarge	Tidak	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua versi PostgreSQL 16, 15, dan 14; 13.7 dan 13 versi yang lebih tinggi; dan 13.4
db.m6gd.2xlarge	Tidak	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua versi PostgreSQL 16, 15, dan 14; 13.7 dan 13 versi yang lebih tinggi; dan 13.4

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6gd.xlarge	Tidak	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua versi PostgreSQL 16, 15, dan 14; 13.7 dan 13 versi yang lebih tinggi; dan 13.4
db.m6gd.large	Tidak	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua versi PostgreSQL 16, 15, dan 14; 13.7 dan 13 versi yang lebih tinggi; dan 13.4
db.m6id – Kelas instans tujuan umum yang didukung oleh prosesor Intel Xeon Scalable generasi ke-3						

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6id.32xlarge	Tidak	MariaDB 10.6.10 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi
db.m6id.24xlarge	Tidak	MariaDB 10.6.10 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6id.16xlarge	Tidak	MariaDB 10.6.10 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi
db.m6id.12xlarge	Tidak	MariaDB 10.6.10 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6id.8xlarge	Tidak	MariaDB 10.6.10 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi
db.m6id.4xlarge	Tidak	MariaDB 10.6.10 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6id.2xlarge	Tidak	MariaDB 10.6.10 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi
db.m6id.xlarge	Tidak	MariaDB 10.6.10 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6id.large	Tidak	MariaDB 10.6.10 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi
db.m6idn – kelas instans tujuan umum dengan prosesor Intel Xeon Scalable Generasi ke-3, penyimpanan SSD, dan pengoptimalan jaringan						
db.m6idn.32xlarge	Tidak	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6idn.24xlarge	Tidak	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi
db.m6idn.16xlarge	Tidak	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6idn.12xlarge	Tidak	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi
db.m6idn.8xlarge	Ya	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6idn.4xlarge	Ya	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi
db.m6idn.2xlarge	Ya	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6idn.xlarge	Ya	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi
db.m6idn.large	Ya	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi
db.m6in – kelas instans tujuan umum yang didukung oleh prosesor Intel Xeon Scalable generasi ke-3						

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6in.32xlarge	Tidak	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua versi PostgreSQL 16 dan 15, 14.3 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, 12.11 dan 12 versi yang lebih tinggi, dan 11.16 dan versi 11 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6in.24xlarge	Tidak	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua versi PostgreSQL 16 dan 15, 14.3 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, 12.11 dan 12 versi yang lebih tinggi, dan 11.16 dan versi 11 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6in.16xlarge	Tidak	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua versi PostgreSQL 16 dan 15, 14.3 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, 12.11 dan 12 versi yang lebih tinggi, dan 11.16 dan versi 11 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6in.12xlarge	Tidak	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua versi PostgreSQL 16 dan 15, 14.3 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, 12.11 dan 12 versi yang lebih tinggi, dan 11.16 dan versi 11 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6in.8xlarge	Ya	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua versi PostgreSQL 16 dan 15, 14.3 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, 12.11 dan 12 versi yang lebih tinggi, dan 11.16 dan versi 11 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6in.4xlarge	Ya	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua versi PostgreSQL 16 dan 15, 14.3 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, 12.11 dan 12 versi yang lebih tinggi, dan 11.16 dan versi 11 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6in.2xlarge	Ya	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua versi PostgreSQL 16 dan 15, 14.3 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, 12.11 dan 12 versi yang lebih tinggi, dan 11.16 dan versi 11 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6in.xlarge	Ya	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua versi PostgreSQL 16 dan 15, 14.3 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, 12.11 dan 12 versi yang lebih tinggi, dan 11.16 dan versi 11 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6in.large	Ya	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua versi PostgreSQL 16 dan 15, 14.3 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, 12.11 dan 12 versi yang lebih tinggi, dan 11.16 dan versi 11 yang lebih tinggi
db.m6i – kelas instans tujuan umum						
db.m6i.32xlarge	Ya	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.15 dan versi 10.5 yang lebih tinggi, dan 10.4.24 dan versi 10.4 yang lebih tinggi	Ya	MySQL versi 8.0.28 dan yang lebih tinggi	Oracle Databas 19c	Semua PostgreSQL 16, 15, dan 14 versi; 13.4, 12.8, dan 11.13 dan versi 11 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6i.24xlarge	Ya	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.15 dan versi 10.5 yang lebih tinggi, dan 10.4.24 dan versi 10.4 yang lebih tinggi	Ya	MySQL versi 8.0.28 dan yang lebih tinggi	Oracle Databas 19c	Semua PostgreSQ L 16, 15, dan 14 versi; 13.4, 12.8, dan 11.13 dan versi 11 yang lebih tinggi
db.m6i.16xlarge	Ya	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.15 dan versi 10.5 yang lebih tinggi, dan 10.4.24 dan versi 10.4 yang lebih tinggi	Ya	MySQL versi 8.0.28 dan yang lebih tinggi	Oracle Databas 19c	Semua PostgreSQ L 16, 15, dan 14 versi; 13.4, 12.8, dan 11.13 dan versi 11 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6i.12xlarge	Ya	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.15 dan versi 10.5 yang lebih tinggi, dan 10.4.24 dan versi 10.4 yang lebih tinggi	Ya	MySQL versi 8.0.28 dan yang lebih tinggi	Oracle Databas 19c	Semua PostgreSQ L 16, 15, dan 14 versi; 13.4, 12.8, dan 11.13 dan versi 11 yang lebih tinggi
db.m6i.8xlarge	Ya	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.15 dan versi 10.5 yang lebih tinggi, dan 10.4.24 dan versi 10.4 yang lebih tinggi	Ya	MySQL versi 8.0.28 dan yang lebih tinggi	Oracle Databas 19c	Semua PostgreSQ L 16, 15, dan 14 versi; 13.4, 12.8, dan 11.13 dan versi 11 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6i.4xlarge	Ya	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.15 dan versi 10.5 yang lebih tinggi, dan 10.4.24 dan versi 10.4 yang lebih tinggi	Ya	MySQL versi 8.0.28 dan yang lebih tinggi	Oracle Databas 19c	Semua PostgreSQ L 16, 15, dan 14 versi; 13.4, 12.8, dan 11.13 dan versi 11 yang lebih tinggi
db.m6i.2xlarge	Ya	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.15 dan versi 10.5 yang lebih tinggi, dan 10.4.24 dan versi 10.4 yang lebih tinggi	Ya	MySQL versi 8.0.28 dan yang lebih tinggi	Oracle Databas 19c	Semua PostgreSQ L 16, 15, dan 14 versi; 13.4, 12.8, dan 11.13 dan versi 11 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m6i.xlarge	Ya	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.15 dan versi 10.5 yang lebih tinggi, dan 10.4.24 dan versi 10.4 yang lebih tinggi	Ya	MySQL versi 8.0.28 dan yang lebih tinggi	Oracle Databas 19c	Semua PostgreSQ L 16, 15, dan 14 versi; 13.4, 12.8, dan 11.13 dan versi 11 yang lebih tinggi
db.m6i.large	Ya	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.15 dan versi 10.5 yang lebih tinggi, dan 10.4.24 dan versi 10.4 yang lebih tinggi	Ya	MySQL versi 8.0.28 dan yang lebih tinggi	Oracle Databas 19c	Semua PostgreSQ L 16, 15, dan 14 versi; 13.4, 12.8, dan 11.13 dan versi 11 yang lebih tinggi
db.m5d – kelas instans tujuan umum						

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m5d.24xlarge	Tidak	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Ya	MySQL 8.0.28 dan yang lebih tinggi	Ya	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, dan 13.4
db.m5d.16xlarge	Tidak	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Ya	MySQL 8.0.28 dan yang lebih tinggi	Ya	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, dan 13.4

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m5d.12xlarge	Tidak	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Ya	MySQL 8.0.28 dan yang lebih tinggi	Ya	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, dan 13.4
db.m5d.8xlarge	Tidak	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Ya	MySQL 8.0.28 dan yang lebih tinggi	Ya	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, dan 13.4

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m5d.4xlarge	Tidak	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Ya	MySQL 8.0.28 dan yang lebih tinggi	Ya	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, dan 13.4
db.m5d.2xlarge	Tidak	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Ya	MySQL 8.0.28 dan yang lebih tinggi	Ya	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, dan 13.4

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m5d.xlarge	Tidak	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Ya	MySQL 8.0.28 dan yang lebih tinggi	Ya	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, dan 13.4
db.m5d.large	Tidak	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Ya	MySQL 8.0.28 dan yang lebih tinggi	Ya	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, dan 13.4
db.m5 – kelas instans tujuan umum						

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m5.24xlarge	Tidak	Ya	Ya	Ya	Ya	Semua versi PostgreSQL 16, 15, 14, 13, 12, dan 11; 10.17 dan versi 10 yang lebih tinggi; dan 9.6.22 dan versi 9 yang lebih tinggi
db.m5.16xlarge	Tidak	Ya	Ya	Ya	Ya	Semua versi PostgreSQL 16, 15, 14, 13, 12, dan 11; 10.17 dan versi 10 yang lebih tinggi; dan 9.6.22 dan versi 9 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m5.12xlarge	Tidak	Ya	Ya	Ya	Ya	Semua versi PostgreSQL 16, 15, 14, 13, 12, dan 11; 10.17 dan versi 10 yang lebih tinggi; dan 9.6.22 dan versi 9 yang lebih tinggi
db.m5.8xlarge	Tidak	Ya	Ya	Ya	Ya	Semua versi PostgreSQL 16, 15, 14, 13, 12, dan 11; 10.17 dan versi 10 yang lebih tinggi; dan 9.6.22 dan versi 9 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m5.4xlarge	Tidak	Ya	Ya	Ya	Ya	Semua versi PostgreSQL 16, 15, 14, 13, 12, dan 11; 10.17 dan versi 10 yang lebih tinggi; dan 9.6.22 dan versi 9 yang lebih tinggi
db.m5.2xlarge	Tidak	Ya	Ya	Ya	Ya	Semua versi PostgreSQL 16, 15, 14, 13, 12, dan 11; 10.17 dan versi 10 yang lebih tinggi; dan 9.6.22 dan versi 9 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m5.xlarge	Tidak	Ya	Ya	Ya	Ya	Semua versi PostgreSQL 16, 15, 14, 13, 12, dan 11; 10.17 dan versi 10 yang lebih tinggi; dan 9.6.22 dan versi 9 yang lebih tinggi
db.m5.large	Tidak	Ya	Ya	Ya	Ya	Semua versi PostgreSQL 16, 15, 14, 13, 12, dan 11; 10.17 dan versi 10 yang lebih tinggi; dan 9.6.22 dan versi 9 yang lebih tinggi
db.m4 – kelas instans tujuan umum						
db.m4.16xlarge	Tidak	Semua versi MariaDB 10.6, 10.5, 10.4, dan 10.3	Ya	MySQL 8.0, 5.7	Dihentikan	Lebih rendah dari PostgreSQL 13

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m4.10xlarge	Tidak	Semua versi MariaDB 10.6, 10.5, 10.4, dan 10.3	Ya	Ya	Dihentikan	Lebih rendah dari PostgreSQL 13
db.m4.4xlarge	Tidak	Semua versi MariaDB 10.6, 10.5, 10.4, dan 10.3	Ya	Ya	Dihentikan	Lebih rendah dari PostgreSQL 13
db.m4.2xlarge	Tidak	Semua versi MariaDB 10.6, 10.5, 10.4, dan 10.3	Ya	Ya	Dihentikan	Lebih rendah dari PostgreSQL 13
db.m4.xlarge	Tidak	Semua versi MariaDB 10.6, 10.5, 10.4, dan 10.3	Ya	Ya	Dihentikan	Lebih rendah dari PostgreSQL 13
db.m4.large	Tidak	Semua versi MariaDB 10.6, 10.5, 10.4, dan 10.3	Ya	Ya	Dihentikan	Lebih rendah dari PostgreSQL 13
db.m3 – kelas instans tujuan umum						
db.m3.2xlarge	Tidak	Tidak	Ya	Ya	Dihentikan	Dihentikan
db.m3.xlarge	Tidak	Tidak	Ya	Ya	Dihentikan	Dihentikan
db.m3.large	Tidak	Tidak	Ya	Ya	Dihentikan	Dihentikan

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m3.medium	Tidak	Tidak	Ya	Ya	Dihentikan	Dihentikan
db.x2g — kelas instance yang dioptimalkan untuk memori yang didukung oleh prosesor Graviton2 AWS						
db.x2g.16xlarge	Tidak	Semua versi MariaDB 10.11, 10.6, 10.5, dan 10.4	Tidak	MySQL 8.0.25 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi
db.x2g.12xlarge	Tidak	Semua versi MariaDB 10.11, 10.6, 10.5, dan 10.4	Tidak	MySQL 8.0.25 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi
db.x2g.8xlarge	Tidak	Semua versi MariaDB 10.11, 10.6, 10.5, dan 10.4	Tidak	MySQL 8.0.25 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x2g.4xlarge	Tidak	Semua versi MariaDB 10.11, 10.6, 10.5, dan 10.4	Tidak	MySQL 8.0.25 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi
db.x2g.2xlarge	Tidak	Semua versi MariaDB 10.11, 10.6, 10.5, dan 10.4	Tidak	MySQL 8.0.25 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi
db.x2g.xlarge	Tidak	Semua versi MariaDB 10.11, 10.6, 10.5, dan 10.4	Tidak	MySQL 8.0.25 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi
db.x2g.large	Tidak	Semua versi MariaDB 10.11, 10.6, 10.5, dan 10.4	Tidak	MySQL 8.0.25 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x2idn – kelas instans dengan memori yang dioptimalkan yang didukung oleh prosesor Intel Xeon Scalable generasi ke-3						
db.x2idn.32xlarge	Tidak	Semua versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Khusus Enterprise Edition	PostgreSQL 15 versi, 14.6, dan 13.9
db.x2idn.24xlarge	Tidak	Semua versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Khusus Enterprise Edition	PostgreSQL 15 versi, 14.6, dan 13.9

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x2idn.16xlarge	Tidak	Semua versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Khusus Enterprise Edition	PostgreSQL 15 versi, 14.6, dan 13.9
db.x2iedn – kelas instans yang memorinya dioptimalkan dengan SSD berbasis NVMe lokal, didukung oleh prosesor Intel Xeon Scalable generasi ke-3						
db.x2iedn.32xlarge	Ya	Semua versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Khusus Enterprise dan Standard Edition, SQL Server 2014 12.00 dan yang lebih tinggi	MySQL 8.0.28 dan yang lebih tinggi	Khusus Enterprise Edition	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, dan 13.4

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x2iedn.24xlarge	Ya	Semua versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Khusus Enterprise dan Standard Edition, SQL Server 2014 12.00 dan yang lebih tinggi	MySQL 8.0.28 dan yang lebih tinggi	Khusus Enterprise Edition	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, dan 13.4
db.x2iedn.16xlarge	Ya	Semua versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Khusus Enterprise dan Standard Edition, SQL Server 2014 12.00 dan yang lebih tinggi	MySQL 8.0.28 dan yang lebih tinggi	Khusus Enterprise Edition	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, dan 13.4

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x2iedn.8xlarge	Ya	Semua versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Khusus Enterprise dan Standard Edition, SQL Server 2014 12.00 dan yang lebih tinggi	MySQL 8.0.28 dan yang lebih tinggi	Khusus Enterprise Edition	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, dan 13.4
db.x2iedn.4xlarge	Ya	Semua versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Khusus Enterprise dan Standard Edition, SQL Server 2014 12.00 dan yang lebih tinggi	MySQL 8.0.28 dan yang lebih tinggi	Enterprise Edition dan Standard Edition 2 (SE2)	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, dan 13.4

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x2iedn.2xlarge	Ya	Semua versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Khusus Enterprise dan Standard Edition, SQL Server 2014 12.00 dan yang lebih tinggi	MySQL 8.0.28 dan yang lebih tinggi	Enterprise Edition dan Standard Edition 2 (SE2)	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, dan 13.4
db.x2iedn.xlarge	Ya	Semua versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Khusus Enterprise dan Standard Edition, SQL Server 2014 12.00 dan yang lebih tinggi	MySQL 8.0.28 dan yang lebih tinggi	Enterprise Edition dan Standard Edition 2 (SE2)	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, dan 13.4
db.x2iezn – kelas instans dengan memori yang dioptimalkan yang didukung oleh prosesor Intel Xeon Scalable generasi ke-2						

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x2iezn.12xlarge	Tidak	Tidak	Tidak	Tidak	Khusus Enterprise Edition	Tidak
db.x2iezn.8xlarge	Tidak	Tidak	Tidak	Tidak	Khusus Enterprise Edition	Tidak
db.x2iezn.6xlarge	Tidak	Tidak	Tidak	Tidak	Khusus Enterprise Edition	Tidak
db.x2iezn.4xlarge	Tidak	Tidak	Tidak	Tidak	Enterprise Edition dan Standard Edition 2 (SE2)	Tidak
db.x2iezn.2xlarge	Tidak	Tidak	Tidak	Tidak	Enterprise Edition dan Standard Edition 2 (SE2)	Tidak
db.z1d – kelas instans dengan memori yang dioptimalkan						

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.z1d.12xlarge	Tidak	Tidak	Ya	Tidak	Ya	Tidak
db.z1d.6xlarge	Tidak	Tidak	Ya	Tidak	Ya	Tidak
db.z1d.3xlarge	Tidak	Tidak	Ya	Tidak	Ya	Tidak
db.z1d.2xlarge	Tidak	Tidak	Ya	Tidak	Ya	Tidak
db.z1d.xlarge	Tidak	Tidak	Ya	Tidak	Ya	Tidak
db.z1d.large	Tidak	Tidak	Ya	Tidak	Ya	Tidak
db.x1e – kelas instan dengan memori yang dioptimalkan						
db.x1e.32xlarge	Tidak	Tidak	Ya	Tidak	Ya	Tidak
db.x1e.16xlarge	Tidak	Tidak	Ya	Tidak	Ya	Tidak
db.x1e.8xlarge	Tidak	Tidak	Ya	Tidak	Ya	Tidak
db.x1e.4xlarge	Tidak	Tidak	Ya	Tidak	Ya	Tidak
db.x1e.2xlarge	Tidak	Tidak	Ya	Tidak	Ya	Tidak
db.x1e.xlarge	Tidak	Tidak	Ya	Tidak	Ya	Tidak
db.x1 – kelas instans dengan memori yang dioptimalkan						
db.x1.32xlarge	Tidak	Tidak	Ya	Tidak	Ya	Tidak
db.x1.16xlarge	Tidak	Tidak	Ya	Tidak	Ya	Tidak
db.r7g — kelas instance yang dioptimalkan untuk memori yang didukung oleh prosesor Graviton3 AWS						

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r7g.16xlarge	Tidak	Versi MariaDB 10.11, 10.6.10 dan versi 10.6 yang lebih tinggi, 10.5.17 dan versi 10.5 yang lebih tinggi, dan 10.4.26 dan versi 10.4 yang lebih tinggi	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.4 dan 13 versi yang lebih tinggi
db.r7g.12xlarge	Tidak	Versi MariaDB 10.11, 10.6.10 dan versi 10.6 yang lebih tinggi, 10.5.17 dan versi 10.5 yang lebih tinggi, dan 10.4.26 dan versi 10.4 yang lebih tinggi	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.4 dan 13 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r7g.8xlarge	Tidak	Versi MariaDB 10.11, 10.6.10 dan versi 10.6 yang lebih tinggi, 10.5.17 dan versi 10.5 yang lebih tinggi, dan 10.4.26 dan versi 10.4 yang lebih tinggi	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.4 dan 13 versi yang lebih tinggi
db.r7g.4xlarge	Tidak	Versi MariaDB 10.11, 10.6.10 dan versi 10.6 yang lebih tinggi, 10.5.17 dan versi 10.5 yang lebih tinggi, dan 10.4.26 dan versi 10.4 yang lebih tinggi	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.4 dan 13 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r7g.2xlarge	Tidak	Versi MariaDB 10.11, 10.6.10 dan versi 10.6 yang lebih tinggi, 10.5.17 dan versi 10.5 yang lebih tinggi, dan 10.4.26 dan versi 10.4 yang lebih tinggi	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.4 dan 13 versi yang lebih tinggi
db.r7g.xlarge	Tidak	Versi MariaDB 10.11, 10.6.10 dan versi 10.6 yang lebih tinggi, 10.5.17 dan versi 10.5 yang lebih tinggi, dan 10.4.26 dan versi 10.4 yang lebih tinggi	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.4 dan 13 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r7g.large	Tidak	Versi MariaDB 10.11, 10.6.10 dan versi 10.6 yang lebih tinggi, 10.5.17 dan versi 10.5 yang lebih tinggi, dan 10.4.26 dan versi 10.4 yang lebih tinggi	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.4 dan 13 versi yang lebih tinggi
db.r6g — kelas instance yang dioptimalkan untuk memori yang didukung oleh prosesor Graviton2 AWS						
db.r6g.16xlarge	Tidak	Semua versi MariaDB 10.11, 10.6, 10.5, dan 10.4	Tidak	MySQL 8.0.23 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi
db.r6g.12xlarge	Tidak	Semua versi MariaDB 10.11, 10.6, 10.5, dan 10.4	Tidak	MySQL 8.0.23 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6g.8xlarge	Tidak	Semua versi MariaDB 10.11, 10.6, 10.5, dan 10.4	Tidak	MySQL 8.0.23 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi
db.r6g.4xlarge	Tidak	Semua versi MariaDB 10.11, 10.6, 10.5, dan 10.4	Tidak	MySQL 8.0.23 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi
db.r6g.2xlarge	Tidak	Semua versi MariaDB 10.11, 10.6, 10.5, dan 10.4	Tidak	MySQL 8.0.23 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi
db.r6g.xlarge	Tidak	Semua versi MariaDB 10.11, 10.6, 10.5, dan 10.4	Tidak	MySQL 8.0.23 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6g.large	Tidak	Semua versi MariaDB 10.11, 10.6, 10.5, dan 10.4	Tidak	MySQL 8.0.23 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi
db.r6gd - kelas instance yang dioptimalkan untuk memori yang didukung oleh prosesor Graviton2 AWS						
db.r6gd.16xlarge	Tidak	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, dan 13.4

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6gd.12xlarge	Tidak	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, dan 13.4
db.r6gd.8xlarge	Tidak	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, dan 13.4

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6gd.4xlarge	Tidak	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, dan 13.4
db.r6gd.2xlarge	Tidak	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, dan 13.4

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6gd.xlarge	Tidak	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, dan 13.4
db.r6gd.large	Tidak	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, dan 13.4
db.r6id – kelas instans dengan memori yang dioptimalkan yang didukung oleh prosesor Intel Xeon Scalable generasi ke-3						

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6id.32xlarge	Tidak	MariaDB 10.6.10 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi
db.r6id.24xlarge	Tidak	MariaDB 10.6.10 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6id.16xlarge	Tidak	MariaDB 10.6.10 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi
db.r6id.12xlarge	Tidak	MariaDB 10.6.10 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6id.8xlarge	Tidak	MariaDB 10.6.10 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi
db.r6id.4xlarge	Tidak	MariaDB 10.6.10 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6id.2xlarge	Tidak	MariaDB 10.6.10 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi
db.r6id.xlarge	Tidak	MariaDB 10.6.10 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6id.large	Tidak	MariaDB 10.6.10 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi
db.r6idn – kelas instans dengan memori yang dioptimalkan yang didukung oleh prosesor Intel Xeon Scalable generasi ke-3						
db.r6idn.32xlarge	Ya	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6idn.24xlarge	Ya	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi
db.r6idn.16xlarge	Ya	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6idn.12xlarge	Ya	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi
db.r6idn.8xlarge	Ya	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6idn.4xlarge	Ya	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi
db.r6idn.2xlarge	Ya	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6idn.xlarge	Ya	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi
db.r6idn.large	Ya	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, dan 13.7 dan 13 versi yang lebih tinggi
db.r6in – kelas instans dengan memori yang dioptimalkan yang didukung oleh prosesor Intel Xeon Scalable generasi ke-3						

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6in.32xlarge	Ya	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua versi PostgreSQL 16 dan 15, 14.3 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, 12.11 dan 12 versi yang lebih tinggi, dan 11.16 dan versi 11 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6in.24xlarge	Ya	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua versi PostgreSQL 16 dan 15, 14.3 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, 12.11 dan 12 versi yang lebih tinggi, dan 11.16 dan versi 11 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6in.16xlarge	Ya	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua versi PostgreSQL 16 dan 15, 14.3 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, 12.11 dan 12 versi yang lebih tinggi, dan 11.16 dan versi 11 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6in.12xlarge	Ya	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua versi PostgreSQL 16 dan 15, 14.3 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, 12.11 dan 12 versi yang lebih tinggi, dan 11.16 dan versi 11 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6in.8xlarge	Ya	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua versi PostgreSQL 16 dan 15, 14.3 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, 12.11 dan 12 versi yang lebih tinggi, dan 11.16 dan versi 11 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6in.4xlarge	Ya	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua versi PostgreSQL 16 dan 15, 14.3 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, 12.11 dan 12 versi yang lebih tinggi, dan 11.16 dan versi 11 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6in.2xlarge	Ya	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua versi PostgreSQL 16 dan 15, 14.3 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, 12.11 dan 12 versi yang lebih tinggi, dan 11.16 dan versi 11 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6in.xlarge	Ya	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua versi PostgreSQL 16 dan 15, 14.3 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, 12.11 dan 12 versi yang lebih tinggi, dan 11.16 dan versi 11 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6in.large	Ya	MariaDB versi 10.6.8 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Tidak	MySQL versi 8.0.28 dan yang lebih tinggi	Tidak	Semua versi PostgreSQL 16 dan 15, 14.3 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, 12.11 dan 12 versi yang lebih tinggi, dan 11.16 dan versi 11 yang lebih tinggi
db.r6i – kelas instans dengan memori yang dioptimalkan						

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6i.32xlarge	Ya	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.15 dan versi 10.5 yang lebih tinggi, dan 10.4.24 dan versi 10.4 yang lebih tinggi	Ya	MySQL versi 8.0.28 dan yang lebih tinggi	Ya	Semua versi PostgreSQL 16, 15, dan 14, 13.4 dan 13 versi yang lebih tinggi, 12.8 dan 12 versi yang lebih tinggi, 11.13 dan versi 11 yang lebih tinggi, dan 10.21 dan versi 10 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6i.24xlarge	Ya	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.15 dan versi 10.5 yang lebih tinggi, dan 10.4.24 dan versi 10.4 yang lebih tinggi	Ya	MySQL versi 8.0.28 dan yang lebih tinggi	Ya	Semua versi PostgreSQL 16, 15, dan 14, 13.4 dan 13 versi yang lebih tinggi, 12.8 dan 12 versi yang lebih tinggi, 11.13 dan versi 11 yang lebih tinggi, dan 10.21 dan versi 10 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6i.16xlarge	Ya	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.15 dan versi 10.5 yang lebih tinggi, dan 10.4.24 dan versi 10.4 yang lebih tinggi	Ya	MySQL versi 8.0.28 dan yang lebih tinggi	Ya	Semua versi PostgreSQL 16, 15, dan 14, 13.4 dan 13 versi yang lebih tinggi, 12.8 dan 12 versi yang lebih tinggi, 11.13 dan versi 11 yang lebih tinggi, dan 10.21 dan versi 10 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6i.12xlarge	Ya	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.15 dan versi 10.5 yang lebih tinggi, dan 10.4.24 dan versi 10.4 yang lebih tinggi	Ya	MySQL versi 8.0.28 dan yang lebih tinggi	Ya	Semua versi PostgreSQL 16, 15, dan 14, 13.4 dan 13 versi yang lebih tinggi, 12.8 dan 12 versi yang lebih tinggi, 11.13 dan versi 11 yang lebih tinggi, dan 10.21 dan versi 10 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6i.8xlarge	Ya	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.15 dan versi 10.5 yang lebih tinggi, dan 10.4.24 dan versi 10.4 yang lebih tinggi	Ya	MySQL versi 8.0.28 dan yang lebih tinggi	Ya	Semua versi PostgreSQL 16, 15, dan 14, 13.4 dan 13 versi yang lebih tinggi, 12.8 dan 12 versi yang lebih tinggi, 11.13 dan versi 11 yang lebih tinggi, dan 10.21 dan versi 10 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6i.4xlarge	Ya	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.15 dan versi 10.5 yang lebih tinggi, dan 10.4.24 dan versi 10.4 yang lebih tinggi	Ya	MySQL versi 8.0.28 dan yang lebih tinggi	Ya	Semua versi PostgreSQL 16, 15, dan 14, 13.4 dan 13 versi yang lebih tinggi, 12.8 dan 12 versi yang lebih tinggi, 11.13 dan versi 11 yang lebih tinggi, dan 10.21 dan versi 10 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6i.2xlarge	Ya	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.15 dan versi 10.5 yang lebih tinggi, dan 10.4.24 dan versi 10.4 yang lebih tinggi	Ya	MySQL versi 8.0.28 dan yang lebih tinggi	Ya	Semua versi PostgreSQL 16, 15, dan 14, 13.4 dan 13 versi yang lebih tinggi, 12.8 dan 12 versi yang lebih tinggi, 11.13 dan versi 11 yang lebih tinggi, dan 10.21 dan versi 10 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6i.xlarge	Ya	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.15 dan versi 10.5 yang lebih tinggi, dan 10.4.24 dan versi 10.4 yang lebih tinggi	Ya	MySQL versi 8.0.28 dan yang lebih tinggi	Ya	Semua versi PostgreSQL 16, 15, dan 14, 13.4 dan 13 versi yang lebih tinggi, 12.8 dan 12 versi yang lebih tinggi, 11.13 dan versi 11 yang lebih tinggi, dan 10.21 dan versi 10 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6i.large	Ya	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.15 dan versi 10.5 yang lebih tinggi, dan 10.4.24 dan versi 10.4 yang lebih tinggi	Ya	MySQL versi 8.0.28 dan yang lebih tinggi	Ya	Semua versi PostgreSQL 16, 15, dan 14, 13.4 dan 13 versi yang lebih tinggi, 12.8 dan 12 versi yang lebih tinggi, 11.13 dan versi 11 yang lebih tinggi, dan 10.21 dan versi 10 yang lebih tinggi
db.r5d – kelas instans dengan memori yang dioptimalkan						
db.r5d.24xlarge	Tidak	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Ya	MySQL 8.0.28 dan yang lebih tinggi	Ya	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, dan 13.4

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5d.16xlarge	Tidak	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Ya	MySQL 8.0.28 dan yang lebih tinggi	Ya	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, dan 13.4
db.r5d.12xlarge	Tidak	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Ya	MySQL 8.0.28 dan yang lebih tinggi	Ya	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, dan 13.4

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5d.8xlarge	Tidak	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Ya	MySQL 8.0.28 dan yang lebih tinggi	Ya	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, dan 13.4
db.r5d.4xlarge	Tidak	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Ya	MySQL 8.0.28 dan yang lebih tinggi	Ya	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, dan 13.4

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5d.2xlarge	Tidak	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Ya	MySQL 8.0.28 dan yang lebih tinggi	Ya	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, dan 13.4
db.r5d.xlarge	Tidak	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Ya	MySQL 8.0.28 dan yang lebih tinggi	Ya	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, dan 13.4

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5d.large	Tidak	Versi MariaDB 10.11, 10.6.7 dan versi 10.6 yang lebih tinggi, 10.5.16 dan versi 10.5 yang lebih tinggi, dan 10.4.25 dan versi 10.4 yang lebih tinggi	Ya	MySQL 8.0.28 dan yang lebih tinggi	Ya	Semua PostgreSQL 16 dan 15 versi, 14.5 dan 14 versi yang lebih tinggi, 13.7 dan 13 versi yang lebih tinggi, dan 13.4
db.r5b – kelas instans dengan memori dioptimalkan yang telah dikonfigurasikan sebelumnya untuk penyimpanan, I/O, dan memori tinggi						
db.r5b.8xlarge.tpc2.mem3x	Tidak	Tidak	Tidak	Tidak	Ya	Tidak
db.r5b.6xlarge.tpc2.mem4x	Tidak	Tidak	Tidak	Tidak	Ya	Tidak
db.r5b.4xlarge.tpc2.mem4x	Tidak	Tidak	Tidak	Tidak	Ya	Tidak
db.r5b.4xlarge.tpc2.mem3x	Tidak	Tidak	Tidak	Tidak	Ya	Tidak
db.r5b.4xlarge.tpc2.mem2x	Tidak	Tidak	Tidak	Tidak	Ya	Tidak
db.r5b.2xlarge.tpc2.mem8x	Tidak	Tidak	Tidak	Tidak	Ya	Tidak
db.r5b.2xlarge.tpc2.mem4x	Tidak	Tidak	Tidak	Tidak	Ya	Tidak

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5b.2xlarge.tpc1.mem2x	Tidak	Tidak	Tidak	Tidak	Ya	Tidak
db.r5b.xlarge.tpc2.mem4x	Tidak	Tidak	Tidak	Tidak	Ya	Tidak
db.r5b.xlarge.tpc2.mem2x	Tidak	Tidak	Tidak	Tidak	Ya	Tidak
db.r5b.large.tpc1.mem2x	Tidak	Tidak	Tidak	Tidak	Ya	Tidak
db.r5b – kelas instans dengan memori yang dioptimalkan						
db.r5b.24xlarge	Tidak	Versi MariaDB 10.11, 10.6.5 dan versi 10.6 yang lebih tinggi, 10.5.12 dan versi 10.5 yang lebih tinggi, 10.4.24 dan versi 10.4 yang lebih tinggi, dan 10.3.34 dan versi 10.3 yang lebih tinggi	Ya	MySQL 8.0.25 dan yang lebih tinggi	Ya	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5b.16xlarge	Tidak	Versi MariaDB 10.11, 10.6.5 dan versi 10.6 yang lebih tinggi, 10.5.12 dan versi 10.5 yang lebih tinggi, 10.4.24 dan versi 10.4 yang lebih tinggi, dan 10.3.34 dan versi 10.3 yang lebih tinggi	Ya	MySQL 8.0.25 dan yang lebih tinggi	Ya	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi
db.r5b.12xlarge	Tidak	Versi MariaDB 10.11, 10.6.5 dan versi 10.6 yang lebih tinggi, 10.5.12 dan versi 10.5 yang lebih tinggi, 10.4.24 dan versi 10.4 yang lebih tinggi, dan 10.3.34 dan versi 10.3 yang lebih tinggi	Ya	MySQL 8.0.25 dan yang lebih tinggi	Ya	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5b.8xlarge	Tidak	Versi MariaDB 10.11, 10.6.5 dan versi 10.6 yang lebih tinggi, 10.5.12 dan versi 10.5 yang lebih tinggi, 10.4.24 dan versi 10.4 yang lebih tinggi, dan 10.3.34 dan versi 10.3 yang lebih tinggi	Ya	MySQL 8.0.25 dan yang lebih tinggi	>Ya	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi
db.r5b.4xlarge	Tidak	Versi MariaDB 10.11, 10.6.5 dan versi 10.6 yang lebih tinggi, 10.5.12 dan versi 10.5 yang lebih tinggi, 10.4.24 dan versi 10.4 yang lebih tinggi, dan 10.3.34 dan versi 10.3 yang lebih tinggi	Ya	MySQL 8.0.25 dan yang lebih tinggi	Ya	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5b.2xlarge	Tidak	Versi MariaDB 10.11, 10.6.5 dan versi 10.6 yang lebih tinggi, 10.5.12 dan versi 10.5 yang lebih tinggi, 10.4.24 dan versi 10.4 yang lebih tinggi, dan 10.3.34 dan versi 10.3 yang lebih tinggi	Ya	MySQL 8.0.25 dan yang lebih tinggi	Ya	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi
db.r5b.xlarge	Tidak	Versi MariaDB 10.11, 10.6.5 dan versi 10.6 yang lebih tinggi, 10.5.12 dan versi 10.5 yang lebih tinggi, 10.4.24 dan versi 10.4 yang lebih tinggi, dan 10.3.34 dan versi 10.3 yang lebih tinggi	Ya	MySQL 8.0.25 dan yang lebih tinggi	Ya	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5b.large	Tidak	Versi MariaDB 10.11, 10.6.5 dan versi 10.6 yang lebih tinggi, 10.5.12 dan versi 10.5 yang lebih tinggi, 10.4.24 dan versi 10.4 yang lebih tinggi, dan 10.3.34 dan versi 10.3 yang lebih tinggi	Ya	MySQL 8.0.25 dan yang lebih tinggi	Ya	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi
db.r5 – kelas instans dengan memori dioptimalkan yang telah dikonfigurasikan sebelumnya untuk penyimpanan, I/O, dan memori tinggi						
db.r5.12xlarge.tpc2.mem2x	Tidak	Tidak	Tidak	Tidak	Ya	Tidak
db.r5.8xlarge.tpc2.mem3x	Tidak	Tidak	Tidak	Tidak	Ya	Tidak
db.r5.6xlarge.tpc2.mem4x	Tidak	Tidak	Tidak	Tidak	Ya	Tidak
db.r5.4xlarge.tpc2.mem4x	Tidak	Tidak	Tidak	Tidak	Ya	Tidak
db.r5.4xlarge.tpc2.mem3x	Tidak	Tidak	Tidak	Tidak	Ya	Tidak

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5.4xlarge.tpc2.mem2x	Tidak	Tidak	Tidak	Tidak	Ya	Tidak
db.r5.2xlarge.tpc2.mem8x	Tidak	Tidak	Tidak	Tidak	Ya	Tidak
db.r5.2xlarge.tpc2.mem4x	Tidak	Tidak	Tidak	Tidak	Ya	Tidak
db.r5.2xlarge.tpc1.mem2x	Tidak	Tidak	Tidak	Tidak	Ya	Tidak
db.r5.xlarge.tpc2.mem4x	Tidak	Tidak	Tidak	Tidak	Ya	Tidak
db.r5.xlarge.tpc2.mem2x	Tidak	Tidak	Tidak	Tidak	Ya	Tidak
db.r5.large.tpc1.mem2x	Tidak	Tidak	Tidak	Tidak	Ya	Tidak
db.r5 – kelas instans dengan memori yang dioptimalkan						
db.r5.24xlarge	Tidak	Ya	Ya	Ya	Ya	Semua versi PostgreSQL 16, 15, 14, 13, 12, dan 11; 10.17 dan versi 10 yang lebih tinggi; dan 9.6.22 dan versi 9 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5.16xlarge	Tidak	Ya	Ya	Ya	Ya	Semua versi PostgreSQL 16, 15, 14, 13, 12, dan 11; 10.17 dan versi 10 yang lebih tinggi; dan 9.6.22 dan versi 9 yang lebih tinggi
db.r5.12xlarge	Tidak	Ya	Ya	Ya	Ya	Semua versi PostgreSQL 16, 15, 14, 13, 12, dan 11; 10.17 dan versi 10 yang lebih tinggi; dan 9.6.22 dan versi 9 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5.8xlarge	Tidak	Ya	Ya	Ya	Ya	Semua versi PostgreSQL 16, 15, 14, 13, 12, dan 11; 10.17 dan versi 10 yang lebih tinggi; dan 9.6.22 dan versi 9 yang lebih tinggi
db.r5.4xlarge	Tidak	Ya	Ya	Ya	Ya	Semua versi PostgreSQL 16, 15, 14, 13, 12, dan 11; 10.17 dan versi 10 yang lebih tinggi; dan 9.6.22 dan versi 9 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5.2xlarge	Tidak	Ya	Ya	Ya	Ya	Semua versi PostgreSQL 16, 15, 14, 13, 12, dan 11; 10.17 dan versi 10 yang lebih tinggi; dan 9.6.22 dan versi 9 yang lebih tinggi
db.r5.xlarge	Tidak	Ya	Ya	Ya	Ya	Semua versi PostgreSQL 16, 15, 14, 13, 12, dan 11; 10.17 dan versi 10 yang lebih tinggi; dan 9.6.22 dan versi 9 yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5.large	Tidak	Ya	Ya	Ya	Ya	Semua versi PostgreSQL 16, 15, 14, 13, 12, dan 11; 10.17 dan versi 10 yang lebih tinggi; dan 9.6.22 dan versi 9 yang lebih tinggi
db.r4 – kelas instans dengan memori yang dioptimalkan						
db.r4.16xlarge	Tidak	Semua versi MariaDB 10.6, 10.5, 10.4, dan 10.3	Ya	Semua MySQL 8.0, 5.7	Dihentikan	Lebih rendah dari PostgreSQL 13
db.r4.8xlarge	Tidak	Semua versi MariaDB 10.6, 10.5, 10.4, dan 10.3	Ya	Semua MySQL 8.0, 5.7	Dihentikan	Lebih rendah dari PostgreSQL 13
db.r4.4xlarge	Tidak	Semua versi MariaDB 10.6, 10.5, 10.4, dan 10.3	Ya	Semua MySQL 8.0, 5.7	Dihentikan	Lebih rendah dari PostgreSQL 13
db.r4.2xlarge	Tidak	Semua versi MariaDB 10.6, 10.5, 10.4, dan 10.3	Ya	Semua MySQL 8.0, 5.7	Dihentikan	Lebih rendah dari PostgreSQL 13

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r4.xlarge	Tidak	Semua versi MariaDB 10.6, 10.5, 10.4, dan 10.3	Ya	Semua MySQL 8.0, 5.7	Dihentikan	Lebih rendah dari PostgreSQL 13
db.r4.large	Tidak	Semua versi MariaDB 10.6, 10.5, 10.4, dan 10.3	Ya	Semua MySQL 8.0, 5.7	Dihentikan	Lebih rendah dari PostgreSQL 13
db.r3 – kelas instans dengan memori yang dioptimalkan						
db.r3.8xlarge**	Tidak	Semua versi MariaDB 10.6, 10.5, 10.4, dan 10.3	Ya	Ya	Dihentikan	Dihentikan
db.r3.4xlarge	Tidak	Semua versi MariaDB 10.6, 10.5, 10.4, dan 10.3	Ya	Ya	Dihentikan	Dihentikan
db.r3.2xlarge	Tidak	Semua versi MariaDB 10.6, 10.5, 10.4, dan 10.3	Ya	Ya	Dihentikan	Dihentikan
db.r3.xlarge	Tidak	Semua versi MariaDB 10.6, 10.5, 10.4, dan 10.3	Ya	Ya	Dihentikan	Dihentikan

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r3.large	Tidak	Semua versi MariaDB 10.6, 10.5, 10.4, dan 10.3	Ya	Ya	Dihentikan	Dihentikan
db.c6gd — kelas instans yang dioptimalkan komputasi (hanya untuk penerapan klaster DB multi-AZ)						
db.c6gd.16xlarge	Tidak	Tidak	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi; 14.5 dan 14 versi yang lebih tinggi; 13.4 dan 13.7 dan 13 versi yang lebih tinggi
db.c6gd.12xlarge	Tidak	Tidak	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi; 14.5 dan 14 versi yang lebih tinggi; 13.4 dan 13.7 dan 13 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.c6gd.8xlarge	Tidak	Tidak	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi; 14.5 dan 14 versi yang lebih tinggi; 13.4 dan 13.7 dan 13 versi yang lebih tinggi
db.c6gd.4xlarge	Tidak	Tidak	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi; 14.5 dan 14 versi yang lebih tinggi; 13.4 dan 13.7 dan 13 versi yang lebih tinggi
db.c6gd.2xlarge	Tidak	Tidak	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi; 14.5 dan 14 versi yang lebih tinggi; 13.4 dan 13.7 dan 13 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.c6gd.xlarge	Tidak	Tidak	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi; 14.5 dan 14 versi yang lebih tinggi; 13.4 dan 13.7 dan 13 versi yang lebih tinggi
db.c6gd.large	Tidak	Tidak	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi; 14.5 dan 14 versi yang lebih tinggi; 13.4 dan 13.7 dan 13 versi yang lebih tinggi
db.c6gd.sedang	Tidak	Tidak	Tidak	MySQL 8.0.28 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16 dan 15 versi; 14.5 dan 14 versi yang lebih tinggi; 13.4 dan 13.7 dan 13 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.t4g — kelas instance berkinerja pecah yang ditenagai oleh prosesor Graviton2 AWS						
db.t4g.2xlarge	Tidak	Semua versi MariaDB 10.11, 10.6, 10.5, dan 10.4	Tidak	MySQL 8.0.25 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi
db.t4g.xlarge	Tidak	Semua versi MariaDB 10.11, 10.6, 10.5, dan 10.4	Tidak	MySQL 8.0.25 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi
db.t4g.large	Tidak	Semua versi MariaDB 10.11, 10.6, 10.5, dan 10.4	Tidak	MySQL 8.0.25 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi
db.t4g.medium	Tidak	Semua versi MariaDB 10.11, 10.6, 10.5, dan 10.4	Tidak	MySQL 8.0.25 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.t4g.small	Tidak	Semua versi MariaDB 10.11, 10.6, 10.5, dan 10.4	Tidak	MySQL 8.0.25 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi
db.t4g.micro	Tidak	Semua versi MariaDB 10.11, 10.6, 10.5, dan 10.4	Tidak	MySQL 8.0.25 dan yang lebih tinggi	Tidak	Semua PostgreSQL 16, 15, 14, dan 13 versi; dan 12.7 dan 12 versi yang lebih tinggi
db.t3 – kelas instans performa yang dapat melonjak						
db.t3.2xlarge	Ya	Ya	Ya	Ya	Ya	Semua PostgreSQL 16, 15, 14, 13, 12, 11, dan 10 versi; dan 9.6.22 dan 9 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.t3.xlarge	Ya	Ya	Ya	Ya	Ya	Semua PostgreSQL 16, 15, 14, 13, 12, 11, dan 10 versi; dan 9.6.22 dan 9 versi yang lebih tinggi
db.t3.large	Ya	Ya	Ya	Ya	Ya	Semua PostgreSQL 16, 15, 14, 13, 12, 11, dan 10 versi; dan 9.6.22 dan 9 versi yang lebih tinggi
db.t3.medium	Ya	Ya	Ya	Ya	Ya	Semua PostgreSQL 16, 15, 14, 13, 12, 11, dan 10 versi; dan 9.6.22 dan 9 versi yang lebih tinggi

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.t3.small	Ya	Ya	Ya	Ya	Ya	Semua PostgreSQL 16, 15, 14, 13, 12, 11, dan 10 versi; dan 9.6.22 dan 9 versi yang lebih tinggi
db.t3.micro	Tidak	Ya	Tidak	Ya	Hanya di Oracle Database 12c Rilis 1 (12.1.0.2), yang telah dihentikan	Semua PostgreSQL 16, 15, 14, 13, 12, 11, dan 10 versi; dan 9.6.22 dan 9 versi yang lebih tinggi
db.t2 – kelas instans performa yang dapat melonjak						
db.t2.2xlarge	Tidak	Semua versi MariaDB 10.6, 10.5, 10.4, dan 10.3	Tidak	Semua MySQL 8.0, 5.7	Dihentikan	Lebih rendah dari PostgreSQL 13
db.t2.xlarge	Tidak	Semua versi MariaDB 10.6, 10.5, 10.4, dan 10.3	Tidak	Semua MySQL 8.0, 5.7	Dihentikan	Lebih rendah dari PostgreSQL 13

Kelas instans	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.t2.large	Tidak	Semua versi MariaDB 10.6, 10.5, 10.4, dan 10.3	Ya	Ya	Dihentikan	Lebih rendah dari PostgreSQL 13
db.t2.medium	Tidak	Semua versi MariaDB 10.6, 10.5, 10.4, dan 10.3	Ya	Ya	Dihentikan	Lebih rendah dari PostgreSQL 13
db.t2.small	Tidak	Semua versi MariaDB 10.6, 10.5, 10.4, dan 10.3	Ya	Ya	Dihentikan	Lebih rendah dari PostgreSQL 13
db.t2.micro	Tidak	Semua versi MariaDB 10.6, 10.5, 10.4, dan 10.3	Ya	Ya	Dihentikan	Lebih rendah dari PostgreSQL 13

Menentukan dukungan kelas instans DB di Wilayah AWS

Untuk menentukan kelas instans DB yang didukung oleh mesin DB di Wilayah AWS tertentu, Anda dapat menggunakan beberapa pendekatan. Anda dapat menggunakan halaman AWS Management Console, Amazon RDS Pricing, atau perintah describe-orderable-db-instance-options untuk AWS Command Line Interface ()AWS CLI.

> **Note**
>
> Ketika Anda melakukan operasi dengan AWS Management Console, secara otomatis menampilkan kelas instans DB yang didukung untuk mesin DB tertentu, versi mesin DB, dan Wilayah AWS. Contoh operasi yang dapat Anda lakukan termasuk membuat dan mengubah instans DB.

Daftar Isi

Menggunakan halaman harga Amazon RDS untuk menentukan dukungan kelas instans DB di Wilayah AWS

Anda dapat menggunakan halaman Harga Amazon RDS untuk menentukan kelas instans DB yang didukung oleh masing-masing mesin DB di Wilayah AWS tertentu.

Untuk menggunakan halaman harga guna menentukan kelas instans DB yang didukung oleh masing-masing mesin DB di sebuah Wilayah

1. Buka Harga Amazon RDS.
2. Di bagian Kalkulator Harga AWS untuk Amazon RDS, pilih Buat perkiraan kustom Anda sekarang.
3. Di Pilih Wilayah, pilih Wilayah AWS.
4. Di Temukan Layanan, masukkan **`Amazon RDS`**.
5. Pilih Konfigurasi untuk opsi konfigurasi dan mesin DB.
6. Gunakan bagian instans yang kompatibel untuk melihat kelas instans DB yang didukung.
7. (Opsional) Pilih opsi lain di kalkulator, lalu pilih Simpan dan lihat ringkasan atau Simpan dan tambahkan layanan.

Menggunakan AWS CLI untuk menentukan dukungan kelas instans DB di Wilayah AWS

Anda dapat menggunakan AWS CLI untuk menentukan kelas instans DB mana yang didukung untuk mesin DB tertentu dan versi mesin DB dalam file Wilayah AWS. Tabel berikut menunjukkan nilai-nilai mesin DB yang valid.

Nama mesin	Nilai-nilai mesin dalam perintah CLI	Informasi selengkapnya tentang versi
Db2	`db2-ae` `db2-se`	Versi-versi Db2 pada Amazon RDS
MariaDB	`mariadb`	Versi-versi MariaDB pada Amazon RDS

Microsoft SQL Server	`sqlserver-ee`
	`sqlserver-se`
	`sqlserver-ex`
	`sqlserver-web`
		Versi Microsoft SQL Server di Amazon RDS

MySQL	`mysql`	Versi MySQL di Amazon RDS
Oracle	`oracle-ee` `oracle-se2`	Catatan Rilis Amazon RDS for Oracle
PostgreSQL	`postgres`	Versi basis data PostgreSQL yang tersedia

Untuk informasi tentang Wilayah AWS nama, lihatAWS Daerah.

Contoh berikut menunjukkan bagaimana menentukan dukungan kelas instance DB dalam Wilayah AWS menggunakan AWS CLI perintah describe-orderable-db-instance-options.

> **Note**
>
> Untuk membatasi output, contoh-contoh ini hanya menunjukkan hasil untuk jenis penyimpanan SSD Tujuan Umum (`gp2`). Jika perlu, Anda dapat mengubah jenis penyimpanan menjadi SSD Tujuan Umum (`gp3`), IOPS yang Tersedia (`io1`), atau magnetik (`standard`) dalam perintah.

Topik

- Menyusun daftar kelas instans DB yang didukung oleh versi mesin DB tertentu di Wilayah AWS
- Menyusun daftar versi mesin DB yang mendukung kelas instans DB tertentu di Wilayah AWS

Menyusun daftar kelas instans DB yang didukung oleh versi mesin DB tertentu di Wilayah AWS

Untuk daftar kelas instans DB yang didukung oleh versi mesin DB tertentu dalam Wilayah AWS, jalankan perintah berikut.

Untuk Linux, macOS, atau Unix:

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

Untuk Windows:

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

Sebagai contoh, perintah berikut menyusun daftar kelas instans DB yang didukung untuk versi 13.6 mesin DB RDS untuk PostgreSQL di AS Timur (Virginia Utara).

Untuk Linux, macOS, atau Unix:

```
aws rds describe-orderable-db-instance-options --engine postgres --engine-version 15.4
 \
    --query "*[].{DBInstanceClass:DBInstanceClass,StorageType:StorageType}|[?
StorageType=='gp2']|[].{DBInstanceClass:DBInstanceClass}" \
    --output text \
    --region us-east-1
```

Untuk Windows:

```
aws rds describe-orderable-db-instance-options --engine postgres --engine-version 15.4
 ^
    --query "*[].{DBInstanceClass:DBInstanceClass,StorageType:StorageType}|[?
StorageType=='gp2']|[].{DBInstanceClass:DBInstanceClass}" ^
    --output text ^
    --region us-east-1
```

Menyusun daftar versi mesin DB yang mendukung kelas instans DB tertentu di Wilayah AWS

Untuk menyusun daftar versi mesin DB yang mendukung kelas instans DB tertentu di Wilayah AWS, jalankan perintah berikut.

Untuk Linux, macOS, atau Unix:

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

Untuk Windows:

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

Sebagai contoh, perintah berikut menyusun daftar versi mesin DB RDS for PostgreSQL yang mendukung kelas instans DB db.r5.large di AS Timur (Virginia Utara).

Untuk Linux, macOS, atau Unix:

```
aws rds describe-orderable-db-instance-options --engine postgres --db-instance-class
 db.m7g.large \
    --query "*[].{EngineVersion:EngineVersion,StorageType:StorageType}|[?
StorageType=='gp2']|[].{EngineVersion:EngineVersion}" \
    --output text \
    --region us-east-1
```

Untuk Windows:

```
aws rds describe-orderable-db-instance-options --engine postgres --db-instance-class
 db.m7g.large ^
    --query "*[].{EngineVersion:EngineVersion,StorageType:StorageType}|[?
StorageType=='gp2']|[].{EngineVersion:EngineVersion}" ^
    --output text ^
    --region us-east-1
```

Mengubah kelas instans DB

Anda dapat mengubah CPU dan memori yang tersedia menjadi instans DB dengan mengubah kelas instans DB-nya. Untuk mengubah kelas instans DB, modifikasi instans DB Anda dengan mengikuti petunjuk di Memodifikasi instans DB Amazon RDS.

Mengonfigurasi prosesor untuk kelas instans DB di RDS for Oracle

Kelas instans DB Amazon RDS mendukung Intel Hyper-Threading Technology, yang memungkinkan beberapa thread berjalan secara bersamaan di satu inti CPU Intel Xeon. Setiap thread direpresentasikan sebagai CPU virtual (vCPU) pada instans DB. Instans DB memiliki jumlah inti CPU default, yang bervariasi sesuai dengan kelas instans DB. Misalnya, kelas instans DB db.m4.xlarge memiliki dua inti CPU dan dua thread per ini secara default—total empat vCPU.

> **Note**
>
> Setiap vCPU adalah hyperthread inti CPU Intel Xeon.

Topik

- Gambaran umum konfigurasi prosesor untuk RDS for Oracle
- Kelas instans DB yang mendukung konfigurasi prosesor
- Mengatur inti CPU dan threads per inti CPU untuk kelas instans DB

Gambaran umum konfigurasi prosesor untuk RDS for Oracle

Ketika menggunakan RDS for Oracle, Anda biasanya dapat menemukan kelas instans DB yang memiliki kombinasi memori dan jumlah vCPU agar sesuai dengan beban kerja. Namun, Anda juga

dapat menentukan fitur prosesor berikut untuk mengoptimalkan instans DB RDS for Oracle untuk beban kerja atau kebutuhan bisnis tertentu:

- Jumlah inti CPU – Anda dapat menyesuaikan jumlah inti CPU untuk instans DB. Anda dapat melakukan ini agar dapat mengoptimalkan biaya lisensi perangkat lunak Anda dengan instans DB yang memiliki jumlah RAM yang cukup untuk beban kerja yang membutuhkan memori intensif tetapi dengan inti CPU yang lebih sedikit.
- Thread per inti – Anda dapat menonaktifkan Intel Hyper-Threading Technology dengan menentukan satu thread untuk setiap inti CPU. Anda dapat melakukannya untuk beban kerja tertentu, seperti beban kerja komputasi performa tinggi (HPC).

Anda dapat mengontrol jumlah inti dan thread CPU untuk setiap inti secara terpisah. Anda dapat mengatur salah satu atau keduanya dalam permintaan. Setelah pengaturan dikaitkan dengan instans DB, pengaturan akan berlanjut hingga Anda mengubahnya.

Pengaturan prosesor untuk instans DB dikaitkan dengan snapshot instans DB. Saat suatu snapshot dipulihkan, instans DB yang dipulihkan menggunakan pengaturan fitur prosesor yang digunakan saat snapshot diambil.

Jika Anda memodifikasi kelas instans DB untuk instans DB dengan pengaturan prosesor non-default, tentukan pengaturan prosesor default atau tentukan secara eksplisit pengaturan prosesor saat modifikasi. Persyaratan ini memastikan Anda mengetahui biaya lisensi pihak ketiga yang mungkin muncul ketika Anda memodifikasi instans DB.

Tidak ada penambahan atau pengurangan biaya untuk menentukan fitur prosesor pada instans DB RDS for Oracle. Anda dikenai biaya yang sama untuk instans DB yang diluncurkan dengan konfigurasi CPU default.

Kelas instans DB yang mendukung konfigurasi prosesor

Anda dapat mengonfigurasi jumlah inti dan thread CPU per inti hanya ketika syarat berikut terpenuhi:

- Anda sedang mengonfigurasi instans DB RDS for Oracle. Untuk informasi tentang kelas instans DB yang didukung oleh basis data Oracle yang berbeda, lihat Kelas instans RDS for Oracle.
- Instans DB Anda menggunakan opsi lisensi Bawa Lisensi Sendiri (BYOL) RDS for Oracle. Untuk informasi selengkapnya tentang opsi lisensi Oracle, lihat Opsi lisensi RDS for Oracle.
- Instans DB Anda bukan milik kelas instans db.r5 atau db.r5b yang memiliki konfigurasi prosesor yang telah ditetapkan. Kelas instans ini memiliki nama

dalam bentuk db.r5.*instance_size*.tpc*threads_per_core*.mem*ratio* atau db.r5b.*instance_size*.tpc*threads_per_core*.mem*ratio*. Misalnya, db.r5b.xlarge.tpc2.mem4x dikonfigurasi sebelumnya dengan 2 threads per inti (tpc2) dan 4x sebanyak memori kelas instans db.r5b.xlarge standar. Anda tidak dapat mengonfigurasi fitur prosesor dari kelas instans yang dioptimalkan ini. Untuk informasi selengkapnya, lihat Kelas instans RDS for Oracle yang didukung.

Pada tabel berikut, Anda dapat menemukan kelas instans DB yang mendukung pengaturan jumlah inti CPU dan thread CPU per inti. Anda juga dapat menemukan nilai default dan nilai yang valid untuk jumlah inti CPU dan thread per inti CPU untuk setiap kelas instans DB.

Kelas instans DB	vCPU default	Inti CPU default	Thread per inti default	Jumlah inti CPU yang valid	Jumlah thread per inti yang valid
db.m6i – kelas instans dengan memori yang dioptimalkan					
db.m6i.large	2	1	2	1	1, 2
db.m6i.xlarge	4	2	2	2	1, 2
db.m6i.2xlarge	8	4	2	2, 4	1, 2
db.m6i.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.m6i.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.m6i.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.m6i.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
db.m6i.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14,	1, 2

Kelas instans DB	vCPU default	Inti CPU default	Thread per inti default	Jumlah inti CPU yang valid	Jumlah thread per inti yang valid
				16, 18, 20, 22, 24, 26, 28, 30, 32	
db.m6i.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.m6i.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
db.m5 – kelas instans tujuan umum					
db.m5.large	2	1	2	1	1, 2
db.m5.xlarge	4	2	2	2	1, 2
db.m5.2xlarge	8	4	2	2, 4	1, 2
db.m5.4xlarge	16	8	2	2, 4, 6, 8	1, 2

Kelas instans DB	vCPU default	Inti CPU default	Thread per inti default	Jumlah inti CPU yang valid	Jumlah thread per inti yang valid
db.m5.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.m5.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
db.m5.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.m5.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.m5d – kelas instans tujuan umum					
db.m5d.large	2	1	2	1	1, 2
db.m5d.xlarge	4	2	2	2	1, 2
db.m5d.2xlarge	8	4	2	2, 4	1, 2
db.m5d.4xlarge	16	8	2	2, 4, 6, 8	1, 2

Kelas instans DB	vCPU default	Inti CPU default	Thread per inti default	Jumlah inti CPU yang valid	Jumlah thread per inti yang valid
db.m5d.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.m5d.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
db.m5d.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.m5d.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.m4 – kelas instans tujuan umum					
db.m4.10xlarge	40	20	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20	1, 2

Kelas instans DB	vCPU default	Inti CPU default	Thread per inti default	Jumlah inti CPU yang valid	Jumlah thread per inti yang valid
db.m4.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.r6i – kelas instans dengan memori yang dioptimalkan					
db.r6i.large	2	1	2	1	1, 2
db.r6i.xlarge	4	2	2	1, 2	1, 2
db.r6i.2xlarge	8	4	2	2, 4	1, 2
db.r6i.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.r6i.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.r6i.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
db.r6i.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2

Kelas instans DB	vCPU default	Inti CPU default	Thread per inti default	Jumlah inti CPU yang valid	Jumlah thread per inti yang valid
db.r6i.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.r6i.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
db.r5 – kelas instans dengan memori yang dioptimalkan					
db.r5.large	2	1	2	1	1, 2
db.r5.xlarge	4	2	2	2	1, 2
db.r5.2xlarge	8	4	2	2, 4	1, 2
db.r5.4xlarge	16	8	2	2, 4, 6, 8	1, 2

Kelas instans DB	vCPU default	Inti CPU default	Thread per inti default	Jumlah inti CPU yang valid	Jumlah thread per inti yang valid
db.r5.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.r5.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
db.r5.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.r5.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.r5 – kelas instans dengan memori yang dioptimalkan					
db.r5b.large	2	1	2	1	1, 2
db.r5b.xlarge	4	2	2	2	1, 2
db.r5b.2xlarge	8	4	2	2, 4	1, 2
db.r5b.4xlarge	16	8	2	2, 4, 6, 8	1, 2

Kelas instans DB	vCPU default	Inti CPU default	Thread per inti default	Jumlah inti CPU yang valid	Jumlah thread per inti yang valid
db.r5b.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.r5b.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
db.r5b.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.r5b.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.r5d – kelas instans dengan memori yang dioptimalkan					
db.r5d.large	2	1	2	1	1, 2
db.r5d.xlarge	4	2	2	2	1, 2
db.r5d.2xlarge	8	4	2	2, 4	1, 2
db.r5d.4xlarge	16	8	2	2, 4, 6, 8	1, 2

Kelas instans DB	vCPU default	Inti CPU default	Thread per inti default	Jumlah inti CPU yang valid	Jumlah thread per inti yang valid
db.r5d.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.r5d.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
db.r5d.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.r5d.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.r4 – kelas instans dengan memori yang dioptimalkan					
db.r4.large	2	1	2	1	1, 2
db.r4.xlarge	4	2	2	1, 2	1, 2
db.r4.2xlarge	8	4	2	1, 2, 3, 4	1, 2
db.r4.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2

Kelas instans DB	vCPU default	Inti CPU default	Thread per inti default	Jumlah inti CPU yang valid	Jumlah thread per inti yang valid
db.r4.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
db.r4.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.r3 – kelas instans dengan memori yang dioptimalkan					
db.r3.large	2	1	2	1	1, 2
db.r3.xlarge	4	2	2	1, 2	1, 2
db.r3.2xlarge	8	4	2	1, 2, 3, 4	1, 2
db.r3.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
db.r3.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.x2idn – kelas instans dengan memori yang dioptimalkan					
db.x2idn.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2

Kelas instans DB	vCPU default	Inti CPU default	Thread per inti default	Jumlah inti CPU yang valid	Jumlah thread per inti yang valid
db.x2idn.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.x2idn.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
db.x2iedn – kelas instans dengan memori yang dioptimalkan					
db.x2iedn.xlarge	4	2	2	1, 2	1, 2
db.x2iedn.2xlarge	8	4	2	2, 4	1, 2
db.x2iedn.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.x2iedn.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2

Kelas instans DB	vCPU default	Inti CPU default	Thread per inti default	Jumlah inti CPU yang valid	Jumlah thread per inti yang valid
db.x2iedn.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.x2iedn.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.x2iedn.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
db.x2iezn – kelas instans dengan memori yang dioptimalkan					
db.x2iezn.2xlarge	8	4	2	2, 4	1, 2
db.x2iezn.4xlarge	16	8	2	2, 4, 6, 8	1, 2

Kelas instans DB	vCPU default	Inti CPU default	Thread per inti default	Jumlah inti CPU yang valid	Jumlah thread per inti yang valid
db.x2iezn.6xlarge	24	12	2	2, 4, 6, 8, 10, 12	1, 2
db.x2iezn.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.x2iezn.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
db.x1 – kelas instans dengan memori yang dioptimalkan					
db.x1.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.x1.32xlarge	128	64	2	4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64	1, 2
db.x1e – kelas instan dengan memori yang dioptimalkan					
db.x1e.xlarge	4	2	2	1, 2	1, 2
db.x1e.2xlarge	8	4	2	1, 2, 3, 4	1, 2
db.x1e.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2

Kelas instans DB	vCPU default	Inti CPU default	Thread per inti default	Jumlah inti CPU yang valid	Jumlah thread per inti yang valid
db.x1e.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
db.x1e.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.x1e.32xlarge	128	64	2	4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64	1, 2
db.z1d – kelas instans dengan memori yang dioptimalkan					
db.z1d.large	2	1	2	1	1, 2
db.z1d.xlarge	4	2	2	2	1, 2
db.z1d.2xlarge	8	4	2	2, 4	1, 2
db.z1d.3xlarge	12	6	2	2, 4, 6	1, 2
db.z1d.6xlarge	24	12	2	2, 4, 6, 8, 10, 12	1, 2

Kelas instans DB	vCPU default	Inti CPU default	Thread per inti default	Jumlah inti CPU yang valid	Jumlah thread per inti yang valid
db.z1d.12xlarge	48	24	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2

> **Note**
>
> Anda dapat menggunakan AWS CloudTrail untuk memantau dan mengaudit perubahan pada konfigurasi proses Amazon RDS for Oracle DB instans. Untuk informasi selengkapnya tentang penggunaan CloudTrail, lihatMemantau panggilan API Amazon RDS di AWS CloudTrail.

Mengatur inti CPU dan threads per inti CPU untuk kelas instans DB

Anda dapat mengonfigurasi jumlah inti CPU dan thread per inti untuk kelas instans DB saat Anda melakukan operasi berikut:

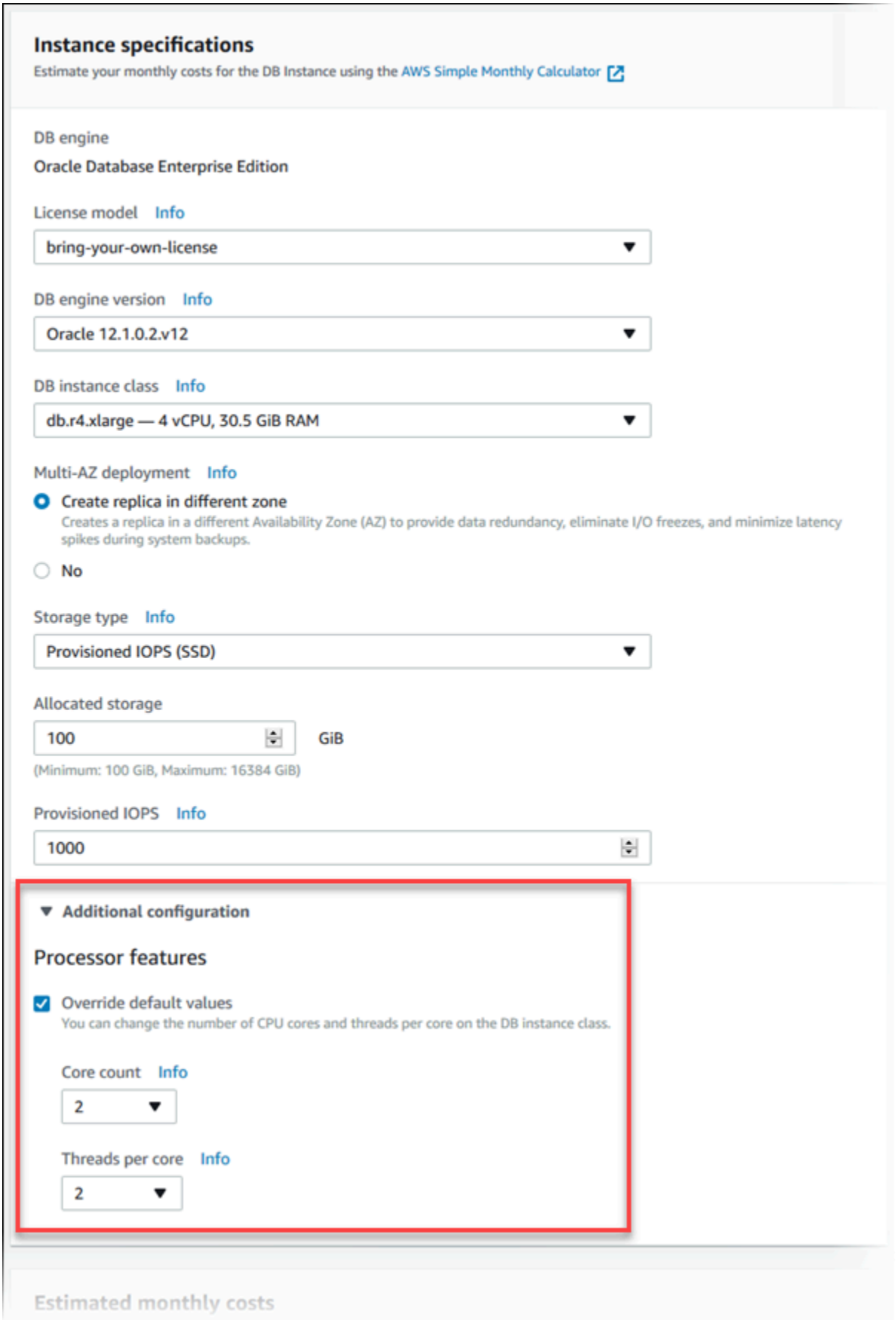
- Membuat instans DB Amazon RDS
- Memodifikasi instans DB Amazon RDS
- Memulihkan dari snapshot DB
- Memulihkan instans DB dengan waktu yang ditentukan

> **Note**
>
> Ketika Anda memodifikasi instans DB untuk mengonfigurasi jumlah inti atau thread per inti CPU, ada gangguan instans DB singkat.

Anda dapat mengatur inti CPU dan utas per inti CPU untuk kelas instans DB menggunakan AWS Management Console, API AWS CLI, atau RDS.

Konsol

Saat membuat, memodifikasi, atau memulihkan instans DB, Anda akan mengatur kelas instans DB di AWS Management Console. Bagian Spesifikasi instans menampilkan opsi untuk prosesor. Gambar berikut menunjukkan opsi fitur prosesor.

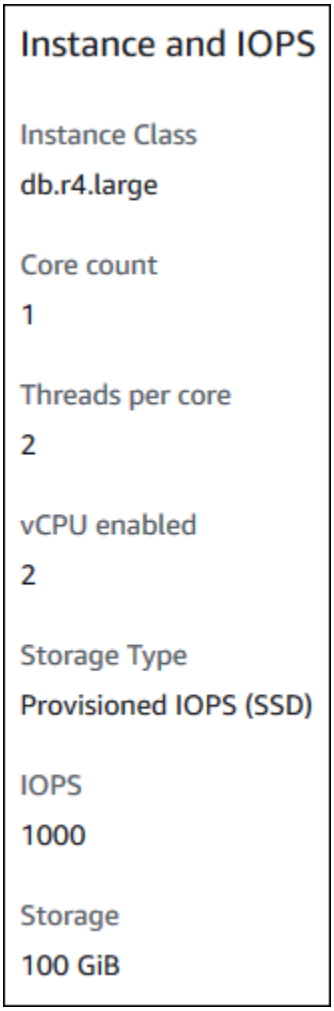

Instance specifications
Estimate your monthly costs for the DB Instance using the AWS Simple Monthly Calculator
DB engine
Oracle Database Enterprise Edition
License model Info
bring-your-own-license
DB engine version Info
Oracle 12.1.0.2.v12
DB instance class Info
db.r4.xlarge — 4 vCPU, 30.5 GiB RAM
Multi-AZ deployment Info
Create replica in different zone
Creates a replica in a different Availability Zone (AZ) to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups.
No
Storage type Info
Provisioned IOPS (SSD)
Allocated storage
100
GiB
(Minimum: 100 GiB, Maximum: 16384 GiB)
Provisioned IOPS Info
1000
Additional configuration
Processor features
Override default values
You can change the number of CPU cores and threads per core on the DB instance class.
Core count Info
2
Threads per core Info
2
Estimated monthly costs

Atur opsi berikut ke nilai yang sesuai untuk kelas instans DB Anda di bagian Fitur prosesor:

- Jumlah inti – Atur jumlah inti CPU menggunakan opsi ini. Nilainya harus sama dengan atau kurang dari jumlah maksimum inti CPU untuk kelas instans DB.
- Thread per inti – Tentukan 2 untuk mengaktifkan beberapa thread per inti, atau tentukan 1 untuk menonaktifkan beberapa thread per inti.

Saat memodifikasi atau memulihkan instans DB, Anda juga dapat mengatur inti CPU dan thread per inti CPU ke default untuk kelas instans.

Saat melihat detail untuk instans DB di konsol, Anda dapat melihat informasi prosesor untuk kelas instans DB di tab Konfigurasi. Gambar berikut menunjukkan kelas instans DB dengan satu inti CPU dan beberapa thread per inti diaktifkan.



Untuk instans DB Oracle, informasi prosesor hanya muncul untuk instans DB Bawa Lisensi Sendiri (BYOL).

AWS CLI

Anda dapat mengatur fitur prosesor untuk instans DB ketika Anda menjalankan salah satu perintah AWS CLI berikut:

- create-db-instance
- modify-db-instance
- restore-db-instance-from-db-snapshot
- restore-db-instance-from-s3
- restore-db-instance-to-point-in-time

Untuk mengkonfigurasi prosesor kelas instans DB untuk instance DB dengan menggunakan AWS CLI, sertakan `--processor-features` opsi dalam perintah. Tentukan jumlah inti CPU dengan nama fitur `coreCount`, dan tentukan apakah beberapa thread per inti diaktifkan dengan nama fitur `threadsPerCore`.

Opsi tersebut memiliki sintaks berikut.

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

Berikut ini adalah contoh yang mengonfigurasi prosesor:

Contoh

- Mengatur jumlah inti CPU untuk instans DB
- Mengatur jumlah inti CPU dan menonaktifkan beberapa thread untuk instans DB
- Melihat nilai prosesor yang valid untuk kelas instans DB
- Mengembalikan ke pengaturan prosesor default untuk instans DB
- Mengembalikan ke jumlah default inti CPU untuk instans DB
- Mengembalikan ke jumlah thread per inti default untuk instans DB

Mengatur jumlah inti CPU untuk instans DB

Example

Contoh berikut menyesuaikan `mydbinstance` dengan mengatur jumlah inti CPU menjadi 4. Perubahan langsung diterapkan dengan menggunakan `--apply-immediately`. Jika Anda ingin

menerapkan perubahan pada periode pemeliharaan terjadwal berikutnya, hilangkan opsi `--apply-immediately`.

Untuk Linux, macOS, atau Unix:

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

Untuk Windows:

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

Mengatur jumlah inti CPU dan menonaktifkan beberapa thread untuk instans DB

Example

Contoh berikut menyesuaikan `mydbinstance` dengan mengatur jumlah inti CPU untuk `4` dan menonaktifkan beberapa thread per inti. Perubahan langsung diterapkan dengan menggunakan `--apply-immediately`. Jika Anda ingin menerapkan perubahan pada periode pemeliharaan terjadwal berikutnya, hilangkan opsi `--apply-immediately`.

Untuk Linux, macOS, atau Unix:

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

Untuk Windows:

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

Melihat nilai prosesor yang valid untuk kelas instans DB

Example

Anda dapat melihat nilai prosesor yang valid untuk kelas instans DB tertentu dengan menjalankan perintah describe-orderable-db-instance-options dan menentukan kelas instance untuk opsi tersebut`--db-instance-class`. Misalnya, output untuk perintah berikut menunjukkan opsi prosesor untuk kelas instans db.r3.large.

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

Berikut ini adalah sampel output untuk perintah dalam format JSON.

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

```
                    {
                        "Name": "threadsPerCore",
                        "DefaultValue": "2",
                        "AllowedValues": "1,2"
                    }
                ],
                "SupportsEnhancedMonitoring": true,
                "SupportsPerformanceInsights": false,
                "MinIopsPerDbInstance": 1000,
                "StorageType": "io1",
                "Vpc": false,
                "SupportsStorageEncryption": true,
                "Engine": "oracle-ee",
                "MultiAZCapable": true
    }
```

Selain itu, Anda dapat menjalankan perintah berikut untuk informasi prosesor kelas instans DB:

- describe-db-instances— Menampilkan informasi prosesor untuk instans DB yang ditentukan.
- describe-db-snapshots— Menampilkan informasi prosesor untuk snapshot DB yang ditentukan.
- describe-valid-db-instance-modifikasi - Menunjukkan modifikasi yang valid pada prosesor untuk instans DB yang ditentukan.

Dalam output perintah sebelumnya, nilai untuk fitur prosesor tidak null hanya jika kondisi berikut terpenuhi:

- Anda menggunakan instans DB RDS for Oracle.
- Instans DB RDS for Oracle Anda mendukung perubahan nilai prosesor.
- Pengaturan inti dan thread CPU saat ini diatur ke nilai nondefault.

Jika kondisi sebelumnya tidak terpenuhi, Anda bisa mendapatkan jenis instance menggunakan. describe-db-instances Anda bisa mendapatkan informasi prosesor untuk jenis instance ini dengan menjalankan operasi describe-instance-typesEC2.

Mengembalikan ke pengaturan prosesor default untuk instans DB

Example

Contoh berikut memodifikasi `mydbinstance` dengan mengembalikan kelas instans DB-nya ke nilai prosesor default. Perubahan langsung diterapkan dengan menggunakan `--apply-immediately`.

Jika Anda ingin menerapkan perubahan pada periode pemeliharaan terjadwal berikutnya, hilangkan opsi `--apply-immediately`.

Untuk Linux, macOS, atau Unix:

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

Untuk Windows:

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

Mengembalikan ke jumlah default inti CPU untuk instans DB

Example

Contoh berikut menyesuaikan `mydbinstance` dengan mengembalikan kelas instans DB ke jumlah inti CPU default. Pengaturan thread per inti tidak diubah. Perubahan langsung diterapkan dengan menggunakan `--apply-immediately`. Jika Anda ingin menerapkan perubahan pada periode pemeliharaan terjadwal berikutnya, hilangkan opsi `--apply-immediately`.

Untuk Linux, macOS, atau Unix:

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

Untuk Windows:

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

Mengembalikan ke jumlah thread per inti default untuk instans DB

Example

Contoh berikut menyesuaikan `mydbinstance` dengan mengembalikan kelas instans DB ke jumlah thread per inti default. Pengaturan jumlah inti CPU tidak berubah. Perubahan langsung diterapkan dengan menggunakan `--apply-immediately`. Jika Anda ingin menerapkan perubahan pada periode pemeliharaan terjadwal berikutnya, hilangkan opsi `--apply-immediately`.

Untuk Linux, macOS, atau Unix:

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

Untuk Windows:

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

API RDS

Anda dapat mengatur fitur prosesor untuk instans DB ketika memanggil salah satu operasi API Amazon RDS berikut:

- CreateDBInstance
- ModifyDBInstance
- InstanceFromDipuhulihkanB DBSnapshot
- DikembalikanB S3 InstanceFrom
- DikembalikanB InstanceToPointInTime

Untuk mengonfigurasi fitur prosesor kelas instans DB untuk instans DB menggunakan API Amazon RDS, sertakan parameter `ProcessFeatures` dalam panggilan.

Parameter tersebut memiliki sintaks berikut.

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

Tentukan jumlah inti CPU dengan nama fitur `coreCount`, dan tentukan apakah beberapa thread per inti diaktifkan dengan nama fitur `threadsPerCore`.

Anda dapat melihat nilai prosesor yang valid untuk kelas instans DB tertentu dengan menjalankan InstanceOptions operasi DescribeOrderableDB dan menentukan kelas instance untuk `DBInstanceClass` parameter. Anda juga dapat menggunakan operasi berikut:

- DescribeDBInstances – Menampilkan informasi prosesor untuk instans DB tertentu.
- DescribeDBSnapshots – Menampilkan informasi prosesor untuk snapshot DB tertentu.
- DescribeValidDB InstanceModifications - Menunjukkan modifikasi yang valid pada prosesor untuk instans DB yang ditentukan.

Dalam output operasi sebelumnya, nilai untuk fitur prosesor tidak null hanya jika kondisi berikut terpenuhi:

- Anda menggunakan instans DB RDS for Oracle.
- Instans DB RDS for Oracle Anda mendukung perubahan nilai prosesor.
- Pengaturan inti dan thread CPU saat ini diatur ke nilai nondefault.

Jika kondisi sebelumnya tidak terpenuhi, Anda bisa mendapatkan jenis instans menggunakan DescribeDBInstances. Anda bisa mendapatkan informasi prosesor untuk jenis instance ini dengan menjalankan operasi DescribeInstanceTypesEC2.

Spesifikasi perangkat keras kelas instans DB

Terminologi berikut digunakan untuk menjelaskan spesifikasi perangkat keras untuk kelas instans DB:

vCPU

Jumlah unit pemrosesan pusat (CPU) virtual. CPU virtual adalah unit kapasitas yang dapat Anda gunakan untuk membandingkan kelas instans DB. Alih-alih membeli atau menyewa prosesor tertentu untuk digunakan selama beberapa bulan atau tahun, Anda menyewa kapasitas per jam. Tujuan kami adalah menyediakan jumlah kapasitas CPU yang konsisten dan spesifik, dalam batas perangkat keras sebenarnya yang mendasarinya.

ECU

Ukuran relatif daya pemrosesan integer dari instans Amazon EC2. Agar mempermudah developer membandingkan kapasitas CPU antara berbagai kelas instans, kami telah mendefinisikan Unit Komputasi Amazon EC2. Jumlah CPU yang dialokasikan ke instans tertentu dinyatakan dalam Unit Komputasi EC2 ini. Satu ECU saat ini menyediakan kapasitas CPU yang setara dengan prosesor 1.0–1.2 GHz 2007 Opteron atau 2007 Xeon.

Memori (GiB)

RAM, dalam gibibyte, dialokasikan ke instans DB. Sering kali ada rasio yang konsisten antara memori dan vCPU. Sebagai contoh, ambil kelas instans db.r4, yang memiliki rasio memori terhadap vCPU serupa dengan kelas instans db.r5. Namun, untuk sebagian besar kasus penggunaan, kelas instans db.r5 memberikan performa yang lebih baik dan lebih konsisten dibandingkan kelas instans db.r4.

Dioptimalkan EBS

Instans DB menggunakan tumpukan konfigurasi yang dioptimalkan dan menyediakan kapasitas khusus tambahan untuk I/O. Pengoptimalan ini memberikan performa terbaik dengan meminimalkan konflik antara I/O dan lalu lintas lain dari instans Anda. Untuk informasi selengkapnya tentang instans yang dioptimalkan Amazon EBS, lihat Instans yang Dioptimalkan Amazon EBS di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Instans yang dioptimalkan EBS memiliki tingkat IOPS dasar dan maksimum. Tingkat IOPS maksimum diberlakukan pada tingkat instans DB. Sekumpulan volume EBS yang digabungkan untuk memiliki tingkat IOPS yang lebih tinggi dari maksimum tidak boleh melebihi ambang batas tingkat instans. Misalnya, jika IOPS maksimum untuk kelas instans DB tertentu adalah 40.000, dan Anda melampirkan empat volume EBS 64.000 IOPS, maka IOPS maksimumnya adalah 40.000, bukan 256.000. Untuk IOPS maksimum yang spesifik untuk setiap jenis instans EC2, lihat Jenis instans yang didukung di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Maks. Bandwidth EBS (Mbps)

Bandwidth EBS maksimum dalam megabit per detik. Bagi dengan 8 untuk mendapatkan throughput yang diharapkan dalam megabyte per detik.

> **Important**
>
> Volume SSD Tujuan Umum (gp2) untuk instans DB Amazon RDS memiliki batas throughput sebesar 250 MiB/dtk dalam banyak kasus. Namun, batas throughput-nya

dapat bervariasi tergantung pada ukuran volume. Untuk informasi selengkapnya, lihat Jenis volume Amazon EBS di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Bandwith jaringan

Kecepatan jaringan relatif terhadap kelas instans DB lainnya.

Pada tabel berikut, Anda dapat menemukan detail perangkat keras tentang kelas instans DB Amazon RDS .

Untuk informasi tentang dukungan mesin DB Amazon RDS untuk setiap kelas instans DB, lihat Mesin DB yang didukung untuk kelas instans DB.

Kelas instans	vCPU	ECU	Memori (GiB)	Penyimpanan instans (GiB)	Maks. Bandwidth EBS (Mbps)	Bandwith jaringan (Gbps)
db.m7g — kelas instance tujuan umum dengan prosesor Graviton3 AWS						
db.m7g.16xlarge	64	—	256	Khusus yang dioptimalkan EBS	20.000	30
db.m7g.12xlarge	48	—	192	Khusus yang dioptimalkan EBS	15.000	22.5
db.m7g.8xlarge	32	—	128	Khusus yang dioptimalkan EBS	10.000	15
db.m7g.4xlarge	16	—	64	Khusus yang dioptimalkan EBS	Hingga 10.000	Hingga 15

Kelas instans	vCPU	ECU	Memori (GiB)	Penyimpanan instans (GiB)	Maks. Bandwidth EBS (Mbps)	Bandwith jaringan (Gbps)
db.m7g.2xlarge*	8	—	32	Khusus yang dioptimalkan EBS	Hingga 10.000	Hingga 15
db.m7g.xlarge*	4	—	16	Khusus yang dioptimalkan EBS	Hingga 10.000	Hingga 12,5
db.m7g.large*	2	—	8	Khusus yang dioptimalkan EBS	Hingga 10.000	Hingga 12,5
db.m6g — kelas instance tujuan umum dengan prosesor Graviton2 AWS						
db.m6g.16xlarge	64	—	256	Khusus yang dioptimalkan EBS	19.000	25
db.m6g.12xlarge	48	—	192	Khusus yang dioptimalkan EBS	13.500	20
db.m6g.8xlarge	32	—	128	Khusus yang dioptimalkan EBS	9.500	12
db.m6g.4xlarge	16	—	64	Khusus yang dioptimalkan EBS	6,800	Hingga 10
db.m6g.2xlarge*	8	—	32	Khusus yang dioptimalkan EBS	Hingga 4.750	Hingga 10

Kelas instans	vCPU	ECU	Memori (GiB)	Penyimpanan instans (GiB)	Maks. Bandwidth EBS (Mbps)	Bandwith jaringan (Gbps)
db.m6g.xlarge*	4	—	16	Khusus yang dioptimalkan EBS	Hingga 4.750	Hingga 10
db.m6g.large*	2	—	8	Khusus yang dioptimalkan EBS	Hingga 4.750	Hingga 10
db.m6gd — kelas instans tujuan umum dengan prosesor Graviton2 dan penyimpanan SSD AWS						
db.m6gd.16xlarge	64	—	256	2 x 1900 NVMe SSD	19.000	25
db.m6gd.12xlarge	48	—	192	2 x 1425 NVMe SSD	13.500	20
db.m6gd.8xlarge	32	—	128	1 x 1900 NVMe SSD	9.000	12
db.m6gd.4xlarge	16	—	64	1 x 950 NVMe SSD	4,750	Hingga 10
db.m6gd.2xlarge	8	—	32	1 x 474 NVMe SSD	Hingga 4.750	Hingga 10
db.m6gd.xlarge	4	—	16	1 x 237 NVMe SSD	Hingga 4.750	Hingga 10
db.m6gd.large	2	—	8	1 x 118 NVMe SSD	Hingga 4.750	Hingga 10
db.m6id – kelas instans tujuan umum dengan prosesor Intel Xeon Scalable Generasi ke-3 dan penyimpanan SSD						

Kelas instans	vCPU	ECU	Memori (GiB)	Penyimpanan instans (GiB)	Maks. Bandwidth EBS (Mbps)	Bandwith jaringan (Gbps)
db.m6id.32xlarge	128	—	512	4 x 1900 NVMe SSD	40.000	50
db.m6id.24xlarge	96	—	384	4 x 1425 NVMe SSD	30.000	37,5
db.m6id.16xlarge	64	—	256	2 x 1900 NVMe SSD	20.000	25
db.m6id.12xlarge	48	—	192	2 x 1425 NVMe SSD	15.000	18.75
db.m6id.8xlarge	32	—	128	1 x 1900 NVMe SSD	10.000	12,5
db.m6id.4xlarge*	16	—	64	1 x 950 NVMe SSD	Hingga 10.000	Hingga 12,5
db.m6id.2xlarge*	8	—	32	1 x 474 NVMe SSD	Hingga 10.000	Hingga 12,5
db.m6id.xlarge*	4	—	16	1 x 237 NVMe SSD	Hingga 10.000	Hingga 12,5
db.m6id.large*	2	—	8	1 x 118 NVMe SSD	Hingga 10.000	Hingga 12,5
db.m6idn – kelas instans tujuan umum dengan prosesor Intel Xeon Scalable Generasi ke-3, penyimpanan SSD, dan pengoptimalan jaringan						
db.m6idn.32xlarge	128	—	512	4 x 1900 NVMe SSD	80.000	200
db.m6idn.24xlarge	96	—	384	4 x 1425 NVMe SSD	60.000	150

Kelas instans	vCPU	ECU	Memori (GiB)	Penyimpanan instans (GiB)	Maks. Bandwidth EBS (Mbps)	Bandwith jaringan (Gbps)
db.m6idn.16xlarge	64	—	256	2 x 1900 NVMe SSD	40.000	100
db.m6idn.12xlarge	48	—	192	2 x 1425 NVMe SSD	30.000	75
db.m6idn.8xlarge	32	—	128	1 x 1900 NVMe SSD	20.000	50
db.m6idn.4xlarge*	16	—	64	1 x 950 NVMe SSD	Hingga 20.000	Hingga 50
db.m6idn.2xlarge*	8	—	32	1 x 474 NVMe SSD	Hingga 20.000	Hingga 40
db.m6idn.xlarge*	4	—	16	1 x 237 NVMe SSD	Hingga 20.000	Hingga 30
db.m6idn.large*	2	—	8	1 x 118 NVMe SSD	Hingga 20.000	Hingga 25
db.m6in – kelas instans tujuan umum dengan prosesor Intel Xeon Scalable Generasi ke-3 dan pengoptimalan jaringan						
db.m6in.32xlarge	128	—	512	Khusus yang dioptimalkan EBS	80.000	200
db.m6in.24xlarge	96	—	384	Khusus yang dioptimalkan EBS	60.000	150
db.m6in.16xlarge	64	—	256	Khusus yang dioptimalkan EBS	40.000	100

Kelas instans	vCPU	ECU	Memori (GiB)	Penyimpanan instans (GiB)	Maks. Bandwidth EBS (Mbps)	Bandwith jaringan (Gbps)
db.m6in.12xlarge	48	—	192	Khusus yang dioptimalkan EBS	30.000	75
db.m6in.8xlarge	32	—	128	Khusus yang dioptimalkan EBS	20.000	50
db.m6in.4xlarge*	16	—	64	Khusus yang dioptimalkan EBS	Hingga 20.000	Hingga 50
db.m6in.2xlarge*	8	—	32	Khusus yang dioptimalkan EBS	Hingga 20.000	Hingga 40
db.m6in.xlarge*	4	—	16	Khusus yang dioptimalkan EBS	Hingga 20.000	Hingga 30
db.m6in.large*	2	—	8	Khusus yang dioptimalkan EBS	Hingga 20.000	Hingga 25
db.m6i – kelas instans tujuan umum dengan prosesor Intel Xeon Scalable Generasi ke-3						
db.m6i.32xlarge	128	—	512	Khusus yang dioptimalkan EBS	50.000	40
db.m6i.24xlarge	96	—	384	Khusus yang dioptimalkan EBS	37.500	30

Kelas instans	vCPU	ECU	Memori (GiB)	Penyimpanan instans (GiB)	Maks. Bandwidth EBS (Mbps)	Bandwith jaringan (Gbps)
db.m6i.16xlarge	64	—	256	Khusus yang dioptimalkan EBS	25.000	20
db.m6i.12xlarge	48	—	192	Khusus yang dioptimalkan EBS	18,750	15
db.m6i.8xlarge	32	—	128	Khusus yang dioptimalkan EBS	12.500	10
db.m6i.4xlarge*	16	—	64	Khusus yang dioptimalkan EBS	Hingga 12.500	Hingga 10
db.m6i.2xlarge*	8	—	32	Khusus yang dioptimalkan EBS	Hingga 12.500	Hingga 10
db.m6i.xlarge*	4	—	16	Khusus yang dioptimalkan EBS	Hingga 12.500	Hingga 10
db.m6i.large*	2	—	8	Khusus yang dioptimalkan EBS	Hingga 12.500	Hingga 10
db.m5d – kelas instans tujuan umum dengan prosesor Intel Xeon Platinum dan penyimpanan SSD						
db.m5d.24xlarge	96	345	384	4 x 900 NVMe SSD	19.000	25
db.m5d.16xlarge	64	262	256	4 x 600 NVMe SSD	13.600	20

Kelas instans	vCPU	ECU	Memori (GiB)	Penyimpanan instans (GiB)	Maks. Bandwidth EBS (Mbps)	Bandwith jaringan (Gbps)
db.m5d.12xlarge	48	173	192	2 x 900 NVMe SSD	9.500	10
db.m5d.8xlarge	32	131	128	2 x 600 NVMe SSD	6,800	10
db.m5d.4xlarge	16	61	64	2 x 300 NVMe SSD	4,750	Hingga 10
db.m5d.2xlarge*	8	31	32	1 x 300 NVMe SSD	Hingga 4.750	Hingga 10
db.m5d.xlarge*	4	15	16	1 x 150 NVMe SSD	Hingga 4.750	Hingga 10
db.m5d.large*	2	10	8	1 x 75 NVMe SSD	Hingga 4.750	Hingga 10
db.m5 – kelas instans tujuan umum dengan prosesor Intel Xeon Platinum						
db.m5.24xlarge	96	345	384	Khusus yang dioptimalkan EBS	19.000	25
db.m5.16xlarge	64	262	256	Khusus yang dioptimalkan EBS	13.600	20
db.m5.12xlarge	48	173	192	Khusus yang dioptimalkan EBS	9.500	10
db.m5.8xlarge	32	131	128	Khusus yang dioptimalkan EBS	6,800	10

Kelas instans	vCPU	ECU	Memori (GiB)	Penyimpanan instans (GiB)	Maks. Bandwidth EBS (Mbps)	Bandwith jaringan (Gbps)
db.m5.4xlarge	16	61	64	Khusus yang dioptimalkan EBS	4,750	Hingga 10
db.m5.2xlarge*	8	31	32	Khusus yang dioptimalkan EBS	Hingga 4.750	Hingga 10
db.m5.xlarge*	4	15	16	Khusus yang dioptimalkan EBS	Hingga 4.750	Hingga 10
db.m5.large*	2	10	8	Khusus yang dioptimalkan EBS	Hingga 4.750	Hingga 10
db.m4 – kelas instans tujuan umum dengan Prosesor Intel Xeon Scalable						
db.m4.16xlarge	64	188	256	Khusus yang dioptimalkan EBS	10.000	25
db.m4.10xlarge	40	124,5	160	Khusus yang dioptimalkan EBS	4.000	10
db.m4.4xlarge	16	53.5	64	Khusus yang dioptimalkan EBS	2.000	Tinggi
db.m4.2xlarge	8	25.5	32	Khusus yang dioptimalkan EBS	1.000	Tinggi

Kelas instans	vCPU	ECU	Memori (GiB)	Penyimpanan instans (GiB)	Maks. Bandwidth EBS (Mbps)	Bandwith jaringan (Gbps)
db.m4.xlarge	4	13	16	Khusus yang dioptimalkan EBS	750	Tinggi
db.m4.large	2	6.5	8	Khusus yang dioptimalkan EBS	450	Sedang
db.m3 – kelas instans tujuan umum						
db.m3.2xlarge	8	26	30	Khusus yang dioptimalkan EBS	1.000	Tinggi
db.m3.xlarge	4	13	15	Khusus yang dioptimalkan EBS	500	Tinggi
db.m3.large	2	6.5	7.5	Khusus EBS	—	Sedang
db.m3.medium	1	3	3,75	Khusus EBS	—	Sedang
db.m1 – kelas instans tujuan umum						
db.m1.xlarge	4	4	15	Khusus yang dioptimalkan EBS	450	Tinggi
db.m1.large	2	2	7.5	Khusus yang dioptimalkan EBS	450	Sedang
db.m1.medium	1	1	3,75	Khusus EBS	—	Sedang

Kelas instans	vCPU	ECU	Memori (GiB)	Penyimpanan instans (GiB)	Maks. Bandwidth EBS (Mbps)	Bandwith jaringan (Gbps)
db.m1.small	1	1	1.7	Khusus EBS	—	Sangat Rendah
db.x2iezn – kelas instans dengan memori yang dioptimalkan						
db.x2iezn.12xlarge	>48	—	1,536	Khusus yang dioptimalkan EBS	19.000	100
db.x2iezn.8xlarge	32	—	1,024	Khusus yang dioptimalkan EBS	12.000	75
db.x2iezn.6xlarge	24	—	768	Khusus yang dioptimalkan EBS	Hingga 9.500	50
db.x2iezn.4xlarge	16	—	512	Khusus yang dioptimalkan EBS	Hingga 4.750	Hingga 25
db.x2iezn.2xlarge	8	—	256	Khusus yang dioptimalkan EBS	Hingga 3.170	Hingga 25
db.x2iedn – kelas instans yang memorinya dioptimalkan dengan pengoptimalan jaringan dan penyimpanan SSD						
db.x2iedn.32xlarge	128	—	4,096	2 x 1900 NVMe SSD	80.000	100
db.x2iedn.24xlarge	96	—	3,072	2 x 1425 NVMe SSD	60.000	75

Kelas instans	vCPU	ECU	Memori (GiB)	Penyimpanan instans (GiB)	Maks. Bandwidth EBS (Mbps)	Bandwith jaringan (Gbps)
db.x2iedn.16xlarge	64	—	2,048	1 x 1900 NVMe SSD	40.000	50
db.x2iedn.8xlarge	32	—	1,024	1 x 950 NVMe SSD	20.000	25
db.x2iedn.4xlarge	16	—	512	1 x 475 NVMe SSD	Hingga 20.000	Hingga 25
db.x2iedn.2xlarge	8	—	256	1 x 237 NVMe SSD	Hingga 20.000	Hingga 25
db.x2iedn.xlarge	4	—	128	1 x 118 NVMe SSD	Hingga 20.000	Hingga 25
db.x2idn – kelas instans yang memorinya dioptimalkan dengan pengoptimalan jaringan dan penyimpanan SSD						
db.x2idn.32xlarge	128	—	2,048	2 x 1900 NVMe SSD	80.000	100
db.x2idn.24xlarge	96	—	1,536	2 x 1425 NVMe SSD	60.000	75
db.x2idn.16xlarge	64	—	1,024	1 x 1900 NVMe SSD	40.000	50
db.x2g – kelas instans dengan memori yang dioptimalkan						
db.x2g.16xlarge	64	—	1024	Khusus yang dioptimalkan EBS	19.000	25

Kelas instans	vCPU	ECU	Memori (GiB)	Penyimpanan instans (GiB)	Maks. Bandwidth EBS (Mbps)	Bandwith jaringan (Gbps)
db.x2g.12xlarge	48	—	768	Khusus yang dioptimalkan EBS	14,250	20
db.x2g.8xlarge	32	—	512	Khusus yang dioptimalkan EBS	9.500	12
db.x2g.4xlarge	16	—	256	Khusus yang dioptimalkan EBS	4,750	Hingga 10
db.x2g.2xlarge	8	—	128	Khusus yang dioptimalkan EBS	Hingga 4.750	Hingga 10
db.x2g.xlarge	4	—	64	Khusus yang dioptimalkan EBS	Hingga 4.750	Hingga 10
db.x2g.large	2	—	32	Khusus yang dioptimalkan EBS	Hingga 4.750	Hingga 10
db.z1d – kelas instans yang memorinya dioptimalkan dengan penyimpanan SSD						
db.z1d.12xlarge	48	271	384	2 x 900 NVMe SSD	14.000	25
db.z1d.6xlarge	24	134	192	1 x 900 NVMe SSD	7.000	10
db.z1d.3xlarge	12	75	96	1 x 450 NVMe SSD	3.500	Hingga 10

Kelas instans	vCPU	ECU	Memori (GiB)	Penyimpanan instans (GiB)	Maks. Bandwidth EBS (Mbps)	Bandwith jaringan (Gbps)
db.z1d.2xlarge	8	53	64	1 x 300 NVMe SSD	2,333	Hingga 10
db.z1d.xlarge*	4	28	32	1 x 150 NVMe SSD	Hingga 2.333	Hingga 10
db.z1d.large*	2	15	16	1 x 75 NVMe SSD	Hingga 2.333	Hingga 10
db.x1e – kelas instan dengan memori yang dioptimalkan						
db.x1e.32xlarge	128	340	3,904	Khusus yang dioptimalkan EBS	14.000	25
db.x1e.16xlarge	64	179	1,952	Khusus yang dioptimalkan EBS	7.000	10
db.x1e.8xlarge	32	91	976	Khusus yang dioptimalkan EBS	3.500	Hingga 10
db.x1e.4xlarge	16	47	488	Khusus yang dioptimalkan EBS	1,750	Hingga 10
db.x1e.2xlarge	8	23	244	Khusus yang dioptimalkan EBS	1.000	Hingga 10
db.x1e.xlarge	4	12	122	Khusus yang dioptimalkan EBS	500	Hingga 10

Kelas instans	vCPU	ECU	Memori (GiB)	Penyimpanan instans (GiB)	Maks. Bandwidth EBS (Mbps)	Bandwith jaringan (Gbps)
db.x1 – kelas instans dengan memori yang dioptimalkan						
db.x1.32xlarge	128	349	1,952	Khusus yang dioptimalkan EBS	14.000	25
db.x1.16xlarge	64	174,5	976	Khusus yang dioptimalkan EBS	7.000	10
db.r7g — kelas instance yang dioptimalkan untuk memori dengan prosesor Graviton3 AWS						
db.r7g.16xlarge	64	—	512	Khusus yang dioptimalkan EBS	20.000	30
db.r7g.12xlarge	48	—	384	Khusus yang dioptimalkan EBS	15.000	22.5
db.r7g.8xlarge	32	—	256	Khusus yang dioptimalkan EBS	10.000	15
db.r7g.4xlarge	16	—	128	Khusus yang dioptimalkan EBS	Hingga 10.000	Hingga 15
db.r7g.2xlarge*	8	—	64	Khusus yang dioptimalkan EBS	Hingga 10.000	Hingga 15
db.r7g.xlarge*	4	—	32	Khusus yang dioptimalkan EBS	Hingga 10.000	Hingga 12,5

Kelas instans	vCPU	ECU	Memori (GiB)	Penyimpanan instans (GiB)	Maks. Bandwidth EBS (Mbps)	Bandwith jaringan (Gbps)
db.r7g.large*	2	—	16	Khusus yang dioptimalkan EBS	Hingga 10.000	Hingga 12,5
db.r6g — kelas instance yang dioptimalkan untuk memori dengan prosesor Graviton2 AWS						
db.r6g.16xlarge	64	—	512	Khusus yang dioptimalkan EBS	19.000	25
db.r6g.12xlarge	48	—	384	Khusus yang dioptimalkan EBS	13.500	20
db.r6g.8xlarge	32	—	256	Khusus yang dioptimalkan EBS	9.000	12
db.r6g.4xlarge	16	—	128	Khusus yang dioptimalkan EBS	4,750	Hingga 10
db.r6g.2xlarge*	8	—	64	Khusus yang dioptimalkan EBS	Hingga 4.750	Hingga 10
db.r6g.xlarge*	4	—	32	Khusus yang dioptimalkan EBS	Hingga 4.750	Hingga 10
db.r6g.large*	2	—	16	Khusus yang dioptimalkan EBS	Hingga 4.750	Hingga 10

Kelas instans	vCPU	ECU	Memori (GiB)	Penyimpanan instans (GiB)	Maks. Bandwidth EBS (Mbps)	Bandwith jaringan (Gbps)
db.r6gd — kelas instans yang dioptimalkan untuk memori dengan prosesor Graviton2 dan penyimpanan SSD AWS						
db.r6gd.16xlarge	64	—	512	2 x 1900 NVMe SSD	19.000	25
db.r6gd.12xlarge	48	—	384	2 x 1425 NVMe SSD	13.500	20
db.r6gd.8xlarge	32	—	256	1 x 1900 NVMe SSD	9.000	12
db.r6gd.4xlarge	16	—	128	1 x 950 NVMe SSD	4,750	Hingga 10
db.r6gd.2xlarge	8	—	64	1 x 474 NVMe SSD	Hingga 4.750	Hingga 10
db.r6gd.xlarge	4	—	32	1 x 237 NVMe SSD	Hingga 4.750	Hingga 10
db.r6gd.large	2	—	16	1 x 118 NVMe SSD	Hingga 4.750	Hingga 10
db.r6id – kelas instans tujuan umum dengan prosesor Intel Xeon Scalable Generasi ke-3 dan penyimpanan SSD						
db.r6id.32xlarge	128	—	1,024	4x1900 NVMe SSD	40.000	50
db.r6id.24xlarge	96	—	768	4x1425 NVMe SSD	30.000	37,5
db.r6id.16xlarge	64	—	512	2x1900 NVMe SSD	20.000	25

Kelas instans	vCPU	ECU	Memori (GiB)	Penyimpanan instans (GiB)	Maks. Bandwidth EBS (Mbps)	Bandwith jaringan (Gbps)
db.r6id.12xlarge	48	—	384	2x1425 NVMe SSD	15.000	18.75
db.r6id.8xlarge	32	—	256	1x1900 NVMe SSD	10.000	12,5
db.r6id.4xlarge*	16	—	128	1x950 NVMe SSD	Hingga 10.000	Hingga 12,5
db.r6id.2xlarge*	8	—	64	1x474 NVMe SSD	Hingga 10.000	Hingga 12,5
db.r6id.xlarge*	4	—	32	1x237 NVMe SSD	Hingga 10.000	Hingga 12,5
db.r6id.large*	2	—	16	1x118 NVMe SSD	Hingga 10.000	Hingga 12,5
db.r6idn – kelas instans yang memorinya dioptimalkan dengan prosesor Intel Xeon Scalable generasi ke-3, penyimpanan SSD, dan pengoptimalan jaringan						
db.r6idn.32xlarge	128	—	1,024	4x1900 NVMe SSD	80.000	200
db.r6idn.24xlarge	96	—	768	4x1425 NVMe SSD	60.000	150
db.r6idn.16xlarge	64	—	512	2x1900 NVMe SSD	40.000	100
db.r6idn.12xlarge	48	—	384	2x1425 NVMe SSD	30.000	75
db.r6idn.8xlarge	32	—	256	1x1900 NVMe SSD	20.000	50

Kelas instans	vCPU	ECU	Memori (GiB)	Penyimpanan instans (GiB)	Maks. Bandwidth EBS (Mbps)	Bandwith jaringan (Gbps)
db.r6idn.4xlarge*	16	—	128	1x950 NVMe SSD	Hingga 20.000	Hingga 50
db.r6idn.2xlarge*	8	—	64	1x474 NVMe SSD	Hingga 20.000	Hingga 40
db.r6idn.xlarge*	4	—	32	1x237 NVMe SSD	Hingga 20.000	Hingga 30
db.r6idn.large*	2	—	16	1x118 NVMe SSD	Hingga 20.000	Hingga 25
db.r6in – kelas instans yang memorinya dioptimalkan dengan prosesor Intel Xeon Scalable generasi ke-3 dan pengoptimalan jaringan						
db.r6in.32xlarge	128	—	1,024	Khusus yang dioptimalkan EBS	80.000	200
db.r6in.24xlarge	96	—	768	Khusus yang dioptimalkan EBS	60.000	150
db.r6in.16xlarge	64	—	512	Khusus yang dioptimalkan EBS	40.000	100
db.r6in.12xlarge	48	—	384	Khusus yang dioptimalkan EBS	30.000	75
db.r6in.8xlarge	32	—	256	Khusus yang dioptimalkan EBS	20.000	50

Kelas instans	vCPU	ECU	Memori (GiB)	Penyimpanan instans (GiB)	Maks. Bandwidth EBS (Mbps)	Bandwith jaringan (Gbps)
db.r6in.4xlarge*	16	—	128	Khusus yang dioptimalkan EBS	Hingga 20.000	Hingga 50
db.r6in.2xlarge*	8	—	64	Khusus yang dioptimalkan EBS	Hingga 20.000	Hingga 40
db.r6in.xlarge*	4	—	32	Khusus yang dioptimalkan EBS	Hingga 20.000	Hingga 30
db.r6in.large*	2	—	16	Khusus yang dioptimalkan EBS	Hingga 20.000	Hingga 25
db.r6id – kelas instans tujuan umum dengan prosesor Intel Xeon Scalable Generasi ke-3 dan penyimpanan SSD						
db.r6id.32xlarge	128	—	1,024	4x1900 NVMe SSD	40.000	50
db.r6id.24xlarge	96	—	768	4x1425 NVMe SSD	30.000	37,5
db.r6id.16xlarge	64	—	512	2x1900 NVMe SSD	20.000	25
db.r6id.12xlarge	48	—	384	2x1425 NVMe SSD	15.000	18.75
db.r6id.8xlarge	32	—	256	1x1900 NVMe SSD	10.000	12,5

Kelas instans	vCPU	ECU	Memori (GiB)	Penyimpanan instans (GiB)	Maks. Bandwidth EBS (Mbps)	Bandwith jaringan (Gbps)
db.r6id.4xlarge*	16	—	128	1x950 NVMe SSD	Hingga 10.000	Hingga 12,5
db.r6id.2xlarge*	8	—	64	1x474 NVMe SSD	Hingga 10.000	Hingga 12,5
db.r6id.xlarge*	4	—	32	1x237 NVMe SSD	Hingga 10.000	Hingga 12,5
db.r6id.large*	2	—	16	1x118 NVMe SSD	Hingga 10.000	Hingga 12,5
db.r6i – kelas instans yang memorinya dioptimalkan dengan prosesor Intel Xeon Scalable generasi ke-3						
db.r6i.32xlarge	128	—	1,024	Khusus yang dioptimalkan EBS	40.000	50
db.r6i.24xlarge	96	—	768	Khusus yang dioptimalkan EBS	30.000	37,5
db.r6i.16xlarge	64	—	512	Khusus yang dioptimalkan EBS	20.000	25
db.r6i.12xlarge	48	—	384	Khusus yang dioptimalkan EBS	15.000	18.75
db.r6i.8xlarge	32	—	256	Khusus yang dioptimalkan EBS	10.000	12,5

Kelas instans	vCPU	ECU	Memori (GiB)	Penyimpanan instans (GiB)	Maks. Bandwidth EBS (Mbps)	Bandwith jaringan (Gbps)
db.r6i.4xlarge*	16	—	128	Khusus yang dioptimalkan EBS	Hingga 10.000	Hingga 12,5
db.r6i.2xlarge*	8	—	64	Khusus yang dioptimalkan EBS	Hingga 10.000	Hingga 12,5
db.r6i.xlarge*	4	—	32	Khusus yang dioptimalkan EBS	Hingga 10.000	Hingga 12,5
db.r6i.large*	2	—	16	Khusus yang dioptimalkan EBS	Hingga 10.000	Hingga 12,5
db.r5d – kelas instans yang memorinya dioptimalkan dengan prosesor Intel Xeon Platinum dan penyimpanan SSD						
db.r5d.24xlarge	96	347	768	4 x 900 NVMe SSD	19.000	25
db.r5d.16xlarge	64	264	512	4 x 600 NVMe SSD	13.600	20
db.r5d.12xlarge	48	173	384	2 x 900 NVMe SSD	9.500	10
db.r5d.8xlarge	32	132	256	2 x 600 NVMe SSD	6,800	10
db.r5d.4xlarge	16	71	128	2 x 300 NVMe SSD	4,750	Hingga 10

Kelas instans	vCPU	ECU	Memori (GiB)	Penyimpanan instans (GiB)	Maks. Bandwidth EBS (Mbps)	Bandwith jaringan (Gbps)
db.r5d.2xlarge*	8	38	64	1 x 300 NVMe SSD	Hingga 4.750	Hingga 10
db.r5d.xlarge*	4	19	32	1 x 150 NVMe SSD	Hingga 4.750	Hingga 10
db.r5d.large*	2	10	16	1 x 75 NVMe SSD	Hingga 4.750	Hingga 10
db.r5b – kelas instans yang memorinya dioptimalkan dengan prosesor Intel Xeon Platinum dan pengoptimalan EBS						
db.r5b.24xlarge	96	347	768	Khusus yang dioptimalkan EBS	60.000	25
db.r5b.16xlarge	64	264	512	Khusus yang dioptimalkan EBS	40.000	20
db.r5b.12xlarge	48	173	384	Khusus yang dioptimalkan EBS	30.000	10
db.r5b.8xlarge	32	132	256	Khusus yang dioptimalkan EBS	20.000	10
db.r5b.4xlarge	16	71	128	Khusus yang dioptimalkan EBS	10.000	Hingga 10
db.r5b.2xlarge*	8	38	64	Khusus yang dioptimalkan EBS	Hingga 10.000	Hingga 10

Kelas instans	vCPU	ECU	Memori (GiB)	Penyimpanan instans (GiB)	Maks. Bandwidth EBS (Mbps)	Bandwith jaringan (Gbps)
db.r5b.xlarge*	4	19	32	Khusus yang dioptimalkan EBS	Hingga 10.000	Hingga 10
db.r5b.large*	2	10	16	Khusus yang dioptimalkan EBS	Hingga 10.000	Hingga 10
db.r5b – Kelas instans Oracle dengan memori yang dioptimalkan yang telah dikonfigurasikan sebelumnya untuk penyimpanan, I/O, dan memori tinggi						
db.r5b.8xlarge.tpc2.mem3x	32	—	768	Khusus yang dioptimalkan EBS	60.000	25
db.r5b.6xlarge.tpc2.mem4x	24	—	768	Khusus yang dioptimalkan EBS	60.000	25
db.r5b.4xlarge.tpc2.mem4x	16	—	512	Khusus yang dioptimalkan EBS	40.000	20
db.r5b.4xlarge.tpc2.mem3x	16	—	384	Khusus yang dioptimalkan EBS	30.000	10
db.r5b.4xlarge.tpc2.mem2x	16	—	256	Khusus yang dioptimalkan EBS	20.000	10
db.r5b.2xlarge.tpc2.mem8x	8	—	512	Khusus yang dioptimalkan EBS	40.000	20

Kelas instans	vCPU	ECU	Memori (GiB)	Penyimpanan instans (GiB)	Maks. Bandwidth EBS (Mbps)	Bandwith jaringan (Gbps)
db.r5b.2xlarge.tpc2.mem4x	8	—	256	Khusus yang dioptimalkan EBS	20.000	10
db.r5b.2xlarge.tpc1.mem2x	8	—	128	Khusus yang dioptimalkan EBS	10.000	Hingga 10
db.r5b.xlarge.tpc2.mem4x	4	—	128	Khusus yang dioptimalkan EBS	10.000	Hingga 10
db.r5b.xlarge.tpc2.mem2x	4	—	64	Khusus yang dioptimalkan EBS	Hingga 10.000	Hingga 10
db.r5b.large.tpc1.mem2x	2	—	32	Khusus yang dioptimalkan EBS	Hingga 10.000	Hingga 10
db.r5 – kelas instans yang memorinya dioptimalkan dengan prosesor Intel Xeon Platinum						
db.r5.24xlarge	96	347	768	Khusus yang dioptimalkan EBS	19.000	25
db.r5.16xlarge	64	264	512	Khusus yang dioptimalkan EBS	13.600	20
db.r5.12xlarge	48	173	384	Khusus yang dioptimalkan EBS	9.500	12

Kelas instans	vCPU	ECU	Memori (GiB)	Penyimpanan instans (GiB)	Maks. Bandwidth EBS (Mbps)	Bandwith jaringan (Gbps)
db.r5.8xlarge	32	132	256	Khusus yang dioptimalkan EBS	6,800	10
db.r5.4xlarge	16	71	128	Khusus yang dioptimalkan EBS	4,750	Hingga 10
db.r5.2xlarge*	8	38	64	Khusus yang dioptimalkan EBS	Hingga 4.750	Hingga 10
db.r5.xlarge*	4	19	32	Khusus yang dioptimalkan EBS	Hingga 4.750	Hingga 10
db.r5.large*	2	10	16	Khusus yang dioptimalkan EBS	Hingga 4.750	Hingga 10
db.r5 – Kelas instans Oracle dengan memori yang dioptimalkan yang telah dikonfigurasikan sebelumnya untuk penyimpanan, I/O, dan memori tinggi						
db.r5.12xlarge.tpc2.mem2x	48	—	768	Khusus yang dioptimalkan EBS	19.000	25
db.r5.8xlarge.tpc2.mem3x	32	—	768	Khusus yang dioptimalkan EBS	19.000	25
db.r5.6xlarge.tpc2.mem4x	24	—	768	Khusus yang dioptimalkan EBS	19.000	25

Kelas instans	vCPU	ECU	Memori (GiB)	Penyimpanan instans (GiB)	Maks. Bandwidth EBS (Mbps)	Bandwith jaringan (Gbps)
db.r5.4xlarge.tpc2.mem4x	16	—	512	Khusus yang dioptimalkan EBS	13.600	20
db.r5.4xlarge.tpc2.mem3x	16	—	384	Khusus yang dioptimalkan EBS	9.500	10
db.r5.4xlarge.tpc2.mem2x	16	—	256	Khusus yang dioptimalkan EBS	6,800	10
db.r5.2xlarge.tpc2.mem8x	8	—	512	Khusus yang dioptimalkan EBS	13.600	20
db.r5.2xlarge.tpc2.mem4x	8	—	256	Khusus yang dioptimalkan EBS	6,800	10
db.r5.2xlarge.tpc1.mem2x	8	—	128	Khusus yang dioptimalkan EBS	4,750	Hingga 10
db.r5.xlarge.tpc2.mem4x	4	—	128	Khusus yang dioptimalkan EBS	4,750	Hingga 10
db.r5.xlarge.tpc2.mem2x	4	—	64	Khusus yang dioptimalkan EBS	Hingga 4.750	Hingga 10
db.r5.large.tpc1.mem2x	2	—	32	Khusus yang dioptimalkan EBS	Hingga 4.750	Hingga 10

Kelas instans	vCPU	ECU	Memori (GiB)	Penyimpanan instans (GiB)	Maks. Bandwidth EBS (Mbps)	Bandwith jaringan (Gbps)
db.r4 – kelas instans yang memorinya dioptimalkan dengan prosesor Intel Xeon Scalable						
db.r4.16xlarge	64	195	488	Khusus yang dioptimalkan EBS	14.000	25
db.r4.8xlarge	32	99	244	Khusus yang dioptimalkan EBS	7.000	10
db.r4.4xlarge	16	53	122	Khusus yang dioptimalkan EBS	3.500	Hingga 10
db.r4.2xlarge	8	27	61	Khusus yang dioptimalkan EBS	1.700	Hingga 10
db.r4.xlarge	4	13,5	30,5	Khusus yang dioptimalkan EBS	850	Hingga 10
db.r4.large	2	7	15.25	Khusus yang dioptimalkan EBS	425	Hingga 10
db.r3 – kelas instans dengan memori yang dioptimalkan						
db.r3.8xlarge	32	104	244	Khusus EBS	—	10
db.r3.4xlarge	16	52	122	Khusus yang dioptimalkan EBS	2.000	Tinggi

Kelas instans	vCPU	ECU	Memori (GiB)	Penyimpanan instans (GiB)	Maks. Bandwidth EBS (Mbps)	Bandwith jaringan (Gbps)
db.r3.2xlarge	8	26	61	Khusus yang dioptimalkan EBS	1.000	Tinggi
db.r3.xlarge	4	13	30,5	Khusus yang dioptimalkan EBS	500	Sedang
db.r3.large	2	6.5	15.25	Khusus yang dioptimalkan EBS	—	Sedang
db.c6gd — kelas instans yang dioptimalkan komputasi (hanya untuk penerapan klaster DB multi-AZ)						
db.c6gd.16xlarge	64	—	128	2 x 1900 NVMe SSD	19.000	25
db.c6gd.12xlarge	48	—	96	2 x 1425 NVMe SSD	13.500	20
db.c6gd.8xlarge	32	—	64	1 x 1900 NVMe SSD	9.000	12
db.c6gd.4xlarge	16	—	32	1 x 950 NVMe SSD	4,750	Hingga 10
db.c6gd.2xlarge	8	—	16	1 x 474 NVMe SSD	Hingga 4.750	Hingga 10
db.c6gd.xlarge	4	—	8	1 x 237 NVMe SSD	Hingga 4.750	Hingga 10
db.c6gd.large	2	—	4	1 x 118 NVMe SSD	Hingga 4.750	Hingga 10

Kelas instans	vCPU	ECU	Memori (GiB)	Penyimpanan instans (GiB)	Maks. Bandwidth EBS (Mbps)	Bandwith jaringan (Gbps)
db.c6gd.sedang	1	—	2	1 x 59 NVMe SSD	Hingga 4.750	Hingga 10
db.t4g — kelas instance performa yang dapat dibobol dengan prosesor Graviton2 AWS						
db.t4g.2xlarge*	8	—	32	Khusus yang dioptimalkan EBS	Hingga 2.780	Hingga 5
db.t4g.xlarge*	4	—	16	Khusus yang dioptimalkan EBS	Hingga 2.780	Hingga 5
db.t4g.large*	2	—	8	Khusus yang dioptimalkan EBS	Hingga 2.780	Hingga 5
db.t4g.medium*	2	—	4	Khusus yang dioptimalkan EBS	Hingga 2.085	Hingga 5
db.t4g.small*	2	—	2	Khusus yang dioptimalkan EBS	Hingga 2.085	Hingga 5
db.t4g.micro*	2	—	1	Khusus yang dioptimalkan EBS	Hingga 2.085	Hingga 5
db.t3 – kelas instans performa yang dapat melonjak						
db.t3.2xlarge*	8	Variat	32	Khusus yang dioptimalkan EBS	Hingga 2.048	Hingga 5

Kelas instans	vCPU	ECU	Memori (GiB)	Penyimpanan instans (GiB)	Maks. Bandwidth EBS (Mbps)	Bandwith jaringan (Gbps)
db.t3.xlarge*	4	Variab	16	Khusus yang dioptimalkan EBS	Hingga 2.048	Hingga 5
db.t3.large*	2	Variab	8	Khusus yang dioptimalkan EBS	Hingga 2.048	Hingga 5
db.t3.medium*	2	Variab	4	Khusus yang dioptimalkan EBS	Hingga 1.536	Hingga 5
db.t3.small*	2	Variab	2	Khusus yang dioptimalkan EBS	Hingga 1.536	Hingga 5
db.t3.micro*	2	Variab	1	Khusus yang dioptimalkan EBS	Hingga 1.536	Hingga 5
db.t2 – kelas instans performa yang dapat melonjak						
db.t2.2xlarge	8	Variab	32	Khusus EBS	—	Sedang
db.t2.xlarge	4	Variab	16	Khusus EBS	—	Sedang
db.t2.large	2	Variab	8	Khusus EBS	—	Sedang
db.t2.medium	2	Variab	4	Khusus EBS	—	Sedang
db.t2.small	1	Variab	2	Khusus EBS	—	Rendah
db.t2.micro	1	Variab	1	Khusus EBS	—	Rendah

* Kelas instans DB ini dapat mendukung performa maksimum selama 30 menit setidaknya sekali setiap 24 jam. Untuk informasi selengkapnya tentang performa dasar jenis instans EC2 yang mendasari, lihat Instans yang dioptimalkan Amazon EBS di Panduan Pengguna Amazon EC2 untuk Instans Linux.

** Kelas instans DB r3.8xlarge tidak memiliki bandwidth EBS khusus dan karenanya tidak menawarkan pengoptimalan EBS. Untuk kelas instans ini, lalu lintas jaringan dan lalu lintas Amazon EBS berbagi antarmuka jaringan 10 gigabit yang sama.

Penyimpanan instans DB Amazon RDS

Instans DB untuk Amazon RDS for Db2, MariaDB, MySQL, PostgreSQL, Oracle, dan Microsoft SQL Server menggunakan volume Amazon Elastic Block Store (Amazon EBS) untuk penyimpanan basis data dan log.

Dalam beberapa kasus, beban kerja basis data Anda mungkin tidak dapat mencapai 100 persen IOPS yang telah Anda sediakan. Untuk informasi selengkapnya, lihat Faktor-faktor yang memengaruhi performa penyimpanan.

Untuk informasi selengkapnya tentang harga penyimpanan instans, lihat Harga Amazon RDS.

Jenis penyimpanan Amazon RDS

Amazon RDS menyediakan tiga jenis penyimpanan: General Purpose SSD (juga dikenal sebagai gp2 dan gp3), Provisioned IOPS SSD (juga dikenal sebagai io1 dan io2 Block Express), dan magnetik (juga dikenal sebagai standar). Ketiganya memiliki karakteristik performa dan harga yang berbeda, artinya Anda dapat menyesuaikan performa penyimpanan dan biaya untuk kebutuhan beban kerja basis data Anda. Anda dapat membuat instans DB RDS for Db2, MySQL, MariaDB, Oracle, dan PostgreSQL dengan penyimpanan hingga 64 tebibyte (TiB). Anda dapat membuat instans DB RDS for SQL Server dengan penyimpanan hingga 16 TiB. Untuk jumlah penyimpanan ini, gunakan jenis penyimpanan SSD IOPS yang Tersedia dan SSD Tujuan Umum. RDS for Db2 hanya mendukung jenis penyimpanan SSD Tujuan Umum gp3 dan jenis penyimpanan SSD IOPS yang Tersedia.

Daftar berikut menjelaskan secara singkat tiga jenis penyimpanan:

- SSD Tujuan Umum – Volume SSD Tujuan Umum menawarkan penyimpanan hemat biaya yang ideal untuk berbagai beban kerja yang berjalan di instans DB ukuran sedang. Penyimpanan Tujuan Umum sangat cocok untuk pengembangan dan pengujian lingkungan.

  Untuk informasi selengkapnya tentang penyimpanan SSD Tujuan Umum, termasuk rentang ukuran penyimpanan, lihat Penyimpanan SSD Tujuan Umum.
- SSD IOPS yang Tersedia – Penyimpanan IOPS yang Tersedia dirancang untuk memenuhi kebutuhan beban kerja intensif I/O, khususnya beban kerja basis data yang memerlukan latensi I/O rendah dan throughput I/O yang konsisten. Penyimpanan IOPS yang Tersedia sangat cocok untuk lingkungan produksi.

  Untuk informasi selengkapnya tentang penyimpanan IOPS yang Tersedia, termasuk rentang ukuran penyimpanannya, lihat Penyimpanan SSD IOPS yang Tersedia.

- Magnetik – Amazon RDS juga mendukung penyimpanan magnetik untuk kompatibilitas mundur. Sebaiknya Anda menggunakan SSD Tujuan Umum atau SSD IOPS yang Tersedia untuk kebutuhan penyimpanan baru. Jumlah penyimpanan maksimum yang diizinkan untuk instans DB pada penyimpanan magnetik lebih kecil dibandingkan jenis penyimpanan lainnya. Untuk informasi selengkapnya, lihat Penyimpanan magnetik.

Jika Anda memilih SSD Tujuan Umum atau SSD IOPS yang Tersedia, tergantung pada mesin yang dipilih dan jumlah penyimpanan yang diminta, Amazon RDS otomatis melakukan striping beberapa volume untuk meningkatkan performa, seperti yang ditunjukkan pada tabel berikut.

Mesin basis data	Ukuran penyimpanan Amazon RDS	Jumlah volume yang disediakan
Db2	Kurang dari 400 GiB	1
Db2	400—65.536 GiB	4
MariaDB, MySQL, dan PostgreSQL	Kurang dari 400 GiB	1
MariaDB, MySQL, dan PostgreSQL	400—65.536 GiB	4
Oracle	Kurang dari 200 GiB	1
Oracle	200—65.536 GiB	4
SQL Server	Setiap	1

Jika Anda mengubah volume SSD Tujuan Umum atau IOPS yang Tersedia, pengubahan tersebut akan melewati urutan status. Saat volume berada pada status `optimizing`, performa volume Anda ada di antara spesifikasi konfigurasi sumber dan target. Performa volume transisi tidak akan kurang dari spesifikasi terendah dari kedua spesifikasi tersebut. Untuk informasi selengkapnya tentang perubahan volume, lihat Memantau kemajuan pengubahan volume di Panduan Pengguna Amazon EC2.

> **Important**
>
> Saat Anda mengubah penyimpanan instans dari satu volume menjadi empat volume, atau saat Anda mengubah instans menggunakan penyimpanan magnetik, Amazon RDS tidak menggunakan fitur Volume Elastis. Sebaliknya, Amazon RDS menyediakan volume baru dan secara transparan memindahkan data dari volume lama ke volume baru. Operasi ini mengonsumsi sejumlah besar IOPS dan throughput dari volume lama dan baru. Bergantung pada ukuran volume dan jumlah beban kerja basis data yang ada selama pengubahan, operasi ini dapat mengonsumsi IOPS dalam jumlah besar, meningkatkan latensi IO secara signifikan, dan memerlukan waktu beberapa jam untuk menyelesaikannya, sementara instans RDS tetap dalam status `Modifying`.

Penyimpanan SSD Tujuan Umum

Penyimpanan SSD Tujuan Umum menawarkan penyimpanan hemat biaya yang dapat diterima untuk sebagian besar beban kerja basis data yang tidak sensitif terhadap latensi.

> **Note**
>
> Instans DB yang menggunakan penyimpanan SSD Tujuan Umum dapat mengalami latensi lebih lama setelah pembuatan replika baca, konversi multi-AZ, dan pemulihan snapshot DB daripada instans yang menggunakan penyimpanan IOPS yang Tersedia. Jika Anda membutuhkan instans DB dengan latensi minimum setelah operasi ini, sebaiknya gunakan Penyimpanan SSD IOPS yang Tersedia.

Amazon RDS menawarkan dua jenis penyimpanan SSD Tujuan Umum: Penyimpanan gp2 dan Penyimpanan gp3.

Penyimpanan gp2

Ketika aplikasi Anda tidak membutuhkan performa penyimpanan yang tinggi, Anda dapat menggunakan penyimpanan gp2 SSD Tujuan Umum. Performa dasar I/O untuk penyimpanan gp2 adalah 3 IOPS untuk setiap GiB, dengan minimal 100 IOPS. Hubungan ini berarti bahwa volume yang lebih besar memiliki performa yang lebih baik. Misalnya, performa dasar untuk satu volume 100-GiB adalah 300 IOPS. Performa dasar untuk satu volume 1.000 GiB adalah 3.000 IOPS. Performa dasar maksimum untuk satu volume gp2 (5334 GiB dan lebih) adalah 16.000 IOPS.

Volume gp2 individual yang berukuran di bawah 1.000 GiB juga memiliki kemampuan untuk melonjak hingga 3.000 IOPS untuk jangka waktu yang lama. Keseimbangan kredit I/O volume menentukan performa lonjakan. Untuk informasi selengkapnya tentang kredit I/O volume, lihat kredit I/O dan performa lonjakan di Panduan Pengguna Amazon EC2. Untuk penjelasan lebih rinci tentang bagaimana kinerja dasar dan saldo kredit I/O mempengaruhi kinerja, lihat posting Memahami burst vs. kinerja dasar dengan Amazon RDS dan gp2 di Blog Database. AWS

Banyak beban kerja tidak pernah mengurangi keseimbangan lonjakan. Namun, beberapa beban kerja dapat menghabiskan keseimbangan kredit penyimpanan lonjakan 3.000 IOPS, sehingga Anda harus merencanakan kapasitas penyimpanan untuk memenuhi kebutuhan beban kerja Anda.

Untuk volume gp2 yang lebih besar dari 1.000 GiB, performa dasar lebih besar dari performa lonjakan. Untuk volume seperti itu, lonjakan tidak relevan karena performa dasar lebih baik daripada performa lonjakan 3.000 IOPS. Namun, untuk instans DB dari mesin dan ukuran tertentu, penyimpanan di-striping menjadi empat volume yang menyediakan empat kali throughput dasar, dan empat kali lipat IOPS lonjakan dari satu volume. Performa penyimpanan untuk volume gp2 pada mesin DB Amazon RDS, termasuk ambang batas, ditunjukkan pada tabel berikut.

Mesin DB	Ukuran penyimpanan RDS	Rentang IOPS dasar	Rentang throughput dasar	IOPS lonjakan
MariaDB, MySQL, dan PostgreSQL	5—399 GIB[1]	100-1197 IOPS	128-250 MiB/dtk	3.000
MariaDB, MySQL, dan PostgreSQL	400—1.335 GiB	1.200-4.005 IOPS	500-1.000 MiB/dtk	12.000
MariaDB, MySQL, dan PostgreSQL	1,336—3,999 GiB	4008-11.997 IOPS	1.000 MiB/dtk	12.000
MariaDB, MySQL, dan PostgreSQL	4.000—65,536 GiB	12.000-64.000 IOPS	1.000 MiB/dtk	N/A[2]

Mesin DB	Ukuran penyimpanan RDS	Rentang IOPS dasar	Rentang throughput dasar	IOPS lonjakan
Oracle	20—199 GiB	100-597 IOPS	128-250 MiB/dtk	3.000
Oracle	200—1.335 GiB	600-4.005 IOPS	500-1.000 MiB/dtk	12.000
Oracle	1,336—3,999 GiB	4008-11.997 IOPS	1.000 MiB/dtk	12.000
Oracle	4.000—65,536 GiB	12.000-64.000 IOPS	1.000 MiB/dtk	N/A[2]
SQL Server	20—333 GiB	100-999 IOPS	128-250 MiB/dtk	3.000
SQL Server	334—999 GiB	1.002-2.997 IOPS	250 MiB/dtk	3.000
SQL Server	1.000—16,384 GiB	3.000-16.000 IOPS	250 MiB/dtk	N/A[2]

> **Note**
>
> [1] Dengan menggunakan AWS Management Console, Anda dapat membuat instans DB dengan ukuran penyimpanan minimum 5 GiB di tingkat Gratis untuk kelas instans db.t3.micro dan db.t4g.micro DB. Jika tidak, ukuran penyimpanan minimum adalah 20 GiB. Batasan ini tidak berlaku untuk API AWS CLI dan RDS.
> [2] Kinerja dasar volume melebihi kinerja burst maksimum.

Penyimpanan gp3

Dengan menggunakan volume penyimpanan gp3 SSD Tujuan Umum, Anda dapat menyesuaikan performa penyimpanan kapasitas penyimpanan secara independen. Performa penyimpanan adalah kombinasi operasi I/O per detik (IOPS) dan seberapa cepat volume penyimpanan dapat melakukan

pembacaan dan penulisan (throughput penyimpanan). Pada volume penyimpanan gp3, Amazon RDS menyediakan performa penyimpanan dasar 3000 IOPS dan 125 MiB/dtk.

Untuk setiap mesin DB RDS kecuali RDS for SQL Server, ketika ukuran penyimpanan untuk volume gp3 mencapai ambang tertentu, performa penyimpanan dasar meningkat menjadi 12.000 IOPS dan 500 MiB/dtk. Ini karena pembagian volume, penyimpanan menggunakan empat volume, bukan satu. RDS for SQL Server tidak mendukung pembagian volume, dan karenanya tidak memiliki nilai ambang batas.

> **Note**
>
> Penyimpanan GP3 SSD Tujuan Umum didukung pada instans DB Single-AZ dan Multi-AZ, dan pada cluster DB multi-AZ. Lihat informasi yang lebih lengkap di Mengonfigurasi dan mengelola deployment Multi-AZ dan the section called “Deployment klaster basis data Multi-AZ”.

Performa penyimpanan untuk volume gp3 pada mesin DB Amazon RDS, termasuk ambang batas, ditunjukkan pada tabel berikut.

Mesin DB	Ukuran penyimpanan	Performa penyimpanan dasar	Rentang IOPS yang Tersedia	Rentang throughput penyimpanan yang disediakan
Db2, MariaDB, MySQL, dan PostgreSQL	Kurang dari 400 GiB	3.000 IOPS/125 MiB/dtk	N/A	N/A
Db2, MariaDB, MySQL, dan PostgreSQL	400 GiB dan lebih tinggi	12.000 IOPS/500 MiB/dtk	12.000–64.000 IOPS	500–4.000 MiB/dtk
Oracle	Kurang dari 200 GiB	3.000 IOPS/125 MiB/dtk	N/A	N/A
Oracle	200 GiB dan lebih tinggi	12.000 IOPS/500 MiB/dtk	12.000–64.000 IOPS	500–4.000 MiB/dtk

Mesin DB	Ukuran penyimpanan	Performa penyimpanan dasar	Rentang IOPS yang Tersedia	Rentang throughput penyimpanan yang disediakan
SQL Server	20—16,384 GiB	3.000 IOPS/125 MiB/dtk	3.000–16.000 IOPS	125–1.000 MiB/dtk

Untuk setiap mesin DB kecuali RDS for SQL Server, Anda dapat menyediakan IOPS tambahan dan throughput penyimpanan saat ukuran penyimpanan berada pada atau di atas nilai ambang batas. Untuk RDS for SQL Server, Anda dapat menyediakan IOPS tambahan dan throughput penyimpanan untuk ukuran penyimpanan yang tersedia. Untuk semua mesin DB, Anda hanya membayar performa penyimpanan tambahan yang disediakan. Untuk informasi selengkapnya, lihat Harga Amazon RDS.

Meskipun tidak bergantung pada ukuran penyimpanan, IOPS yang Tersedia dan throughput penyimpanan yang ditambahkan terkait satu sama lain. Ketika Anda menaikkan IOPS di atas 32.000 untuk MariaDB dan MySQL, nilai throughput penyimpanan secara otomatis meningkat dari 500. MiBps Misalnya, ketika Anda mengatur IOPS ke 40.000 pada RDS untuk MySQL, throughput penyimpanan harus minimal 625. MiBps Peningkatan otomatis tidak dapat dilakukan pada instans DB Oracle, PostgreSQL, dan SQL Server.

Untuk cluster DB multi-AZ, Amazon RDS secara otomatis menetapkan nilai throughput berdasarkan IOPS yang Anda berikan. Anda tidak dapat memodifikasi nilai throughput.

Nilai performa penyimpanan untuk volume gp3 pada RDS memiliki batasan berikut:

- Rasio maksimum throughput penyimpanan IOPS adalah 0,25 untuk semua mesin DB yang didukung.
- Rasio minimum IOPS untuk alokasi penyimpanan (dalam GiB) adalah 0,5 pada RDS for SQL Server. Tidak ada rasio minimum untuk mesin DB lain yang didukung.
- Rasio maksimum IOPS untuk alokasi penyimpanan adalah 500 untuk semua mesin DB yang didukung.
- Jika Anda menggunakan penskalaan otomatis penyimpanan, rasio yang sama antara ambang penyimpanan maksimum dan IOPS (dalam GiB) juga berlaku.

  Untuk informasi selengkapnya tentang penskalaan otomatis penyimpanan, lihat Mengelola kapasitas secara otomatis dengan penskalaan otomatis penyimpanan Amazon RDS.

Penyimpanan SSD IOPS yang Tersedia

Untuk aplikasi produksi yang memerlukan performa I/O yang cepat dan konsisten, kami merekomendasikan penyimpanan IOPS yang Tersedia. Penyimpanan IOPS yang Tersedia adalah jenis penyimpanan yang memberikan performa yang dapat diprediksi, dan latensi rendah yang konsisten. Penyimpanan IOPS yang Tersedia dioptimalkan untuk beban kerja pemrosesan transaksi online (OLTP) yang membutuhkan performa yang konsisten. IOPS yang Tersedia membantu penyelarasan performa beban kerja ini.

Saat membuat instans DB, Anda menentukan tingkat IOPS dan ukuran volumenya. Amazon RDS memberikan tingkat IOPS tersebut untuk instans DB hingga Anda mengubahnya.

Amazon RDS menawarkan dua jenis penyimpanan SSD IOPS yang Disediakan: dan. Penyimpanan io1 penyimpanan io2 Blok Express

Penyimpanan io1

Untuk beban kerja intensif I/O, Anda dapat menggunakan penyimpanan io1 SSD IOPS yang Tersedia dan menghasilkan hingga 256.000 operasi I/O per detik (IOPS). Throughput volume io1 bervariasi berdasarkan jumlah IOPS yang disediakan per volume dan ukuran operasi IO yang dijalankan. Untuk informasi selengkapnya tentang throughput volume io1, lihat Volume IOPS yang Tersedia di Panduan Pengguna Amazon EC2.

Tabel berikut menunjukkan rentang IOPS yang Disediakan dan throughput maksimum untuk setiap mesin database dan rentang ukuran penyimpanan.

Mesin basis data	Rentang ukuran penyimpanan	Rentang IOPS yang Tersedia	Throughput maksimum
Db2, MariaDB, MySQL, dan PostgreSQL	100—399 GiB	1.000–19.950 IOPS	500 MiB/dtk
Db2, MariaDB, MySQL, dan PostgreSQL	400—65.536 GiB	1.000–256.000 IOPS	4.000 MiB/dtk
Oracle	100—199 GiB	1.000–9.950 IOPS	500 MiB/dtk
Oracle	200—65.536 GiB	1.000—256.000 IOPS[1]	4.000 MiB/dtk

Mesin basis data	Rentang ukuran penyimpanan	Rentang IOPS yang Tersedia	Throughput maksimum
SQL Server	20—16,384 GiB	1.000—64.000 IOPS[2]	1.000 MiB/dtk

> **Note**
>
> [1] Untuk Oracle, Anda dapat menyediakan maksimum 256.000 IOPS hanya pada jenis instans r5b.
> [2] Untuk SQL Server, maksimum 64.000 IOPS dijamin hanya pada instans berbasis NITRO yang ada di tipe instans m5*, m6i, r5*, r6i, dan z1d. Jenis instans lain menjamin performa hingga 32.000 IOPS.

Rentang ukuran penyimpanan dan IOPS memiliki batasan berikut:

- Rasio IOPS untuk alokasi penyimpanan (dalam GiB) harus 1-50 pada RDS for SQL Server, dan 0,5-50 pada mesin DB RDS lainnya.
- Jika Anda menggunakan penskalaan otomatis penyimpanan, rasio yang sama antara ambang penyimpanan maksimum dan IOPS (dalam GiB) juga berlaku.

  Untuk informasi selengkapnya tentang penskalaan otomatis penyimpanan, lihat Mengelola kapasitas secara otomatis dengan penskalaan otomatis penyimpanan Amazon RDS.

penyimpanan io2 Blok Express

Untuk beban kerja intensif I/O, Anda dapat menggunakan penyimpanan IOPS SSD io2 Block Express yang disediakan untuk mencapai hingga 256.000 operasi I/O per detik (IOPS). Throughput volume io2 Block Express bervariasi berdasarkan jumlah IOPS yang disediakan per volume dan ukuran operasi IO yang dijalankan.

Semua volume RDS io2 berdasarkan Sistem AWS Nitro adalah volume Blok Ekspres io2 dan memberikan latensi rata-rata sub-milidetik. Instans DB yang tidak didasarkan pada Sistem AWS Nitro adalah volume io2.

Tabel berikut menunjukkan rentang IOPS yang Disediakan dan throughput maksimum untuk setiap mesin database dan rentang ukuran penyimpanan.

Mesin basis data	Rentang ukuran penyimpanan	Rentang IOPS yang Tersedia	Throughput maksimum
Db2, MariaDB, MySQL, dan PostgreSQL	100—65.536 GiB	1.000–256.000 IOPS	4.000 MiB/dtk
Oracle	100—199 GiB	1.000—199.000 IOPS	4.000 MiB/dtk
Oracle	200—65.536 GiB	1.000–256.000 IOPS	4.000 MiB/dtk
SQL Server	20—16,384 GiB	1.000–64.000 IOPS	4.000 MiB/dtk

Rentang ukuran penyimpanan dan IOPS memiliki batasan berikut:

- Rasio IOPS terhadap penyimpanan yang dialokasikan (dalam GiB) harus tidak lebih dari 1000:1. Untuk instans DB yang tidak didasarkan pada Sistem AWS Nitro, rasionya adalah 500:1.
- IOPS maksimum dapat disediakan dengan volume 256 GiB dan lebih besar (1.000 IOPS × 256 GiB = 256.000 IOPS). Untuk instans DB yang tidak didasarkan pada Sistem AWS Nitro, IOPS maksimum dicapai pada 512 GiB (500 IOPS x 512 GiB = 256.000 IOPS).
- Throughput diskalakan secara proporsional hingga 0,256 MiB/dtk per IOPS yang tersedia. Throughput maksimum 4.000 MiB/s dapat dicapai pada 256.000 IOPS dengan ukuran I/O 16-KiB dan 16.000 IOPS atau lebih tinggi dengan ukuran 256-KiB I/O. Untuk instans DB yang tidak didasarkan pada Sistem AWS Nitro, throughput maksimum 2.000 MiB/s dapat dicapai pada 128.000 IOPS dengan ukuran I/O 16-KiB.
- Jika Anda menggunakan penskalaan otomatis penyimpanan, rasio yang sama antara ambang penyimpanan maksimum dan IOPS (dalam GiB) juga berlaku. Untuk informasi selengkapnya tentang penskalaan otomatis penyimpanan, lihat Mengelola kapasitas secara otomatis dengan penskalaan otomatis penyimpanan Amazon RDS.

Volume Amazon RDS io2 Block Express tersedia sebagai berikut: Wilayah AWS

- Asia Pasifik (Hong Kong)
- Asia Pasifik (Mumbai)
- Asia Pasifik (Seoul)
- Asia Pasifik (Singapura)

- Asia Pasifik (Sydney)
- Asia Pasifik (Tokyo)
- Kanada (Pusat)
- Eropa (Frankfurt)
- Eropa (Irlandia)
- Europe (London)
- Eropa (Stockholm)
- Timur Tengah (Bahrain)
- AS Timur (Ohio)
- AS Timur (Virginia Utara)
- AS Barat (California Utara)
- AS Barat (Oregon)

Menggabungkan penyimpanan IOPS yang Tersedia dengan deployment Multi-AZ atau replika baca

Untuk kasus penggunaan OLTP produksi, sebaiknya Anda menggunakan deployment Multi-AZ untuk peningkatan toleransi kesalahan pada penyimpanan IOPS yang Tersedia guna mendapatkan performa yang cepat dan dapat diprediksi.

Anda juga dapat menggunakan penyimpanan SSD IOPS yang Tersedia dengan replika baca untuk MySQL, MariaDB, atau PostgreSQL. Jenis penyimpanan untuk replika baca tidak tergantung pada yang ada di instans DB primer. Misalnya, Anda dapat menggunakan SSD Tujuan Umum untuk replika baca dengan instans DB primer yang menggunakan penyimpanan SSD IOPS yang Tersedia untuk mengurangi biaya. Namun, performa replika baca Anda dalam kasus ini mungkin berbeda dengan konfigurasi ketika instans DB primer dan replika baca menggunakan penyimpanan SSD IOPS yang Tersedia.

Biaya penyimpanan IOPS yang Tersedia

Dengan penyimpanan IOPS yang Tersedia, Anda dikenakan biaya untuk sumber daya yang disediakan, baik jika Anda menggunakannya dalam satu bulan atau tidak.

Untuk informasi selengkapnya tentang harga, lihat Harga Amazon RDS.

Mendapatkan performa terbaik dari penyimpanan SSD IOPS yang Tersedia untuk Amazon RDS

Jika beban kerja Anda dibatasi I/O, menggunakan penyimpanan SSD IOPS yang Tersedia dapat meningkatkan jumlah permintaan I/O yang dapat diproses secara bersamaan oleh sistem. Peningkatan konkurensi memungkinkan penurunan latensi karena permintaan I/O mengurangi waktu dalam antrean. Penurunan latensi memungkinkan penerapan basis data yang lebih cepat, sehingga meningkatkan waktu respons dan memungkinkan throughput basis data yang lebih tinggi.

Penyimpanan SSD IOPS yang Tersedia memberikan cara untuk menyimpan kapasitas I/O dengan menentukan IOPS. Namun, seperti atribut kapasitas sistem lainnya, throughput maksimumnya saat dibebani dibatasi oleh sumber daya yang dikonsumsi terlebih dahulu. Sumber daya tersebut dapat berupa bandwith jaringan, CPU, memori, atau sumber daya internal basis data.

Untuk informasi selengkapnya tentang memaksimalkan volume IOPS yang Tersedia, lihat Performa volume EBS Amazon.

Membandingkan jenis penyimpanan solid-state drive (SSD)

Tabel berikut menunjukkan kasus penggunaan dan karakteristik performa untuk volume penyimpanan SSD yang digunakan oleh Amazon RDS.

Karakteristik	IOPS yang Disediakan (io2 Block Express)	IOPS yang Tersedia (io1)	Tujuan Umum (gp3)	Tujuan Umum (gp2)
Deskripsi	Kinerja tertinggi dalam portofolio penyimpanan RDS (IOPS, throughput, latency) Dirancang untuk beban kerja transaksional yang sensitif terhadap latensi	Performa penyimpanan yang konsisten (IOPS, throughput, latensi) Dirancang untuk beban kerja transaksional yang sensitif terhadap latensi	Fleksibilitas dalam penyediaan penyimpanan, IOPS, dan throughput secara independen Menyeimbangkan performa harga untuk berbagai macam	Memberikan IOPS yang dapat melonjak Menyeimbangkan performa harga untuk berbagai macam beban kerja transaksional

Karakteristik	IOPS yang Disediakan (io2 Block Express)	IOPS yang Tersedia (io1)	Tujuan Umum (gp3)	Tujuan Umum (gp2)
			beban kerja transaksional	
Kasus penggunaan	Beban kerja transaksional penting bisnis yang membutuhkan latensi sub-milidetik dan kinerja IOPS yang berkelanjutan hingga 256.000 IOPS	Beban kerja transaksional yang memerlukan performa IOPS berkelanjutan hingga 256.000 IOPS	Berbagai beban kerja yang berjalan pada basis data relasional berukuran sedang di lingkungan pengembangan/pengujian	Berbagai beban kerja yang berjalan pada basis data relasional berukuran sedang di lingkungan pengembangan/pengujian
Latensi	Sub-milidetik, disediakan secara konsisten 99,9% dari waktu	Satu digit milidetik, disediakan secara konsisten 99,9% dari waktu	Satu digit milidetik, disediakan secara konsisten 99% dari waktu	Satu digit milidetik, disediakan secara konsisten 99% dari waktu
Ukuran volume	100—65.536 GiB (16.384 GiB pada RDS untuk SQL Server)	100—65.536 GiB (20—16.384 GiB pada RDS untuk SQL Server)	20—65.536 GiB (16.384 GiB pada RDS untuk SQL Server)	20—65.536 GiB (16.384 GiB pada RDS untuk SQL Server)

Karakteristik	IOPS yang Disediakan (io2 Block Express)	IOPS yang Tersedia (io1)	Tujuan Umum (gp3)	Tujuan Umum (gp2)
IOPS maksimum	256.000 (64.000 pada RDS for SQL Server)	256.000 (64.000 pada RDS for SQL Server)	64.000 (16.000 pada RDS for SQL Server)	64.000 (16.000 pada RDS for SQL Server) **Note** Anda tidak dapat menyediakan IOPS secara langsung di penyimpanan gp2. IOPS bervariasi dengan ukuran penyimpanan yang dialokasikan.

Karakteristik	IOPS yang Disediakan (io2 Block Express)	IOPS yang Tersedia (io1)	Tujuan Umum (gp3)	Tujuan Umum (gp2)
Throughput maksimum	Menskalakan berdasarkan IOPS yang Tersedia hingga 4.000 MB/dtk Throughput diskalakan secara proporsional hingga 0,256 MiB/dtk per IOPS yang tersedia. Throughput maksimum 4.000 MiB/s dapat dicapai pada 256.000 IOPS dengan ukuran I/O 16-KiB dan 16.000 IOPS atau lebih tinggi dengan ukuran 256-KiB I/O. Untuk contoh yang tidak didasarkan pada Sistem AWS Nitro, throughput maksimum 2.000 MiB/s dapat dicapai pada	Menskalakan berdasarkan IOPS yang Tersedia hingga 4.000 MB/dtk	Menyediakan throughput tambahan hingga 4.000 MB/dtk (1000 MB/dtk pada RDS for SQL Server	1000 MB/dtk (250 MB/dtk pada RDS for SQL Server)

Karakteristik	IOPS yang Disediakan (io2 Block Express)	IOPS yang Tersedia (io1)	Tujuan Umum (gp3)	Tujuan Umum (gp2)
	128.000 IOPS dengan ukuran I/O 16-KiB.			
AWS CLI dan nama API RDS	io2	io1	gp3	gp2

Penyimpanan magnetik

Amazon RDS juga mendukung penyimpanan magnetik untuk kompatibilitas mundur. Sebaiknya Anda menggunakan SSD Tujuan Umum atau SSD IOPS yang Tersedia untuk kebutuhan penyimpanan baru. Berikut ini adalah beberapa batasan untuk penyimpanan magnetik:

- Tidak mengizinkan Anda menskalakan penyimpanan saat menggunakan mesin basis data SQL Server.
- Tidak mendukung penskalaan otomatis penyimpanan.
- Tidak mendukung volume elastis.
- Terbatas hingga ukuran maksimum 3 TiB.
- Terbatas hingga maksimum 1.000 IOPS.

Volume log khusus (DLV)

Anda dapat menggunakan volume log khusus (DLV) untuk instans DB yang menggunakan penyimpanan IOPS Tertentu (PIOPS) dengan menggunakan konsol Amazon RDS, AWS CLI atau Amazon RDS API. DLV memindahkan log transaksi database PostgreSQL dan log redo MySQL/MariaDB dan log biner ke volume penyimpanan yang terpisah dari volume yang berisi tabel database. DLV membuat pencatatan log penulisan transaksi menjadi lebih efisien dan konsisten. DLV ideal untuk basis data dengan penyimpanan besar yang dialokasikan, kebutuhan I/O per detik (IOPS) tinggi, atau beban kerja yang sensitif terhadap latensi.

DLV didukung untuk penyimpanan PIOPS (io1 dan io2 Block Express), dan dibuat dengan ukuran tetap 1.000 GiB dan 3.000 IOPS Provisioned.

Amazon RDS mendukung DLV secara keseluruhan Wilayah AWS untuk versi berikut:

- MariaDB 10.6.7 dan versi 10 yang lebih tinggi
- MySQL 8.0.28 dan versi 8 yang lebih tinggi
- PostgreSQL 13.10 dan versi 13 yang lebih tinggi, 14.7 dan versi 14 yang lebih tinggi, serta 15.2 dan versi 15 yang lebih tinggi

RDS mendukung DLV dengan deployment Multi-AZ. Saat Anda mengubah atau membuat instans Multi-AZ, DLV dibuat untuk instans primer dan sekunder.

RDS mendukung DLV dengan replika baca. Jika instans DB primer memiliki DLV yang aktif, semua replika baca yang dibuat setelah mengaktifkan DLV juga akan memiliki DLV. Setiap replika baca yang dibuat sebelum beralih ke DLV tidak akan mengaktifkan DLV kecuali diubah secara eksplisit untuk mengaktifkannya. Sebaiknya semua replika baca yang dilampirkan ke instans primer sebelum DLV diaktifkan juga diubah secara manual untuk memiliki DLV.

Setelah Anda mengubah pengaturan DLV untuk instans DB, instans DB harus di-boot ulang.

Untuk informasi tentang mengaktifkan DLV, lihat Menggunakan volume log khusus (DLV).

Memantau performa penyimpanan

Amazon RDS menyediakan beberapa metrik yang dapat Anda gunakan untuk menentukan cara kerja instans DB Anda. Anda dapat melihat metrik tersebut di halaman ringkasan untuk instans Anda di Konsol Manajemen Amazon RDS. Anda juga dapat menggunakan Amazon CloudWatch untuk memantau metrik ini. Untuk informasi selengkapnya, lihat Melihat metrik di konsol Amazon RDS. Pemantauan yang Ditingkatkan menyediakan metrik I/O yang lebih detail; untuk informasi selengkapnya, lihat Memantau metrik OS dengan Pemantauan yang Disempurnakan.

Metrik berikut berguna untuk memantau penyimpanan pada instans DB Anda:

- IOPS – Jumlah operasi I/O yang diselesaikan setiap detik. Metrik ini dilaporkan sebagai rata-rata IOPS untuk interval waktu tertentu. Amazon RDS melaporkan IOPS baca dan tulis secara terpisah pada interval 1 menit. Total IOPS adalah jumlah dari IOPS baca dan tulis. Nilai tipikal untuk IOPS berkisar dari nol hingga puluhan ribu per detik.
- Latensi – Waktu yang berlalu antara pengiriman permintaan I/O dan penyelesaiannya. Metrik ini dilaporkan sebagai rata-rata latensi untuk interval waktu tertentu. Amazon RDS melaporkan latensi baca dan tulis secara terpisah pada interval 1 menit. Nilai tipikal untuk latensi adalah dalam milidetik (md).

- Throughput – Jumlah byte setiap detik yang ditransfer ke atau dari disk. Metrik ini dilaporkan sebagai rata-rata throughput untuk interval waktu tertentu. Amazon RDS melaporkan throughput baca dan tulis secara terpisah pada interval 1 menit menggunakan unit megabyte per detik (MB/dtk). Nilai tipikal untuk throughput berkisar dari nol hingga bandwith maksimum saluran I/O.
- Kedalaman Antrean – Jumlah permintaan I/O di dalam antrean yang menunggu dilayani. Ini adalah permintaan I/O yang telah dikirimkan oleh aplikasi tetapi belum dikirimkan ke perangkat karena perangkat sedang sibuk melayani permintaan I/O lainnya. Waktu yang dihabiskan untuk menunggu di dalam antrean merupakan komponen latensi dan waktu layanan (tidak tersedia sebagai metrik). Metrik ini dilaporkan sebagai rata-rata kedalaman antrean untuk interval waktu tertentu. Amazon RDS melaporkan kedalaman antrean pada interval 1 menit. Nilai tipikal untuk kedalaman antrean berkisar dari nol hingga beberapa ratus.

Nilai IOPS yang terukur tidak bergantung pada ukuran operasi I/O individual. Artinya, ketika Anda mengukur performa I/O, pastikan untuk memeriksa throughput instans, bukan hanya jumlah operasi I/O.

Faktor-faktor yang memengaruhi performa penyimpanan

Aktivitas sistem, beban kerja basis data, dan kelas instans DB dapat memengaruhi performa penyimpanan.

Aktivitas sistem

Aktivitas terkait sistem berikut mengonsumsi kapasitas I/O dan mungkin menurunkan performa instans DB saat dalam proses:

- Pembuatan siaga Multi-AZ
- Pembuatan replika baca
- Mengubah jenis penyimpanan

Beban kerja basis data

Dalam beberapa kasus, basis data atau desain aplikasi Anda menimbulkan masalah konkurensi, penguncian, atau bentuk lain dari pertentangan basis data. Dalam hal ini, Anda mungkin tidak dapat menggunakan semua bandwidth yang disediakan secara langsung. Selain itu, Anda mungkin menghadapi situasi terkait beban kerja berikut ini:

- Batas throughput jenis instans yang mendasarinya tercapai.

- Kedalaman antrean selalu kurang dari 1 karena aplikasi Anda tidak cukup mendorong operasi I/O.
- Anda mengalami pertentangan kueri dalam basis data meskipun beberapa kapasitas I/O tidak digunakan.

Dalam beberapa kasus, tidak ada sumber daya sistem yang berada pada atau mendekati batas, dan menambahkan thread tidak akan menambah tingkat transaksi basis data. Dalam kasus demikian, kemacetan kemungkinan besar merupakan pertentangan dalam basis data. Bentuk yang paling umum adalah pertentangan kunci baris dan kunci halaman indeks, tetapi ada banyak kemungkinan lainnya. Jika Anda mengalami situasi ini, silakan minta saran dari pakar penyetelan performa basis data.

Kelas instans DB

Untuk mendapatkan performa maksimal dari instans DB Amazon RDS Anda, pilih jenis instans generasi terbaru dengan bandwidth yang cukup untuk mendukung jenis penyimpanan Anda. Misalnya, Anda dapat memilih instans yang dioptimalkan Amazon EBS dan instans dengan konektivitas jaringan 10 gigabit.

> **Important**
>
> Bergantung pada kelas instans yang Anda gunakan, Anda mungkin melihat performa IOPS yang lebih rendah daripada performa maksimum yang dapat Anda sediakan dengan RDS. Untuk informasi spesifik tentang performa IOPS untuk kelas instans DB, lihat Instans yang dioptimalkan Amazon EBS di Panduan Pengguna Amazon EC2. Sebaiknya Anda menentukan IOPS maksimum untuk kelas instans sebelum menetapkan nilai IOPS yang Tersedia untuk instans DB Anda.

Anda sangat dianjurkan menggunakan instans generasi terbaru agar bisa mendapatkan performa terbaik. Instans DB generasi sebelumnya juga dapat memiliki penyimpanan maksimum yang lebih rendah.

Beberapa sistem file 32-bit yang lebih lama mungkin memiliki kapasitas penyimpanan yang lebih rendah. Untuk menentukan kapasitas penyimpanan instans DB Anda, Anda dapat menggunakan AWS CLI perintah describe-valid-db-instance-modifikasi.

Daftar berikut menunjukkan penyimpanan maksimum yang dapat diskalakan oleh sebagian besar kelas instans DB untuk setiap mesin basis data:

- Db2 – 64 TiB
- MariaDB – 64 TiB
- Microsoft SQL Server - 64 TiB
- MySQL – 64 TiB
- Oracle – 64 TiB
- PostgreSQL – 64 TiB

Tabel berikut menunjukkan beberapa pengecualian untuk penyimpanan maksimum (dalam TiB). Semua instans DB RDS for Microsoft SQL Server memiliki penyimpanan maksimum 16 TiB, sehingga tidak ada entri untuk SQL Server.

Kelas instans	Db2	MariaDB	MySQL	Oracle	PostgreSQL
db.m3 – kelas instans standar					
db.t4g – kelas instans performa yang dapat melonjak					
db.t4g.medium	N/A	16	16	N/A	32
db.t4g.small	N/A	16	16	N/A	16
db.t4g.micro	N/A	6	6	N/A	6
db.t3 – kelas instans performa yang dapat melonjak					
db.t3.medium	32	16	16	32	32
db.t3.small	32	16	16	32	16
db.t3.micro	N/A	6	6	32	6
db.t2 – kelas instans performa yang dapat melonjak					

Untuk detail selengkapnya tentang semua kelas instans yang didukung, lihat Instans DB generasi sebelumnya.

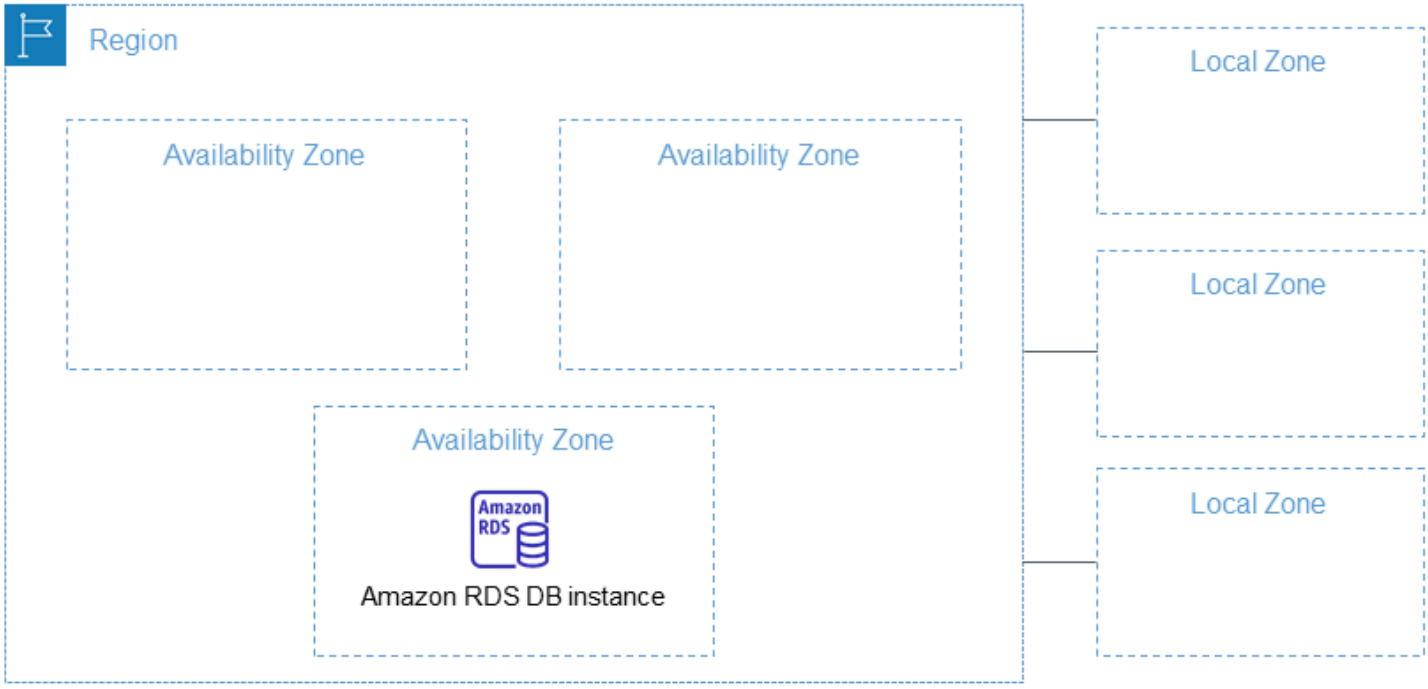
Wilayah, Zona Ketersediaan, dan Zona Lokal

Sumber daya komputasi cloud Amazon di-hosting di beberapa lokasi di seluruh dunia. Lokasi ini terdiri dari AWS Wilayah, Availability Zone, dan Local Zones. Setiap Wilayah AWS adalah wilayah geografis yang terpisah. Setiap AWS Wilayah memiliki beberapa lokasi terisolasi yang dikenal sebagai Availability Zone.

> **Note**
>
> Untuk informasi tentang menemukan Availability Zone untuk suatu AWS Wilayah, lihat Menjelaskan Availability Zone Anda di dokumentasi Amazon EC2.

Dengan menggunakan Zona Lokal, Anda dapat menempatkan sumber daya, seperti komputasi dan penyimpanan, di beberapa lokasi yang lebih dekat dengan pengguna Anda. Amazon RDS memungkinkan Anda menempatkan sumber daya, seperti instans DB, dan data di beberapa lokasi. Sumber daya tidak direplikasi di seluruh AWS Wilayah kecuali Anda melakukannya secara khusus.

Amazon beroperasi state-of-the-art, pusat data yang sangat tersedia. Meskipun jarang, kegagalan yang memengaruhi ketersediaan instans DB yang berada di lokasi yang sama dapat terjadi. Jika Anda meng-hosting semua instans DB di satu lokasi yang terpengaruh oleh kegagalan tersebut, tidak satu pun instans DB Anda akan tersedia.



Penting untuk diingat bahwa setiap AWS Wilayah sepenuhnya independen. Aktivitas Amazon RDS apa pun yang Anda lakukan (misalnya, membuat instance database atau mencantumkan instance database yang tersedia) hanya berjalan di Wilayah default Anda saat ini. AWS AWS Wilayah default dapat diubah di konsol, atau dengan mengatur variabel `AWS_DEFAULT_REGION`lingkungan. Atau dapat diganti dengan menggunakan `--region` parameter dengan (). AWS Command Line Interface AWS CLI Untuk informasi selengkapnya, lihat Mengonfigurasi AWS Command Line Interface, khususnya bagian tentang variabel lingkungan dan opsi baris perintah.

Amazon RDS mendukung AWS Wilayah khusus yang disebut AWS GovCloud (US). Hal ini dirancang untuk memungkinkan lembaga pemerintah AS dan pelanggan memindahkan beban kerja yang lebih sensitif ke cloud. Wilayah AWS GovCloud (US) memenuhi persyaratan peraturan dan kepatuhan khusus pemerintah AS. Untuk informasi lebih lanjut, lihat Apa itu AWS GovCloud (US)?

Untuk membuat atau bekerja dengan instans Amazon RDS DB di AWS Wilayah tertentu, gunakan titik akhir layanan regional yang sesuai.

AWS Daerah

Setiap AWS Wilayah dirancang untuk diisolasi dari AWS Wilayah lain. Rancangan ini mencapai toleransi kesalahan dan stabilitas sebesar mungkin.

Saat Anda melihat sumber daya, Anda hanya melihat sumber daya yang terkait dengan AWS Wilayah yang Anda tentukan. Ini karena AWS Wilayah terisolasi satu sama lain, dan kami tidak secara otomatis mereplikasi sumber daya di seluruh AWS Wilayah.

Ketersediaan wilayah

Tabel berikut menunjukkan AWS Wilayah tempat Amazon RDS saat ini tersedia dan titik akhir untuk setiap Wilayah.

Nama Wilayah	Wilayah	Titik Akhir	Protokol	
AS Timur (Ohio)	us–east-2	rds.us-east-2.amazonaws.com	HTTPS	
		rds-fips.us-east-2.api.aws	HTTPS	
		rds.us-east-2.api.aws	HTTPS	
		rds-fips.us-east-2.amazonaws.com	HTTPS	

Nama Wilayah	Wilayah	Titik Akhir	Protokol
AS Timur (Virginia Utara)	us-east-1	rds.us-east-1.amazonaws.com	HTTPS
		rds-fips.us-east-1.api.aws	HTTPS
		rds-fips.us-east-1.amazonaws.com	HTTPS
		rds.us-east-1.api.aws	HTTPS
AS Barat (California Utara)	us-west-1	rds.us-west-1.amazonaws.com	HTTPS
		rds.us-west-1.api.aws	HTTPS
		rds-fips.us-west-1.amazonaws.com	HTTPS
		rds-fips.us-west-1.api.aws	HTTPS
AS Barat (Oregon)	us-west-2	rds.us-west-2.amazonaws.com	HTTPS
		rds-fips.us-west-2.amazonaws.com	HTTPS
		rds.us-west-2.api.aws	HTTPS
		rds-fips.us-west-2.api.aws	HTTPS
Afrika (Cape Town)	af-south-1	rds.af-south-1.amazonaws.com	HTTPS
		rds.af-south-1.api.aws	HTTPS
Asia Pasifik (Hong Kong)	ap-east-1	rds.ap-east-1.amazonaws.com	HTTPS
		rds.ap-east-1.api.aws	HTTPS
Asia Pasifik (Hyderabad)	ap-south-2	rds.ap-south-2.amazonaws.com	HTTPS
		rds.ap-south-2.api.aws	HTTPS

Nama Wilayah	Wilayah	Titik Akhir	Protokol
Asia Pasifik (Jakarta)	ap-southeast-3	rds.ap-southeast-3.amazonaws.com rds.ap-southeast-3.api.aws	HTTPS HTTPS
Asia Pasifik (Melbourne)	ap-southeast-4	rds.ap-southeast-4.amazonaws.com rds.ap-southeast-4.api.aws	HTTPS HTTPS
Asia Pasifik (Mumbai)	ap-south-1	rds.ap-south-1.amazonaws.com rds.ap-south-1.api.aws	HTTPS HTTPS
Asia Pasifik (Osaka)	ap-northeast-3	rds.ap-northeast-3.amazonaws.com rds.ap-northeast-3.api.aws	HTTPS HTTPS
Asia Pasifik (Seoul)	ap-northeast-2	rds.ap-northeast-2.amazonaws.com rds.ap-northeast-2.api.aws	HTTPS HTTPS
Asia Pasifik (Singapura)	ap-southeast-1	rds.ap-southeast-1.amazonaws.com rds.ap-southeast-1.api.aws	HTTPS HTTPS
Asia Pasifik (Sydney)	ap-southeast-2	rds.ap-southeast-2.amazonaws.com rds.ap-southeast-2.api.aws	HTTPS HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	rds.ap-northeast-1.amazonaws.com rds.ap-northeast-1.api.aws	HTTPS HTTPS

Nama Wilayah	Wilayah	Titik Akhir	Protokol
Kanada (Pusat)	ca-central-1	rds.ca-central-1.amazonaws.com	HTTPS
		rds.ca-central-1.api.aws	HTTPS
		rds-fips.ca-central-1.api.aws	HTTPS
		rds-fips.ca-central-1.amazonaws.com	HTTPS
Kanada Barat (Calgary)	ca-west-1	rds.ca-west-1.amazonaws.com	HTTPS
		rds-fips.ca-west-1.amazonaws.com	HTTPS
Eropa (Frankfurt)	eu-central-1	rds.eu-central-1.amazonaws.com	HTTPS
		rds.eu-central-1.api.aws	HTTPS
Eropa (Irlandia)	eu-west-1	rds.eu-west-1.amazonaws.com	HTTPS
		rds.eu-west-1.api.aws	HTTPS
Eropa (London)	eu-west-2	rds.eu-west-2.amazonaws.com	HTTPS
		rds.eu-west-2.api.aws	HTTPS
Eropa (Milan)	eu-south-1	rds.eu-south-1.amazonaws.com	HTTPS
		rds.eu-south-1.api.aws	HTTPS
Eropa (Paris)	eu-west-3	rds.eu-west-3.amazonaws.com	HTTPS
		rds.eu-west-3.api.aws	HTTPS
Eropa (Spanyol)	eu-south-2	rds.eu-south-2.amazonaws.com	HTTPS
		rds.eu-south-2.api.aws	HTTPS

Nama Wilayah	Wilayah	Titik Akhir	Protokol
Eropa (Stockholm)	eu-north-1	rds.eu-north-1.amazonaws.com rds.eu-north-1.api.aws	HTTPS HTTPS
Eropa (Zürich)	eu-central-2	rds.eu-central-2.amazonaws.com rds.eu-central-2.api.aws	HTTPS HTTPS
Israel (Tel Aviv)	il-central-1	rds.il-central-1.amazonaws.com rds.il-central-1.api.aws	HTTPS HTTPS
Timur Tengah (Bahrain)	me-south-1	rds.me-south-1.amazonaws.com rds.me-south-1.api.aws	HTTPS HTTPS
Timur Tengah (UAE)	me-central-1	rds.me-central-1.amazonaws.com rds.me-central-1.api.aws	HTTPS HTTPS
Amerika Selatan (Sao Paulo)	sa-east-1	rds.sa-east-1.amazonaws.com rds.sa-east-1.api.aws	HTTPS HTTPS
AWS GovCloud (AS-Timur)	us-gov-east-1	rds.us-gov-east-1.amazonaws.com rds.us-gov-east-1.api.aws	HTTPS HTTPS
AWS GovCloud (AS-Barat)	us-gov-west-1	rds.us-gov-west-1.amazonaws.com rds.us-gov-west-1.api.aws	HTTPS HTTPS

Jika Anda tidak menentukan titik akhir secara eksplisit, titik akhir AS Barat (Oregon) menjadi default.

Saat Anda bekerja dengan instans DB menggunakan operasi AWS CLI atau API, pastikan Anda menentukan titik akhir regionalnya.

Zona Ketersediaan

Saat membuat instans DB, Anda dapat memilih Zona Ketersediaan atau meminta Amazon RDS memilih untuk Anda secara acak. Availability Zone diwakili oleh kode AWS Region diikuti oleh pengidentifikasi huruf (misalnya,`us-east-1a`).

Gunakan perintah describe-availability-zonesAmazon EC2 sebagai berikut untuk menjelaskan Availability Zone dalam Wilayah tertentu yang diaktifkan untuk akun Anda.

```
aws ec2 describe-availability-zones --region region-name
```

Misalnya, untuk mendeskripsikan Zona Ketersediaan dalam Wilayah AS Timur (Virginia Utara) (us-east-1) yang diaktifkan untuk akun Anda, jalankan perintah berikut:

```
aws ec2 describe-availability-zones --region us-east-1
```

Anda tidak dapat memilih Zona Ketersediaan untuk instans DB primer dan sekunder dalam deployment Multi-AZ DB. Amazon RDS memilihkannya untuk Anda secara acak. Untuk informasi selengkapnya tentang deployment Multi-AZ, lihat Mengonfigurasi dan mengelola deployment Multi-AZ.

> **Note**
>
> Pilihan acak Zona Ketersediaan oleh RDS tidak menjamin distribusi instans DB antara Zona Ketersediaan yang merata dalam satu akun atau grup subnet DB. Anda dapat meminta AZ tertentu ketika membuat atau memodifikasi instans AZ Tunggal, dan Anda dapat menggunakan grup subnet DB yang lebih spesifik untuk instans Multi-AZ. Lihat informasi selengkapnya di Membuat instans DB Amazon RDS dan Memodifikasi instans DB Amazon RDS.

Zona Lokal

Zona Lokal adalah perpanjangan dari AWS Wilayah yang secara geografis dekat dengan pengguna Anda. Anda dapat memperluas VPC dari Wilayah AWS induk ke Zona Lokal. Untuk melakukannya,

buat subnet baru dan tetapkan ke Zona Lokal AWS . Saat membuat subnet di Zona Lokal, VPC Anda diperluas ke Zona Lokal tersebut. Subnet di Zona Lokal beroperasi sama seperti subnet lain di VPC Anda.

Saat membuat instans DB, Anda dapat memilih subnet di Zona Lokal. Local Zone memiliki koneksi sendiri ke internet dan mendukung AWS Direct Connect. Dengan demikian, sumber daya yang dibuat di Zona Lokal dapat melayani pengguna lokal dengan komunikasi latensi yang sangat rendah. Untuk informasi selengkapnya, lihat Zona Lokal AWS.

Zona Lokal diwakili oleh kode AWS Wilayah diikuti oleh pengidentifikasi yang menunjukkan lokasi, misalnya`aus-west-2-lax-1a`.

> **Note**
>
> Zona Lokal tidak dapat disertakan dalam deployment Multi-AZ.

Cara menggunakan Zona Lokal

1. Aktifkan Zona Lokal pada konsol Amazon EC2.

   Untuk informasi selengkapnya, lihat Mengaktifkan Zona Lokal dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.
2. Buat subnet di Zona Lokal.

   Untuk informasi selengkapnya, lihat Membuat subnet di VPC Anda dalam Panduan Pengguna Amazon VPC.
3. Buat grup subnet DB di Zona Lokal.

   Saat Anda membuat grup subnet DB, pilih grup Zona Ketersediaan untuk Zona Lokal.

   Lihat informasi selengkapnya di Membuat klaster DB dalam VPC.
4. Buat instans DB yang menggunakan grup subnet DB di Zona Lokal.

   Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.

Important

Saat ini, satu-satunya Zona AWS Lokal di mana Amazon RDS tersedia adalah Los Angeles di Wilayah Barat AS (Oregon).

Fitur yang didukung di Amazon RDS oleh Wilayah AWS dan mesin DB

Dukungan untuk fitur dan opsi Amazon RDS bervariasi di seluruh Wilayah AWS dan versi spesifik dari setiap mesin DB. Untuk mengidentifikasi dukungan dan ketersediaan versi mesin DB RDS dalam versi Wilayah AWS tertentu, Anda dapat menggunakan bagian berikut ini.

Fitur Amazon RDS berbeda dari fitur dan opsi asli mesin. Untuk informasi selengkapnya tentang fitur dan opsi asli mesin, lihat Fitur asli mesin.

Topik

Konvensi tabel

Tabel-tabel di bagian fitur tersebut menggunakan pola berikut untuk menentukan nomor versi dan tingkat ketersediaan:

- Versi x.y – Hanya tersedia versi tertentu saja
- Versi x.y dan yang lebih tinggi – Versi yang ditentukan dan semua versi minor yang lebih tinggi dari versi utamanya didukung. Misalnya, “versi 10.11 dan yang lebih tinggi” berarti versi 10.11, 10.11.1, dan 10.12 tersedia.
- — – Fitur tidak tersedia saat ini untuk mesin DB RDS yang dipilih atau yang Wilayah AWS ditentukan.

Referensi cepat fitur

Tabel referensi cepat berikut mencantumkan setiap fitur dan mesin DB RDS yang tersedia. Ketersediaan wilayah dan versi tertentu ditampilkan di bagian fitur selanjutnya.

Fitur	RDS for Db2	RDS for MariaDB	RDS for MySQL	RDS for Oracle	RDS for PostgreSQL	RDS for SQL Server
Deploy t Blue/ Gree n	–	Tersedia	Tersedia	–	Tersedia	–
Penca an otoma lintas wilaya	Tersedia	Tersedia	Tersedia	Tersedia	Tersedia	Tersedia
Replik baca lintas Wilaya	–	Tersedia	Tersedia	Tersedia	Tersedia	Tersedia
Aliran aktivita basis data	–	–	–	Tersedia	–	Tersedia

Fitur	RDS for Db2	RDS for MariaDB	RDS for MySQL	RDS for Oracle	RDS for PostgreSQL	RDS for SQL Server
Mode dual-stack	–	Tersedia	Tersedia	Tersedia	Tersedia	Tersedia
Ekspo Snaps ke Amazo S3	–	Tersedia	Tersedia	–	Tersedia	–
AWS Identit and Acces Manag t (IAM) otentik i basis data	–	Tersedia	Tersedia	–	Tersedia	–
Autent si Kerbe	Tersedia	–	Tersedia	Tersedia	Tersedia	Tersedia
Klaste DB Multi-AZ	–	–	Tersedia	–	Tersedia	–
Wawa Perfor	–	Tersedia	Tersedia	Tersedia	Tersedia	Tersedia

Fitur	RDS for Db2	RDS for MariaDB	RDS for MySQL	RDS for Oracle	RDS for PostgreSQL	RDS for SQL Server
RDS Custor	–	–	–	Tersedia	–	Tersedia
Proksi RDS	–	Tersedia	Tersedia	–	Tersedia	Tersedia
Integra Secret Manag	Tersedia	Tersedia	Tersedia	Tersedia	Tersedia	Tersedia

Deployment Blue/Green

Deployment blue/green menyalin lingkungan basis data produksi di lingkungan penahapan yang terpisah dan tersinkron. Dengan menggunakan Deployment Blue/Green Amazon RDS, Anda dapat membuat perubahan pada basis data di lingkungan penahapan tanpa memengaruhi lingkungan produksi. Misalnya, Anda dapat meningkatkan versi mesin DB besar atau kecil, mengubah parameter basis data, atau membuat perubahan skema di lingkungan penahapannya. Ketika siap, Anda dapat mempromosikan lingkungan penahapan menjadi lingkungan basis data produksi baru. Untuk informasi selengkapnya, lihat Menggunakan Deployment Blue/Green Amazon RDS untuk pembaruan basis data.

Fitur Deployment Blue/Green didukung untuk mesin berikut:

- RDS for MariaDB versi 10.2 dan yang lebih tinggi
- RDS for MySQL versi 5.7 dan yang lebih tinggi
- RDS for MySQL versi 8.0.15 dan yang lebih tinggi
- RDS for PostgreSQL versi 11.21 dan yang lebih tinggi
- RDS for PostgreSQL versi 12.16 dan yang lebih tinggi
- RDS for PostgreSQL versi 13.12 dan yang lebih tinggi
- RDS for PostgreSQL versi 14.9 dan yang lebih tinggi
- RDS for PostgreSQL versi 15.4 dan yang lebih tinggi
- RDS for PostgreSQL versi 16.1 dan yang lebih tinggi

Fitur Deployment Blue/Green tidak didukung untuk mesin berikut:

- RDS for Db2
- RDS for SQL Server
- RDS for Oracle

Fitur Penerapan Biru/Hijau didukung di semua. Wilayah AWS

Pencadangan otomatis lintas Wilayah

Dengan replikasi pencadangan di Amazon RDS, Anda dapat mengonfigurasi instans DB RDS untuk mereplikasi snapshot dan log transaksi ke Wilayah tujuan. Saat replikasi pencadangan dikonfigurasi untuk instans DB, RDS memulai salinan lintas Wilayah untuk semua snapshot dan log transaksi setelah semuanya siap. Untuk informasi selengkapnya, lihat Mereplikasi backup otomatis ke yang lain Wilayah AWS.

Replikasi cadangan tersedia di semua Wilayah AWS kecuali berikut ini:

- Afrika (Cape Town)
- Asia Pasifik (Hong Kong)
- Asia Pasifik (Hyderabad)
- Asia Pasifik (Jakarta)
- Europe (Milan)
- Eropa (Spanyol)
- Eropa (Zürich)
- Timur Tengah (Bahrain)
- Timur Tengah (UEA)

Untuk informasi mendetail tentang batasan Wilayah pencadangan sumber dan tujuan, lihat Mereplikasi backup otomatis ke yang lain Wilayah AWS.

Topik

- Replikasi pencadangan dengan RDS for Db2
- Replikasi pencadangan dengan RDS for MariaDB
- Replikasi pencadangan dengan RDS for MySQL

- Replikasi pencadangan dengan RDS for Oracle
- Replikasi pencadangan dengan RDS for PostgreSQL
- Replikasi pencadangan dengan RDS for SQL Server

Replikasi pencadangan dengan RDS for Db2

Amazon RDS mendukung replikasi pencadangan untuk semua versi RDS for Db2 yang tersedia saat ini.

Replikasi pencadangan dengan RDS for MariaDB

Amazon RDS mendukung replikasi pencadangan untuk semua versi RDS for MariaDByang tersedia saat ini.

Replikasi pencadangan dengan RDS for MySQL

Amazon RDS mendukung replikasi pencadangan untuk semua versi RDS for MySQL yang tersedia saat ini.

Replikasi pencadangan dengan RDS for Oracle

Amazon RDS mendukung replikasi pencadangan untuk semua versi RDS for Oracle yang tersedia saat ini.

Replikasi pencadangan dengan RDS for PostgreSQL

Amazon RDS mendukung replikasi pencadangan untuk semua versi RDS for PostgreSQL yang tersedia saat ini.

Replikasi pencadangan dengan RDS for SQL Server

Amazon RDS mendukung replikasi pencadangan untuk semua versi RDS for SQL Server yang tersedia saat ini.

Replika baca lintas Wilayah

Dengan menggunakan replika baca lintas Wilayah di Amazon RDS, Anda dapat membuat replika baca MariaDB, MySQL, Oracle, PostgreSQL, atau SQL Server di Wilayah yang berbeda dari instans

DB sumber. Untuk informasi tentang replika baca lintas Wilayah, termasuk pertimbangan Wilayah sumber dan tujuan, lihat Membuat replika baca di tempat yang berbeda Wilayah AWS.

Replika baca lintas Wilayah tidak tersedia untuk mesin berikut:

- RDS for Db2

Topik

Replika baca lintas Wilayah dengan RDS for MariaDB

Replika baca lintas Wilayah dengan RDS for MariaDB tersedia di semua Wilayah untuk versi berikut:

- RDS for MariaDB 10.11 (Semua versi yang tersedia)
- RDS for MariaDB 10.6 (Semua versi yang tersedia)
- RDS for MariaDB 10.5 (Semua versi yang tersedia)
- RDS for MariaDB 10.4 (Semua versi yang tersedia)
- RDS for MariaDB 10.3 (Semua versi yang tersedia)

Replika baca lintas Wilayah dengan RDS for MySQL

Replika baca lintas Wilayah dengan RDS for MySQL tersedia di semua Wilayah untuk versi berikut:

- RDS for MySQL 8.0 (Semua versi yang tersedia)
- RDS for MySQL 5.7 (Semua versi yang tersedia)

Replika baca lintas Wilayah dengan RDS for Oracle

Replika baca lintas Wilayah dengan RDS for Oracle tersedia di semua Wilayah dengan batasan versi berikut:

- Untuk RDS for Oracle 21c, replika baca lintas Wilayah tidak tersedia.
- Untuk RDS untuk Oracle 19c, replika baca lintas wilayah tersedia untuk instance Oracle Database 19c yang bukan instance database kontainer (CDB).
- Untuk RDS for Oracle 12c, replika baca lintas Wilayah tersedia untuk Oracle Enterprise Edition (EE) dari Oracle Database 12c Rilis 1 (12.1) yang menggunakan 12.1.0.v10 dan rilis 12c yang lebih tinggi.

Informasi selengkapnya tentang persyaratan tambahan untuk replika baca lintas Wilayah dengan RDS for Oracle dapat dilihat di Persyaratan dan pertimbangan untuk replika RDS for Oracle.

Replika baca lintas Wilayah dengan RDS for PostgreSQL

Replika baca lintas Wilayah dengan RDS for PostgreSQL tersedia di semua Wilayah untuk versi berikut:

- RDS for PostgreSQL 16 (Semua versi yang tersedia)
- RDS for PostgreSQL 15 (Semua versi yang tersedia)
- RDS for PostgreSQL 14 (Semua versi yang tersedia)
- RDS for PostgreSQL 13 (Semua versi yang tersedia)
- RDS for PostgreSQL 12 (Semua versi yang tersedia)
- RDS for PostgreSQL 11 (Semua versi yang tersedia)
- RDS for PostgreSQL 10 (Semua versi yang tersedia)

Replika baca lintas Wilayah dengan RDS for SQL Server

Replika baca lintas Wilayah dengan RDS for SQL Server tersedia di semua Wilayah kecuali berikut ini:

- Afrika (Cape Town)
- Asia Pasifik (Hong Kong)
- Asia Pasifik (Hyderabad)
- Asia Pasifik (Jakarta)
- Asia Pasifik (Melbourne)
- Europe (Milan)

- Eropa (Spanyol)
- Eropa (Zürich)
- Timur Tengah (Bahrain)
- Timur Tengah (UEA)

Replika baca Lintas Wilayah dengan RDS for SQL Server tersedia untuk versi berikut yang menggunakan Microsoft SQL Server Enterprise Edition:

- RDS for SQL Server 2019 (Versi 15.00.4073.23 dan yang lebih tinggi)
- RDS for SQL Server 2017 (Versi 14.00.3281.6 dan yang lebih tinggi)
- RDS for SQL Server 2016 (Versi 13.00.6300.2 dan yang lebih tinggi)

Aliran aktivitas basis data

Dengan menggunakan Aliran Aktivitas Basis Data di Amazon RDS, Anda dapat memantau dan mengatur alarm untuk aktivitas audit di basis data Oracle dan basis data SQL Server. Untuk informasi selengkapnya, lihat Ikhtisar Aliran Aktivitas Basis Data.

Aliran aktivitas basis data tidak tersedia dengan mesin berikut ini:

- RDS for Db2
- RDS for MariaDB
- RDS for MySQL
- RDS for PostgreSQL

Topik

- Aliran aktivitas basis data dengan RDS for Oracle
- Aliran aktivitas basis data dengan RDS for SQL Server

Aliran aktivitas basis data dengan RDS for Oracle

Wilayah dan versi mesin berikut tersedia untuk aliran aktivitas basis data dengan RDS for Oracle.

Untuk informasi selengkapnya tentang persyaratan tambahan untuk aliran aktivitas basis data dengan RDS for Oracle, lihat Ikhtisar Aliran Aktivitas Basis Data.

Wilayah	RDS for Oracle 21c	RDS for Oracle 19c
AS Timur (Ohio)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 dan yang lebih tinggi, menggunakan Enterprise Edition (EE) atau Standard Edition 2 (SE2)
AS Timur (Virginia Utara)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 dan yang lebih tinggi, menggunakan Enterprise Edition (EE) atau Standard Edition 2 (SE2)
AS Barat (California Utara)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 dan yang lebih tinggi, menggunakan Enterprise Edition (EE) atau Standard Edition 2 (SE2)
AS Barat (Oregon)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 dan yang lebih tinggi, menggunakan Enterprise Edition (EE) atau Standard Edition 2 (SE2)
Afrika (Cape Town)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 dan yang lebih tinggi, menggunakan Enterprise Edition (EE) atau Standard Edition 2 (SE2)
Asia Pasifik (Hong Kong)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 dan yang lebih tinggi, menggunakan Enterprise Edition (EE) atau Standard Edition 2 (SE2)
Asia Pasifik (Hyderabad)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 dan yang lebih tinggi, menggunakan Enterprise Edition (EE) atau Standard Edition 2 (SE2)

Wilayah	RDS for Oracle 21c	RDS for Oracle 19c
Asia Pasifik (Jakarta)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 dan yang lebih tinggi, menggunakan Enterprise Edition (EE) atau Standard Edition 2 (SE2)
Asia Pasifik (Melbourne)	–	–
Asia Pasifik (Mumbai)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 dan yang lebih tinggi, menggunakan Enterprise Edition (EE) atau Standard Edition 2 (SE2)
Asia Pasifik (Osaka)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 dan yang lebih tinggi, menggunakan Enterprise Edition (EE) atau Standard Edition 2 (SE2)
Asia Pasifik (Seoul)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 dan yang lebih tinggi, menggunakan Enterprise Edition (EE) atau Standard Edition 2 (SE2)
Asia Pasifik (Singapura)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 dan yang lebih tinggi, menggunakan Enterprise Edition (EE) atau Standard Edition 2 (SE2)
Asia Pasifik (Sydney)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 dan yang lebih tinggi, menggunakan Enterprise Edition (EE) atau Standard Edition 2 (SE2)
Asia Pasifik (Tokyo)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 dan yang lebih tinggi, menggunakan Enterprise Edition (EE) atau Standard Edition 2 (SE2)

Wilayah	RDS for Oracle 21c	RDS for Oracle 19c
Kanada (Pusat)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 dan yang lebih tinggi, menggunakan Enterprise Edition (EE) atau Standard Edition 2 (SE2)
Kanada Barat (Calgary)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 dan yang lebih tinggi, menggunakan Enterprise Edition (EE) atau Standard Edition 2 (SE2)
Tiongkok (Beijing)	–	–
Tiongkok (Ningxia)	–	–
Eropa (Frankfurt)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 dan yang lebih tinggi, menggunakan Enterprise Edition (EE) atau Standard Edition 2 (SE2)
Eropa (Irlandia)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 dan yang lebih tinggi, menggunakan Enterprise Edition (EE) atau Standard Edition 2 (SE2)
Eropa (London)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 dan yang lebih tinggi, menggunakan Enterprise Edition (EE) atau Standard Edition 2 (SE2)
Europe (Milan)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 dan yang lebih tinggi, menggunakan Enterprise Edition (EE) atau Standard Edition 2 (SE2)

Wilayah	RDS for Oracle 21c	RDS for Oracle 19c
Eropa (Paris)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 dan yang lebih tinggi, menggunakan Enterprise Edition (EE) atau Standard Edition 2 (SE2)
Eropa (Spanyol)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 dan yang lebih tinggi, menggunakan Enterprise Edition (EE) atau Standard Edition 2 (SE2)
Eropa (Stockholm)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 dan yang lebih tinggi, menggunakan Enterprise Edition (EE) atau Standard Edition 2 (SE2)
Eropa (Zürich)	–	–
Asia Pasifik (Melbourne)	–	–
Timur Tengah (Bahrain)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 dan yang lebih tinggi, menggunakan Enterprise Edition (EE) atau Standard Edition 2 (SE2)
Timur Tengah (UEA)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 dan yang lebih tinggi, menggunakan Enterprise Edition (EE) atau Standard Edition 2 (SE2)
Amerika Selatan (Sao Paulo)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 dan yang lebih tinggi, menggunakan Enterprise Edition (EE) atau Standard Edition 2 (SE2)
AWS GovCloud (AS-Timur)	–	–

Wilayah	RDS for Oracle 21c	RDS for Oracle 19c
AWS GovCloud (AS-Barat)	–	–

Aliran aktivitas basis data dengan RDS for SQL Server

Wilayah dan versi mesin berikut tersedia untuk aliran aktivitas basis data dengan RDS for SQL Server.

Untuk informasi selengkapnya tentang persyaratan tambahan untuk aliran aktivitas basis data dengan RDS for SQL Server, lihat Ikhtisar Aliran Aktivitas Basis Data.

Wilayah	RDS for SQL Server 2019	RDS for SQL Server 2017	RDS for SQL Server 2016	RDS for SQL Server 2014
AS Timur (Ohio)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
AS Timur (Virginia Utara)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
AS Barat (California Utara)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
AS Barat (Oregon)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Afrika (Cape Town)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Asia Pasifik (Hong Kong)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Asia Pasifik (Hyderabad)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–

Wilayah	RDS for SQL Server 2019	RDS for SQL Server 2017	RDS for SQL Server 2016	RDS for SQL Server 2014
Asia Pasifik (Jakarta)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Asia Pasifik (Melbourne)	–	–	–	–
Asia Pasifik (Mumbai)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Asia Pasifik (Osaka)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Asia Pasifik (Seoul)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Asia Pasifik (Singapura)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Asia Pasifik (Sydney)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Asia Pasifik (Tokyo)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Kanada (Pusat)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Kanada Barat (Calgary)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Tiongkok (Beijing)	–	–	–	–
Tiongkok (Ningxia)	–	–	–	–

Wilayah	RDS for SQL Server 2019	RDS for SQL Server 2017	RDS for SQL Server 2016	RDS for SQL Server 2014
Eropa (Frankfurt)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Eropa (Irlandia)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Eropa (London)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Europe (Milan)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Eropa (Paris)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Eropa (Spanyol)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Eropa (Stockholm)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Eropa (Zürich)	–	–	–	–
Israel (Tel Aviv)	–	–	–	–
Timur Tengah (Bahrain)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Timur Tengah (UEA)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Amerika Selatan (Sao Paulo)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
AWS GovCloud (AS-Timur)	–	–	–	–

Wilayah	RDS for SQL Server 2019	RDS for SQL Server 2017	RDS for SQL Server 2016	RDS for SQL Server 2014
AWS GovCloud (AS-Barat)	–	–	–	–

Mode tumpukan ganda

Dengan menggunakan mode tumpukan ganda dalam RDS, sumber daya dapat berkomunikasi dengan instans DB melalui Protokol Internet versi 4 (IPv4), Protokol Internet versi 6 (IPv6), atau keduanya. Untuk informasi selengkapnya, lihat Mode dual-stack.

Topik

- Mode dual-stack dengan RDS untuk Db2
- Mode tumpukan ganda dengan RDS for MariaDB
- Mode tumpukan ganda dengan RDS for MySQL
- Mode tumpukan ganda dengan RDS for Oracle
- Mode tumpukan ganda dengan RDS for PostgreSQL
- Mode tumpukan ganda dengan RDS for SQL Server

Mode dual-stack dengan RDS untuk Db2

Wilayah dan versi mesin berikut tersedia untuk mode dual-stack dengan RDS untuk Db2.

Wilayah	RDS for Db2 11.5				
AS Timur (Ohio)	Semua versi yang tersedia				
AS Timur (Virginia Utara)	Semua versi yang tersedia				

Wilayah	RDS for Db2 11.5				
AS Barat (California Utara)	Semua versi yang tersedia				
AS Barat (Oregon)	Semua versi yang tersedia				
Afrika (Cape Town)	Semua versi yang tersedia				
Asia Pasifik (Hong Kong)	Semua versi yang tersedia				
Asia Pasifik (Hyderabad)	Semua versi yang tersedia				
Asia Pasifik (Jakarta)	Semua versi yang tersedia				
Asia Pasifik (Melbourne)	Semua versi yang tersedia				
Asia Pasifik (Mumbai)	Semua versi yang tersedia				
Asia Pasifik (Osaka)	Semua versi yang tersedia				
Asia Pasifik (Seoul)	Semua versi yang tersedia				
Asia Pasifik (Singapura)	Semua versi yang tersedia				
Asia Pasifik (Sydney)	Semua versi yang tersedia				

Wilayah	RDS for Db2 11.5				
Asia Pasifik (Tokyo)	Semua versi yang tersedia				
Kanada (Pusat)	Semua versi yang tersedia				
Kanada Barat (Calgary)	–				
China (Beijing)	–				
Tiongkok (Ningxia)	–				
Eropa (Frankfurt)	Semua versi yang tersedia				
Eropa (Irlandia)	Semua versi yang tersedia				
Eropa (London)	Semua versi yang tersedia				
Eropa (Milan)	Semua versi yang tersedia				
Eropa (Paris)	Semua versi yang tersedia				
Eropa (Spanyol)	Semua versi yang tersedia				
Eropa (Stockholm)	Semua versi yang tersedia				

Wilayah	RDS for Db2 11.5				
Eropa (Zürich)	Semua versi yang tersedia				
Israel (Tel Aviv)	–				
Timur Tengah (Bahrain)	Semua versi yang tersedia				
Timur Tengah (UEA)	Semua versi yang tersedia				
Amerika Selatan (Sao Paulo)	Semua versi yang tersedia				
AWS GovCloud (AS-Timur)	–				
AWS GovCloud (AS-Barat)	–				

Mode tumpukan ganda dengan RDS for MariaDB

Wilayah dan versi mesin berikut tersedia untuk mode tumpukan ganda dengan RDS for MariaDB.

Wilayah	RDS for MariaDB 10.11	RDS for MariaDB 10.6	RDS for MariaDB 10.5	RDS for MariaDB 10.4	RDS for MariaDB 10.3
AS Timur (Ohio)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AS Timur (Virginia Utara)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AS Barat (California Utara)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AS Barat (Oregon)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Afrika (Cape Town)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Hong Kong)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Hyderabad)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Jakarta)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Melbourne)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Mumbai)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Osaka)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia

Wilayah	RDS for MariaDB 10.11	RDS for MariaDB 10.6	RDS for MariaDB 10.5	RDS for MariaDB 10.4	RDS for MariaDB 10.3
Asia Pasifik (Seoul)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Singapura)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Sydney)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Tokyo)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Kanada (Pusat)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Kanada Barat (Calgary)	–	–	–	–	–
Tiongkok (Beijing)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Tiongkok (Ningxia)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Frankfurt)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Irlandia)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (London)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Milan)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia

Wilayah	RDS for MariaDB 10.11	RDS for MariaDB 10.6	RDS for MariaDB 10.5	RDS for MariaDB 10.4	RDS for MariaDB 10.3
Eropa (Paris)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Spanyol)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Stockholm)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Zürich)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Israel (Tel Aviv)	–	–	–	–	–
Timur Tengah (Bahrain)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Timur Tengah (UEA)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Amerika Selatan (Sao Paulo)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AWS GovCloud (AS-Timur)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AWS GovCloud (AS-Barat)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia

Mode tumpukan ganda dengan RDS for MySQL

Wilayah dan versi mesin berikut tersedia untuk mode tumpukan ganda dengan RDS for MySQL.

Wilayah	RDS for MySQL 8.0	RDS for MySQL 5.7	RDS for MySQL 5.6
AS Timur (Ohio)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AS Timur (Virginia Utara)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AS Barat (California Utara)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AS Barat (Oregon)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Afrika (Cape Town)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Hong Kong)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Hyderabad)	Semua versi yang tersedia	Semua versi yang tersedia	–
Asia Pasifik (Jakarta)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Melbourne)	Semua versi yang tersedia	Semua versi yang tersedia	–
Asia Pasifik (Mumbai)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Osaka)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia

Wilayah	RDS for MySQL 8.0	RDS for MySQL 5.7	RDS for MySQL 5.6
Asia Pasifik (Seoul)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Singapura)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Sydney)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Tokyo)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Kanada (Pusat)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Kanada Barat (Calgary)	–	–	–
Tiongkok (Beijing)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Tiongkok (Ningxia)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Frankfurt)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Irlandia)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (London)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Milan)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Paris)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia

Wilayah	RDS for MySQL 8.0	RDS for MySQL 5.7	RDS for MySQL 5.6
Eropa (Spanyol)	Semua versi yang tersedia	Semua versi yang tersedia	–
Eropa (Stockholm)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Zürich)	Semua versi yang tersedia	Semua versi yang tersedia	–
Israel (Tel Aviv)	–	–	–
Timur Tengah (Bahrain)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Timur Tengah (UEA)	Semua versi yang tersedia	Semua versi yang tersedia	–
Amerika Selatan (Sao Paulo)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AWS GovCloud (AS-Timur)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AWS GovCloud (AS-Barat)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia

Mode tumpukan ganda dengan RDS for Oracle

Wilayah dan versi mesin berikut tersedia untuk mode tumpukan ganda dengan RDS for Oracle.

Wilayah	RDS for Oracle 21c	RDS for Oracle 19c	RDS for Oracle 12c
AS Timur (Ohio)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AS Timur (Virginia Utara)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia

Wilayah	RDS for Oracle 21c	RDS for Oracle 19c	RDS for Oracle 12c
AS Barat (California Utara)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AS Barat (Oregon)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Afrika (Cape Town)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Hong Kong)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Hyderabad)	–	–	–
Asia Pasifik (Jakarta)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Melbourne)	–	–	–
Asia Pasifik (Mumbai)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Osaka)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Seoul)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Singapura)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Sydney)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Tokyo)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia

Wilayah	RDS for Oracle 21c	RDS for Oracle 19c	RDS for Oracle 12c
Kanada (Pusat)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Kanada Barat (Calgary)	–	–	–
Tiongkok (Beijing)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Tiongkok (Ningxia)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Frankfurt)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Irlandia)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (London)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Milan)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Paris)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Spanyol)	–	–	–
Eropa (Stockholm)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Zürich)	–	–	–
Israel (Tel Aviv)	–	–	–
Timur Tengah (Bahrain)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia

Wilayah	RDS for Oracle 21c	RDS for Oracle 19c	RDS for Oracle 12c
Timur Tengah (UEA)	–	–	–
Amerika Selatan (Sao Paulo)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AWS GovCloud (AS-Timur)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AWS GovCloud (AS-Barat)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia

Mode tumpukan ganda dengan RDS for PostgreSQL

Wilayah dan versi mesin berikut tersedia untuk mode tumpukan ganda dengan RDS for PostgreSQL.

Wilayah	RDS for PostgreSQL 16	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
AS Timur (Ohio)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AS Timur (Virginia Utara)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AS Barat (California Utara)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AS Barat (Oregon)	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi

Wilayah	RDS for PostgreSQL 16	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
	yang tersedia	yang tersedia	yang tersedia	yang tersedia	yang tersedia	yang tersedia	yang tersedia
Afrika (Cape Town)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Hong Kong)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Hyderabad)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Melbourne)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Jakarta)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Mumbai)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia

Wilayah	RDS for PostgreSQL 16	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
Asia Pasifik (Osaka)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Seoul)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Singapura)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Sydney)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Tokyo)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Kanada (Pusat)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Kanada Barat (Calgary)	–	–	–	–	–	–	–

Wilayah	RDS for PostgreSQL 16	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
Tiongkok (Beijing)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Tiongkok (Ningxia)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Frankfurt)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Irlandia)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (London)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Milan)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Paris)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia

Wilayah	RDS for PostgreSQL 16	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
Eropa (Spanyol)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Stockholm)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Zürich)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Israel (Tel Aviv)	–	–	–	–	–	–	–
Timur Tengah (Bahrain)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Timur Tengah (UEA)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Amerika Selatan (Sao Paulo)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia

Wilayah	RDS for PostgreSQL 16	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
AWS GovCloud (AS-Timur)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AWS GovCloud (AS-Barat)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia

Mode tumpukan ganda dengan RDS for SQL Server

Wilayah dan versi mesin berikut tersedia untuk mode tumpukan ganda dengan RDS for SQL Server.

Wilayah	RDS for SQL Server 2019	RDS for SQL Server 2017	RDS for SQL Server 2016	RDS for SQL Server 2014
AS Timur (Ohio)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
AS Timur (Virginia Utara)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
AS Barat (California Utara)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
AS Barat (Oregon)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Afrika (Cape Town)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Asia Pasifik (Hong Kong)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–

Wilayah	RDS for SQL Server 2019	RDS for SQL Server 2017	RDS for SQL Server 2016	RDS for SQL Server 2014
Asia Pasifik (Hyderabad)	–	–	–	–
Asia Pasifik (Jakarta)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Asia Pasifik (Melbourne)	–	–	–	–
Asia Pasifik (Mumbai)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Asia Pasifik (Osaka)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Asia Pasifik (Seoul)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Asia Pasifik (Singapura)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Asia Pasifik (Sydney)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Asia Pasifik (Tokyo)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Kanada (Pusat)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Kanada Barat (Calgary)	–	–	–	–
Tiongkok (Beijing)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–

Wilayah	RDS for SQL Server 2019	RDS for SQL Server 2017	RDS for SQL Server 2016	RDS for SQL Server 2014
Tiongkok (Ningxia)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Eropa (Frankfurt)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Eropa (Irlandia)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Eropa (London)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Eropa (Milan)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Eropa (Paris)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Eropa (Spanyol)	–	–	–	–
Eropa (Stockholm)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Eropa (Zürich)	–	–	–	–
Israel (Tel Aviv)	–	–	–	–
Timur Tengah (Bahrain)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
Timur Tengah (UEA)	–	–	–	–
Amerika Selatan (Sao Paulo)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–

Wilayah	RDS for SQL Server 2019	RDS for SQL Server 2017	RDS for SQL Server 2016	RDS for SQL Server 2014
AWS GovCloud (AS-Timur)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–
AWS GovCloud (AS-Barat)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	–

Ekspor snapshot ke S3

Anda dapat mengekspor data snapshot DB RDS ke bucket Amazon S3. Anda dapat mengekspor semua jenis snapshot DB—termasuk snapshot manual, snapshot sistem otomatis, dan snapshot yang dibuat oleh AWS Backup. Setelah data diekspor, Anda dapat menganalisis data yang diekspor secara langsung melalui alat seperti Amazon Athena atau Amazon Redshift Spectrum. Untuk informasi selengkapnya, lihat Mengekspor data snapshot DB ke Amazon S3.

Mengekspor snapshot ke S3 tidak tersedia untuk mesin berikut:

- RDS for Db2
- RDS for Oracle
- RDS for SQL Server

Topik

- Ekspor snapshot ke S3 dengan RDS for MariaDB
- Ekspor snapshot ke S3 dengan RDS for MySQL
- Ekspor snapshot ke S3 dengan RDS for PostgreSQL

Ekspor snapshot ke S3 dengan RDS for MariaDB

Wilayah dan versi mesin berikut tersedia untuk mengekspor snapshot ke S3 dengan RDS for MariaDB.

Wilayah	RDS for MariaDB 10.11	RDS for MariaDB 10.6	RDS for MariaDB 10.5	RDS for MariaDB 10.4	RDS for MariaDB 10.3
AS Timur (Ohio)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AS Timur (Virginia Utara)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AS Barat (California Utara)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AS Barat (Oregon)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Afrika (Cape Town)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Hong Kong)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Hyderabad)	–	–	–	–	–
Asia Pasifik (Jakarta)	–	–	–	–	–
Asia Pasifik (Melbourne)	–	–	–	–	–
Asia Pasifik (Mumbai)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Osaka)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia

Wilayah	RDS for MariaDB 10.11	RDS for MariaDB 10.6	RDS for MariaDB 10.5	RDS for MariaDB 10.4	RDS for MariaDB 10.3
Asia Pasifik (Seoul)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Singapura)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Sydney)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Tokyo)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Kanada (Pusat)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Kanada Barat (Calgary)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Tiongkok (Beijing)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Tiongkok (Ningxia)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Frankfurt)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Irlandia)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (London)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Europe (Milan)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia

Wilayah	RDS for MariaDB 10.11	RDS for MariaDB 10.6	RDS for MariaDB 10.5	RDS for MariaDB 10.4	RDS for MariaDB 10.3
Eropa (Paris)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Spanyol)	–	–	–	–	–
Eropa (Stockholm)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Zürich)	–	–	–	–	–
Israel (Tel Aviv)	–	–	–	–	–
Timur Tengah (Bahrain)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Timur Tengah (UEA)	–	–	–	–	–
Amerika Selatan (Sao Paulo)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AWS GovCloud (AS-Timur)	–	–	–	–	–
AWS GovCloud (AS-Barat)	–	–	–	–	–

Ekspor snapshot ke S3 dengan RDS for MySQL

Wilayah dan versi mesin berikut tersedia untuk mengekspor snapshot ke S3 dengan RDS for MySQL.

Wilayah	RDS for MySQL 8.0	RDS for MySQL 5.7
AS Timur (Ohio)	Semua versi yang tersedia	Semua versi yang tersedia
AS Timur (Virginia Utara)	Semua versi yang tersedia	Semua versi yang tersedia
AS Barat (California Utara)	Semua versi yang tersedia	Semua versi yang tersedia
AS Barat (Oregon)	Semua versi yang tersedia	Semua versi yang tersedia
Afrika (Cape Town)	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Hong Kong)	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Hyderabad)	–	–
Asia Pasifik (Jakarta)	–	–
Asia Pasifik (Melbourne)	–	–
Asia Pasifik (Mumbai)	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Osaka)	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Seoul)	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Singapura)	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Sydney)	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Tokyo)	Semua versi yang tersedia	Semua versi yang tersedia
Kanada (Pusat)	Semua versi yang tersedia	Semua versi yang tersedia
Kanada Barat (Calgary)	–	–
Tiongkok (Beijing)	Semua versi yang tersedia	Semua versi yang tersedia

Wilayah	RDS for MySQL 8.0	RDS for MySQL 5.7
Tiongkok (Ningxia)	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Frankfurt)	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Irlandia)	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (London)	Semua versi yang tersedia	Semua versi yang tersedia
Europe (Milan)	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Paris)	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Spanyol)	–	–
Eropa (Stockholm)	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Zürich)	–	–
Israel (Tel Aviv)	–	–
Timur Tengah (Bahrain)	Semua versi yang tersedia	Semua versi yang tersedia
Timur Tengah (UEA)	–	–
Amerika Selatan (Sao Paulo)	Semua versi yang tersedia	Semua versi yang tersedia
AWS GovCloud (AS-Timur)	–	–
AWS GovCloud (AS-Barat)	–	–

Ekspor snapshot ke S3 dengan RDS for PostgreSQL

Wilayah dan versi mesin berikut tersedia untuk mengekspor snapshot ke S3 dengan RDS for PostgreSQL.

Wilayah	RDS for PostgreSQL 16	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
AS Timur (Ohio)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AS Timur (Virginia Utara)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AS Barat (California Utara)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AS Barat (Oregon)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Afrika (Cape Town)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Hong Kong)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Hyderabad)	–	–	–	–	–	–	–

Wilayah	RDS for PostgreSQL 16	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
Asia Pasifik (Jakarta)	–	–	–	–	–	–	–
Asia Pasifik (Melbourne)	–	–	–	–	–	–	–
Asia Pasifik (Mumbai)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Osaka)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Seoul)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Singapura)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Sydney)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia

Wilayah	RDS for PostgreSQL 16	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
Asia Pasifik (Tokyo)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Kanada (Pusat)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Kanada Barat (Calgary)	–	–	–	–	–	–	–
Tiongkok (Beijing)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Tiongkok (Ningxia)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Frankfurt)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Irlandia)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia

Wilayah	RDS for PostgreSQL 16	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
Eropa (London)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Europe (Milan)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Paris)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Spanyol)	–	–	–	–	–	–	–
Eropa (Stockholm)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Zürich)	–	–	–	–	–	–	–
Israel (Tel Aviv)	–	–	–	–	–	–	–
Timur Tengah (Bahrain)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia

Wilayah	RDS for PostgreSQL 16	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
Timur Tengah (UEA)	–	–	–	–	–	–	–
Amerika Selatan (Sao Paulo)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AWS GovCloud (AS-Timur)	–	–	–	–	–	–	–
AWS GovCloud (AS-Barat)	–	–	–	–	–	–	–

Autentikasi basis data IAM

Dengan autentikasi basis data IAM di Amazon RDS, Anda dapat melakukan autentikasi tanpa kata sandi saat menghubungkan ke instans DB. Sebagai gantinya, Anda menggunakan token autentikasi. Untuk informasi selengkapnya, lihat Autentikasi basis data IAM untuk MariaDB, MySQL, dan PostgreSQL.

Autentikasi basis data IAM tidak tersedia dengan mesin berikut:

- RDS for Db2
- RDS for Oracle
- RDS for SQL Server

Topik

- Autentikasi basis data IAM dengan RDS for MariaDB
- Autentikasi basis data IAM dengan RDS for MySQL
- Autentikasi basis data IAM dengan RDS for PostgreSQL

Autentikasi basis data IAM dengan RDS for MariaDB

Wilayah dan versi mesin berikut tersedia untuk autentikasi basis data IAM dengan RDS for MariaDB.

Wilayah	RDS for MariaDB 10.11	RDS for MariaDB 10.6	RDS for MariaDB 10.5	RDS for MariaDB 10.4	RDS for MariaDB 10.3
AS Timur (Ohio)	Semua versi yang tersedia	Semua versi yang tersedia	–	–	–
AS Timur (Virginia Utara)	Semua versi yang tersedia	Semua versi yang tersedia	–	–	–
AS Barat (California Utara)	Semua versi yang tersedia	Semua versi yang tersedia	–	–	–
AS Barat (Oregon)	Semua versi yang tersedia	Semua versi yang tersedia	–	–	–
Afrika (Cape Town)	Semua versi yang tersedia	Semua versi yang tersedia	–	–	–
Asia Pasifik (Hong Kong)	Semua versi yang tersedia	Semua versi yang tersedia	–	–	–
Asia Pasifik (Hyderabad)	–	–	–	–	–
Asia Pasifik (Jakarta)	Semua versi yang tersedia	Semua versi yang tersedia	–	–	–

Wilayah	RDS for MariaDB 10.11	RDS for MariaDB 10.6	RDS for MariaDB 10.5	RDS for MariaDB 10.4	RDS for MariaDB 10.3
Asia Pasifik (Melbourne)	–	–	–	–	–
Asia Pasifik (Mumbai)	Semua versi yang tersedia	Semua versi yang tersedia	–	–	–
Asia Pasifik (Osaka)	Semua versi yang tersedia	Semua versi yang tersedia	–	–	–
Asia Pasifik (Seoul)	Semua versi yang tersedia	Semua versi yang tersedia	–	–	–
Asia Pasifik (Singapura)	Semua versi yang tersedia	Semua versi yang tersedia	–	–	–
Asia Pasifik (Sydney)	Semua versi yang tersedia	Semua versi yang tersedia	–	–	–
Asia Pasifik (Tokyo)	Semua versi yang tersedia	Semua versi yang tersedia	–	–	–
Kanada (Pusat)	Semua versi yang tersedia	Semua versi yang tersedia	–	–	–
Kanada Barat (Calgary)	Semua versi yang tersedia	Semua versi yang tersedia	–	–	–
Tiongkok (Beijing)	Semua versi yang tersedia	Semua versi yang tersedia	–	–	–
Tiongkok (Ningxia)	Semua versi yang tersedia	Semua versi yang tersedia	–	–	–
Eropa (Frankfurt)	Semua versi yang tersedia	Semua versi yang tersedia	–	–	–

Wilayah	RDS for MariaDB 10.11	RDS for MariaDB 10.6	RDS for MariaDB 10.5	RDS for MariaDB 10.4	RDS for MariaDB 10.3
Eropa (Irlandia)	Semua versi yang tersedia	Semua versi yang tersedia	–	–	–
Eropa (London)	Semua versi yang tersedia	Semua versi yang tersedia	–	–	–
Europe (Milan)	Semua versi yang tersedia	Semua versi yang tersedia	–	–	–
Eropa (Paris)	Semua versi yang tersedia	Semua versi yang tersedia	–	–	–
Eropa (Spanyol)	–	–	–	–	–
Eropa (Stockholm)	Semua versi yang tersedia	Semua versi yang tersedia	–	–	–
Eropa (Zürich)	–	–	–	–	–
Israel (Tel Aviv)	–	–	–	–	–
Timur Tengah (Bahrain)	Semua versi yang tersedia	Semua versi yang tersedia	–	–	–
Timur Tengah (UEA)	–	–	–	–	–
Amerika Selatan (Sao Paulo)	Semua versi yang tersedia	Semua versi yang tersedia	–	–	–

Wilayah	RDS for MariaDB 10.11	RDS for MariaDB 10.6	RDS for MariaDB 10.5	RDS for MariaDB 10.4	RDS for MariaDB 10.3
AWS GovCloud (AS-Timur)	Semua versi yang tersedia	Semua versi yang tersedia	–	–	–
AWS GovCloud (AS-Barat)	Semua versi yang tersedia	Semua versi yang tersedia	–	–	–

Autentikasi basis data IAM dengan RDS for MySQL

Autentikasi basis data IAM dengan RDS for MySQL tersedia di semua Wilayah untuk versi berikut:

- RDS for MySQL 8.0 – Semua versi yang tersedia
- RDS for MySQL 5.7 – Semua versi yang tersedia

Autentikasi basis data IAM dengan RDS for PostgreSQL

Autentikasi basis data IAM dengan RDS for PostgreSQL tersedia di semua Wilayah untuk versi berikut:

- RDS for PostgreSQL 16 – Semua versi yang tersedia
- RDS for PostgreSQL 15 – Semua versi yang tersedia
- RDS for PostgreSQL 14 – Semua versi yang tersedia
- RDS for PostgreSQL 13 – Semua versi yang tersedia
- RDS for PostgreSQL 12 – Semua versi yang tersedia
- RDS for PostgreSQL 11 – Semua versi yang tersedia
- RDS for PostgreSQL 10 – Semua versi yang tersedia

Autentikasi Kerberos

Dengan menggunakan autentikasi Kerberos di Amazon RDS, Anda dapat mendukung autentikasi eksternal pengguna basis data menggunakan Kerberos dan Microsoft Active Directory. Penggunaan Kerberos dan Active Directory memberikan manfaat masuk tunggal dan autentikasi terpusat pengguna basis data. Untuk informasi selengkapnya, lihat Autentikasi Kerberos.

Autentikasi Kerberos tidak tersedia dengan mesin berikut:

- RDS for MariaDB

Topik

Autentikasi Kerberos dengan RDS for Db2

Wilayah dan versi mesin berikut tersedia untuk autentikasi Kerberos dengan RDS for Db2.

Wilayah	RDS for Db2 11.5
AS Timur (Ohio)	Semua versi
AS Timur (Virginia Utara)	Semua versi
AS Barat (California Utara)	Semua versi
AS Barat (Oregon)	Semua versi
Afrika (Cape Town)	–
Asia Pasifik (Hong Kong)	–
Asia Pasifik (Hyderabad)	–

Wilayah	RDS for Db2 11.5
Asia Pasifik (Jakarta)	–
Asia Pasifik (Melbourne)	–
Asia Pasifik (Mumbai)	Semua versi
Asia Pasifik (Osaka)	–
Asia Pasifik (Seoul)	Semua versi
Asia Pasifik (Singapura)	Semua versi
Asia Pasifik (Sydney)	Semua versi
Asia Pasifik (Tokyo)	Semua versi
Kanada (Pusat)	Semua versi
Kanada Barat (Calgary)	–
Tiongkok (Beijing)	Semua versi
Tiongkok (Ningxia)	Semua versi
Eropa (Frankfurt)	Semua versi
Eropa (Irlandia)	Semua versi
Eropa (London)	Semua versi
Europe (Milan)	–
Eropa (Paris)	–
Eropa (Spanyol)	–
Eropa (Stockholm)	Semua versi
Eropa (Zürich)	–

Wilayah	RDS for Db2 11.5
Israel (Tel Aviv)	–
Timur Tengah (Bahrain)	–
Timur Tengah (UEA)	–
Amerika Selatan (Sao Paulo)	Semua versi
AWS GovCloud (AS-Timur)	–
AWS GovCloud (AS-Barat)	–

Autentikasi Kerberos dengan RDS for MySQL

Wilayah dan versi mesin berikut tersedia untuk autentikasi Kerberos dengan RDS for MySQL.

Wilayah	RDS for MySQL 8.0	RDS for MySQL 5.7	RDS for MySQL 5.6
AS Timur (Ohio)	Semua versi	Semua versi	Semua versi
AS Timur (Virginia Utara)	Semua versi	Semua versi	Semua versi
AS Barat (California Utara)	Semua versi	Semua versi	Semua versi
AS Barat (Oregon)	Semua versi	Semua versi	Semua versi
Afrika (Cape Town)	–	–	–
Asia Pasifik (Hong Kong)	–	–	–
Asia Pasifik (Hyderabad)	–	–	–
Asia Pasifik (Jakarta)	–	–	–

Wilayah	RDS for MySQL 8.0	RDS for MySQL 5.7	RDS for MySQL 5.6
Asia Pasifik (Melbourne)	–	–	–
Asia Pasifik (Mumbai)	Semua versi	Semua versi	Semua versi
Asia Pasifik (Osaka)	–	–	–
Asia Pasifik (Seoul)	Semua versi	Semua versi	Semua versi
Asia Pasifik (Singapura)	Semua versi	Semua versi	Semua versi
Asia Pasifik (Sydney)	Semua versi	Semua versi	Semua versi
Asia Pasifik (Tokyo)	Semua versi	Semua versi	Semua versi
Kanada (Pusat)	Semua versi	Semua versi	Semua versi
Kanada Barat (Calgary)	–	–	–
Tiongkok (Beijing)	Semua versi	Semua versi	Semua versi
Tiongkok (Ningxia)	Semua versi	Semua versi	Semua versi
Eropa (Frankfurt)	Semua versi	Semua versi	Semua versi
Eropa (Irlandia)	Semua versi	Semua versi	Semua versi
Eropa (London)	Semua versi	Semua versi	Semua versi
Europe (Milan)	–	–	–
Eropa (Paris)	–	–	–
Eropa (Spanyol)	–	–	–
Eropa (Stockholm)	Semua versi	Semua versi	Semua versi
Eropa (Zürich)	–	–	–

Wilayah	RDS for MySQL 8.0	RDS for MySQL 5.7	RDS for MySQL 5.6
Israel (Tel Aviv)	–	–	–
Timur Tengah (Bahrain)	–	–	–
Timur Tengah (UEA)	–	–	–
Amerika Selatan (Sao Paulo)	Semua versi	Semua versi	Semua versi
AWS GovCloud (AS-Timur)	–	–	–
AWS GovCloud (AS-Barat)	–	–	–

Autentikasi Kerberos dengan RDS for Oracle

Wilayah dan versi mesin berikut tersedia untuk autentikasi Kerberos dengan RDS for Oracle.

Wilayah	RDS for Oracle 21c	RDS for Oracle 19c	RDS for Oracle 12c
AS Timur (Ohio)	Semua versi	Semua versi	Semua versi
AS Timur (Virginia Utara)	Semua versi	Semua versi	Semua versi
AS Barat (California Utara)	Semua versi	Semua versi	Semua versi
AS Barat (Oregon)	Semua versi	Semua versi	Semua versi
Afrika (Cape Town)	–	–	–
Asia Pasifik (Hong Kong)	–	–	–

Wilayah	RDS for Oracle 21c	RDS for Oracle 19c	RDS for Oracle 12c
Asia Pasifik (Hyderabad)	–	–	–
Asia Pasifik (Jakarta)	–	–	–
Asia Pasifik (Melbourne)	–	–	–
Asia Pasifik (Mumbai)	Semua versi	Semua versi	Semua versi
Asia Pasifik (Osaka)	–	–	–
Asia Pasifik (Seoul)	Semua versi	Semua versi	Semua versi
Asia Pasifik (Singapura)	Semua versi	Semua versi	Semua versi
Asia Pasifik (Sydney)	Semua versi	Semua versi	Semua versi
Asia Pasifik (Tokyo)	Semua versi	Semua versi	Semua versi
Kanada (Pusat)	Semua versi	Semua versi	Semua versi
Kanada Barat (Calgary)	–	–	–
China (Beijing)	–	–	–
Tiongkok (Ningxia)	–	–	–
Eropa (Frankfurt)	Semua versi	Semua versi	Semua versi
Eropa (Irlandia)	Semua versi	Semua versi	Semua versi
Eropa (London)	Semua versi	Semua versi	Semua versi
Europe (Milan)	–	–	–
Eropa (Paris)	–	–	–

Wilayah	RDS for Oracle 21c	RDS for Oracle 19c	RDS for Oracle 12c
Eropa (Spanyol)	–	–	–
Eropa (Stockholm)	Semua versi	Semua versi	Semua versi
Eropa (Zürich)	–	–	–
Israel (Tel Aviv)	–	–	–
Timur Tengah (Bahrain)	–	–	–
Timur Tengah (UEA)	–	–	–
Amerika Selatan (Sao Paulo)	Semua versi	Semua versi	Semua versi
AWS GovCloud (AS-Timur)	Semua versi	Semua versi	Semua versi
AWS GovCloud (AS-Barat)	Semua versi	Semua versi	Semua versi

Autentikasi Kerberos dengan RDS for PostgreSQL

Wilayah dan versi mesin berikut tersedia untuk autentikasi Kerberos dengan RDS for PostgreSQL.

Wilayah	RDS for PostgreSQL 16	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
AS Timur (Ohio)	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi
AS Timur (Virginia Utara)	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi

Wilayah	RDS for PostgreSQL 16	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
AS Barat (California Utara)	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi
AS Barat (Oregon)	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi
Afrika (Cape Town)	–	–	–	–	–	–	–
Asia Pasifik (Hong Kong)	–	–	–	–	–	–	–
Asia Pasifik (Hyderabad)	–	–	–	–	–	–	–
Asia Pasifik (Jakarta)	–	–	–	–	–	–	–
Asia Pasifik (Melbourne)	–	–	–	–	–	–	–
Asia Pasifik (Mumbai)	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi

Wilayah	RDS for PostgreSQL 16	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
Asia Pasifik (Osaka)	–	–	–	–	–	–	–
Asia Pasifik (Seoul)	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi
Asia Pasifik (Singapura)	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi
Asia Pasifik (Sydney)	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi
Asia Pasifik (Tokyo)	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi
Kanada (Pusat)	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi
Kanada Barat (Calgary)	–	–	–	–	–	–	–
Tiongkok (Beijing)	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi
Tiongkok (Ningxia)	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi

Wilayah	RDS for PostgreSQL 16	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
Eropa (Frankfurt)	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi
Eropa (Irlandia)	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi
Eropa (London)	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi
Europe (Milan)	–	–	–	–	–	–	–
Eropa (Paris)	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi
Eropa (Spanyol)	–	–	–	–	–	–	–
Eropa (Stockholm)	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi
Eropa (Zürich)	–	–	–	–	–	–	–
Israel (Tel Aviv)	–	–	–	–	–	–	–
Timur Tengah (Bahrain)	–	–	–	–	–	–	–

Wilayah	RDS for PostgreSQL 16	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
Timur Tengah (UEA)	–	–	–	–	–	–	–
Amerika Selatan (Sao Paulo)	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi
AWS GovCloud (AS-Timur)	–	–	–	–	–	–	–
AWS GovCloud (AS-Barat)	–	–	–	–	–	–	–

Autentikasi Kerberos dengan RDS for SQL Server

Wilayah dan versi mesin berikut tersedia untuk autentikasi Kerberos dengan RDS for SQL Server.

Wilayah	RDS for SQL Server 2019	RDS for SQL Server 2017	RDS for SQL Server 2016	RDS for SQL Server 2014
AS Timur (Ohio)	Semua versi	Semua versi	Semua versi	Semua versi
AS Timur (Virginia Utara)	Semua versi	Semua versi	Semua versi	Semua versi
AS Barat (California Utara)	Semua versi	Semua versi	Semua versi	Semua versi

Wilayah	RDS for SQL Server 2019	RDS for SQL Server 2017	RDS for SQL Server 2016	RDS for SQL Server 2014
AS Barat (Oregon)	Semua versi	Semua versi	Semua versi	Semua versi
Afrika (Cape Town)	Semua versi	Semua versi	Semua versi	Semua versi
Asia Pasifik (Hong Kong)	Semua versi	Semua versi	Semua versi	Semua versi
Asia Pasifik (Hyderabad)	Semua versi	Semua versi	Semua versi	Semua versi
Asia Pasifik (Jakarta)	Semua versi	Semua versi	Semua versi	Semua versi
Asia Pasifik (Melbourne)	Semua versi	Semua versi	Semua versi	Semua versi
Asia Pasifik (Mumbai)	Semua versi	Semua versi	Semua versi	Semua versi
Asia Pasifik (Osaka)	Semua versi	Semua versi	Semua versi	Semua versi
Asia Pasifik (Seoul)	Semua versi	Semua versi	Semua versi	Semua versi
Asia Pasifik (Singapura)	Semua versi	Semua versi	Semua versi	Semua versi
Asia Pasifik (Sydney)	Semua versi	Semua versi	Semua versi	Semua versi
Asia Pasifik (Tokyo)	Semua versi	Semua versi	Semua versi	Semua versi
Kanada (Pusat)	Semua versi	Semua versi	Semua versi	Semua versi

Wilayah	RDS for SQL Server 2019	RDS for SQL Server 2017	RDS for SQL Server 2016	RDS for SQL Server 2014
Kanada Barat (Calgary)	–	–	–	–
Tiongkok (Beijing)	Semua versi	Semua versi	Semua versi	Semua versi
Tiongkok (Ningxia)	Semua versi	Semua versi	Semua versi	Semua versi
Eropa (Frankfurt)	Semua versi	Semua versi	Semua versi	Semua versi
Eropa (Irlandia)	Semua versi	Semua versi	Semua versi	Semua versi
Eropa (London)	Semua versi	Semua versi	Semua versi	Semua versi
Europe (Milan)	Semua versi	Semua versi	Semua versi	Semua versi
Eropa (Paris)	Semua versi	Semua versi	Semua versi	Semua versi
Eropa (Spanyol)	Semua versi	Semua versi	Semua versi	Semua versi
Eropa (Stockholm)	Semua versi	Semua versi	Semua versi	Semua versi
Eropa (Zürich)	Semua versi	Semua versi	Semua versi	Semua versi
Israel (Tel Aviv)	–	–	–	–
Timur Tengah (Bahrain)	Semua versi	Semua versi	Semua versi	Semua versi
Timur Tengah (UAE)	Semua versi	Semua versi	Semua versi	Semua versi
Amerika Selatan (Sao Paulo)	Semua versi	Semua versi	Semua versi	Semua versi

Wilayah	RDS for SQL Server 2019	RDS for SQL Server 2017	RDS for SQL Server 2016	RDS for SQL Server 2014
AWS GovCloud (AS-Timur)	Semua versi	Semua versi	Semua versi	Semua versi
AWS GovCloud (AS-Barat)	Semua versi	Semua versi	Semua versi	Semua versi

Klaster DB Multi-AZ

Deployment klaster DB Multi-AZ di Amazon RDS menyediakan mode deployment ketersediaan tinggi Amazon RDS dengan dua instans DB siaga yang dapat dibaca. Klaster DB Multi-AZ memiliki instans DB penulis dan dua instans DB pembaca di tiga Zona Ketersediaan terpisah di Wilayah yang sama. Klaster DB Multi-AZ menyediakan ketersediaan tinggi, peningkatan kapasitas untuk beban kerja baca, dan latensi tulis yang lebih rendah jika dibandingkan dengan deployment instans DB Multi-AZ. Untuk informasi selengkapnya, lihat Deployment klaster basis data Multi-AZ.

Klaster DB Multi-AZ tidak tersedia dengan mesin berikut:

- RDS for Db2
- RDS for MariaDB
- RDS for Oracle
- RDS for SQL Server

Topik

- Klaster DB Multi-AZ dengan RDS for MySQL
- Klaster DB Multi-AZ dengan RDS for PostgreSQL

Klaster DB Multi-AZ dengan RDS for MySQL

Wilayah dan versi mesin berikut tersedia untuk klaster DB Multi-AZ dengan RDS for MySQL.

Wilayah	RDS for MySQL 8.0
AS Timur (Ohio)	Versi 8.0.28 dan yang lebih tinggi
AS Timur (Virginia Utara)	Versi 8.0.28 dan yang lebih tinggi
AS Barat (California Utara)	–
AS Barat (Oregon)	Versi 8.0.28 dan yang lebih tinggi
Afrika (Cape Town)	Versi 8.0.28 dan yang lebih tinggi
Asia Pasifik (Hong Kong)	Versi 8.0.28 dan yang lebih tinggi
Asia Pasifik (Hyderabad)	–
Asia Pasifik (Jakarta)	Versi 8.0.28 dan yang lebih tinggi
Asia Pasifik (Melbourne)	–
Asia Pasifik (Mumbai)	Versi 8.0.28 dan yang lebih tinggi
Asia Pasifik (Osaka)	Versi 8.0.28 dan yang lebih tinggi
Asia Pasifik (Seoul)	Versi 8.0.28 dan yang lebih tinggi
Asia Pasifik (Singapura)	Versi 8.0.28 dan yang lebih tinggi
Asia Pasifik (Sydney)	Versi 8.0.28 dan yang lebih tinggi
Asia Pasifik (Tokyo)	Versi 8.0.28 dan yang lebih tinggi
Kanada (Pusat)	Versi 8.0.28 dan yang lebih tinggi
Kanada (Pusat)	Versi 8.0.28 dan yang lebih tinggi
Kanada Barat (Calgary)	Versi 8.0.28 dan yang lebih tinggi
Tiongkok (Beijing)	Versi 8.0.28 dan yang lebih tinggi
Tiongkok (Ningxia)	Versi 8.0.28 dan yang lebih tinggi

Wilayah	RDS for MySQL 8.0
Eropa (Frankfurt)	Versi 8.0.28 dan yang lebih tinggi
Eropa (Irlandia)	Versi 8.0.28 dan yang lebih tinggi
Eropa (London)	Versi 8.0.28 dan yang lebih tinggi
Europe (Milan)	Versi 8.0.28 dan yang lebih tinggi
Eropa (Paris)	Versi 8.0.28 dan yang lebih tinggi
Eropa (Spanyol)	–
Eropa (Stockholm)	Versi 8.0.28 dan yang lebih tinggi
Eropa (Zürich)	–
Israel (Tel Aviv)	–
Timur Tengah (Bahrain)	Versi 8.0.28 dan yang lebih tinggi
Timur Tengah (UEA)	–
Amerika Selatan (Sao Paulo)	Versi 8.0.28 dan yang lebih tinggi
AWS GovCloud (AS-Timur)	–
AWS GovCloud (AS-Barat)	–

Anda juga dapat mencantumkan versi yang tersedia di suatu Wilayah untuk kelas instans DB db.r5d.large dengan menjalankan perintah berikut AWS CLI.

Untuk Linux, macOS, atau Unix:

```
aws rds describe-orderable-db-instance-options \
--engine mysql \
--db-instance-class db.r5d.large \
--query '*[]|[?SupportsClusters == `true`].[EngineVersion]'  \
--output text
```

Untuk Windows:

```
aws rds describe-orderable-db-instance-options ^
--engine mysql ^
--db-instance-class db.r5d.large ^
--query "*[]|[?SupportsClusters == `true`].[EngineVersion]"  ^
--output text
```

Anda dapat mengubah kelas instans DB untuk menampilkan versi mesin yang tersedia.

Klaster DB Multi-AZ dengan RDS for PostgreSQL

Wilayah dan versi mesin berikut tersedia untuk klaster DB Multi-AZ dengan RDS for PostgreSQL.

Wilayah	RDS for PostgreSQL 16	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13
AS Timur (Ohio)	Semua versi PostgreSQL 16	Semua versi PostgreSQL 15	Versi 14.5 dan yang lebih tinggi	Versi 13.4 dan versi 13.7 dan yang lebih tinggi
AS Timur (Virginia Utara)	Semua versi PostgreSQL 16	Semua versi PostgreSQL 15	Versi 14.5 dan yang lebih tinggi	Versi 13.4 dan versi 13.7 dan yang lebih tinggi
AS Barat (California Utara)	–	–	–	–
AS Barat (Oregon)	Semua versi PostgreSQL 16	Semua versi PostgreSQL 15	Versi 14.5 dan yang lebih tinggi	Versi 13.4 dan versi 13.7 dan yang lebih tinggi
Afrika (Cape Town)	Semua versi PostgreSQL 16	Semua versi PostgreSQL 15	Versi 14.5 dan yang lebih tinggi	Versi 13.4 dan versi 13.7 dan yang lebih tinggi
Asia Pasifik (Hong Kong)	Semua versi PostgreSQL 16	Semua versi PostgreSQL 15	Versi 14.5 dan yang lebih tinggi	Versi 13.4 dan versi 13.7 dan yang lebih tinggi

Wilayah	RDS for PostgreSQL 16	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13
Asia Pasifik (Hyderabad)	–	–	–	–
Asia Pasifik (Jakarta)	Semua versi PostgreSQL 16	Semua versi PostgreSQL 15	Versi 14.5 dan yang lebih tinggi	Versi 13.4 dan versi 13.7 dan yang lebih tinggi
Asia Pasifik (Melbourne)	–	–	–	–
Asia Pasifik (Mumbai)	Semua versi PostgreSQL 16	Semua versi PostgreSQL 15	Versi 14.5 dan yang lebih tinggi	Versi 13.4 dan versi 13.7 dan yang lebih tinggi
Asia Pasifik (Osaka)	Semua versi PostgreSQL 16	Semua versi PostgreSQL 15	Versi 14.5 dan yang lebih tinggi	Versi 13.4 dan versi 13.7 dan yang lebih tinggi
Asia Pasifik (Seoul)	Semua versi PostgreSQL 16	Semua versi PostgreSQL 15	Versi 14.5 dan yang lebih tinggi	Versi 13.4 dan versi 13.7 dan yang lebih tinggi
Asia Pasifik (Singapura)	Semua versi PostgreSQL 16	Semua versi PostgreSQL 15	Versi 14.5 dan yang lebih tinggi	Versi 13.4 dan versi 13.7 dan yang lebih tinggi
Asia Pasifik (Sydney)	Semua versi PostgreSQL 16	Semua versi PostgreSQL 15	Versi 14.5 dan yang lebih tinggi	Versi 13.4 dan versi 13.7 dan yang lebih tinggi
Asia Pasifik (Tokyo)	Semua versi PostgreSQL 16	Semua versi PostgreSQL 15	Versi 14.5 dan yang lebih tinggi	Versi 13.4 dan versi 13.7 dan yang lebih tinggi

Wilayah	RDS for PostgreSQL 16	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13
Kanada (Pusat)	Semua versi PostgreSQL 16	Semua versi PostgreSQL 15	Versi 14.5 dan yang lebih tinggi	Versi 13.4 dan versi 13.7 dan yang lebih tinggi
Kanada Barat (Calgary)	Semua versi PostgreSQL 16	Semua versi PostgreSQL 15	Versi 14.5 dan yang lebih tinggi	Versi 13.4 dan versi 13.7 dan yang lebih tinggi
Tiongkok (Beijing)	Semua versi PostgreSQL 16	Semua versi PostgreSQL 15	Versi 14.5 dan yang lebih tinggi	Versi 13.4 dan versi 13.7 dan yang lebih tinggi
Tiongkok (Ningxia)	Semua versi PostgreSQL 16	Semua versi PostgreSQL 15	Versi 14.5 dan yang lebih tinggi	Versi 13.4 dan versi 13.7 dan yang lebih tinggi
Eropa (Frankfurt)	Semua versi PostgreSQL 16	Semua versi PostgreSQL 15	Versi 14.5 dan yang lebih tinggi	Versi 13.4 dan versi 13.7 dan yang lebih tinggi
Eropa (Irlandia)	Semua versi PostgreSQL 16	Semua versi PostgreSQL 15	Versi 14.5 dan yang lebih tinggi	Versi 13.4 dan versi 13.7 dan yang lebih tinggi
Eropa (London)	Semua versi PostgreSQL 16	Semua versi PostgreSQL 15	Versi 14.5 dan yang lebih tinggi	Versi 13.4 dan versi 13.7 dan yang lebih tinggi
Europe (Milan)	Semua versi PostgreSQL 16	Semua versi PostgreSQL 15	Versi 14.5 dan yang lebih tinggi	Versi 13.4 dan versi 13.7 dan yang lebih tinggi
Eropa (Paris)	Semua versi PostgreSQL 16	Semua versi PostgreSQL 15	Versi 14.5 dan yang lebih tinggi	Versi 13.4 dan versi 13.7 dan yang lebih tinggi

Wilayah	RDS for PostgreSQL 16	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13
Eropa (Spanyol)	–	–	–	–
Eropa (Stockholm)	Semua versi PostgreSQL 16	Semua versi PostgreSQL 15	Versi 14.5 dan yang lebih tinggi	Versi 13.4 dan versi 13.7 dan yang lebih tinggi
Eropa (Zürich)	–	–	–	–
Israel (Tel Aviv)	–	–	–	–
Timur Tengah (Bahrain)	Semua versi PostgreSQL 16	Semua versi PostgreSQL 15	Versi 14.5 dan yang lebih tinggi	Versi 13.4 dan versi 13.7 dan yang lebih tinggi
Timur Tengah (UEA)	–	–	–	–
Amerika Selatan (Sao Paulo)	Semua versi PostgreSQL 16	Semua versi PostgreSQL 15	Versi 14.5 dan yang lebih tinggi	Versi 13.4 dan versi 13.7 dan yang lebih tinggi
AWS GovCloud (AS-Timur)	–	–	–	
AWS GovCloud (AS-Barat)	–	–	–	–

Anda juga dapat mencantumkan versi yang tersedia di suatu Wilayah untuk kelas instans DB db.r5d.large dengan menjalankan perintah berikut AWS CLI.

Untuk Linux, macOS, atau Unix:

```
aws rds describe-orderable-db-instance-options \
--engine postgres \
--db-instance-class db.r5d.large \
--query '*[]|[?SupportsClusters == `true`].[EngineVersion]'  \
```

```
--output text
```

Untuk Windows:

```
aws rds describe-orderable-db-instance-options ^
--engine postgres ^
--db-instance-class db.r5d.large ^
--query "*[]|[?SupportsClusters == `true`].[EngineVersion]"  ^
--output text
```

Anda dapat mengubah kelas instans DB untuk menampilkan versi mesin yang tersedia.

Wawasan Performa

Wawasan Performa di Amazon RDS memperluas fitur pemantauan Amazon RDS yang sudah ada untuk mengilustrasikan dan membantu Anda menganalisis performa basis data. Dengan dasbor Wawasan Performa, Anda dapat memvisualisasikan pemuatan basis data di instans DB Amazon RDS. Anda juga dapat memfilter pemuatan berdasarkan tunggu, pernyataan SQL, host, atau pengguna. Untuk informasi selengkapnya, lihat Memantau muatan DB dengan Wawasan Performa di Amazon RDS.

Wawasan Performa tersedia untuk semua mesin DB RDS, kecuali RDS for Db2.

Untuk mesin DB yang tersedia, Wawasan Performa tersedia dengan semua versi mesin yang tersedia dan di semua Wilayah AWS.

Untuk informasi dukungan Wilayah, mesin DB, dan kelas instans untuk fitur Wawasan Performa, lihat Dukungan kelas instans, Wilayah, dan mesin DB Amazon RDS untuk fitur Wawasan Performa.

RDS Custom

Amazon RDS Custom mengotomatiskan tugas dan operasi administrasi basis data. Dengan menggunakan RDS Custom, Anda sebagai administrator basis data dapat mengakses dan menyesuaikan lingkungan basis data dan sistem operasi Anda. Dengan RDS Custom, Anda dapat melakukan penyesuaian untuk memenuhi persyaratan aplikasi lama, kustom, dan yang dipaketkan. Untuk informasi selengkapnya, lihat Menggunakan Amazon RDS Custom.

RDS Custom hanya didukung untuk mesin DB berikut:

Topik

- RDS Custom for Oracle

- RDS Custom for SQL Server

RDS Custom for Oracle

Wilayah dan versi mesin berikut tersedia untuk RDS Custom for Oracle.

Wilayah	Oracle Database 19c	Oracle Database 18c	Oracle Database 12c
AS Timur (Ohio)	19c dengan RU/RUR Januari 2021 atau yang lebih tinggi	18c dengan RU/RUR Januari 2021 atau yang lebih tinggi	12.1 dan 12.2 dengan RU/RUR Januari 2021 atau yang lebih tinggi
AS Timur (Virginia Utara)	19c dengan RU/RUR Januari 2021 atau yang lebih tinggi	18c dengan RU/RUR Januari 2021 atau yang lebih tinggi	12.1 dan 12.2 dengan RU/RUR Januari 2021 atau yang lebih tinggi
AS Barat (California Utara)	–	–	–
AS Barat (Oregon)	19c dengan RU/RUR Januari 2021 atau yang lebih tinggi	18c dengan RU/RUR Januari 2021 atau yang lebih tinggi	12.1 dan 12.2 dengan RU/RUR Januari 2021 atau yang lebih tinggi
Afrika (Cape Town)	–	–	–
Asia Pasifik (Hong Kong)	–	–	–
Asia Pasifik (Jakarta)	19c dengan RU/RUR Januari 2021 atau yang lebih tinggi	18c dengan RU/RUR Januari 2021 atau yang lebih tinggi	12.1 dan 12.2 dengan RU/RUR Januari 2021 atau yang lebih tinggi
Asia Pasifik (Melbourne)	–	–	–
Asia Pasifik (Mumbai)	19c dengan RU/RUR Januari 2021 atau yang lebih tinggi	18c dengan RU/RUR Januari 2021 atau yang lebih tinggi	12.1 dan 12.2 dengan RU/RUR Januari 2021 atau yang lebih tinggi

Wilayah	Oracle Database 19c	Oracle Database 18c	Oracle Database 12c
Asia Pasifik (Osaka)	19c dengan RU/RUR Januari 2021 atau yang lebih tinggi	18c dengan RU/RUR Januari 2021 atau yang lebih tinggi	12.1 dan 12.2 dengan RU/RUR Januari 2021 atau yang lebih tinggi
Asia Pasifik (Seoul)	19c dengan RU/RUR Januari 2021 atau yang lebih tinggi	18c dengan RU/RUR Januari 2021 atau yang lebih tinggi	12.1 dan 12.2 dengan RU/RUR Januari 2021 atau yang lebih tinggi
Asia Pasifik (Singapura)	19c dengan RU/RUR Januari 2021 atau yang lebih tinggi	18c dengan RU/RUR Januari 2021 atau yang lebih tinggi	12.1 dan 12.2 dengan RU/RUR Januari 2021 atau yang lebih tinggi
Asia Pasifik (Sydney)	19c dengan RU/RUR Januari 2021 atau yang lebih tinggi	18c dengan RU/RUR Januari 2021 atau yang lebih tinggi	12.1 dan 12.2 dengan RU/RUR Januari 2021 atau yang lebih tinggi
Asia Pasifik (Tokyo)	19c dengan RU/RUR Januari 2021 atau yang lebih tinggi	18c dengan RU/RUR Januari 2021 atau yang lebih tinggi	12.1 dan 12.2 dengan RU/RUR Januari 2021 atau yang lebih tinggi
Kanada (Pusat)	19c dengan RU/RUR Januari 2021 atau yang lebih tinggi	18c dengan RU/RUR Januari 2021 atau yang lebih tinggi	12.1 dan 12.2 dengan RU/RUR Januari 2021 atau yang lebih tinggi
Kanada Barat (Calgary)	–	–	–
China (Beijing)	–	–	–
Tiongkok (Ningxia)	–	–	–
Eropa (Frankfurt)	19c dengan RU/RUR Januari 2021 atau yang lebih tinggi	18c dengan RU/RUR Januari 2021 atau yang lebih tinggi	12.1 dan 12.2 dengan RU/RUR Januari 2021 atau yang lebih tinggi

Wilayah	Oracle Database 19c	Oracle Database 18c	Oracle Database 12c
Eropa (Irlandia)	19c dengan RU/RUR Januari 2021 atau yang lebih tinggi	18c dengan RU/RUR Januari 2021 atau yang lebih tinggi	12.1 dan 12.2 dengan RU/RUR Januari 2021 atau yang lebih tinggi
Eropa (London)	19c dengan RU/RUR Januari 2021 atau yang lebih tinggi	18c dengan RU/RUR Januari 2021 atau yang lebih tinggi	12.1 dan 12.2 dengan RU/RUR Januari 2021 atau yang lebih tinggi
Eropa (Milan)	19c dengan RU/RUR Januari 2021 atau yang lebih tinggi	18c dengan RU/RUR Januari 2021 atau yang lebih tinggi	12.1 dan 12.2 dengan RU/RUR Januari 2021 atau yang lebih tinggi
Eropa (Paris)	19c dengan RU/RUR Januari 2021 atau yang lebih tinggi	18c dengan RU/RUR Januari 2021 atau yang lebih tinggi	12.1 dan 12.2 dengan RU/RUR Januari 2021 atau yang lebih tinggi
Eropa (Stockholm)	19c dengan RU/RUR Januari 2021 atau yang lebih tinggi	18c dengan RU/RUR Januari 2021 atau yang lebih tinggi	12.1 dan 12.2 dengan RU/RUR Januari 2021 atau yang lebih tinggi
Israel (Tel Aviv)	–	–	–
Timur Tengah (Bahrain)	–	–	–
Middle East (UAE)	19c dengan RU/RUR Januari 2021 atau yang lebih tinggi	18c dengan RU/RUR Januari 2021 atau yang lebih tinggi	12.1 dan 12.2 dengan RU/RUR Januari 2021 atau yang lebih tinggi
Amerika Selatan (Sao Paulo)	19c dengan RU/RUR Januari 2021 atau yang lebih tinggi	18c dengan RU/RUR Januari 2021 atau yang lebih tinggi	12.1 dan 12.2 dengan RU/RUR Januari 2021 atau yang lebih tinggi
AWS GovCloud (AS-Timur)	19c dengan RU/RUR Januari 2021 atau yang lebih tinggi	18c dengan RU/RUR Januari 2021 atau yang lebih tinggi	12.1 dan 12.2 dengan RU/RUR Januari 2021 atau yang lebih tinggi

Wilayah	Oracle Database 19c	Oracle Database 18c	Oracle Database 12c
AWS GovCloud (AS-Barat)	19c dengan RU/RUR Januari 2021 atau yang lebih tinggi	18c dengan RU/RUR Januari 2021 atau yang lebih tinggi	12.1 dan 12.2 dengan RU/RUR Januari 2021 atau yang lebih tinggi

RDS Custom for SQL Server

Anda dapat menerapkan RDS Custom untuk SQL Server dengan menggunakan versi mesin yang disediakan RDS (RPEV) atau versi mesin kustom (CEV):

- Jika Anda menggunakan RPEV, instalasi SQL Server dan Amazon Machine Image (AMI) default juga disertakan. Jika Anda menyesuaikan atau memodifikasi sistem operasi (OS), perubahan Anda mungkin tidak akan bertahan selama patching, pemulihan snapshot, atau pemulihan otomatis.
- Jika Anda menggunakan CEV, Anda memilih AMI Anda sendiri dengan Microsoft SQL Server atau SQL Server pra-instal yang Anda instal menggunakan media Anda sendiri. Saat menggunakan CEV yang AWS disediakan, Anda memilih gambar Amazon EC2 (AMI) terbaru yang tersedia AWS, yang memiliki pemutakhiran kumulatif (CU) yang didukung oleh RDS Custom for SQL Server. Dengan CEV, Anda dapat menyesuaikan konfigurasi OS dan SQL Server untuk memenuhi kebutuhan perusahaan Anda.

Versi mesin berikut Wilayah AWS dan DB tersedia untuk RDS Custom untuk SQL Server. Dukungan versi mesin tergantung pada apakah Anda menggunakan RDS Custom for SQL Server dengan RPEV, CEV yang disediakan AWS , atau CEV yang disediakan pelanggan.

Wilayah	RPEV	AWS disediakan CEV	CEV yang disediakan pelanggan
AS Timur (Ohio)	SQL Server 2022 Perusahaan, Standar, atau Web, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Web, dengan CU8, CU17, CU18, CU20, CU22	SQL Server 2022 Perusahaan, Standar, atau Web, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Web, dengan CU17, CU18, CU20, CU22	SQL Server 2022 Perusahaan, Standar, atau Pengembang, dengan CU9. SQL Server 2019 Enterpris e, Standard, atau Developer, dengan

Wilayah	RPEV	AWS disediakan CEV	CEV yang disediakan pelanggan
			CU17, CU18, CU20, CU22
AS Timur (Virginia Utara)	SQL Server 2022 Perusahaan, Standar, atau Web, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Web, dengan CU8, CU17, CU18, CU20, CU22	SQL Server 2022 Perusahaan, Standar, atau Web, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Web, dengan CU17, CU18, CU20, CU22	SQL Server 2022 Perusahaan, Standar, atau Pengembang, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Developer, dengan CU17, CU18, CU20, CU22
AS Barat (California Utara)	–	–	–
AS Barat (Oregon)	SQL Server 2022 Perusahaan, Standar, atau Web, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Web, dengan CU8, CU17, CU18, CU20, CU22	SQL Server 2022 Perusahaan, Standar, atau Web, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Web, dengan CU17, CU18, CU20, CU22	SQL Server 2022 Perusahaan, Standar, atau Pengembang, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Developer, dengan CU17, CU18, CU20, CU22
Afrika (Cape Town)	–	–	–
Asia Pasifik (Hong Kong)	–	–	–
Asia Pasifik (Hyderabad)	–	–	–
Asia Pasifik (Jakarta)	–	–	–

Wilayah	RPEV	AWS disediakan CEV	CEV yang disediakan pelanggan
Asia Pasifik (Melbourne)	–	–	–
Asia Pasifik (Mumbai)	SQL Server 2022 Perusahaan, Standar, atau Web, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Web, dengan CU8, CU17, CU18, CU20, CU22	SQL Server 2022 Perusahaan, Standar, atau Web, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Web, dengan CU17, CU18, CU20, CU22	SQL Server 2022 Perusahaan, Standar, atau Pengembang, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Developer, dengan CU17, CU18, CU20, CU22
Asia Pasifik (Osaka)	–	–	–
Asia Pasifik (Seoul)	SQL Server 2022 Perusahaan, Standar, atau Web, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Web, dengan CU8, CU17, CU18, CU20, CU22	SQL Server 2022 Perusahaan, Standar, atau Web, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Web, dengan CU17, CU18, CU20, CU22	SQL Server 2022 Perusahaan, Standar, atau Pengembang, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Developer, dengan CU17, CU18, CU20, CU22

Wilayah	RPEV	AWS disediakan CEV	CEV yang disediakan pelanggan
Asia Pasifik (Singapura)	SQL Server 2022 Perusahaan, Standar, atau Web, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Web, dengan CU8, CU17, CU18, CU20, CU22	SQL Server 2022 Perusahaan, Standar, atau Web, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Web, dengan CU17, CU18, CU20, CU22	SQL Server 2022 Perusahaan, Standar, atau Pengembang, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Developer, dengan CU17, CU18, CU20, CU22
Asia Pasifik (Sydney)	SQL Server 2022 Perusahaan, Standar, atau Web, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Web, dengan CU8, CU17, CU18, CU20, CU22	SQL Server 2022 Perusahaan, Standar, atau Web, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Web, dengan CU17, CU18, CU20, CU22	SQL Server 2022 Perusahaan, Standar, atau Pengembang, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Developer, dengan CU17, CU18, CU20, CU22
Asia Pasifik (Tokyo)	SQL Server 2022 Perusahaan, Standar, atau Web, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Web, dengan CU8, CU17, CU18, CU20, CU22	SQL Server 2022 Perusahaan, Standar, atau Web, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Web, dengan CU17, CU18, CU20, CU22	SQL Server 2022 Perusahaan, Standar, atau Pengembang, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Developer, dengan CU17, CU18, CU20, CU22

Wilayah	RPEV	AWS disediakan CEV	CEV yang disediakan pelanggan
Kanada (Pusat)	SQL Server 2022 Perusahaan, Standar, atau Web, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Web, dengan CU8, CU17, CU18, CU20, CU22	SQL Server 2022 Perusahaan, Standar, atau Web, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Web, dengan CU17, CU18, CU20, CU22	SQL Server 2022 Perusahaan, Standar, atau Pengembang, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Developer, dengan CU17, CU18, CU20, CU22
Kanada Barat (Calgary)	–	–	–
China (Beijing)	–	–	–
Tiongkok (Ningxia)	–	–	–
Eropa (Frankfurt)	SQL Server 2022 Perusahaan, Standar, atau Web, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Web, dengan CU8, CU17, CU18, CU20, CU22	SQL Server 2022 Perusahaan, Standar, atau Web, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Web, dengan CU17, CU18, CU20, CU22	SQL Server 2022 Perusahaan, Standar, atau Pengembang, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Developer, dengan CU17, CU18, CU20, CU22

Wilayah	RPEV	AWS disediakan CEV	CEV yang disediakan pelanggan
Eropa (Irlandia)	SQL Server 2022 Perusahaan, Standar, atau Web, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Web, dengan CU8, CU17, CU18, CU20, CU22	SQL Server 2022 Perusahaan, Standar, atau Web, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Web, dengan CU17, CU18, CU20, CU22	SQL Server 2022 Perusahaan, Standar, atau Pengembang, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Developer, dengan CU17, CU18, CU20, CU22
Eropa (London)	SQL Server 2022 Perusahaan, Standar, atau Web, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Web, dengan CU8, CU17, CU18, CU20, CU22	SQL Server 2022 Perusahaan, Standar, atau Web, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Web, dengan CU17, CU18, CU20, CU22	SQL Server 2022 Perusahaan, Standar, atau Pengembang, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Developer, dengan CU17, CU18, CU20, CU22
Eropa (Milan)	–	–	–
Eropa (Paris)	–	–	–
Eropa (Spanyol)	–	–	–
Eropa (Stockholm)	SQL Server 2022 Perusahaan, Standar, atau Web, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Web, dengan CU8, CU17, CU18, CU20, CU22	SQL Server 2022 Perusahaan, Standar, atau Web, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Web, dengan CU17, CU18, CU20, CU22	SQL Server 2022 Perusahaan, Standar, atau Pengembang, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Developer, dengan CU17, CU18, CU20, CU22

Wilayah	RPEV	AWS disediakan CEV	CEV yang disediakan pelanggan
Eropa (Zürich)	–	–	–
Israel (Tel Aviv)	–	–	–
Timur Tengah (Bahrain)	–	–	–
Timur Tengah (UEA)	–	–	–
Amerika Selatan (Sao Paulo)	SQL Server 2022 Perusahaan, Standar, atau Web, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Web, dengan CU8, CU17, CU18, CU20, CU22	SQL Server 2022 Perusahaan, Standar, atau Web, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Web, dengan CU17, CU18, CU20, CU22	SQL Server 2022 Perusahaan, Standar, atau Pengembang, dengan CU9. SQL Server 2019 Enterprise, Standard, atau Developer, dengan CU17, CU18, CU20, CU22
AWS GovCloud (AS-Timur)	–	–	–
AWS GovCloud (AS-Barat)	–	–	–

Proksi Amazon RDS

Proksi Amazon RDS adalah proksi basis data yang sepenuhnya terkelola dengan ketersediaan tinggi yang membuat aplikasi lebih dapat ditingkatkan dengan menggabungkan dan berbagi koneksi basis data yang telah dibuat. Untuk informasi selengkapnya, lihat Menggunakan Proksi Amazon RDS.

Proksi RDS tidak tersedia untuk mesin berikut ini:

- RDS for Db2

- RDS for Oracle

Topik

Proksi RDS dengan RDS for MariaDB

Wilayah dan versi mesin berikut tersedia untuk Proksi RDS dengan RDS for MariaDB.

Wilayah	RDS for MariaDB 10.11	RDS for MariaDB 10.6	RDS for MariaDB 10.5	RDS for MariaDB 10.4	RDS for MariaDB 10.3
AS Timur (Ohio)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AS Timur (Virginia Utara)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AS Barat (California Utara)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AS Barat (Oregon)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Afrika (Cape Town)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Hong Kong)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia

Wilayah	RDS for MariaDB 10.11	RDS for MariaDB 10.6	RDS for MariaDB 10.5	RDS for MariaDB 10.4	RDS for MariaDB 10.3
Asia Pasifik (Hyderabad)	–	–	–	–	–
Asia Pasifik (Jakarta)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Melbourne)	–	–	–	–	–
Asia Pasifik (Mumbai)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Osaka)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Seoul)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Singapura)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Sydney)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Tokyo)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Kanada (Pusat)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Kanada Barat (Calgary)	–	–	–	–	–
Tiongkok (Beijing)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia

Wilayah	RDS for MariaDB 10.11	RDS for MariaDB 10.6	RDS for MariaDB 10.5	RDS for MariaDB 10.4	RDS for MariaDB 10.3
Tiongkok (Ningxia)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Frankfurt)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Irlandia)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (London)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Europe (Milan)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Paris)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Spanyol)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Stockholm)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Zürich)	Semua versi yang tersedia	Semua versi yang tersedia	–	–	–
Israel (Tel Aviv)	–	–	–	–	–
Timur Tengah (Bahrain)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia

Wilayah	RDS for MariaDB 10.11	RDS for MariaDB 10.6	RDS for MariaDB 10.5	RDS for MariaDB 10.4	RDS for MariaDB 10.3
Timur Tengah (UEA)	–	–	–	–	–
Amerika Selatan (Sao Paulo)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AWS GovCloud (AS-Timur)	–	–	–	–	–
AWS GovCloud (AS-Barat)	–	–	–	–	–

Proksi RDS dengan RDS for MySQL

Wilayah dan versi mesin berikut tersedia untuk Proksi RDS dengan RDS for MySQL.

Wilayah	RDS for MySQL 8.0	RDS for MySQL 5.7
AS Timur (Ohio)	Semua versi yang tersedia	Semua versi yang tersedia
AS Timur (Virginia Utara)	Semua versi yang tersedia	Semua versi yang tersedia
AS Barat (California Utara)	Semua versi yang tersedia	Semua versi yang tersedia
AS Barat (Oregon)	Semua versi yang tersedia	Semua versi yang tersedia
Afrika (Cape Town)	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Hong Kong)	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Hyderabad)	–	–

Wilayah	RDS for MySQL 8.0	RDS for MySQL 5.7
Asia Pasifik (Jakarta)	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Melbourne)	–	–
Asia Pasifik (Mumbai)	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Osaka)	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Seoul)	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Singapura)	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Sydney)	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Tokyo)	Semua versi yang tersedia	Semua versi yang tersedia
Kanada (Pusat)	Semua versi yang tersedia	Semua versi yang tersedia
Kanada Barat (Calgary)	–	–
Tiongkok (Beijing)	Semua versi yang tersedia	Semua versi yang tersedia
Tiongkok (Ningxia)	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Frankfurt)	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Irlandia)	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (London)	Semua versi yang tersedia	Semua versi yang tersedia
Europe (Milan)	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Paris)	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Spanyol)	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Stockholm)	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Zürich)	–	–

Wilayah	RDS for MySQL 8.0	RDS for MySQL 5.7
Israel (Tel Aviv)	–	–
Timur Tengah (Bahrain)	Semua versi yang tersedia	Semua versi yang tersedia
Timur Tengah (UEA)	–	–
Amerika Selatan (Sao Paulo)	Semua versi yang tersedia	Semua versi yang tersedia
AWS GovCloud (AS-Timur)	–	–
AWS GovCloud (AS-Barat)	–	–

Proksi RDS dengan RDS for PostgreSQL

Wilayah dan versi mesin berikut tersedia untuk Proksi RDS dengan RDS for PostgreSQL.

Wilayah	RDS for PostgreSQL 16	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
AS Timur (Ohio)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AS Timur (Virginia Utara)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AS Barat (California Utara)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AS Barat (Oregon)	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi	Semua versi

Wilayah	RDS for PostgreSQL 16	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
	yang tersedia	yang tersedia	yang tersedia	yang tersedia	yang tersedia	yang tersedia	yang tersedia
Afrika (Cape Town)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Hong Kong)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Hyderabad)	–	–	–	–	–	–	–
Asia Pasifik (Jakarta)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Melbourne)	–	–	–	–	–	–	–
Asia Pasifik (Mumbai)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia

Wilayah	RDS for PostgreSQL 16	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
Asia Pasifik (Osaka)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Seoul)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Singapura)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Sydney)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Tokyo)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Kanada (Pusat)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Kanada Barat (Calgary)	–	–	–	–	–	–	–

Wilayah	RDS for PostgreSQL 16	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
Tiongkok (Beijing)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Tiongkok (Ningxia)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Frankfurt)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Irlandia)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (London)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Europe (Milan)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Paris)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia

Wilayah	RDS for PostgreSQL 16	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
Eropa (Spanyol)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Stockholm)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Zürich)	–	–	–	–	–	–	–
Israel (Tel Aviv)	–	–	–	–	–	–	–
Timur Tengah (Bahrain)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Timur Tengah (UEA)	–	–	–	–	–	–	–
Amerika Selatan (Sao Paulo)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AWS GovCloud (AS-Timur)	–	–	–	–	–	–	–

Wilayah	RDS for PostgreSQL 16	RDS for PostgreSQL 15	RDS for PostgreSQL 14	RDS for PostgreSQL 13	RDS for PostgreSQL 12	RDS for PostgreSQL 11	RDS for PostgreSQL 10
AWS GovCloud (AS-Barat)	–	–	–	–	–	–	–

Proksi RDS dengan RDS for SQL Server

Wilayah dan versi mesin berikut tersedia untuk Proksi RDS dengan RDS for SQL Server.

Wilayah	RDS for SQL Server 2019	RDS for SQL Server 2017	RDS for SQL Server 2016	RDS for SQL Server 2014
AS Timur (Ohio)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AS Timur (Virginia Utara)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AS Barat (California Utara)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AS Barat (Oregon)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Afrika (Cape Town)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Hong Kong)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Hyderabad)	–	–	–	–

Wilayah	RDS for SQL Server 2019	RDS for SQL Server 2017	RDS for SQL Server 2016	RDS for SQL Server 2014
Asia Pasifik (Jakarta)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Melbourne)	–	–	–	–
Asia Pasifik (Mumbai)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Osaka)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Seoul)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Singapura)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Sydney)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Asia Pasifik (Tokyo)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Kanada (Pusat)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Kanada Barat (Calgary)	–	–	–	–
Tiongkok (Beijing)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Tiongkok (Ningxia)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia

Wilayah	RDS for SQL Server 2019	RDS for SQL Server 2017	RDS for SQL Server 2016	RDS for SQL Server 2014
Eropa (Frankfurt)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Irlandia)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (London)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Europe (Milan)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Paris)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Spanyol)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Stockholm)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Eropa (Zürich)	–	–	–	–
Israel (Tel Aviv)	–	–	–	–
Timur Tengah (Bahrain)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
Timur Tengah (UEA)	–	–	–	–
Amerika Selatan (Sao Paulo)	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia	Semua versi yang tersedia
AWS GovCloud (AS-Timur)	–	–	–	–

Wilayah	RDS for SQL Server 2019	RDS for SQL Server 2017	RDS for SQL Server 2016	RDS for SQL Server 2014
AWS GovCloud (AS-Barat)	–	–	–	–

Integrasi Secrets Manager

Dengan AWS Secrets Manager, Anda dapat mengganti kredensi hard-code dalam kode Anda, termasuk kata sandi database, dengan panggilan API ke Secrets Manager untuk mengambil rahasia secara terprogram. Untuk informasi selengkapnya tentang Secrets Manager, lihat Panduan Pengguna AWS Secrets Manager.

Anda dapat menentukan Amazon RDS untuk mengelola kata sandi pengguna utama di Secrets Manager untuk instans DB Amazon RDS atau klaster DB Multi-AZ. RDS menghasilkan kata sandi, menyimpannya di Secrets Manager, dan merotasinya secara teratur. Untuk informasi selengkapnya, lihat Manajemen kata sandi dengan Amazon RDS Aurora dan AWS Secrets Manager.

Integrasi Secrets Manager didukung untuk semua mesin DB RDS dan semua versi.

Integrasi Secrets Manager didukung di semua Wilayah AWS kecuali yang berikut:

- Kanada Barat (Calgary)
- AWS GovCloud (AS-Timur)
- AWS GovCloud (AS-Barat)

Integrasi nol-ETL dengan Amazon Redshift

Integrasi nol-ETL RDS dengan Amazon Redshift adalah solusi yang dikelola sepenuhnya untuk menyediakan data transaksional di Amazon Redshift setelah ditulis ke instan DB Amazon RDS. Untuk informasi selengkapnya, lihat Menggunakan integrasi nol-ETL (pratinjau).

Wilayah dan versi berikut tersedia untuk integrasi nol-ETL dengan Amazon Redshift.

Wilayah	RDS for MySQL 8.0
AS Timur (Virginia Utara)	Versi 8.0.28 dan yang lebih tinggi

Wilayah	RDS for MySQL 8.0
AS Timur (Ohio)	Versi 8.0.28 dan yang lebih tinggi
AS Barat (Oregon)	Versi 8.0.28 dan yang lebih tinggi
Asia Pasifik (Tokyo)	Versi 8.0.28 dan yang lebih tinggi
Eropa (Irlandia)	Versi 8.0.28 dan yang lebih tinggi

Fitur asli mesin

Mesin basis data Amazon RDS juga mendukung banyak fitur dan fungsionalitas asli mesin yang paling umum. Fitur-fitur ini berbeda dari fitur asli Amazon RDS yang tercantum di halaman ini. Beberapa fitur asli mesin mungkin memiliki dukungan yang terbatas atau hak istimewa yang dibatasi.

Untuk informasi selengkapnya tentang fitur asli mesin, lihat:

- Fitur-fitur RDS for Db2
- Dukungan fitur MariaDB di Amazon RDS
- Dukungan fitur MySQL di Amazon RDS
- Fitur-fitur RDS for Oracle
- Menggunakan fitur PostgreSQL yang didukung oleh Amazon RDS for PostgreSQL
- Fitur Microsoft SQL Server di Amazon RDS

Penagihan instans DB untuk Amazon RDS

Instans Amazon RDS ditagih berdasarkan komponen berikut:

- Jam instans DB (per jam) – Berdasarkan kelas instans DB dari instans DB (misalnya, db.t2.small atau db.m4.large). Harga dicantumkan per jam, tetapi tagihan dihitung turun menjadi detik dan menunjukkan waktu dalam bentuk desimal. Penggunaan RDS ditagih dalam setiap kenaikan 1 detik, dengan minimal 10 menit. Untuk mengetahui informasi selengkapnya, lihat Kelas instans DB .
- Penyimpanan (per GiB per bulan) – Kapasitas penyimpanan yang telah Anda sediakan untuk instans DB Anda. Jika Anda menskalakan kapasitas penyimpanan yang telah Anda sediakan dalam bulan tertentu, tagihan Anda akan diprorata. Untuk mengetahui informasi selengkapnya, lihat Penyimpanan instans DB Amazon RDS.
- .
- IOPS yang tersedia (per IOPS per bulan) – Tingkat IOPS yang tersedia, terlepas dari IOPS yang digunakan, untuk penyimpanan gp3 IOPS yang disediakan Amazon RDS (SSD) dan Tujuan Umum (SSD). Penyimpanan yang tersedia untuk volume EBS ditagih dalam setiap kenaikan 1 detik, dengan minimal 10 menit.
- Penyimpanan cadangan (per GiB per bulan) – Penyimpanan cadangan adalah penyimpanan yang terkait dengan cadangan basis data otomatis dan setiap snapshot basis data aktif yang telah Anda ambil. Meningkatkan periode retensi cadangan atau mengambil snapshot basis data tambahan akan meningkatkan penyimpanan cadangan yang digunakan oleh basis data Anda. Tagihan per detik tidak berlaku untuk penyimpanan cadangan (diukur dalam GB-bulan).

  Untuk mengetahui informasi selengkapnya, lihat Mencadangkan, memulihkan, dan mengekspor data.
- Transfer data (per GB) – Transfer data yang masuk dan keluar dari instans DB Anda dari atau ke internet dan Wilayah AWS lainnya.

Amazon RDS menyediakan opsi pembelian berikut yang memungkinkan Anda mengoptimalkan biaya berdasarkan kebutuhan Anda:

- Instans Sesuai Permintaan – Bayar per jam untuk periode instans DB yang Anda gunakan. Harga dicantumkan per jam, tetapi tagihan dihitung turun menjadi detik dan menunjukkan waktu dalam bentuk desimal. Penggunaan RDS kini ditagih setiap kenaikan 1 detik, dengan minimal 10 menit.

- Instans terpesan – Memesan instans DB untuk jangka waktu satu tahun atau tiga tahun dan mendapatkan diskon yang signifikan dibandingkan dengan harga instans DB sesuai permintaan. Dengan penggunaan Instans Terpesan, Anda dapat meluncurkan, menghapus, memulai, atau menghentikan beberapa instans dalam satu jam dan mendapatkan keuntungan Instans Terpesan untuk semua instans.

Untuk mengetahui informasi tentang harga Amazon RDS, lihat Halaman harga Amazon RDS.

Topik

- Instans DB sesuai Permintaan untuk Amazon RDS
- Instans DB terpesan untuk Amazon RDS

Instans DB sesuai Permintaan untuk Amazon RDS

Penagihan instans DB Amazon RDS sesuai permintaan didasarkan pada kelas instans DB (misalnya, db.t3.smal atau db.m5.large). Untuk informasi tentang harga Amazon RDS, lihat Halaman harga Amazon RDS.

Penagihan dimulai untuk instans DB segera setelah instans DB tersedia. Harga dicantumkan per jam, tetapi tagihan dihitung turun menjadi detik dan menunjukkan waktu dalam bentuk desimal. Penggunaan Amazon RDS ditagih dalam setiap kenaikan satu detik, dengan minimum 10 menit. Dalam hal perubahan konfigurasi yang dapat ditagih, seperti penghitungan skala atau kapasitas penyimpanan, Anda dikenai biaya minimum 10 menit. Penagihan berlanjut hingga instans DB berakhir, yaitu pada saat Anda menghapus instans DB atau jika instans DB gagal.

Jika tidak ingin dikenai biaya lagi untuk instans DB, Anda harus menghentikan atau menghapusnya agar tidak ada tagihan untuk instans DB tambahan per jam. Untuk informasi selengkapnya tentang status instans DB yang ditagih, lihat Melihat status instans DB Amazon RDS.

Instans DB tang dihentikan

Ketika instans DB Anda dihentikan, Anda akan dikenai biaya untuk penyimpanan yang disediakan, termasuk IOPS yang Tersedia. Anda juga dikenai biaya untuk penyimpanan cadangan, termasuk penyimpanan untuk snapshot manual dan cadangan otomatis dalam periode retensi yang Anda tentukan. Anda tidak dikenai biaya untuk jam instans DB.

Instans DB Multi-AZ

Jika Anda menentukan bahwa instans DB harus berupa deployment Multi-AZ, Anda akan ditagih berdasarkan harga Multi-AZ yang diposting di halaman harga Amazon RDS.

Instans DB terpesan untuk Amazon RDS

Dengan menggunakan instans DB terpesan, Anda dapat memesan instans DB untuk jangka waktu satu atau tiga tahun. Instans DB terpesan memberikan diskon yang signifikan dibandingkan harga instans DB sesuai permintaan. Instans DB terpesan bukan merupakan instans fisik, melainkan diskon penagihan yang diterapkan untuk penggunaan instans DB tertentu sesuai permintaan dalam akun Anda. Diskon untuk instans DB terpesan terikat dengan jenis instans dan Wilayah AWS.

Proses umum untuk menggunakan instans DB terpesan adalah: Pertama, dapatkan informasi tentang penawaran instans DB terpesan yang tersedia, kemudian beli penawaran instans DB terpesan, dan terakhir dapatkan informasi tentang instans DB terpesan yang ada.

Ikhtisar instans DB terpesan

Saat membeli instans DB terpesan di Amazon RDS, Anda membeli komitmen untuk mendapatkan tarif diskon, pada jenis instans DB tertentu, selama durasi instans DB terpesan. Untuk menggunakan instans DB terpesan Amazon RDS, Anda membuat instans DB baru seperti yang Anda lakukan untuk instans sesuai permintaan.

Instans DB baru yang Anda buat harus memiliki spesifikasi yang sama dengan instans DB terpesan untuk hal berikut:

- Wilayah AWS
- Mesin DB
- Jenis instans DB
- Ukuran instans DB (RDS untuk Microsoft SQL Server dan Amazon RDS for Oracle License Termasuk)
- Edisi (RDS untuk SQL Server dan RDS untuk Oracle)
- Jenis lisensi (termasuk lisensi atau) bring-your-own-license

Jika spesifikasi instans DB baru cocok dengan instans DB terpesan yang sudah ada untuk akun Anda, Anda akan ditagih dengan tarif diskon yang ditawarkan untuk instans DB terpesan. Jika tidak, instans DB ditagih dengan tarif sesuai permintaan.

Anda dapat memodifikasi instans DB yang Anda gunakan sebagai instans DB terpesan. Jika modifikasi sesuai dengan spesifikasi instans DB terpesan, sebagian atau seluruh diskon masih akan berlaku untuk instans DB yang dimodifikasi. Jika modifikasi berada di luar spesifikasi, seperti

mengubah kelas instans, diskon tidak lagi berlaku. Untuk informasi selengkapnya, lihat Instans DB terpesan berukuran fleksibel.

Topik

- Jenis penawaran
- Instans DB terpesan berukuran fleksibel
- Contoh penagihan instans DB terpesan
- Instans DB terpesan untuk klaster DB Multi-AZ
- Menghapus instans DB terpesan

Untuk informasi selengkapnya tentang instans DB terpesan, termasuk harga, lihat instans terpesan Amazon RDS.

Jenis penawaran

Instans DB terpesan tersedia dalam tiga varietas—Tanpa Uang Muka, Uang Muka Sebagian, dan Uang Muka Penuh—yang memungkinkan Anda mengoptimalkan biaya Amazon RDS berdasarkan perkiraan penggunaan Anda.

Tanpa Uang Muka

Opsi ini menyediakan akses ke instans DB terpesan tanpa harus membayar di muka. Instans DB terpesan Tanpa Uang Muka akan menagih tarif per jam yang didiskon untuk setiap jam dalam jangka waktu pemesanan, terlepas dari penggunaannya, dan tidak perlu membayar di muka. Opsi ini hanya tersedia sebagai pemesanan satu tahun.

Uang Muka Sebagian

Opsi ini mengharuskan pembayaran di muka untuk sebagian instans DB terpesan. Sisa jam dalam jangka waktu pemesanan akan ditagih dengan tarif per jam yang didiskon, terlepas dari penggunaannya. Opsi ini adalah pengganti untuk opsi Penggunaan Berat sebelumnya.

Uang Muka Penuh

Pembayaran penuh dilakukan di awal jangka waktu pemesanan, tanpa biaya lain untuk sisa jangka waktu pemesanan, terlepas dari jumlah jam yang digunakan.

Jika Anda menggunakan penagihan gabungan, semua akun dalam organisasi akan dianggap sebagai satu akun. Berarti semua akun dalam organisasi dapat menerima manfaat biaya per jam dari

instans DB terpesan yang dibeli oleh akun lain. Untuk informasi selengkapnya tentang penagihan gabungan, lihat instans DB terpesan Amazon RDS dalam Panduan Pengguna Manajemen Biaya dan Penagihan AWS .

Instans DB terpesan berukuran fleksibel

Saat membeli instans DB terpesan, satu hal yang Anda tentukan adalah kelas instans, misalnya, db.r5.large. Untuk informasi selengkapnya tentang kelas instans DB, lihat Kelas instans DB .

Jika Anda memiliki instans DB, dan perlu menskalakan kapasitasnya ke ukuran yang lebih besar, instans DB terpesan Anda secara otomatis akan diterapkan pada instans DB yang diskalakan. Artinya, instans DB terpesan Anda secara otomatis diterapkan ke semua ukuran kelas instans DB. Instans DB cadangan yang fleksibel dengan ukuran tersedia untuk instans DB dengan mesin database yang sama. Wilayah AWS Instans DB terpesan berukuran fleksibel hanya dapat diskalakan di jenis kelas instans-nya. Misalnya, instans DB terpesan untuk db.r5.large dapat diterapkan ke db.r5.xlarge, tetapi tidak ke db.r6g.large, karena db.r5 dan db.r6g adalah jenis kelas instans yang berbeda.

Manfaat instans DB terpesan juga berlaku untuk konfigurasi Multi-AZ dan AZ-Tunggal. Fleksibilitas berarti bahwa Anda dapat berpindah dengan bebas di antara konfigurasi dengan jenis kelas instans DB yang sama. Misalnya, Anda dapat berpindah dari penerapan AZ tunggal yang berjalan pada satu instans DB besar (empat unit dinormalisasi per jam) ke penerapan multi-AZ yang berjalan pada dua instans DB sedang (2+2 = 4 unit dinormalisasi per jam).

Instans DB terpesan berukuran fleksibel tersedia untuk mesin basis data Amazon RDS berikut:

- RDS for MariaDB
- RDS for MySQL
- RDS untuk Oracle, Bawa Lisensi Anda Sendiri
- RDS for PostgreSQL

Fleksibilitas ukuran tidak berlaku untuk RDS untuk SQL Server dan RDS untuk Lisensi Oracle Termasuk.

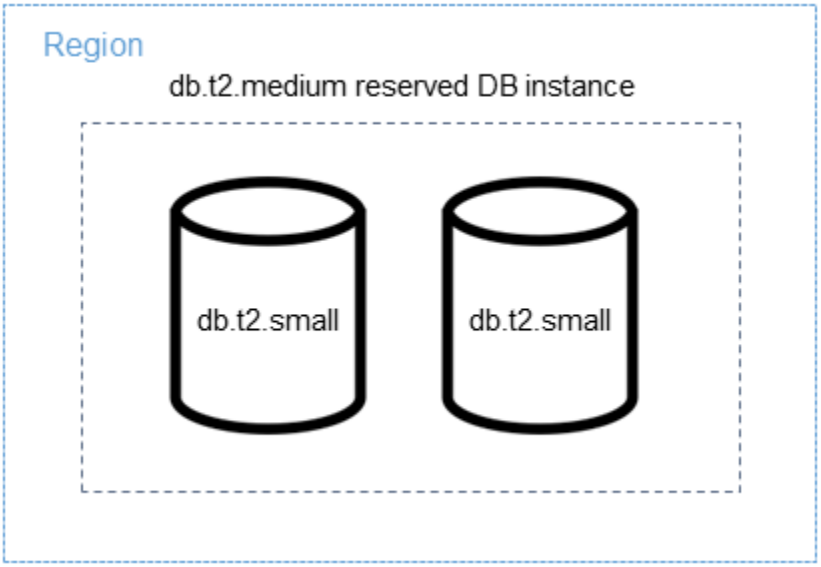
Untuk detail tentang penggunaan instans terpesan berukuran fleksibel dengan Aurora, lihat Instans DB terpesan untuk Aurora.

Anda dapat membandingkan penggunaan untuk ukuran instans DB terpesan yang berbeda dengan menggunakan unit per jam yang dinormalkan. Misalnya, satu unit penggunaan di dua instans DB

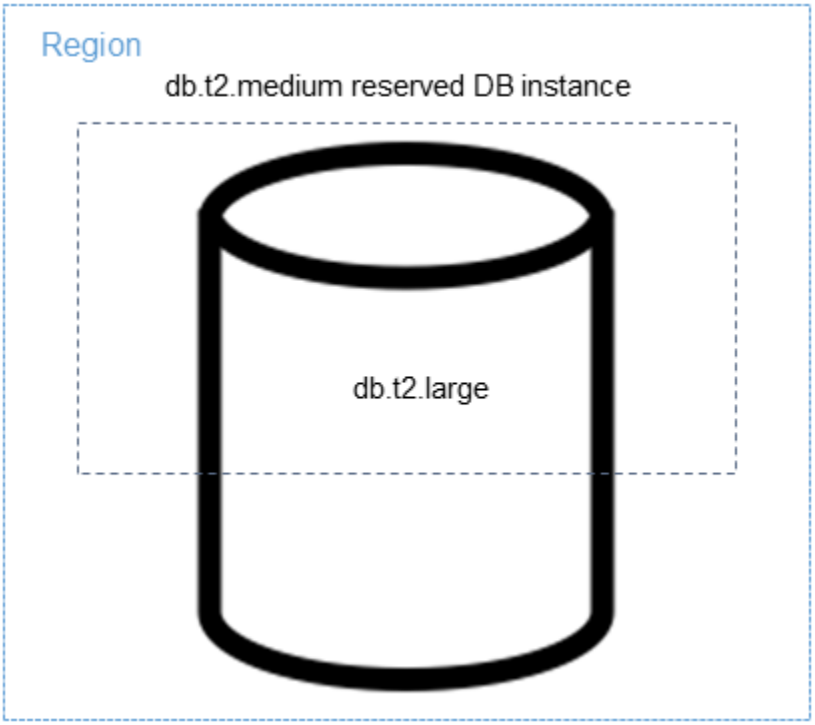
db.r3.large setara dengan delapan unit per jam penggunaan yang dinormalkan di satu db.r3.small. Tabel berikut menunjukkan jumlah unit per jam yang dinormalkan untuk setiap ukuran instans DB.

Ukuran instans	Unit per jam AZ-Tunggal yang dinormalkan (deployment dengan satu instans DB)	Unit per jam yang dinormalkan untuk instans DB Multi-AZ (deployment dengan satu instans DB dan satu fungsi siaga)	Unit per jam yang dinormalkan untuk klaster DB Multi-AZ (deployment dengan satu instans DB dan dua fungsi siaga)
mikro	0,5	1	1.5
kecil	1	2	3
sedang	2	4	6
besar	4	8	12
xlarge	8	16	24
2xlarge	16	32	48
4xlarge	32	64	96
6xlarge	48	96	144
8xlarge	64	128	192
10xlarge	80	160	240
12xlarge	96	192	288
16xlarge	128	256	384
24xlarge	192	384	576
32xlarge	256	512	768

Misalnya, Anda membeli instans DB terpesan `db.t2.medium`, dan Anda memiliki dua instans DB `db.t2.small` yang berjalan di akun Anda pada Wilayah AWS yang sama. Dalam hal ini, manfaat penagihan diterapkan sepenuhnya pada kedua instans.



Atau, jika Anda memiliki satu `db.t2.large` instans yang berjalan di akun Anda dalam hal yang sama Wilayah AWS, manfaat penagihan diterapkan ke 50 persen dari penggunaan instans DB.



Contoh penagihan instans DB terpesan

Harga untuk instans DB terpesan tidak memberikan diskon untuk biaya yang terkait dengan penyimpanan, cadangan, dan I/O. Diskon hanya diberikan pada penggunaan instans sesuai permintaan per jam. Contoh berikut menggambarkan total biaya per bulan untuk instans DB terpesan:

- Kelas instans DB db.r5.large AZ-Tunggal terpesan pada RDS for MySQL di AS Timur (Virginia Utara) dengan opsi Tanpa Uang Muka dengan biaya sebesar $0,12 untuk instans tersebut, atau $90 per bulan
- 400 GiB untuk penyimpanan SSD Tujuan Umum (gp2) dengan biaya $0,115 per GiB per bulan, atau $45,60 per bulan
- 600 GiB untuk penyimpanan cadangan dengan biaya $0,095, atau $19 per bulan (gratis 400 GiB)

Tambahkan semua biaya ini ($90 + $45,60 + $19) dengan instans DB terpesan, dan total biaya per bulannya adalah $154,60.

Jika Anda memilih untuk menggunakan instans DB sesuai permintaan dan bukannya instans DB terpesan, biaya kelas instans DB db.r5.large Tunggal-AZ pada RDS for MySQL di AS Timur (Virginia Utara) adalah sebesar $0,1386 per jam, atau $101,18 per bulan. Jadi, untuk instans DB sesuai permintaan, tambahkan semua opsi ini ($101,18 +$45,60 + $19), dan total biaya per bulannya adalah $165,78. Anda menghemat sekitar $11 per bulan dengan menggunakan instans DB terpesan.

> **Note**
>
> Harga dalam contoh ini adalah harga contoh dan mungkin tidak sesuai dengan harga aktual. Untuk informasi harga Amazon RDS, lihat Harga Amazon RDS.

Instans DB terpesan untuk klaster DB Multi-AZ

Untuk membeli instans DB terpesan yang setara untuk klaster DB Multi-AZ, Anda dapat melakukan salah satu hal berikut:

- Pesan tiga instans DB AZ-Tunggal yang ukurannya sama seperti instans dalam klaster.
- Pesan satu instans DB Multi-AZ dan satu instans DB AZ-Tunggal yang ukurannya sama dengan instans DB dalam klaster.

Misalnya, Anda memiliki satu klaster yang terdiri dari tiga instans DB db.m6gd.large. Dalam hal ini, Anda dapat membeli tiga instans DB terpesan AZ-Tunggal db.m6gd.large, atau satu instans DB terpesan Multi-AZ db.m6gd.large dan satu instans DB terpesan AZ-Tunggal db.m6gd.large. Salah satu dari opsi ini akan memesan diskon instans terpesan maksimum untuk klaster DB Multi-AZ.

Sebagai alternatif, Anda dapat menggunakan instans DB berukuran fleksibel dan membeli instans DB yang lebih besar untuk mencakup instans DB yang lebih kecil dalam satu atau beberapa klaster. Misalnya, jika Anda memiliki dua klaster dengan total enam instans DB db.m6gd.large, Anda dapat membeli tiga instans DB terpesan AZ-Tunggal db.m6gd.xl. Cara ini akan memesan enam instans DB dalam dua klaster. Untuk informasi selengkapnya, lihat Instans DB terpesan berukuran fleksibel.

Anda dapat memesan instans DB yang ukurannya sama dengan instans DB dalam klaster, tetapi memesan lebih sedikit instans DB daripada jumlah total instans DB dalam klaster. Namun, jika Anda melakukannya, klaster hanya dipesan sebagian. Misalnya, Anda memiliki satu klaster dengan tiga instans DB db.m6gd.large, dan Anda membeli satu instans DB terpesan Multi-AZ db.m6gd.large. Dalam hal ini, klaster hanya dipesan sebagian, karena hanya dua dari tiga instans dalam klaster yang dicakup oleh instans DB terpesan. Instans DB yang tersisa akan dikenai biaya dengan tarif per jam db.m6gd.large sesuai permintaan.

Untuk informasi selengkapnya tentang klaster DB Multi-AZ, lihat Deployment klaster basis data Multi-AZ.

Menghapus instans DB terpesan

Syarat untuk instans DB terpesan melibatkan komitmen satu tahun atau tiga tahun. Anda tidak dapat membatalkan instans DB terpesan. Namun, Anda dapat menghapus instans DB yang dicakup oleh diskon instans DB terpesan. Proses penghapusan instans DB yang dicakup oleh diskon instans DB terpesan sama dengan instans DB lainnya.

Anda ditagih untuk biaya di muka, terlepas dari apakah Anda menggunakan sumber daya atau tidak.

Jika Anda menghapus instans DB yang dicakup oleh diskon instans DB terpesan, Anda dapat meluncurkan instans DB lain dengan spesifikasi yang kompatibel. Dalam hal ini, Anda tetap mendapatkan tarif diskon selama jangka waktu reservasi (satu atau tiga tahun).

Bekerja dengan instans DB terpesan

Anda dapat menggunakan API AWS Management Console, the AWS CLI, dan RDS untuk bekerja dengan instans DB yang dicadangkan.

Konsol

Anda dapat menggunakan AWS Management Console untuk bekerja dengan instans DB cadangan seperti yang ditunjukkan dalam prosedur berikut.

Untuk mendapatkan harga dan informasi tentang penawaran instans DB yang tersedia

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Instans terpesan.
3. Pilih Beli Instans DB Terpesan.
4. Untuk Deskripsi produk, pilih mesin DB dan jenis lisensi.
5. Untuk Kelas instans DB, pilih kelas instans DB.
6. Untuk Opsi Deployment, pilih apakah Anda menginginkan deployment instans DB AZ-Tunggal atau Multi-AZ.

   > **Note**
   >
   > Untuk membeli instans DB terpesan yang setara untuk deployment klaster DB Multi-AZ, beli tiga instans DB terpesan AZ-Tunggal, atau satu instans DB terpesan Multi-AZ dan satu instans DB terpesan AZ-Tunggal. Untuk informasi selengkapnya, lihat Instans DB terpesan untuk klaster DB Multi-AZ.

7. Untuk Masa berlaku, pilih jangka waktu untuk memesan instans DB.
8. Untuk Jenis penawaran, pilih jenis penawaran.

   Setelah memilih jenis penawaran, Anda dapat melihat informasi harga.

   > **Important**
   >
   > Pilih Batalkan untuk menghindari pembelian instans DB terpesan dan dikenakan biaya.

Setelah Anda memiliki informasi tentang penawaran instans DB terpesan yang tersedia, Anda dapat menggunakan informasi tersebut untuk membeli penawaran sebagaimana ditunjukkan dalam prosedur berikut.

Untuk membeli instans DB terpesan

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Instans terpesan.
3. Pilih Beli instans DB terpesan.
4. Untuk Deskripsi produk, pilih mesin DB dan jenis lisensi.
5. Untuk Kelas instans DB, pilih kelas instans DB.
6. Untuk Deployment Multi-AZ, pilih apakah Anda menginginkan deployment instans DB AZ-Tunggal atau Multi-AZ.

   > **Note**
   >
   > Untuk membeli instans DB terpesan yang setara untuk deployment klaster DB Multi-AZ, beli tiga instans DB terpesan AZ-Tunggal, atau satu instans DB terpesan Multi-AZ dan satu instans DB terpesan AZ-Tunggal. Untuk informasi selengkapnya, lihat Instans DB terpesan untuk klaster DB Multi-AZ.

7. Untuk Masa berlaku, pilih jangka waktu yang Anda inginkan untuk pemesanan instans DB.
8. Untuk Jenis penawaran, pilih jenis penawaran.

   Setelah memilih jenis penawaran, Anda dapat melihat informasi harga.
9. (Opsional) Anda dapat menetapkan pengidentifikasi Anda sendiri ke instans DB terpesan yang Anda beli untuk membantu Anda melacaknya. Untuk ID Terpesan, ketik pengidentifikasi untuk instans DB terpesan Anda.
10. Pilih Kirim.

    Instans DB terpesan Anda dibeli, lalu ditampilkan dalam daftar Instans terpesan.

Setelah membeli instans DB terpesan, Anda dapat memperoleh informasi tentang instans DB terpesan Anda seperti yang ditunjukkan dalam prosedur berikut.

Untuk mendapatkan informasi tentang instans DB cadangan untuk akun Anda AWS

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.

2. Di panel Navigasi, pilih Instans terpesan.

   Instans DB terpesan untuk akun Anda akan muncul. Untuk melihat informasi terperinci tentang instans DB terpesan tertentu, pilih instans tersebut dalam daftar. Kemudian, Anda dapat melihat informasi terperinci tentang instans tersebut dalam panel detail di bagian bawah konsol.

AWS CLI

Anda dapat menggunakan AWS CLI untuk bekerja dengan instans DB cadangan seperti yang ditunjukkan dalam contoh berikut.

Example mendapatkan penawaran instans DB terpesan yang tersedia

Untuk mendapatkan informasi tentang penawaran instans DB cadangan yang tersedia, hubungi perintah. AWS CLI `describe-reserved-db-instances-offerings`

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

Panggilan ini menghasilkan output serupa dengan berikut ini:

```
OFFERING  OfferingId                            Class         Multi-AZ  Duration  Fixed
 Price  Usage Price  Description  Offering Type
OFFERING  438012d3-4052-4cc7-b2e3-8d3372e0e706  db.r3.large   y         1y
 1820.00 USD  0.368 USD    mysql        Partial  Upfront
OFFERING  649fd0c8-cf6d-47a0-bfa6-060f8e75e95f  db.r3.small   n         1y
 227.50 USD  0.046 USD    mysql        Partial  Upfront
OFFERING  123456cd-ab1c-47a0-bfa6-12345667232f  db.r3.small   n         1y
 162.00 USD   0.00 USD    mysql        All      Upfront
    Recurring Charges:   Amount  Currency  Frequency
    Recurring Charges:   0.123   USD       Hourly
OFFERING  123456cd-ab1c-37a0-bfa6-12345667232d  db.r3.large   y         1y
 700.00 USD   0.00 USD    mysql        All      Upfront
    Recurring Charges:   Amount  Currency  Frequency
    Recurring Charges:   1.25    USD       Hourly
OFFERING  123456cd-ab1c-17d0-bfa6-12345667234e  db.r3.xlarge  n         1y
 4242.00 USD   2.42 USD    mysql        No       Upfront
```

Setelah memiliki informasi tentang penawaran instans DB terpesan yang tersedia, Anda dapat menggunakan informasi tersebut untuk membeli penawaran.

Untuk membeli instans DB cadangan, gunakan AWS CLI perintah [purchase-reserved-db-instances-offering](#)dengan parameter berikut:

- `--reserved-db-instances-offering-id` – ID penawaran yang ingin Anda beli. Lihat contoh sebelumnya untuk mendapatkan ID penawaran.
- `--reserved-db-instance-id` – Anda dapat menetapkan pengindentifikasi Anda sendiri ke instans DB terpesan yang Anda beli untuk membantu melacaknya.

Example membeli instans DB terpesan

Contoh berikut membeli penawaran instans DB cadangan dengan ID *649fd0c8-cf6d-47a0-bfa6-060f8e75e95f*, dan menetapkan pengenal. *MyReservation*

Untuk Linux, macOS, atau Unix:

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

Untuk Windows:

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

Perintah tersebut mengembalikan output serupa dengan berikut ini:

```
RESERVATION  ReservationId      Class        Multi-AZ  Start Time
 Duration  Fixed Price  Usage Price  Count  State            Description  Offering Type
RESERVATION  MyReservation      db.r3.small  y         2011-12-19T00:30:23.247Z  1y
    455.00 USD   0.092 USD    1      payment-pending  mysql        Partial  Upfront
```

Setelah membeli instans DB terpesan, Anda dapat memperoleh informasi tentang instans DB terpesan Anda.

Untuk mendapatkan informasi tentang instans DB cadangan untuk AWS akun Anda, hubungi AWS CLI perintah [describe-reserved-db-instances](#), seperti yang ditunjukkan pada contoh berikut.

Example mendapatkan instans DB terpesan Anda

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

Perintah tersebut mengembalikan output serupa dengan berikut ini:

```
RESERVATION  ReservationId     Class        Multi-AZ  Start Time
 Duration  Fixed Price  Usage Price  Count  State    Description  Offering Type
RESERVATION  MyReservation     db.r3.small  y         2011-12-09T23:37:44.720Z  1y
   455.00 USD   0.092 USD    1      retired  mysql        Partial  Upfront
```

API RDS

Anda dapat menggunakan API RDS untuk bekerja dengan instans DB terpesan:

- Untuk mendapatkan informasi tentang penawaran instans DB terpesan yang tersedia, panggil operasi API Amazon RDS `DescribeReservedDBInstancesOfferings`.
- Setelah memiliki informasi tentang penawaran instans DB terpesan yang tersedia, Anda dapat menggunakan informasi tersebut untuk membeli penawaran. Panggil operasi API RDS `PurchaseReservedDBInstancesOffering` dengan parameter berikut:
  - `--reserved-db-instances-offering-id` – ID penawaran yang ingin Anda beli.
  - `--reserved-db-instance-id` – Anda dapat menetapkan pengindentifikasi Anda sendiri ke instans DB terpesan yang Anda beli untuk membantu melacaknya.
- Setelah membeli instans DB terpesan, Anda dapat memperoleh informasi tentang instans DB terpesan Anda. Panggil operasi API RDS `DescribeReservedDBInstances`.

Melihat penagihan untuk instans DB terpesan Anda

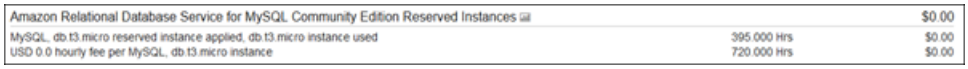
Anda dapat melihat penagihan untuk instans DB terpesan Anda di Dasbor Penagihan di AWS Management Console.

Untuk melihat penagihan instans DB terpesan

1. Masuk ke AWS Management Console.
2. Dari menu akun di kanan atas, pilih Dasbor Penagihan.
3. Pilih Detail Tagihan di kanan atas dasbor.
4. Di bagian Biaya Layanan AWS , perluas Layanan Basis Data Relasional.

5. Perluas Wilayah AWS lokasi instans DB cadangan Anda, misalnya US West (Oregon).

   Instans DB terpesan Anda dan biaya per jamnya untuk bulan berjalan ditampilkan di bagian Layanan Basis Data Relasional Amazon untuk Instans Terpesan ***Mesin Basis Data***.

   Instans DB terpesan dalam contoh ini dibeli dengan opsi Uang Muka Penuh, jadi tidak ada biaya per jam.

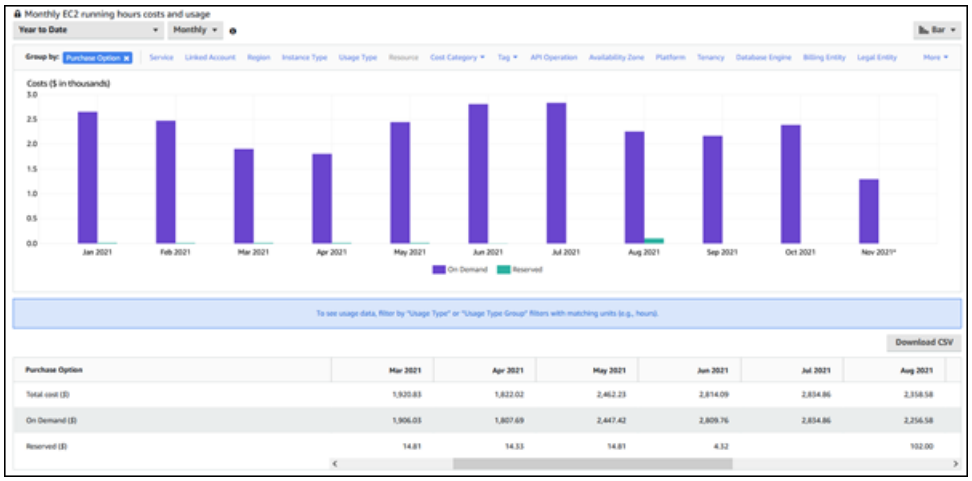
6. Pilih ikon Penjelajah Biaya (grafik batang) di samping judul Instans Terpesan.

   Penjelajah Biaya menampilkan grafik Penggunaan dan biaya waktu pengoperasian EC2 bulanan.

7. Kosongkan filter Grup Jenis Penggunaan di sebelah kanan grafik.

8. Pilih periode waktu dan unit waktu yang Anda inginkan untuk memeriksa biaya penggunaan.

   Contoh berikut menunjukkan biaya penggunaan untuk instans DB sesuai permintaan dan terpesan untuk tahun berjalan menurut bulan.

   Biaya instans DB terpesan dari Januari hingga Juni 2021 adalah biaya bulanan untuk instans Uang Muka Sebagian, sedangkan biaya pada Agustus 2021 adalah biaya satu kali untuk instans Uang Muka Penuh.

   Diskon instans terpesan untuk instans Uang Muka Sebagian habis masa berlakunya pada Juni 2021, tetapi instans DB tidak dihapus. Setelah habis masa berlakunya, biaya dibebankan dengan tarif sesuai permintaan.

Menyiapkan Amazon RDS

Sebelum Anda menggunakan Amazon Relational Database Service untuk pertama kali, selesaikan tugas berikut:

Topik

Jika sudah memiliki Akun AWS, mengetahui persyaratan Amazon RDS, dan lebih memilih menggunakan defaults untuk grup keamanan IAM dan VPC, Anda dapat langsung beralih ke Mulai menggunakan Amazon RDS.

Mendaftar Akun AWS

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar Akun AWS

1. Buka https://portal.aws.amazon.com/billing/signup.
2. Ikuti petunjuk secara online.

   Anda akan diminta untuk menerima panggilan telepon dan memasukkan kode verifikasi pada keypad telepon sebagai bagian dari prosedur pendaftaran.

   Saat Anda mendaftar Akun AWS, Pengguna root akun AWS akan dibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya dalam akun. Sebagai praktik terbaik keamanan, tetapkan akses administratif ke pengguna administratif, dan hanya gunakan pengguna root untuk melakukan tugas yang memerlukan akses pengguna root.

AWS akan mengirimkan email konfirmasi kepada Anda setelah proses pendaftaran selesai. Anda dapat melihat aktivitas akun saat ini dan mengelola akun dengan mengunjungi https://aws.amazon.com/ dan memilih Akun Saya.

Membuat pengguna administratif

Setelah mendaftar Akun AWS, amankan Pengguna root akun AWS, aktifkan AWS IAM Identity Center, dan buat sebuah pengguna administratif sehingga Anda tidak menggunakan pengguna root untuk tugas sehari-hari.

Mengamankan Pengguna root akun AWS Anda

1. Masuk ke AWS Management Console sebagai pemilik akun dengan memilih Pengguna root dan memasukkan alamat email Akun AWS Anda. Di halaman berikutnya, masukkan kata sandi Anda.

   Untuk bantuan masuk menggunakan pengguna root, lihat Masuk sebagai pengguna root dalam Panduan Pengguna AWS Sign-In.
2. Aktifkan autentikasi multi-faktor (MFA) untuk pengguna root Anda.

   Untuk petunjuknya, silakan lihat Mengaktifkan perangkat MFA virtual untuk pengguna root Akun AWS Anda (konsol) dalam Panduan Pengguna IAM.

Membuat pengguna administratif

1. Aktifkan Pusat Identitas IAM.

   Untuk mendapatkan petunjuk, silakan lihat Mengaktifkan AWS IAM Identity Center di Panduan Pengguna AWS IAM Identity Center.
2. Di Pusat Identitas IAM, berikan akses administratif ke sebuah pengguna administratif.

   Untuk mendapatkan tutorial tentang menggunakan Direktori Pusat Identitas IAM sebagai sumber identitas Anda, silakan lihat Mengonfigurasi akses pengguna dengan Direktori Pusat Identitas IAM default di Panduan Pengguna AWS IAM Identity Center.

Masuk sebagai pengguna administratif

- Untuk masuk dengan pengguna Pusat Identitas IAM, gunakan URL masuk yang dikirim ke alamat email Anda saat Anda membuat pengguna Pusat Identitas IAM.

  Untuk bantuan masuk menggunakan pengguna Pusat Identitas IAM, lihat Masuk ke portal akses AWS dalam Panduan Pengguna AWS Sign-In.

Memberikan akses terprogram

Pengguna membutuhkan akses terprogram jika mereka ingin berinteraksi dengan AWS luar dari AWS Management Console. Cara memberikan akses terprogram bergantung pada jenis pengguna yang mengakses AWS.

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

Pengguna mana yang membutuhkan akses terprogram?	Untuk	Oleh
Identitas tenaga kerja (Pengguna yang dikelola di Pusat Identitas IAM)	Gunakan kredensial sementara untuk menandatangani permintaan terprogram ke AWS CLI, SDK AWS, atau API AWS.	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan. • Untuk AWS CLI, lihat Mengonfigurasi AWS CLI untuk menggunakan AWS IAM Identity Center di Panduan Pengguna AWS Command Line Interface. • Untuk SDK AWS, alat, dan API AWS, lihat Autentikasi Pusat Identitas IAM di Panduan Referensi SDK dan Alat AWS.
IAM	Gunakan kredensial sementara untuk menandatangani permintaan terprogram ke AWS CLI, SDK AWS, atau API AWS.	Mengikuti petunjuk dalam Menggunakan kredensial sementara dengan sumber daya AWS di Panduan Pengguna IAM.
IAM	(Tidak direkomendasikan) Gunakan kredensial jangka panjang untuk menandatangani permintaan terprogram	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan.

Pengguna mana yang membutuhkan akses terprogram?	Untuk	Oleh
	ke AWS CLI, SDK AWS, atau API AWS.	• Untuk AWS CLI, lihat Mengautentikasi menggunakan kredensial pengguna IAM di Panduan Pengguna AWS Command Line Interface. • Untuk SDK dan alat AWS, lihat Mengautentikasi menggunakan kredensial jangka panjang di Panduan Referensi SDK dan Alat AWS. • Untuk API AWS, lihat Mengelola kunci akses untuk pengguna IAM di Panduan Pengguna IAM.

Menentukan persyaratan

Blok bangunan dasar Amazon RDS adalah instans DB. Di instans DB, Anda akan membuat basis data Anda. Instans DB menyediakan alamat jaringan yang disebut titik akhir. Aplikasi Anda menggunakan titik akhir ini untuk terhubung ke instans DB Anda. Saat membuat instans DB, Anda menentukan detail seperti penyimpanan, memori, mesin dan versi basis data, konfigurasi, jaringan, dan periode pemeliharaan. Anda mengontrol akses jaringan ke instans DB melalui grup keamanan.

Sebelum membuat grup keamanan dan instans DB, Anda harus mengetahui instans DB dan kebutuhan jaringan Anda. Berikut beberapa hal penting yang perlu dipertimbangkan:

- Kebutuhan sumber daya – Apa saja kebutuhan memori dan prosesor untuk aplikasi atau layanan Anda? Anda menggunakan pengaturan ini untuk membantu Anda menentukan kelas instans DB apa yang akan digunakan. Untuk spesifikasi tentang kelas instans DB, lihat Kelas instans DB .

- VPC, subnet, dan grup keamanan – Instans DB Anda kemungkinan besar akan berada dalam cloud privat virtual (VPC). Untuk menghubungkan ke instans DB Anda, Anda perlu menyiapkan aturan grup keamanan. Aturan ini diatur secara berbeda, tergantung pada jenis VPC yang Anda gunakan dan cara Anda menggunakannya. Misalnya, Anda dapat menggunakan: VPC default atau VPC yang ditentukan pengguna.

  Daftar berikut menjelaskan aturan untuk setiap opsi VPC:

  - VPC Default – Jika akun AWS Anda memiliki VPC default di Wilayah AWS saat ini, VPC tersebut dikonfigurasi untuk mendukung instans DB. Jika Anda menentukan VPC default saat membuat instans DB, lakukan tindakan berikut:
    - Anda harus membuat grup keamanan VPC yang mengizinkan koneksi dari aplikasi atau layanan ke instans DB Amazon RDS. Gunakan opsi Grup Keamanan pada konsol VPC atau AWS CLI untuk membuat grup keamanan VPC. Untuk informasi, lihat Langkah 3: Buat grup keamanan VPC.
    - Tentukan grup subnet DB default. Jika ini adalah instans DB pertama yang Anda buat di Wilayah AWS ini, Amazon RDS membuat grup subnet DB default saat membuat instans DB.
  - VPC yang ditentukan pengguna – Jika Anda ingin menentukan VPC yang ditentukan pengguna ketika Anda membuat instans DB, perhatikan hal berikut:
    - Anda harus membuat grup keamanan VPC yang mengizinkan koneksi dari aplikasi atau layanan ke instans DB Amazon RDS. Gunakan opsi Grup Keamanan pada konsol VPC atau AWS CLI untuk membuat grup keamanan VPC. Untuk informasi, lihat Langkah 3: Buat grup keamanan VPC.
    - VPC harus memenuhi persyaratan tertentu untuk meng-hosting instans DB, seperti memiliki setidaknya dua subnet, yang masing-masing berada di Zona Ketersediaan terpisah. Untuk informasi, lihat Amazon VPC dan Amazon RDS.
    - Pastikan untuk menentukan grup subnet DB yang menentukan subnet di VPC yang dapat digunakan oleh instans DB. Untuk informasi, lihat bagian grup subnet DB di Bekerja dengan klaster DB dalam VPC.
- Ketersediaan tinggi: – Apakah Anda memerlukan dukungan failover? Di Amazon RDS, deployment Multi-AZ membuat instans DB primer dan instans DB siaga sekunder di Zona Ketersediaan lain untuk dukungan failover. Kami merekomendasikan deployment Multi-AZ untuk beban kerja produksi agar menjaga ketersediaan yang tinggi. Untuk tujuan pengembangan dan pengujian, Anda dapat menggunakan deployment yang bukan Multi-AZ. Untuk informasi selengkapnya, lihat Mengonfigurasi dan mengelola deployment Multi-AZ.

- Kebijakan IAM: – Apakah akun AWS Anda memiliki kebijakan yang memberikan izin yang diperlukan untuk melakukan operasi Amazon RDS? Jika Anda terhubung ke AWS menggunakan kredensial IAM, akun IAM Anda harus memiliki kebijakan IAM yang memberikan izin yang diperlukan untuk menjalankan operasi Amazon RDS. Untuk informasi selengkapnya, lihat Manajemen identitas dan akses untuk Amazon RDS.
- Port terbuka: – Port TCP/IP apa yang akan diterima oleh basis data Anda? Firewall di beberapa perusahaan mungkin memblokir koneksi ke port default untuk mesin basis data Anda. Jika firewall perusahaan Anda memblokir port default, pilih port lain untuk instans DB baru. Saat Anda membuat instans DB yang menerima port yang Anda tentukan, Anda dapat mengubah port tersebut dengan mengubah instans DB.
- Wilayah AWS – Wilayah AWS mana yang akan Anda pilih untuk basis data Anda? Memiliki basis data yang dekat dengan aplikasi atau layanan web dapat mengurangi latensi jaringan. Untuk informasi selengkapnya, lihat Wilayah, Zona Ketersediaan, dan Zona Lokal.
- Subsistem disk DB – Apa penyimpanan yang Anda butuhkan? Amazon RDS menyediakan tiga jenis penyimpanan:
  - Tujuan Umum (SSD)
  - IOPS yang Tersedia (PIOPS)
  - Magnetik (juga dikenal sebagai penyimpanan standar)

  Untuk informasi selengkapnya tentang penyimpanan Amazon RDS, lihat Penyimpanan instans DB Amazon RDS.

Setelah memiliki informasi yang Anda perlukan untuk membuat grup keamanan dan instans DB, lanjutkan ke langkah berikutnya.

Memberikan akses ke instans DB di VPC Anda dengan membuat grup keamanan

Grup keamanan VPC menyediakan akses ke instans DB di VPC. Grup keamanan tersebut bertindak sebagai firewall untuk instans DB yang terkait, yang mengontrol lalu lintas masuk dan keluar di tingkat instans DB. Instans DB dibuat secara default dengan firewall dan grup keamanan default yang melindungi instans DB.

Sebelum dapat terhubung ke instans DB, Anda harus menambahkan aturan ke grup keamanan yang memungkinkan Anda untuk terhubung. Gunakan informasi jaringan dan konfigurasi untuk membuat aturan yang akan mengizinkan akses ke instans DB Anda.

Misalnya, katakanlah Anda memiliki aplikasi yang mengakses basis data di instans DB Anda di VPC. Dalam hal ini, Anda harus menambahkan aturan TCP khusus yang menentukan rentang port dan alamat IP yang digunakan aplikasi Anda untuk mengakses basis data. Jika memiliki aplikasi di instans Amazon EC2, Anda dapat menggunakan grup keamanan yang Anda siapkan untuk instans Amazon EC2.

Anda dapat mengonfigurasi konektivitas antara instans Amazon EC2 dab instans DB saat membuat instans DB. Untuk informasi selengkapnya, lihat Konfigurasikan konektivitas jaringan otomatis dengan instans EC2.

> **Tip**
>
> Anda dapat menyiapkan konektivitas jaringan antara instans Amazon EC2 dan instans DB secara otomatis saat membuat instans DB. Untuk informasi selengkapnya, lihat Konfigurasikan konektivitas jaringan otomatis dengan instans EC2.

Untuk informasi tentang skenario umum untuk mengakses instans DB, lihat Skenario untuk mengakses instans DB di VPC.

Untuk membuat grup keamanan VPC

1. Masuk ke AWS Management Console dan buka konsol Amazon VPC di https://console.aws.amazon.com/vpc.

   > **Note**
   >
   > Pastikan Anda berada di konsol VPC, bukan konsol RDS.

2. Di sudut kanan atas AWS Management Console, pilih Wilayah AWS tempat Anda akan membuat grup keamanan VPC dan instans DB. Dalam daftar sumber daya Amazon VPC untuk Wilayah AWS tersebut, Anda akan melihat setidaknya satu VPC dan beberapa subnet. Jika tidak, Anda tidak memiliki VPC default di Wilayah AWS tersebut.
3. Di panel navigasi, pilih Security Groups (Grup Keamanan).
4. Pilih Buat grup keamanan.

Halaman Membuat grup keamanan akan muncul.

5. Dalam Detail dasar, masukkan Nama grup keamanan dan Deskripsi. Untuk VPC, pilih VPC tempat Anda akan membuat instans DB.
6. Di bagian Aturan masuk, pilih Tambahkan aturan.

   a. Untuk Jenis, pilih TCP khusus.

   b. Untuk Rentang port, masukkan nilai port yang akan digunakan untuk klaster DB Anda.

   c. Untuk Sumber, pilih nama grup keamanan atau ketik rentang alamat IP (nilai CIDR) dari tempat Anda mengakses instans DB. Jika Anda memilih IP Saya, pilihan ini akan mengizinkan akses ke instans DB dari alamat IP yang terdeteksi di browser Anda.
7. Jika Anda perlu menambahkan lebih banyak alamat IP atau rentang port yang berbeda, pilih Tambahkan aturan dan masukkan informasi untuk aturan tersebut.
8. (Opsional) Dalam Aturan keluar, tambahkan aturan untuk lalu lintas keluar. Secara default, semua lalu lintas keluar akan diizinkan.
9. Pilih Buat grup keamanan.

Anda dapat menggunakan grup keamanan VPC yang baru saja dibuat sebagai grup keamanan untuk instans DB Anda saat Anda membuatnya.

> **Note**
>
> Jika Anda menggunakan VPC default, grup subnet default yang mencakup semua subnet VPC akan dibuat untuk Anda. Saat membuat instans DB, Anda dapat memilih VPC default dan menggunakan default untuk Grup Subnet DB.

Setelah menyelesaikan persyaratan penyiapan, Anda dapat membuat instans DB menggunakan persyaratan dan grup keamanan Anda. Untuk melakukannya, ikuti petunjuk di Membuat instans DB Amazon RDS. Untuk informasi tentang memulai dengan membuat instans DB yang menggunakan mesin DB tertentu, lihat dokumentasi yang relevan pada tabel berikut.

Mesin basis data	Dokumentasi
MariaDB	Membuat dan menghubungkan ke instans DB MariaDB

Mesin basis data	Dokumentasi
Microsoft SQL Server	Membuat dan menghubungkan ke instans DB Microsoft SQL Server
MySQL	Membuat dan menghubungkan ke instans DB MySQL
Oracle	Membuat dan menghubungkan ke instans DB Oracle
PostgreSQL	Membuat dan menghubungkan ke instans DB PostgreSQL

> **Note**
>
> Jika Anda tidak dapat terhubung ke instans DB setelah membuatnya, lihat informasi pemecahan masalah di Tidak dapat terhubung ke instans DB Amazon RDS.

Mulai menggunakan Amazon RDS

Dalam contoh berikut, Anda dapat menemukan cara membuat dan menghubungkan ke instans DB menggunakan Amazon Relational Database Service (Amazon RDS). Anda dapat membuat instans DB yang menggunakan Db2, MariaDB, MySQL, Microsoft SQL Server, Oracle, atau PostgreSQL.

> **Important**
>
> Sebelum dapat membuat atau terhubung ke instans DB, pastikan untuk menyelesaikan tugas dalam Menyiapkan Amazon RDS.

Pembuatan instans DB dan koneksi ke basis data pada instans DB sedikit berbeda untuk setiap mesin DB. Pilih salah satu mesin DB berikut yang ingin Anda gunakan untuk informasi mendetail tentang pembuatan dan koneksi ke instans DB. Setelah membuat dan terhubung ke instans DB, terdapat petunjuk yang dapat membantu Anda menghapus instans DB.

Topik

- Membuat dan menghubungkan ke instans DB MariaDB
- Membuat dan menghubungkan ke instans DB Microsoft SQL Server
- Membuat dan menghubungkan ke instans DB MySQL
- Membuat dan menghubungkan ke instans DB Oracle
- Membuat dan menghubungkan ke instans DB PostgreSQL
- Tutorial: Membuat server web dan instans DB Amazon RDS
- Tutorial: Menggunakan fungsi Lambda untuk mengakses basis data Amazon RDS

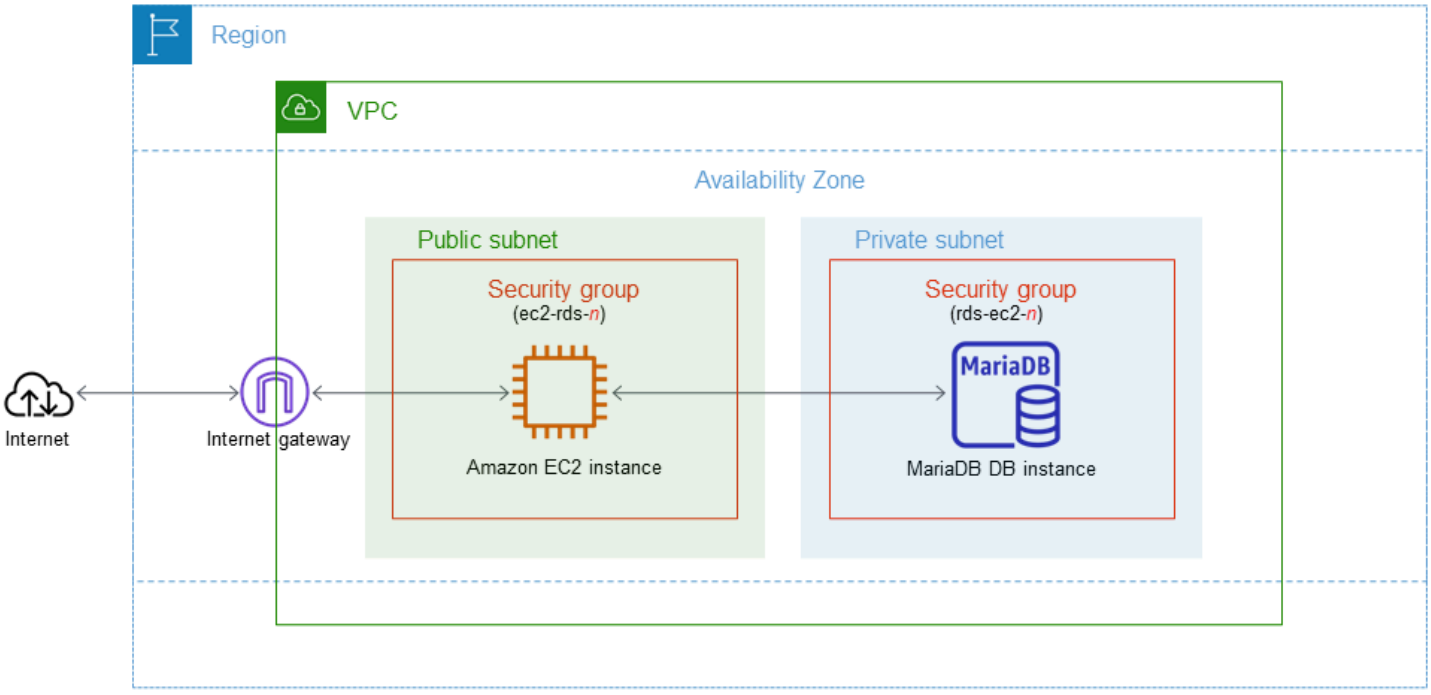
Membuat dan menghubungkan ke instans DB MariaDB

Tutorial ini membuat instans EC2 dan instans DB RDS for MariaDB. Tutorial ini menunjukkan cara mengakses instans DB dari instans EC2 menggunakan klien MySQL standar. Sebagai praktik terbaik, tutorial ini membuat instans DB privat dalam cloud privat virtual (VPC). Dalam kebanyakan kasus, sumber daya lain dalam VPC yang sama, seperti instans EC2, dapat mengakses instans DB, tetapi sumber daya di luar VPC tidak dapat mengaksesnya.

Setelah Anda menyelesaikan tutorial, ada subnet publik dan privat di setiap Zona Ketersediaan di VPC Anda. Dalam satu Zona Ketersediaan, instans EC2 berada di subnet publik, dan instans DB berada di subnet privat.

> **Important**
>
> Tidak ada biaya untuk membuat Akun AWS. Namun, dengan menyelesaikan tutorial ini, Anda mungkin akan dikenai biaya untuk sumber daya yang Anda gunakan. Anda dapat menghapus sumber daya ini setelah menyelesaikan tutorial jika tidak diperlukan lagi.

Diagram berikut menunjukkan konfigurasi setelah tutorial selesai.



Tutorial ini memungkinkan Anda untuk membuat sumber daya Anda dengan menggunakan salah satu metode berikut:

1. Gunakan AWS Management Console - Langkah 1: Buat instans EC2 dan Langkah 2: Buat instans DB MariaDB
2. Gunakan AWS CloudFormation untuk membuat instance database dan instans EC2 - (Opsional) Buat instance VPC, EC2, dan MariaDB menggunakan AWS CloudFormation

Metode pertama menggunakan Easy create untuk membuat instance MariaDB pribadi dengan. AWS Management Console Di sini, Anda hanya menentukan jenis mesin DB, ukuran instans DB, dan pengidentifikasi instans DB. Pembuatan Mudah menggunakan pengaturan default untuk opsi konfigurasi lainnya.

Saat Anda menggunakan Standard create sebagai gantinya, Anda dapat menentukan lebih banyak opsi konfigurasi saat membuat instans DB. Opsi ini mencakup pengaturan untuk ketersediaan, keamanan, cadangan, dan pemeliharaan. Untuk membuat instans DB publik, Anda harus menggunakan Pembuatan Standar. Untuk informasi, lihat Membuat instans DB Amazon RDS.

Topik

Prasyarat

Sebelum memulai, selesaikan langkah-langkah di bagian berikut:

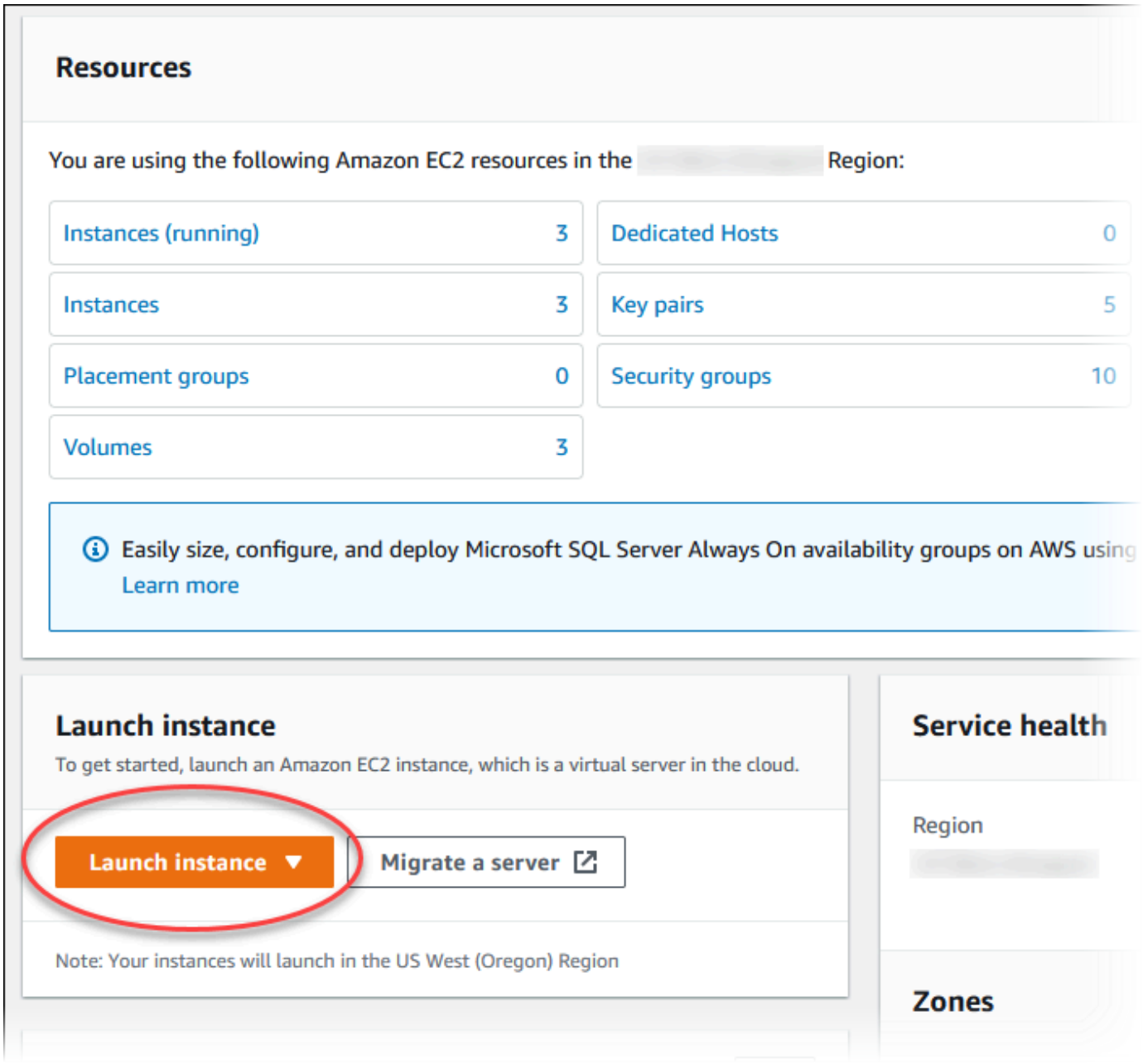
- Mendaftar Akun AWS
- Membuat pengguna administratif

Langkah 1: Buat instans EC2

Buat instans Amazon EC2 yang akan Anda gunakan untuk menghubungkan ke basis data Anda.

Untuk membuat instans EC2

1. Masuk ke AWS Management Console dan buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di sudut kanan atas AWS Management Console, pilih Wilayah AWS di mana Anda ingin membuat instans EC2.
3. Pilih Dasbor EC2, lalu pilih Luncurkan instans seperti yang ditampilkan dalam gambar berikut.

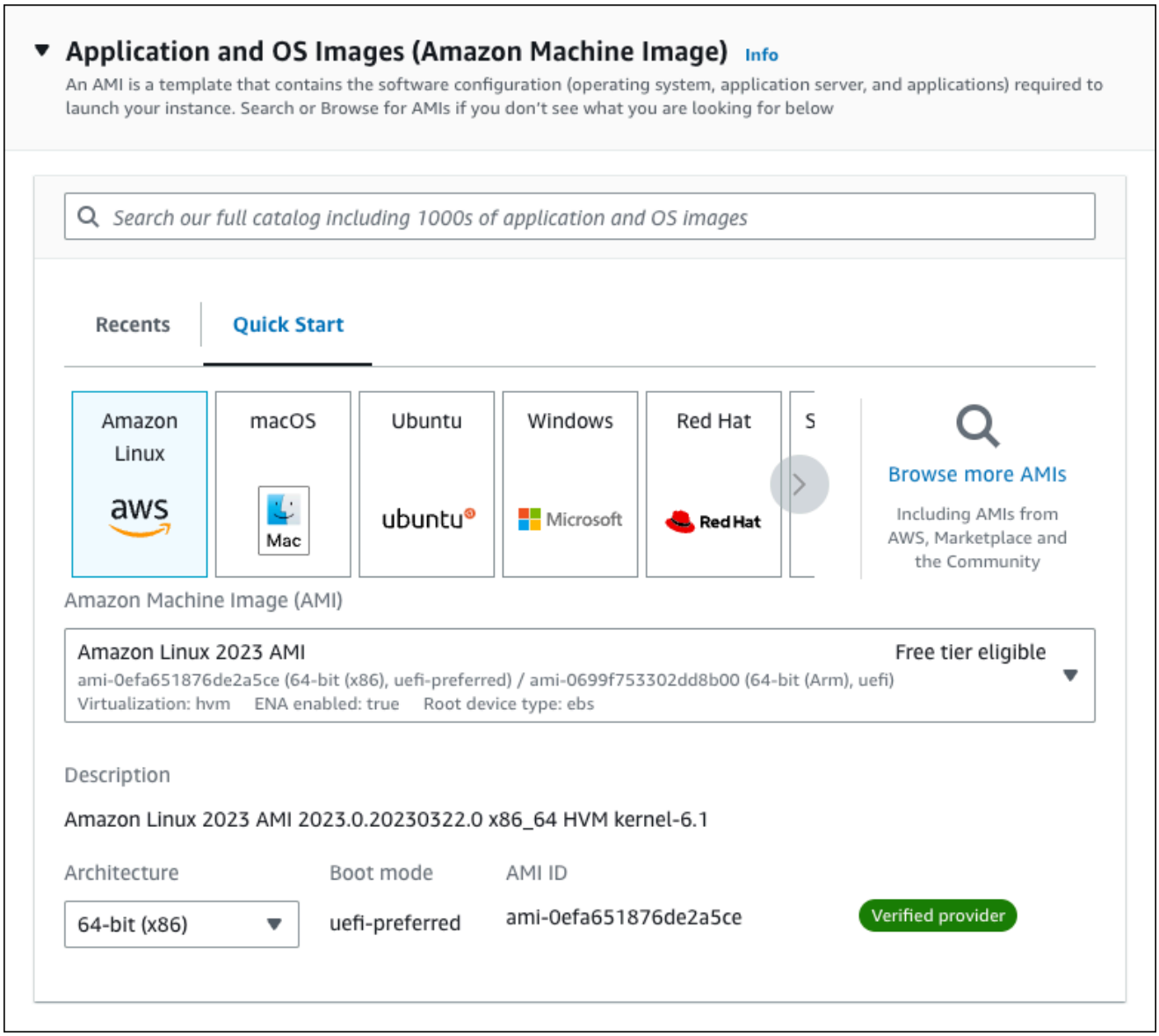


Halaman Meluncurkan instans akan terbuka.

4. Pilih pengaturan berikut di halaman Meluncurkan instans.

   a. Di bagian Nama dan tag, untuk Nama, masukkan **`ec2-database-connect`**.

   b. Di bagian Gambar Aplikasi dan OS (Amazon Machine Image), pilih Amazon Linux, lalu pilih AMI Amazon Linux 2023. Biarkan default untuk pilihan lainnya.



   c. Di bagian Jenis instans, pilih t2.micro.

   d. Di bagian Pasangan kunci (login), pilih nama Pasangan kunci untuk menggunakan pasangan kunci yang ada. Untuk membuat pasangan kunci baru untuk instans Amazon EC2, pilih Buat Pasangan kunci baru lalu gunakan jendela Buat pasangan kunci untuk membuatnya.

Untuk informasi selengkapnya tentang membuat pasangan kunci baru, lihat Membuat pasangan kunci di Panduan Pengguna Amazon EC2 untuk Instans Linux.

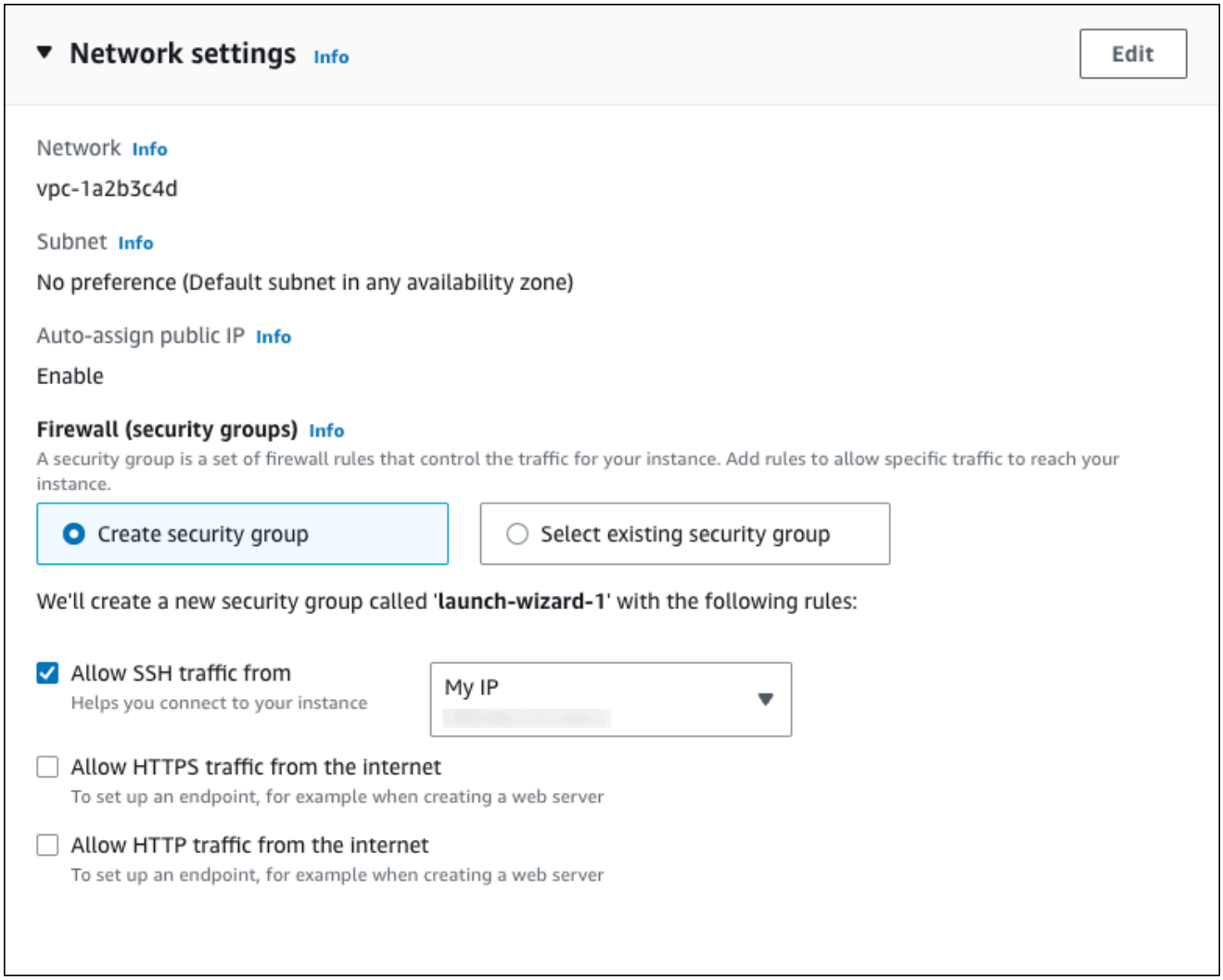
e. Untuk Izinkan lalu lintas SSH di Pengaturan jaringan, pilih sumber koneksi SSH ke instans EC2.

Anda dapat memilih IP Saya jika alamat IP yang ditampilkan benar untuk koneksi SSH. Jika tidak, Anda dapat menentukan alamat IP yang akan digunakan untuk menghubungkan ke instans EC2 di VPC Anda menggunakan Secure Shell (SSH). Untuk menentukan alamat IP publik Anda, Anda dapat membuka layanan di https://checkip.amazonaws.com di jendela atau tab browser lain. Contoh alamat IP adalah 192.0.2.1/32.

Dalam banyak kasus, Anda dapat menghubungkan melalui penyedia layanan Internet (ISP) atau dari belakang firewall Anda tanpa alamat IP statis. Jika demikian, tentukan rentang alamat IP yang digunakan oleh komputer klien.

> ⚠ Warning
>
> Jika menggunakan `0.0.0.0/0` untuk akses SSH, Anda memungkinkan semua alamat IP untuk mengakses instans publik EC2 Anda menggunakan SSH. Hal ini dapat diterima untuk waktu yang singkat di lingkungan pengujian, tetapi tidak aman untuk lingkungan produksi. Dalam produksi, Anda hanya dapat memberikan otorisasi pada alamat IP atau rentang alamat tertentu saja untuk mengakses instans EC2 Anda menggunakan SSH.

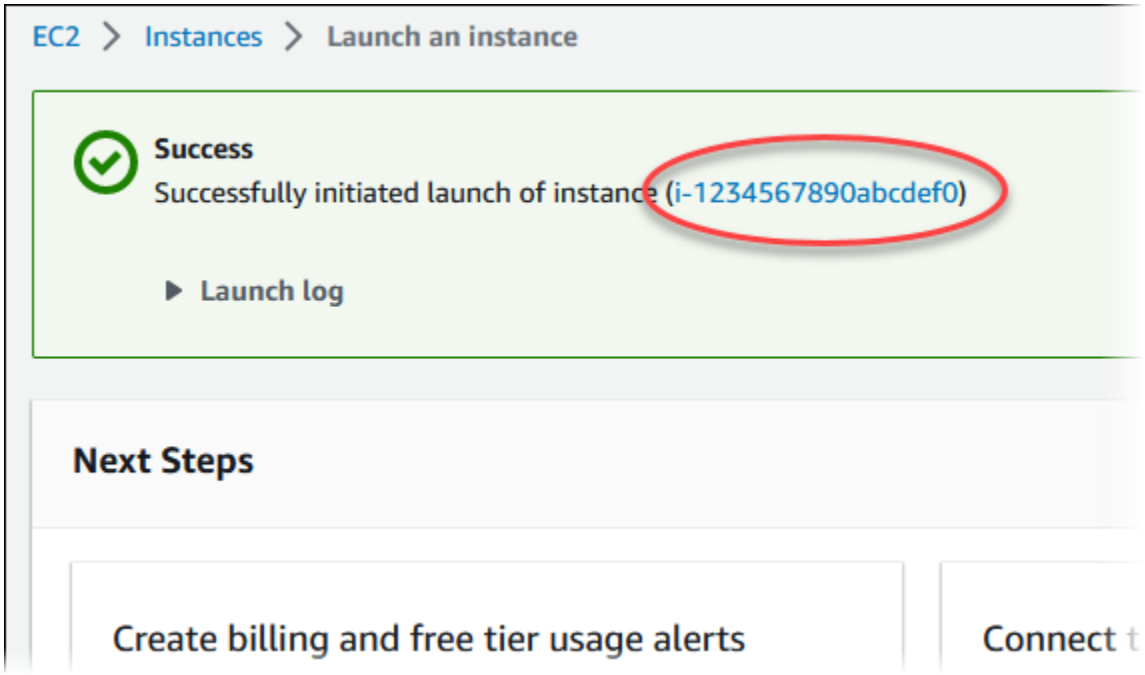
Gambar berikut menunjukkan contoh bagian Pengaturan jaringan.



f. Biarkan nilai default untuk bagian yang lainnya.

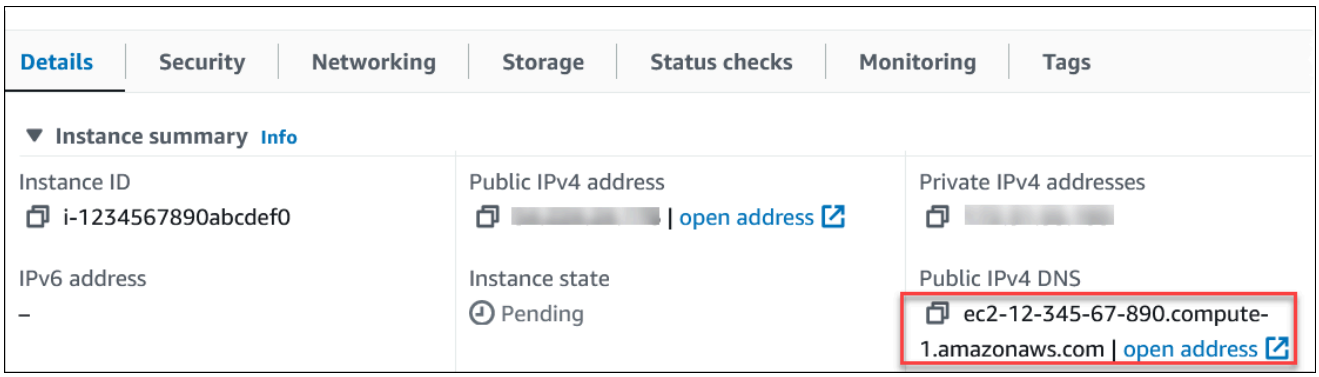
g. Tinjau ringkasan konfigurasi instans EC2 Anda di panel Ringkasan, dan setelah Anda siap, pilih Luncurkan instans.

5. Di halaman Status Peluncuran, catat pengidentifikasi untuk instans EC2 baru Anda, misalnya: `i-1234567890abcdef0`.

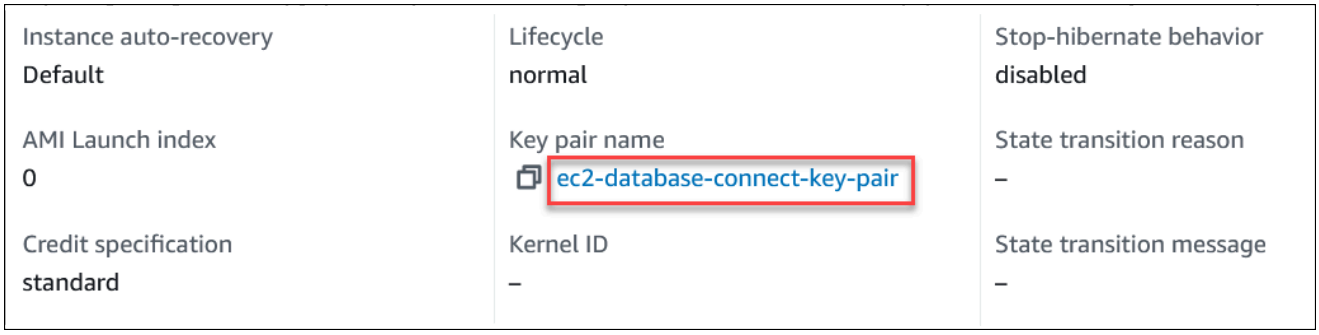


6. Pilih pengidentifikasi instans EC2 untuk membuka daftar instans EC2, lalu pilih instans EC2 Anda.
7. Di tab Detail, catat nilai-nilai berikut, yang akan Anda butuhkan saat menghubungkan menggunakan SSH:

   a. Di Ringkasan instans, catat nilai untuk DNS IPv4 Publik.



   b. Di Detail instans, catat nilai untuk Nama pasangan kunci.



8. Tunggu hingga Status instance untuk instans EC2 Anda berstatus Berjalan sebelum melanjutkan.

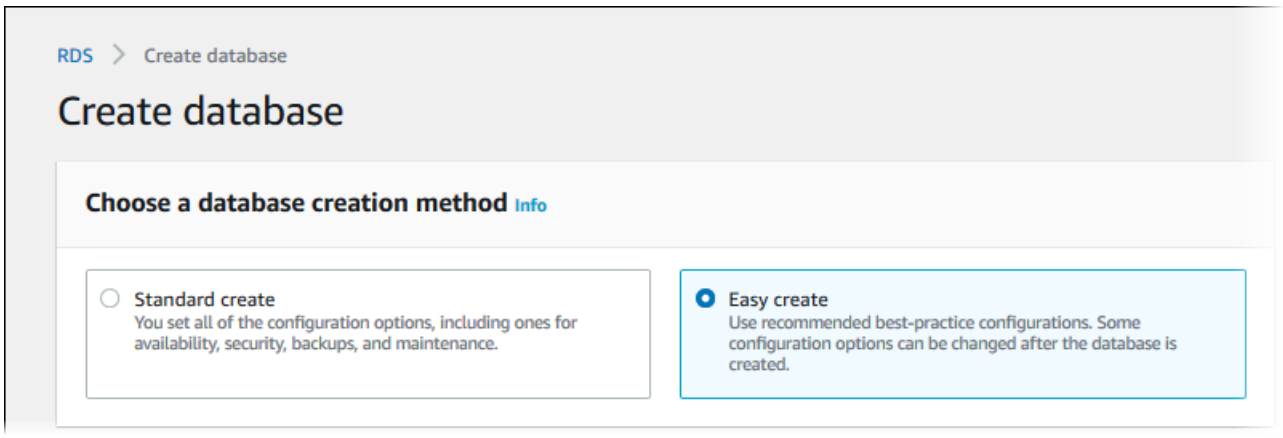
Langkah 2: Buat instans DB MariaDB

Blok bangunan dasar Amazon RDS adalah instans DB. Lingkungan ini adalah tempat Anda menjalankan basis data MariaDB Anda.

Dalam contoh ini, Anda menggunakan Pembuatan Mudah untuk membuat instans DB yang menjalankan mesin basis data MariaDB dengan kelas instans DB db.t3.micro.

Untuk membuat instans DB MariaDB dengan Pembuatan Mudah

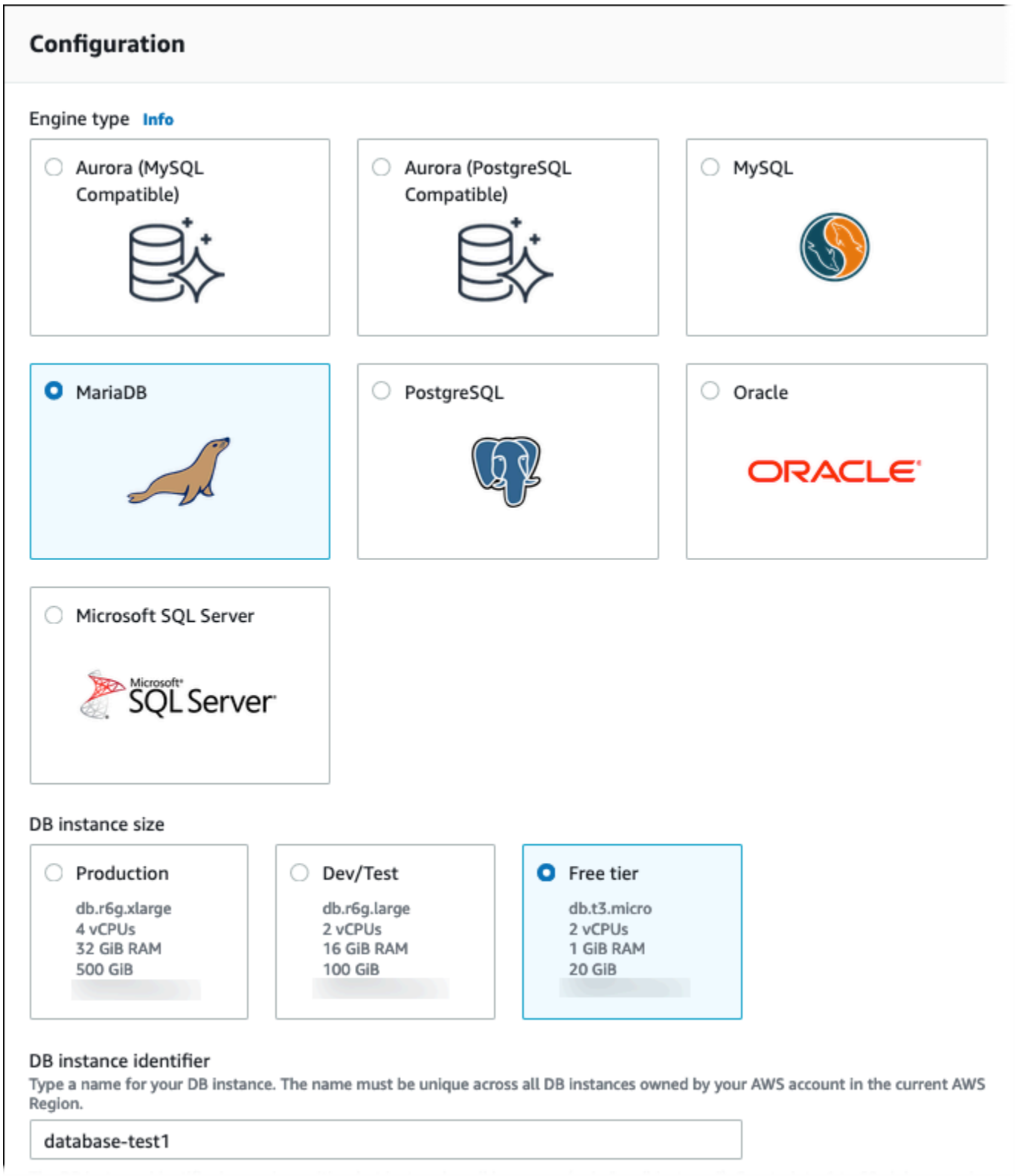
1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di sudut kanan atas konsol Amazon RDS, pilih Wilayah AWS di mana Anda ingin membuat instans DB.
3. Di panel navigasi, pilih Basis Data.
4. Pilih Buat basis data dan pastikan Pembuatan Mudah dipilih.



5. Di Konfigurasi, pilih MariaDB.

6. Untuk Ukuran instans DB, pilih Tingkat gratis.
7. Untuk Pengidentifikasi instans DB, masukkan **`database-test1`**.
8. Untuk Nama pengguna utama, masukkan nama untuk pengguna utama, atau tetap gunakan nama default.

   Tampilan halaman Membuat basis data seperti gambar berikut.

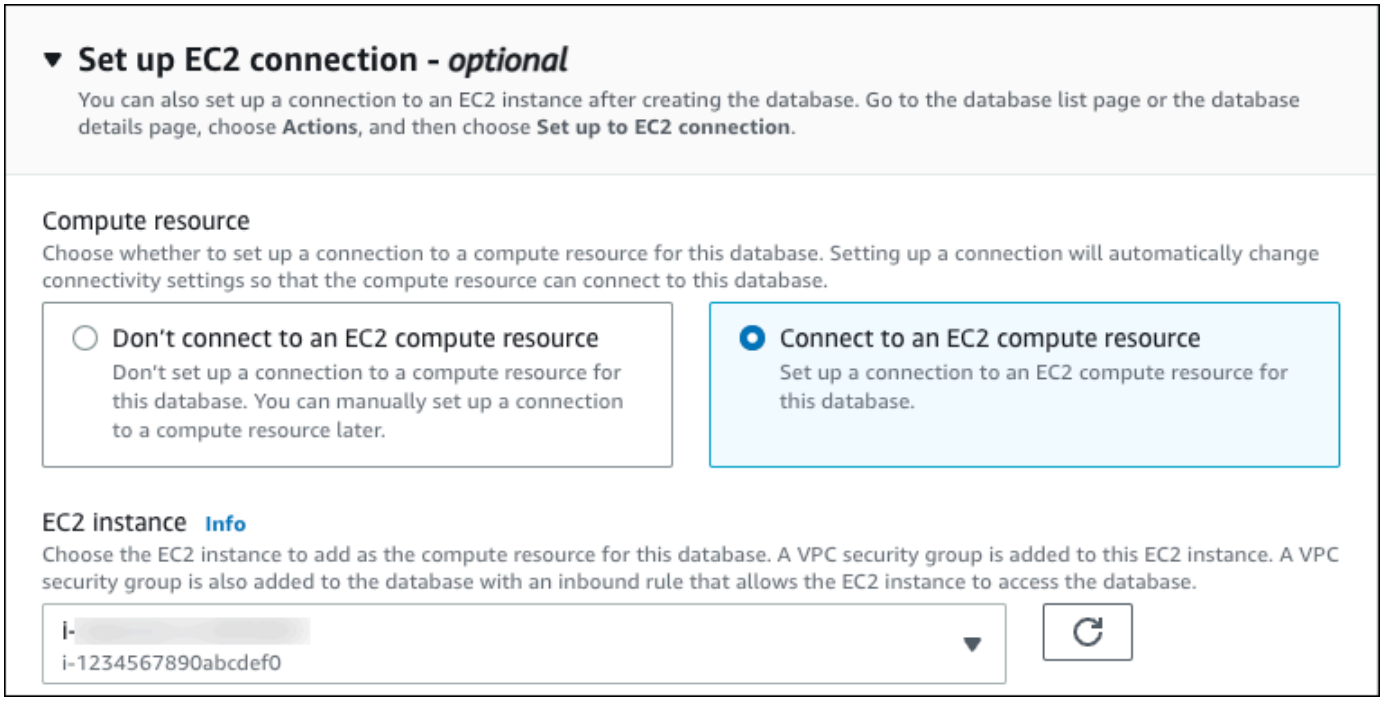


9. Untuk menggunakan kata sandi utama yang dibuat secara otomatis untuk instans DB, pilih Buat kata sandi secara otomatis.

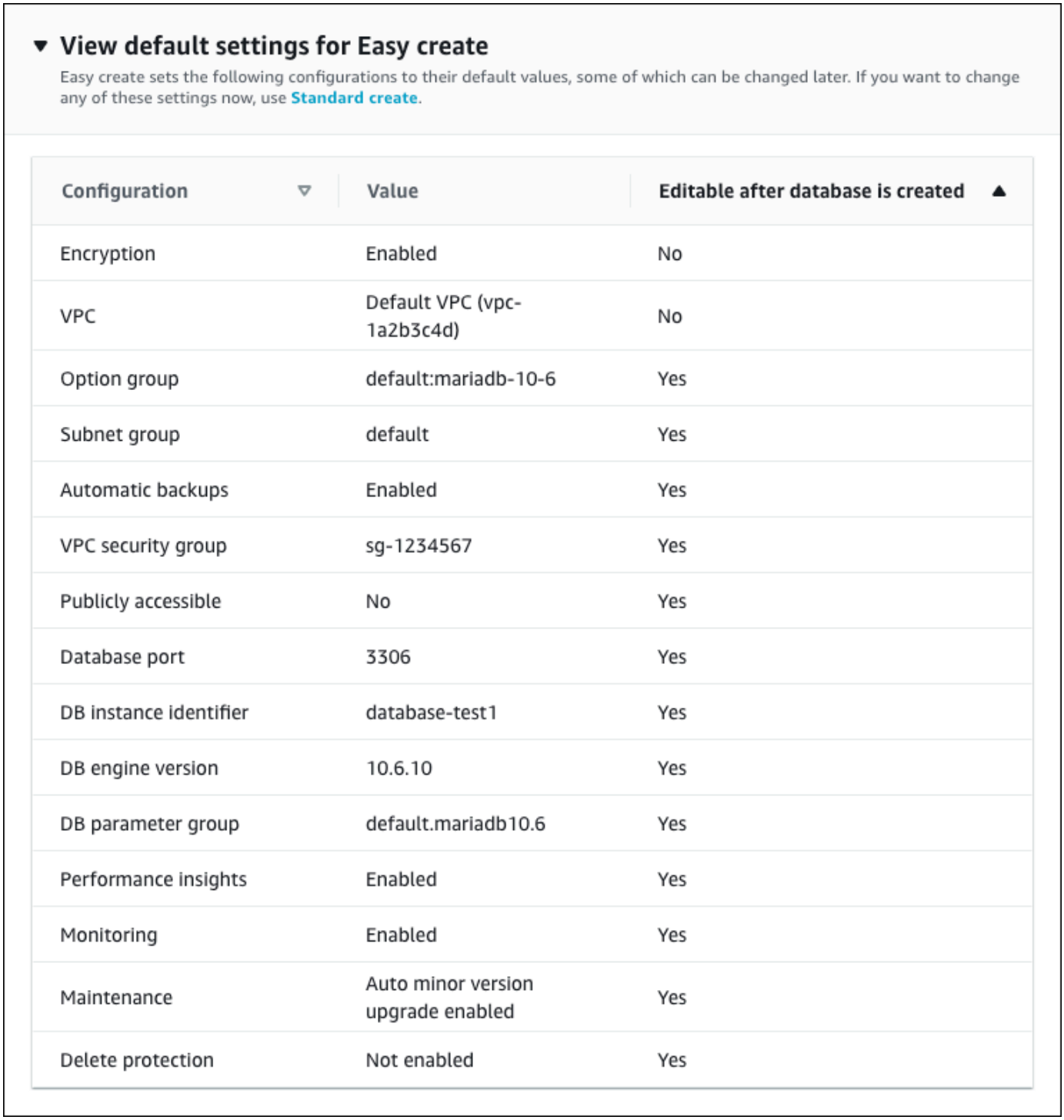
Untuk memasukkan kata sandi utama Anda, hapus centang pada Buat kata sandi secara otomatis, lalu masukkan kata sandi yang sama dalam Kata sandi utama dan Konfirmasi kata sandi.

10. Untuk menyiapkan koneksi dengan instans EC2 yang Anda buat sebelumnya, buka Menyiapkan koneksi EC2 - opsional.

    Pilih Hubungkan ke sumber daya komputasi EC2. Pilih instans EC2 yang Anda buat sebelumnya.



11. Buka Lihat pengaturan default untuk Pembuatan Mudah.



View default settings for Easy create

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

Configuration	Value	Editable after database is created
Encryption	Enabled	No
VPC	Default VPC (vpc-1a2b3c4d)	No
Option group	default:mariadb-10-6	Yes
Subnet group	default	Yes
Automatic backups	Enabled	Yes
VPC security group	sg-1234567	Yes
Publicly accessible	No	Yes
Database port	3306	Yes
DB instance identifier	database-test1	Yes
DB engine version	10.6.10	Yes
DB parameter group	default.mariadb10.6	Yes
Performance insights	Enabled	Yes
Monitoring	Enabled	Yes
Maintenance	Auto minor version upgrade enabled	Yes
Delete protection	Not enabled	Yes

Anda dapat memeriksa pengaturan default yang digunakan dengan Pembuatan mudah. Kolom Dapat diedit setelah basis data dibuat menunjukkan opsi yang dapat Anda ubah setelah membuat basis data.

- Jika pengaturan memiliki Tidak di kolom tersebut, dan Anda menginginkan pengaturan yang berbeda, Anda dapat menggunakan Pembuatan Standar untuk membuat instans DB.
- Jika pengaturan memiliki Ya di kolom tersebut, dan Anda menginginkan pengaturan yang berbeda, Anda dapat menggunakan Pembuatan Standar untuk membuat instans DB, atau mengubah instans DB setelah Anda membuatnya untuk mengubah pengaturan.

12. Pilih Buat basis data.

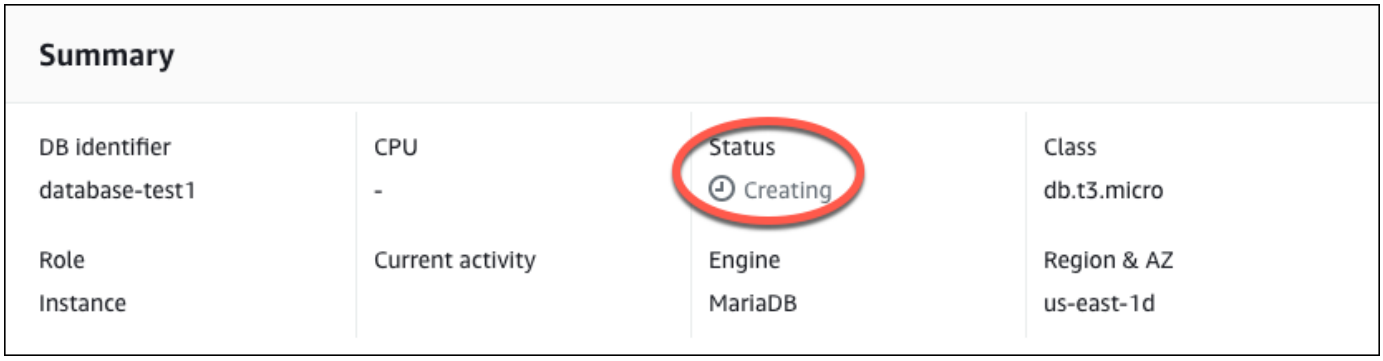
    Untuk melihat nama pengguna dan kata sandi utama untuk instans DB, pilih Lihat detail kredensial.

    Anda dapat menggunakan nama pengguna dan kata sandi yang ditampilkan untuk terhubung ke instans DB sebagai pengguna utama.

    > **Important**
    >
    > Anda tidak dapat melihat kata sandi pengguna utama lagi. Jika tidak mencatatnya, Anda mungkin harus mengubahnya.
    > Jika perlu mengubah kata sandi pengguna utama setelah instans DB tersedia, Anda dapat mengubah instans DB untuk melakukannya. Untuk informasi selengkapnya tentang cara mengubah instans DB, lihat Memodifikasi instans DB Amazon RDS.

13. Dalam daftar Basis Data, pilih nama instans DB MariaDB yang baru untuk menampilkan detailnya.

    Instans DB memiliki status Membuat hingga siap digunakan.



    Saat statusnya berubah menjadi Tersedia, Anda dapat terhubung ke instans DB. Tergantung pada kelas instans DB dan jumlah penyimpanan, diperlukan waktu hingga 20 menit sebelum instans baru tersedia.

(Opsional) Buat instance VPC, EC2, dan MariaDB menggunakan AWS CloudFormation

Alih-alih menggunakan konsol untuk membuat instance VPC, EC2, dan MariaDB, Anda dapat menggunakannya AWS CloudFormation untuk menyediakan AWS sumber daya dengan memperlakukan infrastruktur sebagai kode. Untuk membantu Anda mengatur AWS sumber daya Anda menjadi unit yang lebih kecil dan lebih mudah dikelola, Anda dapat menggunakan fungsionalitas tumpukan AWS CloudFormation bersarang. Untuk informasi selengkapnya, lihat Membuat tumpukan di AWS CloudFormation konsol dan Bekerja dengan tumpukan bersarang.

> **Important**
>
> AWS CloudFormation gratis, tetapi sumber daya yang CloudFormation menciptakan hidup. Anda dikenakan biaya penggunaan standar untuk sumber daya ini sampai Anda menghentikannya. Total biaya akan minimal. Untuk informasi tentang cara meminimalkan biaya apa pun, buka Tingkat AWS Gratis.

Untuk membuat sumber daya Anda menggunakan AWS CloudFormation konsol, selesaikan langkah-langkah berikut:

- Langkah 1: Unduh CloudFormation template
- Langkah 2: Konfigurasikan sumber daya Anda menggunakan CloudFormation

Unduh CloudFormation template

CloudFormation Template adalah file teks JSON atau YAMB yang berisi informasi konfigurasi tentang sumber daya yang ingin Anda buat di tumpukan. Template ini juga membuat VPC dan host bastion untuk Anda bersama dengan instance RDS.

Untuk mengunduh file template, buka tautan berikut, template MariaDB CloudFormation .

Di halaman Github, klik tombol Unduh file mentah untuk menyimpan file YAMAL template.

Konfigurasikan sumber daya Anda menggunakan CloudFormation

> **Note**
>
> Sebelum memulai proses ini, pastikan Anda memiliki pasangan Kunci untuk instans EC2 di. Akun AWS Untuk informasi selengkapnya, lihat Pasangan kunci Amazon EC2 dan instans Linux.

Ketika Anda menggunakan AWS CloudFormation template, Anda harus memilih parameter yang benar untuk memastikan sumber daya Anda dibuat dengan benar. Ikuti langkah-langkah di bawah ini:

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di https://console.aws.amazon.com/cloudformation.
2. Pilih Buat tumpukan.
3. Di bagian Tentukan templat, pilih Unggah file templat dari komputer Anda, lalu pilih Berikutnya.
4. Di halaman Tentukan detail tumpukan, atur parameter berikut:
   a. Tetapkan nama Stack ke MariaDB TestStack.
   b. Di bawah Parameter, atur Availability Zone dengan memilih tiga zona ketersediaan.
   c. Di bawah konfigurasi Linux Bastion Host, untuk Key Name, pilih key pair untuk login ke instans EC2 Anda.
   d. Dalam pengaturan konfigurasi Linux Bastion Host, atur rentang IP yang Diizinkan ke alamat IP Anda. Untuk terhubung ke instans EC2 di VPC Anda menggunakan Secure Shell (SSH), tentukan alamat IP publik Anda menggunakan layanan di https://checkip.amazonaws.com. Contoh alamat IP adalah 192.0.2.1/32.

      > **Warning**
      >
      > Jika menggunakan `0.0.0.0/0` untuk akses SSH, Anda memungkinkan semua alamat IP untuk mengakses instans publik EC2 Anda menggunakan SSH. Hal ini dapat diterima untuk waktu yang singkat di lingkungan pengujian, tetapi tidak aman untuk lingkungan produksi. Dalam produksi, Anda hanya dapat memberikan otorisasi pada alamat IP atau rentang alamat tertentu saja untuk mengakses instans EC2 Anda menggunakan SSH.

   e. Di bawah konfigurasi Database General, atur kelas instance Database ke db.t3.micro.

f. Tetapkan nama Database ke**`database-test1`**.

g. Untuk nama pengguna master Database, masukkan nama untuk pengguna master.

h. Atur Kelola kata sandi pengguna master DB dengan Secrets Manager `false` untuk tutorial ini.

i. Untuk kata sandi Database, tetapkan kata sandi pilihan Anda. Ingat kata sandi ini untuk langkah lebih lanjut dalam tutorial.

j. Di bawah konfigurasi Penyimpanan Database, atur tipe penyimpanan Database ke gp2.

k. Di bawah konfigurasi Pemantauan Database, atur Aktifkan Performance Insights RDS ke false.

l. Biarkan semua pengaturan lainnya sebagai nilai default. Klik Berikutnya untuk melanjutkan.

5. Di halaman tumpukan Tinjauan, pilih Kirim setelah memeriksa database dan opsi host bastion Linux.

Setelah proses pembuatan tumpukan selesai, lihat tumpukan dengan nama BastionStackdan RDSNS untuk mencatat informasi yang Anda butuhkan untuk terhubung ke database. Untuk informasi selengkapnya, lihat Melihat data AWS CloudFormation tumpukan dan sumber daya di AWS Management Console.

Langkah 3: Hubungkan ke instans DB MariaDB

Anda dapat menggunakan aplikasi klien SQL standar untuk menghubungkan ke instans DB. Dalam contoh ini, Anda akan menghubungkan ke instans DB MariaDB menggunakan klien baris perintah mysql.

Untuk menghubungkan ke instans DB MariaDB

1. Temukan titik akhir (nama DNS) dan nomor port untuk instans DB Anda.

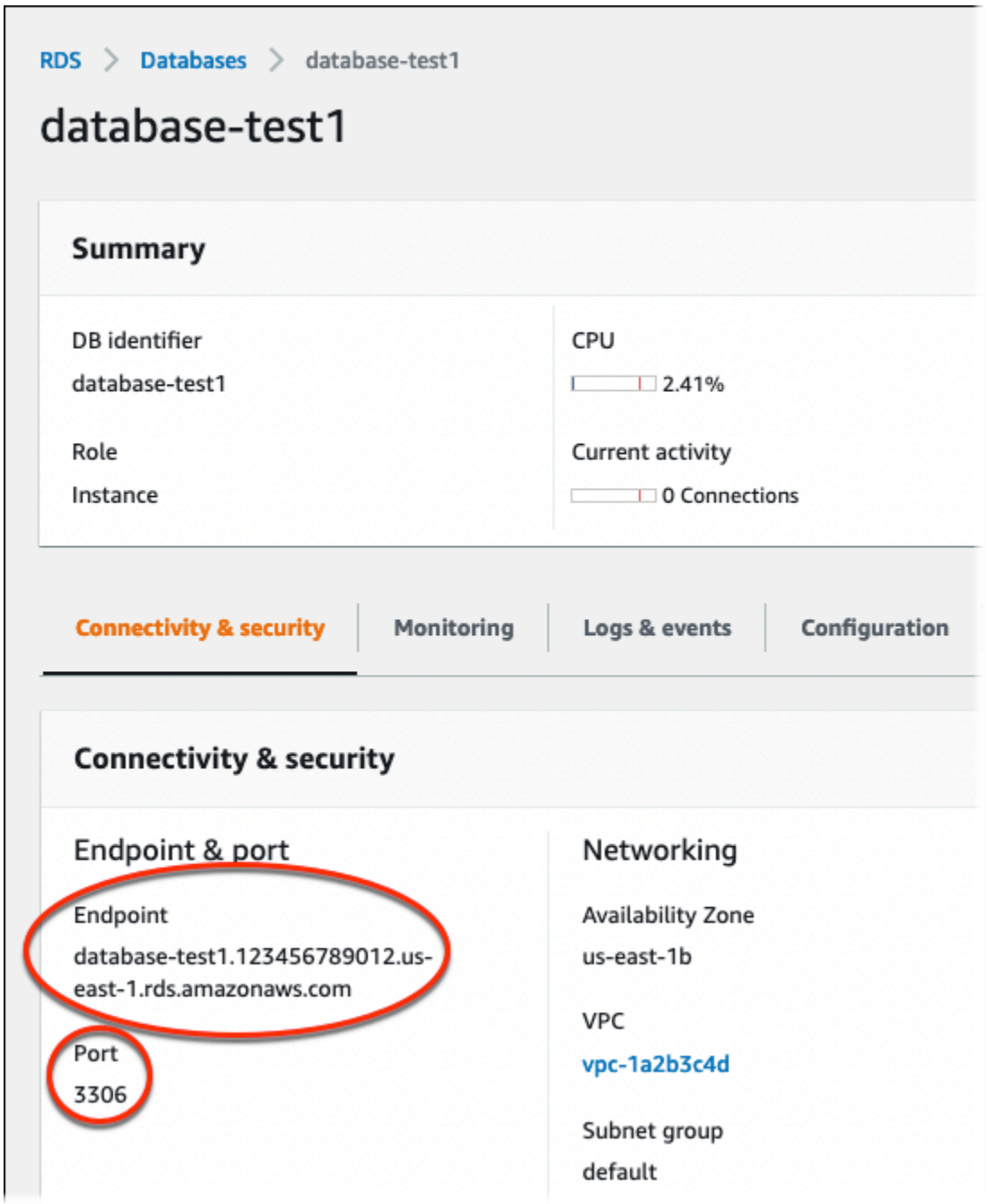
   a. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.

   b. Di sudut kanan atas konsol Amazon RDS, pilih instans DB Wilayah AWS .

   c. Di panel navigasi, pilih Basis Data.

   d. Pilih nama instans DB MariaDB untuk menampilkan detailnya.

   e. Di tab Konektivitas & keamanan, salin titik akhir. Perhatikan juga nomor port. Anda memerlukan titik akhir dan nomor port untuk terhubung ke instans DB.



2. Hubungkan ke instans EC2 yang Anda buat sebelumnya dengan mengikuti langkah-langkah di Menghubungkan ke instans Linux di Panduan Pengguna Amazon EC2 untuk Instans Linux.

   Sebaiknya Anda menghubungkan ke instans EC2 menggunakan SSH. Jika utilitas klien SSH diinstal di Windows, Linux, atau Mac, Anda dapat menghubungkan ke instans menggunakan format perintah berikut:

```
ssh -i location_of_pem_file ec2-user@ec2-instance-public-dns-name
```

Misalnya, asumsikan bahwa `ec2-database-connect-key-pair.pem` disimpan di `/dir1` di Linux, dan DNS IPv4 publik untuk instans EC2 Anda adalah `ec2-12-345-678-90.compute-1.amazonaws.com`. Perintah SSH Anda akan tampak seperti berikut:

```
ssh -i /dir1/ec2-database-connect-key-pair.pem ec2-
user@ec2-12-345-678-90.compute-1.amazonaws.com
```

3. Dapatkan pembaruan keamanan dan perbaikan bug terbaru dengan memperbarui perangkat lunak di instans EC2 Anda. Untuk melakukannya, gunakan perintah berikut.

> **Note**
>
> Opsi `-y` menginstal pembaruan tanpa meminta konfirmasi. Hilangkan opsi ini untuk memeriksa pembaruan sebelum menginstal.

```
sudo dnf update -y
```

4. Instal klien baris perintah mysql dari MariaDB.

   Untuk menginstal klien baris perintah MariaDB di Amazon Linux 2023, jalankan perintah berikut:

```
sudo dnf install mariadb105
```

5. Hubungkan ke instans DB MariaDB. Misalnya, masukkan perintah berikut. Tindakan ini memungkinkan Anda untuk terhubung ke instans DB MariaDB menggunakan klien MySQL.

   Ganti titik akhir instans DB (nama DNS) untuk *`endpoint`*, dan ganti nama pengguna utama yang Anda gunakan untuk *`admin`*. Masukkan kata sandi utama yang Anda gunakan saat dimintai kata sandi.

```
mysql -h endpoint -P 3306 -u admin -p
```

   Setelah memasukkan kata sandi untuk pengguna, Anda akan melihat output yang serupa dengan yang berikut ini.

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
```

```
Your MariaDB connection id is 156
Server version: 10.6.10-MariaDB-log managed by https://aws.amazon.com/rds/

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]>
```

Untuk informasi selengkapnya tentang cara menghubungkan ke instans DB MariaDB, lihat Menghubungkan ke instans DB yang menjalankan mesin basis data MariaDB. Jika Anda tidak dapat terhubung ke instans DB Anda, lihat Tidak dapat terhubung ke instans DB Amazon RDS.

Untuk keamanan, praktik terbaiknya adalah menggunakan koneksi terenkripsi. Hanya gunakan koneksi MariaDB yang tidak terenkripsi saat klien dan server berada di VPC yang sama dan jaringan tepercaya. Untuk informasi tentang cara menggunakan koneksi terenkripsi, lihat Menghubungkan dari klien baris perintah MySQL dengan SSL/TLS (terenkripsi).

6. Jalankan perintah SQL.

   Misalnya, perintah SQL berikut menunjukkan tanggal dan waktu saat ini:

   ```
   SELECT CURRENT_TIMESTAMP;
   ```

Langkah 4: Hapus instans EC2 dan instans DB

Setelah Anda terhubung ke dan menjelajahi instans EC2 dan instans DB sampel yang Anda buat, hapus instans tersebut sehingga Anda tidak lagi dikenakan biaya untuk instans DB tersebut.

Jika Anda biasa AWS CloudFormation membuat sumber daya, lewati langkah ini dan lanjutkan ke langkah berikutnya.

Untuk menghapus instans EC2

1. Masuk ke AWS Management Console dan buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Instans.
3. Pilih instans EC2, dan pilih Status instans, Akhiri instans.
4. Pilih Akhiri saat diminta untuk konfirmasi.

Untuk informasi selengkapnya tentang menghapus instans EC2, lihat Mengakhiri Instans dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Untuk menghapus instans DB tanpa snapshot DB akhir

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data.
3. Pilih instans DB yang ingin Anda hapus.
4. Untuk Tindakan, pilih Hapus.
5. Hapus Buat snapshot akhir? dan Pertahankan pencadangan otomatis.
6. Lengkapi pengakuan dan pilih Hapus.

(Opsional) Hapus instans EC2 dan instans DB yang dibuat dengan CloudFormation

Jika Anda biasa AWS CloudFormation membuat sumber daya, hapus CloudFormation tumpukan setelah Anda terhubung dan jelajahi contoh instans EC2 dan instans DB, sehingga Anda tidak lagi dikenakan biaya untuk itu.

Untuk menghapus sumber CloudFormation daya

1. Buka AWS CloudFormation konsol.
2. Pada halaman Stacks di CloudFormationconsole, pilih tumpukan root (tumpukan tanpa nama VPCStack, BastionStack atau RDSNS).
3. Pilih Hapus.
4. Pilih Hapus tumpukan saat diminta konfirmasi.

Untuk informasi selengkapnya tentang menghapus tumpukan CloudFormation, lihat Menghapus tumpukan di AWS CloudFormation konsol di AWS CloudFormation Panduan Pengguna.

(Opsional) Menghubungkan instans DB Anda ke fungsi Lambda

Anda juga dapat menghubungkan instans DB RDS for MariaDB ke sumber daya komputasi nirserver Lambda. Fungsi Lambda memungkinkan Anda menjalankan kode tanpa menyediakan atau mengelola infrastruktur. Fungsi Lambda juga memungkinkan Anda untuk otomatis merespons

permintaan eksekusi kode pada skala apa pun, mulai dari selusin peristiwa dalam sehari hingga ratusan per detik. Lihat informasi yang lebih lengkap di Menghubungkan secara otomatis fungsi Lambda dan instans basis data.

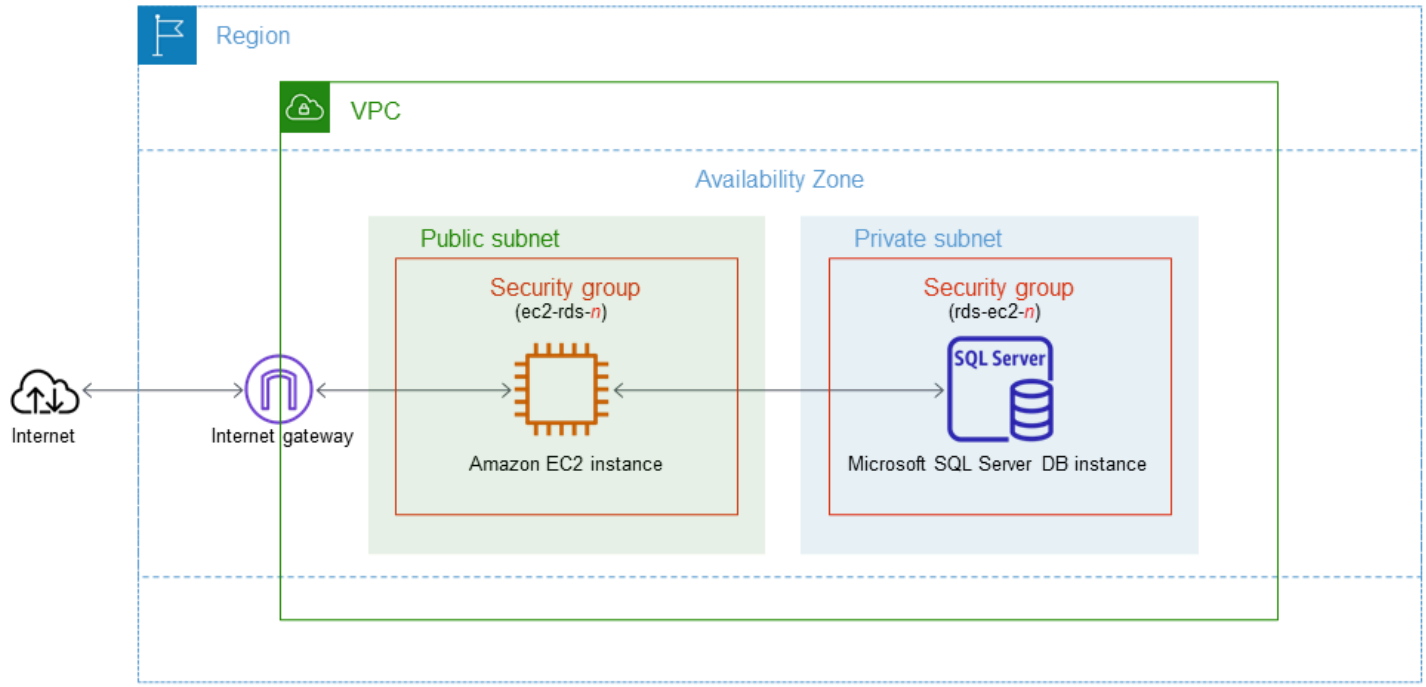
Membuat dan menghubungkan ke instans DB Microsoft SQL Server

Tutorial ini membuat instans EC2 dan instans DB RDS for Microsoft SQL Server. Tutorial ini menunjukkan cara mengakses instans DB dari instans EC2 menggunakan klien Microsoft SQL Server Management Studio. Sebagai praktik terbaik, tutorial ini membuat instans DB privat dalam cloud privat virtual (VPC). Dalam kebanyakan kasus, sumber daya lain dalam VPC yang sama, seperti instans EC2, dapat mengakses instans DB, tetapi sumber daya di luar VPC tidak dapat mengaksesnya.

Setelah Anda menyelesaikan tutorial, ada subnet publik dan privat di setiap Zona Ketersediaan di VPC Anda. Dalam satu Zona Ketersediaan, instans EC2 berada di subnet publik, dan instans DB berada di subnet privat.

> **Important**
>
> Tidak ada biaya untuk membuat AWS akun. Namun, dengan menyelesaikan tutorial ini, Anda mungkin dikenakan biaya untuk AWS sumber daya yang Anda gunakan. Anda dapat menghapus sumber daya ini setelah menyelesaikan tutorial jika tidak diperlukan lagi.

Diagram berikut menunjukkan konfigurasi setelah tutorial selesai.



Tutorial ini memungkinkan Anda untuk membuat sumber daya Anda dengan menggunakan salah satu metode berikut:

1. Gunakan AWS Management Console - Langkah 2: Buat instans DB SQL Server dan Langkah 1: Buat instans EC2
2. Gunakan AWS CloudFormation untuk membuat instance database dan instans EC2 - (Opsional) Buat instance VPC, EC2, dan SQL Server menggunakan AWS CloudFormation

Metode pertama menggunakan Easy create untuk membuat instance SQL Server DB pribadi dengan file. AWS Management Console Di sini, Anda hanya menentukan jenis mesin DB, ukuran instans DB, dan pengidentifikasi instans DB. Pembuatan Mudah menggunakan pengaturan default untuk opsi konfigurasi lainnya.

Saat Anda menggunakan Standard create sebagai gantinya, Anda dapat menentukan lebih banyak opsi konfigurasi saat membuat instans DB. Opsi ini mencakup pengaturan untuk ketersediaan, keamanan, cadangan, dan pemeliharaan. Untuk membuat instans DB publik, Anda harus menggunakan Pembuatan Standar. Untuk informasi, lihat Membuat instans DB Amazon RDS.

Topik

Prasyarat

Sebelum memulai, selesaikan langkah-langkah di bagian berikut:

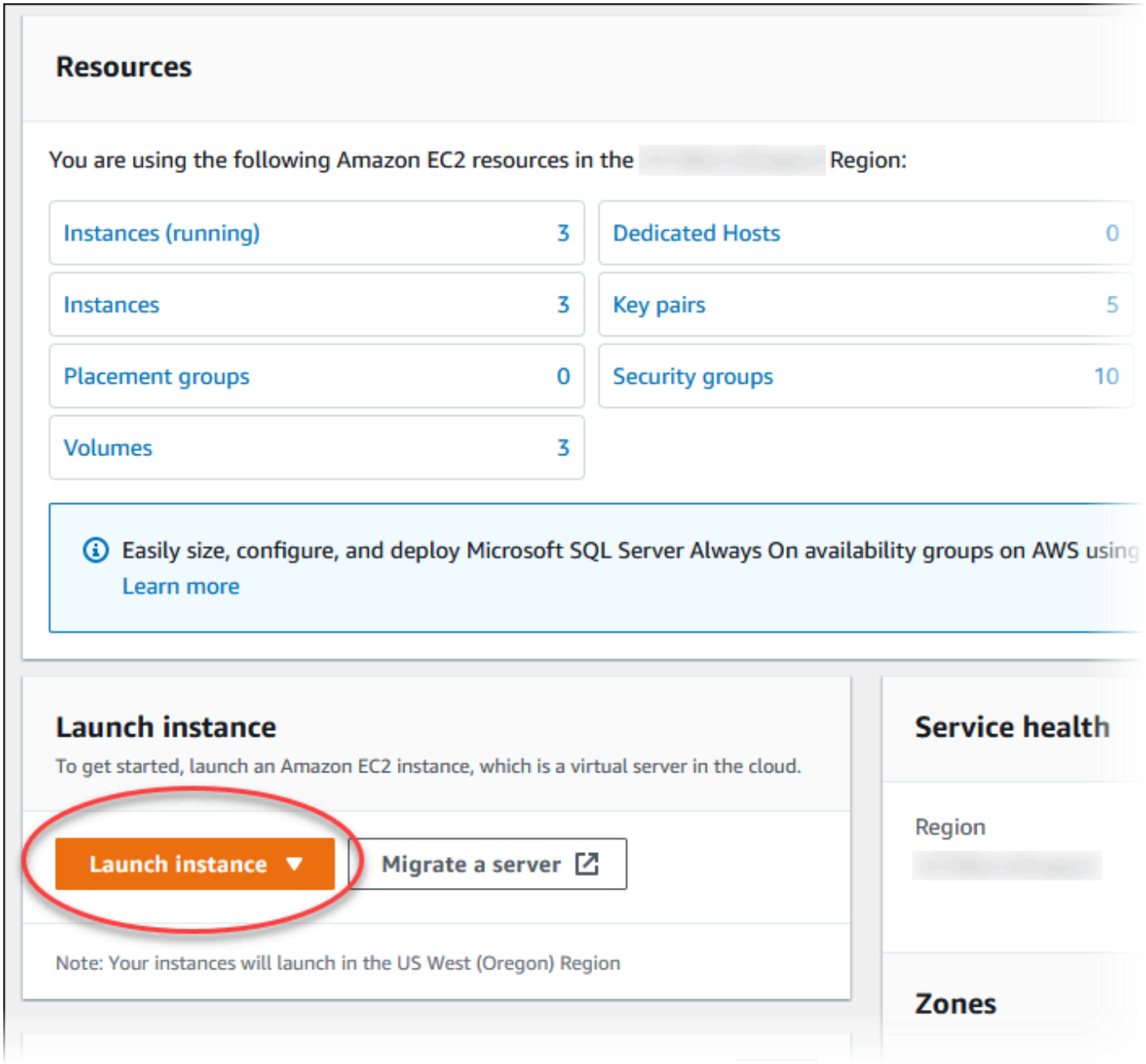
- Mendaftar Akun AWS
- Membuat pengguna administratif

Langkah 1: Buat instans EC2

Buat instans Amazon EC2 yang akan Anda gunakan untuk menghubungkan ke basis data Anda.

Untuk membuat instans EC2

1. Masuk ke AWS Management Console dan buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di sudut kanan atas AWS Management Console, pilih yang Wilayah AWS Anda gunakan untuk database sebelumnya.
3. Pilih Dasbor EC2, lalu pilih Luncurkan instans seperti yang ditampilkan dalam gambar berikut.

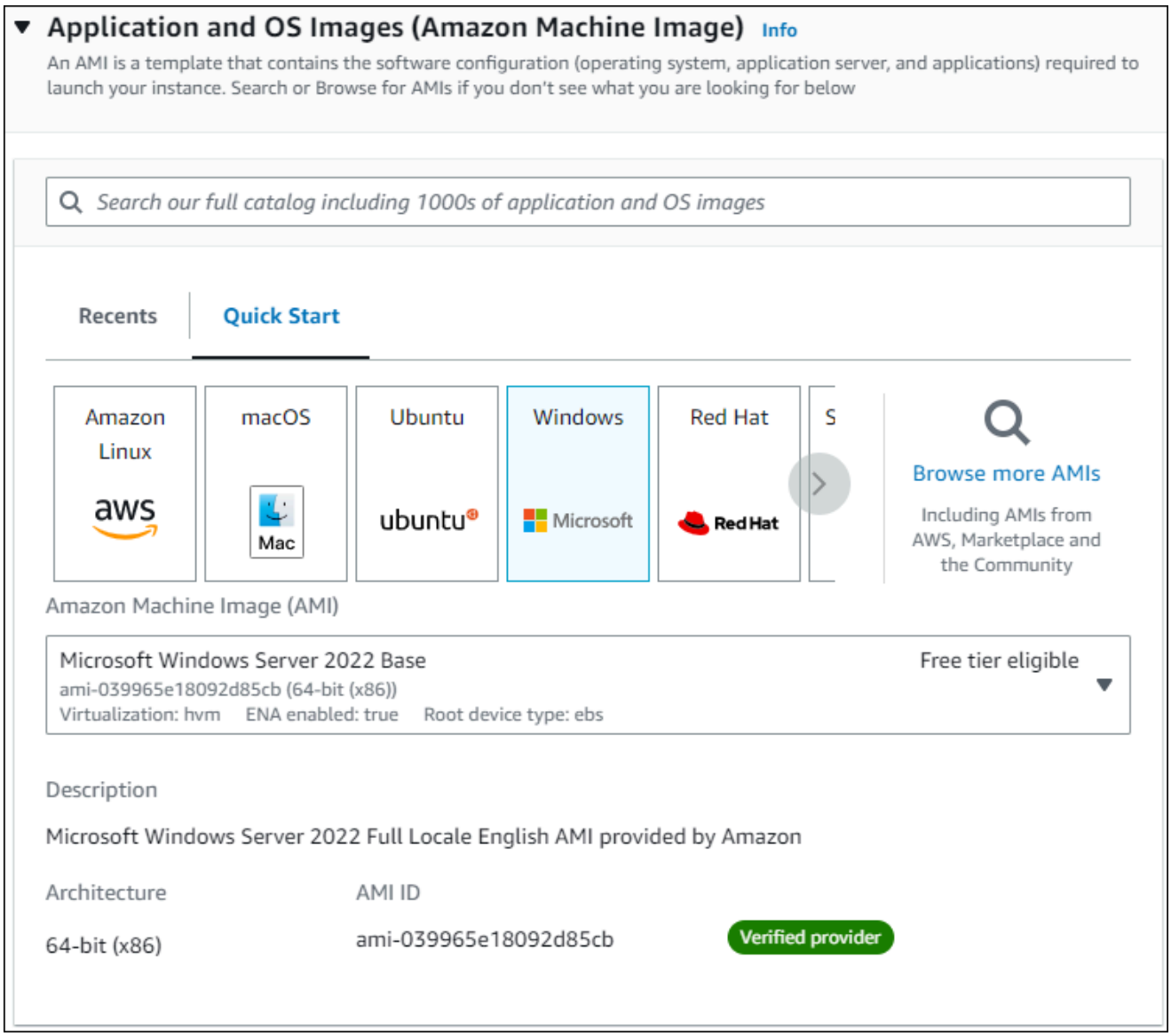


Halaman Meluncurkan instans akan terbuka.

4. Pilih pengaturan berikut di halaman Meluncurkan instans.

   a. Di bagian Nama dan tag, untuk Nama, masukkan **`ec2-database-connect`**.

   b. Pada Gambar Aplikasi dan OS (Amazon Machine Image), pilih Windows, lalu pilih Microsoft Windows Server 2022 Base. Biarkan default untuk pilihan lainnya.



c. Di bagian Jenis instans, pilih t2.micro.

d. Di bagian Pasangan kunci (login), pilih nama Pasangan kunci untuk menggunakan pasangan kunci yang ada. Untuk membuat pasangan kunci baru untuk instans Amazon EC2, pilih Buat Pasangan kunci baru lalu gunakan jendela Buat pasangan kunci untuk membuatnya.

   Untuk informasi selengkapnya tentang membuat pasangan kunci baru, lihat Membuat pasangan kunci di Panduan Pengguna Amazon EC2 untuk Instans Windows.

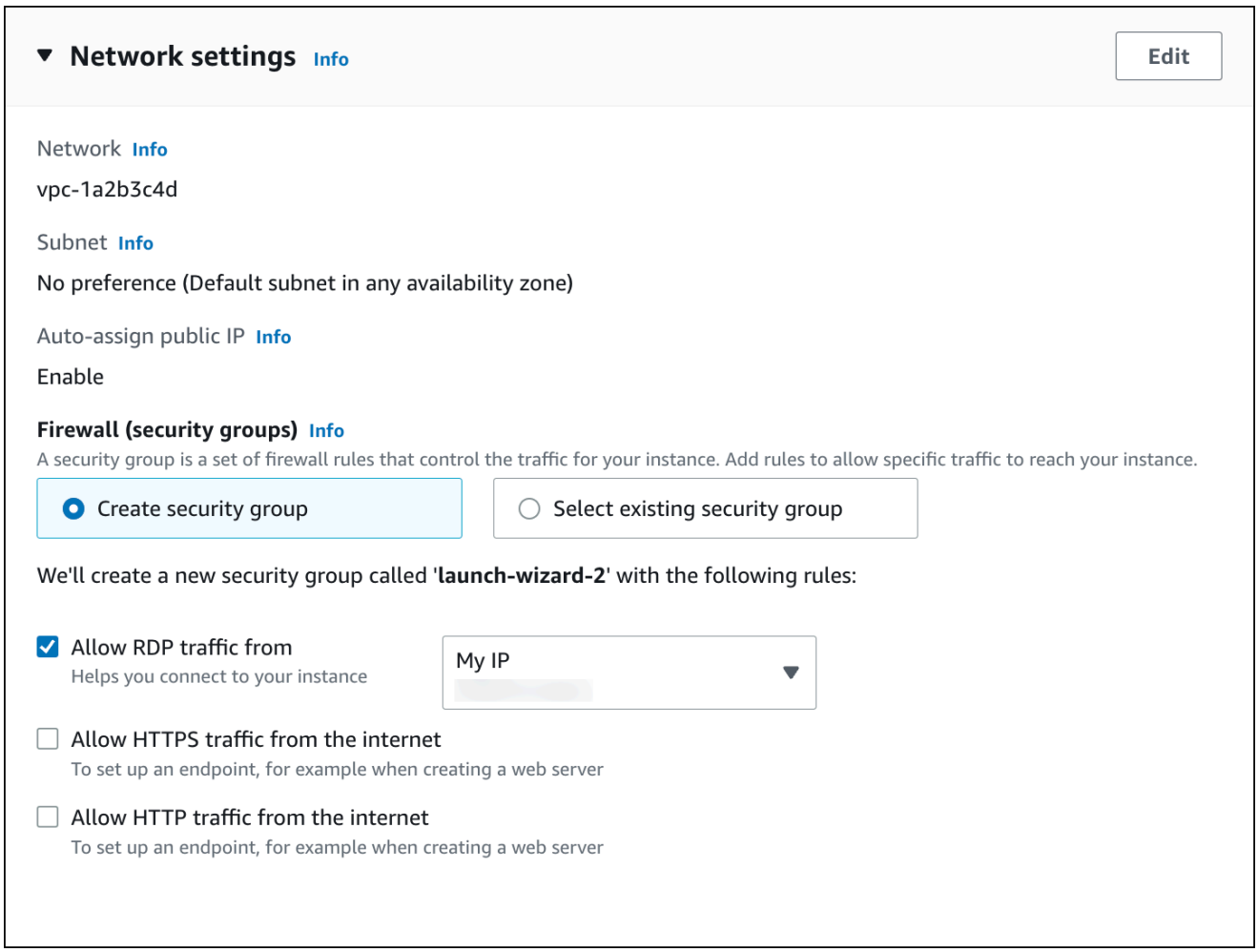
e. Untuk Firewall (grup keamanan) di pengaturan Jaringan, pilih Izinkan lalu lintas RDP dari untuk menghubungkan ke instans EC2.

Anda dapat memilih IP Saya jika alamat IP yang ditampilkan benar untuk koneksi RDP. Jika tidak, Anda dapat menentukan alamat IP yang akan digunakan untuk menghubungkan ke instans EC2 di VPC Anda menggunakan RDP. Untuk menentukan alamat IP publik Anda, Anda dapat membuka layanan di https://checkip.amazonaws.com di jendela atau tab browser lain. Contoh alamat IP adalah 192.0.2.1/32.

Dalam banyak kasus, Anda dapat menghubungkan melalui penyedia layanan Internet (ISP) atau dari belakang firewall Anda tanpa alamat IP statis. Jika demikian, tentukan rentang alamat IP yang digunakan oleh komputer klien.

> **Warning**
>
> Jika menggunakan `0.0.0.0/0` untuk akses RDP, Anda memungkinkan semua alamat IP untuk mengakses instans EC2 publik Anda menggunakan RDP. Hal ini dapat diterima untuk waktu yang singkat di lingkungan pengujian, tetapi tidak aman untuk lingkungan produksi. Dalam produksi, Anda hanya dapat memberikan otorisasi pada alamat IP atau rentang alamat tertentu saja untuk mengakses instans EC2 Anda menggunakan RDP.

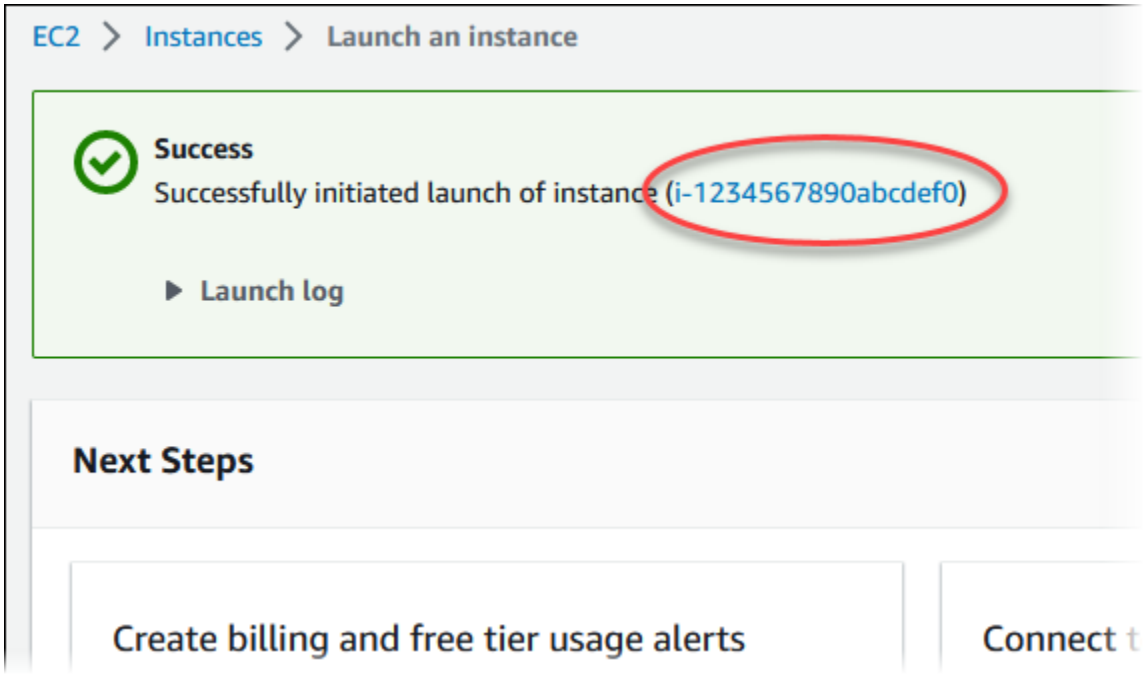
Gambar berikut menunjukkan contoh bagian Pengaturan jaringan.



f. Biarkan nilai default untuk bagian yang lainnya.

g. Tinjau ringkasan konfigurasi instans EC2 Anda di panel Ringkasan, dan setelah Anda siap, pilih Luncurkan instans.

5. Di halaman Status Peluncuran, catat pengidentifikasi untuk instans EC2 baru Anda, misalnya: `i-1234567890abcdef0`.



6. Pilih pengidentifikasi instans EC2 untuk membuka daftar instans EC2.
7. Tunggu hingga Status instance untuk instans EC2 Anda berstatus Berjalan sebelum melanjutkan.

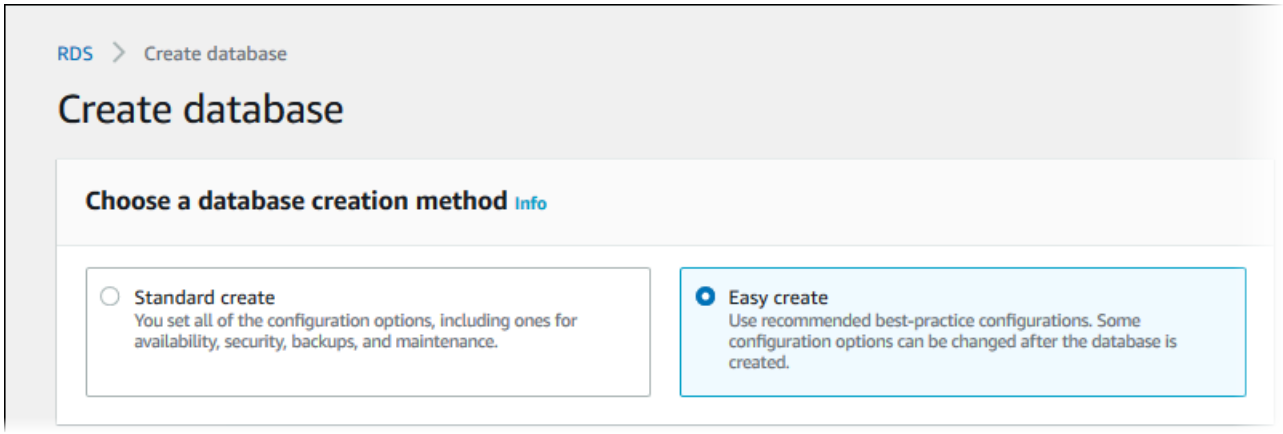
Langkah 2: Buat instans DB SQL Server

Blok bangunan dasar Amazon RDS adalah instans DB. Lingkungan ini adalah tempat Anda menjalankan basis data SQL Server Anda.

Dalam contoh ini, Anda menggunakan Pembuatan Mudah untuk membuat instans DB yang menjalankan mesin basis data SQL Server dengan kelas instans DB db.t2.micro.

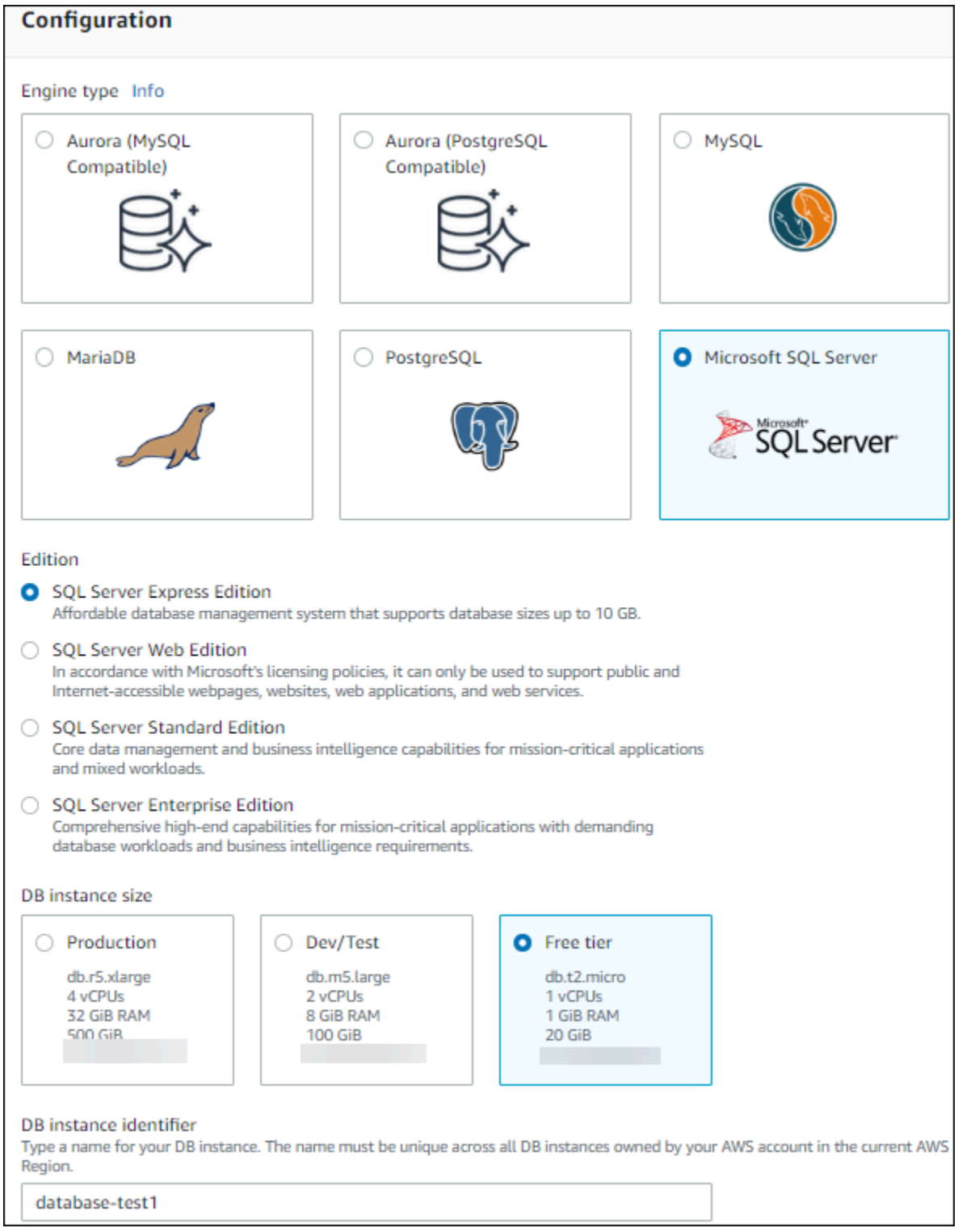
Untuk membuat instans DB Microsoft SQL Server dengan Pembuatan Mudah

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di sudut kanan atas konsol Amazon RDS, pilih Wilayah AWS di mana Anda ingin membuat instans DB.
3. Di panel navigasi, pilih Basis Data.
4. Pilih Buat basis data dan pastikan Pembuatan Mudah dipilih.



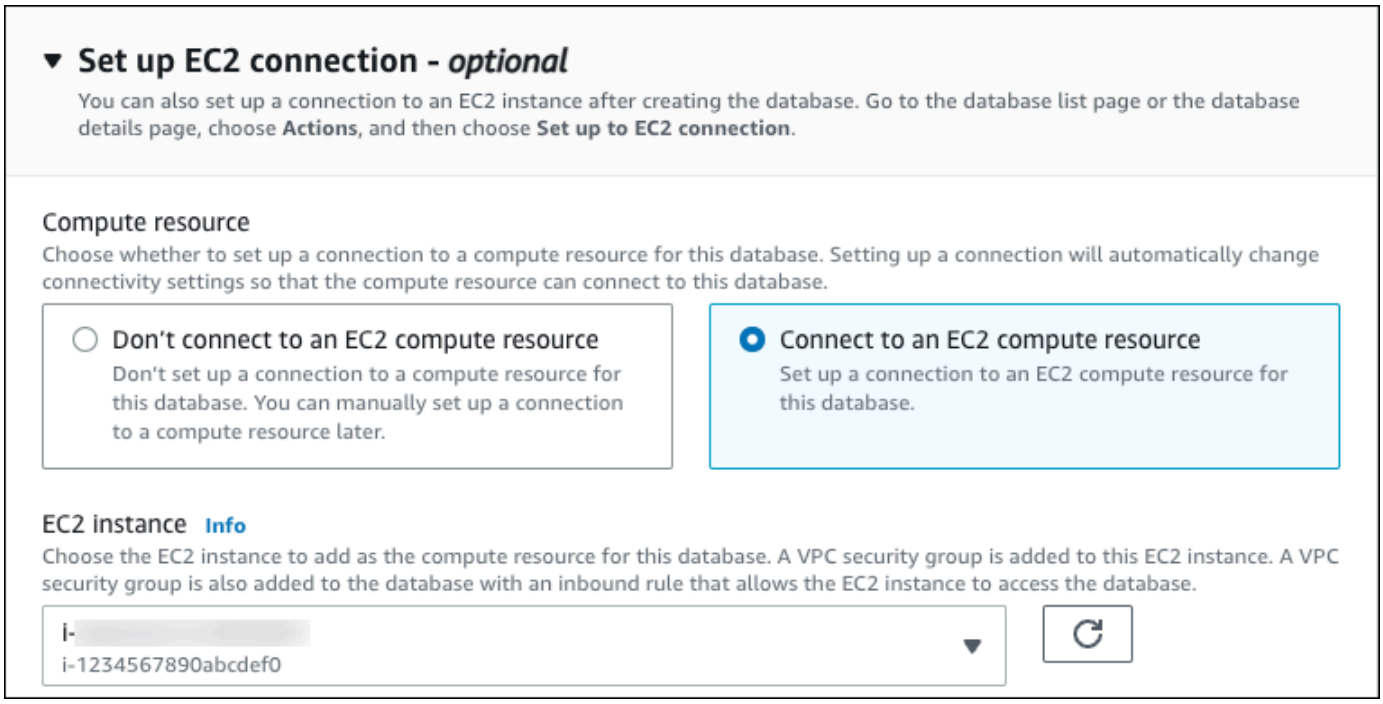
5. Di Konfigurasi, pilih Microsoft SQL Server.
6. Untuk Edisi, pilih SQL Server Express Edition.
7. Untuk Ukuran instans DB, pilih Tingkat gratis.
8. Untuk Pengidentifikasi instans DB, masukkan **`database-test1`**.

   Tampilan halaman Membuat basis data seperti gambar berikut.



9. Untuk Nama pengguna utama, masukkan nama untuk pengguna utama, atau tetap gunakan nama default.
10. Untuk menyiapkan koneksi dengan instans EC2 yang Anda buat sebelumnya, buka Menyiapkan koneksi EC2 - opsional.

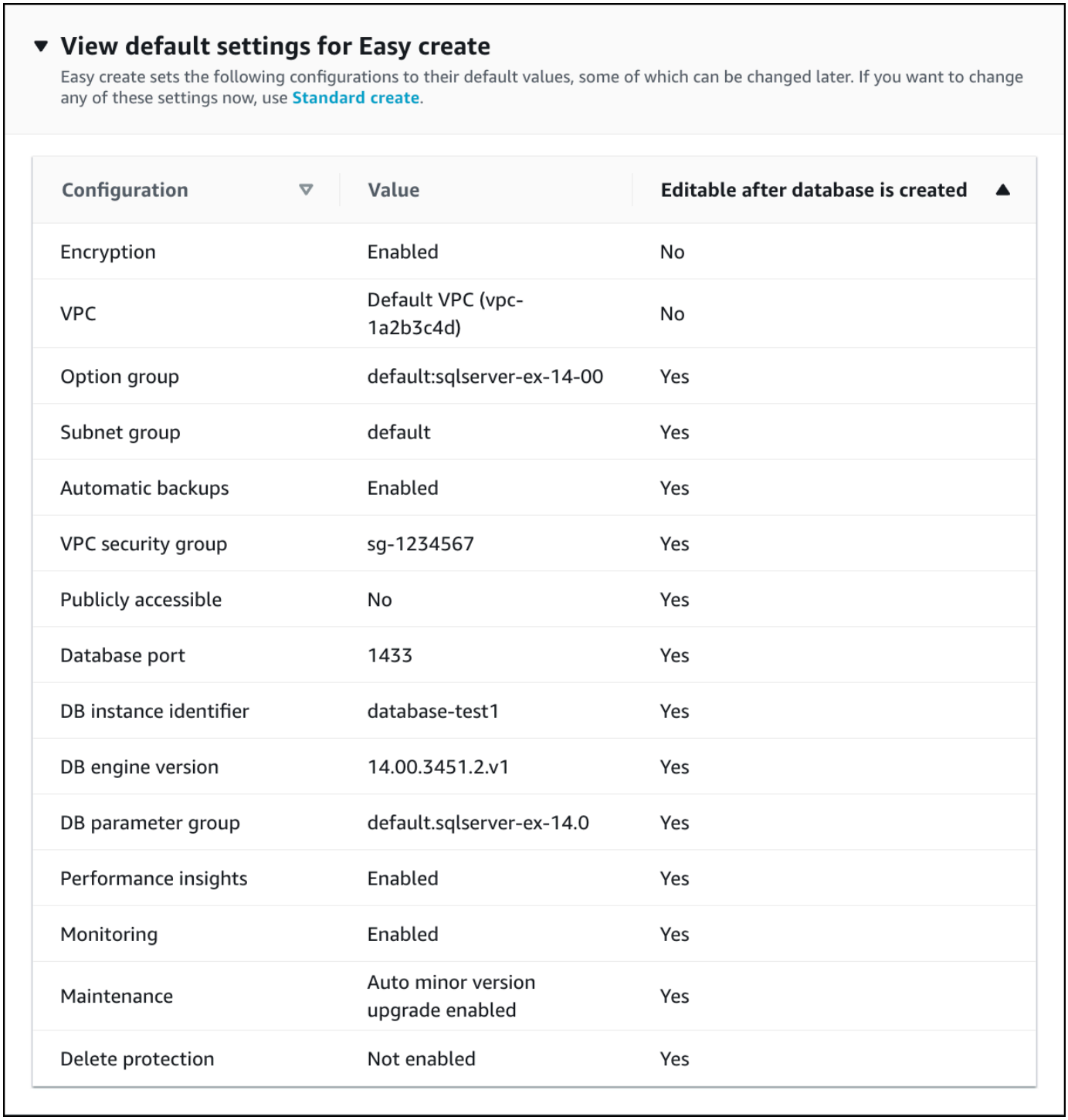
Pilih Hubungkan ke sumber daya komputasi EC2. Pilih instans EC2 yang Anda buat sebelumnya.



11. Untuk menggunakan kata sandi utama yang dibuat secara otomatis untuk instans DB, pilih kotak Buat kata sandi secara otomatis.

    Untuk memasukkan kata sandi utama Anda, hapus centang pada Buat kata sandi secara otomatis, lalu masukkan kata sandi yang sama dalam Kata sandi utama dan Konfirmasi kata sandi.

12. Buka Lihat pengaturan default untuk Pembuatan Mudah.



View default settings for Easy create

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

Configuration	Value	Editable after database is created
Encryption	Enabled	No
VPC	Default VPC (vpc-1a2b3c4d)	No
Option group	default:sqlserver-ex-14-00	Yes
Subnet group	default	Yes
Automatic backups	Enabled	Yes
VPC security group	sg-1234567	Yes
Publicly accessible	No	Yes
Database port	1433	Yes
DB instance identifier	database-test1	Yes
DB engine version	14.00.3451.2.v1	Yes
DB parameter group	default.sqlserver-ex-14.0	Yes
Performance insights	Enabled	Yes
Monitoring	Enabled	Yes
Maintenance	Auto minor version upgrade enabled	Yes
Delete protection	Not enabled	Yes

Anda dapat memeriksa pengaturan default yang digunakan dengan Pembuatan mudah. Kolom Dapat diedit setelah basis data dibuat menunjukkan opsi yang dapat Anda ubah setelah membuat basis data.

- Jika pengaturan memiliki Tidak di kolom tersebut, dan Anda menginginkan pengaturan yang berbeda, Anda dapat menggunakan Pembuatan Standar untuk membuat instans DB.

- Jika pengaturan memiliki Ya di kolom tersebut, dan Anda menginginkan pengaturan yang berbeda, Anda dapat menggunakan Pembuatan Standar untuk membuat instans DB, atau mengubah instans DB setelah Anda membuatnya untuk mengubah pengaturan.

13. Pilih Buat basis data.

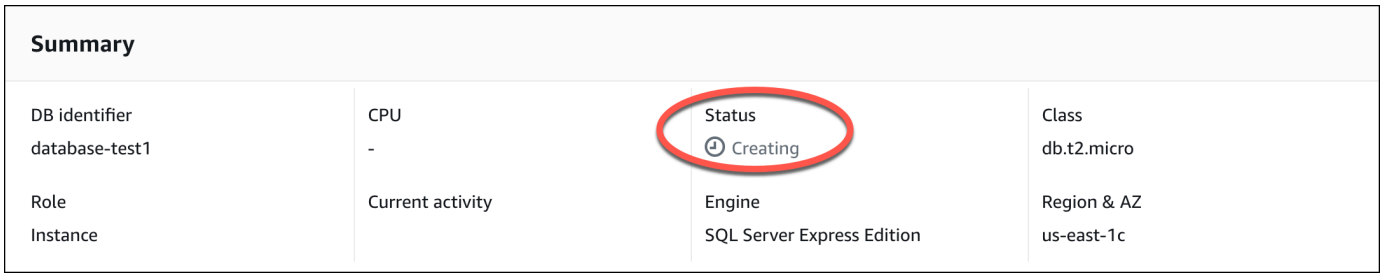
    Untuk melihat nama pengguna dan kata sandi utama untuk instans DB, pilih Lihat detail kredensial.

    Anda dapat menggunakan nama pengguna dan kata sandi yang ditampilkan untuk terhubung ke instans DB sebagai pengguna utama.

    > **Important**
    >
    > Anda tidak dapat melihat kata sandi pengguna utama lagi. Jika tidak mencatatnya, Anda mungkin harus mengubahnya.
    > Jika perlu mengubah kata sandi pengguna utama setelah instans DB tersedia, Anda dapat mengubah instans DB untuk melakukannya. Untuk informasi selengkapnya tentang cara mengubah instans DB, lihat Memodifikasi instans DB Amazon RDS.

14. Dalam daftar Basis data, pilih nama instans DB SQL Server baru untuk menampilkan detailnya.

    Instans DB memiliki status Membuat hingga siap digunakan.



    Saat statusnya berubah menjadi Tersedia, Anda dapat terhubung ke instans DB. Tergantung pada kelas instans DB dan jumlah penyimpanan, diperlukan waktu hingga 20 menit sebelum instans baru tersedia.

(Opsional) Buat instance VPC, EC2, dan SQL Server menggunakan AWS CloudFormation

Alih-alih menggunakan konsol untuk membuat instance VPC, EC2, dan SQL Server, Anda dapat menggunakannya AWS CloudFormation untuk menyediakan AWS sumber daya dengan memperlakukan infrastruktur sebagai kode. Untuk membantu Anda mengatur AWS sumber daya Anda menjadi unit yang lebih kecil dan lebih mudah dikelola, Anda dapat menggunakan fungsionalitas tumpukan AWS CloudFormation bersarang. Untuk informasi selengkapnya, lihat Membuat tumpukan di AWS CloudFormation konsol dan Bekerja dengan tumpukan bersarang..

> **Important**
>
> AWS CloudFormation gratis, tetapi sumber daya yang CloudFormation menciptakan hidup. Anda dikenakan biaya penggunaan standar untuk sumber daya ini sampai Anda menghentikannya. Total biaya akan minimal. Untuk informasi tentang cara meminimalkan biaya apa pun, buka Tingkat AWS Gratis.

Untuk membuat sumber daya Anda menggunakan AWS CloudFormation konsol, selesaikan langkah-langkah berikut:

- Langkah 1: Unduh CloudFormation template
- Langkah 2: Konfigurasikan sumber daya Anda menggunakan CloudFormation

Unduh CloudFormation template

CloudFormation Template adalah file teks JSON atau YAMAL yang berisi informasi konfigurasi tentang sumber daya yang ingin Anda buat di tumpukan. Template ini juga membuat VPC dan host bastion untuk Anda bersama dengan instance RDS.

Untuk mengunduh file template, buka tautan berikut, CloudFormation template SQL Server.

Di halaman Github, klik tombol Unduh file mentah untuk menyimpan file YAMAL template.

Konfigurasikan sumber daya Anda menggunakan CloudFormation

> **Note**
>
> Sebelum memulai proses ini, pastikan Anda memiliki pasangan Kunci untuk instans EC2 di. Akun AWS Untuk informasi selengkapnya, lihat Pasangan kunci Amazon EC2 dan instans Linux.

Ketika Anda menggunakan AWS CloudFormation template, Anda harus memilih parameter yang benar untuk memastikan sumber daya Anda dibuat dengan benar. Ikuti langkah-langkah di bawah ini:

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di https://console.aws.amazon.com/cloudformation.
2. Pilih Buat tumpukan.
3. Di bagian Tentukan templat, pilih Unggah file templat dari komputer Anda, lalu pilih Berikutnya.
4. Di halaman Tentukan detail tumpukan, atur parameter berikut:
   a. Tetapkan nama Stack ke SQL ServerTestStack.
   b. Di bawah Parameter, atur Availability Zone dengan memilih tiga zona ketersediaan.
   c. Di bawah konfigurasi Linux Bastion Host, untuk Key Name, pilih key pair untuk login ke instans EC2 Anda.
   d. Dalam pengaturan konfigurasi Linux Bastion Host, atur rentang IP yang Diizinkan ke alamat IP Anda. Untuk terhubung ke instans EC2 di VPC Anda menggunakan Secure Shell (SSH), tentukan alamat IP publik Anda menggunakan layanan di https://checkip.amazonaws.com. Contoh alamat IP adalah 192.0.2.1/32.

      > **Warning**
      >
      > Jika menggunakan `0.0.0.0/0` untuk akses SSH, Anda memungkinkan semua alamat IP untuk mengakses instans publik EC2 Anda menggunakan SSH. Hal ini dapat diterima untuk waktu yang singkat di lingkungan pengujian, tetapi tidak aman untuk lingkungan produksi. Dalam produksi, Anda hanya dapat memberikan otorisasi pada alamat IP atau rentang alamat tertentu saja untuk mengakses instans EC2 Anda menggunakan SSH.

   e. Di bawah konfigurasi Database General, atur kelas instance Database ke db.t3.micro.

f. Tetapkan nama Database ke**`database-test1`**.

g. Untuk nama pengguna master Database, masukkan nama untuk pengguna master.

h. Atur Kelola kata sandi pengguna master DB dengan Secrets Manager `false` untuk tutorial ini.

i. Untuk kata sandi Database, tetapkan kata sandi pilihan Anda. Ingat kata sandi ini untuk langkah lebih lanjut dalam tutorial.

j. Di bawah konfigurasi Penyimpanan Database, atur tipe penyimpanan Database ke gp2.

k. Di bawah konfigurasi Pemantauan Database, atur Aktifkan Performance Insights RDS ke false.

l. Biarkan semua pengaturan lainnya sebagai nilai default. Klik Berikutnya untuk melanjutkan.

5. Di halaman Configure stack options, tinggalkan semua opsi default. Klik Berikutnya untuk melanjutkan.
6. Di halaman tumpukan Tinjauan, pilih Kirim setelah memeriksa database dan opsi host bastion Linux.

Setelah proses pembuatan tumpukan selesai, lihat tumpukan dengan nama BastionStackdan RDSNS untuk mencatat informasi yang Anda butuhkan untuk terhubung ke database. Untuk informasi selengkapnya, lihat Melihat data AWS CloudFormation tumpukan dan sumber daya di AWS Management Console.

Langkah 3: Hubungkan ke instans DB SQL Server

Dalam prosedur berikut, Anda akan menghubungkan ke instans DB menggunakan Microsoft SQL Server Management Studio (SSMS).

Untuk menghubungkan ke instans DB RDS for SQL Server menggunakan SSMS

1. Temukan titik akhir (nama DNS) dan nomor port untuk instans DB Anda.

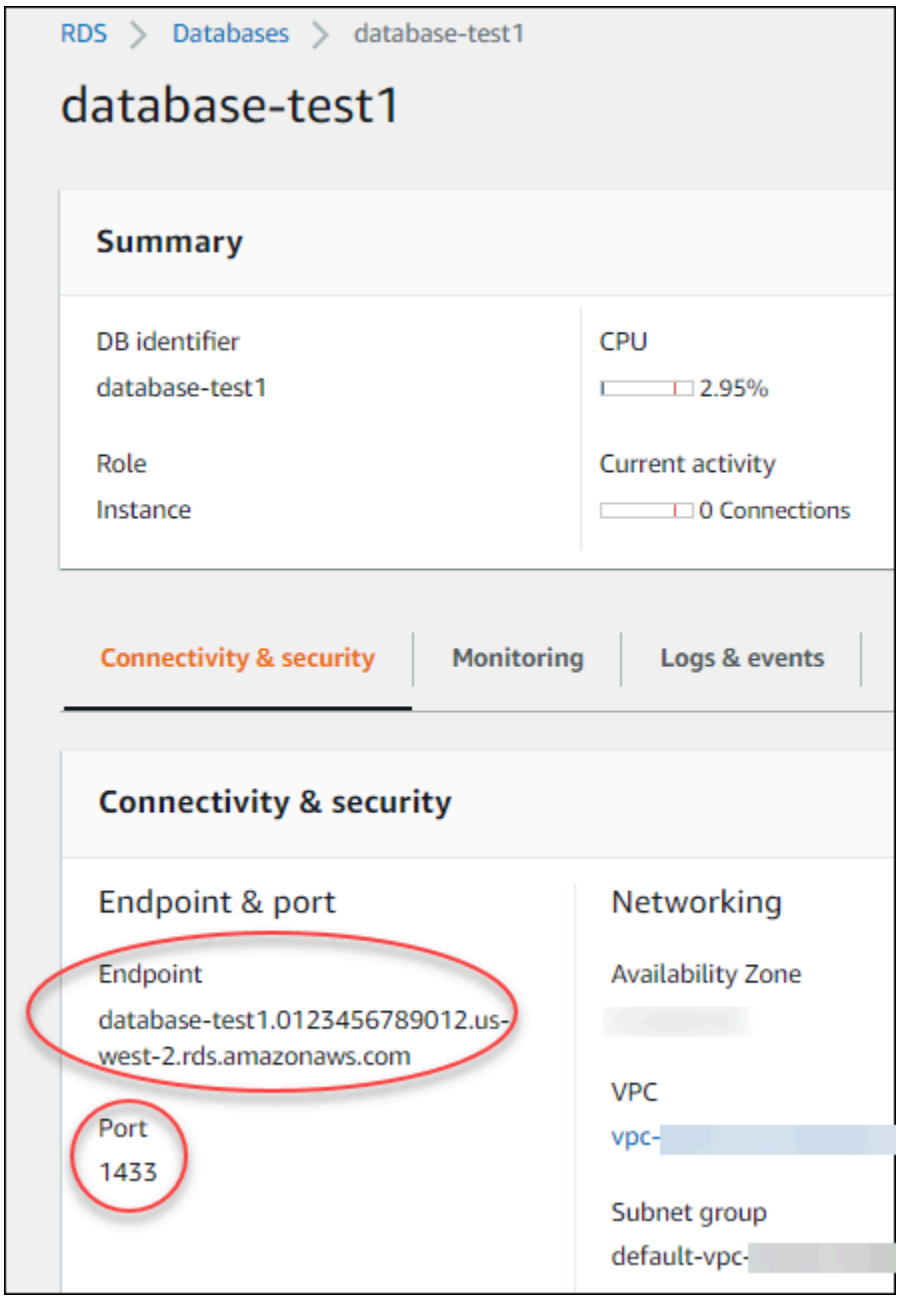
   a. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.

   b. Di sudut kanan atas konsol Amazon RDS, pilih instans DB Wilayah AWS .

   c. Di panel navigasi, pilih Basis Data.

   d. Pilih nama instans DB SQL Server untuk menampilkan detailnya.

   e. Di tab Konektivitas, salin titik akhir. Perhatikan juga nomor port. Anda memerlukan titik akhir dan nomor port untuk terhubung ke instans DB.



2. Hubungkan ke instans EC2 yang Anda buat sebelumnya dengan mengikuti langkah-langkah di Menghubungkan ke instans Microsoft Windows di Panduan Pengguna Amazon EC2 untuk Instans Windows.
3. Instal klien SQL Server Management Studio (SSMS) dari Microsoft.

   Untuk mengunduh versi mandiri SSMS ke instans EC2 Anda, lihat Mengunduh SQL Server Management Studio (SSMS) dalam dokumentasi Microsoft.

   a. Gunakan menu Start untuk membuka Internet Explorer.

b. Gunakan Internet Explorer untuk mengunduh dan menginstal versi SSMS mandiri. Jika ada pemberitahuan bahwa situs tersebut tidak dipercaya, tambahkan situs tersebut ke daftar situs tepercaya.

4. Mulai SQL Server Management Studio (SSMS).

   Kotak dialog Hubungkan ke Server ditampilkan.

5. Masukkan informasi berikut untuk instans DB sampel Anda:

   a. Untuk Jenis server, pilih Mesin Basis Data.

   b. Untuk Nama server, masukkan nama DNS, diikuti dengan koma dan nomor port (port default adalah 1433). Misalnya, nama server Anda akan terlihat seperti berikut:

   ```
   database-test1.0123456789012.us-west-2.rds.amazonaws.com,1433
   ```

   c. Untuk Autentikasi, pilih Autentikasi SQL Server.

   d. Untuk Login, masukkan nama pengguna yang Anda pilih sebagai instans DB sampel. Ini juga dikenal sebagai nama pengguna utama.

   e. Untuk Kata Sandi, masukkan kata sandi yang Anda pilih sebelumnya untuk instans DB sampel. Ini juga dikenal sebagai kata sandi pengguna utama.

6. Pilih Hubungkan.

   Setelah beberapa saat, SSMS akan terhubung ke instans DB Anda. Untuk keamanan, praktik terbaiknya adalah menggunakan koneksi terenkripsi. Hanya gunakan koneksi SQL Server yang tidak terenkripsi saat klien dan server berada di VPC yang sama dan jaringan tepercaya. Untuk informasi tentang cara menggunakan koneksi terenkripsi, lihat Menggunakan SSL dengan instans DB Microsoft SQL Server.

Untuk informasi selengkapnya tentang menghubungkan ke instans DB Microsoft SQL Server, lihat Menghubungkan ke instans DB yang menjalankan mesin basis data Microsoft SQL Server.

Untuk informasi tentang masalah koneksi, lihat Tidak dapat terhubung ke instans DB Amazon RDS.

Langkah 4: Jelajahi instans DB SQL Server sampel

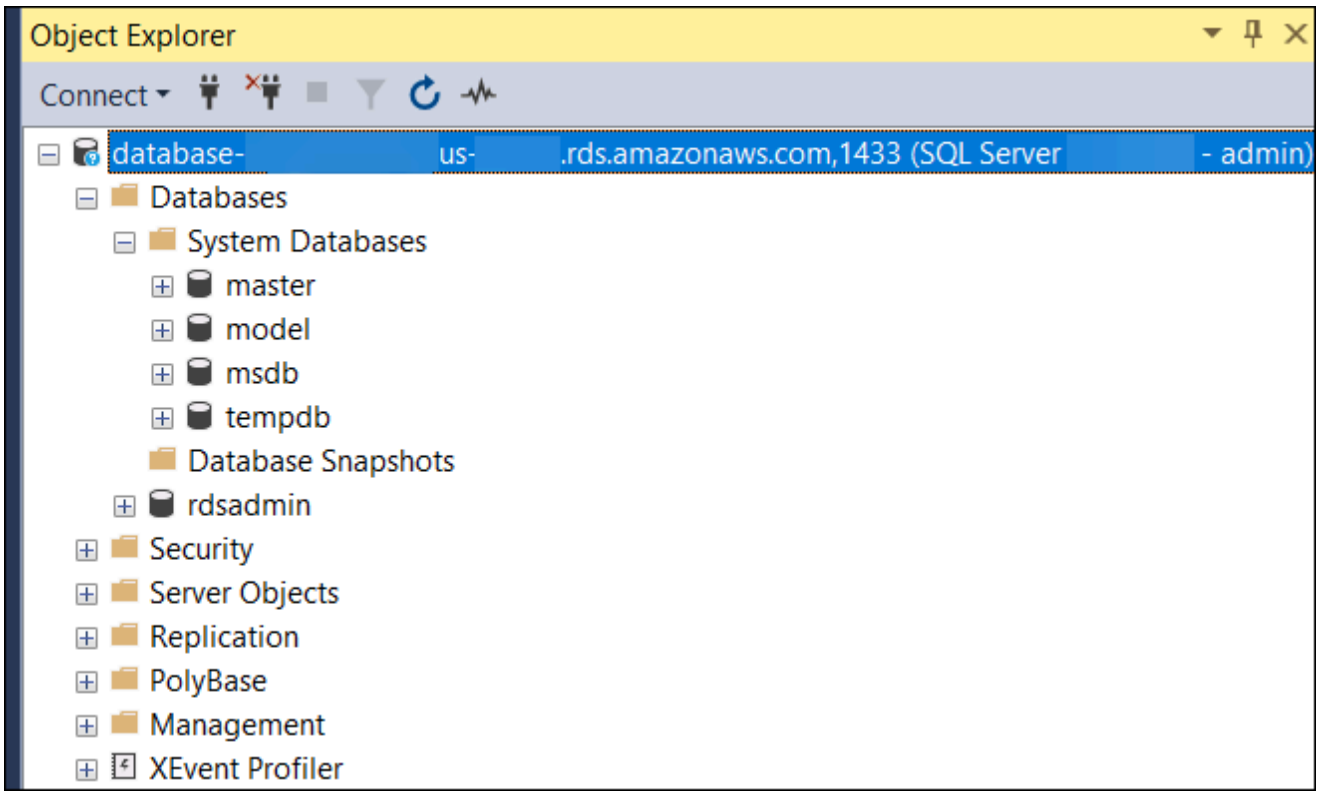
Anda dapat menjelajahi instans DB sampel menggunakan Microsoft SQL Server Management Studio (SSMS).

Untuk menjelajahi instans DB menggunakan SSMS

1. Instans DB SQL Server Anda dilengkapi dengan basis data sistem bawaan standar SQL Server (master, model, msdb, dan tempdb). Untuk menjelajahi basis data sistem, lakukan tindakan berikut:

   a. Di SSMS, pada menu Lihat, pilih Object Explorer.

   b. Perluas instans DB Anda, perluas Basis Data, lalu perluas Basis Data Sistem seperti yang ditunjukkan.



   Instans DB SQL Server Anda juga dilengkapi dengan basis data bernama `rdsadmin`. Amazon RDS menggunakan basis data ini untuk menyimpan objek yang digunakan untuk mengelola basis data Anda. Basis data `rdsadmin` juga mencakup prosedur tersimpan yang dapat Anda jalankan untuk melakukan tugas tingkat lanjut.

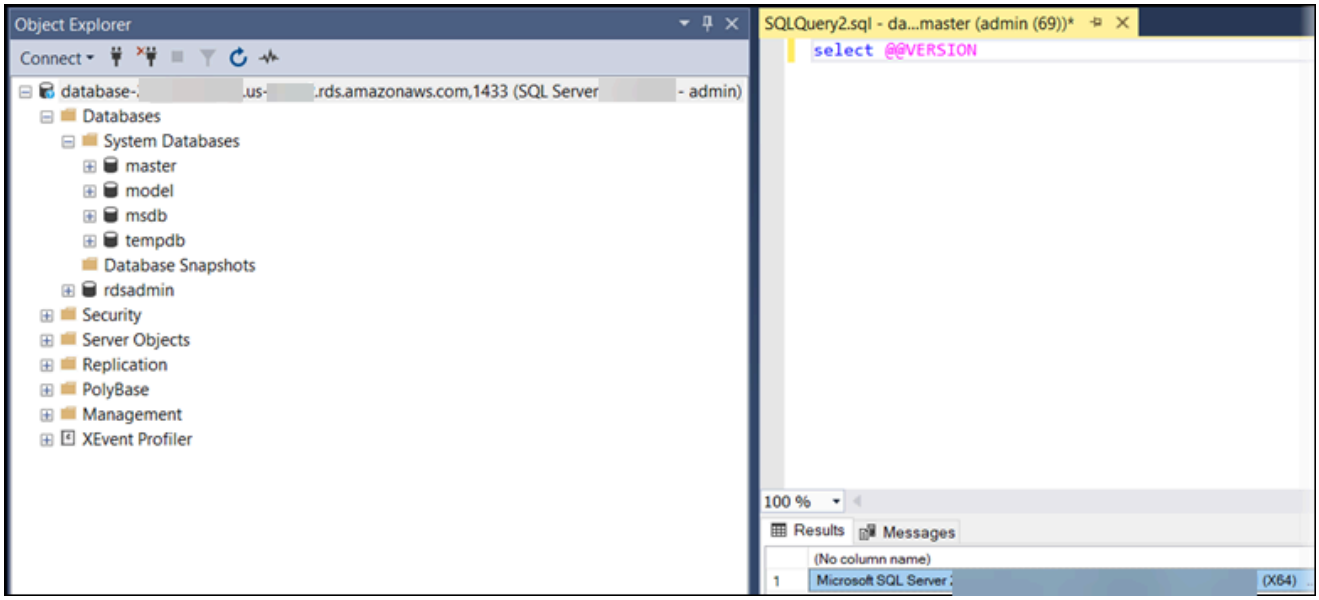
2. Mulai buat basis data Anda sendiri dan jalankan kueri terhadap instans DB dan basis data Anda seperti biasa. Untuk menjalankan kueri uji pada instans DB Anda, lakukan tindakan berikut:

   a. Di SSMS, pada menu File, arahkan ke Baru, lalu pilih Kueri dengan Koneksi Saat Ini.

   b. Masukkan kueri SQL berikut:

```
select @@VERSION
```

c. Jalankan kueri. SSMS mengembalikan versi SQL Server dari instans DB Amazon RDS Anda.



Langkah 5: Hapus instans EC2 dan instans DB

Setelah Anda terhubung ke dan menjelajahi instans EC2 dan instans DB sampel yang Anda buat, hapus instans tersebut sehingga Anda tidak lagi dikenakan biaya untuk instans DB tersebut.

Jika Anda biasa AWS CloudFormation membuat sumber daya, lewati langkah ini dan lanjutkan ke langkah berikutnya.

Untuk menghapus instans EC2

1. Masuk ke AWS Management Console dan buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Instans.
3. Pilih instans EC2, dan pilih Status instans, Akhiri instans.
4. Pilih Akhiri saat diminta untuk konfirmasi.

Untuk informasi selengkapnya tentang menghapus instans EC2, lihat Mengakhiri Instans dalam Panduan Pengguna untuk Instans Windows.

Untuk menghapus instans DB tanpa snapshot DB akhir

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data.
3. Pilih instans DB yang ingin Anda hapus.
4. Untuk Tindakan, pilih Hapus.
5. Hapus Buat snapshot akhir? dan Pertahankan pencadangan otomatis.
6. Lengkapi pengakuan dan pilih Hapus.

(Opsional) Hapus instans EC2 dan instans DB yang dibuat dengan CloudFormation

Jika Anda biasa AWS CloudFormation membuat sumber daya, hapus CloudFormation tumpukan setelah Anda terhubung dan jelajahi contoh instans EC2 dan instans DB, sehingga Anda tidak lagi dikenakan biaya untuk itu.

Untuk menghapus sumber CloudFormation daya

1. Buka AWS CloudFormation konsol.
2. Pada halaman Stacks di CloudFormationconsole, pilih tumpukan root (tumpukan tanpa nama VPCStack, BastionStack atau RDSNS).
3. Pilih Hapus.
4. Pilih Hapus tumpukan saat diminta konfirmasi.

Untuk informasi selengkapnya tentang menghapus tumpukan CloudFormation, lihat Menghapus tumpukan di AWS CloudFormation konsol di AWS CloudFormation Panduan Pengguna.

(Opsional) Menghubungkan instans DB Anda ke fungsi Lambda

Anda juga dapat menghubungkan instans DB RDS for SQL Server ke sumber daya komputasi nirserver Lambda. Fungsi Lambda memungkinkan Anda menjalankan kode tanpa menyediakan atau mengelola infrastruktur. Fungsi Lambda juga memungkinkan Anda untuk otomatis merespons permintaan eksekusi kode pada skala apa pun, mulai dari selusin peristiwa dalam sehari hingga ratusan per detik. Lihat informasi yang lebih lengkap di Menghubungkan secara otomatis fungsi Lambda dan instans basis data.

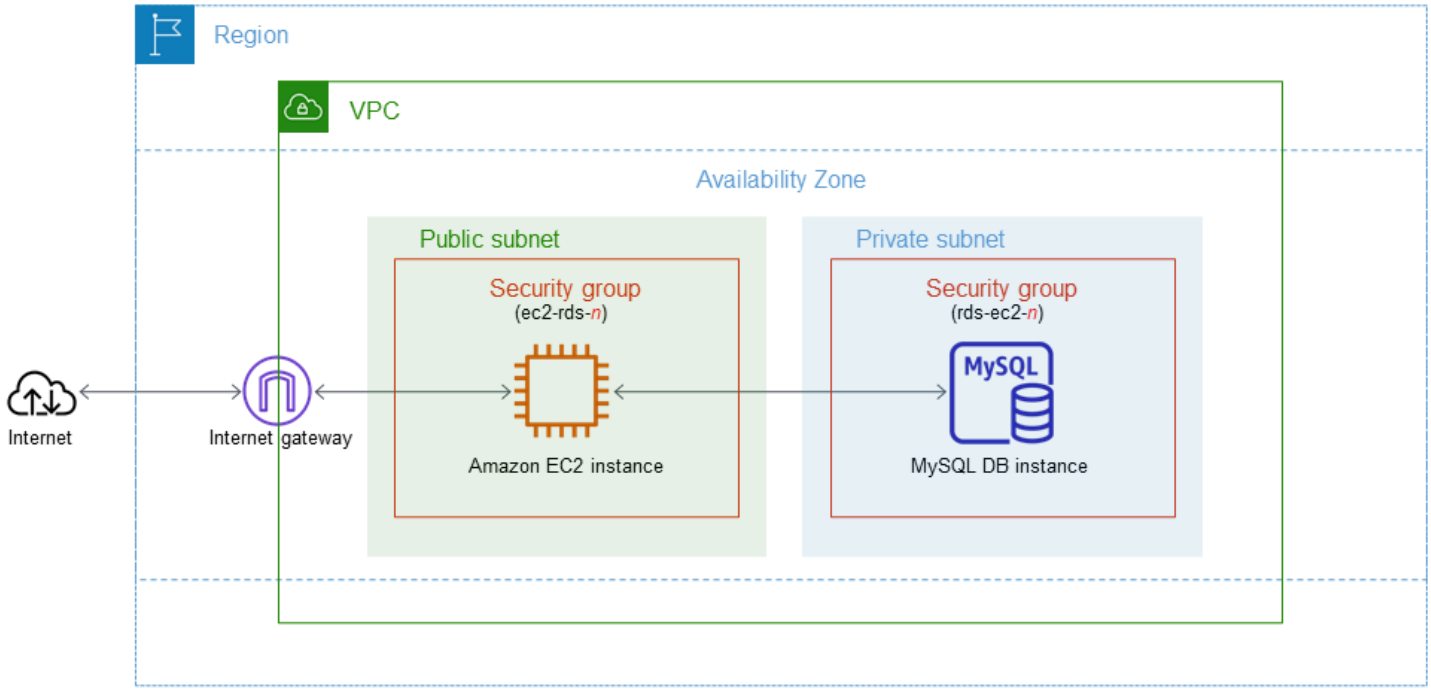
Membuat dan menghubungkan ke instans DB MySQL

Tutorial ini membuat instans EC2 dan instans DB RDS for MySQL. Tutorial ini menunjukkan cara mengakses instans DB dari instans EC2 menggunakan klien MySQL standar. Sebagai praktik terbaik, tutorial ini membuat instans DB privat dalam cloud privat virtual (VPC). Dalam kebanyakan kasus, sumber daya lain dalam VPC yang sama, seperti instans EC2, dapat mengakses instans DB, tetapi sumber daya di luar VPC tidak dapat mengaksesnya.

Setelah Anda menyelesaikan tutorial, ada subnet publik dan privat di setiap Zona Ketersediaan di VPC Anda. Dalam satu Zona Ketersediaan, instans EC2 berada di subnet publik, dan instans DB berada di subnet privat.

> **Important**
>
> Tidak ada biaya untuk membuat AWS akun. Namun, dengan menyelesaikan tutorial ini, Anda mungkin dikenakan biaya untuk AWS sumber daya yang Anda gunakan. Anda dapat menghapus sumber daya ini setelah menyelesaikan tutorial jika tidak diperlukan lagi.

Diagram berikut menunjukkan konfigurasi setelah tutorial selesai.



Tutorial ini memungkinkan Anda untuk membuat sumber daya Anda dengan menggunakan salah satu metode berikut:

1. Gunakan AWS Management Console - Langkah 2: Buat instans DB MySQL dan Langkah 1: Buat instans EC2
2. Gunakan AWS CloudFormation untuk membuat instance database dan instans EC2 - (Opsional) Buat instance VPC, EC2, dan MySQL menggunakan AWS CloudFormation

Metode pertama menggunakan Easy create untuk membuat instance MySQL DB pribadi dengan. AWS Management Console Di sini, Anda hanya menentukan jenis mesin DB, ukuran instans DB, dan pengidentifikasi instans DB. Pembuatan Mudah menggunakan pengaturan default untuk opsi konfigurasi lainnya.

Saat Anda menggunakan Standard create sebagai gantinya, Anda dapat menentukan lebih banyak opsi konfigurasi saat membuat instans DB. Opsi ini mencakup pengaturan untuk ketersediaan, keamanan, cadangan, dan pemeliharaan. Untuk membuat instans DB publik, Anda harus menggunakan Pembuatan Standar. Untuk informasi, lihat Membuat instans DB Amazon RDS.

Topik

Prasyarat

Sebelum memulai, selesaikan langkah-langkah di bagian berikut:

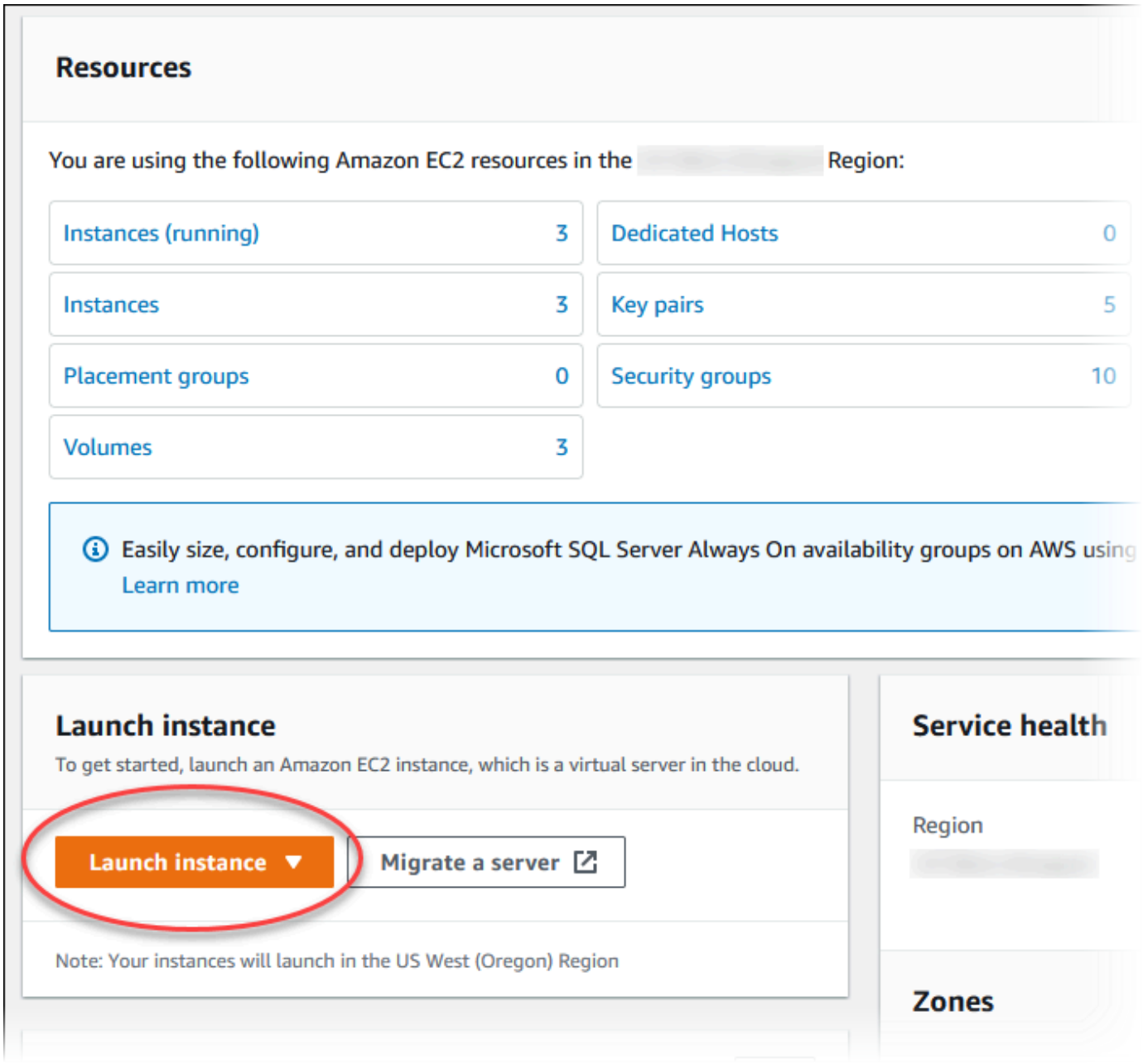
- Mendaftar Akun AWS
- Membuat pengguna administratif

Langkah 1: Buat instans EC2

Buat instans Amazon EC2 yang akan Anda gunakan untuk menghubungkan ke basis data Anda.

Untuk membuat instans EC2

1. Masuk ke AWS Management Console dan buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di sudut kanan atas AWS Management Console, pilih Wilayah AWS di mana Anda ingin membuat instance EC2.
3. Pilih Dasbor EC2, lalu pilih Luncurkan instans seperti yang ditampilkan dalam gambar berikut.

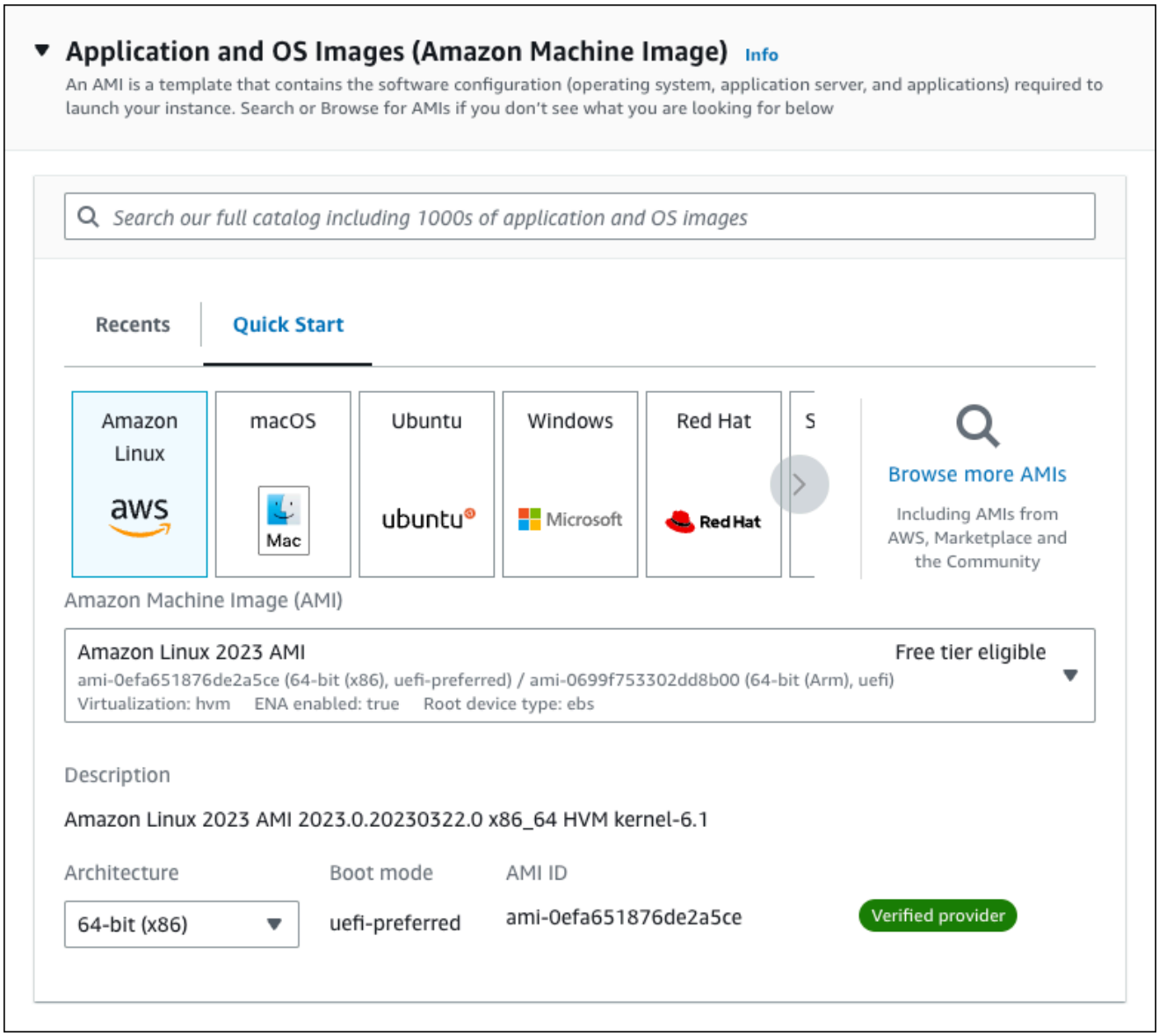


Halaman Meluncurkan instans akan terbuka.

4. Pilih pengaturan berikut di halaman Meluncurkan instans.

   a. Di bagian Nama dan tag, untuk Nama, masukkan **`ec2-database-connect`**.

   b. Di bagian Gambar Aplikasi dan OS (Amazon Machine Image), pilih Amazon Linux, lalu pilih AMI Amazon Linux 2023. Biarkan default untuk pilihan lainnya.



   c. Di bagian Jenis instans, pilih t2.micro.

   d. Di bagian Pasangan kunci (login), pilih nama Pasangan kunci untuk menggunakan pasangan kunci yang ada. Untuk membuat pasangan kunci baru untuk instans Amazon EC2, pilih Buat Pasangan kunci baru lalu gunakan jendela Buat pasangan kunci untuk membuatnya.

Untuk informasi selengkapnya tentang membuat pasangan kunci baru, lihat Membuat pasangan kunci di Panduan Pengguna Amazon EC2 untuk Instans Linux.

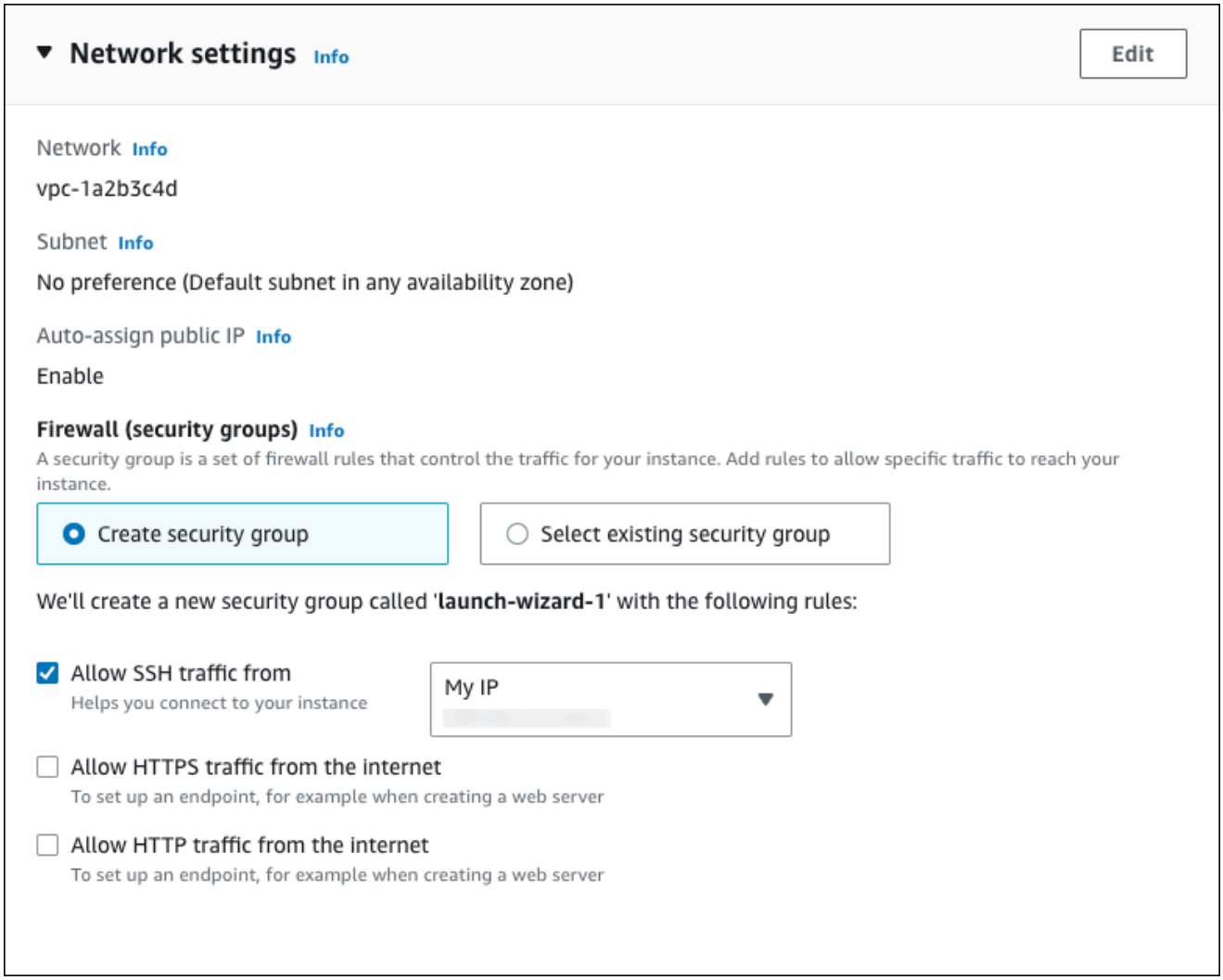
e. Untuk Izinkan lalu lintas SSH di Pengaturan jaringan, pilih sumber koneksi SSH ke instans EC2.

Anda dapat memilih IP Saya jika alamat IP yang ditampilkan benar untuk koneksi SSH. Jika tidak, Anda dapat menentukan alamat IP yang akan digunakan untuk menghubungkan ke instans EC2 di VPC Anda menggunakan Secure Shell (SSH). Untuk menentukan alamat IP publik Anda, Anda dapat membuka layanan di https://checkip.amazonaws.com di jendela atau tab browser lain. Contoh alamat IP adalah 192.0.2.1/32.

Dalam banyak kasus, Anda dapat menghubungkan melalui penyedia layanan Internet (ISP) atau dari belakang firewall Anda tanpa alamat IP statis. Jika demikian, tentukan rentang alamat IP yang digunakan oleh komputer klien.

> ⚠ Warning
>
> Jika menggunakan `0.0.0.0/0` untuk akses SSH, Anda memungkinkan semua alamat IP untuk mengakses instans publik EC2 Anda menggunakan SSH. Hal ini dapat diterima untuk waktu yang singkat di lingkungan pengujian, tetapi tidak aman untuk lingkungan produksi. Dalam produksi, Anda hanya dapat memberikan otorisasi pada alamat IP atau rentang alamat tertentu saja untuk mengakses instans EC2 Anda menggunakan SSH.

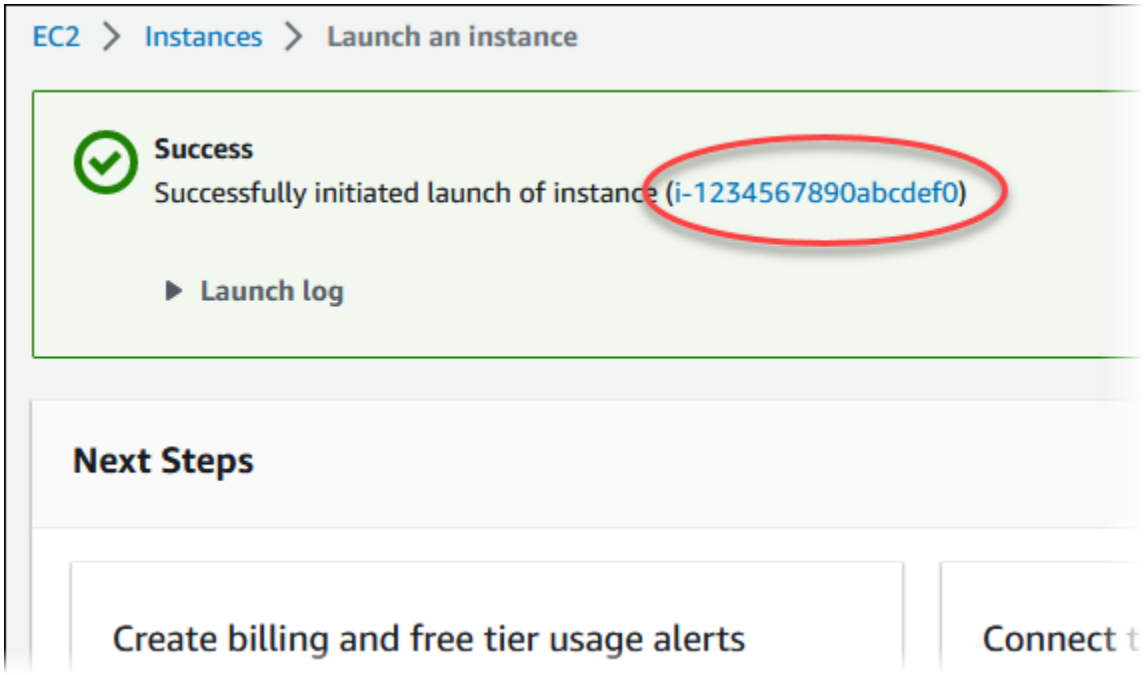
Gambar berikut menunjukkan contoh bagian Pengaturan jaringan.



f. Biarkan nilai default untuk bagian yang lainnya.

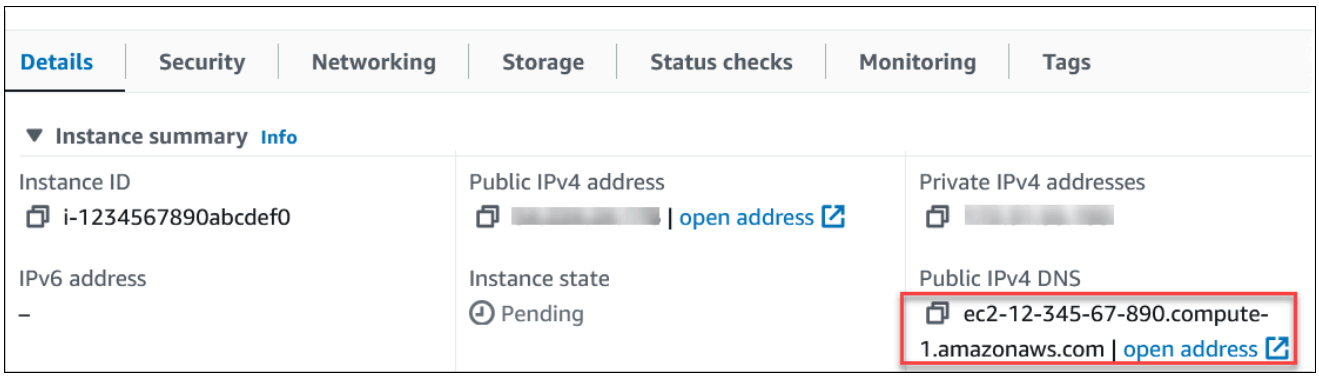
g. Tinjau ringkasan konfigurasi instans EC2 Anda di panel Ringkasan, dan setelah Anda siap, pilih Luncurkan instans.

5. Di halaman Status Peluncuran, catat pengidentifikasi untuk instans EC2 baru Anda, misalnya: `i-1234567890abcdef0`.

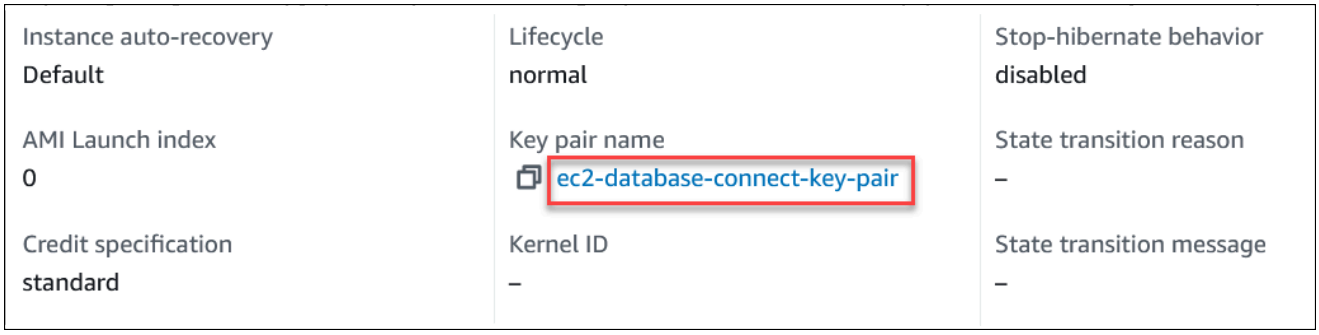


6. Pilih pengidentifikasi instans EC2 untuk membuka daftar instans EC2, lalu pilih instans EC2 Anda.
7. Di tab Detail, catat nilai-nilai berikut, yang akan Anda butuhkan saat menghubungkan menggunakan SSH:

   a. Di Ringkasan instans, catat nilai untuk DNS IPv4 Publik.



   b. Di Detail instans, catat nilai untuk Nama pasangan kunci.



8. Tunggu hingga Status instance untuk instans EC2 Anda berstatus Berjalan sebelum melanjutkan.

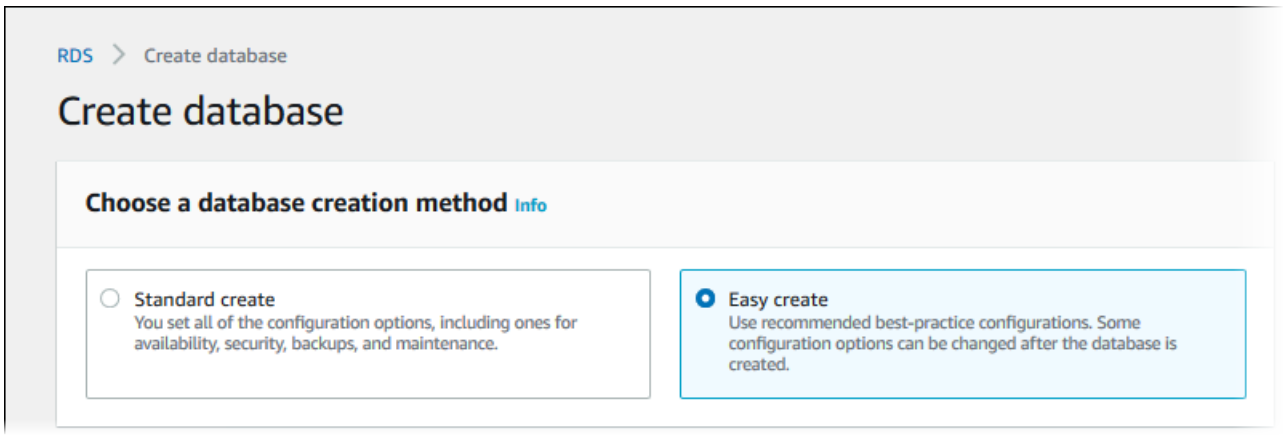
Langkah 2: Buat instans DB MySQL

Blok bangunan dasar Amazon RDS adalah instans DB. Lingkungan ini adalah tempat Anda menjalankan basis data MySQL Anda.

Dalam contoh ini, Anda menggunakan Pembuatan Mudah untuk membuat instans DB yang menjalankan mesin basis data MySQL dengan kelas instans DB db.t3.micro.

Untuk membuat instans DB MySQL dengan Pembuatan Mudah

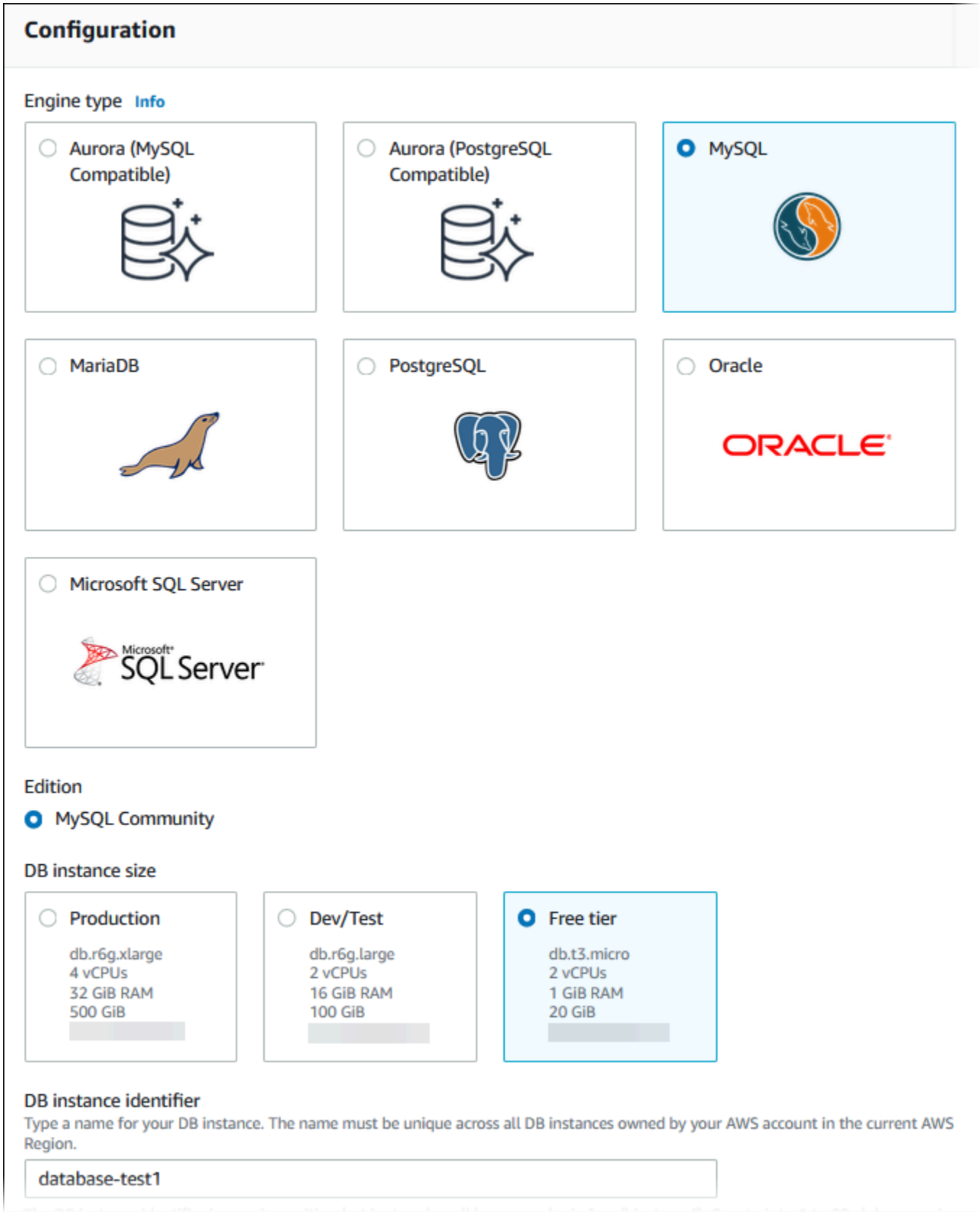
1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di sudut kanan atas konsol Amazon RDS, pilih yang Wilayah AWS Anda gunakan untuk instans EC2 sebelumnya.
3. Di panel navigasi, pilih Basis Data.
4. Pilih Buat basis data dan pastikan Pembuatan Mudah dipilih.



5. Di Konfigurasi, pilih MySQL.

6. Untuk Ukuran instans DB, pilih Tingkat gratis.
7. Untuk Pengidentifikasi instans DB, masukkan **`database-test1`**.
8. Untuk Nama pengguna utama, masukkan nama untuk pengguna utama, atau tetap gunakan nama default.

   Tampilan halaman Membuat basis data seperti gambar berikut.

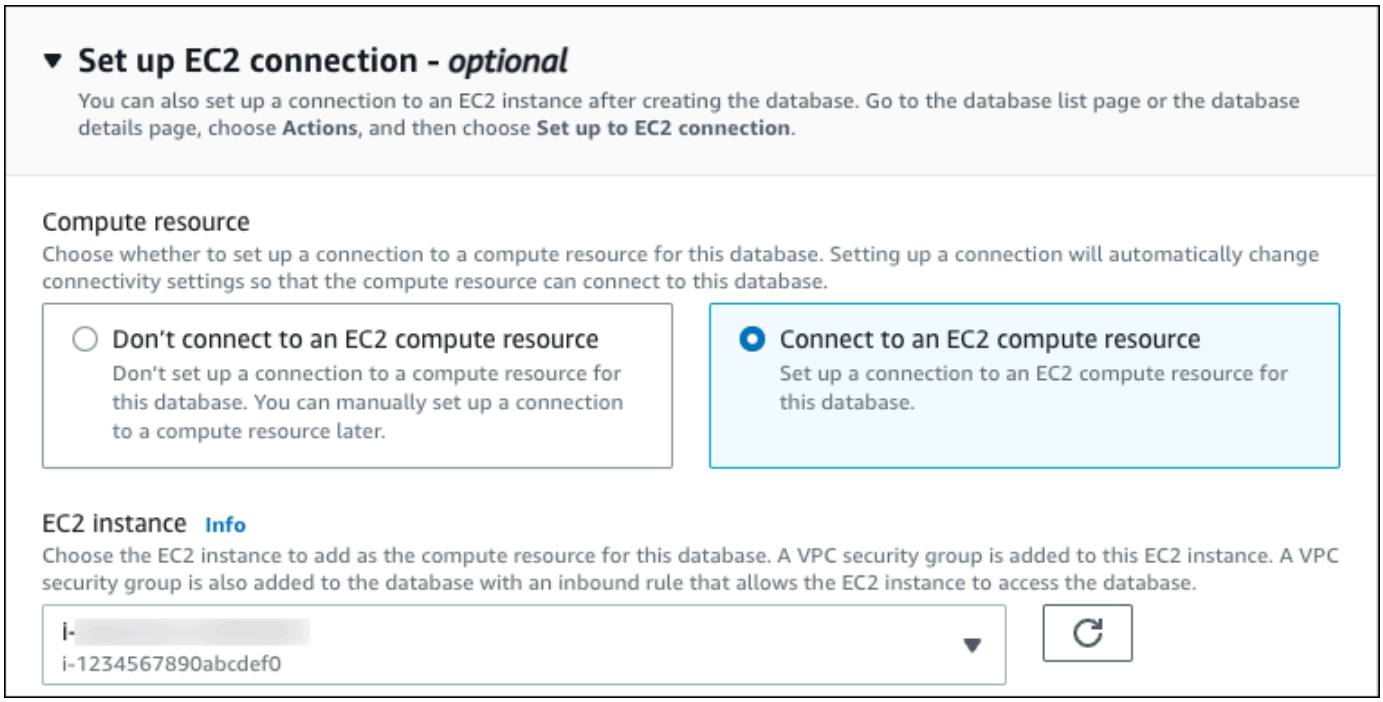

Configuration
Engine type Info
Aurora (MySQL Compatible)
Aurora (PostgreSQL Compatible)
MySQL
MariaDB
PostgreSQL
Oracle
ORACLE
Microsoft SQL Server
Microsoft SQL Server
Edition
MySQL Community
DB instance size
Production
db.r6g.xlarge
4 vCPUs
32 GiB RAM
500 GiB
Dev/Test
db.r6g.large
2 vCPUs
16 GiB RAM
100 GiB
Free tier
db.t3.micro
2 vCPUs
1 GiB RAM
20 GiB
DB instance identifier
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.
database-test1

9. Untuk menggunakan kata sandi utama yang dibuat secara otomatis untuk instans DB, pilih Buat kata sandi secara otomatis.

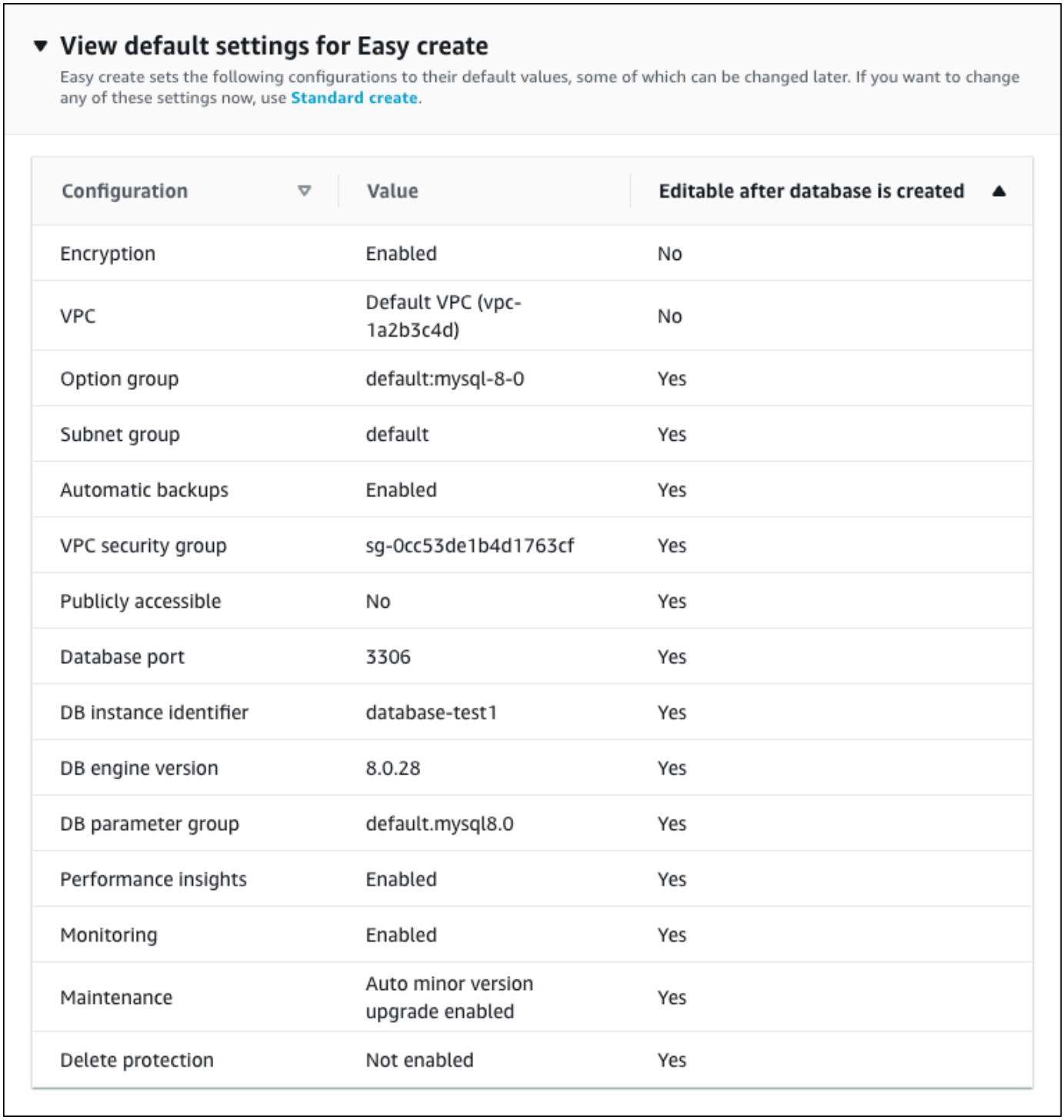
   Untuk memasukkan kata sandi utama Anda, hapus centang pada Buat kata sandi secara otomatis, lalu masukkan kata sandi yang sama dalam Kata sandi utama dan Konfirmasi kata sandi.

10. Untuk menyiapkan koneksi dengan instans EC2 yang Anda buat sebelumnya, buka Menyiapkan koneksi EC2 - opsional.

    Pilih Hubungkan ke sumber daya komputasi EC2. Pilih instans EC2 yang Anda buat sebelumnya.



11. (Opsional) Buka Lihat pengaturan default untuk Pembuatan Mudah.



View default settings for Easy create

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

Configuration	Value	Editable after database is created
Encryption	Enabled	No
VPC	Default VPC (vpc-1a2b3c4d)	No
Option group	default:mysql-8-0	Yes
Subnet group	default	Yes
Automatic backups	Enabled	Yes
VPC security group	sg-0cc53de1b4d1763cf	Yes
Publicly accessible	No	Yes
Database port	3306	Yes
DB instance identifier	database-test1	Yes
DB engine version	8.0.28	Yes
DB parameter group	default.mysql8.0	Yes
Performance insights	Enabled	Yes
Monitoring	Enabled	Yes
Maintenance	Auto minor version upgrade enabled	Yes
Delete protection	Not enabled	Yes

Anda dapat memeriksa pengaturan default yang digunakan dengan Pembuatan mudah. Kolom Dapat diedit setelah basis data dibuat menunjukkan opsi yang dapat Anda ubah setelah membuat basis data.

- Jika pengaturan memiliki Tidak di kolom tersebut, dan Anda menginginkan pengaturan yang berbeda, Anda dapat menggunakan Pembuatan Standar untuk membuat instans DB.
- Jika pengaturan memiliki Ya di kolom tersebut, dan Anda menginginkan pengaturan yang berbeda, Anda dapat menggunakan Pembuatan Standar untuk membuat instans DB, atau mengubah instans DB setelah Anda membuatnya untuk mengubah pengaturan.

12. Pilih Buat basis data.

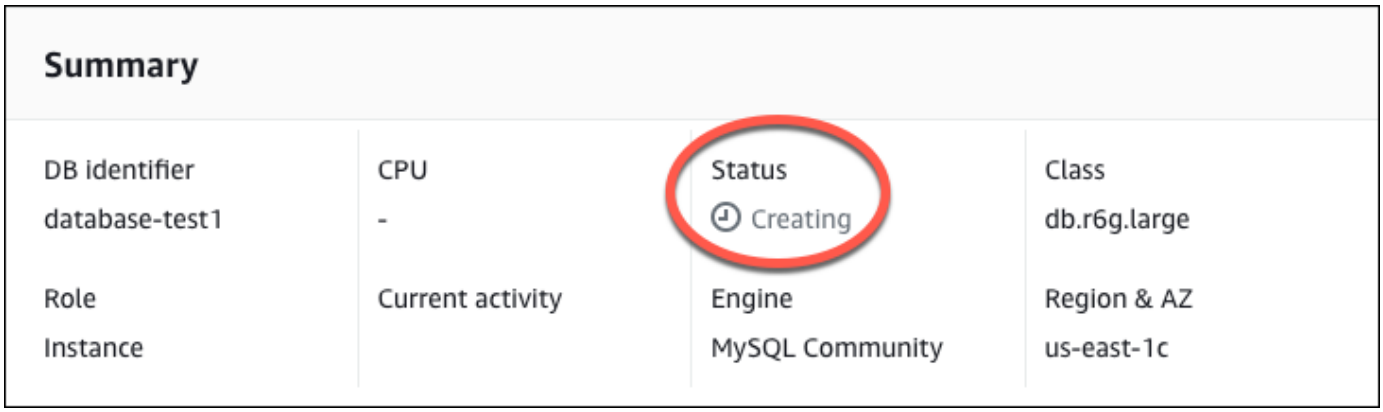
    Untuk melihat nama pengguna dan kata sandi utama untuk instans DB, pilih Lihat detail kredensial.

    Anda dapat menggunakan nama pengguna dan kata sandi yang ditampilkan untuk terhubung ke instans DB sebagai pengguna utama.

    > **Important**
    >
    > Anda tidak dapat melihat kata sandi pengguna utama lagi. Jika tidak mencatatnya, Anda mungkin harus mengubahnya.
    > Jika perlu mengubah kata sandi pengguna utama setelah instans DB tersedia, Anda dapat mengubah instans DB untuk melakukannya. Untuk informasi selengkapnya tentang cara mengubah instans DB, lihat Memodifikasi instans DB Amazon RDS.

13. Dalam daftar Basis Data, pilih nama instans DB MySQL yang baru untuk menampilkan detailnya.

    Instans DB memiliki status Membuat hingga siap digunakan.



    Saat statusnya berubah menjadi Tersedia, Anda dapat terhubung ke instans DB. Tergantung pada kelas instans DB dan jumlah penyimpanan, diperlukan waktu hingga 20 menit sebelum instans baru tersedia.

(Opsional) Buat instance VPC, EC2, dan MySQL menggunakan AWS CloudFormation

Alih-alih menggunakan konsol untuk membuat VPC, instans EC2, dan instance MySQL, Anda dapat menggunakannya AWS CloudFormation untuk menyediakan AWS sumber daya dengan memperlakukan infrastruktur sebagai kode. Untuk membantu Anda mengatur AWS sumber daya Anda menjadi unit yang lebih kecil dan lebih mudah dikelola, Anda dapat menggunakan fungsionalitas tumpukan AWS CloudFormation bersarang. Untuk informasi selengkapnya, lihat Membuat tumpukan di AWS CloudFormation konsol dan Bekerja dengan tumpukan bersarang.

> **Important**
>
> AWS CloudFormation gratis, tetapi sumber daya yang CloudFormation menciptakan hidup. Anda dikenakan biaya penggunaan standar untuk sumber daya ini sampai Anda menghentikannya. Total biaya akan minimal. Untuk informasi tentang cara meminimalkan biaya apa pun, buka Tingkat AWS Gratis.

Untuk membuat sumber daya menggunakan AWS CloudFormation konsol, selesaikan langkah-langkah berikut:

- Langkah 1: Unduh CloudFormation template
- Langkah 2: Konfigurasikan sumber daya Anda menggunakan CloudFormation

Unduh CloudFormation template

CloudFormation Template adalah file teks JSON atau YAMAL yang berisi informasi konfigurasi tentang sumber daya yang ingin Anda buat di tumpukan. Template ini juga membuat VPC dan host bastion untuk Anda bersama dengan instance RDS.

Untuk mengunduh file template, buka tautan berikut, template MySQL CloudFormation .

Di halaman Github, klik tombol Unduh file mentah untuk menyimpan file YAMAL template.

Konfigurasikan sumber daya Anda menggunakan CloudFormation

> **Note**
>
> Sebelum memulai proses ini, pastikan Anda memiliki pasangan Kunci untuk instans EC2 di. Akun AWS Untuk informasi selengkapnya, lihat Pasangan kunci Amazon EC2 dan instans Linux.

Ketika Anda menggunakan AWS CloudFormation template, Anda harus memilih parameter yang benar untuk memastikan sumber daya Anda dibuat dengan benar. Ikuti langkah-langkah di bawah ini:

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di https://console.aws.amazon.com/cloudformation.
2. Pilih Buat tumpukan.
3. Di bagian Tentukan templat, pilih Unggah file templat dari komputer Anda, lalu pilih Berikutnya.
4. Di halaman Tentukan detail tumpukan, atur parameter berikut:
   a. Tetapkan nama Stack ke MySQL TestStack.
   b. Di bawah Parameter, atur Availability Zone dengan memilih tiga zona ketersediaan.
   c. Di bawah konfigurasi Linux Bastion Host, untuk Key Name, pilih key pair untuk login ke instans EC2 Anda.
   d. Dalam pengaturan konfigurasi Linux Bastion Host, atur rentang IP yang Diizinkan ke alamat IP Anda. Untuk terhubung ke instans EC2 di VPC Anda menggunakan Secure Shell (SSH), tentukan alamat IP publik Anda menggunakan layanan di https://checkip.amazonaws.com. Contoh alamat IP adalah 192.0.2.1/32.

      > **Warning**
      >
      > Jika menggunakan `0.0.0.0/0` untuk akses SSH, Anda memungkinkan semua alamat IP untuk mengakses instans publik EC2 Anda menggunakan SSH. Hal ini dapat diterima untuk waktu yang singkat di lingkungan pengujian, tetapi tidak aman untuk lingkungan produksi. Dalam produksi, Anda hanya dapat memberikan otorisasi pada alamat IP atau rentang alamat tertentu saja untuk mengakses instans EC2 Anda menggunakan SSH.

   e. Di bawah konfigurasi Database General, atur kelas instance Database ke db.t3.micro.

f. Tetapkan nama Database ke**`database-test1`**.

g. Untuk nama pengguna master Database, masukkan nama untuk pengguna master.

h. Atur Manage DB master user password dengan Secrets Manager `false` untuk tutorial ini.

i. Untuk kata sandi Database, tetapkan kata sandi pilihan Anda. Ingat kata sandi ini untuk langkah lebih lanjut dalam tutorial.

j. Di bawah konfigurasi Penyimpanan Database, atur tipe penyimpanan Database ke gp2.

k. Di bawah konfigurasi Pemantauan Database, atur Aktifkan Performance Insights RDS ke false.

l. Biarkan semua pengaturan lainnya sebagai nilai default. Klik Berikutnya untuk melanjutkan.

5. Di halaman Configure stack options, tinggalkan semua opsi default. Klik Berikutnya untuk melanjutkan.
6. Di halaman tumpukan Tinjauan, pilih Kirim setelah memeriksa database dan opsi host bastion Linux.

Setelah proses pembuatan tumpukan selesai, lihat tumpukan dengan nama BastionStackdan RDSNS untuk mencatat informasi yang Anda butuhkan untuk terhubung ke database. Untuk informasi selengkapnya, lihat Melihat data AWS CloudFormation tumpukan dan sumber daya di AWS Management Console.

Langkah 3: Hubungkan ke instans DB MySQL

Anda dapat menggunakan aplikasi klien SQL standar untuk menghubungkan ke instans DB. Dalam contoh ini, Anda akan menghubungkan ke instans DB MySQL menggunakan klien baris perintah mysql.

Menghubungkan ke instans DB MySQL

1. Temukan titik akhir (nama DNS) dan nomor port untuk instans DB Anda.

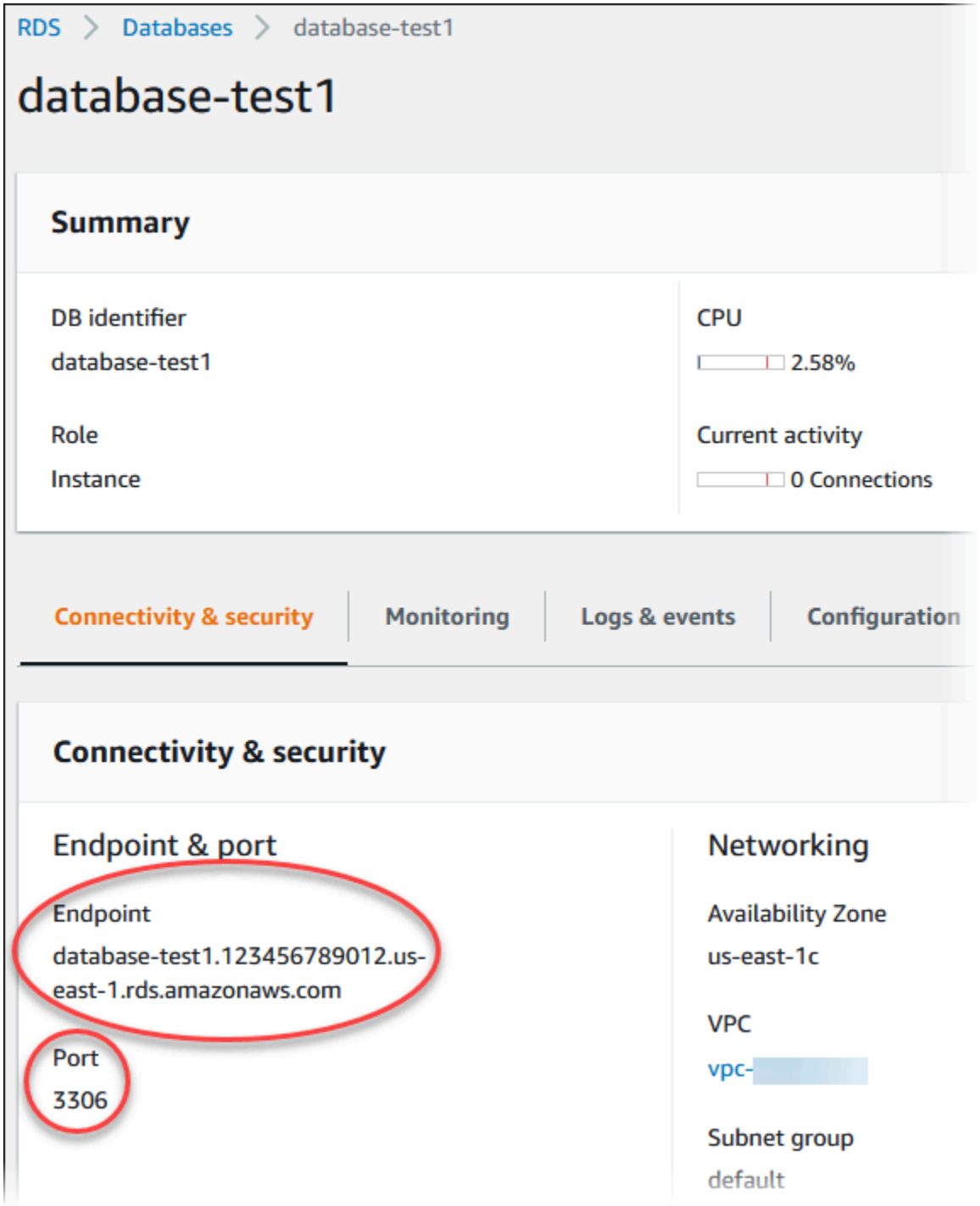
   a. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.

   b. Di sudut kanan atas konsol Amazon RDS, pilih instans DB Wilayah AWS .

   c. Di panel navigasi, pilih Basis data.

   d. Pilih nama instans DB MySQL untuk menampilkan detailnya.

   e. Di tab Konektivitas & keamanan, salin titik akhir. Selain itu, catat nomor porta. Anda memerlukan titik akhir dan nomor port untuk terhubung ke instans DB.



2. Hubungkan ke instans EC2 yang Anda buat sebelumnya dengan mengikuti langkah-langkah di Menghubungkan ke instans Linux di Panduan Pengguna Amazon EC2 untuk Instans Linux.

   Sebaiknya Anda menghubungkan ke instans EC2 menggunakan SSH. Jika utilitas klien SSH diinstal di Windows, Linux, atau Mac, Anda dapat menghubungkan ke instans menggunakan format perintah berikut:

```
ssh -i location_of_pem_file ec2-user@ec2-instance-public-dns-name
```

Misalnya, asumsikan bahwa `ec2-database-connect-key-pair.pem` disimpan di `/dir1` di Linux, dan DNS IPv4 publik untuk instans EC2 Anda adalah `ec2-12-345-678-90.compute-1.amazonaws.com`. Perintah SSH Anda akan tampak seperti berikut:

```
ssh -i /dir1/ec2-database-connect-key-pair.pem ec2-
user@ec2-12-345-678-90.compute-1.amazonaws.com
```

3. Dapatkan pembaruan keamanan dan perbaikan bug terbaru dengan memperbarui perangkat lunak di instans EC2 Anda. Untuk melakukannya, gunakan perintah berikut.

> **Note**
>
> Opsi `-y` menginstal pembaruan tanpa meminta konfirmasi. Hilangkan opsi ini untuk memeriksa pembaruan sebelum menginstal.

```
sudo dnf update -y
```

4. Untuk menginstal klien baris perintah mysql dari MariaDB di Amazon Linux 2023, jalankan perintah berikut:

```
sudo dnf install mariadb105
```

5. Hubungkan ke instans DB MySQL. Misalnya, masukkan perintah berikut. Tindakan ini memungkinkan Anda untuk terhubung ke instans DB MySQL menggunakan klien MySQL.

   Ganti titik akhir instans DB (nama DNS) untuk *`endpoint`*, dan ganti nama pengguna utama yang Anda gunakan untuk *`admin`*. Masukkan kata sandi utama yang Anda gunakan saat dimintai kata sandi.

```
mysql -h endpoint -P 3306 -u admin -p
```

   Setelah memasukkan kata sandi untuk pengguna, Anda akan melihat output yang serupa dengan yang berikut ini.

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 3082
Server version: 8.0.28 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MySQL [(none)]>
```

Untuk informasi selengkapnya tentang cara menghubungkan ke instans DB MySQL, lihat Menghubungkan ke instans DB yang menjalankan mesin basis data MySQL. Jika Anda tidak dapat terhubung ke instans DB Anda, lihat Tidak dapat terhubung ke instans DB Amazon RDS.

Untuk keamanan, praktik terbaiknya adalah menggunakan koneksi terenkripsi. Hanya gunakan koneksi MySQL yang tidak terenkripsi saat klien dan server berada di VPC yang sama dan jaringan tepercaya. Untuk informasi tentang cara menggunakan koneksi terenkripsi, lihat Menghubungkan dari klien baris perintah MySQL dengan SSL/TLS (terenkripsi).

6. Jalankan perintah SQL.

   Misalnya, perintah SQL berikut menunjukkan tanggal dan waktu saat ini:

   ```
   SELECT CURRENT_TIMESTAMP;
   ```

Langkah 4: Hapus instans EC2 dan instans DB

Setelah Anda terhubung ke dan menjelajahi instans EC2 dan instans DB sampel yang Anda buat, hapus instans tersebut sehingga Anda tidak lagi dikenakan biaya untuk instans DB tersebut.

Jika Anda biasa AWS CloudFormation membuat sumber daya, lewati langkah ini dan lanjutkan ke langkah berikutnya.

Untuk menghapus instans EC2

1. Masuk ke AWS Management Console dan buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Instans.
3. Pilih instans EC2, dan pilih Status instans, Akhiri instans.

4. Pilih Akhiri saat diminta untuk konfirmasi.

Untuk informasi selengkapnya tentang menghapus instans EC2, lihat Mengakhiri Instans dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Untuk menghapus instans DB tanpa snapshot DB akhir

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data.
3. Pilih instans DB yang ingin Anda hapus.
4. Untuk Tindakan, pilih Hapus.
5. Hapus Buat snapshot akhir? dan Pertahankan pencadangan otomatis.
6. Lengkapi pengakuan dan pilih Hapus.

(Opsional) Hapus instans EC2 dan instans DB yang dibuat dengan CloudFormation

Jika Anda biasa AWS CloudFormation membuat sumber daya, hapus CloudFormation tumpukan setelah Anda terhubung dan jelajahi contoh instans EC2 dan instans DB, sehingga Anda tidak lagi dikenakan biaya untuk itu.

Untuk menghapus sumber CloudFormation daya

1. Buka AWS CloudFormation konsol.
2. Pada halaman Stacks di CloudFormationconsole, pilih tumpukan root (tumpukan tanpa nama VPCStack, BastionStack atau RDSNS).
3. Pilih Hapus.
4. Pilih Hapus tumpukan saat diminta konfirmasi.

Untuk informasi selengkapnya tentang menghapus tumpukan CloudFormation, lihat Menghapus tumpukan di AWS CloudFormation konsol di AWS CloudFormation Panduan Pengguna.

(Opsional) Menghubungkan instans DB Anda ke fungsi Lambda

Anda juga dapat menghubungkan instans DB RDS for MySQL ke sumber daya komputasi nirserver Lambda. Fungsi Lambda memungkinkan Anda menjalankan kode tanpa menyediakan atau mengelola infrastruktur. Fungsi Lambda juga memungkinkan Anda untuk otomatis merespons permintaan eksekusi kode pada skala apa pun, mulai dari selusin peristiwa dalam sehari hingga ratusan per detik. Lihat informasi yang lebih lengkap di Menghubungkan secara otomatis fungsi Lambda dan instans basis data.

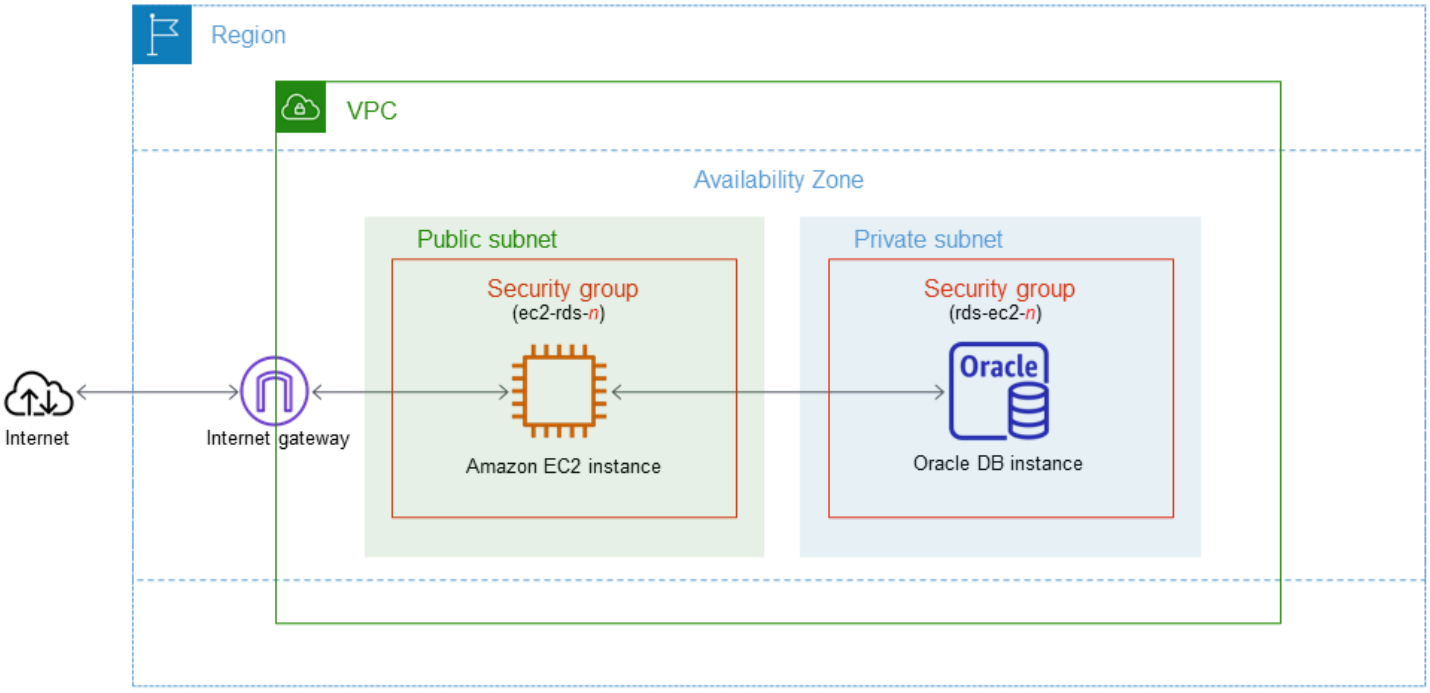
Membuat dan menghubungkan ke instans DB Oracle

Tutorial ini membuat instans EC2 dan instans DB RDS for Oracle. Tutorial ini menunjukkan cara mengakses instans DB dari instans EC2 menggunakan klien Oracle standar. Sebagai praktik terbaik, tutorial ini membuat instans DB privat dalam cloud privat virtual (VPC). Dalam kebanyakan kasus, sumber daya lain dalam VPC yang sama, seperti instans EC2, dapat mengakses instans DB, tetapi sumber daya di luar VPC tidak dapat mengaksesnya.

Setelah Anda menyelesaikan tutorial, ada subnet publik dan privat di setiap Zona Ketersediaan di VPC Anda. Dalam satu Zona Ketersediaan, instans EC2 berada di subnet publik, dan instans DB berada di subnet privat.

> **Important**
>
> Tidak ada biaya untuk membuat AWS akun. Namun, dengan menyelesaikan tutorial ini, Anda mungkin dikenakan biaya untuk AWS sumber daya yang Anda gunakan. Anda dapat menghapus sumber daya ini setelah menyelesaikan tutorial jika tidak diperlukan lagi.

Diagram berikut menunjukkan konfigurasi setelah tutorial selesai.



Tutorial ini memungkinkan Anda untuk membuat sumber daya Anda dengan menggunakan salah satu metode berikut:

1. Gunakan AWS Management Console - Langkah 2: Buat instans DB Oracle dan Langkah 1: Buat instans EC2
2. Gunakan AWS CloudFormation untuk membuat instance database dan instans EC2 - (Opsional) Buat VPC, instans EC2, dan instans Oracle DB menggunakan AWS CloudFormation

Metode pertama menggunakan Easy create untuk membuat instance Oracle DB pribadi dengan. AWS Management Console Di sini, Anda hanya menentukan jenis mesin DB, ukuran instans DB, dan pengidentifikasi instans DB. Pembuatan Mudah menggunakan pengaturan default untuk opsi konfigurasi lainnya.

Saat Anda menggunakan Standard create sebagai gantinya, Anda dapat menentukan lebih banyak opsi konfigurasi saat membuat instans DB. Opsi ini mencakup pengaturan untuk ketersediaan, keamanan, cadangan, dan pemeliharaan. Untuk membuat instans DB publik, Anda harus menggunakan Pembuatan Standar. Untuk informasi, lihat Membuat instans DB Amazon RDS.

Topik

Prasyarat

Sebelum memulai, selesaikan langkah-langkah di bagian berikut:

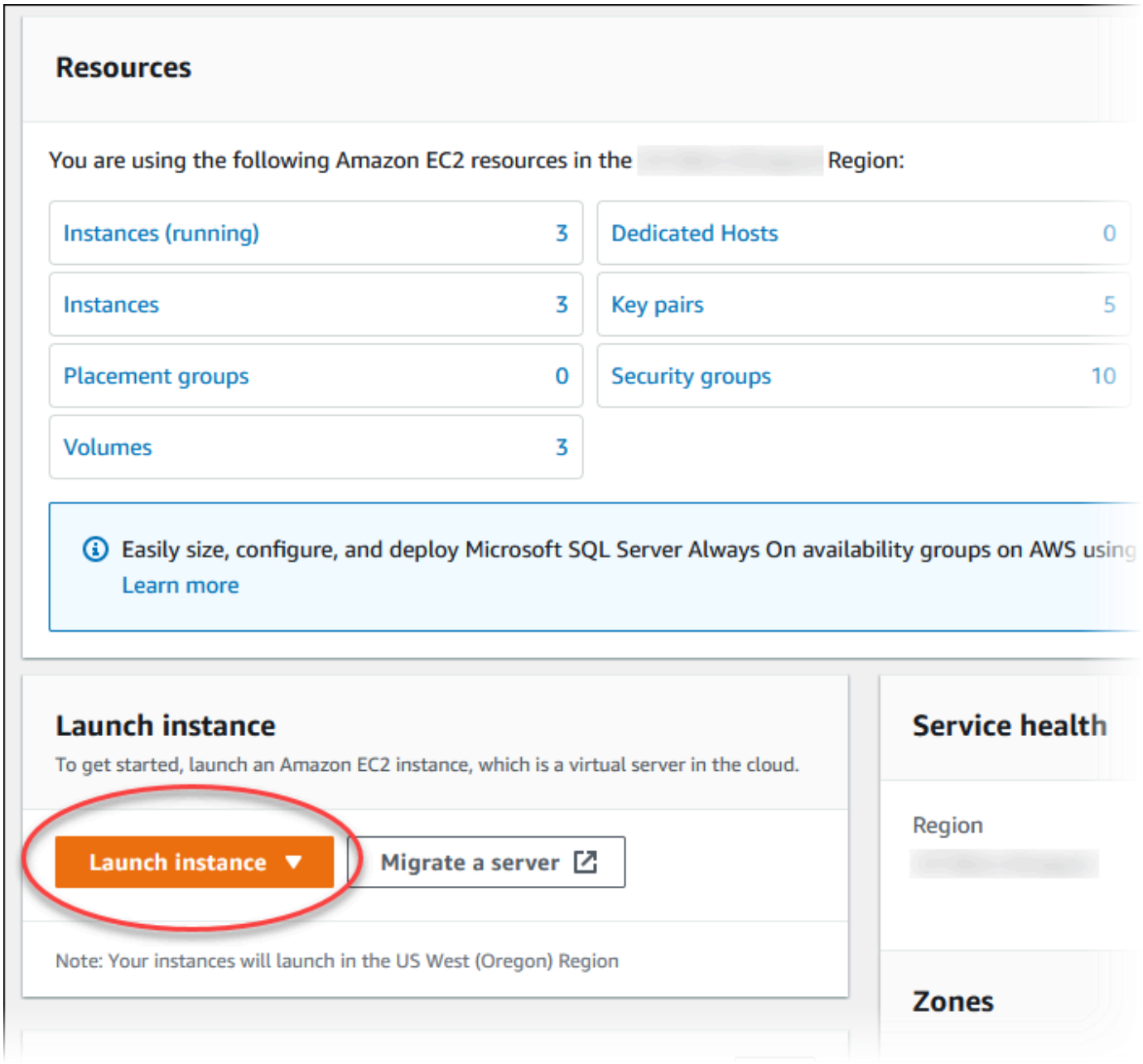
- Mendaftar Akun AWS
- Membuat pengguna administratif

Langkah 1: Buat instans EC2

Buat instans Amazon EC2 yang akan Anda gunakan untuk menghubungkan ke basis data Anda.

Untuk membuat instans EC2

1. Masuk ke AWS Management Console dan buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di sudut kanan atas AWS Management Console, pilih Wilayah AWS di mana Anda ingin membuat instans EC2.
3. Pilih Dasbor EC2, lalu pilih Luncurkan instans seperti yang ditampilkan dalam gambar berikut.

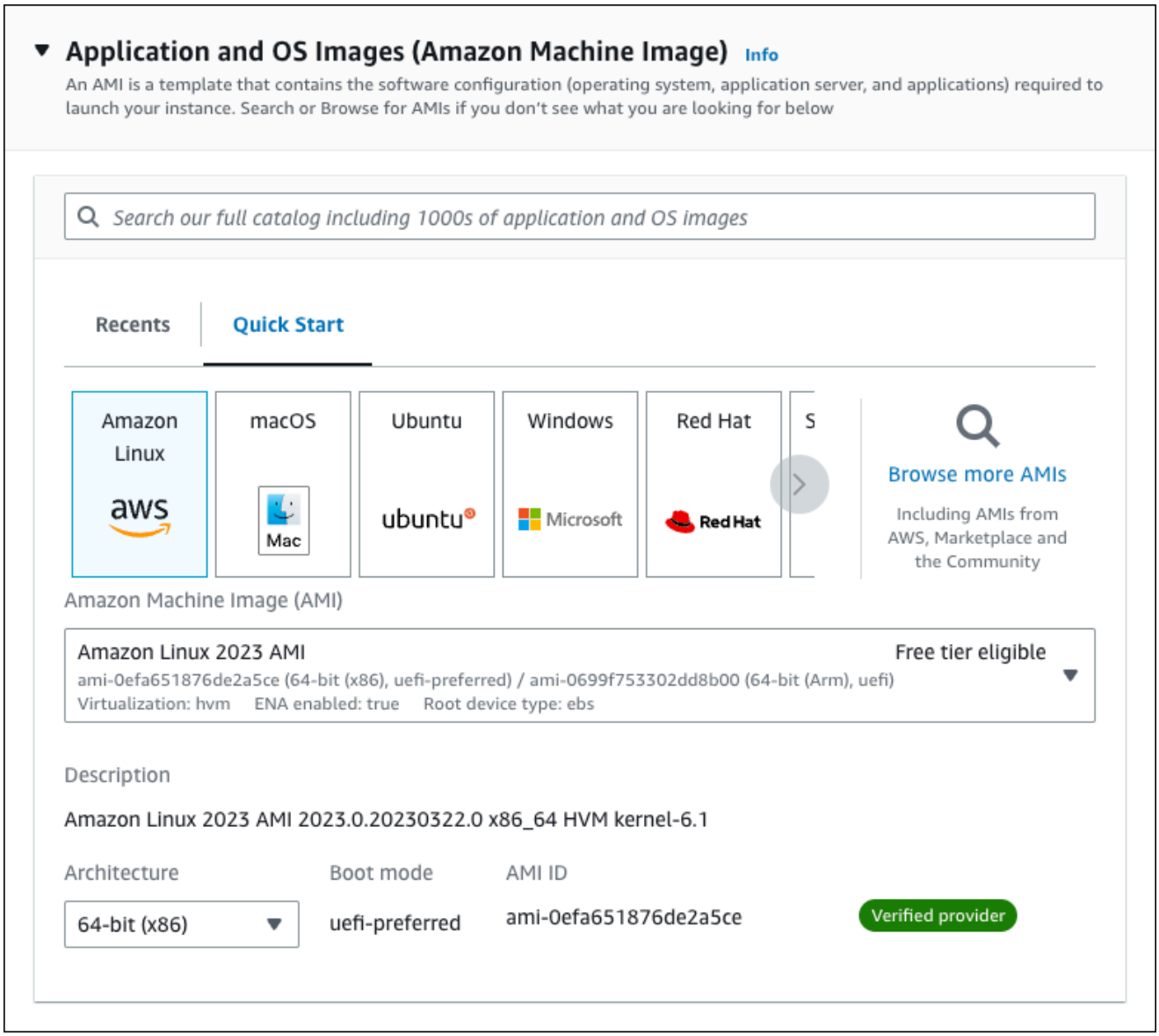


Halaman Meluncurkan instans akan terbuka.

4. Pilih pengaturan berikut di halaman Meluncurkan instans.

   a. Di bagian Nama dan tag, untuk Nama, masukkan **`ec2-database-connect`**.

   b. Di bagian Gambar Aplikasi dan OS (Amazon Machine Image), pilih Amazon Linux, lalu pilih AMI Amazon Linux 2023. Biarkan default untuk pilihan lainnya.



   c. Di bagian Jenis instans, pilih t2.micro.

   d. Di bagian Pasangan kunci (login), pilih nama Pasangan kunci untuk menggunakan pasangan kunci yang ada. Untuk membuat pasangan kunci baru untuk instans Amazon EC2, pilih Buat Pasangan kunci baru lalu gunakan jendela Buat pasangan kunci untuk membuatnya.

Untuk informasi selengkapnya tentang membuat pasangan kunci baru, lihat Membuat pasangan kunci di Panduan Pengguna Amazon EC2 untuk Instans Linux.

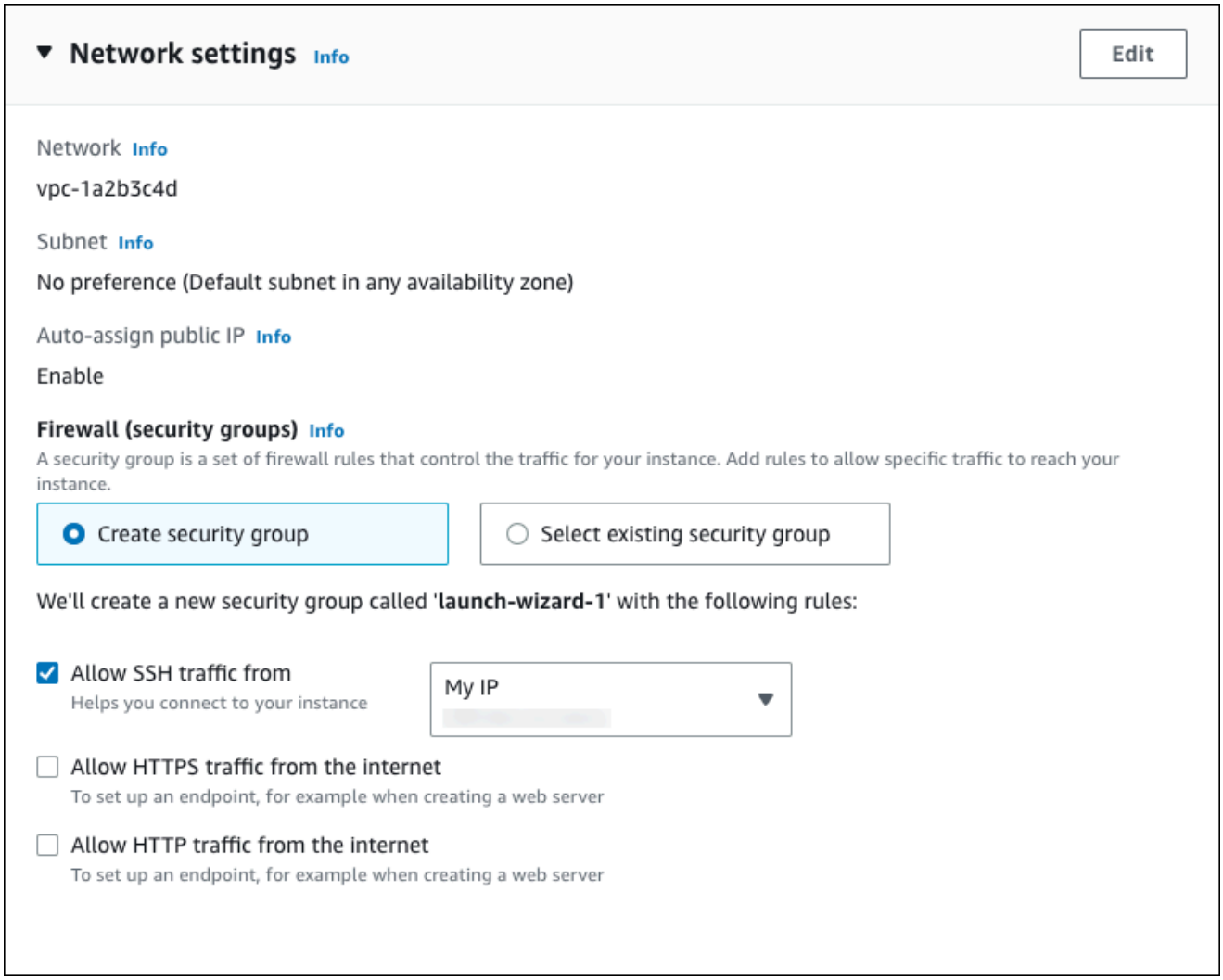
e. Untuk Izinkan lalu lintas SSH di Pengaturan jaringan, pilih sumber koneksi SSH ke instans EC2.

Anda dapat memilih IP Saya jika alamat IP yang ditampilkan benar untuk koneksi SSH. Jika tidak, Anda dapat menentukan alamat IP yang akan digunakan untuk menghubungkan ke instans EC2 di VPC Anda menggunakan Secure Shell (SSH). Untuk menentukan alamat IP publik Anda, Anda dapat membuka layanan di https://checkip.amazonaws.com di jendela atau tab browser lain. Contoh alamat IP adalah 192.0.2.1/32.

Dalam banyak kasus, Anda dapat menghubungkan melalui penyedia layanan Internet (ISP) atau dari belakang firewall Anda tanpa alamat IP statis. Jika demikian, tentukan rentang alamat IP yang digunakan oleh komputer klien.

> ⚠ Warning
>
> Jika menggunakan `0.0.0.0/0` untuk akses SSH, Anda memungkinkan semua alamat IP untuk mengakses instans publik EC2 Anda menggunakan SSH. Hal ini dapat diterima untuk waktu yang singkat di lingkungan pengujian, tetapi tidak aman untuk lingkungan produksi. Dalam produksi, Anda hanya dapat memberikan otorisasi pada alamat IP atau rentang alamat tertentu saja untuk mengakses instans EC2 Anda menggunakan SSH.

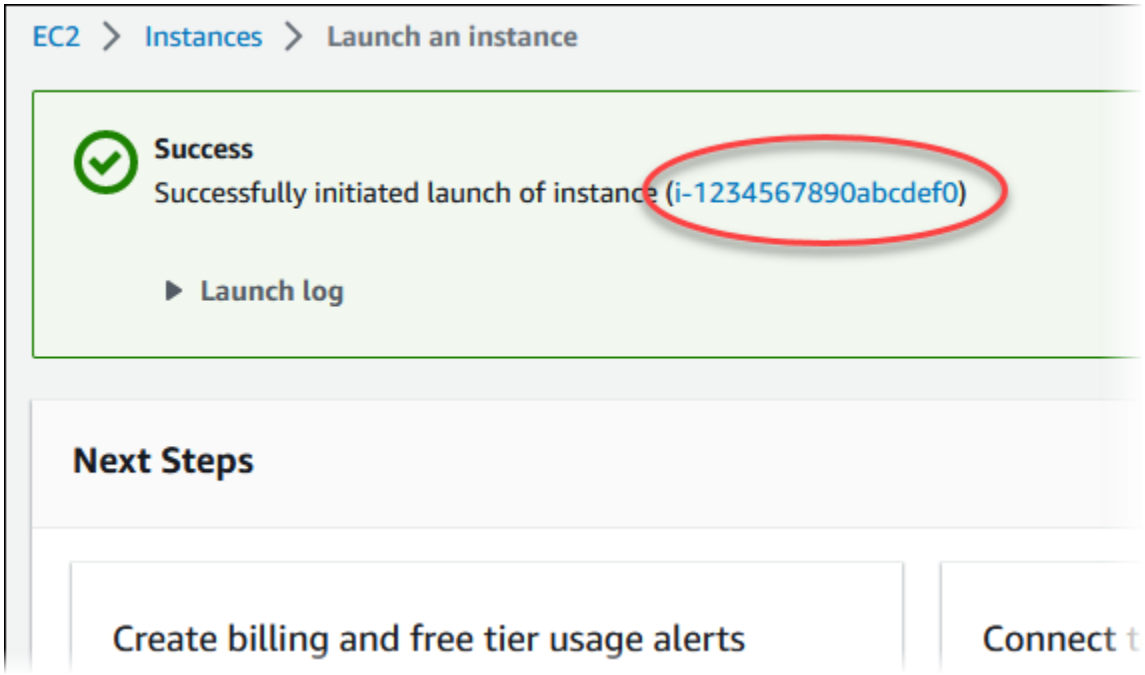
Gambar berikut menunjukkan contoh bagian Pengaturan jaringan.



f. Biarkan nilai default untuk bagian yang lainnya.

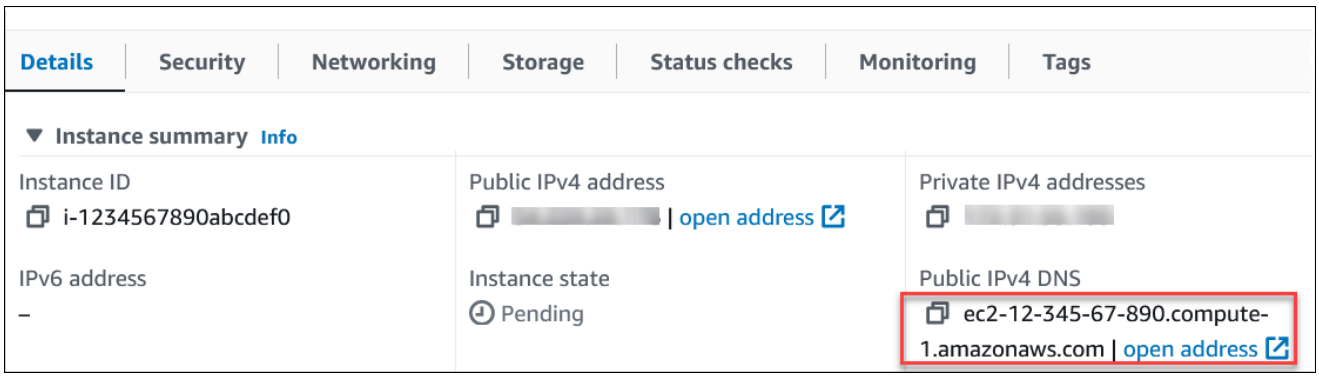
g. Tinjau ringkasan konfigurasi instans EC2 Anda di panel Ringkasan, dan setelah Anda siap, pilih Luncurkan instans.

5. Di halaman Status Peluncuran, catat pengidentifikasi untuk instans EC2 baru Anda, misalnya: `i-1234567890abcdef0`.

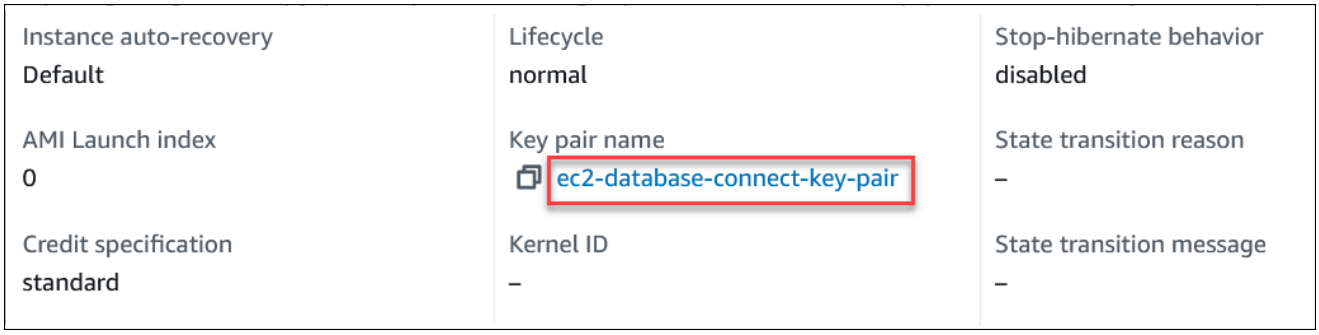


6. Pilih pengidentifikasi instans EC2 untuk membuka daftar instans EC2, lalu pilih instans EC2 Anda.
7. Di tab Detail, catat nilai-nilai berikut, yang akan Anda butuhkan saat menghubungkan menggunakan SSH:

   a. Di Ringkasan instans, catat nilai untuk DNS IPv4 Publik.



   b. Di Detail instans, catat nilai untuk Nama pasangan kunci.



8. Tunggu hingga Status instance untuk instans EC2 Anda berstatus Berjalan sebelum melanjutkan.

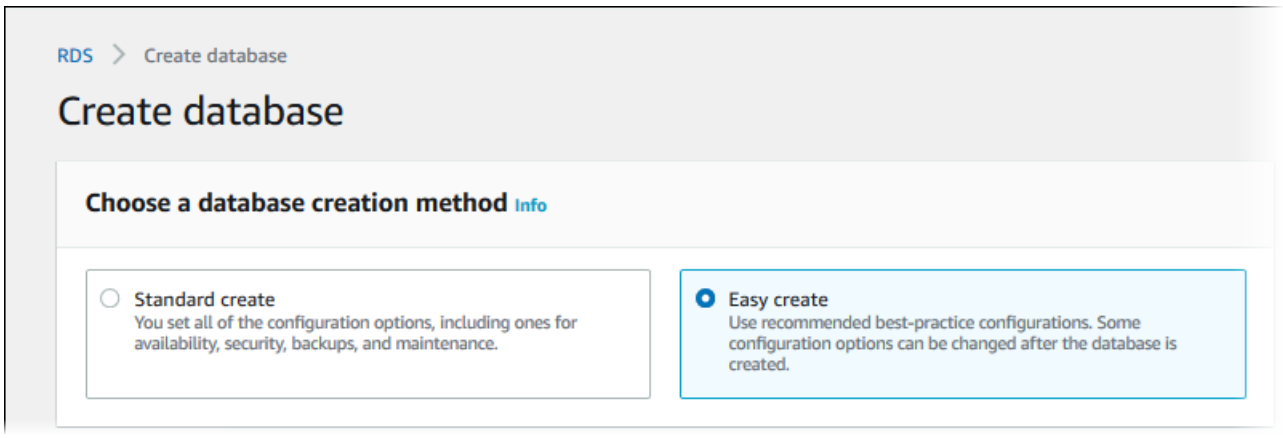
Langkah 2: Buat instans DB Oracle

Blok bangunan dasar Amazon RDS adalah instans DB. Lingkungan ini adalah tempat Anda menjalankan basis data Oracle Anda.

Dalam contoh ini, Anda menggunakan Pembuatan Mudah untuk membuat instans DB yang menjalankan mesin basis data Oracle dengan kelas instans DB db.m5.large.

Untuk membuat instans DB Oracle DB dengan Pembuatan Mudah

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di sudut kanan atas konsol Amazon RDS, pilih Wilayah AWS di mana Anda ingin membuat instans DB.
3. Di panel navigasi, pilih Basis Data.
4. Pilih Buat basis data dan pastikan Pembuatan Mudah dipilih.



5. Di Konfigurasi, pilih Oracle.

6. Untuk Ukuran instans DB, pilih Dev/Tes.
7. Untuk Pengidentifikasi instans DB, masukkan **`database-test1`**.
8. Untuk Nama pengguna utama, masukkan nama untuk pengguna utama, atau tetap gunakan nama default.

   Tampilan halaman Membuat basis data seperti gambar berikut.

Configuration

Engine type Info

- Aurora (MySQL Compatible)
- Aurora (PostgreSQL Compatible)
- MySQL
- MariaDB
- PostgreSQL
- Oracle
- Microsoft SQL Server

Edition

- Oracle Enterprise Edition
  Affordable and full-featured database management system supporting up to 16 vCPUs.
- Oracle Standard Edition Two
  Affordable and full-featured database management system supporting up to 16 vCPUs. Oracle Database Standard Edition Two is a replacement for Standard Edition and Standard Edition One.

DB instance size

- Production
  db.r5.large
  2 vCPUs
  16 GiB RAM
  500 GiB
- Dev/Test
  db.m5.large
  2 vCPUs
  8 GiB RAM
  100 GiB

DB instance identifier

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

database-test1

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

Master username Info

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

admin

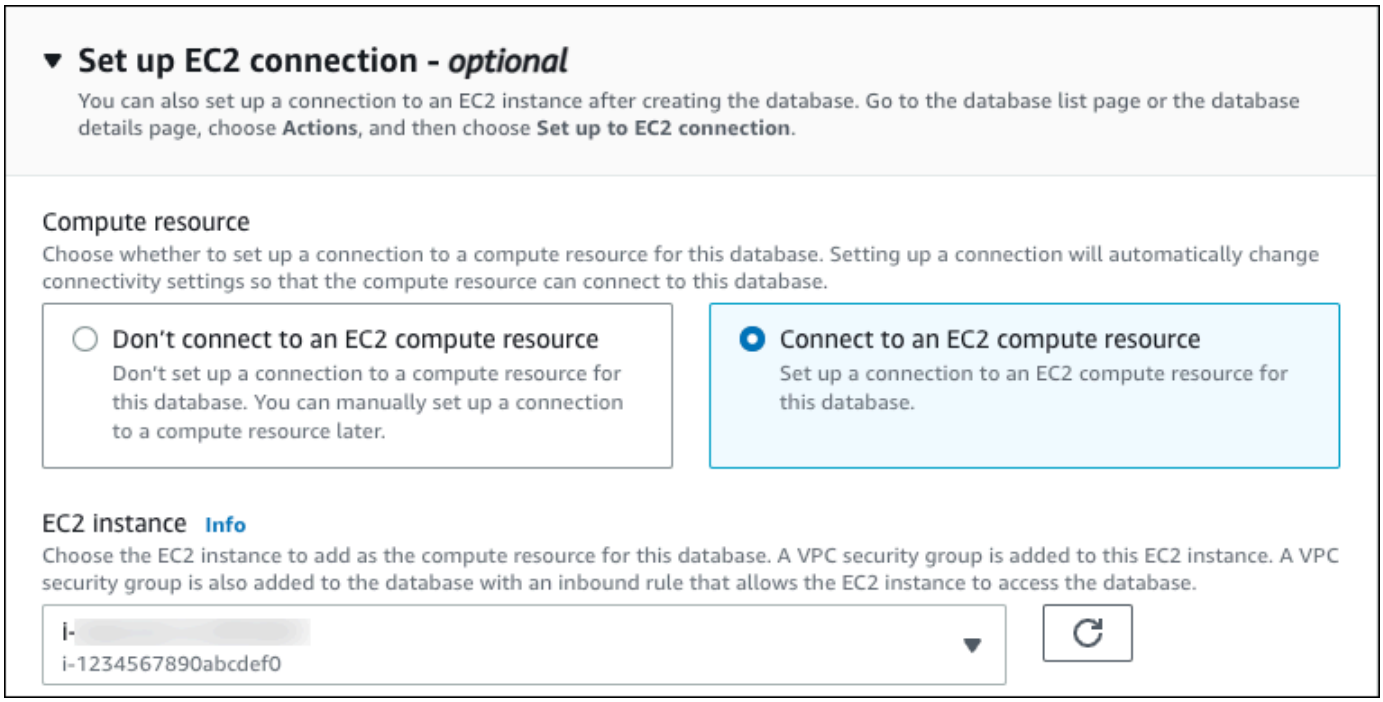
1 to 16 alphanumeric characters. First character must be a letter.

9. Untuk menggunakan kata sandi utama yang dibuat secara otomatis untuk instans DB, pilih Buat kata sandi secara otomatis.

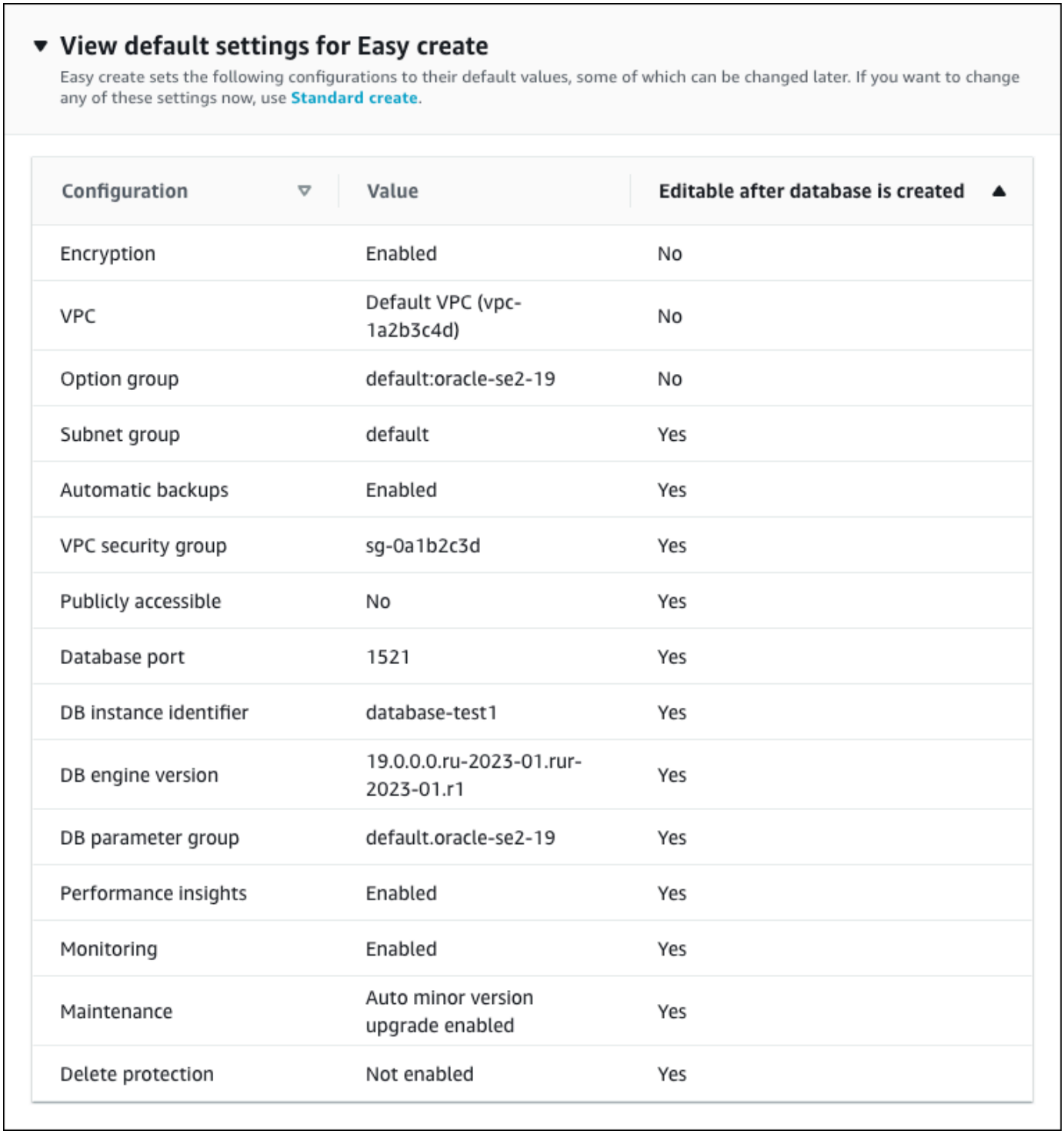
   Untuk memasukkan kata sandi utama Anda, hapus centang pada Buat kata sandi secara otomatis, lalu masukkan kata sandi yang sama dalam Kata sandi utama dan Konfirmasi kata sandi.

10. Untuk menyiapkan koneksi dengan instans EC2 yang Anda buat sebelumnya, buka Menyiapkan koneksi EC2 - opsional.

    Pilih Hubungkan ke sumber daya komputasi EC2. Pilih instans EC2 yang Anda buat sebelumnya.



11. Buka Lihat pengaturan default untuk Pembuatan Mudah.



View default settings for Easy create

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

Configuration	Value	Editable after database is created
Encryption	Enabled	No
VPC	Default VPC (vpc-1a2b3c4d)	No
Option group	default:oracle-se2-19	No
Subnet group	default	Yes
Automatic backups	Enabled	Yes
VPC security group	sg-0a1b2c3d	Yes
Publicly accessible	No	Yes
Database port	1521	Yes
DB instance identifier	database-test1	Yes
DB engine version	19.0.0.0.ru-2023-01.rur-2023-01.r1	Yes
DB parameter group	default.oracle-se2-19	Yes
Performance insights	Enabled	Yes
Monitoring	Enabled	Yes
Maintenance	Auto minor version upgrade enabled	Yes
Delete protection	Not enabled	Yes

Anda dapat memeriksa pengaturan default yang digunakan dengan Pembuatan mudah. Kolom Dapat diedit setelah basis data dibuat menunjukkan opsi yang dapat Anda ubah setelah membuat basis data.

- Jika pengaturan memiliki Tidak di kolom tersebut, dan Anda menginginkan pengaturan yang berbeda, Anda dapat menggunakan Pembuatan Standar untuk membuat instans DB.
- Jika pengaturan memiliki Ya di kolom tersebut, dan Anda menginginkan pengaturan yang berbeda, Anda dapat menggunakan Pembuatan Standar untuk membuat instans DB, atau mengubah instans DB setelah Anda membuatnya untuk mengubah pengaturan.

12. Pilih Buat basis data.

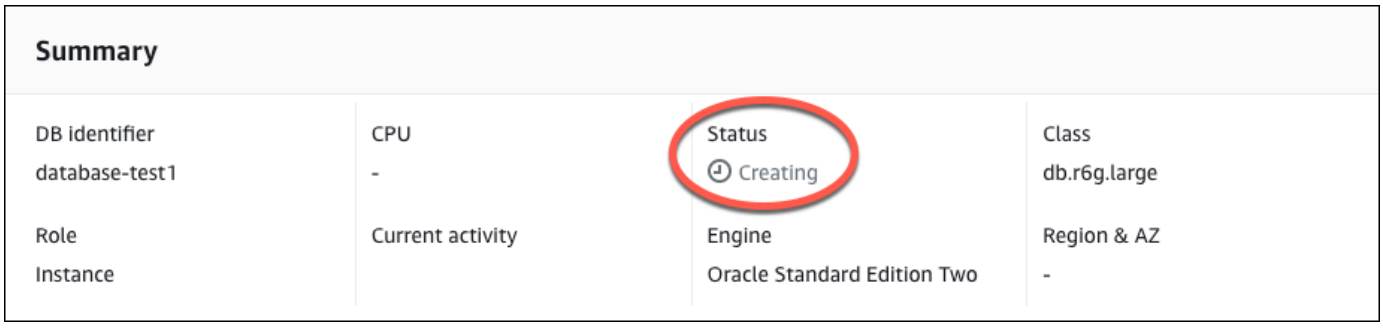
    Untuk melihat nama pengguna dan kata sandi utama untuk instans DB, pilih Lihat detail kredensial.

    Anda dapat menggunakan nama pengguna dan kata sandi yang ditampilkan untuk terhubung ke instans DB sebagai pengguna utama.

    > **Important**
    >
    > Anda tidak dapat melihat kata sandi pengguna utama lagi. Jika tidak mencatatnya, Anda mungkin harus mengubahnya.
    > Jika perlu mengubah kata sandi pengguna utama setelah instans DB tersedia, Anda dapat mengubah instans DB untuk melakukannya. Untuk informasi selengkapnya tentang cara mengubah instans DB, lihat Memodifikasi instans DB Amazon RDS.

13. Dalam daftar Basis Data, pilih nama instans DB Oracle yang baru untuk menampilkan detailnya.

    Instans DB memiliki status Membuat hingga siap digunakan.



    Saat statusnya berubah menjadi Tersedia, Anda dapat terhubung ke instans DB. Tergantung pada kelas instans DB dan jumlah penyimpanan, diperlukan waktu hingga 20 menit sebelum instans baru tersedia. Saat instans DB sedang dibuat, Anda dapat melanjutkan ke langkah berikutnya dan membuat instans EC2.

(Opsional) Buat VPC, instans EC2, dan instans Oracle DB menggunakan AWS CloudFormation

Alih-alih menggunakan konsol untuk membuat VPC, instans EC2, dan instans Oracle DB, Anda dapat menggunakannya AWS CloudFormation untuk menyediakan AWS sumber daya dengan memperlakukan infrastruktur sebagai kode. Untuk membantu Anda mengatur AWS sumber daya Anda menjadi unit yang lebih kecil dan lebih mudah dikelola, Anda dapat menggunakan fungsionalitas tumpukan AWS CloudFormation bersarang. Untuk informasi selengkapnya, lihat Membuat tumpukan di AWS CloudFormation konsol dan Bekerja dengan tumpukan bersarang.

> **Important**
>
> AWS CloudFormation gratis, tetapi sumber daya yang CloudFormation menciptakan hidup. Anda dikenakan biaya penggunaan standar untuk sumber daya ini sampai Anda menghentikannya. Total biaya akan minimal. Untuk informasi tentang cara meminimalkan biaya apa pun, buka Tingkat AWS Gratis.

Untuk membuat sumber daya Anda menggunakan AWS CloudFormation konsol, selesaikan langkah-langkah berikut:

- Langkah 1: Unduh CloudFormation template
- Langkah 2: Konfigurasikan sumber daya Anda menggunakan CloudFormation

Unduh CloudFormation template

CloudFormation Template adalah file teks JSON atau YAMAL yang berisi informasi konfigurasi tentang sumber daya yang ingin Anda buat di tumpukan. Template ini juga membuat VPC dan host bastion untuk Anda bersama dengan instance RDS.

Untuk mengunduh file template, buka tautan berikut, Oracle CloudFormation Template.

Di halaman Github, klik tombol Unduh file mentah untuk menyimpan file YAMAL template.

Konfigurasikan sumber daya Anda menggunakan CloudFormation

> **Note**
>
> Sebelum memulai proses ini, pastikan Anda memiliki pasangan Kunci untuk instans EC2 di. Akun AWS Untuk informasi selengkapnya, lihat Pasangan kunci Amazon EC2 dan instans Linux.

Ketika Anda menggunakan AWS CloudFormation template, Anda harus memilih parameter yang benar untuk memastikan sumber daya Anda dibuat dengan benar. Ikuti langkah-langkah di bawah ini:

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di https://console.aws.amazon.com/cloudformation.
2. Pilih Buat tumpukan.
3. Di bagian Tentukan templat, pilih Unggah file templat dari komputer Anda, lalu pilih Berikutnya.
4. Di halaman Tentukan detail tumpukan, atur parameter berikut:
   a. Tetapkan nama Stack ke OracleTestStack.
   b. Di bawah Parameter, atur Availability Zone dengan memilih tiga zona ketersediaan.
   c. Di bawah konfigurasi Linux Bastion Host, untuk Key Name, pilih key pair untuk login ke instans EC2 Anda.
   d. Dalam pengaturan konfigurasi Linux Bastion Host, atur rentang IP yang Diizinkan ke alamat IP Anda. Untuk terhubung ke instans EC2 di VPC Anda menggunakan Secure Shell (SSH), tentukan alamat IP publik Anda menggunakan layanan di https://checkip.amazonaws.com. Contoh alamat IP adalah 192.0.2.1/32.

      > **Warning**
      >
      > Jika menggunakan `0.0.0.0/0` untuk akses SSH, Anda memungkinkan semua alamat IP untuk mengakses instans publik EC2 Anda menggunakan SSH. Hal ini dapat diterima untuk waktu yang singkat di lingkungan pengujian, tetapi tidak aman untuk lingkungan produksi. Dalam produksi, Anda hanya dapat memberikan otorisasi pada alamat IP atau rentang alamat tertentu saja untuk mengakses instans EC2 Anda menggunakan SSH.

   e. Di bawah konfigurasi Database General, atur kelas instance Database ke db.t3.micro.

f. Tetapkan nama Database ke**`database-test1`**.

g. Untuk nama pengguna master Database, masukkan nama untuk pengguna master.

h. Atur Kelola kata sandi pengguna master DB dengan Secrets Manager `false` untuk tutorial ini.

i. Untuk kata sandi Database, tetapkan kata sandi pilihan Anda. Ingat kata sandi ini untuk langkah lebih lanjut dalam tutorial.

j. Di bawah konfigurasi Penyimpanan Database, atur tipe penyimpanan Database ke gp2.

k. Di bawah konfigurasi Pemantauan Database, atur Aktifkan Performance Insights RDS ke false.

l. Biarkan semua pengaturan lainnya sebagai nilai default. Klik Berikutnya untuk melanjutkan.

5. Di halaman Configure stack options, tinggalkan semua opsi default. Klik Berikutnya untuk melanjutkan.
6. Di halaman tumpukan Tinjauan, pilih Kirim setelah memeriksa database dan opsi host bastion Linux.

Setelah proses pembuatan tumpukan selesai, lihat tumpukan dengan nama BastionStackdan RDSNS untuk mencatat informasi yang Anda butuhkan untuk terhubung ke database. Untuk informasi selengkapnya, lihat Melihat data AWS CloudFormation tumpukan dan sumber daya di AWS Management Console.

Langkah 3: Hubungkan klien SQL Anda ke instans DB Oracle

Anda dapat menggunakan aplikasi klien SQL standar untuk menghubungkan ke instans DB Anda. Dalam contoh ini, Anda akan menghubungkan ke instans DB Oracle menggunakan klien baris perintah Oracle.

Menghubungkan ke instans DB Oracle

1. Temukan titik akhir (nama DNS) dan nomor port untuk instans DB Anda.

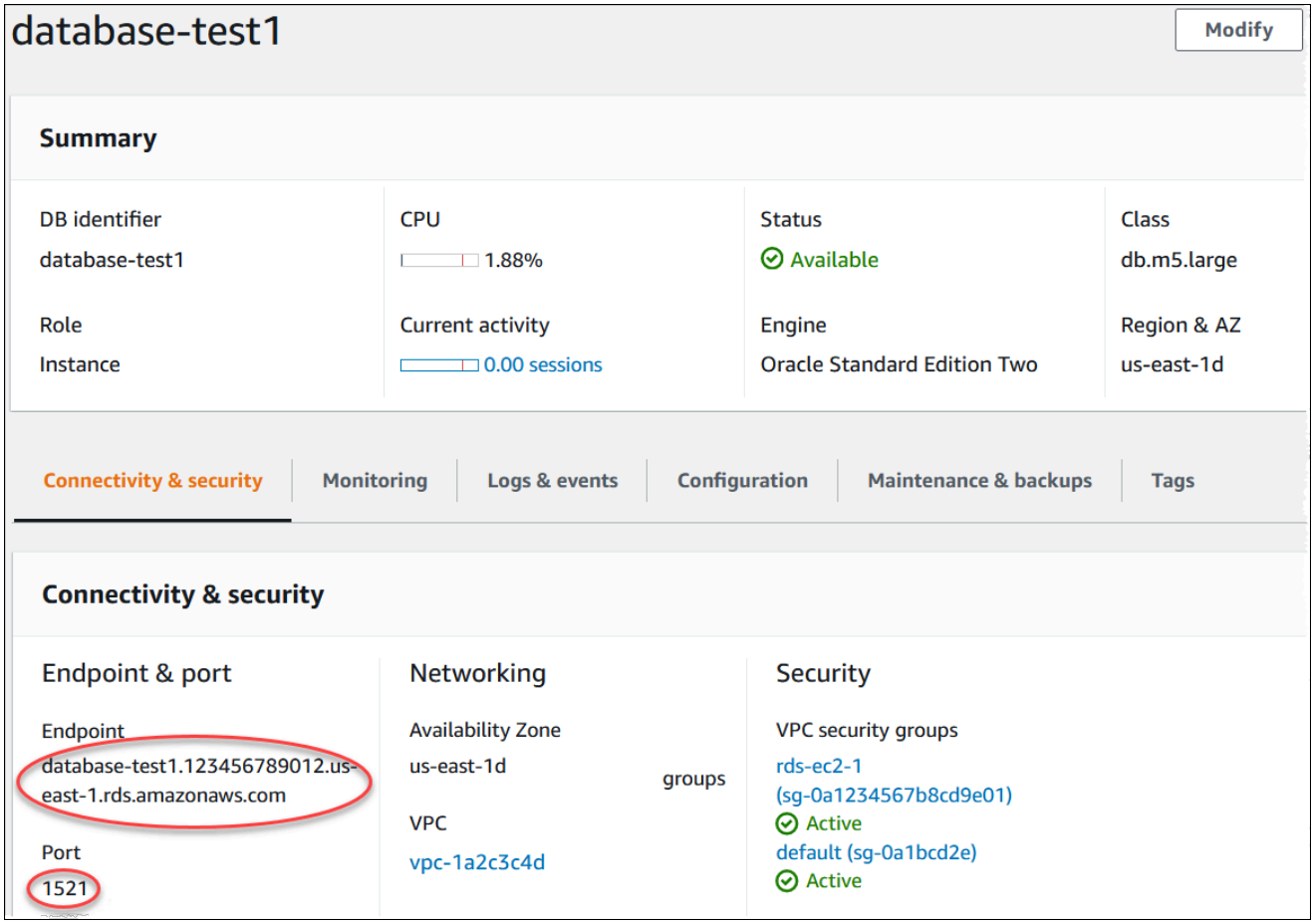
   a. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.

   b. Di sudut kanan atas konsol Amazon RDS, pilih Wilayah AWS untuk instans DB.

   c. Di panel navigasi, pilih Basis Data.

   d. Pilih nama instans DB Oracle untuk menampilkan detailnya.

   e. Di tab Konektivitas & keamanan, salin titik akhir. Selain itu, catat nomor porta. Anda memerlukan titik akhir dan nomor port untuk terhubung ke instans DB.



2. Hubungkan ke instans EC2 yang Anda buat sebelumnya dengan mengikuti langkah-langkah di Menghubungkan ke instans Linux di Panduan Pengguna Amazon EC2 untuk Instans Linux.

   Sebaiknya Anda menghubungkan ke instans EC2 menggunakan SSH. Jika utilitas klien SSH diinstal di Windows, Linux, atau Mac, Anda dapat menghubungkan ke instans menggunakan format perintah berikut:

   ```
   ssh -i location_of_pem_file ec2-user@ec2-instance-public-dns-name
   ```

   Misalnya, asumsikan bahwa `ec2-database-connect-key-pair.pem` disimpan di `/dir1` di Linux, dan DNS IPv4 publik untuk instans EC2 Anda adalah `ec2-12-345-678-90.compute-1.amazonaws.com`. Perintah SSH Anda akan tampak seperti berikut:

   ```
   ssh -i /dir1/ec2-database-connect-key-pair.pem ec2-
   user@ec2-12-345-678-90.compute-1.amazonaws.com
   ```

3. Dapatkan pembaruan keamanan dan perbaikan bug terbaru dengan memperbarui perangkat lunak di instans EC2 Anda. Untuk melakukannya, gunakan perintah berikut.

   > **Note**
   >
   > Opsi `-y` menginstal pembaruan tanpa meminta konfirmasi. Hilangkan opsi ini untuk memeriksa pembaruan sebelum menginstal.

   ```
   sudo dnf update -y
   ```

4. Di browser web, buka https://www.oracle.com/database/technologies/instant-client/linux-x86-64-downloads.html.
5. Untuk versi basis data terbaru yang muncul di halaman web, salin tautan .rpm (bukan tautan .zip) untuk Instant Client Basic Package dan SQL*Plus Package. Sebagai contoh, tautan berikut adalah untuk Oracle Database versi 21.9:

   - https://download.oracle.com/otn_software/linux/instantclient/219000/ oracle-instantclient-basic -21.9.0.0.0-1.el8.x86_64.rpm
   - https://download.oracle.com/otn_software/linux/instantclient/219000/ oracle-instantclient-sqlplus -21.9.0.0.0-1.el8.x86_64.rpm

6. Dalam sesi SSH Anda, jalankan perintah `wget` untuk mengunduh file .rpm dari tautan yang Anda peroleh pada langkah sebelumnya. Contoh berikut mengunduh file .rpm untuk Oracle Database versi 21.9:

   ```
   wget https://download.oracle.com/otn_software/linux/instantclient/219000/oracle-instantclient-basic-21.9.0.0.0-1.el8.x86_64.rpm
   wget https://download.oracle.com/otn_software/linux/instantclient/219000/oracle-instantclient-sqlplus-21.9.0.0.0-1.el8.x86_64.rpm
   ```

7. Instal paket dengan menjalankan perintah `dnf` sebagai berikut:

   ```
   sudo dnf install oracle-instantclient-*.rpm
   ```

8. Mulai SQL*Plus dan hubungkan ke instans DB Oracle. Misalnya, masukkan perintah berikut.

   Ganti titik akhir instans DB (nama DNS) untuk *`oracle-db-instance-endpoint`*, dan ganti nama pengguna utama yang Anda gunakan untuk *`admin`*. Jika Anda menggunakan Pembuatan

Mudah untuk Oracle, nama basis datanya adalah DATABASE. Masukkan kata sandi utama yang Anda gunakan saat dimintai kata sandi.

```
sqlplus admin@oracle-db-instance-endpoint:1521/DATABASE
```

Setelah memasukkan kata sandi untuk pengguna, Anda akan melihat output yang serupa dengan yang berikut ini.

```
SQL*Plus: Release 21.0.0.0.0 - Production on Wed Mar 1 16:41:28 2023
Version 21.9.0.0.0

Copyright (c) 1982, 2022, Oracle.  All rights reserved.

Enter password:
Last Successful login time: Wed Mar 01 2023 16:30:52 +00:00

Connected to:
Oracle Database 19c Standard Edition 2 Release 19.0.0.0.0 - Production
Version 19.18.0.0.0

SQL>
```

Untuk informasi selengkapnya tentang cara menghubungkan ke instans DB Oracle, lihat Menghubungkan ke instans RDS for Oracle DB. Jika Anda tidak dapat terhubung ke instans DB Anda, lihat Tidak dapat terhubung ke instans DB Amazon RDS.

Untuk keamanan, praktik terbaiknya adalah menggunakan koneksi terenkripsi. Hanya gunakan koneksi Oracle yang tidak terenkripsi saat klien dan server berada di VPC yang sama dan jaringan tepercaya. Untuk informasi tentang cara menggunakan koneksi terenkripsi, lihat Mengamankan koneksi instans DB Oracle.

9. Jalankan perintah SQL.

   Misalnya, perintah SQL berikut menunjukkan tanggal saat ini:

   ```
   SELECT SYSDATE FROM DUAL;
   ```

Langkah 4: Hapus instans EC2 dan instans DB

Setelah Anda terhubung ke dan menjelajahi instans EC2 dan instans DB sampel yang Anda buat, hapus instans tersebut sehingga Anda tidak lagi dikenakan biaya untuk instans DB tersebut.

Jika Anda biasa AWS CloudFormation membuat sumber daya, lewati langkah ini dan lanjutkan ke langkah berikutnya.

Untuk menghapus instans EC2

1. Masuk ke AWS Management Console dan buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Instans.
3. Pilih instans EC2, dan pilih Status instans, Akhiri instans.
4. Pilih Akhiri saat diminta untuk konfirmasi.

Untuk informasi selengkapnya tentang menghapus instans EC2, lihat Mengakhiri Instans dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Untuk menghapus instans DB tanpa snapshot DB akhir

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data.
3. Pilih instans DB yang ingin Anda hapus.
4. Untuk Tindakan, pilih Hapus.
5. Hapus Buat snapshot akhir? dan Pertahankan pencadangan otomatis.
6. Lengkapi pengakuan dan pilih Hapus.

(Opsional) Hapus instans EC2 dan instans DB yang dibuat dengan CloudFormation

Jika Anda biasa AWS CloudFormation membuat sumber daya, hapus CloudFormation tumpukan setelah Anda terhubung dan jelajahi contoh instans EC2 dan instans DB, sehingga Anda tidak lagi dikenakan biaya untuk itu.

Untuk menghapus sumber CloudFormation daya

1. Buka AWS CloudFormation konsol.
2. Pada halaman Stacks di CloudFormationconsole, pilih tumpukan root (tumpukan tanpa nama VPCStack, BastionStack atau RDSNS).
3. Pilih Hapus.
4. Pilih Hapus tumpukan saat diminta konfirmasi.

Untuk informasi selengkapnya tentang menghapus tumpukan CloudFormation, lihat Menghapus tumpukan di AWS CloudFormation konsol di AWS CloudFormation Panduan Pengguna.

(Opsional) Menghubungkan instans DB Anda ke fungsi Lambda

Anda juga dapat menghubungkan instans DB RDS for Oracle ke sumber daya komputasi nirserver Lambda. Fungsi Lambda memungkinkan Anda menjalankan kode tanpa menyediakan atau mengelola infrastruktur. Fungsi Lambda juga memungkinkan Anda untuk otomatis merespons permintaan eksekusi kode pada skala apa pun, mulai dari selusin peristiwa dalam sehari hingga ratusan per detik. Lihat informasi yang lebih lengkap di Menghubungkan secara otomatis fungsi Lambda dan instans basis data.

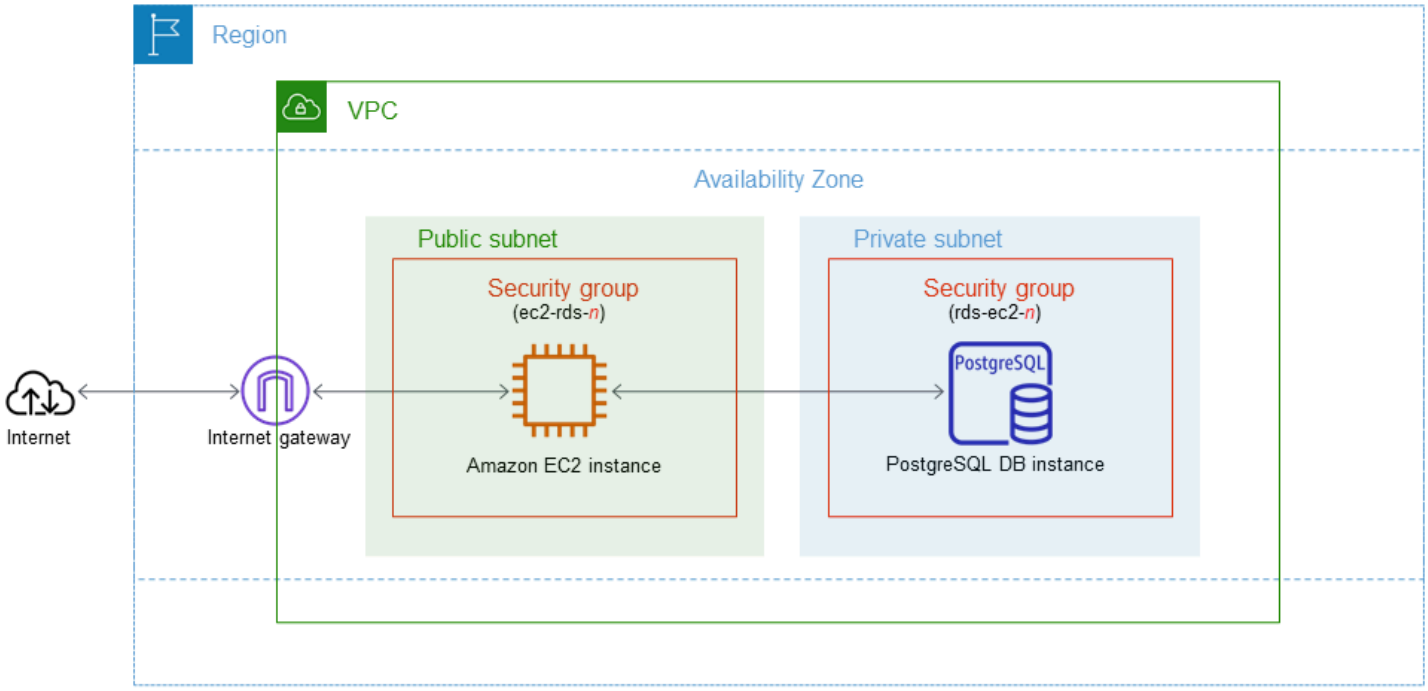
Membuat dan menghubungkan ke instans DB PostgreSQL

Tutorial ini membuat instans EC2 dan instans DB RDS for PostgreSQL. Tutorial ini menunjukkan cara mengakses instans DB dari instans EC2 menggunakan klien PostgreSQL standar. Sebagai praktik terbaik, tutorial ini membuat instans DB privat dalam cloud privat virtual (VPC). Dalam kebanyakan kasus, sumber daya lain dalam VPC yang sama, seperti instans EC2, dapat mengakses instans DB, tetapi sumber daya di luar VPC tidak dapat mengaksesnya.

Setelah Anda menyelesaikan tutorial, ada subnet publik dan privat di setiap Zona Ketersediaan di VPC Anda. Dalam satu Zona Ketersediaan, instans EC2 berada di subnet publik, dan instans DB berada di subnet privat.

> **Important**
>
> Tidak ada biaya untuk membuat AWS akun. Namun, dengan menyelesaikan tutorial ini, Anda mungkin dikenakan biaya untuk AWS sumber daya yang Anda gunakan. Anda dapat menghapus sumber daya ini setelah menyelesaikan tutorial jika tidak diperlukan lagi.

Diagram berikut menunjukkan konfigurasi setelah tutorial selesai.



Tutorial ini memungkinkan Anda untuk membuat sumber daya Anda dengan menggunakan salah satu metode berikut:

1. Gunakan AWS Management Console - Langkah 1: Buat instans EC2 dan Langkah 2: Buat instans DB PostgreSQL
2. Gunakan AWS CloudFormation untuk membuat instance database dan instans EC2 - (Opsional) Buat instance VPC, EC2, dan PostgreSQL menggunakan AWS CloudFormation

Metode pertama menggunakan Easy create untuk membuat instance PostgreSQL DB pribadi dengan. AWS Management Console Di sini, Anda hanya menentukan jenis mesin DB, ukuran instans DB, dan pengidentifikasi instans DB. Pembuatan Mudah menggunakan pengaturan default untuk opsi konfigurasi lainnya.

Saat Anda menggunakan Standard create sebagai gantinya, Anda dapat menentukan lebih banyak opsi konfigurasi saat membuat instans DB. Opsi ini mencakup pengaturan untuk ketersediaan, keamanan, cadangan, dan pemeliharaan. Untuk membuat instans DB publik, Anda harus menggunakan Pembuatan Standar. Untuk informasi, lihat Membuat instans DB Amazon RDS.

Topik

Prasyarat

Sebelum memulai, selesaikan langkah-langkah di bagian berikut:

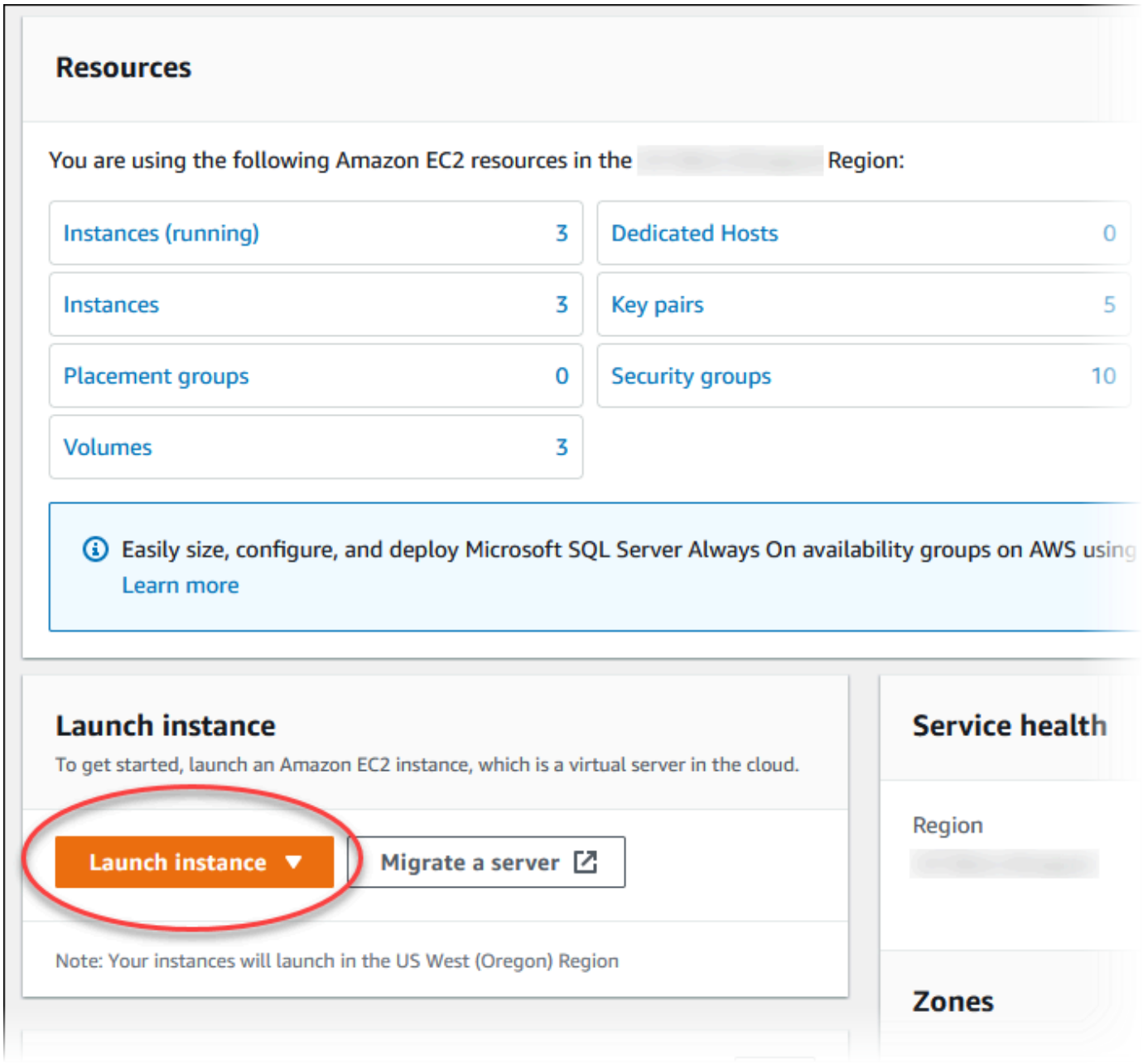
- Mendaftar Akun AWS
- Membuat pengguna administratif

Langkah 1: Buat instans EC2

Buat instans Amazon EC2 yang akan Anda gunakan untuk menghubungkan ke basis data Anda.

Untuk membuat instans EC2

1. Masuk ke AWS Management Console dan buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di sudut kanan atas AWS Management Console, pilih Wilayah AWS di mana Anda ingin membuat instans EC2.
3. Pilih Dasbor EC2, lalu pilih Luncurkan instans seperti yang ditampilkan dalam gambar berikut.

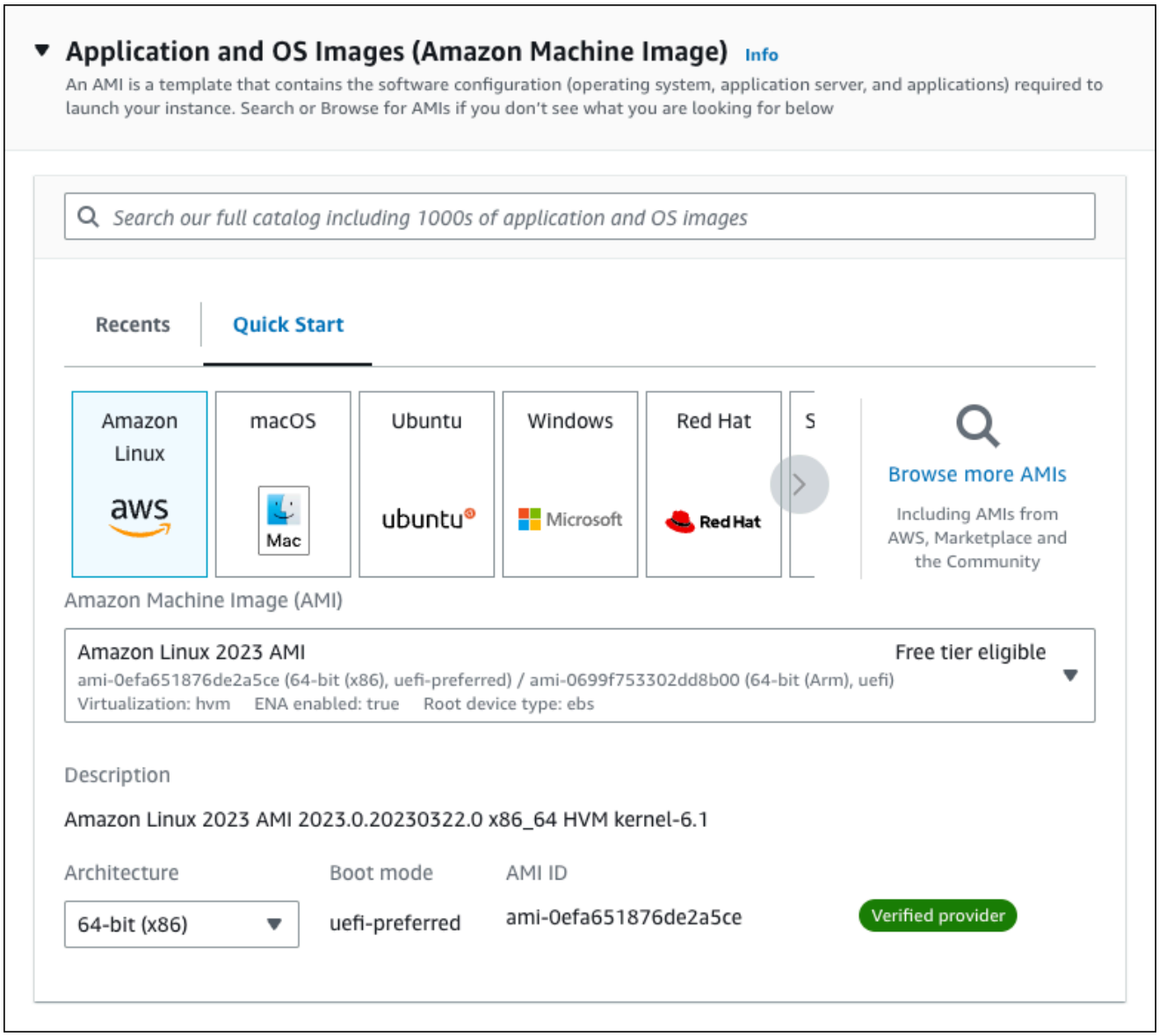


Halaman Meluncurkan instans akan terbuka.

4. Pilih pengaturan berikut di halaman Meluncurkan instans.

   a. Di bagian Nama dan tag, untuk Nama, masukkan **`ec2-database-connect`**.

   b. Di bagian Gambar Aplikasi dan OS (Amazon Machine Image), pilih Amazon Linux, lalu pilih AMI Amazon Linux 2023. Biarkan default untuk pilihan lainnya.



   c. Di bagian Jenis instans, pilih t2.micro.

   d. Di bagian Pasangan kunci (login), pilih nama Pasangan kunci untuk menggunakan pasangan kunci yang ada. Untuk membuat pasangan kunci baru untuk instans Amazon EC2, pilih Buat Pasangan kunci baru lalu gunakan jendela Buat pasangan kunci untuk membuatnya.

Untuk informasi selengkapnya tentang membuat pasangan kunci baru, lihat Membuat pasangan kunci di Panduan Pengguna Amazon EC2 untuk Instans Linux.

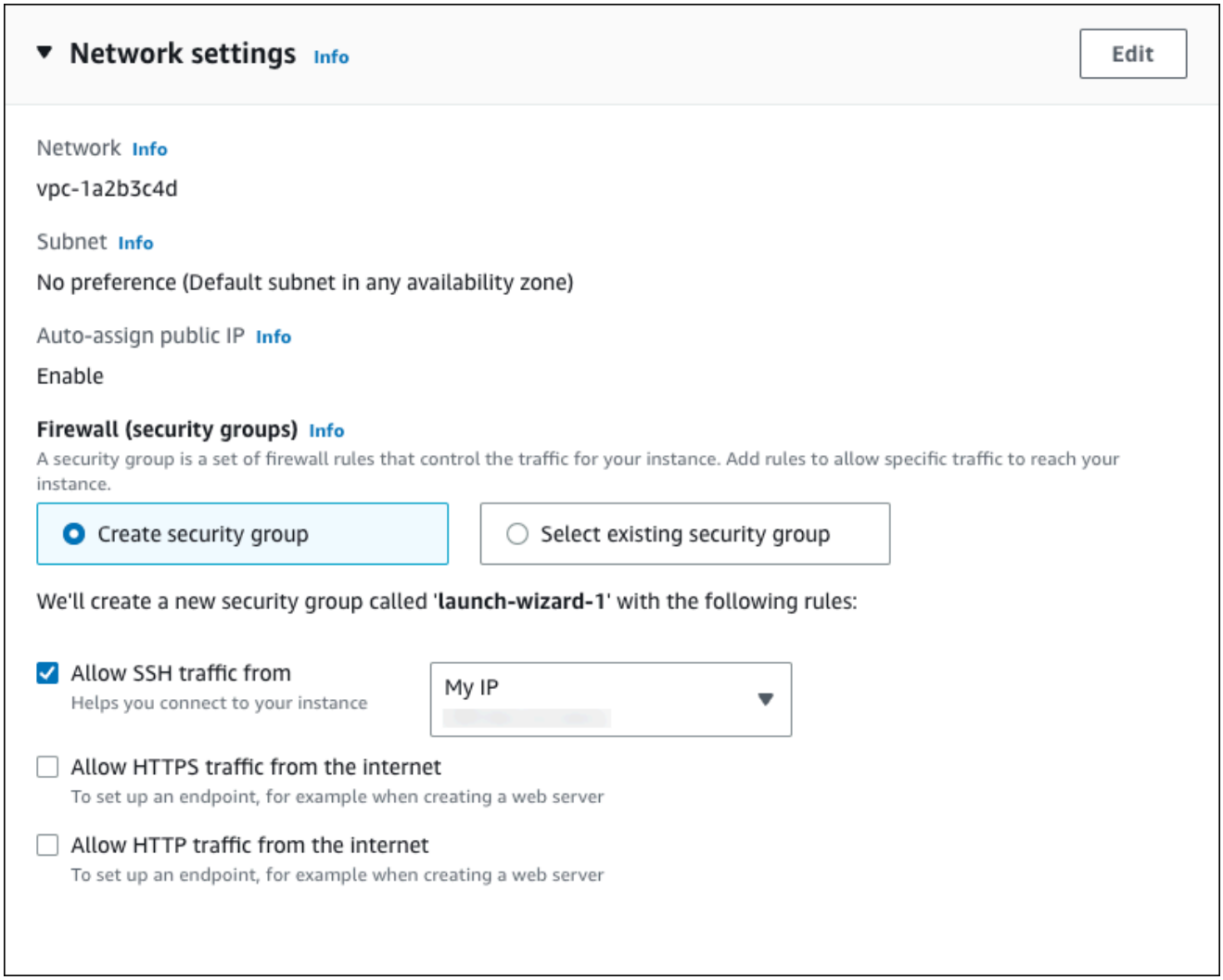
e. Untuk Izinkan lalu lintas SSH di Pengaturan jaringan, pilih sumber koneksi SSH ke instans EC2.

Anda dapat memilih IP Saya jika alamat IP yang ditampilkan benar untuk koneksi SSH. Jika tidak, Anda dapat menentukan alamat IP yang akan digunakan untuk menghubungkan ke instans EC2 di VPC Anda menggunakan Secure Shell (SSH). Untuk menentukan alamat IP publik Anda, Anda dapat membuka layanan di https://checkip.amazonaws.com di jendela atau tab browser lain. Contoh alamat IP adalah 192.0.2.1/32.

Dalam banyak kasus, Anda dapat menghubungkan melalui penyedia layanan Internet (ISP) atau dari belakang firewall Anda tanpa alamat IP statis. Jika demikian, tentukan rentang alamat IP yang digunakan oleh komputer klien.

> ⚠ Warning
>
> Jika menggunakan `0.0.0.0/0` untuk akses SSH, Anda memungkinkan semua alamat IP untuk mengakses instans publik EC2 Anda menggunakan SSH. Hal ini dapat diterima untuk waktu yang singkat di lingkungan pengujian, tetapi tidak aman untuk lingkungan produksi. Dalam produksi, Anda hanya dapat memberikan otorisasi pada alamat IP atau rentang alamat tertentu saja untuk mengakses instans EC2 Anda menggunakan SSH.

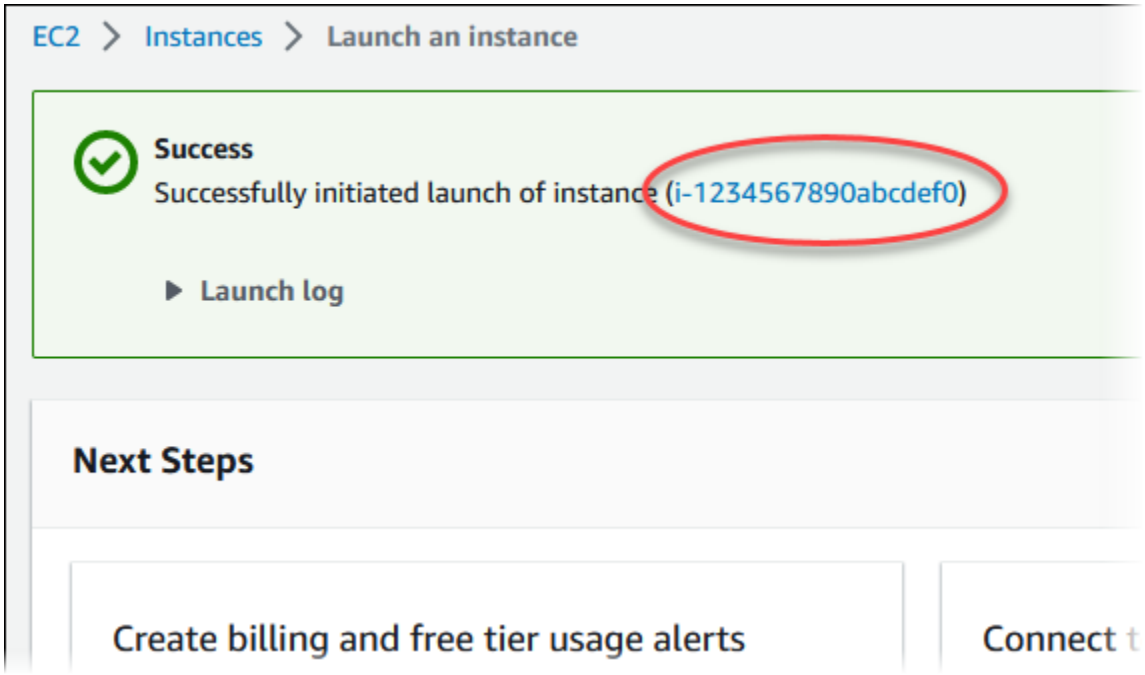
Gambar berikut menunjukkan contoh bagian Pengaturan jaringan.



f. Biarkan nilai default untuk bagian yang lainnya.

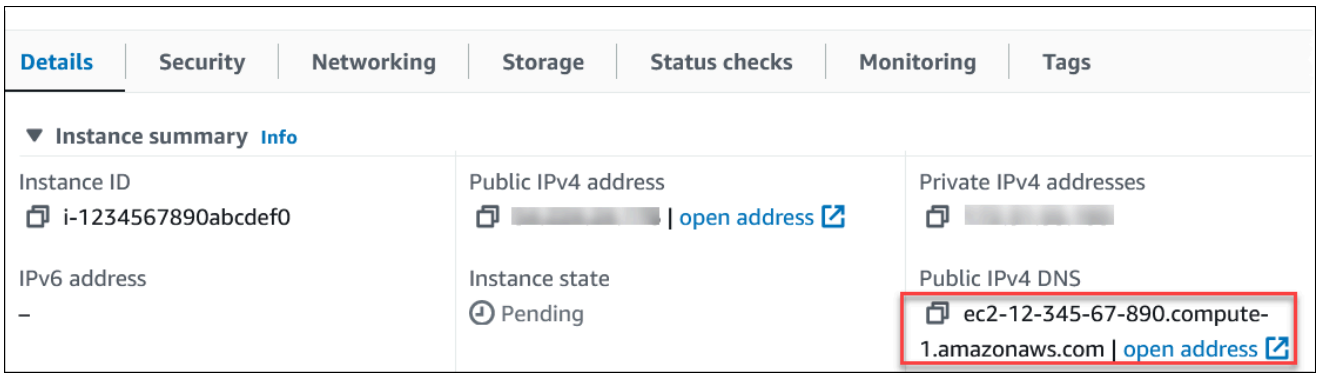
g. Tinjau ringkasan konfigurasi instans EC2 Anda di panel Ringkasan, dan setelah Anda siap, pilih Luncurkan instans.

5. Di halaman Status Peluncuran, catat pengidentifikasi untuk instans EC2 baru Anda, misalnya: `i-1234567890abcdef0`.

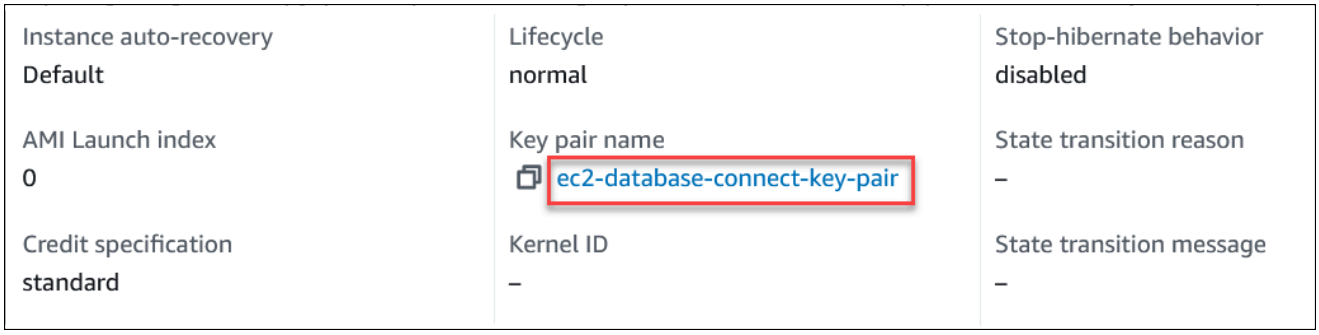


6. Pilih pengidentifikasi instans EC2 untuk membuka daftar instans EC2, lalu pilih instans EC2 Anda.
7. Di tab Detail, catat nilai-nilai berikut, yang akan Anda butuhkan saat menghubungkan menggunakan SSH:

   a. Di Ringkasan instans, catat nilai untuk DNS IPv4 Publik.



   b. Di Detail instans, catat nilai untuk Nama pasangan kunci.



8. Tunggu hingga Status instance untuk instans EC2 Anda berstatus Berjalan sebelum melanjutkan.

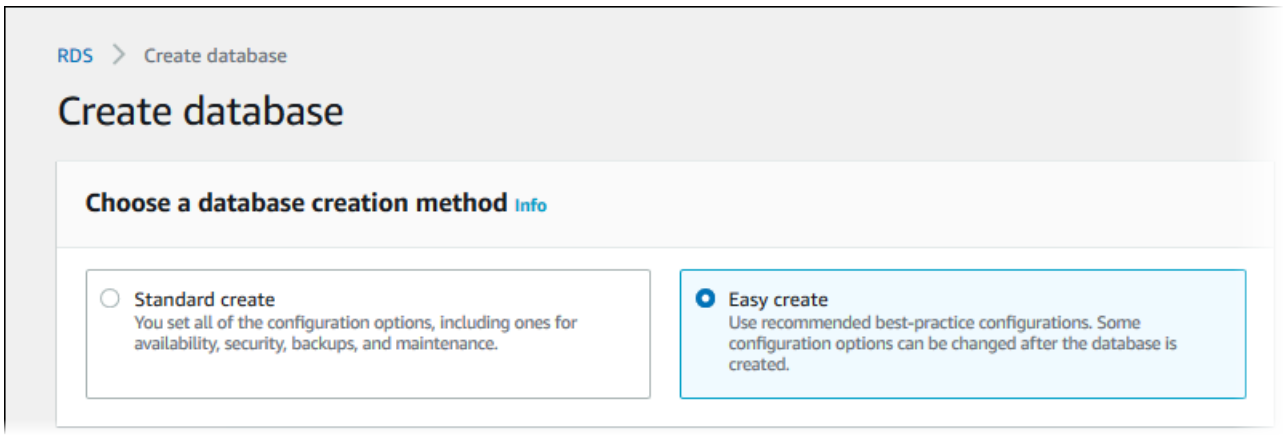
Langkah 2: Buat instans DB PostgreSQL

Blok bangunan dasar Amazon RDS adalah instans DB. Lingkungan ini adalah tempat Anda menjalankan basis data PostgreSQL Anda.

Dalam contoh ini, Anda menggunakan Pembuatan Mudah untuk membuat instans DB yang menjalankan mesin basis data PostgreSQL dengan kelas instans DB db.t3.micro.

Untuk membuat instans DB PostgreSQL dengan Pembuatan Mudah

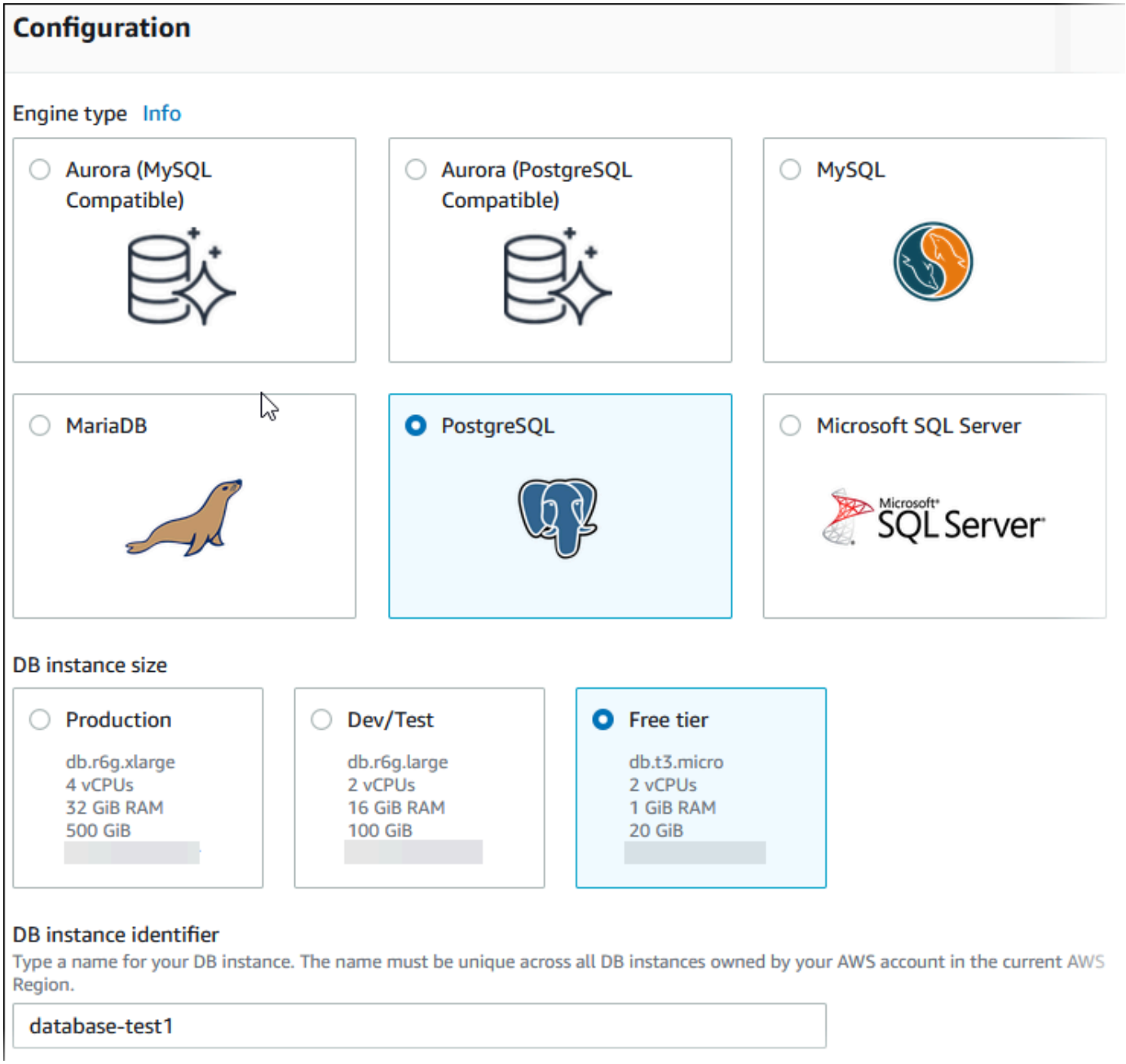
1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di sudut kanan atas konsol Amazon RDS, pilih AWS Wilayah tempat Anda ingin membuat instans DB.
3. Di panel navigasi, pilih Basis Data.
4. Pilih Buat basis data dan pastikan Pembuatan Mudah dipilih.



5. Di Konfigurasi, pilih PostgreSQL.

6. Untuk Ukuran instans DB, pilih Tingkat gratis.
7. Untuk Pengidentifikasi instans DB, masukkan **`database-test1`**.
8. Untuk Nama pengguna utama, masukkan nama untuk pengguna utama, atau tetap gunakan nama default (**`postgres`**).

   Tampilan halaman Membuat basis data seperti gambar berikut.

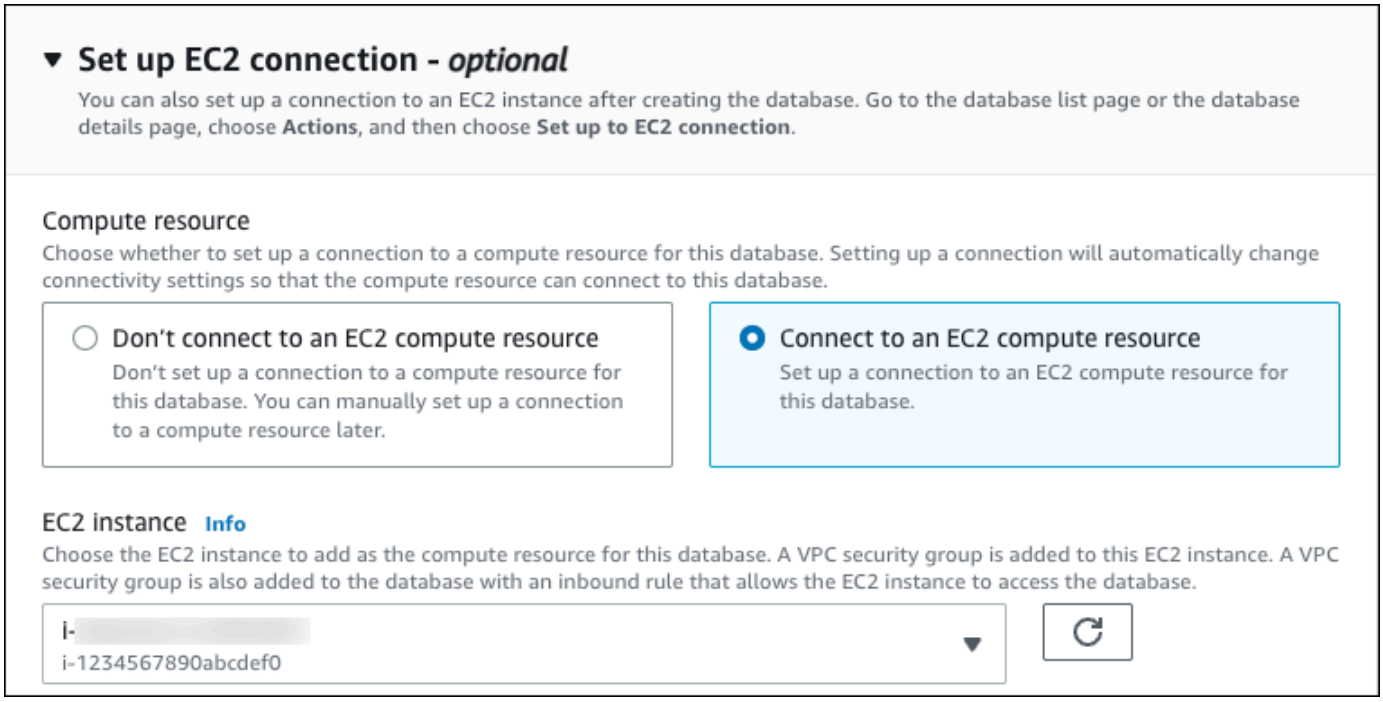


9. Untuk menggunakan kata sandi utama yang dibuat secara otomatis untuk instans DB, pilih Buat kata sandi secara otomatis.

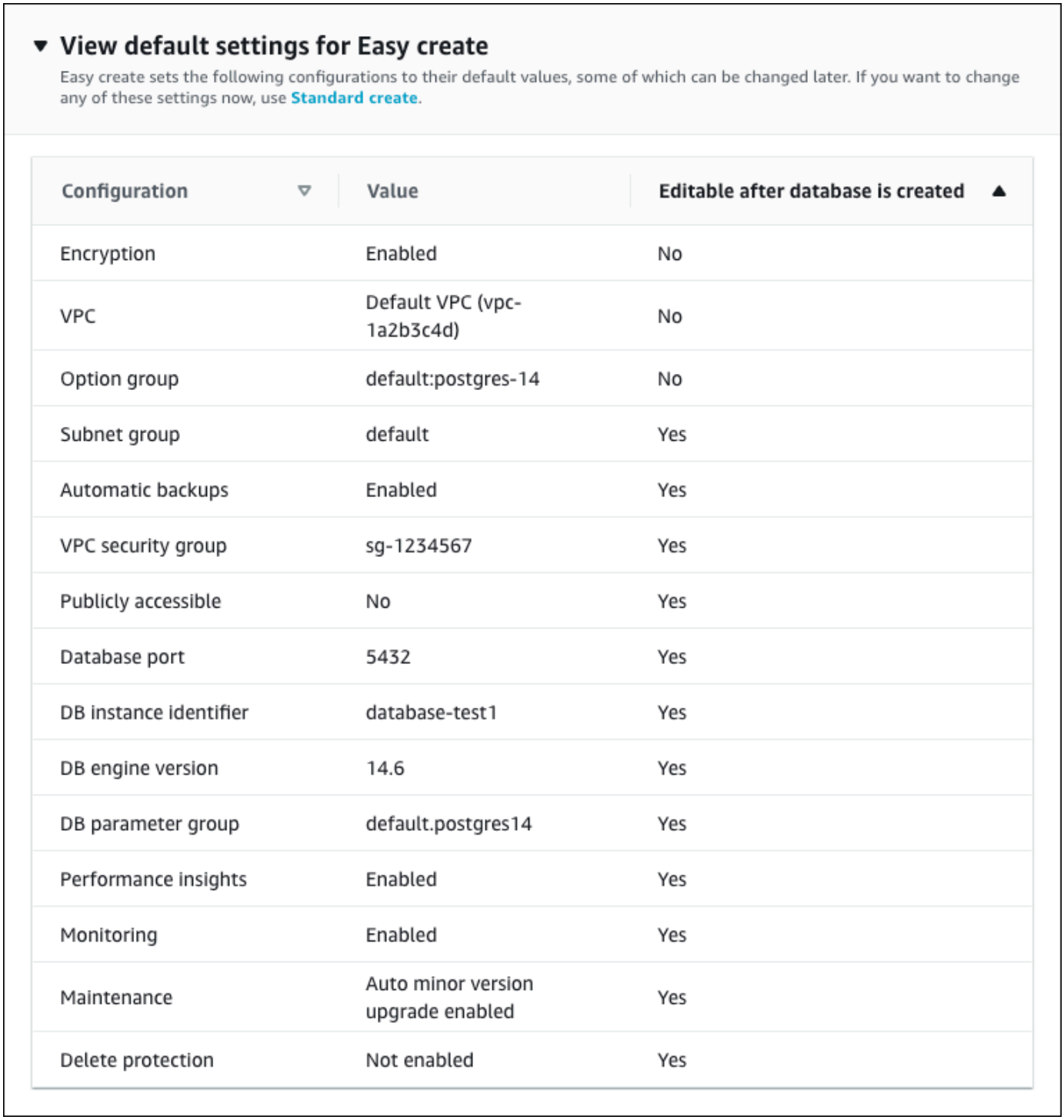
Untuk memasukkan kata sandi utama Anda, hapus centang pada Buat kata sandi secara otomatis, lalu masukkan kata sandi yang sama dalam Kata sandi utama dan Konfirmasi kata sandi.

10. Untuk menyiapkan koneksi dengan instans EC2 yang Anda buat sebelumnya, buka Menyiapkan koneksi EC2 - opsional.

    Pilih Hubungkan ke sumber daya komputasi EC2. Pilih instans EC2 yang Anda buat sebelumnya.



11. Buka Lihat pengaturan default untuk Pembuatan Mudah.



View default settings for Easy create

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

Configuration	Value	Editable after database is created
Encryption	Enabled	No
VPC	Default VPC (vpc-1a2b3c4d)	No
Option group	default:postgres-14	No
Subnet group	default	Yes
Automatic backups	Enabled	Yes
VPC security group	sg-1234567	Yes
Publicly accessible	No	Yes
Database port	5432	Yes
DB instance identifier	database-test1	Yes
DB engine version	14.6	Yes
DB parameter group	default.postgres14	Yes
Performance insights	Enabled	Yes
Monitoring	Enabled	Yes
Maintenance	Auto minor version upgrade enabled	Yes
Delete protection	Not enabled	Yes

Anda dapat memeriksa pengaturan default yang digunakan dengan Pembuatan mudah. Kolom Dapat diedit setelah basis data dibuat menunjukkan opsi yang dapat Anda ubah setelah membuat basis data.

- Jika pengaturan memiliki Tidak di kolom tersebut, dan Anda menginginkan pengaturan yang berbeda, Anda dapat menggunakan Pembuatan Standar untuk membuat instans DB.
- Jika pengaturan memiliki Ya di kolom tersebut, dan Anda menginginkan pengaturan yang berbeda, Anda dapat menggunakan Pembuatan Standar untuk membuat instans DB, atau mengubah instans DB setelah Anda membuatnya untuk mengubah pengaturan.

12. Pilih Buat basis data.

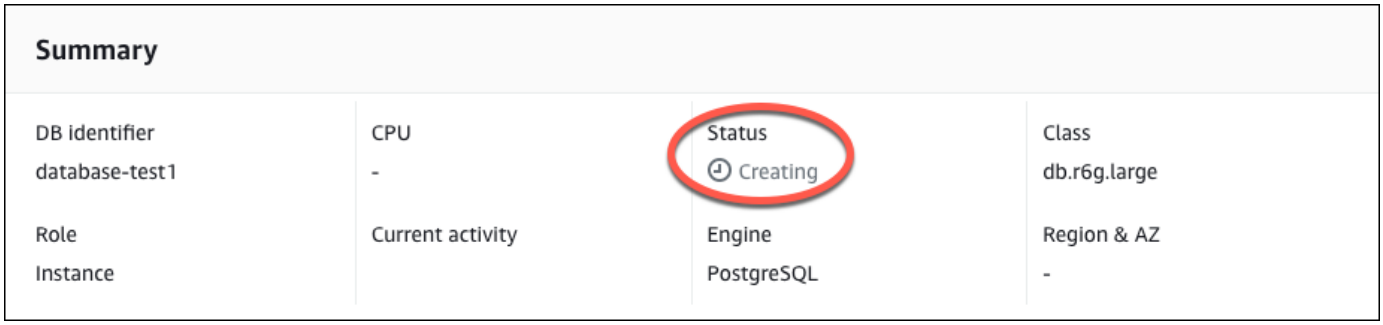
    Untuk melihat nama pengguna dan kata sandi utama untuk instans DB, pilih Lihat detail kredensial.

    Anda dapat menggunakan nama pengguna dan kata sandi yang ditampilkan untuk terhubung ke instans DB sebagai pengguna utama.

    > **Important**
    >
    > Anda tidak dapat melihat kata sandi pengguna utama lagi. Jika tidak mencatatnya, Anda mungkin harus mengubahnya.
    > Jika perlu mengubah kata sandi pengguna utama setelah instans DB tersedia, Anda dapat mengubah instans DB untuk melakukannya. Untuk informasi selengkapnya tentang cara mengubah instans DB, lihat Memodifikasi instans DB Amazon RDS.

13. Dalam daftar Basis Data, pilih nama instans DB PostgreSQL yang baru untuk menampilkan detailnya.

    Instans DB memiliki status Membuat hingga siap digunakan.



    Saat statusnya berubah menjadi Tersedia, Anda dapat terhubung ke instans DB. Tergantung pada kelas instans DB dan jumlah penyimpanan, diperlukan waktu hingga 20 menit sebelum instans baru tersedia.

(Opsional) Buat instance VPC, EC2, dan PostgreSQL menggunakan AWS CloudFormation

Alih-alih menggunakan konsol untuk membuat instance VPC, EC2, dan PostgreSQL, Anda dapat menggunakannya AWS CloudFormation untuk menyediakan sumber daya dengan memperlakukan infrastruktur sebagai kode. AWS Untuk membantu Anda mengatur AWS sumber daya Anda menjadi unit yang lebih kecil dan lebih mudah dikelola, Anda dapat menggunakan fungsionalitas tumpukan AWS CloudFormation bersarang. Untuk informasi selengkapnya, lihat Membuat tumpukan di AWS CloudFormation konsol dan Bekerja dengan tumpukan bersarang.

> **Important**
>
> AWS CloudFormation gratis, tetapi sumber daya yang CloudFormation menciptakan hidup. Anda dikenakan biaya penggunaan standar untuk sumber daya ini sampai Anda menghentikannya. Total biaya akan minimal. Untuk informasi tentang cara meminimalkan biaya apa pun, buka Tingkat AWS Gratis.

Untuk membuat sumber daya menggunakan AWS CloudFormation konsol, selesaikan langkah-langkah berikut:

- Langkah 1: Unduh CloudFormation template
- Langkah 2: Konfigurasikan sumber daya Anda menggunakan CloudFormation

Unduh CloudFormation template

CloudFormation Template adalah file teks JSON atau YAMAL yang berisi informasi konfigurasi tentang sumber daya yang ingin Anda buat di tumpukan. Template ini juga membuat VPC dan host bastion untuk Anda bersama dengan instance RDS.

Untuk men-download file template, buka link berikut, PostgreSQL CloudFormation template.

Di halaman Github, klik tombol Unduh file mentah untuk menyimpan file YAMAL template.

Konfigurasikan sumber daya Anda menggunakan CloudFormation

> **Note**
>
> Sebelum memulai proses ini, pastikan Anda memiliki pasangan Kunci untuk instans EC2 di. Akun AWS Untuk informasi selengkapnya, lihat Pasangan kunci Amazon EC2 dan instans Linux.

Ketika Anda menggunakan AWS CloudFormation template, Anda harus memilih parameter yang benar untuk memastikan sumber daya Anda dibuat dengan benar. Ikuti langkah-langkah di bawah ini:

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol di https://console.aws.amazon.com/cloudformation.
2. Pilih Buat tumpukan.
3. Di bagian Tentukan templat, pilih Unggah file templat dari komputer Anda, lalu pilih Berikutnya.
4. Di halaman Tentukan detail tumpukan, atur parameter berikut:
   a. Tetapkan nama Stack ke PostgreSQL TestStack.
   b. Di bawah Parameter, atur Availability Zone dengan memilih tiga zona ketersediaan.
   c. Di bawah konfigurasi Linux Bastion Host, untuk Key Name, pilih key pair untuk login ke instans EC2 Anda.
   d. Dalam pengaturan konfigurasi Linux Bastion Host, atur rentang IP yang Diizinkan ke alamat IP Anda. Untuk menyambung ke instans EC2 di VPC Anda menggunakan Secure Shell (SSH), tentukan alamat IP publik Anda menggunakan layanan di https://checkip.amazonaws.com. Contoh alamat IP adalah 192.0.2.1/32.

      > **Warning**
      >
      > Jika menggunakan `0.0.0.0/0` untuk akses SSH, Anda memungkinkan semua alamat IP untuk mengakses instans publik EC2 Anda menggunakan SSH. Hal ini dapat diterima untuk waktu yang singkat di lingkungan pengujian, tetapi tidak aman untuk lingkungan produksi. Dalam produksi, Anda hanya dapat memberikan otorisasi pada alamat IP atau rentang alamat tertentu saja untuk mengakses instans EC2 Anda menggunakan SSH.

   e. Di bawah konfigurasi Database General, atur kelas instance Database ke db.t3.micro.

f. Tetapkan nama Database ke**`database-test1`**.

g. Untuk nama pengguna master Database, masukkan nama untuk pengguna master.

h. Atur Kelola kata sandi pengguna master DB dengan Secrets Manager `false` untuk tutorial ini.

i. Untuk kata sandi Database, tetapkan kata sandi pilihan Anda. Ingat kata sandi ini untuk langkah lebih lanjut dalam tutorial.

j. Di bawah konfigurasi Penyimpanan Database, atur tipe penyimpanan Database ke gp2.

k. Di bawah konfigurasi Pemantauan Database, atur Aktifkan Performance Insights RDS ke false.

l. Biarkan semua pengaturan lainnya sebagai nilai default. Klik Berikutnya untuk melanjutkan.

5. Di halaman Configure stack options, tinggalkan semua opsi default. Klik Berikutnya untuk melanjutkan.
6. Di halaman tumpukan Tinjauan, pilih Kirim setelah memeriksa database dan opsi host bastion Linux.

Setelah proses pembuatan tumpukan selesai, lihat tumpukan dengan nama BastionStackdan RDSNS untuk mencatat informasi yang Anda butuhkan untuk terhubung ke database. Untuk informasi selengkapnya, lihat Melihat data AWS CloudFormation tumpukan dan sumber daya di AWS Management Console.

Langkah 3: Hubungkan ke instans DB PostgreSQL

Anda dapat menghubungkan ke instans DB menggunakan pgadmin atau psql. Contoh ini menjelaskan cara menghubungkan ke instans DB PostgreSQL menggunakan klien baris perintah psql.

Untuk menghubungkan ke instans DB PostgreSQL menggunakan psql

1. Temukan titik akhir (nama DNS) dan nomor port untuk instans DB Anda.

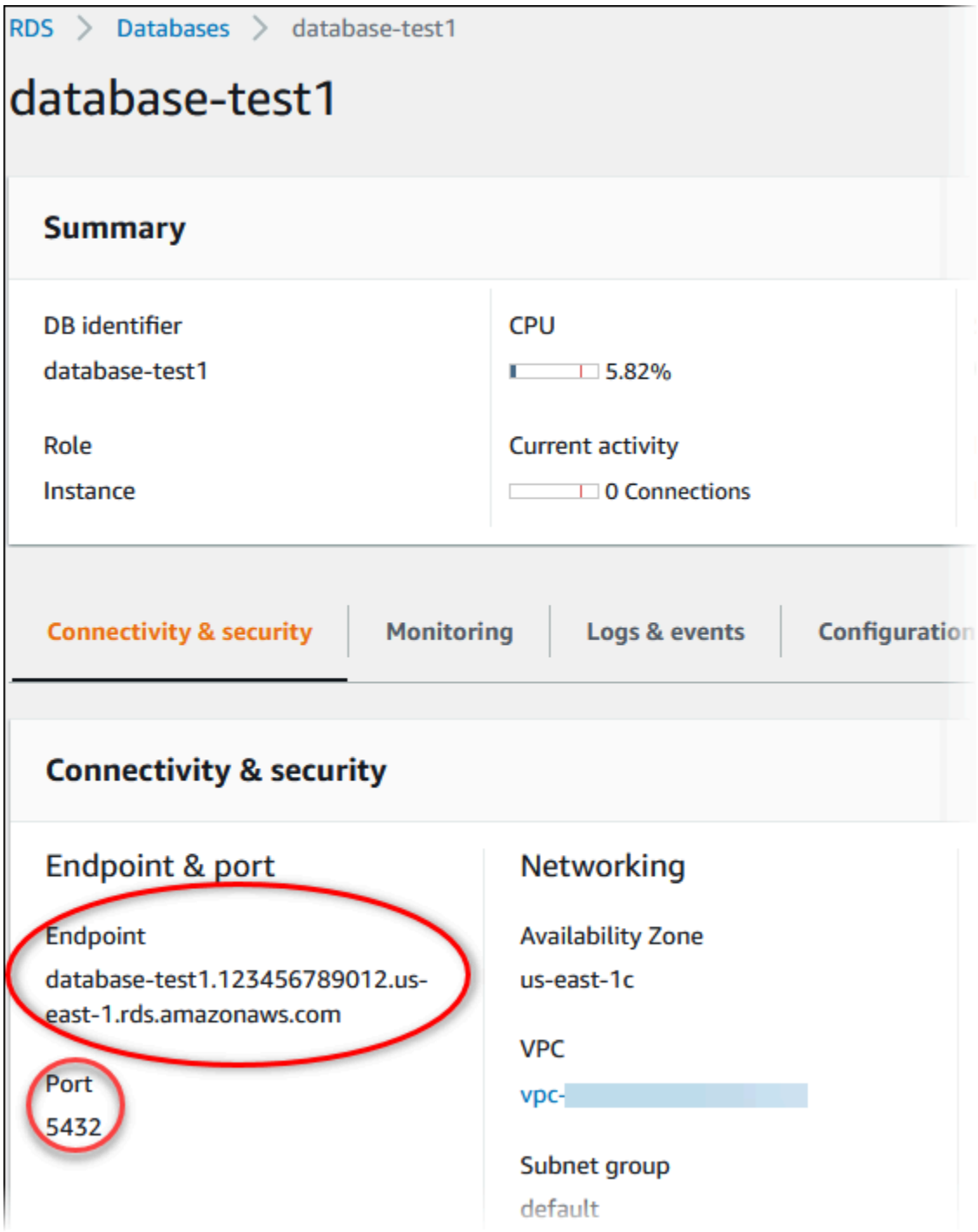
   a. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.

   b. Di sudut kanan atas konsol Amazon RDS, pilih Wilayah AWS untuk instans DB.

   c. Di panel navigasi, pilih Basis Data.

   d. Pilih nama instans DB PostgreSQL untuk menampilkan detailnya.

   e. Di tab Konektivitas & keamanan, salin titik akhir. Perhatikan juga nomor port. Anda memerlukan titik akhir dan nomor port untuk terhubung ke instans DB.



2. Hubungkan ke instans EC2 yang Anda buat sebelumnya dengan mengikuti langkah-langkah di Menghubungkan ke instans Linux di Panduan Pengguna Amazon EC2 untuk Instans Linux.

   Sebaiknya Anda menghubungkan ke instans EC2 menggunakan SSH. Jika utilitas klien SSH diinstal di Windows, Linux, atau Mac, Anda dapat menghubungkan ke instans menggunakan format perintah berikut:

```
ssh -i location_of_pem_file ec2-user@ec2-instance-public-dns-name
```

Misalnya, asumsikan bahwa `ec2-database-connect-key-pair.pem` disimpan di `/dir1` di Linux, dan DNS IPv4 publik untuk instans EC2 Anda adalah `ec2-12-345-678-90.compute-1.amazonaws.com`. Perintah SSH Anda akan tampak seperti berikut:

```
ssh -i /dir1/ec2-database-connect-key-pair.pem ec2-
user@ec2-12-345-678-90.compute-1.amazonaws.com
```

3. Dapatkan pembaruan keamanan dan perbaikan bug terbaru dengan memperbarui perangkat lunak di instans EC2 Anda. Untuk melakukannya, gunakan perintah berikut.

> **Note**
>
> Opsi `-y` menginstal pembaruan tanpa meminta konfirmasi. Hilangkan opsi ini untuk memeriksa pembaruan sebelum menginstal.

```
sudo dnf update -y
```

4. Untuk mengnstal klien baris perintah psql dari PostgreSQL di Amazon Linux 2023, jalankan perintah berikut:

```
sudo dnf install postgresql15
```

5. Hubungkan ke instans DB PostgreSQL. Misalnya, masukkan perintah berikut pada prompt perintah di komputer klien. TIndakan ini memungkinkan Anda terhubung ke instans DB PostgreSQL menggunakan klien psql.

   Ganti titik akhir instans DB (nama DNS) untuk *`endpoint`*, ganti nama basis data `--dbname` yang ingin Anda hubungkan untuk *`postgres`*, dan ganti nama pengguna utama yang Anda gunakan untuk *`postgres`*. Masukkan kata sandi utama yang Anda gunakan saat dimintai kata sandi.

```
psql --host=endpoint --port=5432 --dbname=postgres --username=postgres
```

Setelah memasukkan kata sandi untuk pengguna, Anda akan melihat output yang serupa dengan yang berikut ini.

```
psql (14.3, server 14.6)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256,
 compression: off)
Type "help" for help.

postgres=>
```

Untuk informasi selengkapnya tentang cara menghubungkan ke instans DB PostgreSQL, lihat Menghubungkan ke instans DB yang menjalankan mesin basis data PostgreSQL. Jika Anda tidak dapat terhubung ke instans DB Anda, lihat Memecahkan masalah koneksi ke instans RDS for PostgreSQL Anda.

Untuk keamanan, praktik terbaiknya adalah menggunakan koneksi terenkripsi. Hanya gunakan koneksi PostgreSQL yang tidak terenkripsi saat klien dan server berada di VPC yang sama dan jaringan tepercaya. Untuk informasi tentang cara menggunakan koneksi terenkripsi, lihat Menghubungkan ke instans DB PostgreSQL melalui SSL.

6. Jalankan perintah SQL.

   Misalnya, perintah SQL berikut menunjukkan tanggal dan waktu saat ini:

   ```
   SELECT CURRENT_TIMESTAMP;
   ```

Langkah 4: Hapus instans EC2 dan instans DB

Setelah Anda terhubung ke dan menjelajahi instans EC2 dan instans DB sampel yang Anda buat, hapus instans tersebut sehingga Anda tidak lagi dikenakan biaya untuk instans DB tersebut.

Jika Anda biasa AWS CloudFormation membuat sumber daya, lewati langkah ini dan lanjutkan ke langkah berikutnya.

Untuk menghapus instans EC2

1. Masuk ke AWS Management Console dan buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Instans.

3. Pilih instans EC2, dan pilih Status instans, Akhiri instans.
4. Pilih Akhiri saat diminta untuk konfirmasi.

Untuk informasi selengkapnya tentang menghapus instans EC2, lihat Mengakhiri Instans dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Untuk menghapus instans DB tanpa snapshot DB akhir

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data.
3. Pilih instans DB yang ingin Anda hapus.
4. Untuk Tindakan, pilih Hapus.
5. Hapus Buat snapshot akhir? dan Pertahankan pencadangan otomatis.
6. Lengkapi pengakuan dan pilih Hapus.

(Opsional) Hapus instans EC2 dan instans DB yang dibuat dengan CloudFormation

Jika Anda biasa AWS CloudFormation membuat sumber daya, hapus CloudFormation tumpukan setelah Anda terhubung dan jelajahi contoh instans EC2 dan instans DB, sehingga Anda tidak lagi dikenakan biaya untuk itu.

Untuk menghapus sumber CloudFormation daya

1. Buka AWS CloudFormation konsol.
2. Pada halaman Stacks di CloudFormationconsole, pilih tumpukan root (tumpukan tanpa nama VPCStack, BastionStack atau RDSNS).
3. Pilih Hapus.
4. Pilih Hapus tumpukan saat diminta konfirmasi.

Untuk informasi selengkapnya tentang menghapus tumpukan CloudFormation, lihat Menghapus tumpukan di AWS CloudFormation konsol di AWS CloudFormation Panduan Pengguna.

(Opsional) Menghubungkan instans DB Anda ke fungsi Lambda

Anda juga dapat menghubungkan instans DB RDS for PostgreSQL ke sumber daya komputasi nirserver Lambda. Fungsi Lambda memungkinkan Anda menjalankan kode tanpa menyediakan atau mengelola infrastruktur. Fungsi Lambda juga memungkinkan Anda untuk otomatis merespons permintaan eksekusi kode pada skala apa pun, mulai dari selusin peristiwa dalam sehari hingga ratusan per detik. Lihat informasi yang lebih lengkap di Menghubungkan secara otomatis fungsi Lambda dan instans basis data.

Tutorial: Membuat server web dan instans DB Amazon RDS

Tutorial ini menunjukkan cara menginstal server web Apache dengan PHP dan membuat basis data MariaDB, MySQL, atau PostgreSQL. Server web berjalan di instans Amazon EC2 menggunakan Amazon Linux 2023, dan Anda dapat memilih antara instans DB MySQL atau PostgreSQL. Baik instans Amazon EC2 maupun instans DB berjalan di cloud privat virtual (VPC) berdasarkan layanan Amazon VPC.

Important

Pembuatan akun AWS tidak dikenakan biaya. Namun, dengan menyelesaikan tutorial ini, Anda mungkin akan dikenai biaya untuk sumber daya AWS yang Anda gunakan. Anda dapat menghapus sumber daya ini setelah menyelesaikan tutorial jika tidak diperlukan lagi.

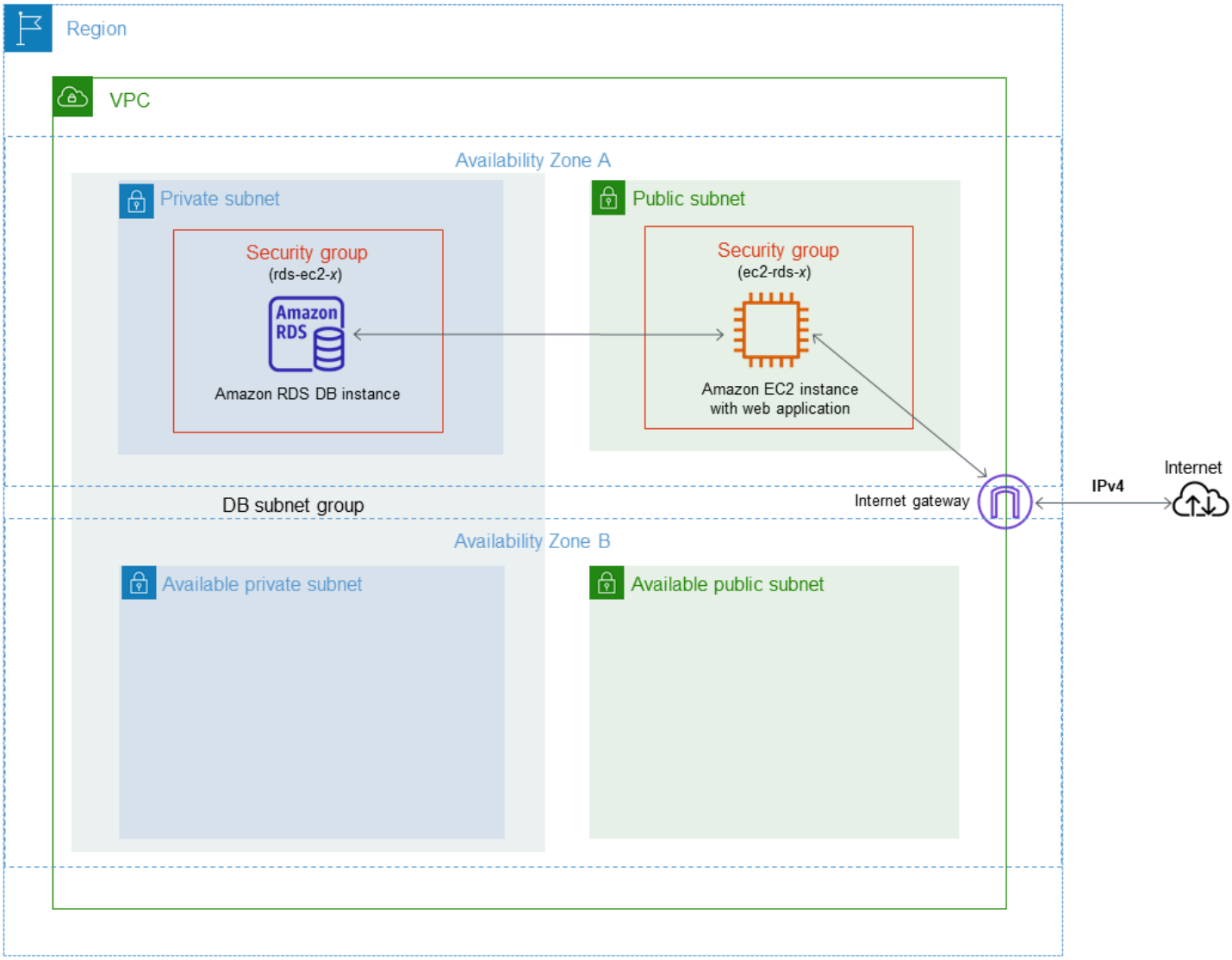
Note

Tutorial ini berfungsi dengan Amazon Linux 2023 dan mungkin tidak berfungsi untuk versi Linux lainnya.

Dalam tutorial berikut, Anda membuat instans EC2 yang menggunakan VPC, subnet, dan grup keamanan default untuk Akun AWS Anda. Tutorial ini menunjukkan cara membuat instans DB dan secara otomatis menyiapkan konektivitas dengan instans EC2 yang Anda buat. Tutorial kemudian menunjukkan cara menginstal server web pada instans EC2. Anda menghubungkan server web ke instans DB di VPC menggunakan titik akhir instans DB.

1. Meluncurkan instans EC2
2. Membuat instans DB Amazon RDS
3. Menginstal server web di instans EC2 Anda

Diagram berikut menunjukkan konfigurasi setelah tutorial selesai.



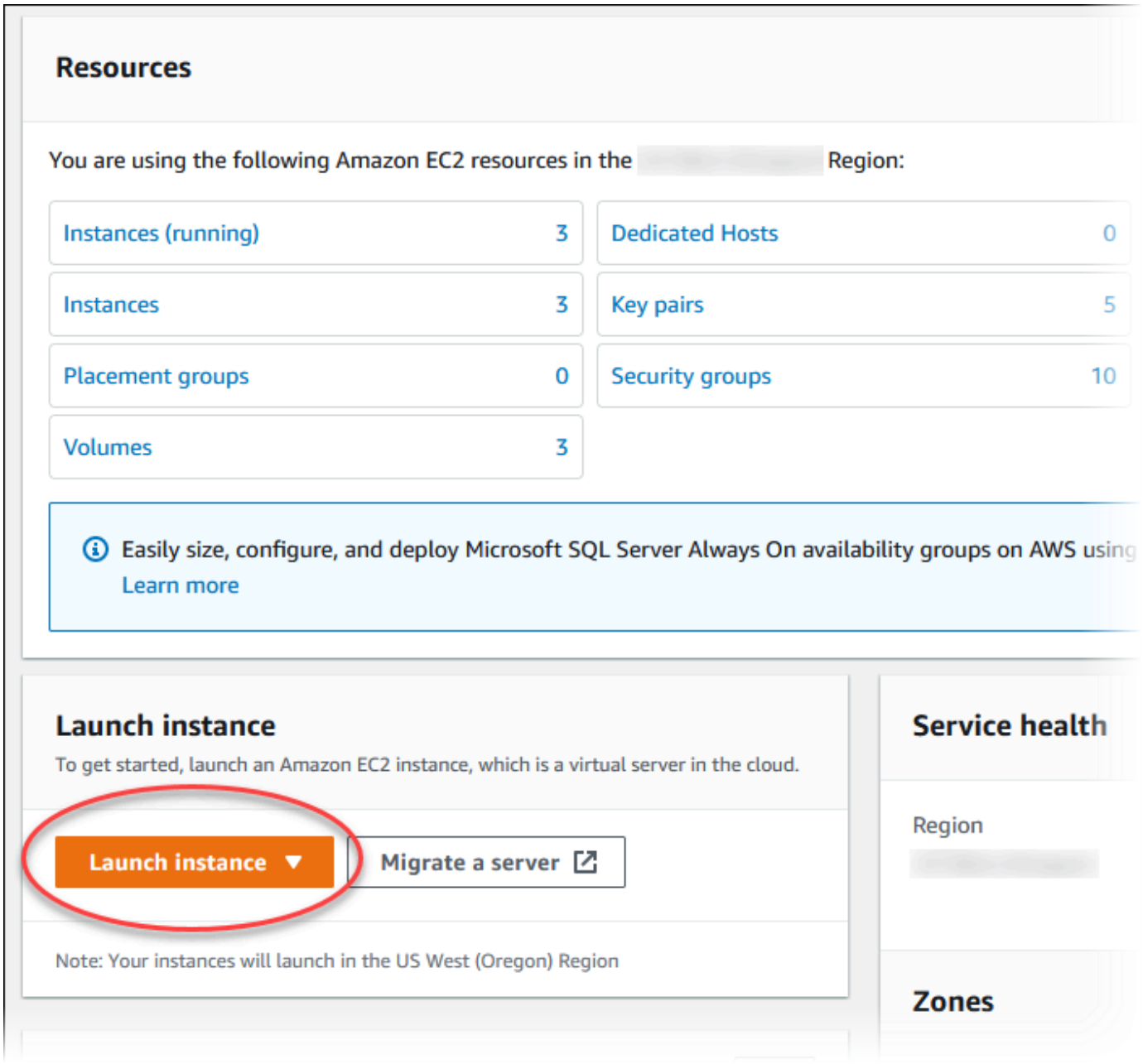
> **Note**
>
> Setelah Anda menyelesaikan tutorial, ada subnet publik dan privat di setiap Zona Ketersediaan di VPC Anda. Tutorial ini menggunakan VPC default untuk Akun AWS Anda dan secara otomatis menyiapkan konektivitas antara instans EC2 dan klaster DB. Jika Anda lebih suka mengonfigurasi VPC baru untuk skenario ini, selesaikan tugas di Tutorial: Membuat VPC untuk digunakan dengan instans DB (khusus IPv4).

Meluncurkan instans EC2

Buat instans Amazon EC2 di subnet publik VPC Anda.

Untuk meluncurkan instans EC2

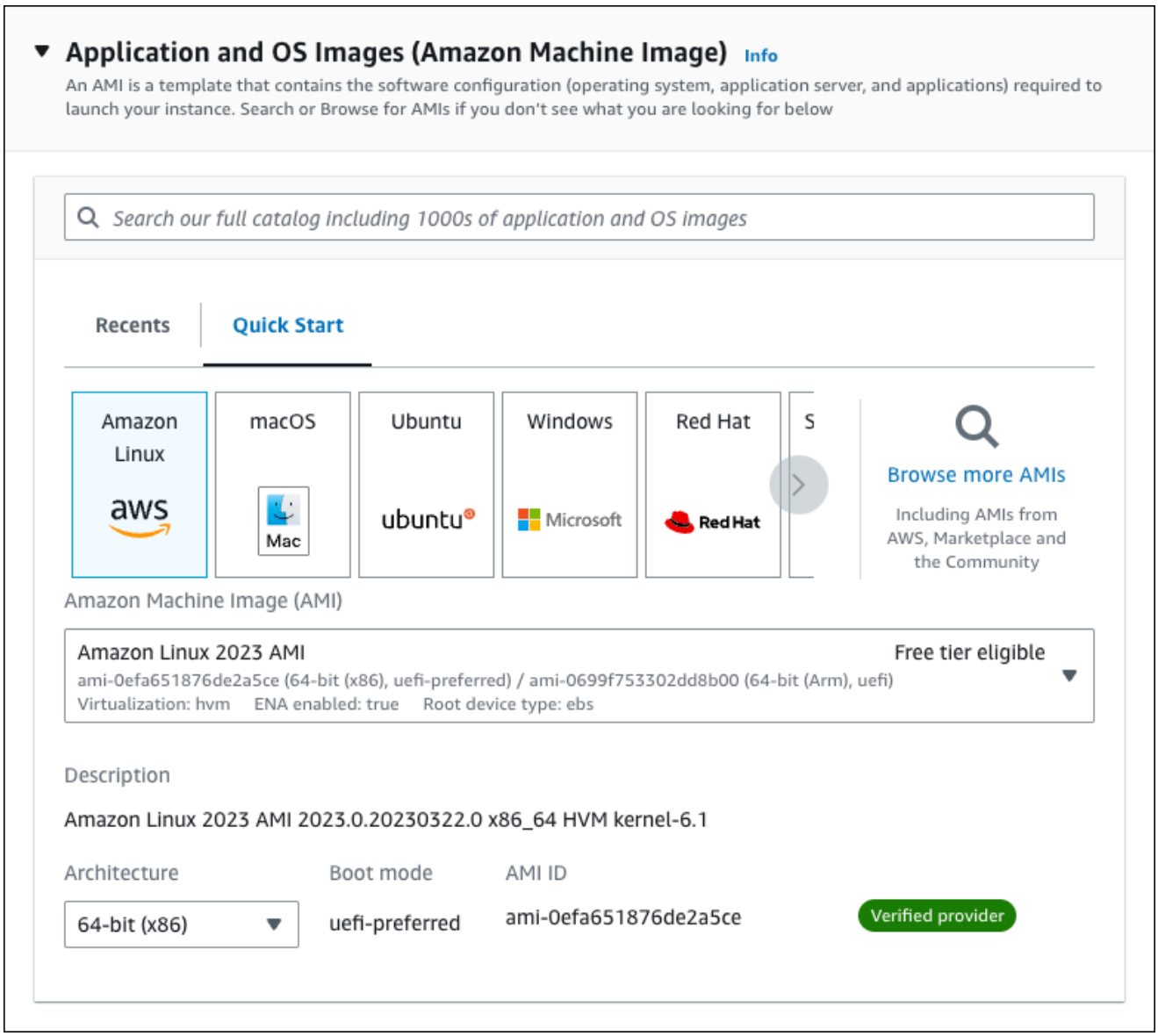
1. Masuk ke AWS Management Console dan buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di sudut kanan atas AWS Management Console, pilih Wilayah AWS tempat Anda akan membuat instans EC2.
3. Pilih Dasbor EC2, lalu pilih Luncurkan instans, seperti yang ditampilkan berikut ini.



4. Pilih pengaturan berikut di halaman Luncurkan instans.

a. Di bagian Nama dan tag, untuk Nama, masukkan **`tutorial-ec2-instance-web-server`**.

b. Di bagian Gambar Aplikasi dan OS (Amazon Machine Image), pilih Amazon Linux, lalu pilih AMI Amazon Linux 2023. Biarkan default untuk pilihan lain.



c. Di bagian Jenis instans, pilih t2.micro.

d. Di bagian Pasangan kunci (login), pilih nama Pasangan kunci untuk menggunakan pasangan kunci yang ada. Untuk membuat pasangan kunci baru untuk instans Amazon EC2, pilih Buat Pasangan kunci baru lalu gunakan jendela Buat pasangan kunci untuk membuatnya.

Untuk informasi selengkapnya tentang membuat pasangan kunci baru, lihat Membuat pasangan kunci di Panduan Pengguna Amazon EC2 untuk Instans Linux.

e. Di bagian Pengaturan jaringan, atur nilai-nilai ini dan biarkan nilai-nilai lainnya sebagai default:

- Untuk Izinkan lalu lintas SSH dari, pilih sumber koneksi SSH ke instans EC2.

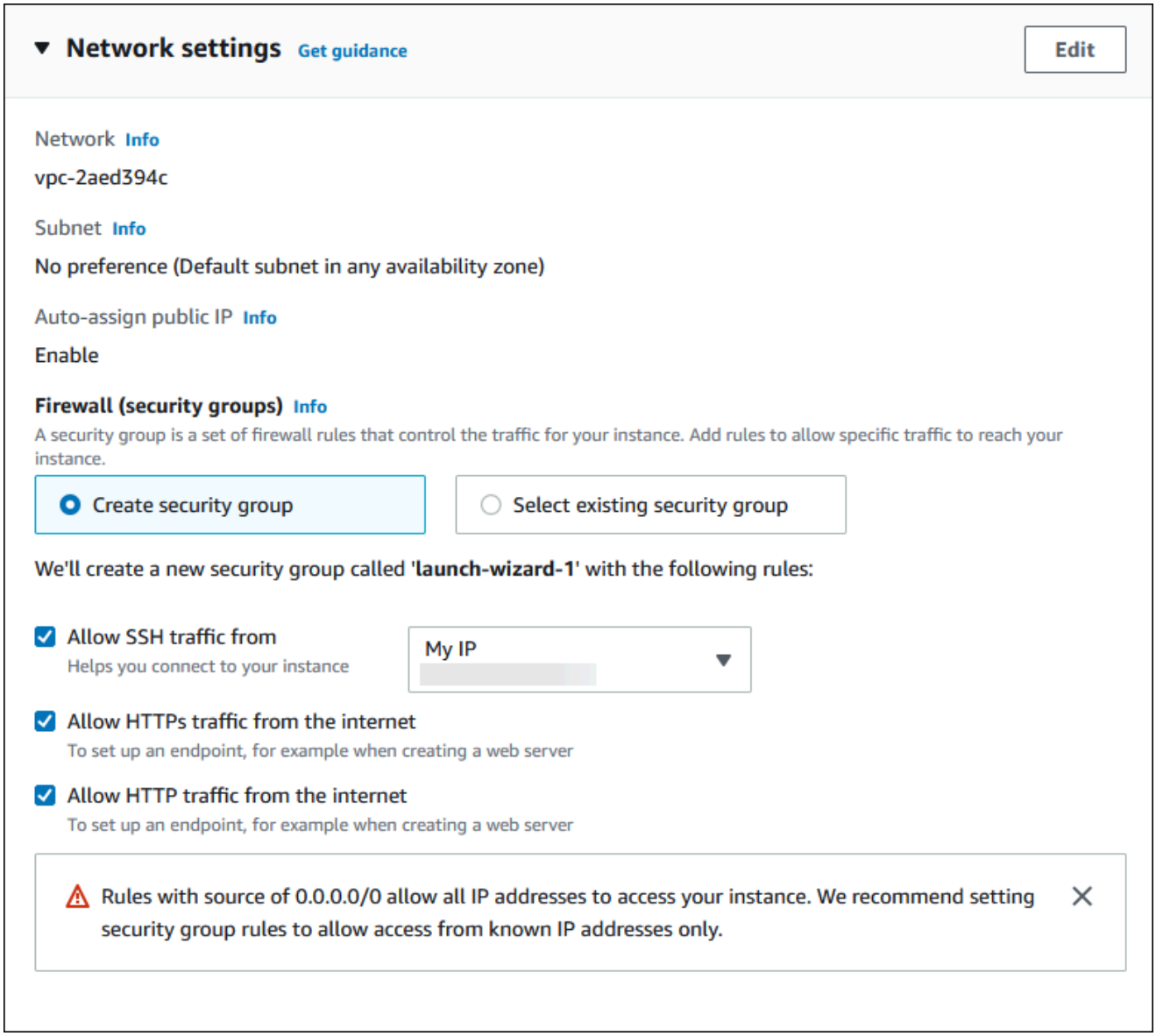
  Anda dapat memilih IP Saya jika alamat IP yang ditampilkan benar untuk koneksi SSH.

  Jika tidak, Anda dapat menentukan alamat IP yang akan digunakan untuk menghubungkan ke instans EC2 di VPC Anda menggunakan Secure Shell (SSH). Untuk menentukan alamat IP publik Anda, Anda dapat membuka layanan di https://checkip.amazonaws.com di jendela atau tab browser lain. Contoh alamat IP adalah `203.0.113.25/32`.

  Dalam banyak kasus, Anda dapat menghubungkan melalui penyedia layanan Internet (ISP) atau dari belakang firewall Anda tanpa alamat IP statis. Jika demikian, tentukan rentang alamat IP yang digunakan oleh komputer klien.

  > **Warning**
  >
  > Jika Anda menggunakan `0.0.0.0/0` untuk akses SSH, Anda memungkinkan semua alamat IP untuk mengakses instans publik Anda menggunakan SSH. Hal ini dapat diterima untuk waktu yang singkat di lingkungan pengujian, tetapi tidak aman untuk lingkungan produksi. Dalam produksi, Anda hanya dapat memberikan otorisasi pada alamat IP atau rentang alamat tertentu saja untuk mengakses instans-instans Anda menggunakan SSH.

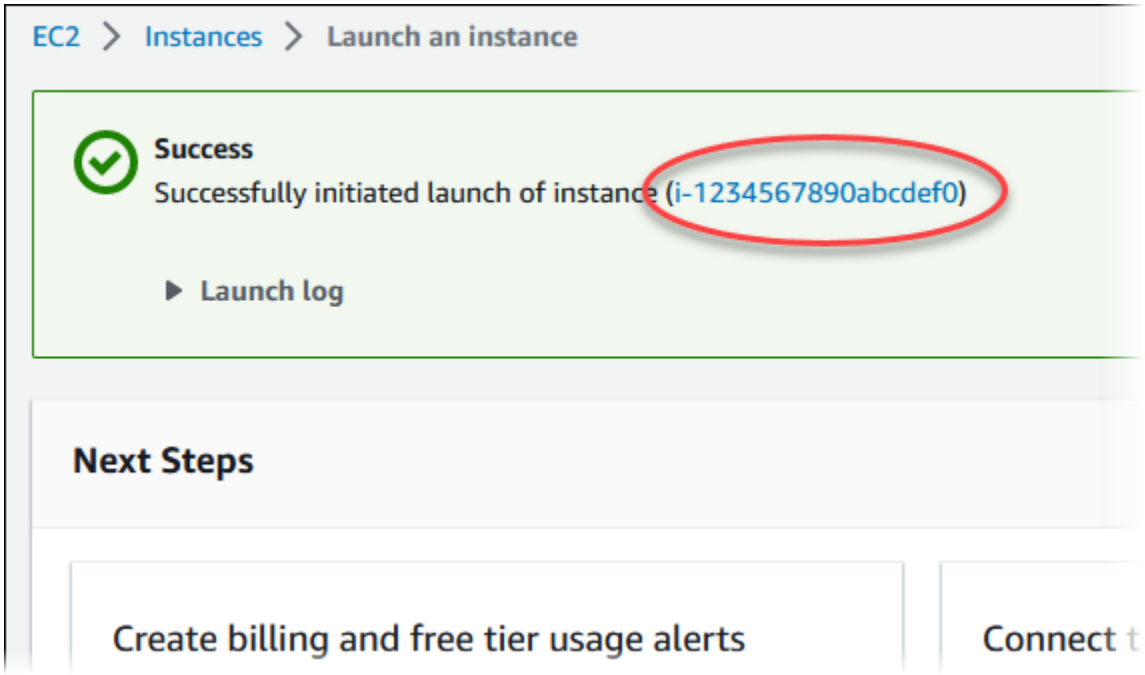
- Aktifkan Izinkan lalu lintas HTTPS dari internet.
- Aktifkan Izinkan lalu lintas HTTP dari internet.



f. Biarkan nilai default untuk bagian yang lainnya.

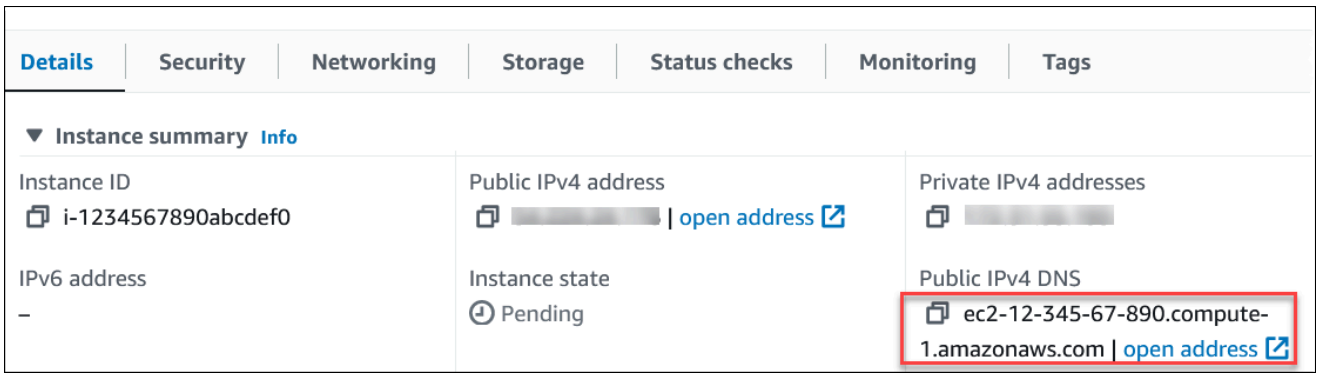
g. Tinjau ringkasan konfigurasi instans di panel Ringkasan, dan ketika Anda siap, pilih Luncurkan instans.

5. Di halaman Status Peluncuran, catat pengidentifikasi untuk instans EC2 baru Anda, misalnya: `i-1234567890abcdef0`.

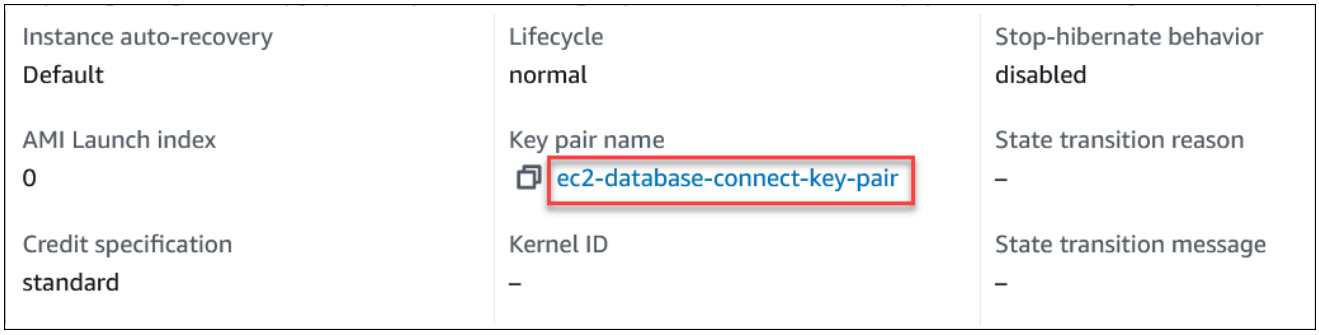


6. Pilih pengidentifikasi instans EC2 untuk membuka daftar instans EC2, lalu pilih instans EC2 Anda.
7. Di tab Detail, catat nilai-nilai berikut, yang akan Anda butuhkan saat menghubungkan menggunakan SSH:

   a. Di Ringkasan instans, catat nilai untuk DNS IPv4 Publik.



   b. Di Detail instans, catat nilai untuk Nama pasangan kunci.



8. Tunggu hingga status instans untuk instans Anda Berjalan sebelum melanjutkan.
9. Selesaikan Membuat instans DB Amazon RDS.

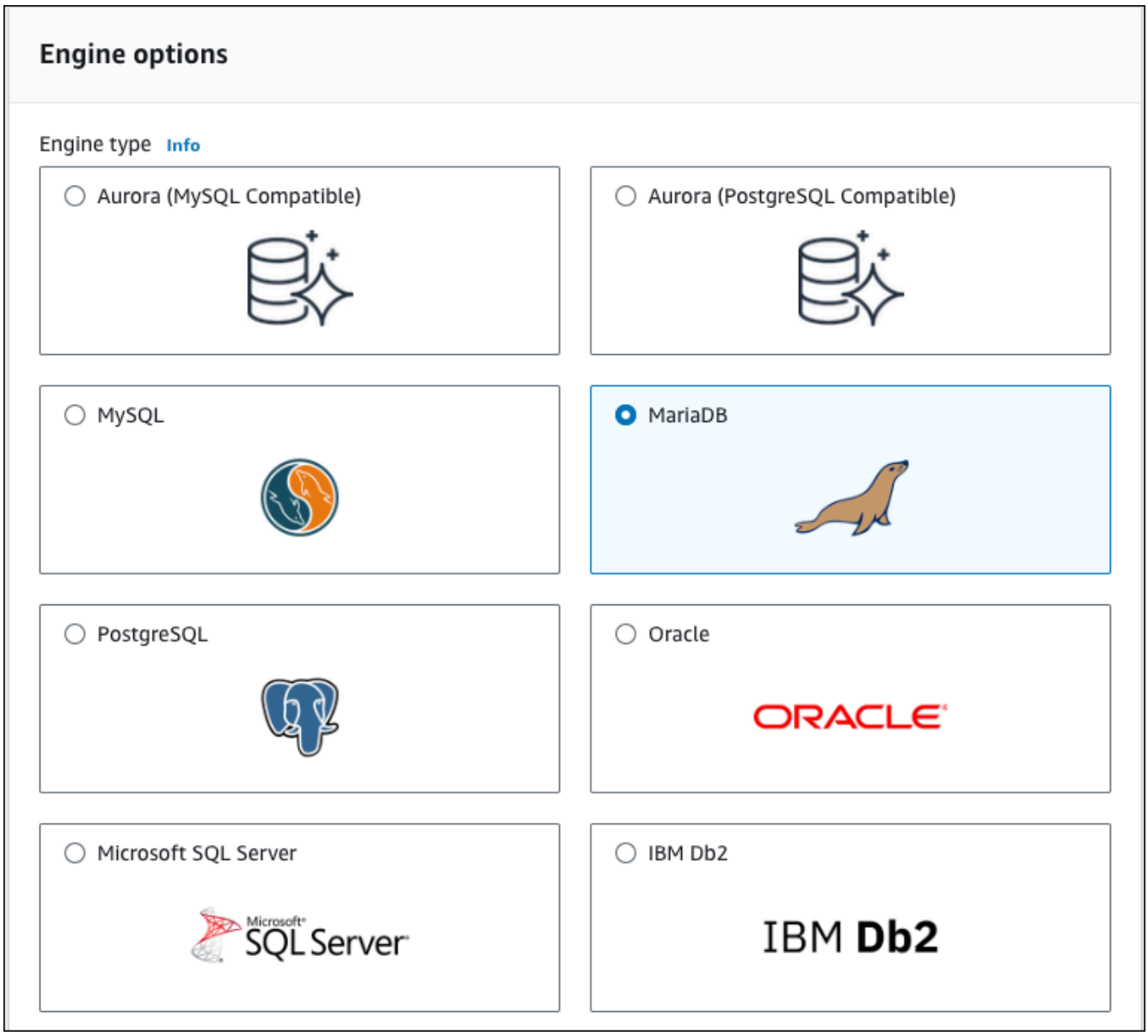
Membuat instans DB Amazon RDS

Buat instans DB RDS untuk MariaDB, RDS for MySQL, atau RDS for PostgreSQL yang mempertahankan data yang digunakan oleh aplikasi web.

RDS for MariaDB

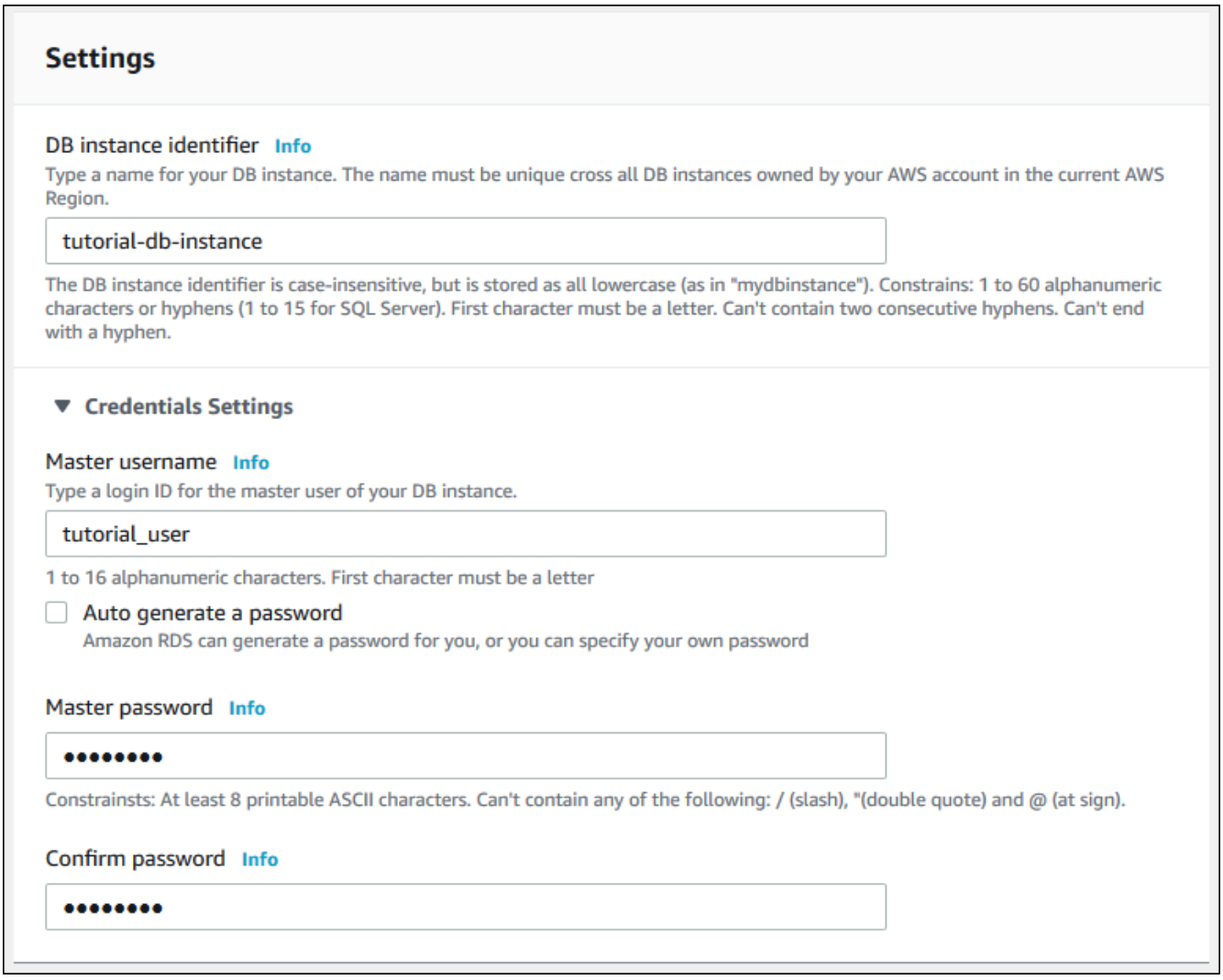
Untuk membuat instans MariaDB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di sudut kanan atas AWS Management Console, periksa Wilayah AWS. Ini sama dengan tempat Anda membuat instans EC2.
3. Di panel navigasi, pilih Basis Data.
4. Pilih Buat basis data.
5. Di halaman Buat basis data, pilih Pembuatan standar.
6. Untuk opsi Mesin, pilih MariaDB.

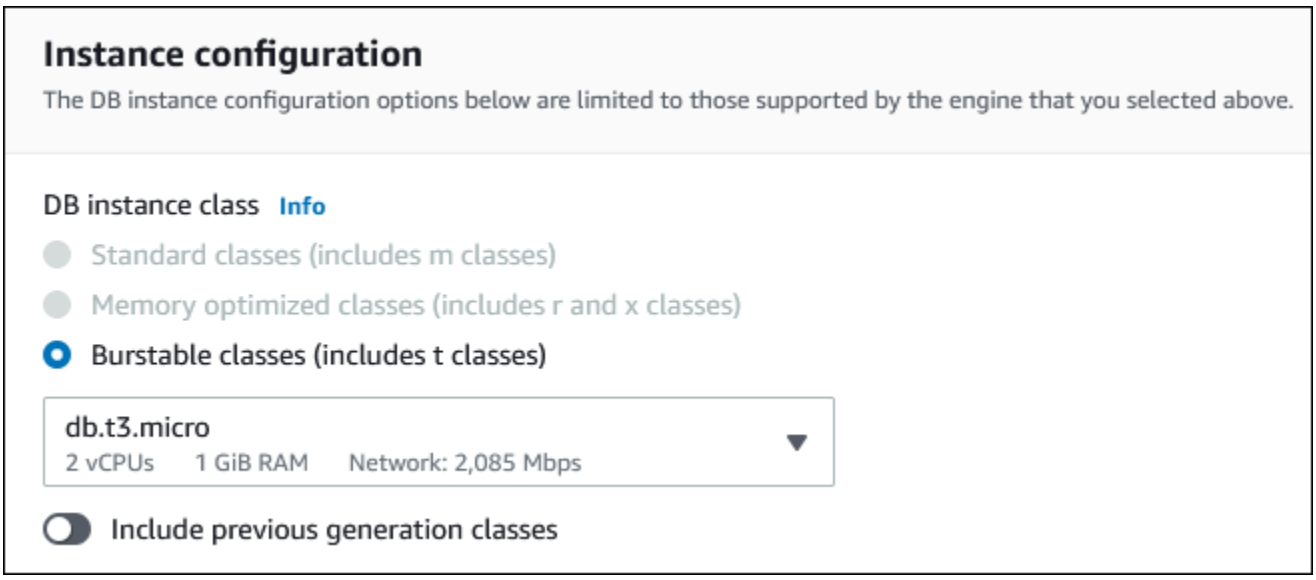


7. Untuk Templat, pilih Tingkat gratis.

8. Di bagian Ketersediaan dan daya tahan, pertahankan default-nya.
9. Di bagian Pengaturan, atur nilai-nilai ini:
   - Pengidentifikasi instans DB – Ketikkan **`tutorial-db-instance`**.
   - Nama pengguna utama – Ketikkan **`tutorial_user`**.
   - Buat kata sandi secara otomatis – Biarkan opsi nonaktif.
   - Kata sandi utama – Ketikkan kata sandi.
   - Konfirmasi kata sandi – Ketik ulang kata sandi.

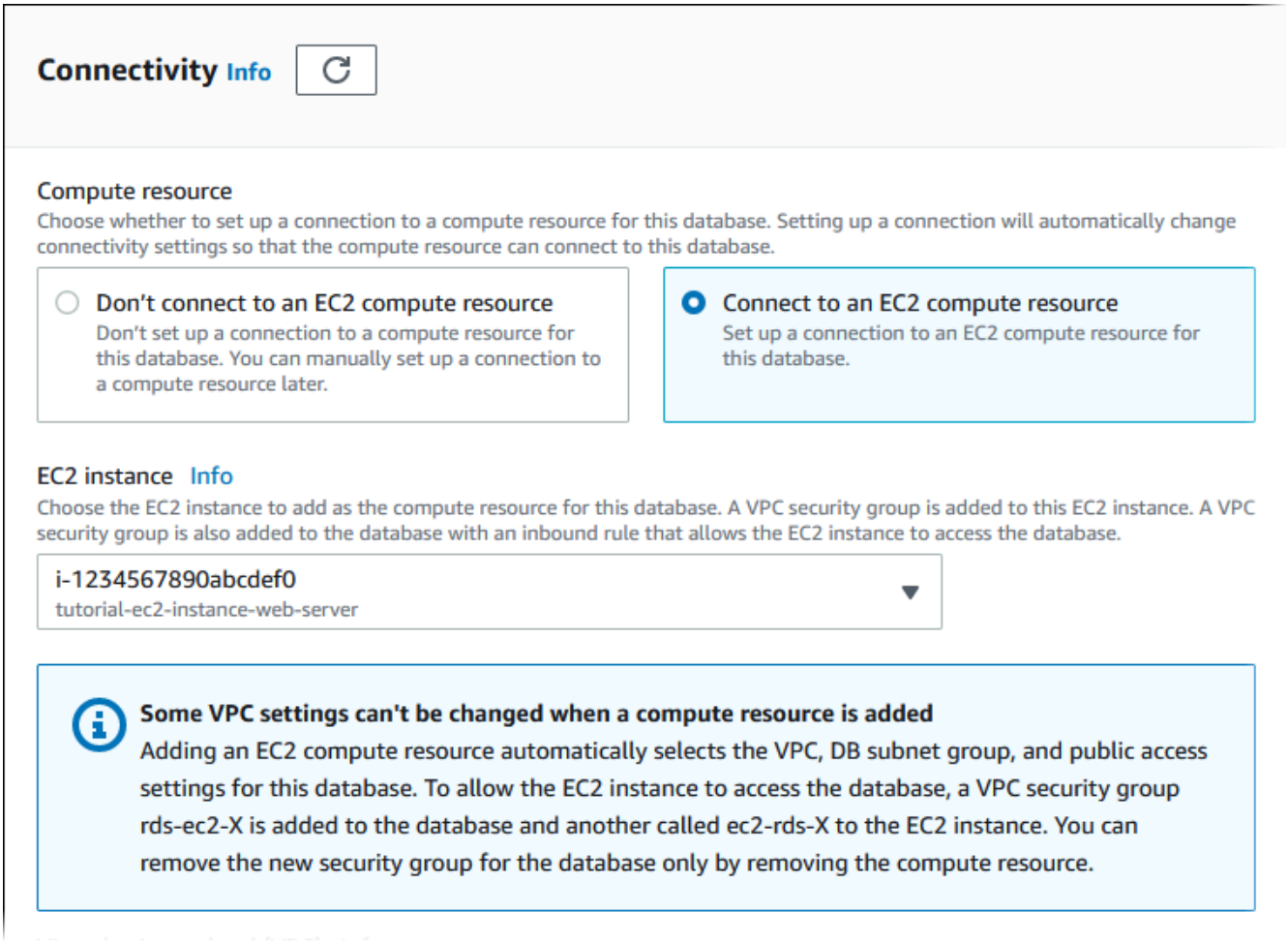


10. Di bagian Konfigurasi instans, atur nilai-nilai ini:
    - Kelas runtutan (termasuk kelas t)
    - db.t3.micro



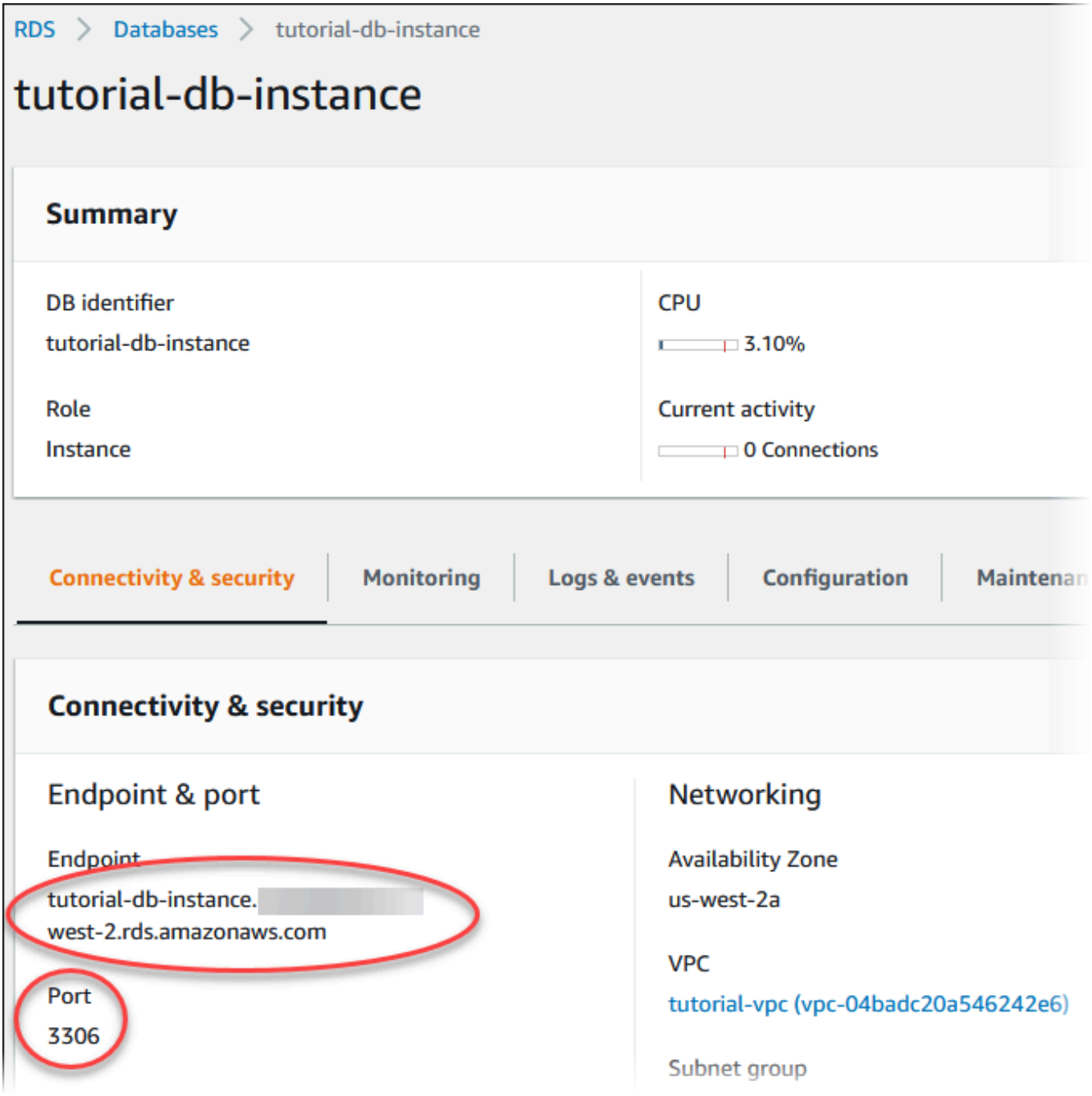
11. Di bagian Penyimpanan, pertahankan default-nya.
12. Di bagian Konektivitas, atur nilai-nilai ini dan biarkan nilai lainnya sebagai default:

    - Untuk sumber daya Komputasi, pilih Hubungkan ke sumber daya komputasi EC2.
    - Untuk contoh EC2, pilih instans EC2 yang Anda buat sebelumnya, seperti tutorial-ec2 -. instance-web-server



13. Di bagian Autentikasi basis data, pastikan Autentikasi kata sandi dipilih.
14. Buka bagian Konfigurasi tambahan, dan masukkan **`sample`** untuk Nama database awal. Biarkan opsi lainnya menggunakan pengaturan default.
15. Untuk membuat instans MariaDB, pilih Buat basis data.

    Instans DB baru Anda muncul di daftar Basis data dengan status Membuat.
16. Tunggu sampai Status instans DB baru Anda menampilkan status Tersedia. Lalu pilih nama instans DB untuk menampilkan detailnya.
17. Di bagian Konektivitas & keamanan, lihat Titik Akhir dan Port instans DB.



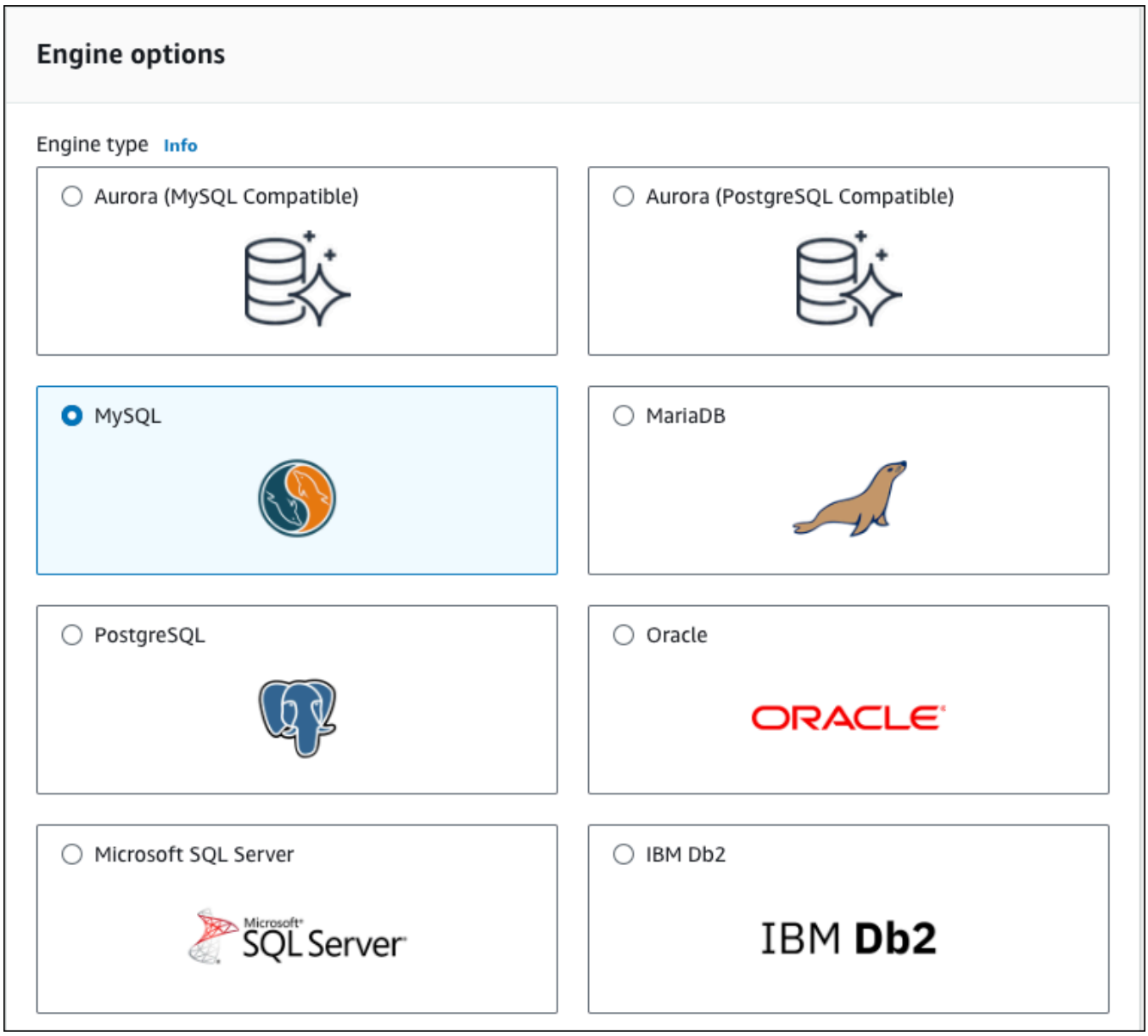
Catat titik akhir dan port untuk instans DB Anda. Gunakan informasi ini untuk menghubungkan server web Anda ke instans DB Anda.

18. Selesaikan Menginstal server web di instans EC2 Anda.

RDS for MySQL

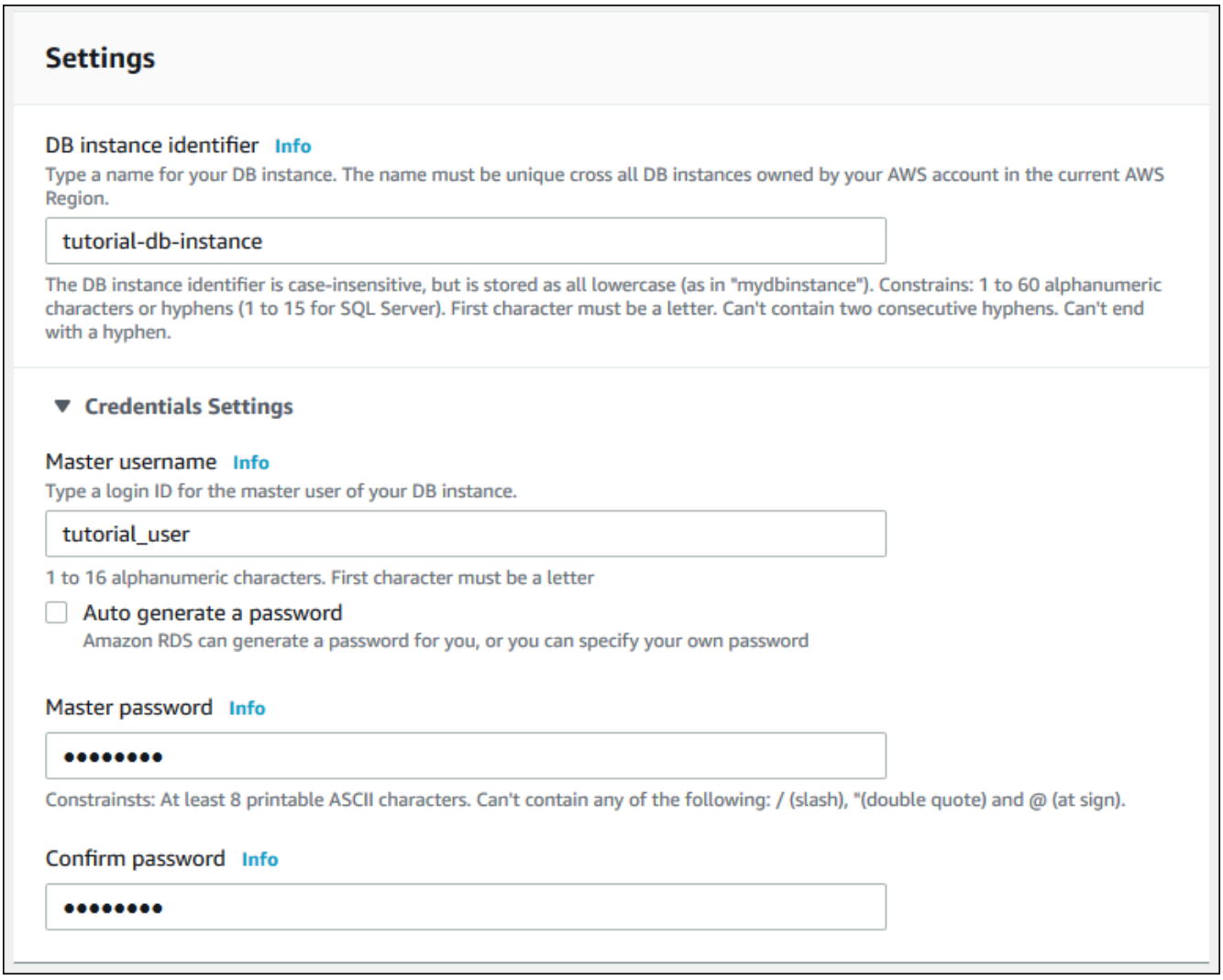
Untuk membuat instans DB MySQL

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di sudut kanan atas AWS Management Console, periksa Wilayah AWS. Ini sama dengan tempat Anda membuat instans EC2.
3. Di panel navigasi, pilih Basis Data.
4. Pilih Buat basis data.
5. Di halaman Buat basis data, pilih Pembuatan standar.
6. Untuk opsi Mesin, pilih MySQL.

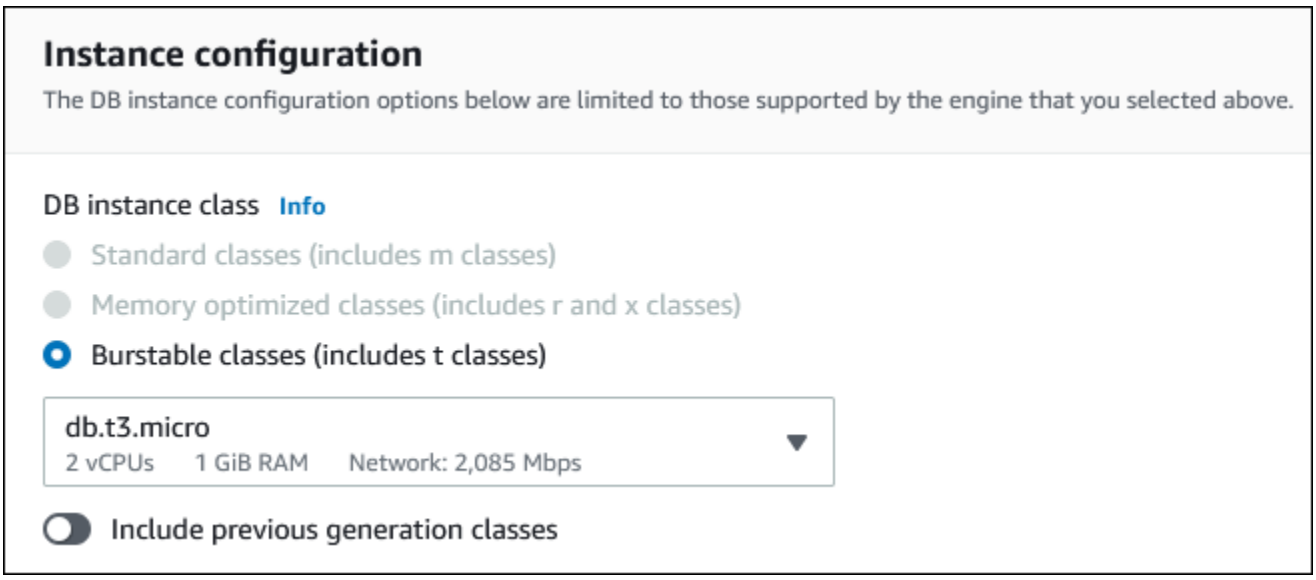


7. Untuk Templat, pilih Tingkat gratis.

8. Di bagian Ketersediaan dan daya tahan, pertahankan default-nya.
9. Di bagian Pengaturan, atur nilai-nilai ini:
   - Pengidentifikasi instans DB – Ketikkan **`tutorial-db-instance`**.
   - Nama pengguna utama – Ketikkan **`tutorial_user`**.
   - Buat kata sandi secara otomatis – Biarkan opsi nonaktif.
   - Kata sandi utama – Ketikkan kata sandi.
   - Konfirmasi kata sandi – Ketik ulang kata sandi.

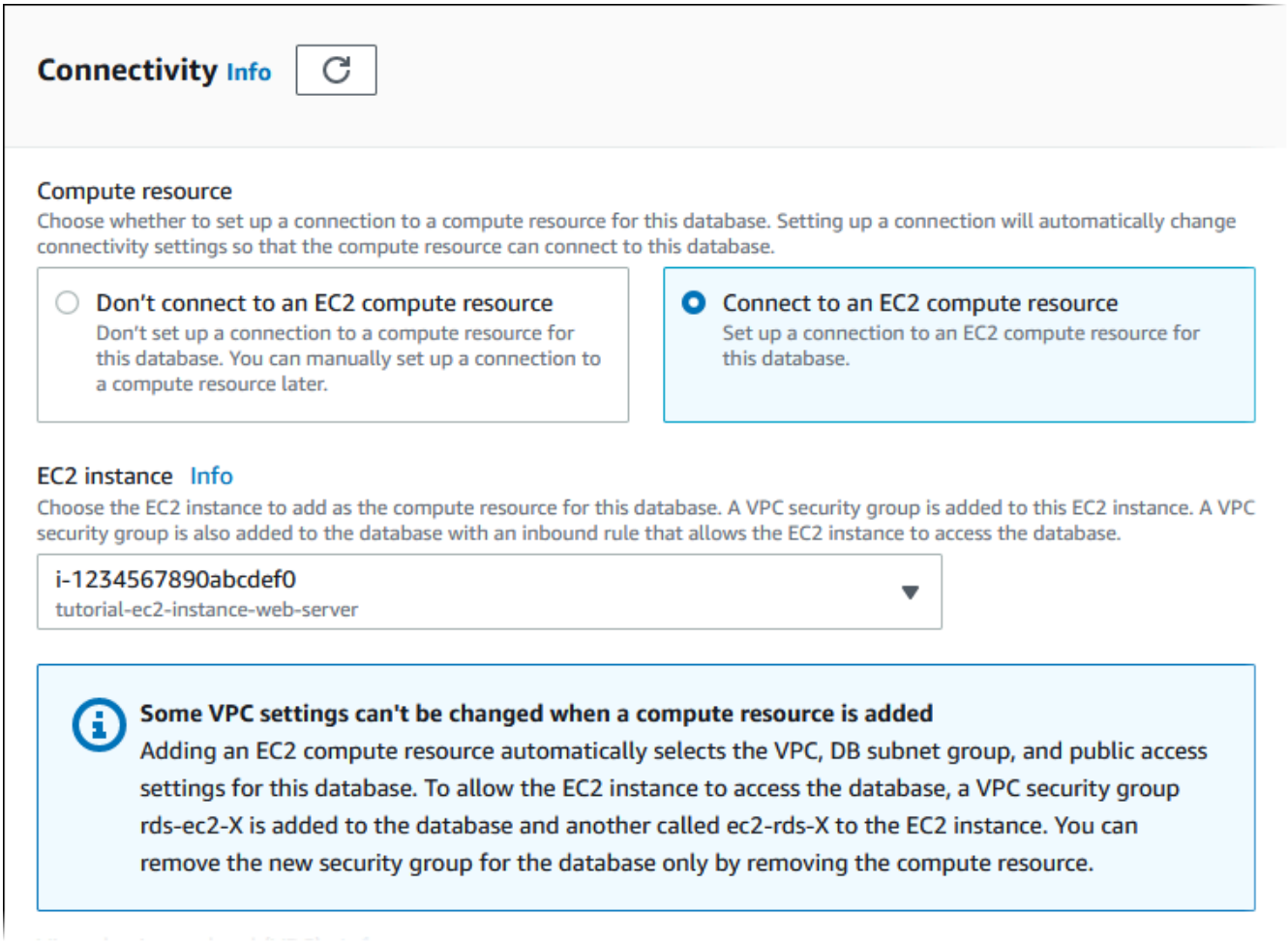


10. Di bagian Konfigurasi instans, atur nilai-nilai ini:
    - Kelas runtutan (termasuk kelas t)
    - db.t3.micro



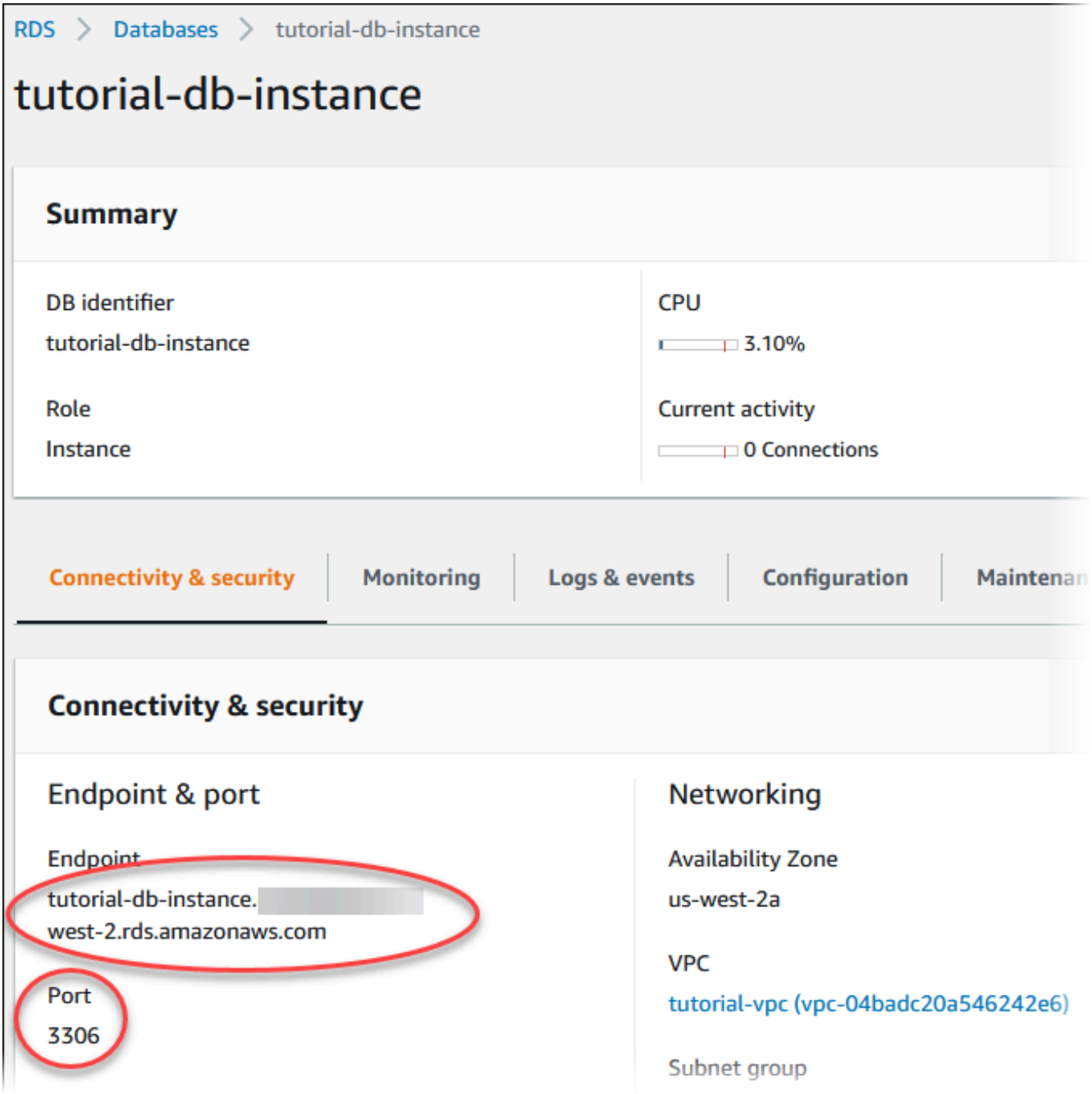
11. Di bagian Penyimpanan, pertahankan default-nya.
12. Di bagian Konektivitas, atur nilai-nilai ini dan biarkan nilai lainnya sebagai default:

    - Untuk sumber daya Komputasi, pilih Hubungkan ke sumber daya komputasi EC2.
    - Untuk contoh EC2, pilih instans EC2 yang Anda buat sebelumnya, seperti tutorial-ec2 -. instance-web-server



13. Di bagian Autentikasi basis data, pastikan Autentikasi kata sandi dipilih.
14. Buka bagian Konfigurasi tambahan, dan masukkan **`sample`** untuk Nama database awal. Biarkan opsi lainnya menggunakan pengaturan default.
15. Untuk membuat instans DB MySQL, pilih Buat basis data.

    Instans DB baru Anda muncul di daftar Basis data dengan status Membuat.
16. Tunggu sampai Status instans DB baru Anda menampilkan status Tersedia. Lalu pilih nama instans DB untuk menampilkan detailnya.
17. Di bagian Konektivitas & keamanan, lihat Titik Akhir dan Port instans DB.



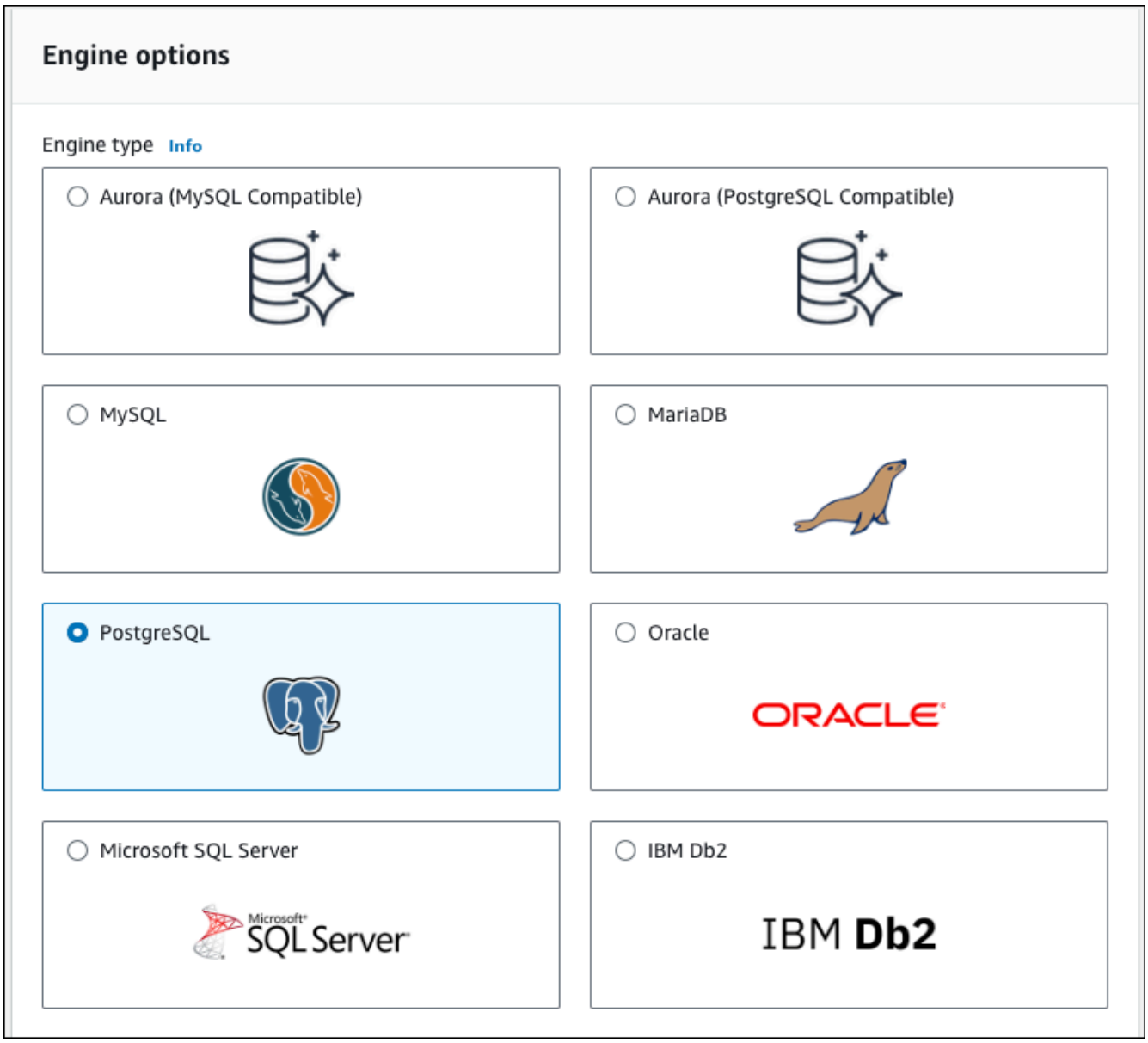
Catat titik akhir dan port untuk instans DB Anda. Gunakan informasi ini untuk menghubungkan server web Anda ke instans DB Anda.

18. Selesaikan Menginstal server web di instans EC2 Anda.

RDS for PostgreSQL

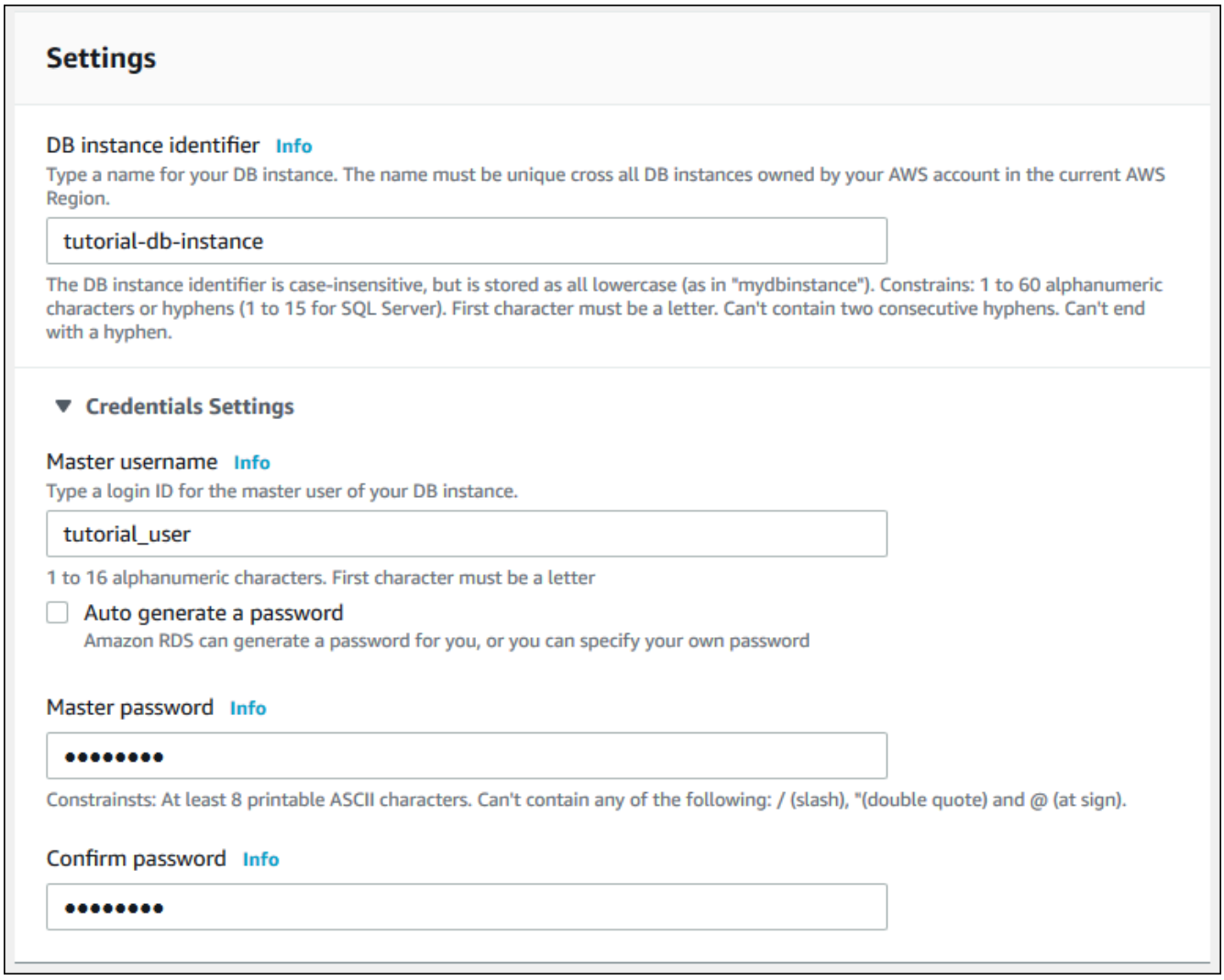
Untuk membuat instans DB PostgreSQL

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di sudut kanan atas AWS Management Console, periksa Wilayah AWS. Ini sama dengan tempat Anda membuat instans EC2.
3. Di panel navigasi, pilih Basis Data.
4. Pilih Buat basis data.
5. Di halaman Buat basis data, pilih Pembuatan standar.
6. Untuk opsi Mesin, pilih PostgreSQL.

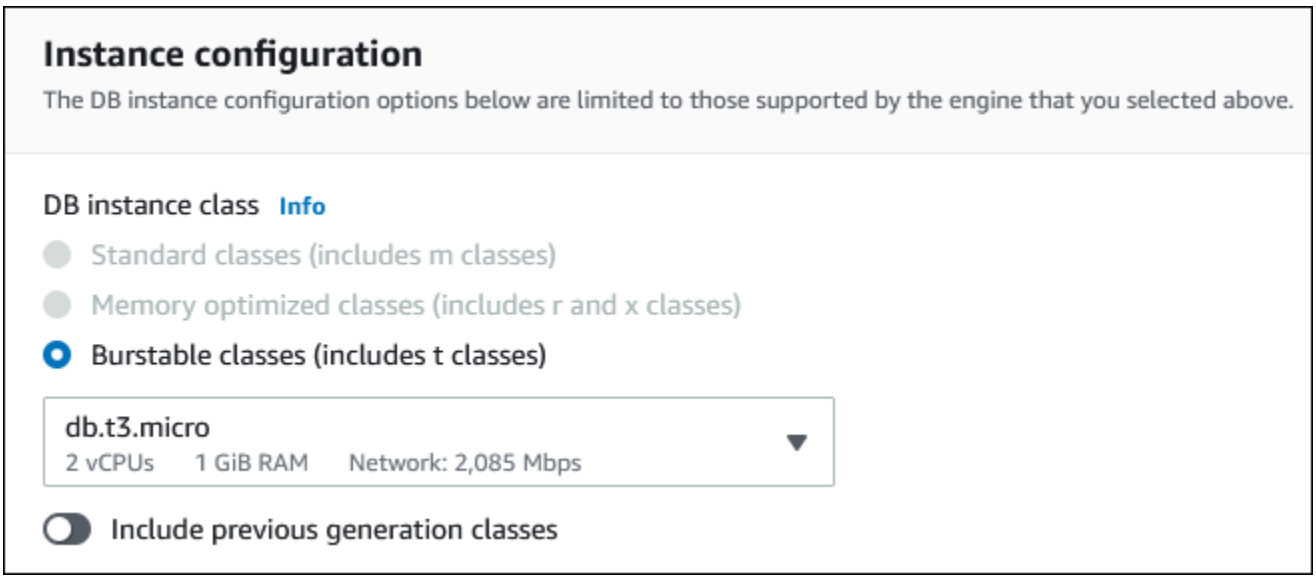


7. Untuk Templat, pilih Tingkat gratis.

8. Di bagian Ketersediaan dan daya tahan, pertahankan default-nya.
9. Di bagian Pengaturan, atur nilai-nilai ini:
   - Pengidentifikasi instans DB – Ketikkan **`tutorial-db-instance`**.
   - Nama pengguna utama – Ketikkan **`tutorial_user`**.
   - Buat kata sandi secara otomatis – Biarkan opsi nonaktif.
   - Kata sandi utama – Ketikkan kata sandi.
   - Konfirmasi kata sandi – Ketik ulang kata sandi.

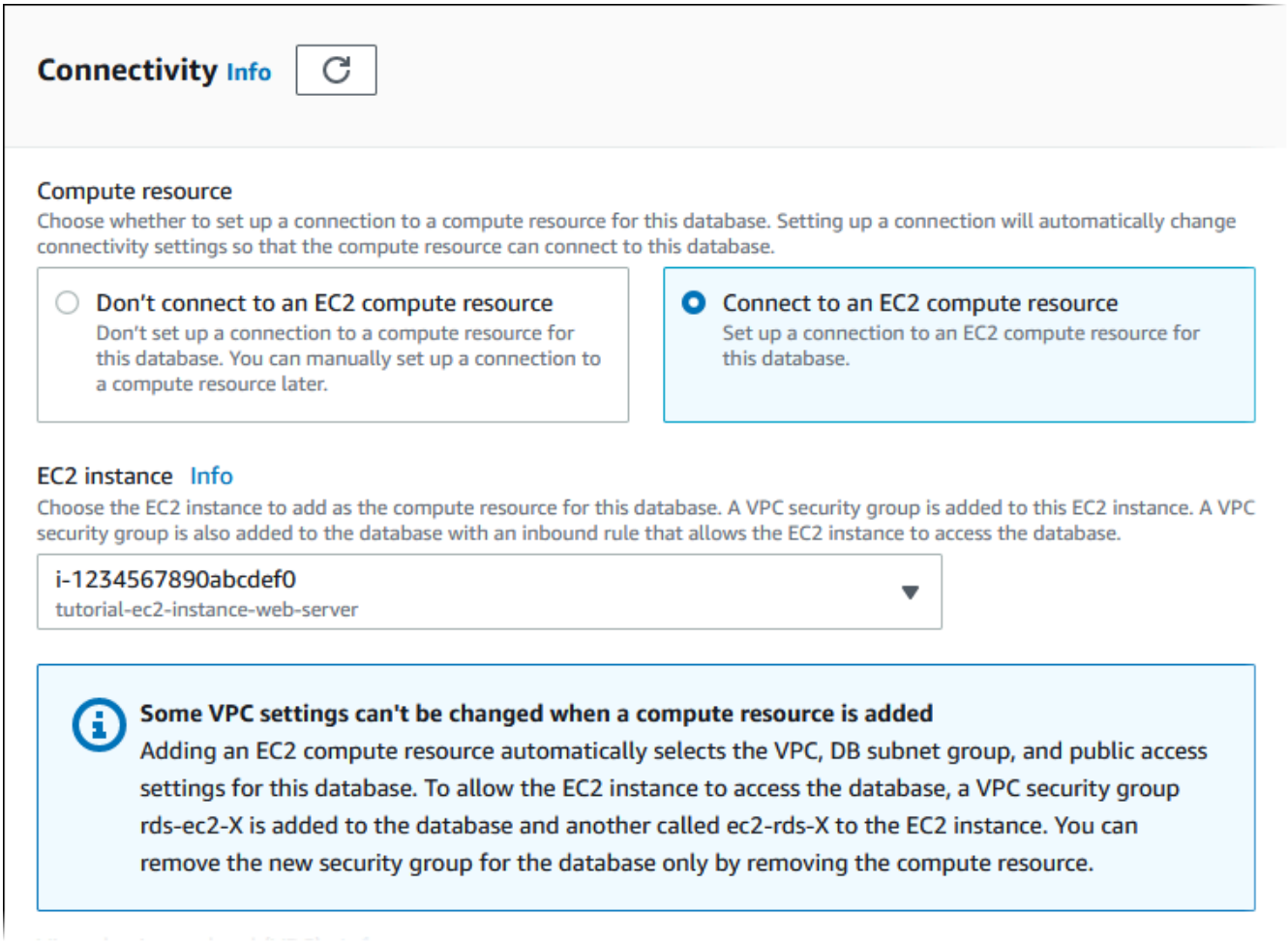


10. Di bagian Konfigurasi instans, atur nilai-nilai ini:
    - Kelas runtutan (termasuk kelas t)
    - db.t3.micro



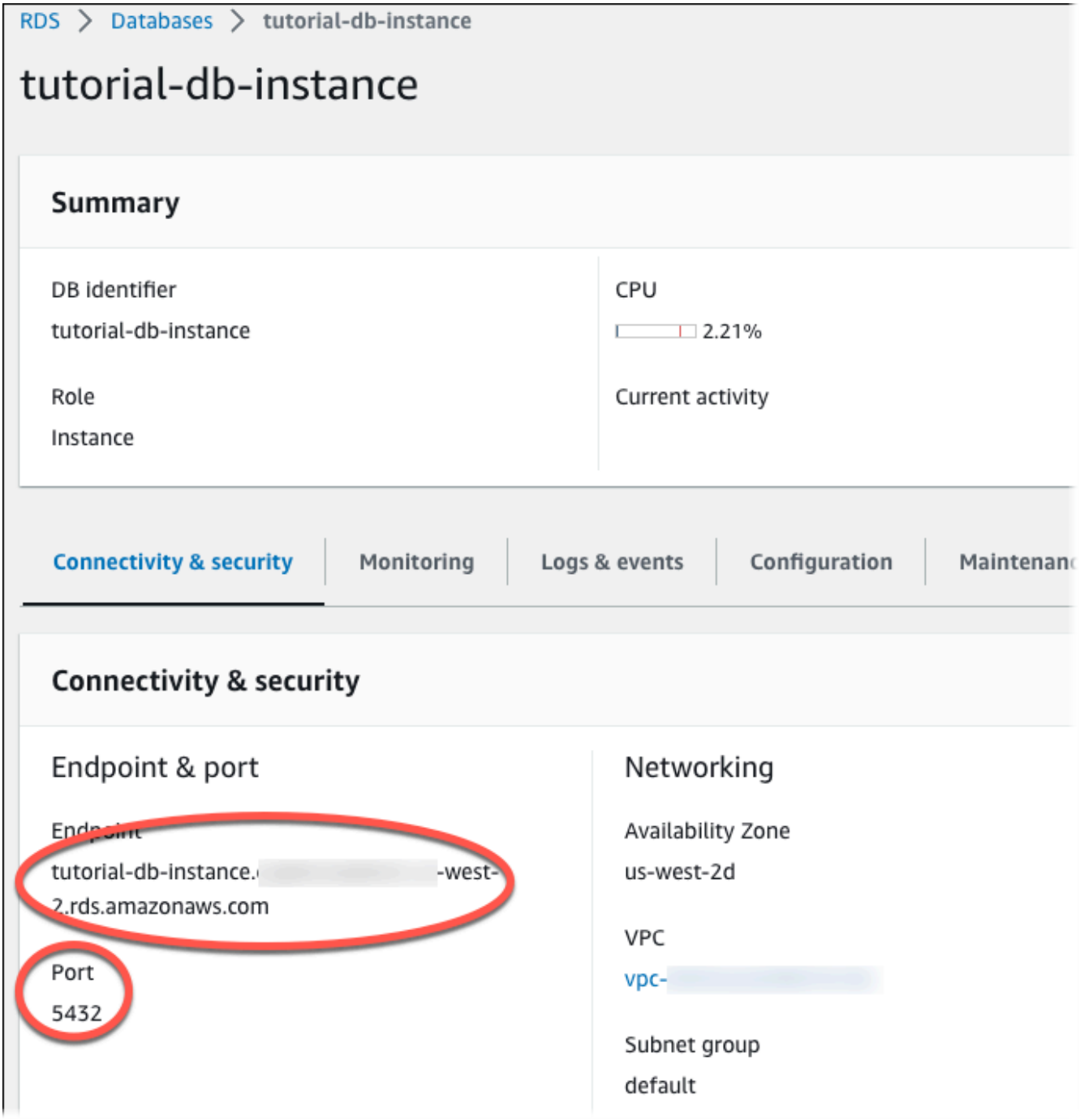
11. Di bagian Penyimpanan, pertahankan default-nya.
12. Di bagian Konektivitas, atur nilai-nilai ini dan biarkan nilai lainnya sebagai default:

    - Untuk sumber daya Komputasi, pilih Hubungkan ke sumber daya komputasi EC2.
    - Untuk contoh EC2, pilih instans EC2 yang Anda buat sebelumnya, seperti tutorial-ec2 -. instance-web-server



13. Di bagian Autentikasi basis data, pastikan Autentikasi kata sandi dipilih.
14. Buka bagian Konfigurasi tambahan, dan masukkan **`sample`** untuk Nama database awal. Biarkan opsi lainnya menggunakan pengaturan default.
15. Untuk membuat instans DB PostgreSQL, pilih Buat basis data.

    Instans DB baru Anda muncul di daftar Basis data dengan status Membuat.
16. Tunggu sampai Status instans DB baru Anda menampilkan status Tersedia. Lalu pilih nama instans DB untuk menampilkan detailnya.
17. Di bagian Konektivitas & keamanan, lihat Titik Akhir dan Port instans DB.



Catat titik akhir dan port untuk instans DB Anda. Gunakan informasi ini untuk menghubungkan server web Anda ke instans DB Anda.

18. Selesaikan Menginstal server web di instans EC2 Anda.

Menginstal server web di instans EC2 Anda

Instal server web pada instans EC2 yang Anda buat di Meluncurkan instans EC2. Server web ini terhubung ke instans DB Amazon RDS yang Anda buat di Membuat instans DB Amazon RDS.

Menginstal server web Apache dengan PHP dan MariaDB

Hubungkan ke instans EC2 Anda dan instal server web.

Menghubungkan ke instans EC2 dan menginstal server web Apache dengan PHP

1. Hubungkan ke instans EC2 yang Anda buat sebelumnya dengan mengikuti langkah-langkah di Menghubungkan ke instans Linux di Panduan Pengguna Amazon EC2 untuk Instans Linux.

   Sebaiknya Anda menghubungkan ke instans EC2 menggunakan SSH. Jika utilitas klien SSH diinstal di Windows, Linux, atau Mac, Anda dapat menghubungkan ke instans menggunakan format perintah berikut:

   ```
   ssh -i location_of_pem_file ec2-user@ec2-instance-public-dns-name
   ```

   Misalnya, asumsikan bahwa `ec2-database-connect-key-pair.pem` disimpan di `/dir1` di Linux, dan DNS IPv4 publik untuk instans EC2 Anda adalah `ec2-12-345-678-90.compute-1.amazonaws.com`. Perintah SSH Anda akan tampak seperti berikut:

   ```
   ssh -i /dir1/ec2-database-connect-key-pair.pem ec2-
   user@ec2-12-345-678-90.compute-1.amazonaws.com
   ```

2. Dapatkan pembaruan keamanan dan perbaikan bug terbaru dengan memperbarui perangkat lunak di instans EC2 Anda. Untuk melakukannya, gunakan perintah berikut.

   > **Note**
   >
   > Opsi `-y` menginstal pembaruan tanpa meminta konfirmasi. Hilangkan opsi ini untuk memeriksa pembaruan sebelum menginstal.

   ```
   sudo dnf update -y
   ```

3. Setelah pembaruan selesai, instal server web Apache, PHP, dan MariaDB atau perangkat lunak PostgreSQL menggunakan perintah berikut. Perintah ini menginstal beberapa paket perangkat lunak dan dependensi terkait bersamaan.

   MariaDB & MySQL

   ```
   sudo dnf install -y httpd php php-mysqli mariadb105
   ```

   PostgreSQL

   ```
   sudo dnf install -y httpd php php-pgsql postgresql15
   ```

   Jika Anda mengalami kesalahan, instans Anda mungkin tidak diluncurkan dengan AMI Amazon Linux 2023. Sebaiknya gunakan AMI Amazon Linux 2 AMI. Anda dapat melihat versi Amazon Linux Anda menggunakan perintah berikut.

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

   Untuk informasi selengkapnya, lihat Memperbarui perangkat lunak instans.

4. Mulai server web dengan perintah yang ditampilkan berikut ini.

   ```
   sudo systemctl start httpd
   ```

   Anda dapat menguji apakah server web Anda terinstal dan berjalan dengan benar. Untuk melakukannya, masukkan nama Sistem Nama Domain (DNS) publik dari instans EC2 Anda di bilah alamat browser web, misalnya: `http://ec2-42-8-168-21.us-west-1.compute.amazonaws.com`. Jika server web Anda berjalan, maka Anda akan melihat halaman uji Apache.

   Jika Anda tidak melihat halaman uji Apache, periksa aturan masuk Anda untuk grup keamanan VPC yang Anda buat di Tutorial: Membuat VPC untuk digunakan dengan instans DB (khusus IPv4). Pastikan aturan masuk Anda menyertakan aturan yang mengizinkan akses HTTP (port 80) untuk alamat IP agar terhubung ke server web.

> **Note**
>
> Halaman uji Apache hanya muncul jika tidak ada konten di direktori root dokumen, `/var/www/html`. Setelah konten ditambahkan ke direktori root dokumen, konten tersebut akan muncul di alamat DNS publik dari instans EC2 Anda. Sebelumnya, konten tersebut muncul di halaman uji Apache.

5. Konfigurasikan server web untuk memulai setiap boot sistem menggunakan perintah `systemctl`.

```
sudo systemctl enable httpd
```

Untuk mengizinkan `ec2-user` mengelola file di direktori root default untuk server web Apache Anda, ubah kepemilikan dan izin direktori `/var/www`. Ada banyak cara untuk menyelesaikan tugas ini. Dalam tutorial ini, Anda menambahkan `ec2-user` ke grup `apache`, untuk memberikan kepemilikan grup `apache` atas direktori `/var/www` dan menetapkan izin tulis ke grup.

Mengatur izin file untuk server web Apache

1. Tambahkan pengguna `ec2-user` ke grup `apache`.

```
sudo usermod -a -G apache ec2-user
```

2. Keluar untuk menyegarkan izin Anda dan masukkan grup `apache` baru.

```
exit
```

3. Masuk kembali dan verifikasi apakah grup `apache` ada dengan perintah `groups`.

```
groups
```

   Output Anda akan terlihat seperti berikut ini:

```
ec2-user adm wheel apache systemd-journal
```

4. Ubah kepemilikan grup atas direktori `/var/www` dan kontennya ke grup `apache`.

```
sudo chown -R ec2-user:apache /var/www
```

5. Ubah izin direktori atas `/var/www` dan subdirektorinya untuk menambahkan izin tulis grup dan atur ID grup pada subdirektori yang dibuat di masa mendatang.

```
sudo chmod 2775 /var/www
find /var/www -type d -exec sudo chmod 2775 {} \;
```

6. Ubah izin file secara berulang di direktori `/var/www` dan subdirektorinya untuk menambahkan izin tulis grup.

```
find /var/www -type f -exec sudo chmod 0664 {} \;
```

Sekarang, `ec2-user` (dan setiap anggota grup `apache` mendatang) dapat menambahkan, menghapus, dan mengedit file pada root dokumen Apache. Ini memungkinkan Anda untuk menambahkan konten, seperti situs web statis atau aplikasi PHP.

> **Note**
>
> Server web yang menjalankan protokol HTTP tidak memberikan keamanan transportasi untuk data yang dikirim atau diterimanya. Saat Anda menghubungkan ke server HTTP menggunakan browser web, banyak informasi yang terlihat oleh penyadap di mana saja di sepanjang jalur jaringan. Informasi ini mencakup URL yang Anda kunjungi, konten halaman web yang Anda terima, dan konten (termasuk kata sandi) dari setiap formulir HTML. Praktik terbaik untuk mengamankan server web Anda adalah dengan menginstal dukungan untuk HTTPS (HTTP Secure). Protokol ini melindungi data Anda dengan enkripsi SSL/TLS. Untuk informasi selengkapnya, lihat Tutorial: Mengonfigurasi SSL/TLS dengan Amazon Linux AMI di Panduan Pengguna Amazon EC2.

Menghubungkan server web Apache ke instans DB

Selanjutnya, Anda menambahkan konten ke server web Apache yang terhubung ke instans DB Amazon RDS.

Menambahkan konten ke server web Apache yang terhubung ke instans DB Anda

1. Saat masih terhubung ke instans EC2, ubah direktori ke `/var/www` dan buat subdirektori baru yang diberi nama `inc`.

```
cd /var/www
mkdir inc
cd inc
```

2. Buat file baru dalam direktori `inc` yang diberi nama `dbinfo.inc`, lalu edit file tersebut dengan menggunakan nano (atau editor pilihan Anda).

```
>dbinfo.inc
nano dbinfo.inc
```

3. Tambahkan konten berikut ini ke file `dbinfo.inc`. Di sini, *db_instance_endpoint* adalah titik akhir instans DB Anda, tanpa port, untuk instans DB Anda.

> **Note**
>
> Sebaiknya tempatkan nama pengguna dan informasi kata sandi dalam folder yang bukan bagian dari root dokumen untuk server web Anda. Hal ini mengurangi kemungkinan informasi keamanan Anda terungkap.
> Pastikan untuk mengubah `master password` ke kata sandi yang sesuai di aplikasi Anda.

```
<?php

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

4. Simpan dan tutup file `dbinfo.inc`. Jika Anda menggunakan nano, simpan dan tutup file dengan menggunakan Ctrl+S dan Ctrl+X.
5. Ubah direktori ke `/var/www/html`.

```
cd /var/www/html
```

6. Buat file baru dalam direktori `html` yang diberi nama `SamplePage.php`, lalu edit file tersebut dengan menggunakan nano (atau editor pilihan Anda).

```
>SamplePage.php
nano SamplePage.php
```

7. Tambahkan konten berikut ini ke file `SamplePage.php`:

   MariaDB & MySQL

```
<?php include "../inc/dbinfo.inc"; ?>
<html>
<body>
<h1>Sample page</h1>
<?php

  /* Connect to MySQL and select the database. */
  $connection = mysqli_connect(DB_SERVER, DB_USERNAME, DB_PASSWORD);

  if (mysqli_connect_errno()) echo "Failed to connect to MySQL: " .
 mysqli_connect_error();

  $database = mysqli_select_db($connection, DB_DATABASE);

  /* Ensure that the EMPLOYEES table exists. */
  VerifyEmployeesTable($connection, DB_DATABASE);

  /* If input fields are populated, add a row to the EMPLOYEES table. */
  $employee_name = htmlentities($_POST['NAME']);
  $employee_address = htmlentities($_POST['ADDRESS']);

  if (strlen($employee_name) || strlen($employee_address)) {
    AddEmployee($connection, $employee_name, $employee_address);
  }
?>

<!-- Input form -->
<form action="<?PHP echo $_SERVER['SCRIPT_NAME'] ?>" method="POST">
  <table border="0">
    <tr>
```

```
      <td>NAME</td>
      <td>ADDRESS</td>
    </tr>
    <tr>
      <td>
        <input type="text" name="NAME" maxlength="45" size="30" />
      </td>
      <td>
        <input type="text" name="ADDRESS" maxlength="90" size="60" />
      </td>
      <td>
        <input type="submit" value="Add Data" />
      </td>
    </tr>
  </table>
</form>

<!-- Display table data. -->
<table border="1" cellpadding="2" cellspacing="2">
  <tr>
    <td>ID</td>
    <td>NAME</td>
    <td>ADDRESS</td>
  </tr>

<?php

$result = mysqli_query($connection, "SELECT * FROM EMPLOYEES");

while($query_data = mysqli_fetch_row($result)) {
  echo "<tr>";
  echo "<td>",$query_data[0], "</td>",
       "<td>",$query_data[1], "</td>",
       "<td>",$query_data[2], "</td>";
  echo "</tr>";
}
?>

</table>

<!-- Clean up. -->
<?php

  mysqli_free_result($result);
```

```
  mysqli_close($connection);

?>

</body>
</html>


<?php

/* Add an employee to the table. */
function AddEmployee($connection, $name, $address) {
   $n = mysqli_real_escape_string($connection, $name);
   $a = mysqli_real_escape_string($connection, $address);

   $query = "INSERT INTO EMPLOYEES (NAME, ADDRESS) VALUES ('$n', '$a');";

   if(!mysqli_query($connection, $query)) echo("<p>Error adding employee data.</
p>");
}

/* Check whether the table exists and, if not, create it. */
function VerifyEmployeesTable($connection, $dbName) {
  if(!TableExists("EMPLOYEES", $connection, $dbName))
  {
     $query = "CREATE TABLE EMPLOYEES (
         ID int(11) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
         NAME VARCHAR(45),
         ADDRESS VARCHAR(90)
       )";

     if(!mysqli_query($connection, $query)) echo("<p>Error creating table.</
p>");
  }
}

/* Check for the existence of a table. */
function TableExists($tableName, $connection, $dbName) {
  $t = mysqli_real_escape_string($connection, $tableName);
  $d = mysqli_real_escape_string($connection, $dbName);

  $checktable = mysqli_query($connection,
      "SELECT TABLE_NAME FROM information_schema.TABLES WHERE TABLE_NAME = '$t'
 AND TABLE_SCHEMA = '$d'");
```

```
  if(mysqli_num_rows($checktable) > 0) return true;

  return false;
}
?>
```

PostgreSQL

```
<?php include "../inc/dbinfo.inc"; ?>

<html>
<body>
<h1>Sample page</h1>
<?php

/* Connect to PostgreSQL and select the database. */
$constring = "host=" . DB_SERVER . " dbname=" . DB_DATABASE . " user=" .
 DB_USERNAME . " password=" . DB_PASSWORD ;
$connection = pg_connect($constring);

if (!$connection){
 echo "Failed to connect to PostgreSQL";
 exit;
}

/* Ensure that the EMPLOYEES table exists. */
VerifyEmployeesTable($connection, DB_DATABASE);

/* If input fields are populated, add a row to the EMPLOYEES table. */
$employee_name = htmlentities($_POST['NAME']);
$employee_address = htmlentities($_POST['ADDRESS']);

if (strlen($employee_name) || strlen($employee_address)) {
  AddEmployee($connection, $employee_name, $employee_address);
}

?>

<!-- Input form -->
<form action="<?PHP echo $_SERVER['SCRIPT_NAME'] ?>" method="POST">
  <table border="0">
```

```
    <tr>
      <td>NAME</td>
      <td>ADDRESS</td>
    </tr>
    <tr>
      <td>
    <input type="text" name="NAME" maxlength="45" size="30" />
      </td>
      <td>
    <input type="text" name="ADDRESS" maxlength="90" size="60" />
      </td>
      <td>
    <input type="submit" value="Add Data" />
      </td>
    </tr>
  </table>
</form>
<!-- Display table data. -->
<table border="1" cellpadding="2" cellspacing="2">
  <tr>
    <td>ID</td>
    <td>NAME</td>
    <td>ADDRESS</td>
  </tr>

<?php

$result = pg_query($connection, "SELECT * FROM EMPLOYEES");

while($query_data = pg_fetch_row($result)) {
  echo "<tr>";
  echo "<td>",$query_data[0], "</td>",
       "<td>",$query_data[1], "</td>",
       "<td>",$query_data[2], "</td>";
  echo "</tr>";
}
?>
</table>

<!-- Clean up. -->
<?php

  pg_free_result($result);
  pg_close($connection);
```

```
?>
</body>
</html>


<?php

/* Add an employee to the table. */
function AddEmployee($connection, $name, $address) {
   $n = pg_escape_string($name);
   $a = pg_escape_string($address);
   echo "Forming Query";
   $query = "INSERT INTO EMPLOYEES (NAME, ADDRESS) VALUES ('$n', '$a');";

   if(!pg_query($connection, $query)) echo("<p>Error adding employee data.</
p>");
}

/* Check whether the table exists and, if not, create it. */
function VerifyEmployeesTable($connection, $dbName) {
  if(!TableExists("EMPLOYEES", $connection, $dbName))
  {
     $query = "CREATE TABLE EMPLOYEES (
         ID serial PRIMARY KEY,
         NAME VARCHAR(45),
         ADDRESS VARCHAR(90)
       )";

     if(!pg_query($connection, $query)) echo("<p>Error creating table.</p>");
  }
}
/* Check for the existence of a table. */
function TableExists($tableName, $connection, $dbName) {
  $t = strtolower(pg_escape_string($tableName)); //table name is case sensitive
  $d = pg_escape_string($dbName); //schema is 'public' instead of 'sample' db
 name so not using that

  $query = "SELECT TABLE_NAME FROM information_schema.TABLES WHERE TABLE_NAME =
 '$t';";
  $checktable = pg_query($connection, $query);

  if (pg_num_rows($checktable) >0) return true;
  return false;
```

```
}
?>
```

8. Simpan dan tutup file `SamplePage.php`.
9. Verifikasi bahwa server web Anda berhasil terhubung ke instans DB Anda dengan membuka browser web dan menelusuri ke `http://`*`EC2 instance endpoint`*`/SamplePage.php`, misalnya: `http://ec2-12-345-67-890.us-west-2.compute.amazonaws.com/SamplePage.php`.

Anda dapat menggunakan `SamplePage.php` untuk menambahkan data ke instans DB Anda. Data yang Anda tambahkan kemudian ditampilkan di halaman. Untuk memverifikasi apakah data dimasukkan ke dalam tabel, instal klien MySQL pada instans Amazon EC2. Kemudian, hubungkan ke instans DB dan kueri tabelnya.

Untuk informasi tentang menginstal klien MySQL dan menghubungkan ke instans DB, lihat Menghubungkan ke instans DB yang menjalankan mesin basis data MySQL.

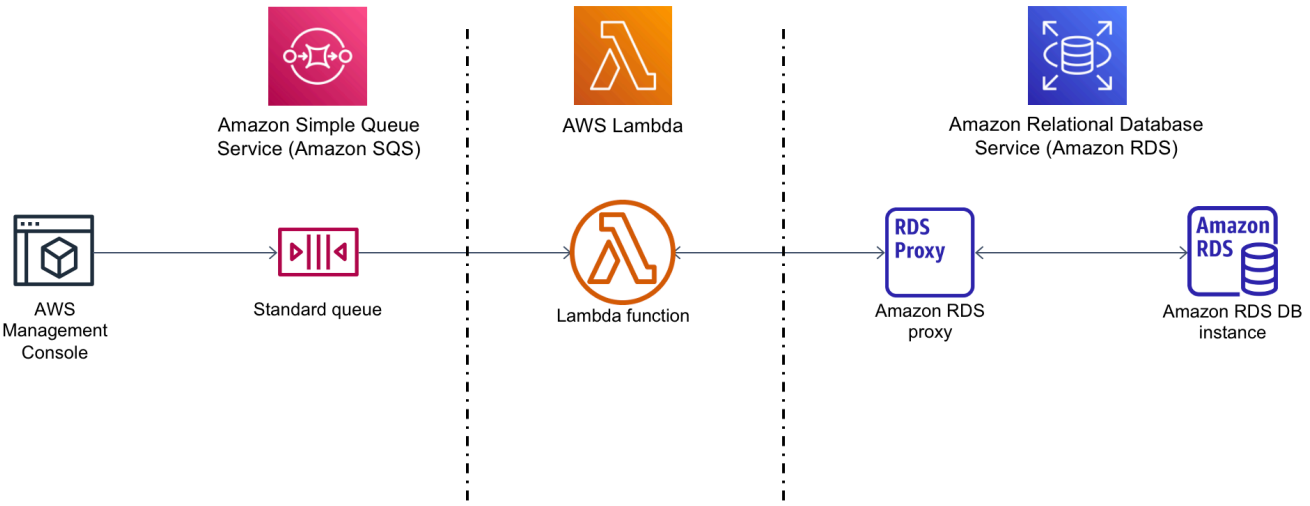
Untuk memastikan instans DB Anda seaman mungkin, verifikasi bahwa sumber di luar VPC tidak dapat menghubungkan ke instans DB Anda.

Setelah selesai menguji server web dan database, Anda harus menghapus instans DB dan instans Amazon EC2 Anda.

- Untuk menghapus instans DB, ikuti petunjuk di Menghapus instans DB. Anda tidak perlu membuat snapshot terakhir.
- Untuk mengakhiri instans Amazon EC2, ikuti instruksi di Mengakhiri instans Anda dalam Panduan Pengguna Amazon EC2.

Tutorial: Menggunakan fungsi Lambda untuk mengakses basis data Amazon RDS

Dalam tutorial ini, Anda menggunakan fungsi Lambda untuk menulis data ke basis data Amazon Relational Database Service (Amazon RDS) melalui Proksi RDS. Fungsi Lambda Anda membaca catatan dari antrean Amazon Simple Queue Service (Amazon SQS) dan menulis item baru ke tabel di basis data Anda setiap kali ada pesan yang ditambahkan. Dalam contoh ini, Anda menggunakan AWS Management Console untuk menambahkan pesan secara manual ke antrean. Diagram berikut menunjukkan AWS sumber daya yang Anda gunakan untuk menyelesaikan tutorial.



Dengan Amazon RDS, Anda dapat menjalankan basis data relasional terkelola di cloud menggunakan produk basis data umum seperti Microsoft SQL Server, MariaDB, MySQL, Oracle Database, dan PostgreSQL. Dengan mengakses basis data menggunakan Lambda, Anda dapat membaca dan menulis data sebagai respons terhadap peristiwa, seperti pelanggan baru yang mendaftar ke situs web Anda. Fungsi, instans basis data, dan proksi secara otomatis diskalakan untuk memenuhi periode saat permintaan sedang tinggi.

Untuk menyelesaikan tutorial ini, lakukan tugas berikut:

1. Luncurkan RDS untuk instance database MySQL dan proxy di VPC default Anda Akun AWS.
2. Buat dan uji fungsi Lambda yang membuat tabel baru di basis data Anda dan menulis data ke dalamnya.

3. Buat antrean Amazon SQS dan konfigurasikan untuk menginvokasi fungsi Lambda Anda setiap kali ada pesan baru yang ditambahkan.
4. Uji penyiapan lengkap dengan menambahkan pesan ke antrian Anda menggunakan AWS Management Console dan memantau hasilnya menggunakan CloudWatch Log.

Dengan menyelesaikan langkah-langkah ini, Anda belajar:

- Cara menggunakan Amazon RDS untuk membuat instans basis data dan proksi, serta menghubungkan fungsi Lambda ke proksi.
- Cara menggunakan Lambda untuk melakukan operasi pembuatan dan pembacaan pada basis data Amazon RDS.
- Cara menggunakan Amazon SQS untuk menginvokasi fungsi Lambda.

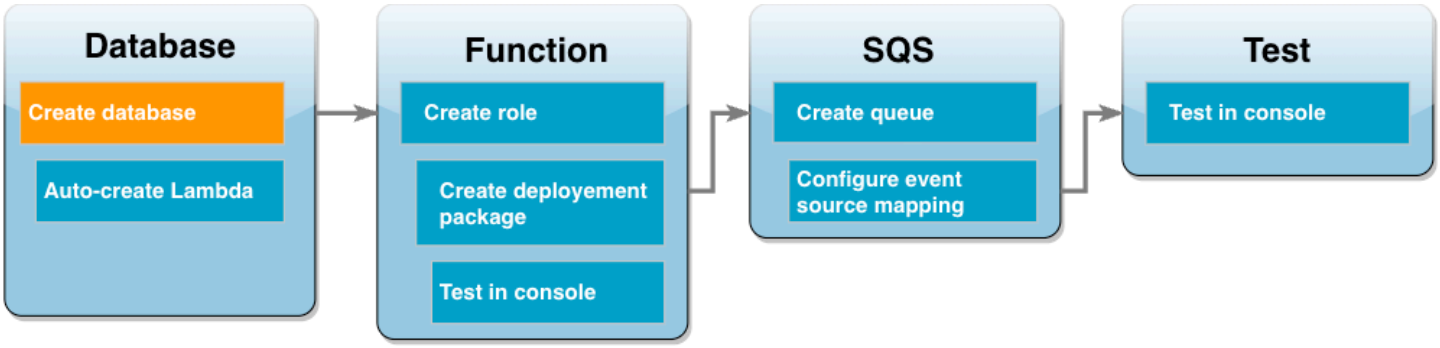
Anda dapat menyelesaikan tutorial ini menggunakan AWS Management Console atau AWS Command Line Interface (AWS CLI).

Prasyarat

Sebelum memulai, selesaikan langkah-langkah di bagian berikut:

- Mendaftar Akun AWS
- Membuat pengguna administratif

Buat instans DB Amazon RDS



Instans DB Amazon RDS adalah lingkungan basis data terisolasi yang berjalan di AWS Cloud. Instans dapat berisi satu atau beberapa basis data yang dibuat pengguna. Kecuali Anda menentukan sebaliknya, Amazon RDS membuat instance database baru di VPC default yang disertakan dalam

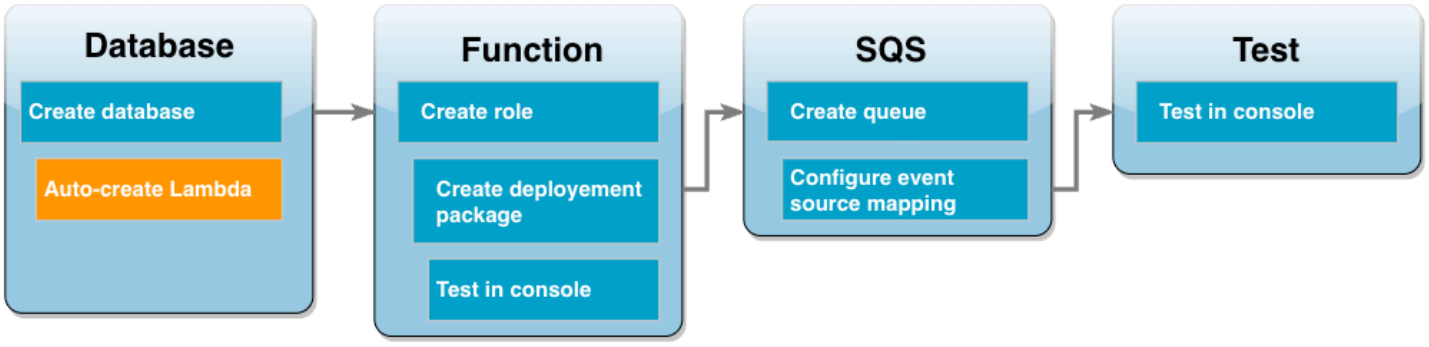
file Anda. Akun AWS Untuk informasi selengkapnya tentang Amazon VPC, lihat Panduan Pengguna Amazon Virtual Private Cloud.

Dalam tutorial ini, Anda membuat instance baru di VPC default Anda Akun AWS dan membuat database bernama `ExampleDB` dalam contoh itu. Anda dapat membuat instans dan database DB Anda menggunakan salah satu AWS Management Console atau AWS CLI.

Untuk membuat instans basis data

1. Buka konsol Amazon RDS dan pilih Buat basis data.
2. Pilih Pembuatan standar, lalu di Opsi mesin, pilih MySQL.
3. Di Templat, pilih Tingkat gratis.
4. Di Pengaturan, untuk Pengidentifikasi instans DB, masukkan **`MySQLForLambda`**.
5. Tetapkan nama pengguna dan kata sandi Anda dengan melakukan hal berikut:
   a. Di Pengaturan kredensial, atur Nama pengguna utama sebagai `admin`.
   b. Untuk Kata sandi utama, masukkan dan konfirmasikan kata sandi untuk mengakses basis data Anda.
6. Tentukan nama basis data dengan melakukan hal berikut:
   - Pilih semua sesuai opsi default-nya, kemudian gulir ke bawah ke bagian Konfigurasi tambahan.
   - Perluas bagian ini dan masukkan **`ExampleDB`** sebagai Nama basis data awal.
7. Pilih semua sesuai opsi default-nya dan pilih Buat basis data.

Buat fungsi Lambda dan proksi



Anda dapat menggunakan konsol RDS untuk membuat fungsi Lambda dan proksi di VPC yang sama dengan basis data.

> **Note**
>
> Anda hanya dapat membuat sumber daya terkait ini jika basis data Anda telah selesai dibuat dan berstatus Tersedia.

Untuk membuat fungsi dan proksi terkait

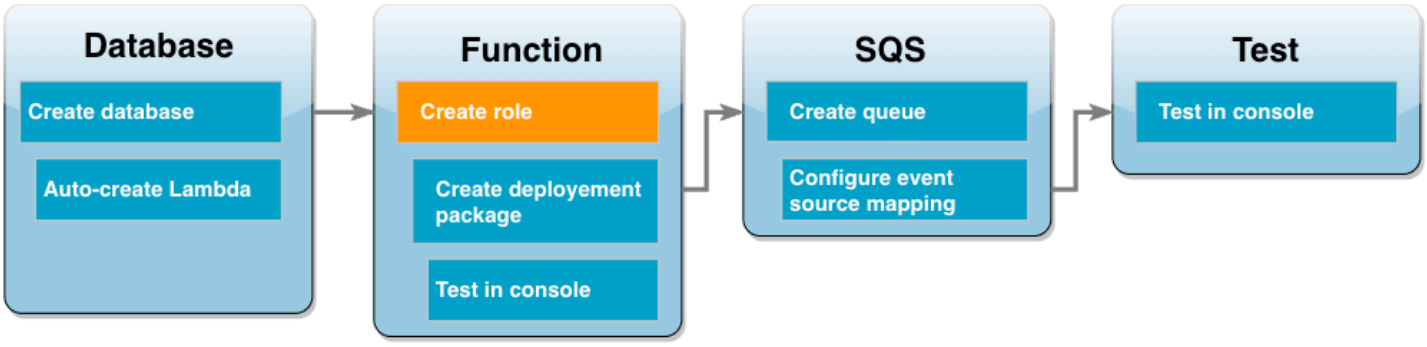
1. Dari halaman Basis Data, periksa apakah basis data Anda sudah berstatus Tersedia. Jika sudah, lanjutkan ke langkah berikutnya. Jika belum, tunggu hingga basis data tersedia.
2. Pilih basis data dan pilih Siapkan koneksi Lambda dari Tindakan.
3. Di halaman Siapkan koneksi Lambda, pilih Buat fungsi baru.

   Atur Nama fungsi Lambda baru ke **`LambdaFunctionWithRDS`**.
4. Di bagian Proksi RDS, pilih opsi Hubungkan menggunakan Proksi RDS. Selanjutnya pilih Buat proksi baru.

   - Untuk Kredensial basis data, pilih Nama pengguna dan kata sandi basis data.
   - Untuk Nama pengguna, pilih `admin`.
   - Untuk Kata sandi, masukkan kata sandi untuk instans basis data Anda.
5. Pilih Siapkan untuk menyelesaikan pembuatan proksi dan fungsi Lambda.

Wizard menyelesaikan penyiapan dan menyediakan tautan ke konsol Lambda untuk meninjau fungsi baru Anda. Catat titik akhir proksi sebelum beralih ke konsol Lambda.

Buat peran eksekusi fungsi



Sebelum membuat fungsi Lambda, Anda membuat peran eksekusi untuk memberikan izin yang diperlukan ke fungsi Anda. Untuk tutorial ini, Lambda memerlukan izin untuk mengelola koneksi

jaringan ke VPC yang berisi instans basis data Anda dan untuk melakukan polling pesan dari antrean Amazon SQS.

Untuk memberikan izin yang diperlukan oleh fungsi Lambda, tutorial ini menggunakan kebijakan terkelola IAM. Ini adalah kebijakan yang memberikan izin untuk banyak kasus penggunaan umum dan tersedia di Akun AWS Anda. Untuk informasi selengkapnya tentang penggunaan kebijakan terkelola, lihat Praktik terbaik kebijakan.

Untuk membuat peran eksekusi Lambda

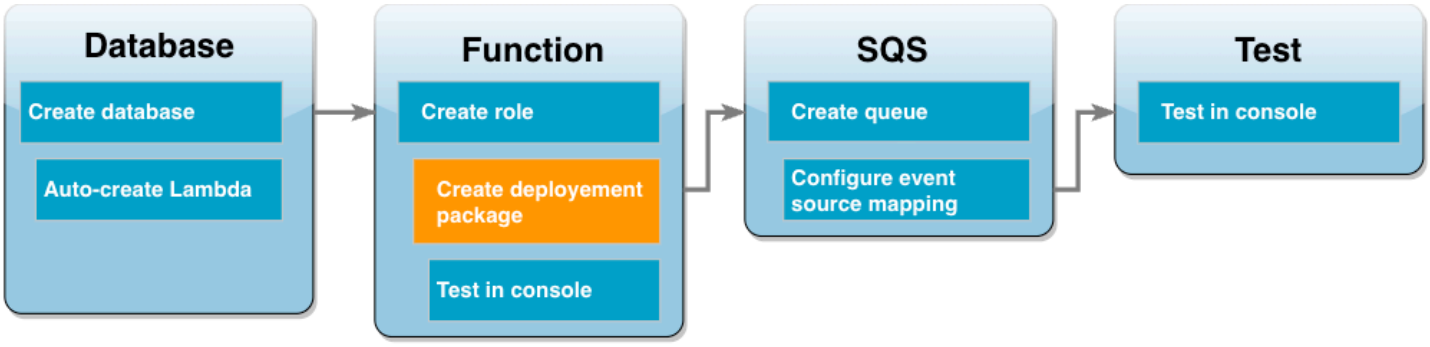
1. Buka halaman Peran di konsol IAM dan pilih Buat peran.
2. Untuk Jenis entitas tepercaya, pilih Layanan AWS , dan untuk Kasus penggunaan, pilih Lambda.
3. Pilih Berikutnya.
4. Tambahkan kebijakan terkelola IAM dengan melakukan hal berikut:
    a. Menggunakan kotak pencarian kebijakan, cari **`AWSLambdaSQSQueueExecutionRole`**.
    b. Di dalam daftar hasil, centang kotak di samping peran, lalu pilih Hapus filter.
    c. Menggunakan kotak pencarian kebijakan, cari **`AWSLambdaVPCAccessExecutionRole`**.
    d. Di dalam daftar hasil, centang kotak di samping peran, lalu pilih Berikutnya.
5. Untuk Nama peran, masukkan **`lambda-vpc-sqs-role`**, lalu pilih Buat peran.

Nantinya dalam tutorial ini, Anda memerlukan Amazon Resource Name (ARN) untuk peran eksekusi yang baru saja Anda buat.

Untuk menemukan ARN peran eksekusi

1. Buka halaman Peran di konsol IAM dan pilih peran (`lambda-vpc-sqs-role`) Anda.
2. Salin ARN yang ditampilkan di bagian Ringkasan.

Buat paket deployment Lambda



Contoh kode Python berikut menggunakan paket PyMySQL untuk membuka koneksi ke database Anda. Saat pertama kali diinvokasi, fungsi Anda juga membuat tabel baru bernama `Customer`. Tabel menggunakan skema berikut, dengan kunci primer `CustID`:

```
Customer(CustID, Name)
```

Fungsi ini juga menggunakan PyMy SQL untuk menambahkan catatan ke tabel ini. Fungsi ini menambahkan catatan menggunakan ID pelanggan dan nama yang ditentukan dalam pesan yang akan Anda tambahkan ke antrean Amazon SQS.

Kode ini membuat koneksi ke basis data Anda di luar fungsi handler. Membuat koneksi dalam kode inisialisasi memungkinkan koneksi untuk digunakan kembali oleh invokasi fungsi berikutnya dan meningkatkan performa. Dalam aplikasi produksi, Anda juga dapat menggunakan konkurensi yang tersedia untuk menginisialisasi sejumlah permintaan koneksi basis data. Koneksi ini langsung tersedia setelah fungsi Anda dipanggil.

```
import sys
import logging
import pymysql
import json
import os

# rds settings
user_name = os.environ['USER_NAME']
password = os.environ['PASSWORD']
rds_proxy_host = os.environ['RDS_PROXY_HOST']
db_name = os.environ['DB_NAME']

logger = logging.getLogger()
logger.setLevel(logging.INFO)
```

```
# create the database connection outside of the handler to allow connections to be
# re-used by subsequent function invocations.
try:
        conn = pymysql.connect(host=rds_proxy_host, user=user_name, passwd=password,
 db=db_name, connect_timeout=5)
except pymysql.MySQLError as e:
    logger.error("ERROR: Unexpected error: Could not connect to MySQL instance.")
    logger.error(e)
    sys.exit(1)

logger.info("SUCCESS: Connection to RDS for MySQL instance succeeded")

def lambda_handler(event, context):
    """
    This function creates a new RDS database table and writes records to it
    """
    message = event['Records'][0]['body']
    data = json.loads(message)
    CustID = data['CustID']
    Name = data['Name']

    item_count = 0
    sql_string = f"insert into Customer (CustID, Name) values(%s, %s)"

    with conn.cursor() as cur:
        cur.execute("create table if not exists Customer ( CustID  int NOT NULL, Name
 varchar(255) NOT NULL, PRIMARY KEY (CustID))")
        cur.execute(sql_string, (CustID, Name))
        conn.commit()
        cur.execute("select * from Customer")
        logger.info("The following items have been added to the database:")
        for row in cur:
            item_count += 1
            logger.info(row)
    conn.commit()

    return "Added %d items to RDS for MySQL table" %(item_count)
```

> **Note**
>
> Dalam contoh ini, kredensial akses basis data Anda disimpan sebagai variabel lingkungan. Dalam aplikasi produksi, sebaiknya gunakan AWS Secrets Manager sebagai opsi yang lebih aman. Perlu diketahui bahwa jika fungsi Lambda Anda berada di dalam VPC, untuk terhubung ke Secrets Manager, Anda perlu membuat titik akhir VPC. Untuk mempelajari selengkapnya, lihat Cara terhubung ke layanan Secrets Manager dalam Virtual Private Cloud.

Untuk menyertakan ketergantungan PyMy SQL dengan kode fungsi Anda, buat paket.zip deployment. Perintah berikut dapat digunakan untuk Linux, macOS, atau Unix:

Untuk membuat paket deployment .zip

1. Simpan kode contoh sebagai file bernama `lambda_function.py`.
2. Di direktori yang sama di mana Anda membuat `lambda_function.py` file Anda, buat direktori baru bernama `package` dan instal perpustakaan PyMy SQL.

   ```
   mkdir package
   pip install --target package pymysql
   ```

3. Buat file zip yang berisi kode aplikasi Anda dan perpustakaan PyMy SQL. Di Linux atau macOS, jalankan perintah CLI berikut. Di Windows, gunakan alat zip pilihan Anda untuk membuat file `lambda_function.zip`. File kode sumber `lambda_function.py` dan folder yang berisi dependensi Anda harus diinstal di root file .zip.

   ```
   cd package
   zip -r ../lambda_function.zip .
   cd ..
   zip lambda_function.zip lambda_function.py
   ```

   Anda juga dapat membuat paket deployment menggunakan lingkungan virtual Python. Lihat Melakukan deployment fungsi Lambda Python dengan arsip file .zip.

Perbarui fungsi Lambda

Menggunakan paket .zip yang baru saja Anda buat, perbarui fungsi Lambda Anda menggunakan konsol Lambda. Agar fungsi dapat mengakses basis data, variabel lingkungan juga perlu dikonfigurasikan dengan kredensial akses.

Untuk memperbarui fungsi Lambda

1. Buka halaman Fungsi di konsol Lambda dan pilih fungsi `LambdaFunctionWithRDS`.
2. Di tab Runtime settings, pilih Edit untuk mengubah Runtime fungsi ke Python 3.10.
3. Ubah Handler ke `lambda_function.lambda_handler`.
4. Di tab Kode, pilih Unggah dari kemudian File .zip.
5. Pilih file `lambda_function.zip` yang Anda buat di tahap sebelumnya dan pilih Simpan.

Sekarang konfigurasikan fungsi dengan peran eksekusi yang Anda buat sebelumnya. Dengan demikian, fungsi dapat memperoleh izin yang diperlukan untuk mengakses instans basis data Anda dan melakukan polling antrean Amazon SQS.

Untuk mengonfigurasi peran eksekusi fungsi

1. Di halaman Fungsi pada konsol Lambda, pilih tab Konfigurasi, lalu pilih Izin.
2. Di Peran eksekusi, pilih Edit.
3. Di Peran yang ada, pilih peran eksekusi Anda (`lambda-vpc-sqs-role`).
4. Pilih Simpan.

Untuk mengonfigurasi variabel lingkungan fungsi Anda

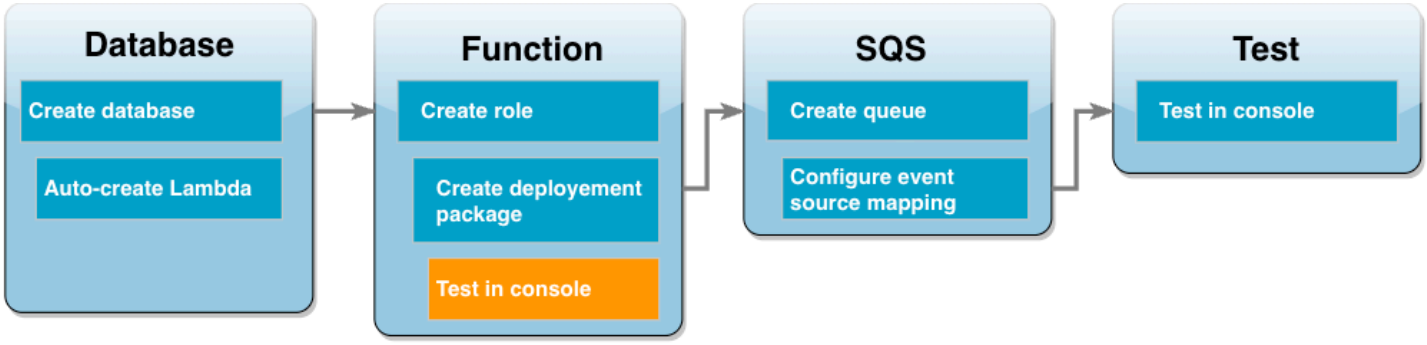
1. Di halaman Fungsi pada konsol Lambda, pilih tab Konfigurasi, lalu pilih Variabel lingkungan.
2. Pilih Edit.
3. Untuk menambahkan kredensial akses basis data, lakukan hal berikut:
    a. Pilih Tambahkan variabel lingkungan, lalu untuk Kunci masukkan **`USER_NAME`** dan untuk Nilai masukkan **`admin`**.
    b. Pilih Tambahkan variabel lingkungan, lalu untuk Kunci masukkan **`DB_NAME`** dan untuk Nilai masukkan **`ExampleDB`**.

c. Pilih Tambahkan variabel lingkungan, lalu untuk Kunci masukkan **PASSWORD** dan untuk Nilai masukkan kata sandi yang Anda pilih saat membuat basis data.

d. Pilih Tambahkan variabel lingkungan, lalu untuk Kunci masukkan **RDS_PROXY_HOST** dan untuk Nilai masukkan titik akhir Proksi RDS yang Anda catat sebelumnya.

e. Pilih Simpan.

Uji fungsi Lambda Anda di konsol



Anda dapat menggunakan konsol Lambda untuk menguji fungsi. Anda membuat peristiwa pengujian yang meniru data yang akan diterima fungsi saat diinvokasi menggunakan Amazon SQS pada tahap akhir tutorial ini. Peristiwa pengujian berisi objek JSON yang menentukan ID pelanggan dan nama pelanggan yang akan ditambahkan ke tabel `Customer` buatan fungsi Anda.

Untuk menguji fungsi Lambda

1. Buka halaman Fungsi di konsol Lambda dan pilih fungsi Anda.
2. Pilih bagian Uji.
3. Pilih Buat peristiwa baru dan masukkan **myTestEvent** sebagai nama peristiwa.
4. Salin kode berikut ke JSON peristiwa dan pilih Simpan.

```
{
  "Records": [
    {
      "messageId": "059f36b4-87a3-44ab-83d2-661975830a7d",
      "receiptHandle": "AQEBwJnKyrHigUMZj6rYigCgxlaS3SLy0a...",
      "body": "{\n     \"CustID\": 1021,\n     \"Name\": \"Martha Rivera\"\n}",
      "attributes": {
        "ApproximateReceiveCount": "1",
        "SentTimestamp": "1545082649183",
        "SenderId": "AIDAIENQZJOLO23YVJ4VO",
```

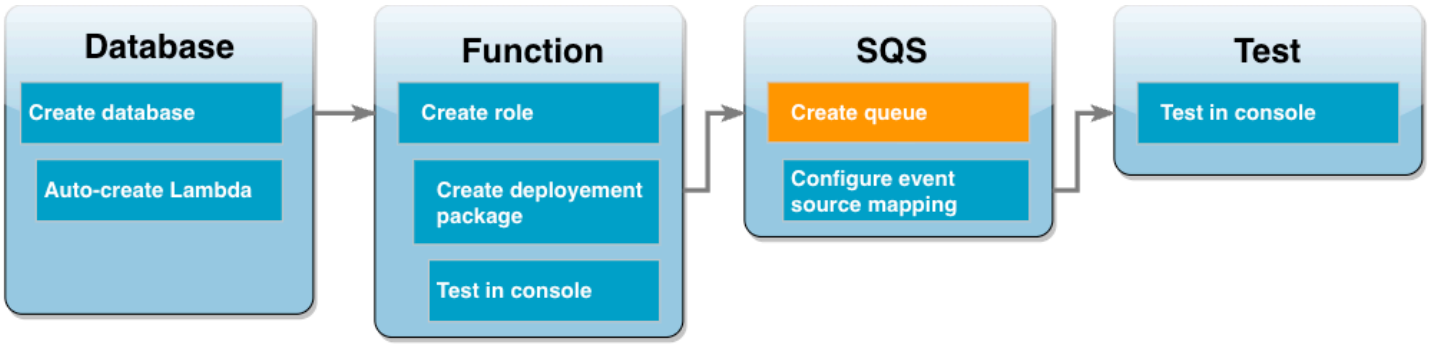
```
        "ApproximateFirstReceiveTimestamp": "1545082649185"
      },
      "messageAttributes": {},
      "md5OfBody": "e4e68fb7bd0e697a0ae8f1bb342846b3",
      "eventSource": "aws:sqs",
      "eventSourceARN": "arn:aws:sqs:us-west-2:123456789012:my-queue",
      "awsRegion": "us-west-2"
    }
  ]
}
```

5. Pilih Uji.

Di tab Hasil eksekusi, Anda akan melihat hasil yang mirip dengan yang ditampilkan di Log Fungsi:

```
[INFO] 2023-02-14T19:31:35.149Z bdd06682-00c7-4d6f-9abb-89f4bbb4a27f The following
 items have been added to the database:
[INFO] 2023-02-14T19:31:35.149Z bdd06682-00c7-4d6f-9abb-89f4bbb4a27f (1021, 'Martha
 Rivera')
```

Buat antrean Amazon SQS

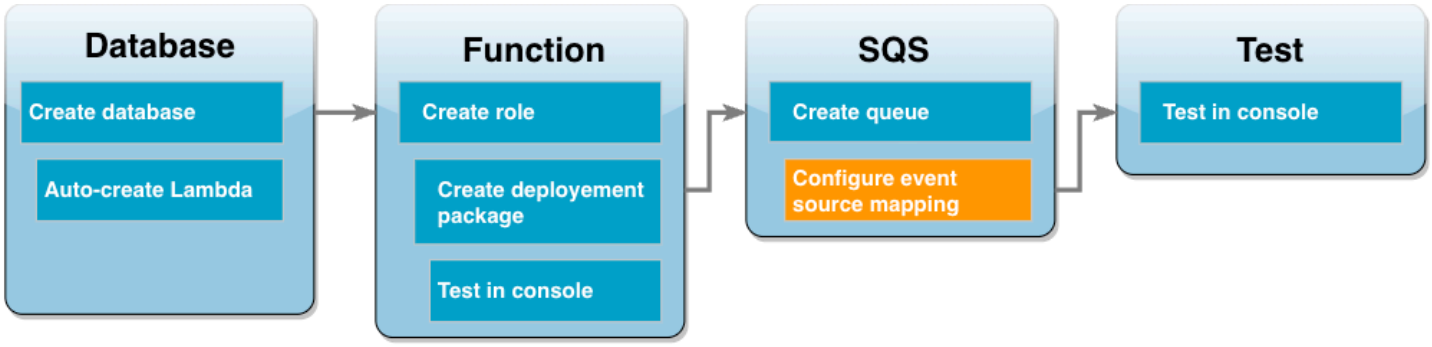


Anda telah berhasil menguji integrasi fungsi Lambda dan instans basis data Amazon RDS. Sekarang Anda membuat antrean Amazon SQS yang akan Anda gunakan untuk menginvokasi fungsi Lambda di tahap akhir tutorial ini.

Untuk membuat antrean Amazon SQS (konsol)

1. Buka halaman Antrean di konsol Amazon SQS dan pilih Buat antrean.
2. Di bagian Jenis pilih Standar masukkan **`LambdaRDSQueue`** untuk nama antrean.
3. Pilih semua sesuai opsi default-nya dan pilih Buat antrean.

Buat pemetaan sumber peristiwa untuk menginvokasi fungsi Lambda Anda



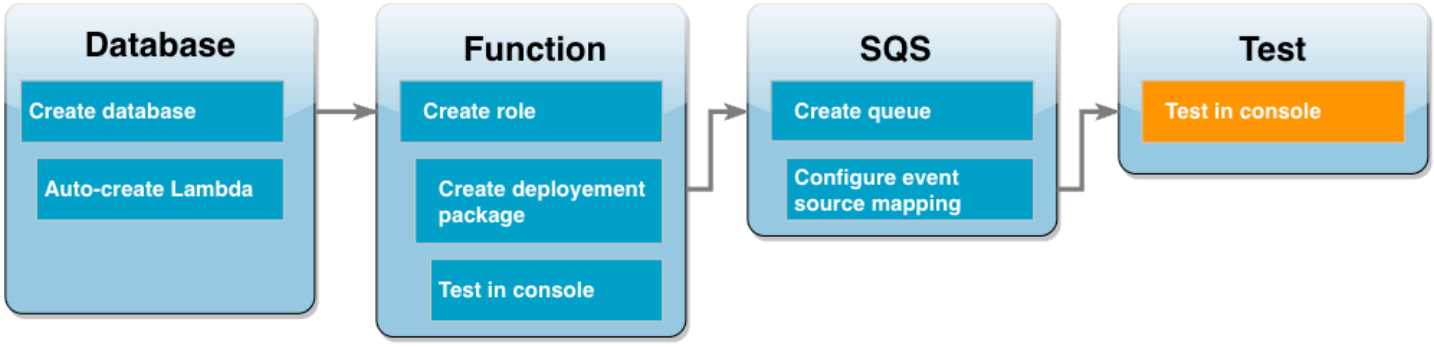
Pemetaan sumber peristiwa adalah sumber daya Lambda yang membaca item dari aliran atau antrean dan menginvokasi fungsi Lambda. Saat mengonfigurasi pemetaan sumber peristiwa, Anda dapat menentukan ukuran batch sehingga catatan dari aliran atau antrean Anda dikumpulkan ke dalam satu payload. Dalam contoh ini, Anda mengatur ukuran batch ke 1 sehingga fungsi Lambda akan diinvokasi setiap kali Anda mengirim sebuah pesan ke antrean. Anda dapat mengonfigurasi pemetaan sumber peristiwa menggunakan konsol AWS CLI atau Lambda.

Untuk membuat pemetaan sumber peristiwa (konsol)

1. Buka halaman Fungsi di konsol Lambda dan pilih fungsi Anda (`LambdaFunctionWithRDS`).
2. Di bagian Ikhtisar fungsi, pilih Tambahkan pemicu.
3. Untuk sumbernya, pilih Amazon SQS, lalu pilih nama antrean (`LambdaRDSQueue`).
4. Untuk Ukuran batch, masukkan **1**.
5. Biarkan semua opsi lain diatur ke nilai default dan pilih Tambahkan.

Kini semua penyiapan Anda sudah siap diuji dengan menambahkan pesan ke antrean Amazon SQS Anda.

Uji dan pantau pengaturan Anda



Untuk menguji seluruh pengaturan Anda, tambahkan pesan ke antrean Amazon SQS menggunakan konsol. Anda kemudian menggunakan CloudWatch Log untuk mengonfirmasi bahwa fungsi Lambda Anda menulis catatan ke database Anda seperti yang diharapkan.

Untuk menguji dan memantau pengaturan Anda

1. Buka halaman Antrean di konsol Amazon SQS dan pilih antrean (`LambdaRDSQueue`).
2. Pilih Kirim dan terima pesan dan tempelkan JSON berikut ke Konten pesan di bagian Kirim pesan.

```
{
    "CustID": 1054,
    "Name": "Richard Roe"
}
```

3. Pilih Kirim pesan.

   Mengirim pesan ke antrean akan membuat Lambda menginvokasi fungsi Anda melalui pemetaan sumber peristiwa. Untuk mengonfirmasi bahwa Lambda telah memanggil fungsi Anda seperti yang diharapkan, gunakan CloudWatch Log untuk memverifikasi bahwa fungsi tersebut telah menulis nama pelanggan dan ID ke tabel database Anda.
4. Buka halaman Grup log CloudWatch konsol dan pilih grup log untuk fungsi Anda (`/aws/lambda/LambdaFunctionWithRDS`).
5. Di bagian Log stream, pilih log stream terbaru.

   Tabel Anda harus berisi dua catatan pelanggan, satu dari setiap invokasi fungsi Anda. Di log stream, Anda akan melihat pesan yang mirip dengan pesan berikut:

```
[INFO] 2023-02-14T19:06:43.873Z 45368126-3eee-47f7-88ca-3086ae6d3a77 The following
 items have been added to the database:
[INFO] 2023-02-14T19:06:43.873Z 45368126-3eee-47f7-88ca-3086ae6d3a77 (1021, 'Martha
 Rivera')
[INFO] 2023-02-14T19:06:43.873Z 45368126-3eee-47f7-88ca-3086ae6d3a77 (1054,
 'Richard Roe')
```

Bersihkan sumber daya Anda

Sekarang Anda dapat menghapus sumber daya yang Anda buat untuk tutorial ini, kecuali Anda ingin mempertahankannya. Dengan menghapus AWS sumber daya yang tidak lagi Anda gunakan, Anda mencegah tagihan yang tidak perlu ke AWS akun Anda.

Untuk menghapus fungsi Lambda

1. Buka halaman Fungsi di konsol Lambda.
2. Pilih fungsi yang Anda buat.
3. Pilih Tindakan, Hapus.
4. Pilih Hapus.

Untuk menghapus peran eksekusi

1. Buka halaman Peran dari konsol IAM.
2. Pilih peran eksekusi yang Anda buat.
3. Pilih Hapus peran.
4. Pilih Ya, Hapus.

Untuk menghapus instans DB MySQL

1. Buka halaman Basis Data di konsol Amazon RDS.
2. Pilih basis data yang Anda buat.
3. Pilih Tindakan, Hapus.
4. Hapus kotak centang Buat snapshot terakhir.
5. Masukkan **`delete me`** di kotak teks.
6. Pilih Hapus.

Untuk menghapus antrean Amazon SQS

1. Masuk ke AWS Management Console dan buka konsol Amazon SQS di https://console.aws.amazon.com/sqs/.
2. Pilih antrean yang Anda buat.
3. Pilih Hapus.

4. Masukkan **`delete`** di kotak teks.
5. Pilih Hapus.

Tutorial Amazon RDS dan kode sampel

AWS Dokumentasi mencakup beberapa tutorial yang memandu Anda melalui kasus penggunaan Amazon RDS Aurora yang umum. Banyak dari tutorial ini menunjukkan cara menggunakan Amazon RDS Aurora dengan AWS layanan lain. Selain itu, Anda dapat mengakses kode sampel di GitHub.

> **Note**
>
> Tutorial lainnya dapat dilihat di Blog Basis Data AWS. Untuk informasi tentang pelatihan, lihat Pelatihan dan Sertifikasi AWS.

Topik

Tutorial dalam panduan ini

Tutorial berikut dalam panduan ini menunjukkan cara melakukan tugas umum dengan Amazon RDS:

- Tutorial: Membuat VPC untuk digunakan dengan instans DB (khusus IPv4)

  Pelajari cara menyertakan instans DB dalam cloud privat virtual (VPC) berdasarkan layanan Amazon VPC. Dalam hal ini, VPC membagikan data dengan server web yang dijalankan di instans Amazon EC2 dalam VPC yang sama.
- Tutorial: Membuat VPC untuk digunakan dengan instans DB (mode dual-stack)

  Pelajari cara menyertakan instans DB dalam cloud privat virtual (VPC) berdasarkan layanan Amazon VPC. Dalam hal ini, VPC membagikan data dengan instans Amazon EC2 dalam VPC yang sama. Dalam tutorial ini, Anda akan membuat VPC untuk skenario ini yang berfungsi dengan basis data yang berjalan dalam mode tumpukan ganda.

- Tutorial: Membuat server web dan instans DB Amazon RDS

  Pelajari cara menginstal server web Apache dengan PHP dan membuat basis data MySQL. Server web yang berjalan di instans Amazon EC2 menggunakan Amazon Linux, dan basis data MySQL adalah instans DB MySQL. Kedua instans Amazon EC2 dan instans DB tersebut berjalan di Amazon VPC.
- Tutorial: Memulihkan instans DB Amazon RDS dari snapshot DB

  Pelajari cara memulihkan instans DB dari snapshot DB.
- Tutorial: Menggunakan fungsi Lambda untuk mengakses basis data Amazon RDS

  Pelajari cara membuat fungsi Lambda dari konsol RDS untuk mengakses basis data melalui proksi, membuat tabel, menambahkan beberapa catatan, dan mengambil catatan dari tabel. Pelajari juga cara menginvokasi fungsi Lambda dan memverifikasikan hasil kueri.
- Tutorial: Menggunakan tag untuk menentukan instans DB yang akan dihentikan

  Pelajari cara menggunakan tag untuk menentukan instans DB yang akan dihentikan.
- Tutorial: Mencatat log perubahan status instans DB menggunakan Amazon EventBridge

  Pelajari cara mencatat perubahan status instans DB menggunakan Amazon EventBridge dan AWS Lambda.
- Tutorial: Membuat alarm Amazon CloudWatch untuk kelambatan replika klaster basis data Multi-AZ

  Pelajari cara membuat CloudWatch alarm yang mengirimkan pesan Amazon SNS saat lag replika untuk cluster DB multi-AZ telah melampaui ambang batas. Alarm mengawasi metrik `ReplicaLag` selama periode waktu yang Anda tentukan. Tindakannya adalah notifikasi yang dikirim ke topik Amazon SNS atau kebijakan Amazon EC2 Auto Scaling.

Tutorial dalam AWS panduan lain

Tutorial berikut di AWS panduan lain menunjukkan kepada Anda cara melakukan tugas-tugas umum dengan Amazon RDS Aurora:

- Tutorial: Memutar Rahasia untuk AWS Database di AWS Secrets Manager Panduan Pengguna

  Pelajari cara membuat rahasia untuk AWS database dan mengkonfigurasi rahasia untuk memutar pada jadwal. Anda memicu satu rotasi secara manual, kemudian mengonfirmasi bahwa versi baru rahasia terus memberikan akses.

- Tutorial dan sampel di Panduan Developer AWS Elastic Beanstalk

  Pelajari cara menerapkan aplikasi yang menggunakan database Amazon RDS. AWS Elastic Beanstalk

- Menggunakan Data dari Basis Data Amazon RDS untuk Membuat Sumber Data Amazon ML di Panduan Developer Amazon Machine Learning

  Pelajari cara membuat objek sumber data Amazon Machine Learning (Amazon ML) dari data yang disimpan dalam instans DB MySQL.

- Mengaktifkan Akses ke Instans Amazon RDS secara Manual di VPC di Panduan Pengguna Amazon QuickSight

  Pelajari cara mengaktifkan QuickSight akses Amazon ke instans Amazon RDS DB di VPC.

AWS lokakarya dan portal konten lab untuk Amazon RDS Aurora PostgreSQL

Kumpulan lokakarya dan konten praktis berikut ini membantu Anda memperoleh pemahaman tentang fitur dan kapabilitas Amazon RDS PostgreSQL:

- Membuat instans DB

  Pelajari cara membuat instans DB.

- Pemantauan Performa dengan Alat RDS

  Pelajari cara menggunakan AWS dan alat SQL (Cloudwatch, Enhanced Monitoring, Log Kueri Lambat, Performance Insights, PostgreSQL Catalog Views) untuk memahami masalah kinerja dan mengidentifikasi cara untuk meningkatkan kinerja database Anda.

AWS lokakarya dan portal konten lab untuk Amazon RDS

Kumpulan lokakarya dan konten praktis berikut ini membantu Anda memperoleh pemahaman tentang fitur dan kapabilitas Amazon RDS MySQL:

- Membuat instans DB

  Pelajari cara membuat instans DB.

- Menggunakan Wawasan Performa

  Pelajari cara memantau dan menyetel instans DB Anda menggunakan Wawasan performa.

Tutorial dan kode sampel di GitHub

Tutorial dan kode contoh berikut GitHub menunjukkan kepada Anda cara melakukan tugas-tugas umum dengan Amazon RDS Aurora:

- Membuat pelacak item Amazon Relational Database Service

  Pelajari cara membuat aplikasi yang melacak dan melaporkan item pekerjaan. Aplikasi ini menggunakan Amazon RDS, Amazon Simple Email Service, Elastic Beanstalk, dan SDK for Java 2.x.

Menggunakan layanan ini dengan AWS SDK

AWS kit pengembangan perangkat lunak (SDK) 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
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 SDK for Python (Boto3)	AWS SDK for Python (Boto3) contoh kode

Dokumentasi SDK	Contoh kode
AWS SDK for Ruby	AWS SDK for Ruby contoh kode
AWS SDK for Rust	AWS SDK for Rust contoh kode
AWS SDK untuk SAP ABAP	AWS SDK untuk SAP ABAP contoh kode
AWS SDK for Swift	AWS SDK for Swift contoh kode

Untuk contoh yang spesifik untuk layanan ini, lihat Contoh kode untuk Amazon RDS menggunakan AWS SDK.

Ketersediaan contoh

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

Praktik terbaik untuk Amazon RDS

Pelajari praktik terbaik untuk menggunakan Amazon RDS. Saat praktik terbaik yang baru diidentifikasi, kami akan terus memperbarui bagian ini.

Topik

> **Note**
>
> Untuk rekomendasi umum terkait Amazon RDS, lihat Melihat dan menanggapi rekomendasi Amazon Aurora RDS.

Pedoman operasional dasar Amazon RDS

Berikut ini adalah pedoman operasional dasar yang harus diikuti setiap orang saat menggunakan Amazon RDS. Perhatikan bahwa Perjanjian Tingkat Layanan Amazon RDS mewajibkan Anda untuk mengikuti pedoman ini:

- Gunakan metrik untuk memantau memori, CPU, jeda replika, dan penggunaan penyimpanan Anda. Anda dapat mengatur Amazon CloudWatch untuk memberi tahu Anda saat pola penggunaan

berubah atau saat penerapan mendekati batas kapasitas. Ini memungkinkan Anda untuk mempertahankan kinerja dan ketersediaan sistem.

- Tingkatkan skala instans DB Anda saat mendekati batas kapasitas penyimpanan. Anda akan memiliki buffer dalam penyimpanan dan memori untuk mengakomodasi peningkatan permintaan yang tidak terduga dari aplikasi Anda.
- Aktifkan pencadangan otomatis dan atur periode pencadangan agar pencadangan dilakukan selama IOPS tulis terendah harian. Pada saat itulah pencadangan tidak terlalu mengganggu penggunaan basis data Anda.
- Jika beban kerja basis data Anda memerlukan lebih banyak I/O daripada yang Anda sediakan, pemulihan setelah failover atau kegagalan basis data akan lambat. Untuk meningkatkan kapasitas I/O instans DB, lakukan salah satu atau semua hal berikut:
  - Migrasi ke kelas instans DB lain dengan kapasitas I/O tinggi.
  - Konversi dari penyimpanan magnetis ke penyimpanan Tujuan Umum atau IOPS yang Tersedia, bergantung pada jumlah peningkatan yang Anda butuhkan. Untuk informasi tentang jenis penyimpanan yang tersedia, lihat Jenis penyimpanan Amazon RDS.

    Jika Anda mengonversi ke penyimpanan IOPS yang Tersedia, pastikan Anda juga menggunakan kelas instans DB yang dioptimalkan untuk IOPS yang Tersedia. Untuk informasi tentang IOPS yang Tersedia, lihat Penyimpanan SSD IOPS yang Tersedia.
  - Jika Anda sudah menggunakan penyimpanan IOPS yang Tersedia, sediakan kapasitas throughput tambahan.
- Jika aplikasi klien Anda menyimpan data Domain Name Service (DNS) dari instans DB Anda, tetapkan nilai time-to-live (TTL) kurang dari 30 detik. Alamat IP yang mendasari untuk instans DB dapat berubah setelah failover. Menyimpan data DNS dalam cache untuk waktu yang lama dapat menyebabkan kegagalan koneksi. Aplikasi Anda mungkin mencoba untuk menghubungkan ke alamat IP yang sudah tidak berada dalam layanan.
- Uji failover untuk instans DB Anda guna mengetahui berapa lama proses untuk kasus penggunaan khusus Anda. Uji failover untuk memastikan bahwa aplikasi yang mengakses instans DB Anda dapat terhubung secara otomatis ke instans DB baru setelah failover terjadi.

Rekomendasi RAM instans DB

Praktik terbaik performa Amazon RDS adalah mengalokasikan RAM yang cukup sehingga set kerja Anda hampir seluruhnya berada di memori. Set kerja adalah data dan indeks yang sering Anda

gunakan pada instans. Makin banyak Anda menggunakan instans DB, makin banyak set kerja yang akan tumbuh.

Untuk mengetahui apakah set kerja Anda hampir semuanya ada dalam memori, periksa metrik ReadiOps (menggunakan CloudWatch Amazon) saat instans DB sedang dimuat. Nilai ReadIOPS harus kecil dan stabil. Dalam beberapa kasus, penskalaan kelas instans DB ke kelas yang memiliki lebih banyak RAM menghasilkan penurunan ReadIOPS yang signifikan. Dalam kasus ini, set kerja Anda hampir tidak seluruhnya berada di memori. Terus naikkan skala hingga ReadIOPS tidak lagi turun secara drastis setelah operasi penskalaan, atau ReadIOPS berkurang dengan jumlah yang sangat kecil. Untuk informasi tentang pemantauan metrik instans DB, lihat Melihat metrik di konsol Amazon RDS.

Menggunakan Pemantauan yang Ditingkatkan untuk mengidentifikasi masalah sistem operasi

Jika Pemantauan yang Ditingkatkan diaktifkan, Amazon RDS menyediakan metrik secara waktu nyata untuk sistem operasi (OS) yang dijalankan oleh instans DB Anda. Anda dapat melihat metrik instans DB menggunakan konsol tersebut. Anda juga dapat menggunakan output Enhanced Monitoring JSON dari Amazon CloudWatch Logs dalam sistem pemantauan pilihan Anda. Untuk informasi selengkapnya tentang Pemantauan yang Ditingkatkan, lihat Memantau metrik OS dengan Pemantauan yang Disempurnakan.

Menggunakan metrik untuk mengidentifikasi masalah performa

Untuk mengidentifikasi masalah performa yang disebabkan oleh sumber daya yang tidak mencukupi dan kemacetan umum lainnya, Anda dapat memantau metrik yang tersedia untuk instans DB Amazon RDS Anda.

Melihat metrik performa

Anda harus memantau metrik performa secara rutin untuk mengetahui nilai rata-rata, maksimum, dan minimum dalam berbagai rentang waktu. Sehingga Anda dapat mengidentifikasi saat performa menurun. Anda juga dapat mengatur CloudWatch alarm Amazon untuk ambang metrik tertentu sehingga Anda diberi tahu jika mereka tercapai.

Untuk memecahkan masalah performa, penting untuk memahami performa dasar sistem. Saat Anda menyiapkan instans DB dan menjalankannya dengan beban kerja umum, catat nilai rata-rata, maksimum, dan minimum dari semua metrik performa. Lakukan hal tersebut pada sejumlah interval

yang berbeda (misalnya, satu jam, 24 jam, satu minggu, dua minggu). Tindakan ini dapat memberi Anda gambaran tentang kondisi normal. Sehingga membantu mendapatkan perbandingan untuk jam sibuk dan tidak sibuk. Kemudian, Anda dapat menggunakan informasi ini untuk mengidentifikasi saat performa turun di bawah tingkat standar.

Jika Anda menggunakan klaster DB Multi-AZ, pantau selisih waktu antara transaksi terbaru pada instans DB penulis dan transaksi terbaru yang diterapkan pada instans DB pembaca. Selisih ini disebut jeda replika. Untuk informasi selengkapnya, lihat Kelambatan replika dan klaster basis data Multi-AZ.

Anda dapat melihat gabungan Performance Insights dan CloudWatch metrik di dasbor Performance Insights dan memantau instans DB Anda. Untuk menggunakan tampilan pemantauan ini, Wawasan Performa harus diaktifkan untuk instans DB Anda. Untuk informasi tentang tampilan pemantauan ini, lihat Menampilkan metrik gabungan di konsol Amazon RDS.

Anda dapat membuat laporan analisis performa untuk periode waktu tertentu dan melihat wawasan yang diidentifikasi serta rekomendasi untuk menyelesaikan masalah. Untuk informasi selengkapnya, lihat Membuat laporan analisis performa.

Untuk melihat metrik performa

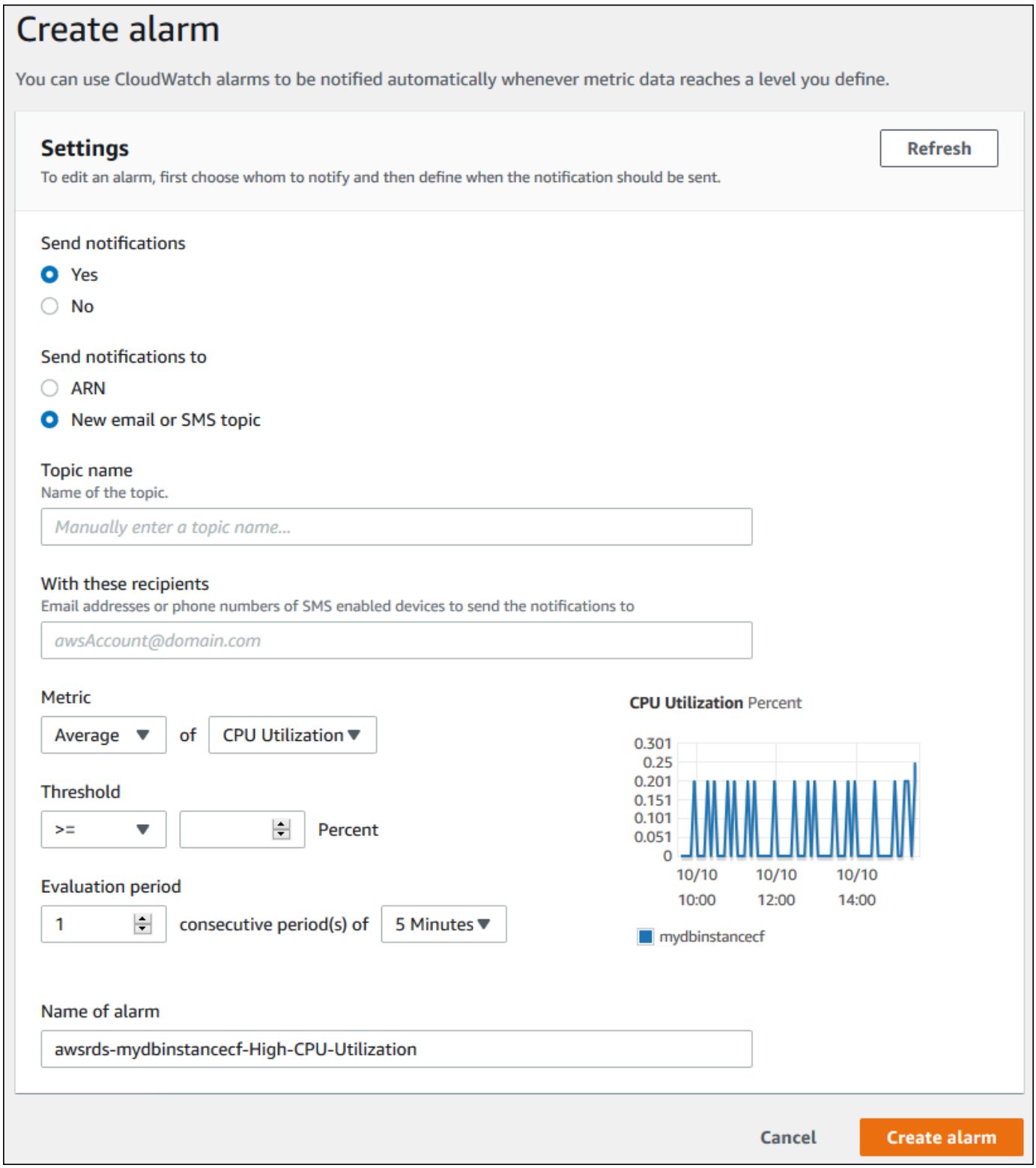
1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data, lalu pilih instans DB.
3. Pilih Pemantauan.

   Dasbor menyediakan metrik performa. Metrik default untuk menampilkan informasi selama tiga jam terakhir.
4. Gunakan tombol bernomor di kanan atas untuk menelusuri halaman metrik tambahan, atau sesuaikan pengaturan untuk melihat lebih banyak metrik.
5. Pilih metrik performa untuk menyesuaikan rentang waktu agar dapat melihat data untuk selain hari ini. Anda dapat mengubah nilai Statistik, Rentang Waktu, dan Periode untuk menyesuaikan informasi yang ditampilkan. Misalnya, Anda mungkin ingin melihat nilai puncak untuk metrik pada masing-masing hari dalam dua minggu terakhir. Jika demikian, atur Statistik ke Maksimum, Rentang Waktu ke 2 Minggu Terakhir, dan Periode ke Hari.

Anda juga dapat melihat metrik performa menggunakan CLI atau API. Untuk informasi selengkapnya, lihat Melihat metrik di konsol Amazon RDS.

Untuk mengatur CloudWatch alarm

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data, lalu pilih instans DB.
3. Pilih Log & peristiwa.
4. Di bagian CloudWatch alarm, pilih Buat alarm.



5. Untuk Kirim pemberitahuan, pilih Ya, dan untuk Kirim pemberitahuan ke, pilih Topik SMS atau email baru.

6. Untuk Nama topik, masukkan nama untuk pemberitahuan, dan untuk Dengan penerima ini, masukkan daftar alamat email dan nomor telepon yang dipisahkan dengan koma.
7. Untuk Metrik, pilih statistik dan metrik alarm yang akan diatur.
8. Untuk Ambang Batas, tentukan apakah metrik harus lebih besar dari, kurang dari, atau sama dengan ambang batas, dan tentukan nilai ambang batas.
9. Untuk Periode evaluasi, pilih periode evaluasi untuk alarm. Untuk periode berurutan, pilih periode ketika ambang batas harus sudah tercapai untuk memicu alarm.
10. Untuk Nama alarm, masukkan nama untuk alarm.
11. Pilih Buat Alarm.

Alarm muncul di bagian CloudWatch alarm.

Mengevaluasi metrik performa

Instans DB memiliki sejumlah kategori metrik yang berbeda, dan cara menentukan nilai yang dapat diterima bergantung pada metrik.

CPU

- Pemanfaatan CPU – Persentase kapasitas pemrosesan komputer yang digunakan.

Memori

- Freeable Memory — Berapa banyak RAM yang tersedia pada instans DB, dalam byte. Garis merah dalam metrik tab Pemantauan ditandai pada 75% untuk Metrik CPU, Memori, dan Penyimpanan. Jika penggunaan memori instans sering melewati garis tersebut, hal ini menunjukkan bahwa Anda harus memeriksa beban kerja atau meningkatkan instans.
- Penggunaan Swap — Berapa banyak ruang swap yang digunakan oleh instans DB, dalam byte.

Ruang disk

- Ruang Penyimpanan Kosong – Jumlah ruang disk yang saat ini tidak digunakan oleh instans DB, dalam megabyte.

Operasi input/output

- IOPS Baca, IOPS Tulis – Jumlah rata-rata operasi baca atau tulis disk per detik.
- Latensi Baca, Latensi Tulis – Waktu rata-rata untuk operasi baca atau tulis dalam milidetik.
- Throughput Baca, Throughput Tulis – Rata-rata jumlah megabyte yang dibaca dari atau ditulis ke disk per detik.
- Kedalaman Antrean – Jumlah operasi I/O yang menunggu untuk ditulis ke atau dibaca dari disk.

Lalu lintas jaringan

- Throughput Penerimaan Jaringan, Throughput Pengiriman Jaringan – Tingkat lalu lintas jaringan ke dan dari instans DB dalam byte per detik.

Koneksi basis data

- Koneksi DB – Jumlah sesi klien yang terhubung ke instans DB.

Untuk deskripsi individual yang lebih mendetail tentang metrik performa yang tersedia, lihat Memantau metrik Amazon RDS dengan Amazon CloudWatch.

Secara umum, nilai yang dapat diterima untuk metrik performa bergantung pada seperti apa garis dasar Anda dan apa yang dilakukan aplikasi Anda. Periksa varian yang konsisten atau sedang tren dari garis dasar Anda. Informasi tentang jenis metrik khusus sebagai berikut:

- Penggunaan CPU atau RAM yang tinggi – Nilai yang tinggi untuk penggunaan CPU atau RAM mungkin sesuai. Misalnya, hal tersebut mungkin terjadi jika sesuai dengan tujuan aplikasi Anda (seperti throughput atau konkurensi) dan memang diharapkan.
- Penggunaan ruang disk – Periksa penggunaan ruang disk jika ruang yang digunakan selalu berada di atau di atas 85 persen dari total ruang disk. Periksa apakah ada kemungkinan untuk menghapus data dari instans atau mengarsipkan data ke sistem yang berbeda guna mengosongkan sebagian ruang.
- Lalu lintas jaringan – Untuk lalu lintas jaringan, bicaralah kepada administrator sistem Anda untuk memahami throughput yang diharapkan bagi jaringan domain dan koneksi internet Anda. Periksa lalu lintas jaringan jika throughput selalu lebih rendah dari yang diharapkan.
- Koneksi basis data – Sebaiknya batasi koneksi basis data jika Anda melihat jumlah koneksi pengguna yang tinggi sehubungan dengan penurunan performa instans dan waktu respons.

Jumlah koneksi pengguna terbaik untuk instans DB Anda akan bervariasi berdasarkan kelas instans Anda dan kerumitan operasi yang dilakukan. Untuk menentukan jumlah koneksi basis data, kaitkan instans DB Anda dengan grup parameter. Dalam grup ini, atur parameter Koneksi Pengguna ke selain 0 (tidak terbatas). Anda dapat menggunakan grup parameter yang sudah ada atau membuat grup baru. Untuk informasi selengkapnya, lihat Bekerja dengan grup parameter.

- Metrik IOPS – Nilai yang diharapkan untuk metrik IOPS bergantung pada spesifikasi disk dan konfigurasi server; jadi, gunakan acuan dasar Anda untuk mengetahui nilai yang lazim. Periksa apakah nilainya selalu berbeda dari garis dasar Anda. Untuk performa IOPS terbaik, pastikan set kerja Anda sesuai dengan memori untuk meminimalkan operasi baca dan tulis.

Untuk masalah pada metrik performa, langkah pertama untuk meningkatkan performa adalah menyetel kueri yang paling sering digunakan dan paling menghabiskan biaya. Setel kueri tersebut untuk melihat apakah langkah ini dapat menurunkan tekanan pada sumber daya sistem. Untuk informasi selengkapnya, lihat Menyetel kueri.

Jika kueri sudah disetel dan masalah tetap ada, sebaiknya tingkatkan Amazon RDS Kelas instans DB . Anda dapat meningkatkannya ke versi yang memiliki lebih banyak sumber daya (CPU, RAM, ruang disk, bandwidth jaringan, kapasitas I/O) yang terkait dengan masalah.

Menyetel kueri

Salah satu cara terbaik untuk meningkatkan performa instans DB adalah dengan menyetel kueri yang paling sering digunakan dan paling sarat sumber daya. Di sini, Anda menyetelnya agar dapat dijalankan dengan biaya yang lebih terjangkau. Untuk informasi tentang peningkatan kueri, gunakan sumber daya berikut:

- MySQL – Lihat Optimizing SELECT statements dalam dokumentasi MySQL. Untuk sumber daya tambahan tentang penyetelan kueri, lihat Sumber daya penyetelan dan pengoptimalan performa MySQL.
- Oracle – Lihat Database SQL Tuning Guide dalam dokumentasi Oracle Database.
- SQL Server – Lihat Analyzing a query dalam dokumentasi Microsoft. Anda juga dapat menggunakan tampilan pengelolaan data terkait eksekusi, indeks, dan I/O (DMV) yang dijelaskan di System Dynamic Management Views dalam dokumentasi Microsoft untuk memecahkan masalah kueri SQL Server.

  Aspek umum penyetelan kueri adalah membuat indeks yang efektif. Untuk perbaikan indeks instans DB yang dapat dilakukan, lihat Database Engine Tuning Advisor dalam dokumentasi

Microsoft. Untuk informasi tentang penggunaan Tuning Advisor di RDS for SQL Server, lihat Menganalisis beban kerja basis data di instans DB Amazon RDS for SQL Server dengan basis data Engine Tuning Advisor.

- PostgreSQL – Lihat Using EXPLAIN dalam dokumentasi PostgreSQL untuk mempelajari cara menganalisis rencana kueri. Anda dapat menggunakan informasi ini untuk mengubah kueri atau tabel yang mendasarinya guna meningkatkan performa kueri.

  Untuk informasi tentang cara menentukan gabungan dalam kueri Anda untuk performa terbaik, lihat Mengontrol perencana dengan klausul JOIN eksplisit.
- MariaDB – Lihat Query optimizations dalam dokumentasi MariaDB.

Praktik terbaik dalam menggunakan MySQL

Ukuran tabel dan jumlah tabel dalam basis data MySQL dapat memengaruhi performa.

Ukuran tabel

Biasanya, batasan sistem operasi pada ukuran file menentukan ukuran tabel maksimum yang efektif untuk basis data MySQL. Jadi, batasannya biasanya tidak ditentukan oleh batasan MySQL internal.

Untuk instans DB MySQL, hindari tabel di basis data Anda yang tumbuh terlalu besar. Meskipun batas penyimpanan umum adalah 64 TiB, batas penyimpanan yang tersedia membatasi ukuran maksimum file tabel MySQL hingga 16 TiB. Partisi tabel besar Anda sehingga ukuran file berada di bawah batas 16 TiB. Pendekatan ini juga dapat meningkatkan performa dan waktu pemulihan. Untuk informasi selengkapnya, lihat Batas ukuran file MySQL di Amazon RDS.

Tabel yang sangat besar (dengan ukuran lebih dari 100 GB) dapat memengaruhi performa baik secara negatif untuk baca dan tulis (termasuk pernyataan DML dan terutama pernyataan DDL). Indeks pada tabel besar dapat secara signifikan meningkatkan performa tertentu, tetapi juga dapat menurunkan performa laporan DML. Pernyataan DDL, seperti `ALTER TABLE`, dapat menjadi lebih lambat secara signifikan untuk tabel besar karena operasi tersebut mungkin sepenuhnya membangun ulang tabel dalam beberapa kasus. Pernyataan DDL ini dapat mengunci tabel selama operasi berlangsung.

Jumlah memori yang diperlukan oleh MySQL untuk baca dan tulis tergantung pada tabel yang terlibat dalam operasi. Praktik terbaiknya adalah memiliki setidaknya RAM yang cukup untuk menyimpan indeks tabel yang digunakan secara aktif. Untuk menemukan sepuluh tabel dan indeks terbesar dalam basis data, gunakan kueri berikut:

```
select table_schema, TABLE_NAME, dat, idx from
(SELECT table_schema, TABLE_NAME,
       ( data_length ) / 1024 / 1024 as dat,
       ( index_length ) / 1024 / 1024 as idx
FROM information_schema.TABLES
order by 3 desc ) a
order by 3 desc
limit 10;
```

Jumlah tabel

Sistem file yang mendasarinya mungkin memiliki batas jumlah file yang mewakili tabel. Namun, MySQL tidak memiliki batasan jumlah tabel. Meskipun demikian, total jumlah tabel dalam mesin penyimpanan MySQL InnoDB dapat berkontribusi pada penurunan performa, terlepas dari ukuran tabel tersebut. Untuk membatasi dampak sistem operasi, Anda dapat memisahkan tabel di beberapa basis data dalam instans DB MySQL yang sama. Tindakan tersebut dapat membatasi jumlah file dalam direktori, tetapi tidak akan menyelesaikan masalah secara keseluruhan.

Penurunan performa karena sejumlah besar tabel (lebih dari 10 ribu) disebabkan oleh MySQL yang bekerja dengan file penyimpanan, termasuk membuka dan menutupnya. Untuk mengatasi masalah ini, Anda dapat meningkatkan ukuran parameter `table_open_cache` dan `table_definition_cache`. Namun, meningkatkan nilai parameter tersebut dapat secara signifikan meningkatkan jumlah penggunaan memori MySQL, dan bahkan mungkin menggunakan semua memori yang tersedia. Untuk informasi selengkapnya, lihat How MySQL Opens and Closes Tables dalam dokumentasi MySQL.

Selain itu, terlalu banyak tabel dapat secara signifikan memengaruhi waktu pemulaian MySQL. Pematian dan pengaktifan ulang yang bersih dan pemulihan crash dapat terpengaruh, khususnya dalam versi sebelum MySQL 8.0.

Kami merekomendasikan total kurang dari 10.000 tabel di semua basis data dalam instans DB. Untuk kasus penggunaan dengan sejumlah besar tabel dalam basis data, lihat Satu Juta Tabel di MySQL 8.0.

Mesin penyimpanan

Fitur point-in-time pemulihan dan pemulihan snapshot Amazon RDS for MySQL memerlukan mesin penyimpanan yang dapat dipulihkan dengan crash. Fitur ini hanya didukung untuk mesin penyimpanan InnoDB. Meskipun MySQL mendukung banyak mesin penyimpanan dengan berbagai

kemampuan, tidak semuanya dioptimalkan untuk pemulihan crash dan durabilitas data. Misalnya, mesin penyimpanan MyISAM tidak mendukung pemulihan kerusakan yang andal dan dapat mencegah point-in-time pemulihan atau pemulihan snapshot berfungsi sebagaimana dimaksud. Hal ini dapat mengakibatkan hilangnya atau rusaknya data ketika MySQL dimulai ulang setelah terjadi kerusakan.

InnoDB adalah mesin penyimpanan yang direkomendasikan dan didukung untuk instans DB MySQL di Amazon RDS. Instans InnoDB juga dapat dimigrasikan ke Aurora, sementara instans MyISAM tidak dapat dimigrasikan. Namun, MyISAM berperforma lebih baik daripada InnoDB jika Anda memerlukan pencarian teks penuh yang intens. Jika Anda masih memilih untuk menggunakan MyISAM dengan Amazon RDS, mengikuti langkah-langkah yang diuraikan di Cadangan otomatis dengan mesin penyimpanan MySQL yang tidak didukung dapat membantu dalam skenario tertentu untuk fungsi pemulihan snapshot.

Jika ingin mengonversi tabel MyISAM yang sudah ada ke tabel InnoDB, Anda dapat menggunakan proses yang diuraikan dalam dokumentasi MySQL. MyISAM dan InnoDB memiliki keunggulan dan kekurangan yang berbeda, jadi Anda harus sepenuhnya mengevaluasi dampak dari pembuatan penggantian ini pada aplikasi Anda sebelum melakukannya.

Selain itu, Federated Storage Engine saat ini tidak didukung oleh Amazon RDS for MySQL.

Praktik terbaik untuk menggunakan MariaDB

Ukuran tabel dan jumlah tabel dalam basis data MariaDB dapat memengaruhi performa.

Ukuran tabel

Biasanya, batasan sistem operasi pada ukuran file menentukan ukuran tabel maksimum yang efektif untuk basis data MariaDB. Jadi, batasannya biasanya tidak ditentukan oleh batasan MariaDB internal.

Untuk instans DB MariaDB, hindari tabel di basis data Anda yang tumbuh terlalu besar. Meskipun batas penyimpanan umum adalah 64 TiB, batas penyimpanan yang tersedia membatasi ukuran maksimum file tabel MariaDB hingga 16 TiB. Partisi tabel besar Anda sehingga ukuran file berada di bawah batas 16 TiB. Pendekatan ini juga dapat meningkatkan performa dan waktu pemulihan.

Tabel yang sangat besar (dengan ukuran lebih dari 100 GB) dapat memengaruhi performa baik secara negatif untuk baca dan tulis (termasuk pernyataan DML dan terutama pernyataan DDL).

Indeks pada tabel besar dapat secara signifikan meningkatkan performa tertentu, tetapi juga dapat menurunkan performa laporan DML. Pernyataan DDL, seperti `ALTER TABLE`, dapat menjadi lebih lambat secara signifikan untuk tabel besar karena operasi tersebut mungkin sepenuhnya membangun ulang tabel dalam beberapa kasus. Pernyataan DDL ini dapat mengunci tabel selama operasi berlangsung.

Jumlah memori yang diperlukan oleh MariaDB untuk baca dan tulis tergantung pada tabel yang terlibat dalam operasi. Praktik terbaiknya adalah memiliki setidaknya RAM yang cukup untuk menyimpan indeks tabel yang digunakan secara aktif. Untuk menemukan sepuluh tabel dan indeks terbesar dalam basis data, gunakan kueri berikut:

```
select table_schema, TABLE_NAME, dat, idx from
(SELECT table_schema, TABLE_NAME,
       ( data_length ) / 1024 / 1024 as dat,
       ( index_length ) / 1024 / 1024 as idx
FROM information_schema.TABLES
order by 3 desc ) a
order by 3 desc
limit 10;
```

Jumlah tabel

Sistem file yang mendasarinya mungkin memiliki batas jumlah file yang mewakili tabel. Namun, MariaDB tidak memiliki batasan jumlah tabel. Meskipun demikian, total jumlah tabel dalam mesin penyimpanan MariaDB InnoDB dapat berkontribusi pada penurunan performa, terlepas dari ukuran tabel tersebut. Untuk membatasi dampak sistem operasi, Anda dapat memisahkan tabel di beberapa basis data dalam instans DB MariaDB yang sama. Tindakan tersebut dapat membatasi jumlah file dalam direktori, tetapi tidak menyelesaikan masalah secara keseluruhan.

Penurunan performa karena sejumlah besar tabel (lebih dari 10.000) disebabkan oleh MariaDB yang bekerja dengan file penyimpanan. Pekerjaan ini mencakup pembukaan dan penutupan file penyimpanan MariaDB. Untuk mengatasi masalah ini, Anda dapat meningkatkan ukuran parameter `table_open_cache` dan `table_definition_cache`. Namun, meningkatkan nilai parameter tersebut dapat secara signifikan meningkatkan jumlah penggunaan memori MariaDB. Bahkan mungkin menggunakan semua memori yang tersedia. Untuk informasi selengkapnya, lihat Optimizing table_open_cache dalam dokumentasi MariaDB.

Selain itu, terlalu banyak tabel dapat secara signifikan memengaruhi waktu pemulaian MariaDB. Pematian dan pengaktifan ulang yang bersih dan pemulihan crash dapat terpengaruh. Kami

merekomendasikan jumlah total kurang dari sepuluh ribu tabel di semua basis data dalam instans DB.

Mesin penyimpanan

Fitur point-in-time pemulihan dan pemulihan snapshot Amazon RDS for MariaDB memerlukan mesin penyimpanan yang dapat dipulihkan dengan crash. Meskipun MariaDB mendukung banyak mesin penyimpanan dengan berbagai kemampuan, tidak semuanya dioptimalkan untuk pemulihan crash dan durabilitas data. Misalnya, meskipun Aria adalah pengganti yang aman untuk myISAM, Aria mungkin masih mencegah point-in-time pemulihan atau pemulihan snapshot berfungsi sebagaimana dimaksud. Hal ini dapat mengakibatkan hilangnya atau rusaknya data ketika MariaDB dimulai ulang setelah terjadi kerusakan. InnoDB adalah mesin penyimpanan yang direkomendasikan dan didukung untuk instans DB MariaDB di Amazon RDS. Jika Anda masih memilih untuk menggunakan Aria dengan Amazon RDS, mengikuti langkah-langkah yang diuraikan di Cadangan otomatis dengan mesin penyimpanan MariaDB yang tidak didukung dapat membantu dalam skenario tertentu untuk fungsi pemulihan snapshot.

Jika ingin mengonversi tabel MyISAM yang sudah ada ke tabel InnoDB, Anda dapat menggunakan proses yang diuraikan dalam dokumentasi MariaDB. MyISAM dan InnoDB memiliki keunggulan dan kekurangan yang berbeda, jadi Anda harus sepenuhnya mengevaluasi dampak dari pembuatan penggantian ini pada aplikasi Anda sebelum melakukannya.

Praktik terbaik untuk menggunakan Oracle

Informasi tentang praktik terbaik untuk menggunakan Amazon RDS for Oracle dapat dilihat di Praktik terbaik untuk menjalankan basis data Oracle di Amazon Web Services.

Lokakarya AWS virtual 2020 menyertakan presentasi tentang menjalankan basis data Oracle produksi di Amazon RDS. Video presentasi tersedia di sini.

Praktik terbaik untuk menggunakan PostgreSQL

Dari dua area penting tempat Anda dapat meningkatkan performa dengan RDS for PostgreSQL, salah satunya adalah saat memuat data ke instans DB. Area lainnya adalah saat menggunakan fitur autovacuum PostgreSQL. Bagian berikut mencakup beberapa praktik yang direkomendasikan untuk kedua area tersebut.

Untuk informasi tentang cara Amazon RDS mengimplementasikan tugas DBA umum untuk PostgreSQL lainnya, lihat Tugas DBA umum untuk Amazon RDS for PostgreSQL.

Memuat data ke dalam instans DB PostgreSQL

Saat memuat data ke instans DB Amazon RDS for PostgreSQL, ubah pengaturan instans DB dan nilai grup parameter DB Anda. Ubah pengaturan dan nilai tersebut untuk memungkinkan pengimporan data yang paling efisien ke instans DB Anda.

Ubah pengaturan instans DB Anda sebagai berikut:

- Nonaktifkan pencadangan instans DB (ubah backup_retention menjadi 0)
- Nonaktifkan Multi-AZ

Ubah grup parameter DB Anda untuk menyertakan pengaturan berikut. Selain itu, uji juga pengaturan parameter untuk menemukan pengaturan yang paling efisien untuk instans DB Anda.

- Tingkatkan nilai parameter `maintenance_work_mem`. Untuk informasi selengkapnya tentang parameter penggunaan sumber daya PostgreSQL, lihat dokumentasi PostgreSQL.
- Tingkatkan nilai parameter `max_wal_size` dan `checkpoint_timeout` untuk mengurangi jumlah penulisan ke log write-ahead log (WAL).
- Nonaktifkan parameter `synchronous_commit`.
- Nonaktifkan parameter autovacuum PostgreSQL.
- Pastikan tidak ada satu pun tabel yang Anda impor yang tidak masuk log. Data yang disimpan dalam tabel yang tidak masuk log dapat hilang selama failover. Untuk informasi selengkapnya, lihat MEMBUAT TABEL YANG TIDAK MASUK LOG.

Gunakan perintah `pg_dump -Fc` (terkompresi) atau `pg_restore -j` (paralel) dengan pengaturan ini.

Setelah operasi pemuatan selesai, pulihkan instans DB dan parameter DB Anda ke pengaturan normalnya.

Menggunakan fitur autovacuum PostgreSQL

Fitur autovacuum untuk basis data PostgreSQL adalah fitur yang sangat kami rekomendasikan bagi Anda untuk menjaga kondisi instans DB PostgreSQL. Autovacuum mengotomatiskan eksekusi perintah VACUUM dan ANALYZE. Penggunaan autovacuum diwajibkan oleh PostgreSQL, tidak diberlakukan oleh Amazon RDS, dan penggunaannya sangat penting untuk performa yang baik.

Fitur ini diaktifkan secara default untuk semua instans DB Amazon RDS for PostgreSQL baru, dan parameter konfigurasi terkait diatur dengan tepat secara default.

Administrator basis data Anda perlu mengetahui dan memahami operasi pemeliharaan ini. Untuk dokumentasi PostgreSQL tentang autovacuum, lihat Daemon Autovacuum.

Autovacuum bukan merupakan operasi "bebas sumber daya", tetapi operasi yang bekerja di latar belakang dan menghasilkan operasi pengguna sebanyak mungkin. Saat diaktifkan, autovacuum memeriksa tabel yang memiliki banyak tuple yang diperbarui atau dihapus. Autovacuum juga melindungi dari kehilangan data yang berumur sangat lama karena penyelesaian ID transaksi. Untuk informasi selengkapnya, lihat Mencegah kegagalan penyelesaian ID transaksi.

Autovacuum sebaiknya tidak dianggap sebagai operasi dengan overhead tinggi yang dapat dikurangi untuk mendapatkan performa yang lebih baik. Sebaliknya, tabel yang memiliki pembaruan dan penghapusan dengan kecepatan tinggi akan dengan cepat memburuk seiring waktu jika autovacuum tidak dijalankan.

> **Important**
>
> Tidak menjalankan autovacuum dapat mengakibatkan gangguan yang pada akhirnya diperlukan untuk melakukan operasi vakum yang jauh lebih mengganggu. Dalam beberapa kasus, instans DB RDS for PostgreSQL mungkin menjadi tidak tersedia karena penggunaan autovacuum yang terlalu konservatif. Dalam kasus ini, basis data PostgreSQL dimatikan untuk melindungi dirinya sendiri. Pada saat itu, Amazon RDS harus melakukan vakum single-user-mode penuh langsung pada instans DB. Vakum penuh ini dapat mengakibatkan gangguan selama beberapa jam. Oleh karena itu, kami sangat merekomendasikan agar Anda tidak mematikan autovacuum, yang diaktifkan secara default.

Parameter autovacuum menentukan kapan dan seberapa keras autovacuum bekerja. Parameter `autovacuum_vacuum_threshold` dan `autovacuum_vacuum_scale_factor` menentukan kapan autovacuum dijalankan. Parameter `autovacuum_max_workers`, `autovacuum_nap_time`, `autovacuum_cost_limit`, dan `autovacuum_cost_delay` menentukan seberapa keras autovacuum bekerja. Untuk informasi selengkapnya tentang autovacuum, kapan autovacuum berjalan, dan parameter apa yang diperlukan, lihat Routine Vacuuming dalam dokumentasi PostgreSQL.

Kueri berikut ini menunjukkan jumlah tuple "mati" dalam tabel dengan nama table1:

```
SELECT relname, n_dead_tup, last_vacuum, last_autovacuum FROM
pg_catalog.pg_stat_all_tables
WHERE n_dead_tup > 0 and relname = 'table1';
```

Hasil kueri akan seperti berikut ini:

```
relname | n_dead_tup | last_vacuum | last_autovacuum
---------+------------+-------------+-----------------
 tasks   |   81430522 |             |
(1 row)
```

Video praktik terbaik Amazon RDS for PostgreSQL

Konferensi re AWS : Invent 2020 menyertakan presentasi tentang fitur-fitur baru dan praktik terbaik untuk bekerja dengan PostgreSQL di Amazon RDS. Video presentasi tersedia di sini.

Praktik terbaik untuk menggunakan SQL Server

Praktik terbaik untuk deployment Multi-AZ dengan instans DB SQL Server mencakup hal-hal berikut:

- Gunakan peristiwa DB Amazon RDS untuk memantau failover. Misalnya, Anda akan mendapatkan pemberitahuan melalui pesan teks atau email jika ada kegagalan instans DB. Untuk informasi selengkapnya tentang peristiwa Amazon RDS, lihat Menggunakan pemberitahuan peristiwa Amazon RDS.
- Jika aplikasi Anda menyimpan nilai DNS dalam cache, atur waktu untuk beroperasi (TTL) menjadi kurang dari 30 detik. Mengatur TTL adalah praktik yang baik jika terjadi failover. Dalam failover, alamat IP mungkin berubah dan nilai yang disimpan dalam cache mungkin tidak lagi digunakan.
- Sebaiknya Anda tidak mengaktifkan mode berikut karena mode ini menonaktifkan pencatatan transaksi dalam log, yang diperlukan untuk Multi-AZ:
  - Mode pemulihan sederhana
  - Mode offline
  - Mode hanya baca
- Lakukan pengujian untuk menentukan durasi yang dibutuhkan bagi instans DB Anda untuk failover. Waktu failover dapat bervariasi berdasarkan jenis basis data, kelas instans, dan jenis penyimpanan yang Anda gunakan. Anda juga harus menguji kemampuan aplikasi Anda untuk terus bekerja jika terjadi failover.

- Untuk mempersingkat waktu failover, lakukan hal berikut:
  - Pastikan Anda memiliki cukup IOPS yang Tersedia, yang dialokasikan untuk beban kerja Anda. I/O yang tidak memadai dapat memperpanjang waktu failover. Pemulihan basis data memerlukan I/O.
  - Gunakan transaksi yang lebih kecil. Pemulihan basis data bergantung pada transaksi, jadi jika Anda dapat memecah transaksi besar menjadi beberapa transaksi yang lebih kecil, waktu failover Anda akan lebih singkat.
- Pertimbangkan kenaikan latensi selama failover. Sebagai bagian dari proses failover, Amazon RDS mereplikasi data Anda ke instans siaga baru secara otomatis. Replikasi ini berarti bahwa data baru dimasukkan ke dua instans DB yang berbeda. Jadi, mungkin ada beberapa latensi hingga instans DB siaga berhasil mengimbangi instans DB primer yang baru.
- Deploy aplikasi Anda di semua Zona Ketersediaan. Jika Zona Ketersediaan tidak aktif, aplikasi Anda di Zona Ketersediaan lain akan tetap tersedia.

Ketika menggunakan deployment Multi-AZ dari SQL Server, perlu diingat bahwa Amazon RDS membuat replika untuk semua basis data SQL Server di instans Anda. Jika Anda tidak ingin basis data tertentu memiliki replika sekunder, atur instans DB terpisah yang tidak menggunakan Multi-AZ untuk basis data tersebut.

Video praktik terbaik Amazon RDS for SQL Server

Konferensi AWS re:Invent 2019 menyertakan presentasi tentang fitur-fitur baru dan praktik terbaik untuk bekerja dengan SQL Server di Amazon RDS. Video presentasi tersedia di sini.

Menggunakan grup parameter DB

Sebaiknya Anda mencoba perubahan grup parameter DB pada instans DB pengujian sebelum menerapkan perubahan grup parameter pada instans DB produksi Anda. Pengaturan parameter mesin DB yang tidak tepat dalam grup parameter DB dapat memiliki efek merugikan yang tidak diinginkan, termasuk penurunan performa dan ketidakstabilan sistem. Selalu berhati-hati saat memodifikasi parameter mesin DB dan mencadangkan instans DB sebelum memodifikasi grup parameter DB.

Untuk informasi tentang mencadangkan instans DB, lihat Mencadangkan, memulihkan, dan mengekspor data.

Praktik terbaik untuk mengotomatiskan pembuatan instans DB

Ini adalah praktik terbaik Amazon RDS untuk membuat instans DB dengan versi minor pilihan mesin basis data. Anda dapat menggunakan AWS CLI, Amazon RDS API, atau AWS CloudFormation untuk mengotomatiskan pembuatan instans DB. Jika menggunakan metode ini, Anda hanya dapat menentukan versi utama dan Amazon RDS otomatis membuat instans dengan versi minor pilihan. Misalnya, jika PostgreSQL 12.5 adalah versi minor pilihan, dan jika Anda menentukan versi 12 dengan `create-db-instance`, instans DB akan menjadi versi 12.5.

Untuk menentukan versi minor pilihan, Anda dapat menjalankan perintah `describe-db-engine-versions` dengan opsi `--default-only` seperti yang ditunjukkan dalam contoh berikut.

```
aws rds describe-db-engine-versions --default-only --engine postgres

{
    "DBEngineVersions": [
        {
            "Engine": "postgres",
            "EngineVersion": "12.5",
            "DBParameterGroupFamily": "postgres12",
            "DBEngineDescription": "PostgreSQL",
            "DBEngineVersionDescription": "PostgreSQL 12.5-R1",
            ...some output truncated...
        }
    ]
}
```

Untuk informasi tentang pembuatan instans DB secara pemrograman, lihat sumber daya berikut:

- Menggunakan AWS CLI - create-db-instance
- Menggunakan API Amazon RDS – CreateDBInstance
- Menggunakan AWS CloudFormation — AWS: :RDS: :DBInstance

Video presentasi fitur baru dan praktik terbaik Amazon RDS

Konferensi re AWS : Invent 2019 menyertakan presentasi tentang fitur Amazon RDS baru dan praktik terbaik untuk memantau, menganalisis, dan menyetel kinerja basis data menggunakan RDS. Video presentasi tersedia di sini.

Mengonfigurasi instans DB Amazon RDS

Bagian ini menunjukkan cara menyiapkan instans DB Amazon RDS. Sebelum membuat instans DB, tentukan kelas instans DB yang akan menjalankan instans DB. Juga, tentukan di mana instans DB akan berjalan dengan memilih AWS Wilayah. Selanjutnya, buat instans DB.

Anda dapat mengonfigurasi instans DB dengan grup opsi dan grup parameter DB.

- Grup opsi menentukan fitur, opsi yang dipanggil, yang tersedia untuk instans DB Amazon RDS tertentu.
- Grup parameter DB bertindak sebagai kontainer untuk nilai konfigurasi mesin yang diterapkan pada satu atau beberapa instans DB.

Opsi dan parameter yang tersedia tergantung pada mesin DB dan versi mesin DB. Anda dapat menentukan grup opsi dan grup parameter DB saat membuat instans DB. Anda juga dapat mengubah instans DB untuk menentukannya.

Topik

Membuat instans DB Amazon RDS

Blok bangunan dasar Amazon RDS adalah instans DB, tempat Anda membuat basis data. Anda memilih karakteristik spesifik mesin instans DB saat Anda membuatnya. Anda juga memilih kapasitas penyimpanan, CPU, memori, dan sebagainya dari AWS instance di mana server database berjalan.

Topik

Prasyarat instans DB

> **Important**
>
> Sebelum Anda dapat membuat instans DB Amazon RDS, selesaikan tugas-tugas dalam Menyiapkan Amazon RDS.

Berikut ini adalah prasyarat untuk membuat instans DB RDS.

Topik

Konfigurasikan jaringan untuk instans DB

Anda dapat membuat instans DB Amazon RDS hanya di cloud privat virtual (VPC) berdasarkan layanan Amazon VPC. Juga, itu harus dalam Wilayah AWS yang memiliki setidaknya dua Availability Zone. Grup subnet DB yang Anda pilih untuk instans DB harus mencakup setidaknya dua Zona Ketersediaan. Konfigurasi ini memastikan Anda dapat mengonfigurasi deployment Multi-AZ saat membuat instans DB atau dengan mudah beralih ke instans DB di masa mendatang.

Untuk mengatur konektivitas antara instans DB baru Anda dan instans Amazon EC2 di VPC yang sama, lakukan saat Anda membuat instans DB. Untuk terhubung ke instans DB Anda dari sumber daya selain instans EC2 di VPC yang sama, konfigurasikan koneksi jaringan secara manual.

Topik

Konfigurasikan konektivitas jaringan otomatis dengan instans EC2

Ketika Anda membuat instans RDS DB, Anda dapat menggunakan AWS Management Console untuk mengatur konektivitas antara instans EC2 dan instans DB baru. Ketika Anda melakukannya, RDS mengonfigurasi VPC dan pengaturan jaringan secara otomatis. Instans DB dibuat di VPC yang sama dengan instans EC2 sehingga instans EC2 dapat mengakses instans DB.

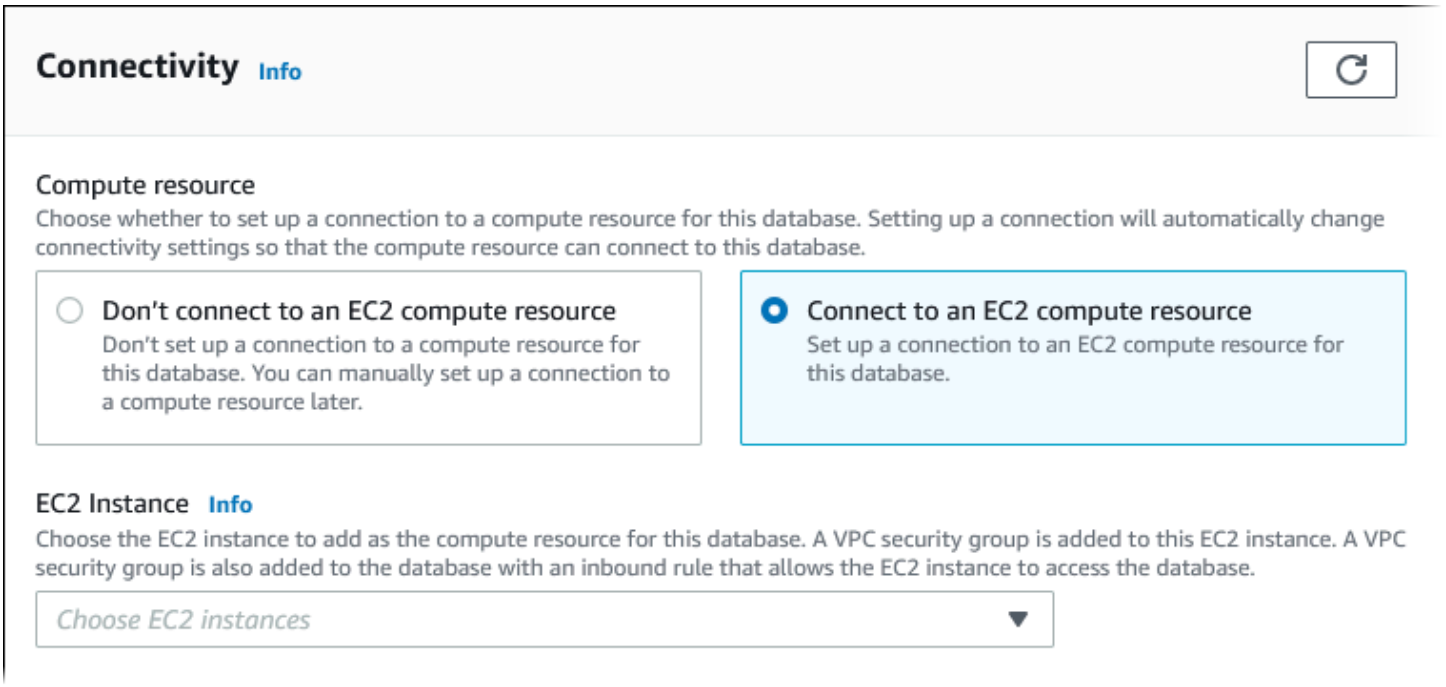
Berikut ini adalah persyaratan untuk menghubungkan instans EC2 dengan instans DB:

- Instans EC2 harus ada Wilayah AWS sebelum Anda membuat instans DB.

  Jika tidak ada instans EC2 di Wilayah AWS, konsol menyediakan tautan untuk membuatnya.
- Pengguna yang membuat instans DB harus memiliki izin untuk melakukan operasi berikut ini:
  - `ec2:AssociateRouteTable`
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:AuthorizeSecurityGroupIngress`
  - `ec2:CreateRouteTable`
  - `ec2:CreateSubnet`
  - `ec2:CreateSecurityGroup`
  - `ec2:DescribeInstances`
  - `ec2:DescribeNetworkInterfaces`
  - `ec2:DescribeRouteTables`
  - `ec2:DescribeSecurityGroups`
  - `ec2:DescribeSubnets`
  - `ec2:ModifyNetworkInterfaceAttribute`
  - `ec2:RevokeSecurityGroupEgress`

Menggunakan opsi ini membuat instans DB privat. Instans DB menggunakan grup subnet DB dengan hanya subnet privat untuk membatasi akses ke sumber daya dalam VPC.

Untuk menghubungkan instans EC2 ke instans DB, pilih Hubungkan ke sumber daya komputasi EC2 di bagian Konektivitas pada halaman Buat basis data.



Saat Anda memilih Hubungkan ke sumber daya komputasi EC2, RDS menetapkan opsi berikut secara otomatis. Anda tidak dapat mengubah pengaturan ini kecuali Anda memilih untuk tidak mengatur konektivitas dengan instans EC2 dengan memilih Jangan hubungkan ke sumber daya komputasi EC2.

Opsi konsol	Pengaturan otomatis
Jenis jaringan	RDS mengatur jenis jaringan ke IPv4. Saat ini, mode tumpukan ganda tidak didukung saat Anda mengatur koneksi antara instans EC2 dan instans DB.
Cloud Privat Virtual (VPC)	RDS mengatur VPC ke yang terkait dengan instans EC2.
Grup subnet DB	RDS membutuhkan grup subnet DB dengan subnet privat di Zona Ketersediaan yang sama dengan instans EC2. Jika grup subnet DB yang memenuhi persyaratan ini ada, RDS menggunakan grup subnet DB yang ada. Secara default, opsi ini diatur ke Pengaturan otomatis.

Opsi konsol	Pengaturan otomatis
	Ketika Anda memilih Pengaturan otomatis dan tidak ada grup subnet DB yang memenuhi persyaratan ini, tindakan berikut terjadi. RDS menggunakan tiga subnet privat yang tersedia di tiga Zona Ketersediaan dengan salah satu Zona Ketersediaan sama dengan instans EC2. Jika subnet privat tidak tersedia di Zona Ketersediaan, RDS membuat subnet privat di Zona Ketersediaan. Kemudian RDS membuat grup subnet DB. Ketika subnet privat tersedia, RDS menggunakan tabel rute yang terkait dengan subnet dan menambahkan subnet apa pun yang dibuatnya ke tabel rute ini. Ketika tidak ada subnet privat yang tersedia, RDS membuat tabel rute tanpa akses gateway internet dan menambahkan subnet yang dibuatnya ke tabel rute. RDS juga memungkinkan Anda untuk menggunakan grup subnet DB yang ada. Pilih Pilih yang ada jika Anda ingin menggunakan grup subnet DB pilihan Anda yang sudah ada.
Akses publik	RDS memilih Tidak sehingga instans DB tidak dapat diakses publik. Untuk keamanan, ini adalah praktik terbaik untuk menjaga basis data tetap privat dan memastikannya tidak dapat diakses dari internet.

Opsi konsol	Pengaturan otomatis
Grup keamanan VPC (firewall)	RDS membuat grup keamanan baru yang terkait dengan instans DB. Grup keamanan diberi nama `rds-ec2-`*n*, di mana *n* merupakan nomor. Grup keamanan ini disertai aturan masuk dengan grup keamanan VPC EC2 (firewall) sebagai sumbernya . Grup keamanan ini yang terkait dengan instans DB memungkinkan instans EC2 untuk mengakses instans DB. RDS juga membuat grup keamanan baru yang terkait dengan instans EC2. Grup keamanan diberi nama `ec2-rds-`*n*, di mana *n* merupakan nomor. Grup keamanan ini disertai aturan keluar dengan grup keamanan VPC dari instans DB sebagai sumbernya. Grup keamanan ini memungkinkan instans EC2 untuk mengirim lalu lintas ke instans DB. Anda dapat menambahkan grup keamanan baru lainnya dengan memilih Buat baru dan mengetik nama grup keamanan baru. Anda dapat menambahkan grup keamanan yang ada dengan memilih Pilih yang ada dan memilih grup keamanan untuk ditambahkan.
Zona Ketersediaan	Saat Anda memilih instans DB tunggal dalam Ketersediaan & durabilitas (deployment Satu AZ), RDS memilih Zona Ketersediaan instans EC2. Saat Anda memilih Instans DB Multi-AZ dalam Ketersediaan & durabilitas (deployment instans DB Multi-AZ), RDS memilih Zona Ketersediaan instans EC2 untuk satu instans DB dalam deployment. RDS secara acak memilih Zona Ketersediaan yang berbeda untuk instans DB lainnya. Instans DB utama atau replika siaga dibuat di Zona Ketersediaan yang sama dengan instans EC2. Saat Anda memilih Instans DB Multi-AZ, ada kemungkinan biaya lintas Zona Ketersediaan jika instans DB dan instans EC2 berada di Zona Ketersediaan yang berbeda.

Untuk informasi selengkapnya tentang pengaturan ini, lihat Pengaturan untuk instans DB.

Jika Anda mengubah pengaturan ini setelah instans DB dibuat, perubahan dapat memengaruhi koneksi antara instans EC2 dan instans DB.

Konfigurasikan jaringan secara manual

Untuk terhubung ke instans DB Anda dari sumber daya selain instans EC2 di VPC yang sama, konfigurasikan koneksi jaringan secara manual. Jika Anda menggunakan AWS Management Console untuk membuat instans DB Anda, Anda dapat meminta Amazon RDS secara otomatis membuat VPC untuk Anda. Atau Anda dapat menggunakan VPC yang ada atau membuat VPC baru untuk instans DB Anda. Dengan pendekatan apa pun, VPC Anda membutuhkan setidaknya satu subnet di masing-masing dari setidaknya dua Zona Ketersediaan untuk digunakan dengan instans DB RDS.

Secara default, Amazon RDS membuat instans DB sebagai Zona Ketersediaan secara otomatis untuk Anda. Untuk memilih Zona Ketersediaan tertentu, Anda perlu mengubah pengaturan Ketersediaan & durabilitas ke Instans DB tunggal. Melakukan hal ini akan memperlihatkan pengaturan Zona Ketersediaan yang memungkinkan Anda memilih dari antara Zona Ketersediaan di VPC Anda. Namun, jika Anda memilih deployment Multi-AZ, RDS memilih Zona Ketersediaan instans DB utama atau penulis secara otomatis, dan pengaturan Zona Ketersediaan tidak akan ditampilkan.

Dalam beberapa kasus, Anda mungkin tidak memiliki VPC default atau belum membuat VPC. Jika demikian, Anda dapat meminta Amazon RDS untuk membuat VPC secara otomatis ketika Anda membuat instans DB menggunakan konsol. Jika tidak, lakukan tindakan berikut:

- Buat VPC dengan setidaknya satu subnet di masing-masing setidaknya dua Availability Zones di Wilayah AWS mana Anda ingin menyebarkan instans DB Anda. Lihat informasi yang lebih lengkap di Bekerja dengan klaster DB dalam VPC dan Tutorial: Membuat VPC untuk digunakan dengan instans DB (khusus IPv4).
- Tentukan grup keamanan VPC yang mengizinkan koneksi ke instans DB Anda. Lihat informasi yang lebih lengkap di Memberikan akses ke instans DB di VPC Anda dengan membuat grup keamanan dan Mengontrol akses dengan grup keamanan.
- Tentukan grup subnet DB RDS yang menentukan setidaknya dua subnet di VPC yang dapat digunakan oleh instans DB tersebut. Untuk informasi selengkapnya, lihat Bekerja dengan grup subnet DB.

Jika Anda ingin terhubung ke sumber daya yang tidak berada di VPC yang sama dengan instans DB, lihat skenario yang sesuai di Skenario untuk mengakses instans DB di VPC.

Prasyarat tambahan

Sebelum Anda membuat instans DB, pertimbangkan prasyarat tambahan berikut ini:

- Jika Anda terhubung untuk AWS menggunakan kredensyal AWS Identity and Access Management (IAM), AWS akun Anda harus memiliki kebijakan IAM tertentu. Ini memberikan izin yang diperlukan untuk melakukan operasi Amazon RDS. Untuk informasi selengkapnya, lihat Manajemen identitas dan akses untuk Amazon RDS.

  Untuk menggunakan IAM untuk mengakses konsol RDS, masuk ke AWS Management Console dengan kredensyal pengguna IAM Anda. Kemudian buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
- Untuk menyesuaikan parameter konfigurasi instans DB Anda, tentukan grup parameter DB dengan pengaturan parameter yang diperlukan. Untuk informasi tentang membuat atau memodifikasi grup parameter DB, lihat Bekerja dengan grup parameter.

  > **Important**
  >
  > Jika Anda menggunakan model BYOL untuk RDS untuk Db2, sebelum membuat instans DB, Anda harus terlebih dahulu membuat grup parameter kustom yang berisi IBM Site ID dan IBM Customer ID. Untuk informasi selengkapnya, lihat Bawa Lisensi Sendiri.
- Tentukan nomor port TCP/IP untuk menentukan instans DB Anda. Firewall di beberapa perusahaan memblokir koneksi ke port default untuk instans DB RDS. Jika firewall perusahaan Anda memblokir port default, pilih port lain untuk instans DB Anda. Port default untuk mesin DB Amazon RDS adalah:

RDS for Db2	RDS for MariaDB	RDS for MySQL	RDS for Oracle	RDS for PostgreSQL	RDS for SQL Server
50000	3306	3306	1521	5432	1433

  Untuk RDS for SQL Server, port berikut disimpan, dan Anda tidak dapat menggunakannya saat membuat instans DB: `1234`, `1434`, `3260`, `3343`, `3389`, `47001`, dan `49152-49156`.

Membuat instans DB

Anda dapat membuat instans Amazon RDS DB menggunakan AWS Management Console, API AWS CLI, atau RDS.

Konsol

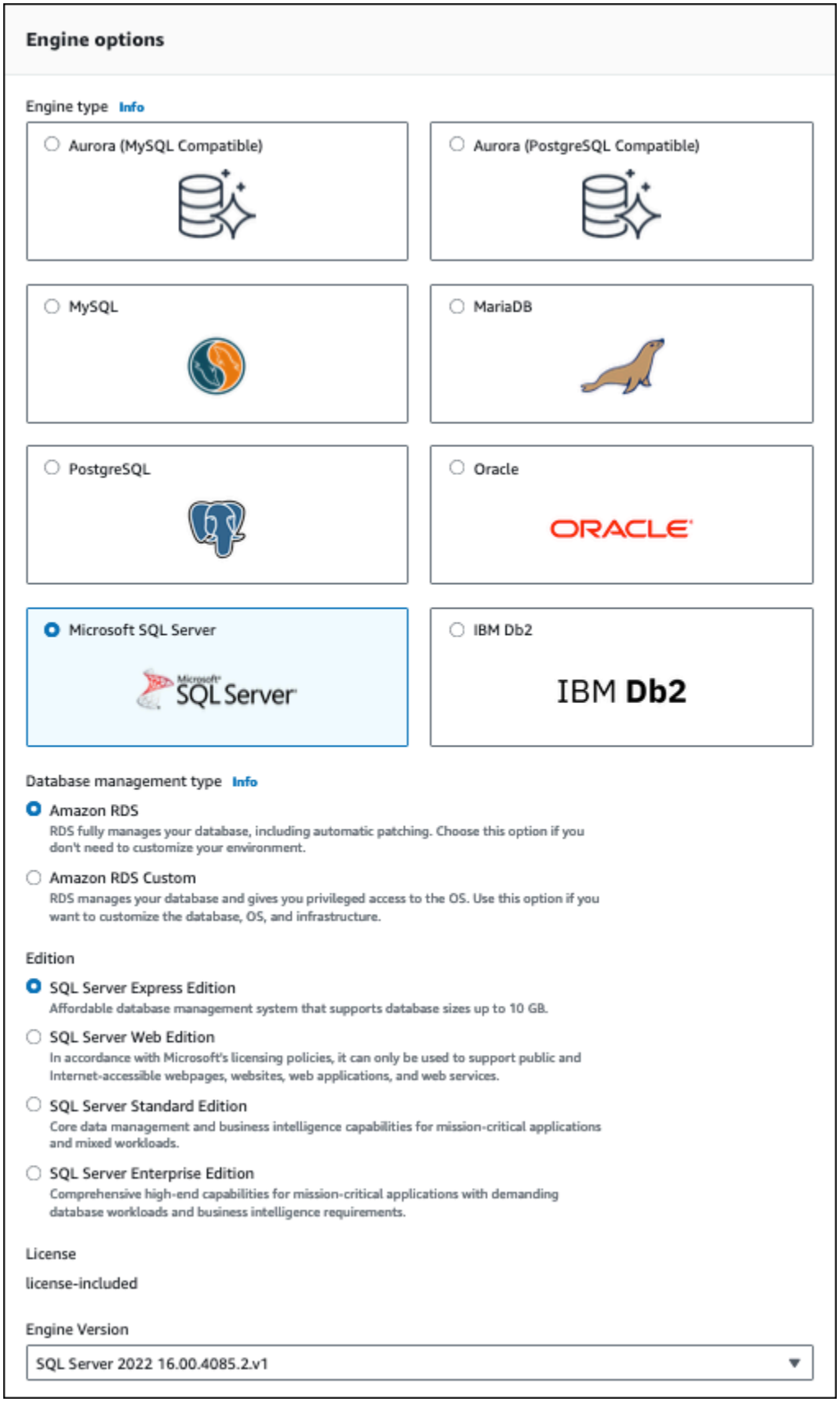
Anda dapat membuat instans DB dengan menggunakan AWS Management Console with Easy create diaktifkan atau tidak diaktifkan. Dengan Pembuatan mudah aktif, Anda hanya menentukan jenis mesin DB, ukuran instans DB, dan pengidentifikasi instans DB. Pembuatan mudah menggunakan pengaturan default untuk opsi konfigurasi lainnya. Dengan Mudah dibuat tidak diaktifkan, Anda menentukan lebih banyak opsi konfigurasi saat Anda membuat basis data, termasuk opsi untuk ketersediaan, keamanan, pencadangan, dan pemeliharaan.

> **Note**
>
> Dalam prosedur berikut, Pembuatan standar diaktifkan, dan Mudah dibuat tidak diaktifkan. Prosedur ini menggunakan Microsoft SQL Server sebagai contoh.
> Untuk contoh yang menggunakan Mudah dibuat yang akan memandu Anda dalam membuat dan menghubungkan ke contoh instans DB untuk setiap mesin, lihat Mulai menggunakan Amazon RDS.

Untuk membuat instans DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di sudut kanan atas konsol Amazon RDS, pilih Wilayah AWS tempat Anda ingin membuat instans DB.
3. Di panel navigasi, pilih Basis data.
4. Pilih Buat basis data, lalu pilih Pembuatan standar.
5. Untuk Jenis mesin, pilih IBM Db2, MariaDB, Microsoft SQL Server, MySQL, Oracle, atau PostgreSQL.

   Microsoft SQL Server ditampilkan di sini.


Engine options
Engine type Info
Aurora (MySQL Compatible)
Aurora (PostgreSQL Compatible)
MySQL
MariaDB
PostgreSQL
Oracle
ORACLE
Microsoft SQL Server
SQL Server
IBM Db2
IBM Db2
Database management type Info
Amazon RDS
RDS fully manages your database, including automatic patching. Choose this option if you don't need to customize your environment.
Amazon RDS Custom
RDS manages your database and gives you privileged access to the OS. Use this option if you want to customize the database, OS, and infrastructure.
Edition
SQL Server Express Edition
Affordable database management system that supports database sizes up to 10 GB.
SQL Server Web Edition
In accordance with Microsoft's licensing policies, it can only be used to support public and Internet-accessible webpages, websites, web applications, and web services.
SQL Server Standard Edition
Core data management and business intelligence capabilities for mission-critical applications and mixed workloads.
SQL Server Enterprise Edition
Comprehensive high-end capabilities for mission-critical applications with demanding database workloads and business intelligence requirements.
License
license-included
Engine Version
SQL Server 2022 16.00.4085.2.v1

6. Untuk Jenis manajemen basis data, jika Anda menggunakan Oracle atau SQL Server, pilih Amazon RDS atau Amazon RDS Custom.

   Amazon RDS ditampilkan di sini. Untuk informasi selengkapnya tentang RDS Custom, lihat Menggunakan Amazon RDS Custom.

7. Untuk Edisi, jika Anda menggunakan Db2, Oracle, atau SQL Server, pilih edisi mesin DB yang ingin Anda gunakan.

   MySQL hanya memiliki satu opsi untuk edisi, sementara MariaDB dan PostgreSQL tidak memiliki satu pun.

8. Untuk Versi, pilih versi mesin.

9. Di Templat, pilih templat yang sesuai dengan kasus penggunaan Anda. Jika Anda memilih Produksi, hal berikut sudah dipilih sebelumnya di langkah selanjutnya:

   - Opsi failover Multi-AZ
   - Opsi penyimpanan IOPS SSD (io1) yang tersedia
   - Opsi Aktifkan perlindungan penghapusan

   Kami menyarankan fitur ini untuk lingkungan produksi apa pun.

   > **Note**
   >
   > Pilihan templat bervariasi berdasarkan edisi.

10. Untuk memasukkan kata sandi utama, lakukan hal berikut:

    a. Di bagian Pengaturan, buka Pengaturan Kredensial.

    b. Jika Anda ingin menentukan kata sandi, kosongkan kotak centang Buat kata sandi secara otomatis jika dipilih.

    c. (Opsional) Ubah nilai Nama pengguna master.

    d. Masukkan kata sandi yang sama di Kata sandi master dan Konfirmasikan kata sandi.

11. (Opsional) Atur koneksi ke sumber daya komputasi untuk instans DB ini.

    Anda dapat mengonfigurasi konektivitas antara instans Amazon EC2 dan instans DB baru selama pembuatan instans DB. Untuk informasi selengkapnya, lihat Konfigurasikan konektivitas jaringan otomatis dengan instans EC2.

12. Di bagian Konektivitas di bawah Grup keamanan VPC (firewall), jika Anda memilih Buat baru, grup keamanan VPC dibuat dengan aturan masuk yang memungkinkan alamat IP komputer lokal Anda mengakses basis data.
13. Untuk bagian yang tersisa, tentukan pengaturan instans DB Anda. Untuk informasi tentang setiap pengaturan, lihat Pengaturan untuk instans DB.
14. Pilih Buat basis data.

    Jika Anda memilih untuk menggunakan kata sandi yang dibuat otomatis, tombol Lihat detail kredensial muncul pada halaman Basis data.

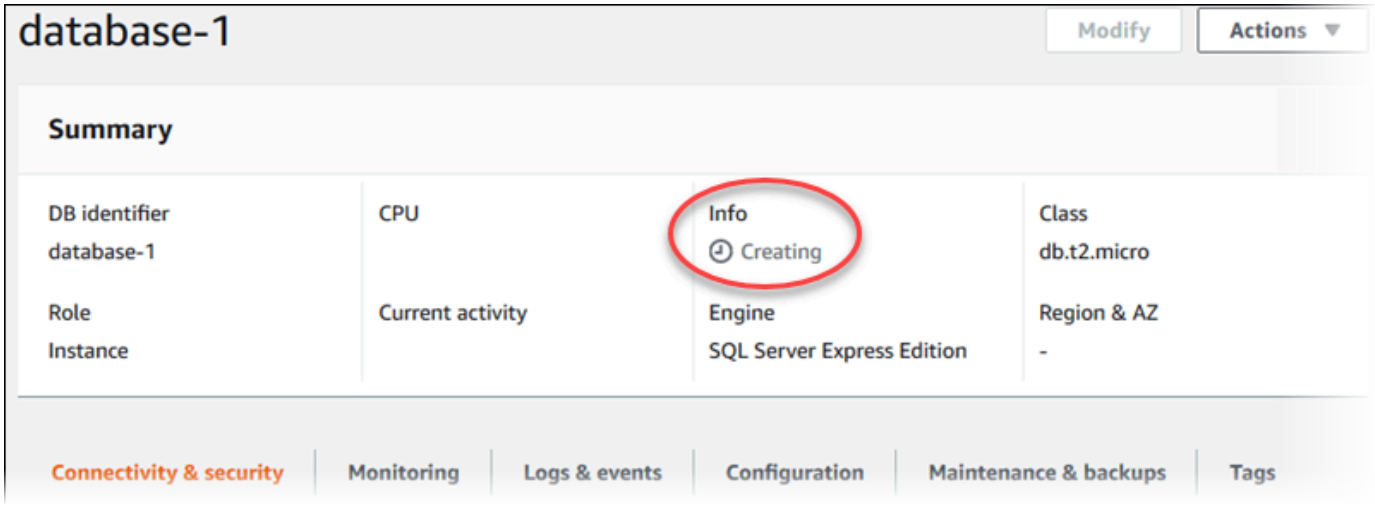
    Untuk melihat nama pengguna dan kata sandi master untuk instans DB, pilih Lihat detail kredensial.

    Untuk terhubung ke instans DB sebagai pengguna master, gunakan nama pengguna dan kata sandi yang muncul.

    > **Important**
    >
    > Anda tidak dapat melihat kata sandi pengguna master lagi. Jika Anda tidak mencatatnya, Anda mungkin harus mengubahnya. Jika Anda perlu mengubah kata sandi pengguna master setelah instans DB tersedia, ubah instans DB untuk melakukannya. Untuk informasi selengkapnya tentang mengubah instans DB, lihat Memodifikasi instans DB Amazon RDS.

15. Untuk Basis data, pilih nama instans DB baru.

    Pada konsol RDS, detail untuk instans DB baru muncul. Instans DB akan berstatus Sedang dibuat hingga proses pembuatannya selesai dan siap untuk digunakan. Saat statusnya berubah menjadi Tersedia, Anda dapat terhubung ke instans DB. Bergantung pada kelas instans DB dan penyimpanan yang dialokasikan, perlu waktu beberapa menit agar instans baru tersedia.



AWS CLI

Untuk membuat instance DB dengan menggunakan AWS CLI, panggil create-db-instanceperintah dengan parameter berikut:

- `--db-instance-identifier`
- `--db-instance-class`
- `--vpc-security-group-ids`
- `--db-subnet-group`
- `--engine`
- `--master-username`
- `--master-user-password`
- `--allocated-storage`
- `--backup-retention-period`

Untuk informasi tentang setiap pengaturan, lihat Pengaturan untuk instans DB.

Contoh ini menggunakan Microsoft SQL Server.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-instance \
    --engine sqlserver-se \
```

```
    --db-instance-identifier mymsftsqlserver \
    --allocated-storage 250 \
    --db-instance-class db.t3.large \
    --vpc-security-group-ids mysecuritygroup \
    --db-subnet-group mydbsubnetgroup \
    --master-username masterawsuser \
    --manage-master-user-password \
    --backup-retention-period 3
```

Untuk Windows:

```
aws rds create-db-instance ^
    --engine sqlserver-se ^
    --db-instance-identifier mydbinstance ^
    --allocated-storage 250 ^
    --db-instance-class db.t3.large ^
    --vpc-security-group-ids mysecuritygroup ^
    --db-subnet-group mydbsubnetgroup ^
    --master-username masterawsuser ^
    --manage-master-user-password ^
    --backup-retention-period 3
```

Perintah ini menghasilkan output yang serupa dengan yang berikut ini.

```
DBINSTANCE  mydbinstance  db.t3.large  sqlserver-se  250  sa  creating  3  ****  n
 10.50.2789
SECGROUP  default  active
PARAMGRP  default.sqlserver-se-14  in-sync
```

API RDS

Untuk membuat instans DB menggunakan Amazon RDS API, panggil operasi createDBInstance.

Untuk informasi tentang setiap pengaturan, lihat Pengaturan untuk instans DB.

Pengaturan untuk instans DB

Dalam tabel berikut, Anda dapat menemukan detail tentang pengaturan yang Anda pilih saat membuat instans DB. Tabel juga menampilkan mesin DB yang mendukung setiap pengaturan.

Anda dapat membuat instance DB menggunakan konsol, perintah create-db-instanceCLI, atau operasi CreateDBInstance RDS API.

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung

Penyimpanan yang dialokasikan	Jumlah penyimpanan yang dialokasikan untuk instans DB Anda (dalam gibibyte). Dalam beberapa kasus, mengalokasikan jumlah penyimpanan yang lebih tinggi untuk instans DB Anda daripada ukuran basis data Anda dapat meningkatkan performa I/O.
	
	Untuk informasi selengkapnya, lihat Penyimpanan instans DB Amazon RDS.
		Opsi CLI:
		
		`--allocated-storage`
		
		Parameter API:
		
		`AllocatedStorage`
			Semua

Pengaturan arsitektur	Jika Anda memilih Arsitektur multi-penghuni Oracle, RDS untuk Oracle membuat basis data kontainer (CDB). Jika Anda tidak memilih opsi ini, RDS for Oracle membuat non-CDB. Sebuah non-CDB menggunakan arsitektur basis data Oracle tradisional. Sebuah CDB dapat berisi basis data pluggable (PDB) sedangkan non-CDB tidak bisa. Oracle Database 21c hanya menggunakan arsitektur CDB. Oracle Database 19c dapat menggunakan arsitektur CDB atau non-CDB. Rilis yang lebih rendah dari Oracle Database 19c hanya menggunakan arsitektur non-CDB. Untuk informasi selengkapnya, lihat Ikhtisar CDB RDS for Oracle.	Opsi CLI: `--engine oracle-ee-cdb` (Multi-penghuni Oracle) `--engine oracle-se2-cdb` (Multi-penghuni Oracle) `--engine oracle-ee` (tradisional) `--engine oracle-se2` (tradisional) Parameter API: `Engine`	Oracle
Konfigurasi arsitektur	Pengaturan ini hanya berlaku ketika Anda memilih Arsitektur multi-penghuni Oracle untuk Pengaturan arsitektur. Pilih	Opsi CLI:	Oracle

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung
	salah satu dari pengaturan tambahan berikut: • Dengan Konfigurasi multi-penghuni, instans RDS for Oracle CDB Anda dapat berisi 1–30 basis data penghuni, bergantung pada edisi basis data dan lisensi opsi apa pun yang diperlukan. Dalam konteks basis data Oracle, basis data penghuni adalah PDB. PDB aplikasi dan PDB proksi tidak didukung. Instans DB Anda dibuat dengan 1 basis data penghuni awal. Pilih nilai untuk Nama basis data penghuni, Nama pengguna master basis data penghuni, Kata sandi master basis data penghuni, dan Set karakter basis data penghuni. Konfigurasi multi-penghuni bersifat permanen. Dengan demikian, Anda tidak dapat mengonversi konfigurasi multi-penghuni kembali ke konfigurasi penghuni tunggal. Pembaruan rilis (RU) minimum yang didukung untuk konfigurasi multi-penghuni adalah 19.0.0.0.ru-2022-01.rur-2022.r1.	`--multi-tenant` (konfigurasi multi-penghuni) `--no-multi-tenant` (konfigurasi penghuni tunggal) Parameter API: `MultiTenant`	

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung
	Note Fitur Amazon RDS disebut “multi-penghuni” dan bukannya “multi-penghuni” karena merupakan kemampuan platform RDS, bukan hanya mesin Oracle DB. Istilah “Oracle multi-penghuni” mengacu secara eksklusif ke arsitektur basis data Oracle, yang kompatibel dengan deployment RDS dan on-premise. • Dengan Konfigurasi penghuni tunggal, RDS for Oracle CDB Anda berisi 1 PDB. Ini adalah konfigurasi default saat Anda membuat CDB. Anda tidak dapat menghapus PDB awal atau menambahkan lebih banyak PDB. Anda nantinya dapat mengonversi konfigurasi penghuni tunggal CDB Anda ke konfigurasi multi-penghuni, tetapi Anda tidak dapat mengonver si kembali ke konfigurasi penghuni tunggal. Terlepas dari konfigurasi mana yang Anda pilih, CDB Anda berisi satu PDB		

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung
	awal. Dalam konfigurasi multi-penghuni, Anda dapat membuat lebih banyak PDB di lain waktu menggunakan API RDS. Untuk informasi selengkapnya, lihat Ikhtisar CDB RDS for Oracle.		

Peningkatan versi minor otomatis	Pilih Aktifkan peningkatan versi minor otomatis untuk mengaktifkan instans DB Anda agar menerima peningkatan versi mesin DB minor pilihan secara otomatis saat tersedia. Ini adalah perilaku default. Amazon RDS melakukan peningkatan versi minor otomatis selama jendela pemeliharaan. Jika Anda tidak memilih Aktifkan peningkatan versi minor otomatis, instans DB Anda tidak ditingkatkan secara otomatis saat versi minor baru tersedia.
	
	Untuk informasi selengkapnya, lihat Meng-upgrade versi mesin minor secara otomatis.
		Opsi CLI:
		
		`--auto-minor-version-upgrade`
		
		`--no-auto-minor-version-upgrade`
		
		Parameter API:
		
		`AutoMinorVersionUpgrade`
			Semua

Zona ketersediaan	Zona Ketersediaan untuk instans DB Anda. Gunakan nilai default dari Tidak ada preferensi kecuali jika Anda ingin menentukan Zona Ketersediaan.
	
	Untuk informasi selengkapnya, lihat Wilayah, Zona Ketersediaan, dan Zona Lokal.
		Opsi CLI:
		
		`--availability-zone`
		
		Parameter API:
		
		`AvailabilityZone`
			Semua

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung

AWS KMS key	Hanya tersedia jika Enkripsi diatur ke Aktifkan enkripsi. Pilih AWS KMS key untuk mengenkripsi instans DB ini. Untuk informasi selengkapnya, lihat Mengenkripsi sumber daya Amazon RDS.	Opsi CLI:
		`--kms-key-id`
		Parameter API:
		`KmsKeyId`
			Semua

Replikasi cadangan	Pilih Aktifkan replikasi di Wilayah AWS lain untuk membuat cadangan di Wilayah tambahan untuk pemulihan bencana. Lalu pilih Wilayah tujuan untuk cadangan tambahan.	Tidak tersedia saat membuat instans DB. Untuk informasi tentang mengaktifkan pencadangan Lintas wilayah menggunakan AWS CLI atau RDS API, lihat. Mengaktifkan pencadangan otomatis lintas Wilayah	Oracle PostgreSQL SQL Server

Periode retensi cadangan	Jumlah hari yang Anda inginkan untuk menyimpan cadangan otomatis instans DB Anda. Untuk setiap instans DB nontrivial, atur nilai ini ke **1** atau lebih besar.
	Untuk informasi selengkapnya, lihat Pengantar cadangan.
		Opsi CLI:
		`--backup-retention-period`
		Parameter API:
		`BackupRetentionPeriod`
			Semua

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung

Target cadangan	Pilih AWS Clouduntuk menyimpan cadangan otomatis dan snapshot manual di Wilayah induk. AWS Pilih Outposts (on-premise) untuk menyimpannya secara titik waktu di Outpost Anda.
	
	Pengaturan opsi ini hanya berlaku untuk RDS di Outpost. Untuk informasi selengkapnya, lihat Membuat instans DB untuk Amazon RDS on AWS Outposts.
		Opsi CLI:
		
		`--backup-target`
		
		Parameter API:
		
		`BackupTarget`
			MySQL, PostgreSQL, SQL Server

Jendela cadangan	Periode waktu di mana Amazon RDS secara otomatis membuat cadangan instans DB Anda. Kecuali jika Anda memiliki waktu tertentu di mana Anda ingin basis data Anda dicadangkan, gunakan default Tidak ada preferensi.
	
	Untuk informasi selengkapnya, lihat Pengantar cadangan.
		Opsi CLI:
		
		`--preferred-backup-window`
		
		Parameter API:
		
		`PreferredBackupWindow`
			Semua

Otoritas sertifikat	Otoritas sertifikat (CA) untuk sertifikat server yang digunakan oleh instans DB.
	
	Untuk informasi selengkapnya, lihat .
		Opsi CLI:
		
		`--ca-certificate-identifier`
		
		Parameter API RDS:
		
		`CACertificateIdentifier`
			Semua

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung

Set karakter	Set karakter untuk instans DB Anda. Nilai default AL32UTF8 untuk set karakter DB adalah untuk set karakter Universal Unicode 5.0 UTF-8. Anda tidak dapat mengubah set karakter DB setelah Anda membuat instans DB.
	
	Dalam konfigurasi penghuni tunggal, set karakter DB non-default hanya memengaruhi PDB, bukan CDB. Untuk informasi selengkapnya, lihat Konfigurasi satu penghuni pada arsitektur CDB.
	
	Set karakter DB berbeda dari set karakter nasional, yang disebut set karakter NCHAR. Tidak seperti set karakter DB, set karakter NCHAR menentukan pengodean untuk kolom tipe data NCHAR (NCHAR, NVARCHAR2, dan NCLOB) tanpa memengaruhi metadata basis data.
	
	Untuk informasi selengkapnya, lihat Set karakter RDS for Oracle.
		Opsi CLI:
		
		`--character-set-name`
		
		Parameter API:
		
		`CharacterSetName`
			Oracle

Kolasi	Kolasi tingkat server untuk instans DB Anda.
	
	Untuk informasi selengkapnya, lihat Kolasi tingkat server untuk Microsoft SQL Server.
		Opsi CLI:
		
		`--character-set-name`
		
		Parameter API:
		
		`CharacterSetName`
			SQL Server

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung

Salin tanda ke snapshot	Opsi ini menyalin tag instans DB apa pun ke snapshot DB saat Anda membuat snapshot.
	
	Untuk informasi selengkapnya, lihat Memberi tag pada sumber daya Amazon RDS.
		Opsi CLI:
		
		`--copy-tags-to-snapshot`
		
		`--no-copy-tags-to-snapshot`
		
		Parameter API RDS:
		
		`CopyTagsToSnapshot`
			Semua

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung

Autentikasi basis data	Opsi autentikasi basis data yang ingin Anda gunakan.
	
	Pilih Autentikasi kata sandi untuk mengautentikasi pengguna basis data dengan kata sandi basis data saja.
	
	Pilih Autentikasi kata sandi dan DB IAM untuk mengautentikasi pengguna basis data dengan kata sandi basis data dan kredensial pengguna melalui pengguna dan peran. Untuk informasi selengkapnya, lihat Autentikasi basis data IAM untuk MariaDB, MySQL, dan PostgreSQL. Opsi ini hanya didukung untuk MySQL dan PostgreSQL.
	
	Pilih Otentikasi Kata Sandi dan Kerberos untuk mengautentikasi pengguna database dengan kata sandi database dan otentikasi Kerberos melalui yang dibuat dengan. AWS Managed Microsoft AD AWS Directory Service Selanjutnya, pilih direktori atau pilih Buat direktori baru.
	
	Untuk informasi selengkapnya, lihat salah satu dari berikut ini:
	
	• Menggunakan autentikasi Kerberos untuk RDS for Db2
	• Menggunakan autentikasi Kerberos untuk MySQL
		IAM:
		
		Opsi CLI:
		
		`--enable-iam-database-authentication`
		
		`--no-enable-iam-database-authentication`
		
		Parameter API RDS:
		
		`EnableIAMDatabaseAuthentication`
		
		Kerberos:
		
		Opsi CLI:
		
		`--domain`
		
		`--domain-iam-role-name`
		
		Parameter API RDS:
		
		`Domain`
		
		`DomainIAMRoleName`
			Bervariasi bergantung pada tipe autentikasi

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung
	• Mengonfigurasi autentikasi Kerberos untuk Amazon RDS for Oracle • Menggunakan autentikasi Kerberos dengan Amazon RDS for PostgreSQL		
Jenis manajemen basis data	Pilih Amazon RDS jika Anda tidak perlu menyesuaikan lingkungan Anda. Pilih Amazon RDS Custom jika Anda ingin menyesuaikan basis data, OS, dan infrastruktur. Untuk informasi selengkapnya, lihat Menggunakan Amazon RDS Custom.	Untuk CLI dan API, Anda menentukan jenis mesin basis data.	Oracle SQL Server

Port basis data	Port yang ingin Anda gunakan untuk mengakses instans DB. Port default akan ditampilkan.
	Note
	Firewall di beberapa perusahaan memblokir koneksi ke port MariaDB, MySQL, dan PostgreSQL default. Jika firewall perusahaan Anda memblokir port default, masukkan port lain untuk instans DB Anda.
		Opsi CLI:
		`--port`
		Parameter API RDS:
		`Port`
			Semua

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung

Versi mesin DB	Versi mesin basis data yang ingin Anda gunakan.	Opsi CLI:
		`--engine-version`
		Parameter API RDS:
		`EngineVersion`
			Semua

Kelas instans DB	Konfigurasi untuk instans DB Anda. Misalnya, kelas instans DB db.t3.small memiliki memori 2 GiB, 2 vCPU, 1 inti virtual, sebuah ECU variabel, dan kapasitas I/O sedang.
	Jika memungkinkan, pilih kelas instans DB yang cukup besar sehingga set kerja kueri yang umum dapat disimpan di memori. Ketika set kerja disimpan di memori, sistem dapat menghindari menulis pada disk, yang akan meningkatkan performa. Untuk informasi selengkapnya, lihat Kelas instans DB .
	Di RDS for Oracle, Anda dapat memilih Sertakan konfigurasi memori tambahan. Konfigurasi ini dioptimalkan untuk rasio tinggi memori berbanding vCPU. Misalnya, db.r5.6xlarge.tpc2.mem4x adalah instans DB db.r5.8x yang memiliki 2 utas per inti (tpc2) dan 4x memori instans DB db.r5.6xlarge standar. Untuk informasi selengkapnya, lihat Kelas instans RDS for Oracle.
		Opsi CLI:
		`--db-instance-class`
		Parameter API RDS:
		`DBInstanceClass`
			Semua

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung

Pengidentifikasi instans DB	Nama untuk instans DB Anda. Beri nama instans DB Anda dengan cara yang sama seperti cara Anda menamai server on-premise Anda. Pengidentifikasi instans DB Anda dapat berisi hingga 63 karakter alfanumerik, dan harus unik untuk akun Anda di Wilayah yang Anda pilih. AWS	Opsi CLI:
		`--db-instance-identifier`
		Parameter API RDS:
		`DBInstanceIdentifier`
			Semua

Grup parameter DB	Grup parameter untuk instans DB Anda. Anda dapat memilih grup parameter default, atau Anda dapat membuat grup parameter kustom.
	Jika Anda menggunakan model BYOL untuk RDS untuk Db2, sebelum membuat instans DB, Anda harus terlebih dahulu membuat grup parameter kustom yang berisi IBM Site ID dan IBM Customer ID. Untuk informasi selengkapnya, lihat Bawa Lisensi Sendiri.
	Untuk informasi selengkapnya, lihat Bekerja dengan grup parameter.
		Opsi CLI:
		`--db-parameter-group-name`
		Parameter API RDS:
		`DBParameterGroupName`
			Semua

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung

Grup subnet DB	Grup subnet DB yang Anda ingin gunakan untuk klaster DB.
	Pilih Pilih yang ada untuk menggunakan grup subnet DB yang ada. Kemudian pilih grup subnet yang diperlukan dari daftar dropdown Grup subnet DB yang ada.
	
	Pilih Pengaturan otomatis untuk membiarkan RDS memilih grup subnet DB yang kompatibel. Jika tidak ada, RDS membuat grup subnet baru untuk klaster Anda.
	
	Untuk informasi selengkapnya, lihat Bekerja dengan grup subnet DB.
		Opsi CLI:
		
		`--db-subnet-group-name`
		
		Parameter API RDS:
		
		`DBSubnetGroupName`
			Semua

Volume Log Khusus	Gunakan volume log khusus (DLV) untuk menyimpan log transaksi basis data pada volume penyimpanan yang terpisah dari volume yang berisi tabel basis data.
	
	Untuk informasi selengkapnya, lihat Menggunakan volume log khusus (DLV).
		Opsi CLI:
		
		`--dedicated-log-volume`
		
		Parameter API RDS:
		
		`DedicatedLogVolume`
			Semua

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung

Perlindungan penghapusan	Aktifkan perlindungan penghapusan agar instans DB Anda tidak terhapus. Jika Anda membuat instans DB produksi dengan perlindungan penghapusan AWS Management Console, diaktifkan secara default.
	
	Untuk informasi selengkapnya, lihat Menghapus instans DB.
		Opsi CLI:
		
		`--deletion-protection`
		
		`--no-deletion-protection`
		
		Parameter API RDS:
		
		`DeletionProtection`
			Semua

Enkripsi	Aktifkan Enkripsi untuk mengaktifkan enkripsi saat diam untuk instans DB ini.
	
	Untuk informasi selengkapnya, lihat Mengenkripsi sumber daya Amazon RDS.
		Opsi CLI:
		
		`--storage-encrypted`
		
		`--no-storage-encrypted`
		
		Parameter API RDS:
		
		`StorageEncrypted`
			Semua

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung

Peningkatan Pemantauan	Aktifkan Pemantauan yang ditingkatkan untuk memungkinkan pengumpulan metrik secara waktu nyata untuk sistem operasi yang dijalankan oleh instans DB Anda.
	
	Untuk informasi selengkapnya, lihat Memantau metrik OS dengan Pemantauan yang Disempurnakan.
		Opsi CLI:
		
		`--monitoring-interval`
		
		`--monitoring-role-arn`
		
		Parameter API RDS:
		
		`MonitoringInterval`
		
		`MonitoringRoleArn`
			Semua

Jenis mesin	Pilih mesin basis data yang akan digunakan untuk instans DB ini.	Opsi CLI:
		
		`--engine`
		
		Parameter API RDS:
		
		`Engine`
			Semua

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung

Nama basis data awal	Nama untuk basis data di instans DB Anda. Jika Anda tidak memberikan nama, Amazon RDS tidak membuat basis data di instans DB (kecuali untuk Oracle dan PostgreSQL). Nama tidak boleh berupa kata yang disimpan oleh mesin basis data, dan memiliki batasan lain, bergantung pada mesin DB.
	
	Db2:
	
	• Harus berisi antara 1–8 karakter alfanumerik.
	• Itu harus dimulai dengan a-z, A-Z, @, $, atau #, dan diikuti oleh a-z, A-Z, 0-9, _, @, #, atau $.
	• Nama tidak boleh berisi spasi.
	• Untuk informasi selengkapnya, lihat Pertimbangan tambahan.
	
	MariaDB dan MySQL:
	
	• Harus berisi antara 1–64 karakter alfanumerik.
	
	Oracle:
	
	•
		Opsi CLI:
		
		`--db-name`
		
		Parameter API RDS:
		
		`DBName`
			Semua kecuali SQL Server

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung
	Harus berisi antara 1–8 karakter alfanumerik. • Nama tidak boleh `NULL`. Nilai default-nya adalah `ORCL`. • Harus dimulai dengan huruf. PostgreSQL: • Harus berisi antara 1–63 karakter alfanumerik. • Harus dimulai dengan sebuah huruf atau garis bawah. Karakter selanjutnya dapat berupa huruf, garis bawah, atau digit (0-9). • Nama basis data awal adalah `postgres`.		

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung

Lisensi	Nilai yang valid untuk model lisensi:
	• bring-your-own-licenseuntuk Db2.
	• general-public-licenseuntuk MariaDB.
	• license-included untuk Microsoft SQL Server.
	• general-public-licenseuntuk MySQL.
	• termasuk lisensi atau bring-your-own-licenseuntuk Oracle.
	• postgresql-license untuk PostgreSQL.
		Opsi CLI:
		`--license-model`
		Parameter API RDS:
		`LicenseModel`
			Semua

Log ekspor	Jenis file log database untuk dipublikasikan ke Amazon CloudWatch Logs.
	Untuk informasi selengkapnya, lihat Menerbitkan log basis data ke Log Amazon CloudWatch.
		Opsi CLI:
		`--enable-cloudwatch-logs-exports`
		Parameter API RDS:
		`EnableCloudwatchLogsExports`
			Semua

Jendela pemeliharaan	Jendela waktu 30 menit di mana modifikasi yang tertunda untuk instans DB Anda diterapkan. Jika jangka waktu tidak penting, pilih Tidak Ada Preferensi.
	Untuk informasi selengkapnya, lihat Periode pemeliharaan Amazon RDS.
		Opsi CLI:
		`--preferred-maintenance-window`
		Parameter API RDS:
		`PreferredMaintenanceWindow`
			Semua

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung

Kelola kredensyal master di AWS Secrets Manager	Pilih Kelola kredensial master di AWS Secrets Manager untuk mengelola kata sandi pengguna master dalam rahasia di Secrets Manager.
	
	Anda juga dapat memilih kunci KMS yang akan digunakan untuk melindungi rahasia. Pilih dari kunci KMS di akun Anda, atau masukkan kunci dari akun yang berbeda.
	
	Untuk informasi selengkapnya, lihat Manajemen kata sandi dengan Amazon RDS Aurora dan AWS Secrets Manager.
		Opsi CLI:
		
		`--manage-master-user-password | --no-manage-master-user-password`
		
		`--master-user-secret-kms-key-id`
		
		Parameter API RDS:
		
		`ManageMasterUserPassword`
		
		`MasterUserSecretKmsKeyId`
			Semua

Kata sandi master	Kata sandi untuk akun pengguna master Anda. Kata sandi memiliki jumlah karakter ASCII yang dapat dicetak berikut ini (tidak termasuk `/`, `"`, spasi, dan `@`), bergantung pada mesin DB:
	
	• Db2: 8–255
	• Oracle: 8–30
	• MariaDB dan MySQL: 8–41
	• SQL Server dan PostgreSQL: 8–128
		Opsi CLI:
		
		`--master-user-password`
		
		Parameter API RDS:
		
		`MasterUserPassword`
			Semua

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung

Nama pengguna master	Nama yang Anda gunakan sebagai nama pengguna master untuk masuk ke instans DB Anda dengan semua hak istimewa basis data. Perhatikan batasan penamaan berikut:
	• Nama dapat berisi 1–16 karakter alfanumerik dan garis bawah.
	• Karakter pertama harus berupa huruf.
	• Nama tidak dapat berupa kata yang disimpan oleh mesin basis data.
	
	Anda tidak dapat mengubah nama pengguna master setelah Anda membuat instans DB.
	
	Untuk Db2, kami menyarankan Anda menggunakan nama pengguna master yang sama dengan nama instans Db2 yang dikelola sendiri.
	
	Untuk informasi selengkapnya tentang hak istimewa yang diberikan kepada pengguna master, lihat Hak akses akun pengguna master.
		Opsi CLI:
		`--master-username`
		Parameter API RDS:
		`MasterUsername`
			Semua

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung

Autentikasi Windows Microsoft SQL Server	Aktifkan autentikasi Windows Microsoft SQL Server, lalu Telusuri Direktori untuk memilih direktori tempat Anda ingin mengizinkan pengguna domain resmi mengautentikasi dengan instans SQL Server ini menggunakan Autentikasi Windows.	Opsi CLI:
		`--domain`
		`--domain-iam-role-name`
		Parameter API RDS:
		`Domain`
		`DomainIAMRoleName`
			SQL Server

Deployment Multi-AZ	Buat instans siaga untuk membuat replika sekunder pasif instans DB Anda di Zona Ketersediaan lainnya untuk dukungan failover. Kami merekomendasikan Multi-AZ untuk beban kerja produksi agar menjaga ketersediaan tetap tinggi.
	Untuk pengembangan dan pengujian, Anda dapat memilih Jangan buat instans siaga.
	Untuk informasi selengkapnya, lihat Mengonfigurasi dan mengelola deployment Multi-AZ.
		Opsi CLI:
		`--multi-az`
		`--no-multi-az`
		Parameter API RDS:
		`MultiAZ`
			Semua

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung

Set karakter nasional (NCHAR)	Set karakter nasional untuk instans DB Anda, biasanya disebut set karakter NCHAR. Anda dapat mengatur set karakter nasional ke AL16UTF16 (default) atau UTF-8. Anda tidak dapat mengubah set karakter nasional setelah Anda membuat instans DB.
	
	Set karakter nasional berbeda dari set karakter DB. Tidak seperti set karakter DB, set karakter nasional menentukan pengodean hanya untuk kolom tipe data NCHAR (NCHAR, NVARCHAR2, dan NCLOB) tanpa memengaruhi metadata basis data.
	
	Untuk informasi selengkapnya, lihat Set karakter RDS for Oracle.
		Opsi CLI:
		
		`--nchar-character-set-name`
		
		Parameter API:
		
		`NcharCharacterSetName`
			Oracle

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung

Jenis jaringan	Protokol alamat IP didukung oleh instans DB.
	
	IPv4 (default) untuk menentukan bahwa sumber daya dapat berkomunikasi dengan instans DB hanya melalui protokol alamat Internet Protocol versi 4 (IPv4).
	
	Mode tumpukan ganda untuk menentukan bahwa sumber daya dapat berkomunikasi dengan instans DB melalui IPv4, Internet Protocol versi 6 (IPv6), atau keduanya. Gunakan mode tumpukan ganda jika Anda memiliki sumber daya yang harus berkomunikasi dengan instans DB melalui protokol pengalamatan IPv6. Selain itu, pastikan Anda mengaitkan blok CIDR IPv6 dengan semua subnet dalam grup subnet DB yang Anda tentukan.
	
	Untuk informasi selengkapnya, lihat Penentuan alamat IP Amazon RDS.
		Opsi CLI:
		
		`--network-type`
		
		Parameter API RDS:
		
		`NetworkType`
			Semua

Grup opsi	Grup opsi untuk instans DB Anda. Anda dapat memilih grup opsi default atau Anda dapat membuat grup opsi kustom.
	
	Untuk informasi selengkapnya, lihat Menggunakan grup opsi.
		Opsi CLI:
		
		`--option-group-name`
		
		Parameter API RDS:
		
		`OptionGroupName`
			Semua

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung

Wawasan Performa	Aktifkan Wawasan Performa untuk memantau beban instans DB Anda sehingga Anda dapat menganalisis dan memecahkan masalah performa basis data Anda.
	
	Pilih periode retensi untuk menentukan berapa banyak riwayat data Wawasan Performa yang akan disimpan. Pengaturan retensi di tingkat gratis adalah Default (7 hari). Untuk mempertahankan data performa Anda lebih lama, tetapkan 1–24 bulan. Untuk informasi selengkapnya tentang periode retensi,lihat Harga dan retensi data untuk Wawasan Performa.
	
	Pilih kunci KMS yang akan digunakan untuk melindungi kunci yang digunakan untuk mengenkripsi volume basis data ini. Pilih dari kunci KMS di akun Anda, atau masukkan kunci dari akun yang berbeda.
	
	Untuk informasi selengkapnya, lihat Memantau muatan DB dengan Wawasan Performa di Amazon RDS.
		Opsi CLI:
		
		`--enable-performance-insights`
		
		`--no-enable-performance-insights`
		
		`--performance-insights-retention-period`
		
		`--performance-insights-kms-key-id`
		
		Parameter API RDS:
		
		`EnablePerformanceInsights`
		
		`PerformanceInsightsRetentionPeriod`
		
		`PerformanceInsightsKMSKeyId`
			Semua kecuali Db2

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung

IOPS yang Tersedia	Nilai IOPS (operasi I/O per detik) yang Tersedia untuk instans DB. Pengaturan ini hanya tersedia jika Anda memilih salah satu Jenis penyimpanan berikut:
	• SSD tujuan umum (gp3)
	• SSD IOPS yang tersedia (io1)
	• IOPS SSD yang disediakan (io2)
	
	Untuk informasi selengkapnya, lihat Penyimpanan instans DB Amazon RDS.
		Opsi CLI:
		`--iops`
		Parameter API RDS:
		`Iops`
			Semua

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung

Akses publik	Ya untuk memberikan alamat IP publik instans DB, yang berarti instans dapat diakses di luar VPC. Agar dapat diakses oleh publik, instans DB juga harus berada di subnet publik di VPC.
	
	Tidak untuk membuat instans DB hanya dapat diakses dari dalam VPC.
	
	Untuk informasi selengkapnya, lihat Menyembunyikan klaster DB dalam VPC dari internet.
	
	Untuk terhubung ke instans DB dari luar VPC, instans DB harus dapat diakses publik. Selain itu, akses harus diberikan menggunakan aturan masuk grup keamanan instans DB. Selain itu, persyaratan lain harus dipenuhi. Untuk informasi selengkapnya, lihat Tidak dapat terhubung ke instans DB Amazon RDS.
	
	Jika instans DB Anda tidak dapat diakses publik, gunakan koneksi VPN AWS Site-to-Site AWS Direct Connect atau koneksi untuk mengaksesnya dari jaringan pribadi. Untuk informasi selengkapnya, lihat Privasi lalu lintas jaringan internet.
		Opsi CLI:
		
		`--publicly-accessible`
		
		`--no-publicly-accessible`
		
		Parameter API RDS:
		
		`PubliclyAccessible`
			Semua

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung

RDS Extended Support	Pilih Aktifkan RDS Extended Support untuk memungkinkan versi mesin utama yang didukung untuk terus berjalan melewati akhir RDS dari tanggal dukungan standar.
	
	Saat Anda membuat instans DB, Amazon RDS default ke RDS Extended Support. Untuk mencegah pembuatan instans DB baru setelah RDS berakhir pada tanggal dukungan standar dan untuk menghindari biaya untuk RDS Extended Support, nonaktifkan pengaturan ini. Instans DB Anda yang ada tidak akan dikenakan biaya hingga tanggal mulai penetapan harga RDS Extended Support.
	
	Untuk informasi selengkapnya, lihat Menggunakan Dukungan Diperpanjang Amazon RDS.
		Opsi CLI:
		
		`--engine-lifecycle-support`
		
		Parameter API RDS:
		
		`EngineLifecycleSupport`
			MySQL
			
			PostgreSQL

Proksi RDS	Pilih Buat Proksi RDS untuk membuat proksi untuk instans DB Anda. Amazon RDS secara otomatis membuat peran IAM dan rahasia Secrets Manager untuk proksi. Untuk informasi selengkapnya, lihat Menggunakan Proksi Amazon RDS.	Tidak tersedia saat membuat instans DB.	MariaDB MySQL PostgreSQL

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung

Penskalaan otomatis penyimpanan	Aktifkan penskalaan otomatis penyimpanan agar Amazon RDS dapat secara otomatis menambah penyimpanan saat diperlukan untuk menghindari kehabisan ruang penyimpanan pada instans DB Anda.
	
	Gunakan Ambang batas penyimpanan maksimum untuk mengatur batas maksimum penambahan peningkatan penyimpanan otomatis untuk instans DB Anda oleh Amazon RDS. Defaultnya adalah 1.000 GiB.
	
	Untuk informasi selengkapnya, lihat Mengelola kapasitas secara otomatis dengan penskalaan otomatis penyimpanan Amazon RDS.
		Opsi CLI:
		
		`--max-allocated-storage`
		
		Parameter API RDS:
		
		`MaxAllocatedStorage`
			Semua

Throughput penyimpanan	Nilai throughput penyimpanan untuk instans DB. Pengaturan ini hanya tersedia jika Anda memilih SSD tujuan umum (gp3) untuk Jenis penyimpanan.
	
	Untuk informasi selengkapnya, lihat Penyimpanan gp3.
		Opsi CLI:
		
		`--storage-throughput`
		
		Parameter API RDS:
		
		`StorageThroughput`
			Semua

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung

Jenis penyimpanan	Jenis penyimpanan untuk instans DB Anda.
	
	Jika Anda memilih SSD Tujuan Umum (gp3), Anda dapat menyediakan IOPS yang tersedia tambahan dan throughput penyimpanan di bawah Pengaturan lanjutan.
	
	Jika Anda memilih Provisioned IOPS SSD (io1) atau Provisioned IOPS SSD (io2), masukkan nilai IOPS Provisioned.
	
	Untuk informasi selengkapnya, lihat Jenis penyimpanan Amazon RDS.
		Opsi CLI:
		
		`--storage-type`
		
		Parameter API RDS:
		
		`StorageType`
			Semua

Grup subnet	Grup subnet DB untuk dihubungkan dengan instans DB ini.
	
	Untuk informasi selengkapnya, lihat Bekerja dengan grup subnet DB.
		Opsi CLI:
		
		`--db-subnet-group-name`
		
		Parameter API RDS:
		
		`DBSubnetGroupName`
			Semua

Nama basis data penghuni	Nama PDB awal Anda dalam konfigurasi multi-penghuni arsitektur Oracle. Pengaturan ini hanya tersedia jika Anda memilih Konfigurasi multi-penghuni untuk Konfigurasi arsitektur.
	
	Nama basis data penghuni harus berbeda dari nama CDB Anda, yang diberi nama `RDSCDB`. Anda tidak dapat mengubah nama CDB.
		Opsi CLI:
		
		`--db-name`
		
		Parameter API RDS:
		
		`DBName`
			Oracle

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung

Nama pengguna master basis data penghuni	Nama yang Anda gunakan sebagai nama pengguna master untuk masuk ke basis data penghuni (PDB) Anda dengan semua hak istimewa basis data. Pengaturan ini hanya tersedia jika Anda memilih Konfigurasi multi-penghuni untuk Konfigurasi arsitektur.
	
	Perhatikan batasan penamaan berikut:
	
	• Nama dapat berisi 1–16 karakter alfanumerik dan garis bawah.
	• Karakter pertama harus berupa huruf.
	• Nama tidak dapat berupa kata yang disimpan oleh mesin basis data.
	
	Anda tidak dapat melakukan hal berikut:
	
	• Mengubah nama pengguna master penghuni setelah Anda membuat basis data penghuni.
	• Masuk dengan nama pengguna master penghuni ke CDB.
		Opsi CLI:
		
		`--master-username`
		
		Parameter API RDS:
		
		`MasterUsername`
			Oracle

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung

Kata sandi master basis data penghuni	Kata sandi untuk akun pengguna master basis data penghuni (PDB) Anda. Pengaturan ini hanya tersedia jika Anda memilih Konfigurasi multi-penghuni untuk Konfigurasi arsitektur.
	
	Kata sandi memiliki 8-30 karakter ASCII yang dapat dicetak, tidak termasuk `/`, `"`, spasi, dan `@`.
		Opsi CLI:
		
		`--master-password`
		
		Parameter API RDS:
		
		`MasterPassword`
			Oracle

Set karakter basis data penghuni	Set karakter dari basis data penghuni awal. Pengaturan ini hanya tersedia jika Anda memilih Konfigurasi multi-penghuni untuk Konfigurasi arsitektur. Hanya instans RDS for Oracle CDB yang didukung.
	
	Nilai default AL32UTF8 untuk set karakter basis data penghuni adalah untuk set karakter Universal Unicode 5.0 UTF-8. Anda dapat memilih set karakter basis data penghuni yang berbeda dari set karakter CDB.
	
	Untuk informasi selengkapnya, lihat Set karakter RDS for Oracle.
		Opsi CLI:
		
		`--character-set-name`
		
		Parameter API RDS:
		
		`CharacterSetName`
			Oracle

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung

Set karakter nasional basis data penghuni	Set karakter nasional untuk basis data penghuni Anda, biasanya disebut set karakter NCHAR. Pengaturan ini hanya tersedia jika Anda memilih Konfigura si multi-penghuni untuk Konfigurasi arsitektur. Hanya instans RDS for Oracle CDB yang didukung.
	
	Anda dapat mengatur set karakter nasional ke AL16UTF16 (default) atau UTF-8. Anda tidak dapat mengubah set karakter nasional setelah Anda membuat basis data penghuni.
	
	Set karakter nasional basis data penghuni berbeda dari set karakter basis data penghuni. Set karakter nasional menentukan pengodean hanya untuk kolom yang menggunakan tipe data NCHAR (NCHAR, NVARCHAR2 , dan NCLOB) serta tidak memengaruhi metadata basis data.
	
	Untuk informasi selengkapnya, lihat Set karakter RDS for Oracle.
		Opsi CLI:
		
		`--nchar-character-set-name`
		
		Parameter API:
		
		`NcharCharacterSetName`
			Oracle

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Mesin DB yang didukung

Zona waktu	Zona waktu untuk instans DB Anda. Jika Anda tidak memilih zona waktu, instans DB Anda menggunakan zona waktu default. Anda tidak dapat mengubah zona waktu setelah instans DB dibuat.
	
	Untuk informasi selengkapnya, lihat Zona waktu lokal untuk instans DB Microsoft SQL Server.
		Opsi CLI:
		
		`--timezone`
		
		Parameter API RDS:
		
		`Timezone`
			SQL Server
			
			RDS Custom for SQL Server

Cloud Privat Virtual (VPC)	VPC berbasis layanan Amazon VPC yang akan dihubungkan dengan instans DB ini. Untuk informasi selengkapnya, lihat Amazon VPC dan Amazon RDS.	Untuk CLI dan API, Anda menentukan ID grup keamanan VPC.	Semua

Grup keamanan VPC (firewall)	Grup keamanan untuk dihubungkan dengan instans DB.
	
	Untuk informasi selengkapnya, lihat Ikhtisar grup keamanan VPC.
		Opsi CLI:
		
		`--vpc-security-group-ids`
		
		Parameter API RDS:
		
		`VpcSecurityGroupIds`
			Semua

Membuat sumber daya Amazon RDS dengan AWS CloudFormation

Amazon RDS terintegrasi dengan AWS CloudFormation, layanan yang membantu Anda memodelkan dan mengatur sumber daya AWS sehingga Anda dapat menghabiskan lebih sedikit waktu untuk membuat dan mengelola sumber daya dan infrastruktur Anda. Anda membuat templat yang menjelaskan semua sumber daya AWS yang Anda inginkan (seperti instans DB dan grup parameter DB), dan AWS CloudFormation menyediakan dan mengonfigurasi sumber daya tersebut untuk Anda.

Saat menggunakan AWS CloudFormation, Anda dapat menggunakan kembali templat Anda untuk mengatur sumber daya RDS secara konsisten dan berulang kali. Jelaskan sumber daya Anda sekali, lalu sediakan sumber daya yang sama berulang-ulang dalam beberapa akun dan Wilayah AWS.

RDS dan templat AWS CloudFormation

Untuk menyediakan dan mengonfigurasi sumber daya bagi RDS dan layanan terkait, Anda harus memahami templat AWS CloudFormation. Templat adalah file teks dengan format JSON atau YAML. Templat ini menjelaskan sumber daya yang ingin Anda sediakan di tumpukan AWS CloudFormation. Jika tidak terbiasa dengan JSON atau YAML, Anda dapat menggunakan AWS CloudFormation Designer untuk membantu Anda memulai dengan templat AWS CloudFormation. Untuk informasi selengkapnya, lihat Apa itu AWS CloudFormation Designer? di Panduan Pengguna AWS CloudFormation.

RDS mendukung pembuatan sumber daya di AWS CloudFormation. Untuk informasi selengkapnya, termasuk contoh templat JSON dan YAML untuk sumber daya ini, lihat referensi jenis sumber daya RDS di Panduan Pengguna AWS CloudFormation.

Pelajari selengkapnya tentang AWS CloudFormation

Untuk mempelajari selengkapnya tentang AWS CloudFormation, lihat sumber daya berikut:

- AWS CloudFormation
- Panduan Pengguna AWS CloudFormation
- Referensi AWS CloudFormation API
- Panduan Pengguna Antarmuka Baris Perintah AWS CloudFormation

Menghubungkan ke instans DB Amazon RDS

Sebelum dapat menghubungkan ke instans DB, Anda harus membuat instans DB. Untuk informasi, lihat Membuat instans DB Amazon RDS. Setelah Amazon RDS menyediakan instans DB Anda, gunakan aplikasi klien standar atau utilitas untuk mesin DB Anda guna terhubung ke instans DB tersebut. Dalam string koneksi, tentukan alamat DNS dari titik akhir instans DB sebagai parameter host. Tentukan juga nomor port dari titik akhir instans DB sebagai parameter port.

Topik

Menemukan informasi koneksi untuk instans DB Amazon RDS

Informasi koneksi untuk instans DB mencakup titik akhir, port, dan pengguna basis data yang valid, seperti pengguna utama. Misalnya, untuk instans DB MySQL, asumsikan bahwa nilai titik akhirnya adalah `mydb.123456789012.us-east-1.rds.amazonaws.com`. Dalam hal ini, nilai port-nya adalah `3306`, dan pengguna basis datanya adalah `admin`. Dengan informasi ini, Anda menentukan nilai-nilai berikut dalam string koneksi:

- Untuk host atau nama host, atau nama DNS, tentukan `mydb.123456789012.us-east-1.rds.amazonaws.com`.
- Untuk port, tentukan `3306`.
- Untuk pengguna, tentukan `admin`.

Titik akhir bersifat unik untuk setiap instans DB, dan nilai-nilai port serta pengguna dapat bervariasi. Daftar berikut ini menunjukkan port yang paling umum untuk setiap mesin DB:

- Db2 – 50000
- MariaDB – 3306
- Microsoft SQL Server – 1433

- MySQL – 3306
- Oracle – 1521
- PostgreSQL – 5432

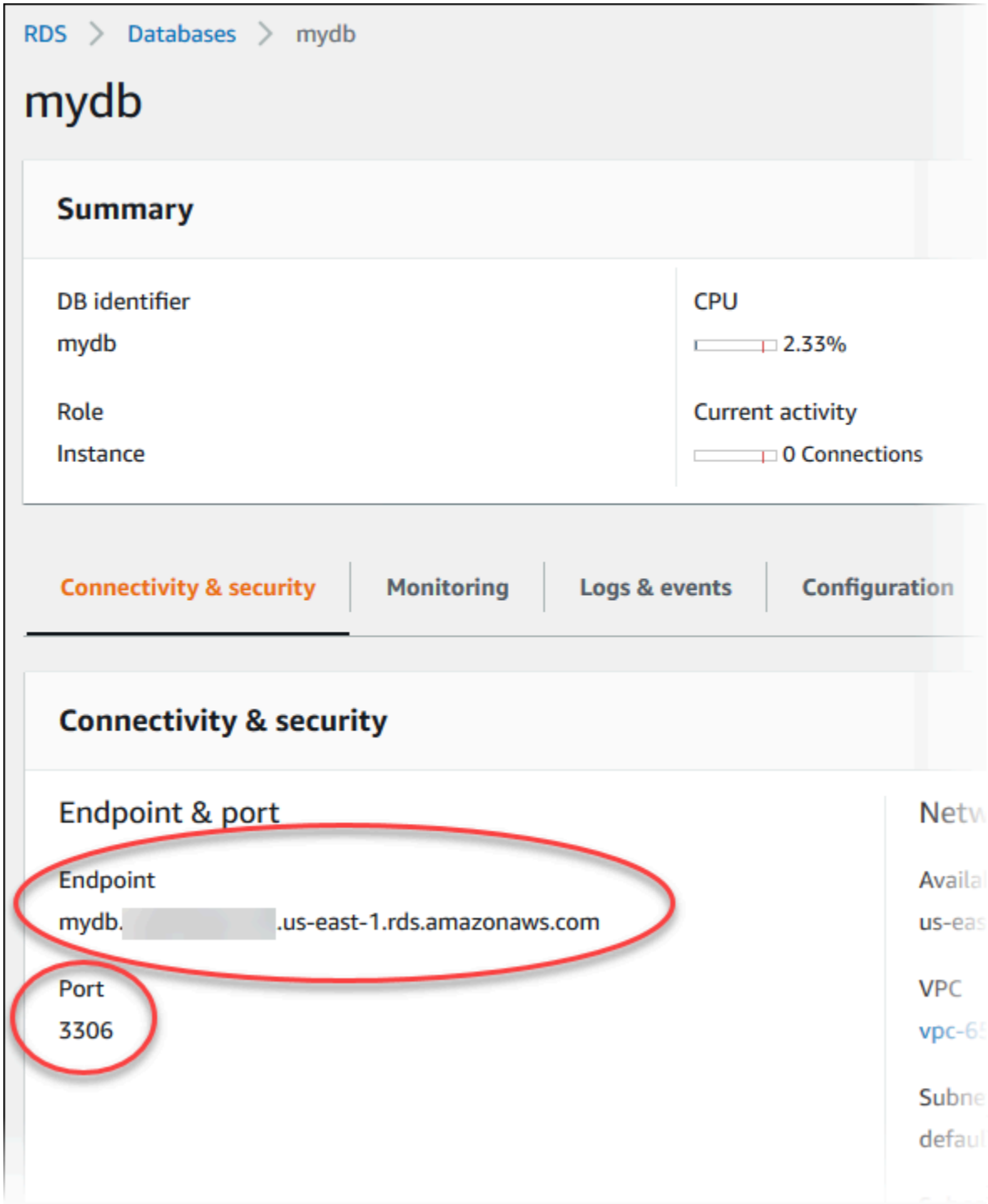
Untuk terhubung ke instans DB, gunakan klien apa saja untuk mesin DB. Misalnya, Anda dapat menggunakan utilitas mysql untuk terhubung ke instans DB MariaDB atau MySQL. Anda dapat menggunakan Microsoft SQL Server Management Studio untuk terhubung ke instans DB SQL Server. Anda dapat menggunakan Oracle SQL Developer untuk terhubung ke instans DB Oracle. Demikian pula, Anda dapat menggunakan utilitas baris perintah psql untuk terhubung ke instans DB PostgreSQL.

Untuk menemukan informasi koneksi instans DB, gunakan AWS Management Console. Anda juga dapat menggunakan describe-db-instancesperintah AWS Command Line Interface (AWS CLI) atau operasi RDS API DescribedBInstances.

Konsol

Untuk menemukan informasi koneksi instans DB di AWS Management Console

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data untuk menampilkan daftar instans DB Anda.
3. Pilih nama instans DB untuk menampilkan detailnya.
4. Di tab Konektivitas & keamanan, salin titik akhir. Selain itu, catat nomor porta. Anda memerlukan titik akhir dan nomor port untuk terhubung ke instans DB.



5. Jika Anda perlu menemukan nama pengguna utama, pilih tab Konfigurasi dan lihat nilai Nama pengguna utama.

AWS CLI

Untuk menemukan informasi koneksi untuk instans DB dengan menggunakanAWS CLI, panggil describe-db-instancesperintah. Dalam panggilan tersebut, buat kueri untuk ID instans DB, titik akhir, port, dan nama pengguna utama.

Untuk Linux, macOS, atau Unix:

```
aws rds describe-db-instances \
  --query "*[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername]"
```

Untuk Windows:

```
aws rds describe-db-instances ^
  --query "*[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername]"
```

Output Anda akan terlihat seperti berikut ini.

```
[
    [
        "mydb",
        "mydb.123456789012.us-east-1.rds.amazonaws.com",
        3306,
        "admin"
    ],
    [
        "myoracledb",
        "myoracledb.123456789012.us-east-1.rds.amazonaws.com",
        1521,
        "dbadmin"
    ],
    [
        "mypostgresqldb",
        "mypostgresqldb.123456789012.us-east-1.rds.amazonaws.com",
        5432,
        "postgresadmin"
    ]
]
```

API RDS

Untuk menemukan informasi koneksi instans DB dengan menggunakan API Amazon RDS, panggil operasi DescribeDBInstances. Dalam output, temukan nilai untuk alamat titik akhir, port titik akhir, dan nama pengguna utama.

Opsi autentikasi basis data

Amazon RDS mendukung cara-cara berikut untuk mengautentikasi pengguna basis data:

- Autentikasi kata sandi – Instans DB Anda melakukan semua administrasi akun pengguna. Anda membuat pengguna dan menentukan kata sandi dengan pernyataan SQL. Pernyataan SQL yang dapat Anda gunakan tergantung pada mesin DB Anda.
- Autentikasi basis data (IAM) AWS Identity and Access Management – Anda tidak perlu menggunakan kata sandi saat menghubungkan ke instans DB. Sebagai gantinya, Anda menggunakan token autentikasi.
- Autentikasi Kerberos – Anda menggunakan autentikasi eksternal pengguna basis data menggunakan Kerberos dan Microsoft Active Directory. Kerberos adalah protokol autentikasi jaringan yang menggunakan tiket dan kriptografi kunci-simetris agar tidak perlu mentransmisikan kata sandi melalui jaringan. Kerberos telah disematkan dalam Active Directory dan dirancang untuk mengautentikasi pengguna ke sumber daya jaringan, seperti basis data.

Autentikasi basis data IAM dan autentikasi Kerberos hanya tersedia untuk mesin dan versi DB tertentu.

Untuk informasi selengkapnya, lihat Autentikasi basis data dengan Amazon RDS.

Koneksi terenkripsi

Anda dapat menggunakan Lapisan Soket Aman (SSL) atau Keamanan Lapisan Pengangkutan (TLS) dari aplikasi Anda untuk mengenkripsi koneksi ke instans DB. Setiap mesin DB memiliki proses sendiri untuk menerapkan SSL/TLS. Untuk informasi selengkapnya, lihat .

Skenario untuk mengakses instans DB di VPC

Dengan menggunakan Amazon Virtual Private Cloud (Amazon VPC), Anda dapat meluncurkan sumber daya AWS, seperti instans DB Amazon RDS, ke dalam cloud privat virtual (VPC). Saat menggunakan Amazon VPC, Anda dapat mengontrol lingkungan jaringan virtual Anda. Anda dapat

memilih rentang alamat IP Anda sendiri, membuat subnet, dan mengonfigurasi daftar kontrol akses dan perutean.

Grup keamanan VPC mengontrol akses ke instans DB di dalam VPC. Setiap aturan grup keamanan VPC memungkinkan sumber tertentu untuk mengakses instans DB di VPC yang terkait dengan grup keamanan VPC tersebut. Sumber tersebut dapat berupa rentang alamat (misalnya, 203.0.113.0/24), atau grup keamanan VPC lainnya. Dengan menentukan grup keamanan VPC sebagai sumber, Anda mengizinkan lalu lintas masuk dari semua instans (biasanya server aplikasi) yang menggunakan grup keamanan VPC sumber.

Sebelum mencoba menghubungkan ke instans DB Anda, konfigurasikan VPC untuk kasus penggunaan Anda. Berikut ini adalah skenario umum untuk mengakses instans DB di VPC:

- Instans DB di VPC yang diakses oleh instans Amazon EC2 di VPC yang sama – Penggunaan umum instans DB di VPC adalah membagikan data dengan server aplikasi yang berjalan di instans EC2 dalam VPC yang sama. Instans EC2 mungkin menjalankan server web dengan aplikasi yang berinteraksi dengan instans DB tersebut.
- Instans DB di VPC yang diakses oleh instans EC2 di VPC yang berbeda – Dalam beberapa kasus, instans DB Anda berada di VPC yang berbeda dari instans EC2 yang Anda gunakan untuk mengaksesnya. Jika demikian, Anda dapat menggunakan peering VPC untuk mengakses instans DB.
- Instans DB di VPC yang diakses oleh aplikasi klien melalui internet – Untuk mengakses instans DB di VPC dari aplikasi klien melalui internet, Anda mengonfigurasi VPC dengan subnet publik tunggal. Anda juga mengonfigurasi gateway internet untuk memungkinkan komunikasi melalui internet.

  Untuk terhubung ke instans DB dari luar VPC-nya, instans DB harus dapat diakses secara publik. Selain itu, akses harus diberikan menggunakan aturan masuk grup keamanan instans DB, dan persyaratan lain harus terpenuhi. Untuk informasi selengkapnya, lihat Tidak dapat terhubung ke instans DB Amazon RDS.
- Instans DB di VPC yang diakses oleh jaringan privat – Jika instans DB Anda tidak dapat diakses secara publik, Anda dapat menggunakan salah satu opsi berikut untuk mengaksesnya dari jaringan privat:
  - Koneksi Site-to-Site VPN AWS
  - Koneksi AWS Direct Connect
  - Koneksi AWS Client VPN

Untuk informasi selengkapnya, lihat Skenario untuk mengakses instans DB di VPC.

Menghubungkan ke instans DB yang menjalankan mesin DB tertentu

Untuk informasi tentang cara menghubungkan ke instans DB yang menjalankan mesin DB tertentu, ikuti petunjuk untuk mesin DB Anda:

- Menghubungkan ke instans DB RDS untuk Db2 Anda
- Menghubungkan ke instans DB yang menjalankan mesin basis data MariaDB
- Menghubungkan ke instans DB yang menjalankan mesin basis data Microsoft SQL Server
- Menghubungkan ke instans DB yang menjalankan mesin basis data MySQL
- Menghubungkan ke instans RDS for Oracle DB
- Menghubungkan ke instans DB yang menjalankan mesin basis data PostgreSQL

Mengelola koneksi dengan Proksi RDS

Anda juga dapat menggunakan Proksi Amazon RDS untuk mengelola koneksi ke instans DB RDS for PostgreSQL, RDS for MariaDB, RDS for Microsoft SQL Server, dan RDS for MySQL. Proksi RDS memungkinkan aplikasi untuk menyatukan dan berbagi koneksi basis data guna meningkatkan skalabilitas. Untuk informasi selengkapnya, lihat Menggunakan Proksi Amazon RDS.

Menggunakan grup opsi

Beberapa mesin DB menawarkan fitur-fitur tambahan yang memudahkan pengelolaan data dan basis data, serta untuk menyediakan keamanan tambahan bagi basis data Anda. Amazon RDS menggunakan grup opsi untuk mengaktifkan dan mengonfigurasi fitur ini. Grup opsi dapat menentukan fitur, opsi yang dipanggil, yang tersedia untuk instans DB Amazon RDS tertentu. Opsi dapat memiliki pengaturan yang menentukan cara kerja opsi. Saat Anda mengaitkan instans DB dengan grup opsi, pengaturan opsi dan opsi yang ditentukan diaktifkan untuk instans DB tersebut.

Amazon RDS mendukung opsi untuk mesin basis data berikut:

Mesin basis data	Dokumentasi terkait
`MariaDB`	Opsi untuk mesin basis data MariaDB
`Microsoft SQL Server`	Opsi untuk mesin basis data Microsoft SQL Server
`MySQL`	Opsi untuk instans DB MySQL
`Oracle`	Menambahkan opsi untuk instans DB Oracle
`PostgreSQL`	PostgreSQL tidak menggunakan opsi dan grup opsi. PostgreSQL menggunakan ekstensi dan modul untuk memberikan fitur tambahan. Untuk informasi selengkapnya, lihat Versi ekstensi PostgreSQL yang didukung.

Gambaran umum grup opsi

Amazon RDS menyediakan grup opsi default kosong untuk setiap instans DB baru. Anda tidak dapat mengubah atau menghapus grup opsi default ini, tetapi grup opsi baru yang Anda buat akan mendapatkan pengaturannya dari grup opsi default. Untuk menerapkan opsi ke instans DB, Anda harus melakukan hal berikut:

1. Buat grup opsi baru, atau salin atau ubah grup opsi yang ada.
2. Tambahkan satu atau beberapa opsi ke grup opsi.
3. Kaitkan grup opsi dengan instans DB.

Untuk mengaitkan grup opsi dengan instans DB, ubah instans DB. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Instans DB dan snapshot DB dapat dikaitkan dengan grup opsi. Dalam beberapa kasus, Anda mungkin memulihkan dari snapshot DB atau melakukan point-in-time pemulihan untuk instans DB. Dalam hal ini, grup opsi yang terkait dengan snapshot DB atau instans DB akan secara default dikaitkan dengan instans DB yang dipulihkan. Anda dapat mengaitkan grup opsi yang berbeda dengan instans DB yang dipulihkan. Namun, grup opsi baru harus berisi opsi persisten atau permanen yang disertakan dalam grup opsi asli. Opsi persisten dan permanen dijelaskan sebagai berikut.

Opsi memerlukan memori tambahan agar bisa berjalan pada instans DB. Dengan demikian, Anda mungkin perlu meluncurkan instans yang lebih besar untuk menggunakannya, bergantung pada penggunaan Anda saat ini atas instans DB Anda. Misalnya, Oracle Enterprise Manager Database Control menggunakan sekitar 300 MB RAM. Jika Anda mengaktifkan opsi ini untuk instans DB kecil, Anda mungkin mengalami masalah atau out-of-memory kesalahan kinerja.

Opsi persisten dan permanen

Dua jenis opsi, persisten dan permanen, memerlukan pertimbangan khusus ketika Anda menambahkannya ke grup opsi.

Opsi persisten tidak dapat dihapus dari grup opsi sementara instans DB dikaitkan dengan grup opsi. Contoh opsi yang persisten adalah opsi TDE untuk enkripsi data transparan (TDE) Microsoft SQL Server. Anda harus membatalkan pengaitan semua instans DB dari grup opsi sebelum opsi persisten dapat dihapus dari grup opsi. Dalam beberapa kasus, Anda mungkin memulihkan atau melakukan point-in-time pemulihan dari snapshot DB. Dalam hal ini, jika grup opsi yang terkait dengan snapshot DB tersebut berisi opsi persisten, Anda hanya dapat mengaitkan instans DB yang dipulihkan dengan grup opsi tersebut.

Opsi permanen, seperti opsi TDE untuk Oracle Advanced Security TDE, tidak dapat dihapus dari grup opsi. Anda dapat mengubah grup opsi instans DB yang menggunakan opsi permanen. Namun, grup opsi yang terkait dengan instans DB tersebut harus menyertakan opsi permanen yang sama. Dalam beberapa kasus, Anda mungkin memulihkan atau melakukan point-in-time pemulihan dari snapshot DB. Dalam hal ini, jika grup opsi yang terkait dengan snapshot DB tersebut berisi opsi permanen, Anda hanya dapat mengaitkan instans DB yang dipulihkan dengan grup opsi dengan opsi permanen tersebut.

Untuk instans DB Oracle, Anda dapat menyalin snapshot DB yang dibagikan yang memiliki opsi `Timezone` atau `OLS` (atau keduanya). Untuk melakukannya, tentukan grup opsi target yang mencakup opsi ini saat Anda menyalin snapshot DB. Opsi OLS bersifat permanen dan persisten hanya untuk instans DB Oracle yang menjalankan Oracle versi 12.2 atau lebih tinggi. Untuk informasi selengkapnya tentang opsi ini, lihat Zona waktu Oracle dan Keamanan Label Oracle.

Pertimbangan VPC

Grup opsi yang terkait dengan instans DB akan ditautkan ke VPC instans DB. Hal ini berarti bahwa Anda tidak dapat menggunakan grup opsi yang ditetapkan ke instans DB jika Anda mencoba memulihkan instans tersebut ke VPC yang berbeda. Jika memulihkan instans DB ke VPC yang berbeda, Anda dapat melakukan salah satu dari berikut ini:

- Tetapkan grup opsi default untuk instans DB.
- Tetapkan grup opsi yang ditautkan ke VPC tersebut.
- Buat grup opsi baru dan tetapkan ke instans DB.

Dengan opsi persisten atau permanen, seperti Oracle TDE, Anda harus membuat grup opsi baru. Grup opsi ini harus menyertakan opsi persisten atau permanen saat memulihkan instans DB ke VPC yang berbeda.

Pengaturan opsi mengendalikan perilaku opsi. Misalnya, opsi Oracle Advanced Security `NATIVE_NETWORK_ENCRYPTION` memiliki pengaturan yang dapat Anda gunakan untuk menentukan algoritma enkripsi untuk lalu lintas jaringan ke dan dari Pembaruan DB. Beberapa pengaturan opsi dioptimalkan untuk digunakan dengan Amazon RDS dan tidak dapat diubah.

Opsi yang saling eksklusif

Beberapa opsi bersifat saling eksklusif. Anda dapat menggunakan salah satunya, tetapi tidak keduanya sekaligus. Opsi-opsi berikut ini bersifat saling eksklusif:

- Oracle Enterprise Manager Database Express dan Oracle Management Agent untuk Kontrol Cloud Enterprise Manager.
- Enkripsi jaringan asli Oracle dan Lapisan Soket Aman Oracle.

Membuat grup opsi

Anda dapat membuat grup opsi baru yang mendapatkan pengaturannya dari grup opsi default. Anda kemudian perlu menambahkan satu atau beberapa opsi ke grup opsi baru. Atau, jika Anda sudah memiliki grup opsi, Anda dapat menyalin grup opsi tersebut dengan semua opsinya ke grup opsi baru. Untuk informasi selengkapnya, lihat Menyalin grup opsi.

Setelah Anda membuat grup opsi baru, grup ini tidak memiliki opsi. Untuk mempelajari cara menambahkan opsi ke grup opsi, lihat Menambahkan opsi ke grup opsi. Setelah menambahkan opsi yang diinginkan, Anda kemudian dapat mengaitkan grup opsi dengan instans DB. Dengan cara ini, opsi menjadi tersedia pada instans DB. Untuk informasi tentang mengaitkan grup opsi dengan instans DB, lihat dokumentasi untuk mesin Anda dalam Menggunakan grup opsi.

Konsol

Salah satu cara untuk membuat grup opsi adalah dengan menggunakan AWS Management Console.

Untuk membuat grup opsi baru menggunakan konsol

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup opsi.
3. Pilih Buat grup.
4. Di jendela Buat grup opsi, lakukan hal berikut:
    a. Untuk Nama, ketikkan nama untuk grup opsi yang unik di dalam AWS akun Anda. Nama tersebut hanya boleh berisi huruf, angka, dan tanda hubung.
    b. Untuk Deskripsi, ketikkan deskripsi singkat grup opsi. Deskripsi digunakan untuk tampilan.
    c. Untuk Mesin, pilih mesin DB yang Anda inginkan.
    d. Untuk Versi mesin utama, pilih versi mayor mesin DB yang Anda inginkan.
5. Untuk melanjutkan, pilih Buat. Untuk membatalkan operasi, pilih Batal.

AWS CLI

Untuk membuat grup opsi, gunakan AWS CLI `create-option-group`perintah dengan parameter yang diperlukan berikut.

- `--option-group-name`

- `--engine-name`
- `--major-engine-version`
- `--option-group-description`

Example

Contoh berikut membuat grup opsi bernama `testoptiongroup`, yang terkait dengan mesin DB Oracle Enterprise Edition. Deskripsi diapit dalam tanda petik.

Untuk Linux, macOS, atau Unix:

```
aws rds create-option-group \
    --option-group-name testoptiongroup \
    --engine-name oracle-ee \
    --major-engine-version 12.1 \
    --option-group-description "Test option group"
```

Untuk Windows:

```
aws rds create-option-group ^
    --option-group-name testoptiongroup ^
    --engine-name oracle-ee ^
    --major-engine-version 12.1 ^
    --option-group-description "Test option group"
```

API RDS

Untuk membuat grup opsi, panggil operasi `CreateOptionGroup` API Amazon RDS. Sertakan parameter berikut:

- `OptionGroupName`
- `EngineName`
- `MajorEngineVersion`
- `OptionGroupDescription`

Menyalin grup opsi

Anda dapat menggunakan AWS CLI atau Amazon RDS API menyalin grup opsi. Menyalin grup opsi bisa lebih mudah. Contohnya adalah ketika Anda memiliki grup opsi yang sudah ada dan ingin menyertakan sebagian besar parameter dan nilai kustomnya di grup opsi baru. Anda juga dapat membuat salinan grup opsi yang Anda gunakan dalam produksi lalu memodifikasi salinan tersebut untuk menguji pengaturan opsi lainnya.

> **Note**
>
> Saat ini, Anda tidak dapat menyalin grup opsi ke AWS Wilayah lain.

AWS CLI

Untuk menyalin grup opsi, gunakan AWS CLI copy-option-groupperintah. Sertakan opsi wajib berikut:

- `--source-option-group-identifier`
- `--target-option-group-identifier`
- `--target-option-group-description`

Example

Contoh berikut ini membuat grup opsi bernama `new-option-group`, yaitu salinan lokal dari grup opsi `my-option-group`.

Untuk Linux, macOS, atau Unix:

```
aws rds copy-option-group \
    --source-option-group-identifier my-option-group \
    --target-option-group-identifier new-option-group \
    --target-option-group-description "My new option group"
```

Untuk Windows:

```
aws rds copy-option-group ^
    --source-option-group-identifier my-option-group ^
    --target-option-group-identifier new-option-group ^
    --target-option-group-description "My new option group"
```

API RDS

Untuk menyalin grup opsi, hubungi CopyOptionGroupoperasi Amazon RDS API. Sertakan parameter wajib berikut.

- `SourceOptionGroupIdentifier`
- `TargetOptionGroupIdentifier`
- `TargetOptionGroupDescription`

Menambahkan opsi ke grup opsi

Anda dapat menambahkan opsi ke grup opsi yang sudah ada. Setelah menambahkan opsi yang diinginkan, Anda kemudian dapat mengaitkan grup opsi dengan instans DB sehingga opsi tersedia di instans DB. Untuk informasi tentang mengaitkan grup opsi dengan instans DB, lihat dokumentasi untuk mesin DB spesifik Anda yang tercantum dalam Menggunakan grup opsi.

Perubahan grup opsi harus diterapkan segera dalam dua kasus:

- Saat Anda menambahkan opsi yang menambahkan atau memperbarui nilai port, seperti opsi `OEM`.
- Saat Anda menambahkan atau menghapus grup opsi dengan opsi yang menyertakan nilai port.

Dalam kasus ini, pilih opsi Terapkan Segera pada konsol. Atau Anda dapat menyertakan opsi `--apply-immediately` saat menggunakan AWS CLI atau atur parameter `ApplyImmediately` ke `true` saat menggunakan API Amazon RDS. Opsi yang tidak menyertakan nilai port dapat segera diterapkan, atau dapat diterapkan selama periode pemeliharaan berikutnya untuk instans DB.

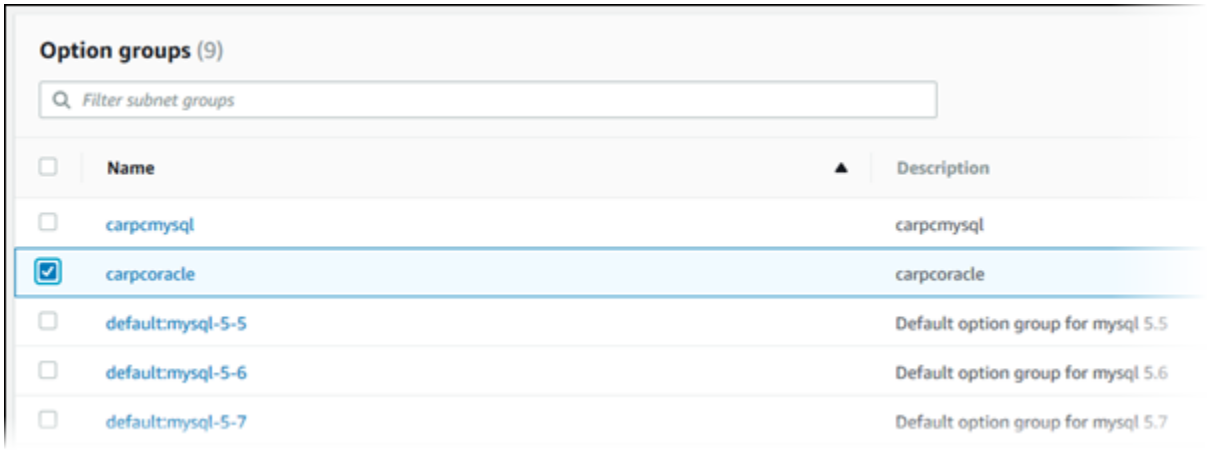
> **Note**
>
> Jika Anda menetapkan grup keamanan sebagai nilai untuk opsi dalam grup opsi, Anda perlu mengelola grup keamanan ini dengan memodifikasi grup opsi. Anda tidak dapat mengubah atau menghapus grup keamanan ini dengan memodifikasi instans DB. Selain itu, grup keamanan tidak muncul di detail instans DB di AWS Management Console atau di output untuk AWS CLI perintah`describe-db-instances`.

Konsol

Anda dapat menggunakan AWS Management Console untuk menambahkan opsi ke grup opsi.

Untuk menambahkan opsi ke grup opsi menggunakan konsol

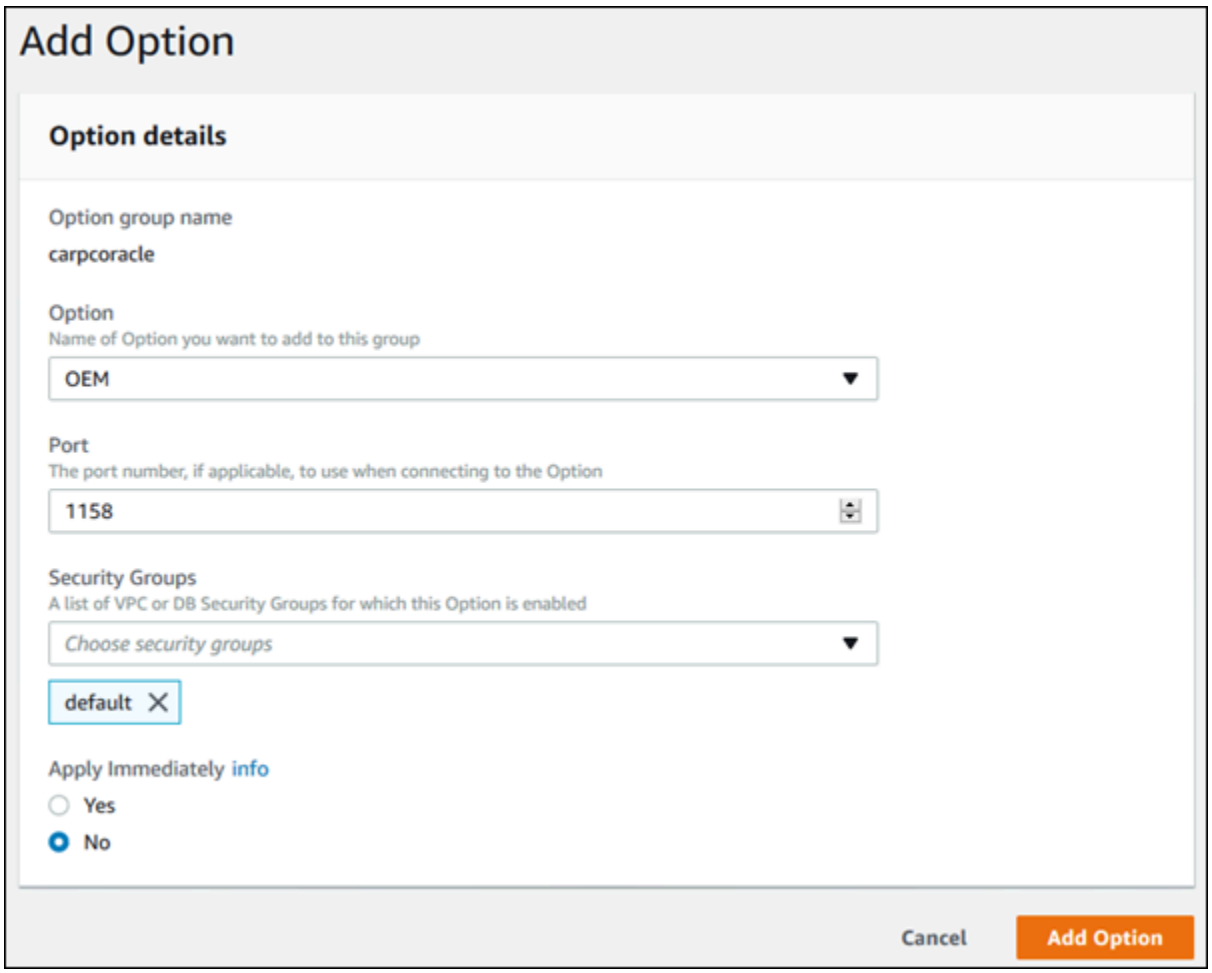
1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup opsi.
3. Pilih grup opsi yang ingin Anda ubah, lalu pilih Tambahkan opsi.



4. Di jendela Tambahkan opsi, lakukan hal berikut:

   a. Pilih opsi yang ingin Anda tambahkan. Anda mungkin perlu memberikan nilai tambahan, bergantung pada opsi yang Anda pilih. Misalnya, saat Anda memilih opsi `OEM`, Anda juga harus mengetikkan nilai port dan menentukan grup keamanan.

   b. Untuk mengaktifkan opsi pada semua instans DB terkait segera setelah Anda menambahkannya, untuk Terapkan Segera, pilih Ya. Jika Anda memilih Tidak (default), opsi diaktifkan untuk setiap instans DB terkait selama periode pemeliharaan berikutnya.



5. Jika pengaturan sudah sesuai keinginan Anda, pilih Tambahkan opsi.

AWS CLI

Untuk menambahkan opsi ke grup opsi, jalankan perintah AWS CLI add-option-to-option-group dengan opsi yang ingin Anda tambahkan. Untuk segera mengaktifkan opsi baru pada semua instans DB terkait, sertakan parameter `--apply-immediately`. Secara default, opsi tersebut diaktifkan untuk setiap instans DB terkait selama periode pemeliharaan berikutnya. Sertakan parameter wajib berikut:

- `--option-group-name`

Example

Contoh berikut menambahkan opsi `Timezone`, dengan pengaturan `America/Los_Angeles`, ke grup opsi `testoptiongroup` dan segera mengaktifkannya.

Untuk Linux, macOS, atau Unix:

```
aws rds add-option-to-option-group \
    --option-group-name testoptiongroup \
    --options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=America/
Los_Angeles}]" \
    --apply-immediately
```

Untuk Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name testoptiongroup ^
    --options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=America/
Los_Angeles}]" ^
    --apply-immediately
```

Output perintah akan seperti yang berikut ini:

```
...{
       "OptionName": "Timezone",
       "OptionDescription": "Change time zone",
       "Persistent": true,
       "Permanent": false,
       "OptionSettings": [
           {
               "Name": "TIME_ZONE",
               "Value": "America/Los_Angeles",
               "DefaultValue": "UTC",
               "Description": "Specifies the timezone the user wants to change the
 system time to",
               "ApplyType": "DYNAMIC",
               "DataType": "STRING",
               "AllowedValues": "Africa/Cairo,...",
               "IsModifiable": true,
               "IsCollection": false
           }
       ],
```

```
        "DBSecurityGroupMemberships": [],
        "VpcSecurityGroupMemberships": []
    }...
```

Example

Contoh berikut menambahkan opsi Oracle OEM ke grup opsi. Hal ini juga menentukan port kustom dan sepasang grup keamanan VPC Amazon EC2 yang akan digunakan untuk port tersebut.

Untuk Linux, macOS, atau Unix:

```
aws rds add-option-to-option-group \
    --option-group-name testoptiongroup \
    --options OptionName=OEM,Port=5500,VpcSecurityGroupMemberships="sg-test1,sg-test2"
 \
    --apply-immediately
```

Untuk Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name testoptiongroup ^
    --options OptionName=OEM,Port=5500,VpcSecurityGroupMemberships="sg-test1,sg-test2"
 ^
    --apply-immediately
```

Output perintah akan seperti yang berikut ini:

```
OPTIONGROUP  False  oracle-ee  12.1 arn:aws:rds:us-east-1:1234567890:og:testoptiongroup
   Test Option Group  testoptiongroup vpc-test
OPTIONS Oracle 12c EM Express   OEM     False   False   5500
VPCSECURITYGROUPMEMBERSHIPS     active  sg-test1
VPCSECURITYGROUPMEMBERSHIPS     active  sg-test2
```

Example

Contoh berikut menambahkan opsi Oracle `NATIVE_NETWORK_ENCRYPTION` ke grup opsi dan menentukan pengaturan opsi. Jika tidak ada pengaturan opsi yang ditentukan, nilai default akan digunakan.

Untuk Linux, macOS, atau Unix:

```
aws rds add-option-to-option-group \
```

```
    --option-group-name testoptiongroup \
    --options '[{"OptionSettings":
[{"Name":"SQLNET.ENCRYPTION_SERVER","Value":"REQUIRED"},
{"Name":"SQLNET.ENCRYPTION_TYPES_SERVER","Value":"AES256,AES192,DES"}],"OptionName":"NATIVE_NET
 \
    --apply-immediately
```

Untuk Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name testoptiongroup ^
    --options "OptionSettings"=[{"Name"="SQLNET.ENCRYPTION_SERVER","Value"="REQUIRED"},
{"Name"="SQLNET.ENCRYPTION_TYPES_SERVER","Value"="AES256\,AES192\,DES"}],"OptionName"="NATIVE_N
 ^
    --apply-immediately
```

Output perintah akan seperti yang berikut ini:

```
...{
    "OptionName": "NATIVE_NETWORK_ENCRYPTION",
    "OptionDescription": "Native Network Encryption",
    "Persistent": false,
    "Permanent": false,
    "OptionSettings": [
        {
            "Name": "SQLNET.ENCRYPTION_TYPES_SERVER",
             "Value": "AES256,AES192,DES",
            "DefaultValue":
 "RC4_256,AES256,AES192,3DES168,RC4_128,AES128,3DES112,RC4_56,DES,RC4_40,DES40",
            "Description": "Specifies list of encryption algorithms in order of
 intended use",
            "ApplyType": "STATIC",
            "DataType": "STRING",
            "AllowedValues":
 "RC4_256,AES256,AES192,3DES168,RC4_128,AES128,3DES112,RC4_56,DES,RC4_40,DES40",
            "IsModifiable": true,
            "IsCollection": true
        },
        {
            "Name": "SQLNET.ENCRYPTION_SERVER",
            "Value": "REQUIRED",
            "DefaultValue": "REQUESTED",
            "Description": "Specifies the desired encryption behavior",
```

```
            "ApplyType": "STATIC",
            "DataType": "STRING",
            "AllowedValues": "ACCEPTED,REJECTED,REQUESTED,REQUIRED",
            "IsModifiable": true,
            "IsCollection": false
        },...
```

API RDS

Untuk menambahkan opsi ke grup opsi menggunakan Amazon RDS API, panggil ModifyOptionGroupoperasi dengan opsi yang ingin Anda tambahkan. Untuk segera mengaktifkan opsi baru pada semua instans DB terkait, sertakan parameter `ApplyImmediately` dan atur ke `true`. Secara default, opsi tersebut diaktifkan untuk setiap instans DB terkait selama periode pemeliharaan berikutnya. Sertakan parameter wajib berikut:

- `OptionGroupName`

Menampilkan daftar opsi dan pengaturan opsi untuk grup opsi

Anda dapat menampilkan daftar opsi dan pengaturan opsi untuk grup opsi.

Konsol

Anda dapat AWS Management Console menggunakan daftar semua opsi dan pengaturan opsi untuk grup opsi.

Untuk menampilkan daftar opsi dan pengaturan opsi untuk grup opsi

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup opsi.
3. Pilih nama grup opsi untuk menampilkan detailnya. Opsi dan pengaturan opsi dalam grup opsi akan ditampilkan daftarnya.

AWS CLI

Untuk membuat daftar opsi dan pengaturan opsi untuk grup opsi, gunakan AWS CLI `describe-option-groups`perintah. Tentukan nama grup opsi yang opsi dan pengaturannya ingin Anda lihat. Jika Anda tidak menentukan nama grup opsi, semua grup opsi akan dideskripsikan.

Example

Contoh berikut menampilkan daftar opsi dan pengaturan opsi untuk semua grup opsi.

```
aws rds describe-option-groups
```

Example

Contoh berikut menampilkan daftar opsi dan pengaturan opsi untuk semua grup opsi bernama `testoptiongroup`.

```
aws rds describe-option-groups --option-group-name testoptiongroup
```

API RDS

Untuk menampilkan daftar opsi dan pengaturan opsi untuk grup opsi, gunakan operasi API RDS `DescribeOptionGroups`. Tentukan nama grup opsi yang opsi dan pengaturannya ingin Anda lihat. Jika Anda tidak menentukan nama grup opsi, semua grup opsi akan dideskripsikan.

Memodifikasi pengaturan opsi

Setelah Anda menambahkan opsi yang memiliki pengaturan opsi yang dapat diubah, Anda dapat mengubah pengaturan ini kapan saja. Jika Anda mengubah opsi atau pengaturan opsi dalam grup opsi, perubahan tersebut diterapkan untuk semua instans DB yang terkait dengan grup opsi tersebut. Untuk informasi selengkapnya tentang pengaturan yang tersedia untuk berbagai opsi, lihat dokumentasi untuk mesin Anda dalam Menggunakan grup opsi.

Perubahan grup opsi harus diterapkan segera dalam dua kasus:

- Saat Anda menambahkan opsi yang menambahkan atau memperbarui nilai port, seperti opsi `OEM`.
- Saat Anda menambahkan atau menghapus grup opsi dengan opsi yang menyertakan nilai port.

Dalam kasus ini, pilih opsi Terapkan Segera pada konsol. Atau Anda dapat menyertakan opsi `--apply-immediately` saat menggunakan AWS CLI atau mengatur parameter `ApplyImmediately` ke `true` saat menggunakan API RDS. Opsi yang tidak menyertakan nilai port dapat segera diterapkan, atau dapat diterapkan selama periode pemeliharaan berikutnya untuk instans DB.

> **Note**
>
> Jika Anda menetapkan grup keamanan sebagai nilai untuk opsi dalam grup opsi, Anda perlu mengelola grup keamanan ini dengan memodifikasi grup opsi. Anda tidak dapat mengubah atau menghapus grup keamanan ini dengan memodifikasi instans DB. Selain itu, grup keamanan tidak muncul di detail instans DB di AWS Management Console atau di output untuk AWS CLI perintah`describe-db-instances`.

Konsol

Anda dapat menggunakan AWS Management Console untuk memodifikasi pengaturan opsi.

Untuk mengubah pengaturan opsi dengan menggunakan konsol

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup opsi.
3. Pilih grup opsi yang opsinya ingin Anda ubah, lalu pilih Ubah opsi.
4. Pada jendela Ubah opsi, dari Opsi Terinstal, pilih opsi yang pengaturannya ingin Anda ubah. Buat perubahan yang Anda inginkan.
5. Untuk segera mengaktifkan opsi setelah Anda menambahkannya, untuk Terapkan Segera, pilih Ya. Jika Anda memilih Tidak (default), opsi diaktifkan untuk setiap instans DB terkait selama periode pemeliharaan berikutnya.
6. Jika pengaturan ini sudah sesuai keinginan Anda, pilih Ubah Opsi.

AWS CLI

Untuk mengubah pengaturan opsi, gunakan AWS CLI `add-option-to-option-group`perintah dengan grup opsi dan opsi yang ingin Anda ubah. Secara default, opsi tersebut diaktifkan untuk setiap instans DB terkait selama periode pemeliharaan berikutnya. Untuk segera menerapkan perubahan pada semua instans DB terkait, sertakan parameter `--apply-immediately`. Untuk mengubah pengaturan opsi, gunakan argumen `--settings`.

Example

Contoh berikut memodifikasi port yang digunakan oleh Oracle Enterprise Manager Database Control (OEM) dalam grup opsi bernama `testoptiongroup` dan segera menerapkan perubahan.

Untuk Linux, macOS, atau Unix:

```
aws rds add-option-to-option-group \
    --option-group-name testoptiongroup \
    --options OptionName=OEM,Port=5432,DBSecurityGroupMemberships=default \
    --apply-immediately
```

Untuk Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name testoptiongroup ^
    --options OptionName=OEM,Port=5432,DBSecurityGroupMemberships=default ^
    --apply-immediately
```

Output perintah akan seperti yang berikut ini:

```
OPTIONGROUP   False  oracle-ee  12.1  arn:aws:rds:us-
east-1:1234567890:og:testoptiongroup   Test Option Group    testoptiongroup
OPTIONS Oracle 12c EM Express   OEM     False   False   5432
DBSECURITYGROUPMEMBERSHIPS   default  authorized
```

Example

Contoh berikut memodifikasi opsi Oracle `NATIVE_NETWORK_ENCRYPTION` dan mengubah pengaturan opsi.

Untuk Linux, macOS, atau Unix:

```
aws rds add-option-to-option-group \
    --option-group-name testoptiongroup \
    --options '[{"OptionSettings":
[{"Name":"SQLNET.ENCRYPTION_SERVER","Value":"REQUIRED"},
{"Name":"SQLNET.ENCRYPTION_TYPES_SERVER","Value":"AES256,AES192,DES,RC4_256"}],"OptionName":"NA
 \
    --apply-immediately
```

Untuk Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name testoptiongroup ^
```

```
    --options "OptionSettings"=[{"Name"="SQLNET.ENCRYPTION_SERVER","Value"="REQUIRED"},
{"Name"="SQLNET.ENCRYPTION_TYPES_SERVER","Value"="AES256\,AES192\,DES
\,RC4_256"}],"OptionName"="NATIVE_NETWORK_ENCRYPTION" ^
    --apply-immediately
```

Output perintah akan seperti yang berikut ini:

```
OPTIONGROUP   False  oracle-ee  12.1  arn:aws:rds:us-
east-1:1234567890:og:testoptiongroup   Test Option Group    testoptiongroup

OPTIONS Oracle Advanced Security - Native Network Encryption
 NATIVE_NETWORK_ENCRYPTION       False   False
OPTIONSETTINGS
 RC4_256,AES256,AES192,3DES168,RC4_128,AES128,3DES112,RC4_56,DES,RC4_40,DES40 STATIC
 STRING
   RC4_256,AES256,AES192,3DES168,RC4_128,AES128,3DES112,RC4_56,DES,RC4_40,DES40
 Specifies list of encryption algorithms in order of intended use
   True     True    SQLNET.ENCRYPTION_TYPES_SERVER  AES256,AES192,DES,RC4_256
OPTIONSETTINGS  ACCEPTED,REJECTED,REQUESTED,REQUIRED    STATIC  STRING  REQUESTED
 Specifies the desired encryption behavior  False   True  SQLNET.ENCRYPTION_SERVER
 REQUIRED
OPTIONSETTINGS  SHA1,MD5   STATIC  STRING  SHA1,MD5    Specifies list of
 checksumming algorithms in order of intended use      True    True
 SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER  SHA1,MD5
OPTIONSETTINGS  ACCEPTED,REJECTED,REQUESTED,REQUIRED  STATIC  STRING
 REQUESTED     Specifies the desired data integrity behavior   False   True
 SQLNET.CRYPTO_CHECKSUM_SERVER  REQUESTED
```

API RDS

Untuk mengubah pengaturan opsi, gunakan perintah API Amazon RDS `ModifyOptionGroup` dengan grup opsi dan opsi yang ingin Anda ubah. Secara default, opsi tersebut diaktifkan untuk setiap instans DB terkait selama periode pemeliharaan berikutnya. Untuk segera menerapkan perubahan pada semua instans DB terkait, sertakan parameter `ApplyImmediately` dan tetapkan ke `true`.

Menghapus opsi dari grup opsi

Beberapa opsi dapat dihapus dari grup opsi, dan beberapa tidak dapat dihapus. Opsi persisten tidak dapat dihapus dari grup opsi hingga semua instans DB yang dikaitkan dengan grup opsi tersebut dibatalkan pengaitannya. Opsi permanen tidak akan dapat dihapus dari grup opsi. Untuk informasi

selengkapnya tentang opsi yang dapat dihapus, lihat dokumentasi untuk mesin spesifik Anda yang tercantum dalam Menggunakan grup opsi.

Jika Anda menghapus semua opsi dari grup opsi, Amazon RDS tidak menghapus grup opsi. Instans DB yang dikaitkan dengan grup opsi kosong akan terus dikaitkan dengannya; instans ini tidak akan memiliki opsi aktif. Alternatifnya, untuk menghapus semua opsi dari instans DB, Anda dapat mengaitkan instans DB dengan grup opsi default (kosong).

Konsol

Anda dapat menggunakan opsi AWS Management Console untuk menghapus opsi dari grup opsi.

Untuk menghapus opsi dari grup opsi menggunakan konsol

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup opsi.
3. Pilih grup opsi yang opsinya ingin Anda hapus, lalu pilih Hapus opsi.
4. Di jendela Hapus opsi, lakukan hal berikut:
   - Pilih kotak centang untuk opsi yang ingin Anda hapus.
   - Agar penghapusan berlaku segera setelah Anda melakukannya, untuk Terapkan segera, pilih Ya. Jika Anda memilih Tidak (default), opsi dihapus untuk setiap instans DB terkait selama periode pemeliharaan berikutnya.

5. Jika pengaturan ini sudah sesuai keinginan Anda, pilih Ya, Hapus.

AWS CLI

Untuk menghapus opsi dari grup opsi, gunakan AWS CLI `remove-option-from-option-group`perintah dengan opsi yang ingin Anda hapus. Secara default, opsi tersebut dihapus dari setiap instans DB terkait selama periode pemeliharaan berikutnya. Untuk segera menerapkan perubahan, sertakan parameter `--apply-immediately`.

Example

Contoh berikut menghapus opsi Oracle Enterprise Manager Database Control (OEM) dari grup opsi bernama `testoptiongroup` dan segera menerapkan perubahan.

Untuk Linux, macOS, atau Unix:

```
aws rds remove-option-from-option-group \
    --option-group-name testoptiongroup \
    --options OEM \
    --apply-immediately
```

Untuk Windows:

```
aws rds remove-option-from-option-group ^
    --option-group-name testoptiongroup ^
    --options OEM ^
    --apply-immediately
```

Output perintah akan seperti yang berikut ini:

```
OPTIONGROUP    testoptiongroup oracle-ee   12.1    Test option group
```

API RDS

Untuk menghapus opsi dari grup opsi, gunakan tindakan API Amazon RDS `ModifyOptionGroup`. Secara default, opsi tersebut dihapus dari setiap instans DB terkait selama periode pemeliharaan berikutnya. Untuk segera menerapkan perubahan, sertakan parameter `ApplyImmediately` dan tetapkan ke `true`.

Sertakan parameter berikut:

- `OptionGroupName`
- `OptionsToRemove.OptionName`

Menghapus grup opsi

Anda dapat menghapus grup opsi hanya jika memenuhi kriteria berikut:

- Ini tidak terkait dengan sumber daya Amazon RDS apa pun. Grup opsi dapat dikaitkan dengan instans DB, snapshot DB manual, atau snapshot DB otomatis.
- Ini bukan grup opsi default.

Untuk mengidentifikasi grup opsi yang digunakan oleh instance DB dan snapshot DB Anda, Anda dapat menggunakan perintah CLI berikut:

```
aws rds describe-db-instances \
    --query 'DBInstances[*].
[DBInstanceIdentifier,OptionGroupMemberships[].OptionGroupName]'

aws rds describe-db-snapshots | jq -r '.DBSnapshots[] | "\(.DBInstanceIdentifier),
\(.OptionGroupName)"' | sort | uniq
```

Jika Anda mencoba menghapus grup opsi yang terkait dengan sumber daya RDS, kesalahan seperti berikut ini akan ditampilkan.

```
An error occurred (InvalidOptionGroupStateFault) when calling the DeleteOptionGroup
 operation: The option group 'optionGroupName' cannot be deleted because it is in use.
```

Untuk menemukan sumber daya Amazon RDS yang dikaitkan dengan grup opsi

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup opsi.
3. Pilih nama grup opsi untuk menampilkan detailnya.
4. Periksa bagian Instans dan Snapshot yang Dikaitkan untuk sumber daya Amazon RDS terkait.

Jika instans DB dikaitkan dengan grup opsi, ubah instans DB untuk menggunakan grup opsi yang berbeda. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Jika snapshot DB manual dikaitkan dengan grup opsi, ubah snapshot DB untuk menggunakan grup opsi yang berbeda. Anda dapat melakukannya dengan menggunakan AWS CLI `modify-db-snapshot`perintah.

> **Note**
>
> Anda tidak dapat mengubah grup opsi dari snapshot DB otomatis.

Konsol

Salah satu cara untuk menghapus grup opsi adalah dengan menggunakan AWS Management Console.

Untuk menghapus grup opsi baru menggunakan konsol

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup opsi.
3. Pilih grup opsi.
4. Pilih Hapus grup.
5. Di halaman konfirmasi, pilih Hapus untuk menyelesaikan penghapusan grup opsi, atau pilih Batal untuk membatalkan penghapusan.

AWS CLI

Untuk menghapus grup opsi, gunakan AWS CLI `delete-option-group`perintah dengan parameter yang diperlukan berikut.

- `--option-group-name`

Example

Contoh berikut akan menghapus grup opsi bernama `testoptiongroup`.

Untuk Linux, macOS, atau Unix:

```
aws rds delete-option-group \
    --option-group-name testoptiongroup
```

Untuk Windows:

```
aws rds delete-option-group ^
    --option-group-name testoptiongroup
```

API RDS

Untuk menghapus grup opsi, panggil operasi API Amazon RDS `DeleteOptionGroup`. Sertakan parameter berikut:

- `OptionGroupName`

Bekerja dengan grup parameter

Parameter basis data menentukan konfigurasi basis data. Misalnya, parameter basis data dapat menentukan jumlah sumber daya, seperti memori, untuk dialokasikan ke basis data.

Anda mengelola konfigurasi basis data dengan mengaitkan instans DB dan klaster DB Multi-AZ dengan grup parameter. Amazon RDS menentukan grup parameter dengan pengaturan default. Anda dapat juga menentukan grup parameter Anda sendiri dengan pengaturan yang disesuaikan.

> **Note**
>
> Beberapa mesin DB menawarkan fitur tambahan yang dapat Anda tambahkan ke basis data sebagai opsi dalam grup opsi. Untuk informasi tentang grup opsi, lihat Menggunakan grup opsi.

Topik

Ikhtisar grup parameter

Grup parameter DB bertindak sebagai kontainer untuk nilai konfigurasi mesin yang diterapkan pada satu atau beberapa instans DB.

Grup parameter klaster DB hanya berlaku untuk klaster DB Multi-AZ. Pada klaster DB Multi-AZ, pengaturan di grup parameter klaster DB berlaku untuk semua instans DB di kluster. Grup parameter DB default untuk mesin DB dan versi mesin DB digunakan untuk setiap instans DB di klaster DB.

Topik

- Parameter set karakter
- Nilai parameter dan parameter yang didukung

Grup parameter kustom dan default

Jika Anda membuat instans DB tanpa menentukan grup parameter DB, instans DB akan menggunakan grup parameter DB default. Demikian pula, jika Anda membuat klaster DB Multi-AZ tanpa menentukan grup parameter klaster DB, klaster DB tersebut akan menggunakan grup parameter klaster DB default. Setiap grup parameter default berisi default mesin basis data dan default sistem Amazon RDS berdasarkan mesin, kelas komputasi, dan penyimpanan yang dialokasikan dari instans.

Anda tidak dapat memodifikasi pengaturan parameter dari grup parameter default. Anda dapat melakukan hal berikut sebagai gantinya:

1. Buat grup parameter baru.
2. Ubah pengaturan parameter yang Anda inginkan. Tidak semua parameter mesin DB dalam grup parameter memenuhi syarat untuk dimodifikasi.
3. Ubah instans DB atau cluster DB Anda untuk mengaitkan grup parameter baru.

   Saat Anda mengaitkan grup parameter DB baru dengan instance DB, asosiasi segera terjadi. Untuk informasi tentang cara memodifikasi instans DB, lihat Memodifikasi instans DB Amazon RDS. Untuk mengetahui informasi tentang cara memodifikasi klaster DB multi-AZ, lihat. Mengubah klaster basis data Multi-AZ.

   > **Note**
   >
   > Jika Anda telah memodifikasi instans DB Anda untuk menggunakan grup parameter kustom, dan Anda memulai instans DB, RDS secara otomatis me-reboot instans DB sebagai bagian dari proses startup.

RDS menerapkan parameter statis dan dinamis yang dimodifikasi dalam grup parameter yang baru terkait hanya setelah instance DB di-boot ulang. Namun, jika Anda memodifikasi parameter dinamis dalam grup parameter DB setelah Anda mengaitkannya dengan instans DB, perubahan ini akan diterapkan segera tanpa reboot. Untuk informasi selengkapnya tentang cara mengubah grup parameter DB, lihat Memodifikasi instans DB Amazon RDS.

Jika Anda memperbarui parameter dalam grup parameter DB, perubahan berlaku untuk semua instans DB yang terkait dengan grup parameter tersebut. Demikian pula, jika Anda memperbarui parameter dalam grup parameter klaster DB Multi-AZ, perubahan berlaku untuk semua klaster DB Aurora yang terkait dengan grup parameter klaster DB tersebut.

Jika Anda tidak ingin membuat grup parameter dari awal, Anda dapat menyalin grup parameter yang ada dengan AWS CLI copy-db-parameter-groupperintah atau perintah copy-db-cluster-parameter-group. Anda mungkin mendapati bahwa menyalin grup parameter berguna dalam beberapa kasus. Misalnya, Anda mungkin ingin menyertakan sebagian besar parameter dan nilai kustom grup parameter DB yang ada dalam grup parameter DB baru.

Parameter instans DB statis dan dinamis

Parameter instans DB bersifat statis atau dinamis. Perbedaannya terletak dalam hal berikut:

- Saat Anda mengubah parameter statis dan menyimpan grup parameter DB, perubahan parameter akan diterapkan setelah Anda me-reboot instans DB terkait secara manual. Untuk parameter statis, konsol selalu menggunakan `pending-reboot` untuk `ApplyMethod`.
- Saat Anda mengubah parameter dinamis, perubahan parameter akan langsung diterapkan secara default, tanpa harus di-reboot. Saat Anda menggunakan AWS Management Console untuk mengubah nilai parameter instans DB, selalu digunakan `immediate` untuk parameter dinamis `ApplyMethod` for. Untuk menunda perubahan parameter hingga setelah Anda me-reboot instans DB terkait, gunakan AWS CLI atau RDS API. Atur `ApplyMethod` ke `pending-reboot` untuk perubahan parameter.

> **Note**
>
> Menggunakan `pending-reboot` dengan parameter dinamis di AWS CLI atau RDS API pada RDS untuk instance SQL Server DB menghasilkan kesalahan. Gunakan `apply-immediately` di RDS for SQL Server.

Untuk informasi selengkapnya tentang menggunakan AWS CLI untuk mengubah nilai parameter, lihat modify-db-parameter-group. Untuk informasi selengkapnya tentang penggunaan RDS API untuk mengubah nilai parameter, lihat ParameterGroupModifyDB.

Jika instans DB tidak menggunakan perubahan terbaru pada grup parameter DB terkait, konsol menunjukkan status reboot-tertunda untuk grup parameter DB. Status ini tidak menyebabkan reboot

otomatis selama periode pemeliharaan berikutnya. Untuk menerapkan perubahan parameter terbaru pada instans DB tersebut, reboot instans DB secara manual.

Parameter klaster DB statis dan dinamis

Parameter klaster DB bersifat statis atau dinamis. Perbedaannya terletak dalam hal berikut:

- Saat Anda mengubah parameter statis dan menyimpan grup parameter DB klaster, perubahan parameter akan diterapkan setelah Anda me-reboot klaster DB terkait secara manual. Untuk parameter statis, konsol selalu menggunakan `pending-reboot` untuk `ApplyMethod`.
- Saat Anda mengubah parameter dinamis, perubahan parameter akan langsung diterapkan secara default, tanpa harus di-reboot. Saat Anda menggunakan AWS Management Console untuk mengubah nilai parameter cluster DB, selalu digunakan `immediate` untuk parameter dinamis `ApplyMethod` for. Untuk menunda perubahan parameter hingga cluster DB terkait di-boot ulang, gunakan AWS CLI atau RDS API. Atur `ApplyMethod` ke `pending-reboot` untuk perubahan parameter.

Untuk informasi selengkapnya tentang menggunakan AWS CLI untuk mengubah nilai parameter, lihat modify-db-cluster-parameter-group. Untuk informasi selengkapnya tentang penggunaan RDS API untuk mengubah nilai parameter, lihat ClusterParameterGroupModifyDB.

Parameter set karakter

Sebelum Anda membuat klaster DB Multi-AZ atau instans DB, atur parameter apa pun yang terkait dengan set karakter atau kolasi basis data di grup parameter Anda. Lakukan juga sebelum Anda membuat basis data di dalamnya. Dengan cara ini, Anda memastikan basis data default dan basis data baru menggunakan nilai kolasi dan set karakter yang Anda tentukan. Jika Anda mengubah parameter kolasi atau set karakter, perubahan parameter tidak diterapkan ke basis data yang sudah ada.

Untuk beberapa mesin DB, Anda dapat mengubah nilai kolasi atau set karakter untuk basis data yang sudah ada menggunakan perintah `ALTER DATABASE`, contohnya:

```
ALTER DATABASE database_name CHARACTER SET character_set_name COLLATE collation;
```

Untuk informasi selengkapnya tentang cara mengubah nilai set kolasi atau set karakter untuk basis data, periksa dokumentasi untuk mesin DB Anda.

Nilai parameter dan parameter yang didukung

Untuk menentukan parameter yang didukung untuk mesin DB, lihat parameter di grup parameter DB dan grup parameter klaster DB yang digunakan oleh instans DB atau klaster DB. Untuk informasi selengkapnya, lihat Melihat nilai parameter untuk grup parameter DB dan Melihat nilai parameter untuk grup parameter klaster DB.

Anda juga dapat menentukan parameter bilangan bulat dan Boolean menggunakan ekspresi, rumus, dan fungsi. Fungsi dapat mencakup ekspresi log matematis. Namun, tidak semua parameter mendukung ekspresi, rumus, dan fungsi untuk nilai parameter. Untuk informasi selengkapnya, lihat Menentukan parameter DB.

Pengaturan parameter yang tidak tepat dalam grup parameter dapat menimbulkan dampak buruk yang tidak diinginkan, termasuk penurunan performa dan ketidakstabilan sistem. Selalu berhati-hatilah saat memodifikasi parameter basis data, dan cadangkan data Anda sebelum memodifikasi grup parameter. Cobalah menerapkan perubahan pengaturan grup parameter pada instans DB atau klaster DB uji coba sebelum menerapkan perubahan grup parameter tersebut ke instans DB atau klaster DB produksi.

Bekerja dengan grup parameter DB dalam instance DB

Instans DB menggunakan grup parameter DB. Bagian berikut menjelaskan cara mengonfigurasi dan mengelola grup parameter instans DB.

Topik

- Membuat grup parameter DB
- Mengaitkan grup parameter DB dengan instans DB
- Memodifikasi parameter dalam grup parameter DB
- Mengatur ulang parameter dalam grup parameter DB ke nilai defaultnya
- Menyalin grup parameter DB
- Mencantumkan grup parameter DB
- Melihat nilai parameter untuk grup parameter DB
- Menghapus grup parameter DB

Membuat grup parameter DB

Anda dapat membuat grup parameter DB baru menggunakan AWS Management Console, AWS CLI, atau RDS API.

Batasan berikut berlaku untuk nama grup parameter DB:

- Nama harus berisi 1 sampai 255 huruf, angka, atau tanda hubung.

  Nama grup parameter default boleh menyertakan titik, seperti `default.mysql8.0`. Namun, nama grup parameter kustom tidak boleh menyertakan titik.
- Karakter pertamanya harus berupa huruf.
- Nama tidak boleh diakhiri dengan tanda hubung atau berisi dua tanda hubung berturut-turut.

Konsol

Untuk membuat grup parameter DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup parameter.
3. Pilih Buat grup parameter.

   Jendela Buat grup parameter akan muncul.
4. Dalam daftar Kelompok grup parameter, pilih kelompok grup parameter DB.
5. Dalam daftar Jenis, jika relevan, pilih Grup Parameter DB.
6. Di kotak Nama grup, masukkan nama grup parameter DB yang baru.
7. Di kotak Deskripsi, masukkan deskripsi untuk grup parameter DB yang baru.
8. Pilih Buat.

AWS CLI

Untuk membuat grup parameter DB, gunakan AWS CLI `create-db-parameter-group`perintah. Contoh berikut membuat grup parameter DB yang bernama mydbparametergroup untuk MySQL versi 8.0 dengan deskripsi "Grup parameter baru saya".

Sertakan parameter wajib berikut:

- `--db-parameter-group-name`
- `--db-parameter-group-family`
- `--description`

Untuk mencantumkan semua kelompok grup parameter yang tersedia, gunakan perintah berikut:

```
aws rds describe-db-engine-versions --query "DBEngineVersions[].DBParameterGroupFamily"
```

> **Note**
>
> Output berisi duplikat.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-parameter-group \
    --db-parameter-group-name mydbparametergroup \
    --db-parameter-group-family MySQL8.0 \
    --description "My new parameter group"
```

Untuk Windows:

```
aws rds create-db-parameter-group ^
    --db-parameter-group-name mydbparametergroup ^
    --db-parameter-group-family MySQL8.0 ^
    --description "My new parameter group"
```

Perintah ini menghasilkan output yang serupa dengan yang berikut:

```
DBPARAMETERGROUP  mydbparametergroup  mysql8.0  My new parameter group
```

API RDS

Untuk membuat grup parameter DB, gunakan operasi `CreateDBParameterGroup` API RDS.

Sertakan parameter wajib berikut:

- `DBParameterGroupName`

- `DBParameterGroupFamily`
- `Description`

Mengaitkan grup parameter DB dengan instans DB

Anda dapat membuat grup parameter DB Anda sendiri dengan pengaturan yang disesuaikan. Anda dapat mengaitkan grup parameter DB dengan instans DB menggunakan AWS CLI,, atau RDS API. AWS Management Console Anda dapat melakukannya saat membuat atau memodifikasi instans DB.

Untuk informasi tentang cara membuat grup parameter DB, lihat Membuat grup parameter DB. Untuk informasi selengkapnya tentang cara membuat instans DB, lihat Membuat instans DB Amazon RDS. Untuk informasi tentang cara memodifikasi instans DB, lihat Memodifikasi instans DB Amazon RDS.

> **Note**
>
> Ketika Anda mengaitkan grup parameter DB baru dengan instans DB, parameter statis dan dinamis yang dimodifikasi diterapkan hanya setelah instans DB di-reboot. Namun, jika Anda memodifikasi parameter dinamis dalam grup parameter DB setelah Anda mengaitkannya dengan instans DB, perubahan ini diterapkan segera tanpa reboot.

Konsol

Untuk mengaitkan grup parameter DB dengan instans DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data, lalu pilih instans DB yang ingin Anda modifikasi.
3. Pilih Modifikasi. Halaman Modifikasi instans DB akan muncul.
4. Ubah pengaturan Grup parameter DB.
5. Pilih Lanjutkan dan periksa ringkasan modifikasi.
6. (Opsional) Pilih Terapkan langsung untuk langsung menerapkan perubahan. Memilih opsi ini dapat menyebabkan penonaktifan dalam beberapa kasus. Untuk informasi selengkapnya, lihat Menggunakan pengaturan Terapkan Segera.
7. Di halaman konfirmasi, tinjau perubahan Anda. Jika sudah benar, pilih Modifikasi instans DB untuk menyimpan perubahan Anda.

Atau pilih Kembali untuk mengedit perubahan atau Batal untuk membatalkan perubahan.

AWS CLI

Untuk mengaitkan grup parameter DB dengan instans DB, gunakan AWS CLI `modify-db-instance`perintah dengan opsi berikut:

- `--db-instance-identifier`
- `--db-parameter-group-name`

Contoh berikut mengaitkan Grup parameter DB `mydbpg` dengan instans DB `database-1`. Perubahan langsung diterapkan dengan menggunakan `--apply-immediately`. Gunakan `--no-apply-immediately` untuk menerapkan perubahan selama masa pemeliharaan berikutnya. Untuk informasi selengkapnya, lihat Menggunakan pengaturan Terapkan Segera.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier database-1 \
    --db-parameter-group-name mydbpg \
    --apply-immediately
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier database-1 ^
    --db-parameter-group-name mydbpg ^
    --apply-immediately
```

API RDS

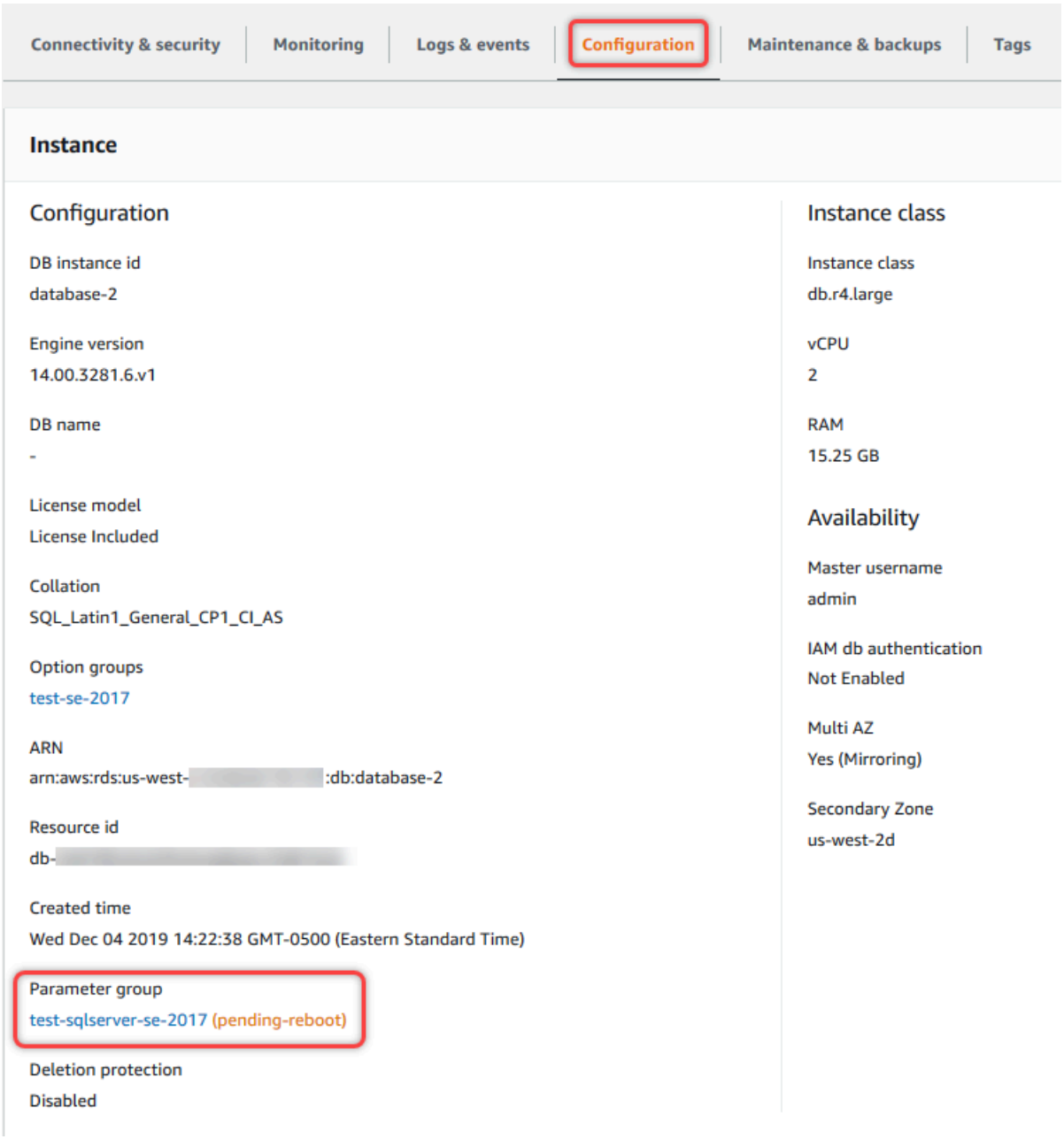
Untuk mengaitkan grup parameter DB dengan instans DB, gunakan operasi `ModifyDBInstance` API RDS dengan parameter berikut:

- `DBInstanceName`
- `DBParameterGroupName`

Memodifikasi parameter dalam grup parameter DB

Anda dapat memodifikasi nilai parameter dalam grup parameter DB buatan pelanggan; Anda tidak dapat mengubah nilai parameter dalam grup parameter DB default. Perubahan pada parameter dalam grup parameter DB buatan pelanggan diterapkan ke semua instans DB yang dikaitkan dengan grup parameter DB.

Perubahan pada beberapa parameter diterapkan langsung ke instans DB tanpa reboot. Perubahan pada parameter lain diterapkan hanya setelah instans DB di-reboot. Konsol RDS menampilkan status grup parameter DB yang terkait dengan instans DB pada tab Konfigurasi. Misalnya, instans DB tidak menggunakan perubahan terbaru pada grup parameter DB terkait. Jika demikian, konsol RDS menampilkan grup parameter DB dengan status Reboot tertunda. Untuk menerapkan perubahan parameter terbaru pada instans DB tersebut, reboot instans DB secara manual.



Konsol

Untuk memodifikasi grup parameter DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup parameter.
3. Dalam daftar, pilih grup parameter yang ingin Anda modifikasi.
4. Untuk Tindakan grup parameter, pilih Edit.

5. Ubah nilai parameter yang ingin Anda modifikasi. Anda dapat menggulir parameter menggunakan tombol panah di bagian kanan atas kotak dialog.

   Anda tidak dapat mengubah nilai dalam grup parameter default.

6. Pilih Simpan perubahan.

AWS CLI

Untuk memodifikasi grup parameter DB, gunakan AWS CLI `modify-db-parameter-group`perintah dengan opsi yang diperlukan berikut:

- `--db-parameter-group-name`
- `--parameters`

Contoh berikut memodifikasi nilai `max_connections` dan `max_allowed_packet` dalam grup parameter DB yang bernama mydbparametergroup.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name mydbparametergroup \
    --parameters
 "ParameterName=max_connections,ParameterValue=250,ApplyMethod=immediate" \

 "ParameterName=max_allowed_packet,ParameterValue=1024,ApplyMethod=immediate"
```

Untuk Windows:

```
aws rds modify-db-parameter-group ^
    --db-parameter-group-name mydbparametergroup ^
    --parameters
 "ParameterName=max_connections,ParameterValue=250,ApplyMethod=immediate" ^

 "ParameterName=max_allowed_packet,ParameterValue=1024,ApplyMethod=immediate"
```

Perintah menghasilkan output seperti berikut:

```
DBPARAMETERGROUP  mydbparametergroup
```

API RDS

Untuk memodifikasi grup parameter DB, gunakan operasi `ModifyDBParameterGroup` API RDS dengan parameter wajib berikut:

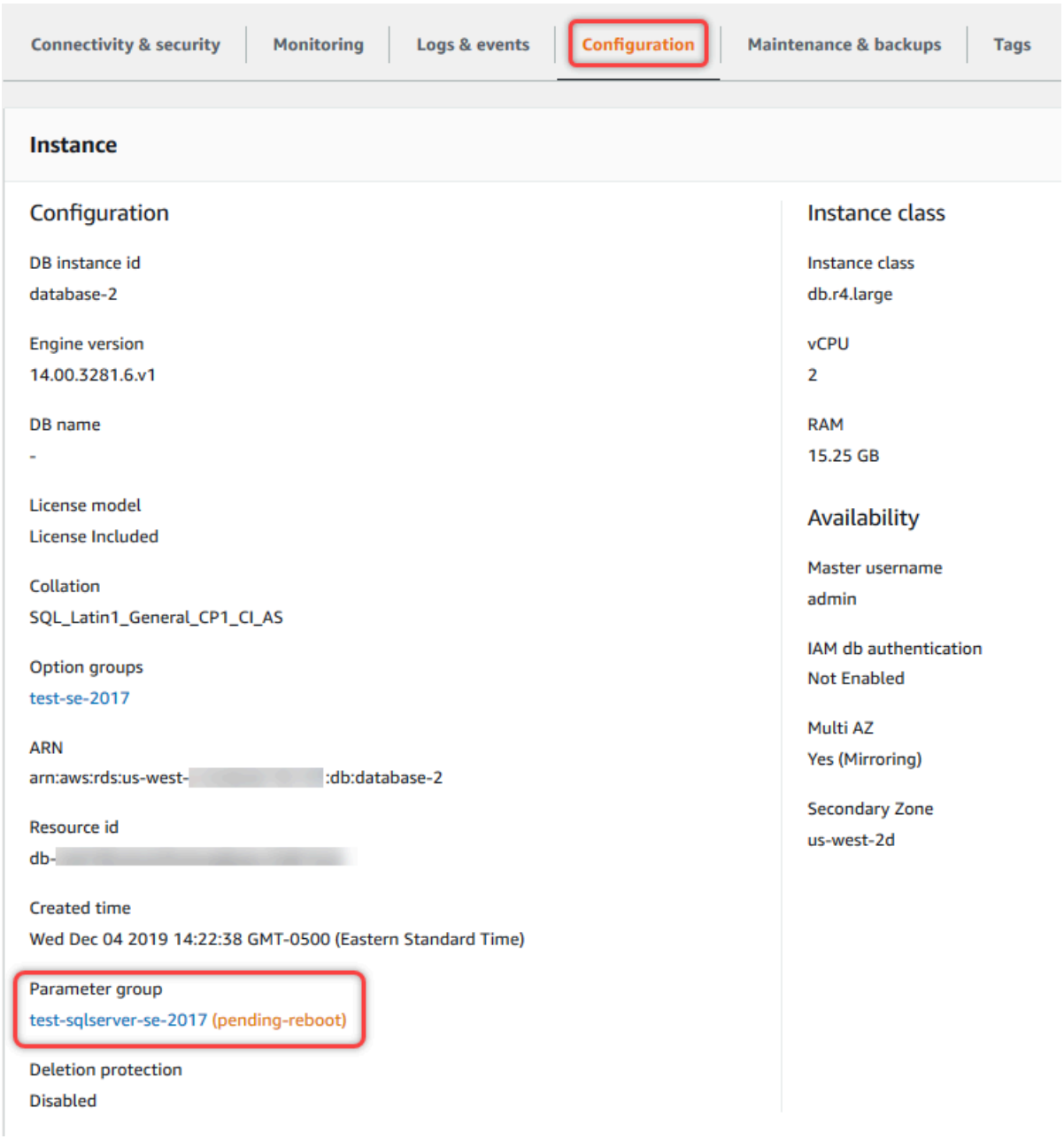
- `DBParameterGroupName`
- `Parameters`

Mengatur ulang parameter dalam grup parameter DB ke nilai defaultnya

Anda dapat mengatur ulang nilai parameter dalam grup parameter DB buatan pelanggan ke nilai defaultnya. Perubahan pada parameter dalam grup parameter DB buatan pelanggan diterapkan ke semua instans DB yang dikaitkan dengan grup parameter DB.

Saat Anda menggunakan konsol, Anda dapat mengatur ulang parameter tertentu ke nilai defaultnya. Namun, Anda tidak dapat dengan mudah mengatur ulang semua parameter dalam grup parameter DB sekaligus. Saat Anda menggunakan AWS CLI atau RDS API, Anda dapat mengatur ulang parameter tertentu ke nilai defaultnya. Anda juga dapat mengatur ulang semua parameter dalam grup parameter DB sekaligus.

Perubahan pada beberapa parameter diterapkan langsung ke instans DB tanpa reboot. Perubahan pada parameter lain diterapkan hanya setelah instans DB di-reboot. Konsol RDS menampilkan status grup parameter DB yang terkait dengan instans DB pada tab Konfigurasi. Misalnya, instans DB tidak menggunakan perubahan terbaru pada grup parameter DB terkait. Jika demikian, konsol RDS menampilkan grup parameter DB dengan status Reboot tertunda. Untuk menerapkan perubahan parameter terbaru pada instans DB tersebut, reboot instans DB secara manual.



> **Note**
>
> Dalam grup parameter DB default, parameter selalu ditetapkan ke nilai defaultnya.

Konsol

Untuk mengatur ulang parameter dalam grup parameter DB ke nilai defaultnya

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup parameter.
3. Dalam daftar, pilih grup parameter.
4. Untuk Tindakan grup parameter, pilih Edit.
5. Pilih parameter yang ingin Anda atur ulang ke nilai defaultnya. Anda dapat menggulir parameter menggunakan tombol panah di bagian kanan atas kotak dialog.

   Anda tidak dapat mengatur ulang nilai dalam grup parameter default.
6. Pilih Atur ulang lalu konfirmasikan dengan memilih Atur ulang parameter.

AWS CLI

Untuk mengatur ulang beberapa atau semua parameter dalam grup parameter DB, gunakan AWS CLI `reset-db-parameter-group`perintah dengan opsi wajib berikut:`--db-parameter-group-name`.

Untuk mengatur ulang semua parameter dalam grup parameter DB, tentukan opsi `--reset-all-parameters`. Untuk mengatur ulang parameter tertentu, tentukan opsi `--parameters`.

Contoh berikut mengatur ulang semua parameter dalam grup parameter DB yang bernama mydbparametergroup ke nilai defaultnya.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds reset-db-parameter-group \
    --db-parameter-group-name mydbparametergroup \
    --reset-all-parameters
```

Untuk Windows:

```
aws rds reset-db-parameter-group ^
    --db-parameter-group-name mydbparametergroup ^
```

```
    --reset-all-parameters
```

Contoh berikut mengatur ulang opsi `max_connections` dan `max_allowed_packet` ke nilai defaultnya dalam grup parameter DB yang bernama mydbparametergroup.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds reset-db-parameter-group \
    --db-parameter-group-name mydbparametergroup \
    --parameters "ParameterName=max_connections,ApplyMethod=immediate" \
                 "ParameterName=max_allowed_packet,ApplyMethod=immediate"
```

Untuk Windows:

```
aws rds reset-db-parameter-group ^
    --db-parameter-group-name mydbparametergroup ^
    --parameters "ParameterName=max_connections,ApplyMethod=immediate" ^
                 "ParameterName=max_allowed_packet,ApplyMethod=immediate"
```

Perintah menghasilkan output seperti berikut:

```
DBParameterGroupName  mydbparametergroup
```

API RDS

Untuk mengatur ulang parameter dalam grup parameter DB ke nilai defaultnya, gunakan perintah `ResetDBParameterGroup` API RDS dengan parameter wajib berikut: `DBParameterGroupName`.

Untuk mengatur ulang semua parameter dalam grup parameter DB, atur parameter `ResetAllParameters` ke `true`. Untuk mengatur ulang parameter tertentu, tentukan parameter `Parameters`.

Menyalin grup parameter DB

Anda dapat menyalin grup parameter DB kustom yang Anda buat. Menyalin grup parameter bisa menjadi solusi yang mudah. Contohnya adalah saat Anda telah membuat grup parameter DB dan ingin menyertakan sebagian besar parameter dan nilai kustomnya dalam grup parameter DB yang baru. Anda dapat menyalin grup parameter DB dengan menggunakan file AWS Management

Console. Anda juga dapat menggunakan AWS CLI copy-db-parameter-groupperintah atau operasi RDS API CopyDB ParameterGroup.

Setelah Anda menyalin grup parameter DB, tunggu minimal 5 menit sebelum membuat instans DB pertama Anda yang menggunakan grup parameter DB tersebut sebagai grup parameter default. Cara ini memungkinkan Amazon RDS menyelesaikan sepenuhnya tindakan penyalinan sebelum grup parameter digunakan. Tindakan ini harus dilakukan terutama untuk parameter yang sangat penting saat membuat basis data default untuk instans DB. Contohnya adalah kumpulan karakter untuk basis data default yang ditentukan oleh parameter `character_set_database`. Gunakan opsi Parameter Grup konsol Amazon RDS atau describe-db-parametersperintah untuk memverifikasi bahwa grup parameter DB Anda dibuat.

> **Note**
>
> Anda tidak dapat menyalin grup parameter default. Namun, Anda dapat membuat grup parameter baru yang didasarkan pada grup parameter default.
> Anda tidak dapat menyalin grup parameter DB ke grup parameter yang berbeda Akun AWS atau Wilayah AWS.

Konsol

Untuk menyalin grup parameter DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup parameter.
3. Dalam daftar, pilih grup parameter kustom yang ingin Anda salin.
4. Untuk Tindakan grup parameter, pilih Salin.
5. Di Pengidentifikasi grup parameter DB baru, masukkan nama untuk grup parameter baru.
6. Di Deskripsi, masukkan deskripsi untuk grup parameter DB yang baru.
7. Pilih Salin.

AWS CLI

Untuk menyalin grup parameter DB, gunakan AWS CLI `copy-db-parameter-group`perintah dengan opsi yang diperlukan berikut:

- `--source-db-parameter-group-identifier`
- `--target-db-parameter-group-identifier`
- `--target-db-parameter-group-description`

Contoh berikut membuat grup parameter DB baru yang bernama `mygroup2` yang merupakan salinan dari grup parameter DB `mygroup1`.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds copy-db-parameter-group \
    --source-db-parameter-group-identifier mygroup1 \
    --target-db-parameter-group-identifier mygroup2 \
    --target-db-parameter-group-description "DB parameter group 2"
```

Untuk Windows:

```
aws rds copy-db-parameter-group ^
    --source-db-parameter-group-identifier mygroup1 ^
    --target-db-parameter-group-identifier mygroup2 ^
    --target-db-parameter-group-description "DB parameter group 2"
```

API RDS

Untuk menyalin grup parameter DB, gunakan operasi `CopyDBParameterGroup` API RDS dengan parameter wajib berikut:

- `SourceDBParameterGroupIdentifier`
- `TargetDBParameterGroupIdentifier`
- `TargetDBParameterGroupDescription`

Mencantumkan grup parameter DB

Anda dapat mencantumkan grup parameter DB yang telah Anda buat untuk AWS akun Anda.

> **Note**
>
> Grup parameter default secara otomatis dibuat dari template parameter default ketika Anda membuat instans DB untuk mesin dan versi DB tertentu. Grup parameter default ini berisi pengaturan parameter pilihan dan tidak dapat dimodifikasi. Saat membuat grup parameter kustom, Anda dapat memodifikasi pengaturan parameter.

Konsol

Untuk mencantumkan semua grup parameter DB untuk AWS akun

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup parameter.

   Grup parameter DB akan muncul dalam daftar.

AWS CLI

Untuk membuat daftar semua grup parameter DB untuk AWS akun, gunakan AWS CLI `describe-db-parameter-groups`perintah.

Example

Contoh berikut mencantumkan semua grup parameter DB yang tersedia untuk akun AWS .

```
aws rds describe-db-parameter-groups
```

Perintah memberikan respons seperti berikut:

```
DBPARAMETERGROUP  default.mysql8.0     mysql8.0  Default parameter group for MySQL8.0
DBPARAMETERGROUP  mydbparametergroup   mysql8.0  My new parameter group
```

Contoh berikut menjelaskan grup parameter mydbparamgroup1.

Untuk Linux, macOS, atau Unix:

```
aws rds describe-db-parameter-groups \
    --db-parameter-group-name mydbparamgroup1
```

Untuk Windows:

```
aws rds describe-db-parameter-groups ^
    --db-parameter-group-name mydbparamgroup1
```

Perintah memberikan respons seperti berikut:

```
DBPARAMETERGROUP  mydbparametergroup1  mysql8.0  My new parameter group
```

API RDS

Untuk mencantumkan semua grup parameter DB untuk AWS akun, gunakan `DescribeDBParameterGroups`operasi RDS API.

Melihat nilai parameter untuk grup parameter DB

Anda dapat memperoleh daftar semua parameter dalam grup parameter DB beserta nilainya.

Konsol

Untuk melihat nilai parameter untuk grup parameter DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup parameter.

   Grup parameter DB akan muncul dalam daftar.
3. Pilih nama grup parameter untuk melihat daftar parameternya.

AWS CLI

Untuk melihat nilai parameter untuk grup parameter DB, gunakan AWS CLI `describe-db-parameters`perintah dengan parameter yang diperlukan berikut.

- `--db-parameter-group-name`

Example

Contoh berikut mencantumkan parameter dan nilai parameter untuk grup parameter DB yang bernama mydbparametergroup.

```
aws rds describe-db-parameters --db-parameter-group-name mydbparametergroup
```

Perintah memberikan respons seperti berikut:

```
DBPARAMETER  Parameter Name            Parameter Value  Source           Data Type
 Apply Type  Is Modifiable
DBPARAMETER  allow-suspicious-udfs                      engine-default   boolean
 static      false
DBPARAMETER  auto_increment_increment                   engine-default   integer
 dynamic     true
DBPARAMETER  auto_increment_offset                      engine-default   integer
 dynamic     true
DBPARAMETER  binlog_cache_size         32768            system           integer
 dynamic     true
DBPARAMETER  socket                    /tmp/mysql.sock  system           string
 static      false
```

API RDS

Untuk melihat nilai parameter untuk grup parameter DB, gunakan perintah `DescribeDBParameters` API RDS dengan parameter wajib berikut.

- `DBParameterGroupName`

Menghapus grup parameter DB

Anda dapat menghapus grup parameter DB menggunakan AWS Management Console, AWS CLI, atau RDS API. Grup parameter memenuhi syarat untuk dihapus hanya jika tidak terkait dengan instans DB.

Konsol

Untuk menghapus grup parameter DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup parameter.

   Grup parameter DB akan muncul dalam daftar.
3. Pilih nama grup parameter yang akan dihapus.

4. Pilih Tindakan dan kemudian Hapus.
5. Tinjau nama grup parameter dan kemudian pilih Hapus.

AWS CLI

Untuk menghapus grup parameter DB, gunakan AWS CLI `delete-db-parameter-group`perintah dengan parameter yang diperlukan berikut.

- `--db-parameter-group-name`

Example

Contoh berikut menghapus kelompok parameter DB bernama mydbparametergroup.

```
aws rds delete-db-parameter-group --db-parameter-group-name mydbparametergroup
```

API RDS

Untuk menghapus grup parameter DB, gunakan `DeleteDBParameterGroup`perintah RDS API dengan parameter wajib berikut.

- `DBParameterGroupName`

Bekerja dengan grup parameter klaster DB untuk klaster DB Multi-AZ

Klaster DB Multi-AZ menggunakan grup parameter klaster DB. Bagian berikut menjelaskan cara mengonfigurasi dan mengelola grup parameter klaster DB.

Topik

- Membuat grup parameter klaster DB
- Mengubah parameter dalam grup parameter klaster DB
- Mengatur ulang parameter dalam grup parameter klaster DB
- Menyalin grup parameter klaster DB
- Mencantumkan grup parameter klaster DB
- Melihat nilai parameter untuk grup parameter klaster DB

- Menghapus grup parameter cluster DB

Membuat grup parameter klaster DB

Anda dapat membuat grup parameter cluster DB baru menggunakan AWS Management Console, AWS CLI, atau RDS API.

Setelah Anda membuat grup parameter klaster DB, tunggu minimal 5 menit sebelum membuat klaster DB yang menggunakan grup parameter klaster DB tersebut. Dengan demikian, Amazon RDS dapat sepenuhnya membuat grup parameter sebelum digunakan oleh klaster DB baru. Anda dapat menggunakan halaman grup Parameter di konsol Amazon RDS atau describe-db-cluster-parametersperintah untuk memverifikasi bahwa grup parameter cluster DB Anda dibuat.

Batasan berikut berlaku untuk nama grup parameter klaster DB:

- Nama harus berisi 1 sampai 255 huruf, angka, atau tanda hubung.

  Nama grup parameter default boleh menyertakan titik, seperti `default.aurora-mysql5.7`. Namun, nama grup parameter kustom tidak boleh menyertakan titik.
- Karakter pertamanya harus berupa huruf.
- Nama tidak boleh diakhiri dengan tanda hubung atau berisi dua tanda hubung berturut-turut.

Konsol

Untuk membuat grup parameter klaster DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup parameter.
3. Pilih Buat grup parameter.

   Jendela Buat grup parameter akan muncul.
4. Dalam daftar Kelompok grup parameter, pilih kelompok grup parameter DB.
5. Dalam daftar Type, pilih grup parameter cluster DB.
6. Di kotak Nama grup, masukkan nama grup parameter klaster DB yang baru.
7. Di kotak Deskripsi, masukkan deskripsi untuk grup parameter klaster DB yang baru.
8. Pilih Buat.

AWS CLI

Untuk membuat grup parameter cluster DB, gunakan AWS CLI `create-db-cluster-parameter-group`perintah.

Contoh berikut membuat grup parameter klaster DB yang bernama mydbclusterparametergroup untuk RDS for MySQL versi 8.0 dengan deskripsi "Grup parameter klaster baru saya".

Sertakan parameter wajib berikut:

- `--db-cluster-parameter-group-name`
- `--db-parameter-group-family`
- `--description`

Untuk mencantumkan semua kelompok grup parameter yang tersedia, gunakan perintah berikut:

```
aws rds describe-db-engine-versions --query "DBEngineVersions[].DBParameterGroupFamily"
```

> **Note**
>
> Output berisi duplikat.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-cluster-parameter-group \
    --db-cluster-parameter-group-name mydbclusterparametergroup \
    --db-parameter-group-family mysql8.0 \
    --description "My new cluster parameter group"
```

Untuk Windows:

```
aws rds create-db-cluster-parameter-group ^
    --db-cluster-parameter-group-name mydbclusterparametergroup ^
    --db-parameter-group-family mysql8.0 ^
    --description "My new cluster parameter group"
```

Perintah ini menghasilkan output yang serupa dengan yang berikut:

```
{
    "DBClusterParameterGroup": {
        "DBClusterParameterGroupName": "mydbclusterparametergroup",
        "DBParameterGroupFamily": "mysql8.0",
        "Description": "My new cluster parameter group",
        "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:cluster-
pg:mydbclusterparametergroup2"
    }
}
```

API RDS

Untuk membuat grup parameter klaster DB, gunakan tindakan `CreateDBClusterParameterGroup` API RDS.

Sertakan parameter wajib berikut:

- `DBClusterParameterGroupName`
- `DBParameterGroupFamily`
- `Description`

Mengubah parameter dalam grup parameter klaster DB

Anda dapat mengubah nilai parameter dalam grup parameter klaster DB buatan pelanggan. Anda tidak dapat mengubah nilai parameter dalam grup parameter klaster DB default. Perubahan pada parameter dalam grup parameter klaster DB buatan pelanggan diterapkan ke semua klaster DB yang dikaitkan dengan grup parameter klaster DB.

Konsol

Untuk mengubah grup parameter klaster DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup parameter.
3. Dalam daftar, pilih grup parameter yang ingin Anda modifikasi.
4. Untuk Tindakan grup parameter, pilih Edit.

5. Ubah nilai parameter yang ingin Anda modifikasi. Anda dapat menggulir parameter menggunakan tombol panah di bagian kanan atas kotak dialog.

   Anda tidak dapat mengubah nilai dalam grup parameter default.
6. Pilih Simpan perubahan.
7. Reboot instance DB utama (penulis) di cluster untuk menerapkan perubahan padanya.
8. Kemudian reboot instans DB pembaca untuk menerapkan perubahan pada mereka.

AWS CLI

Untuk memodifikasi grup parameter cluster DB, gunakan AWS CLI `modify-db-cluster-parameter-group`perintah dengan parameter yang diperlukan berikut:

- `--db-cluster-parameter-group-name`
- `--parameters`

Contoh berikut memodifikasi nilai `server_audit_logging` dan `server_audit_logs_upload` dalam grup parameter klaster DB yang bernama mydbclusterparametergroup.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-cluster-parameter-group \
    --db-cluster-parameter-group-name mydbclusterparametergroup \
    --parameters
 "ParameterName=server_audit_logging,ParameterValue=1,ApplyMethod=immediate" \

 "ParameterName=server_audit_logs_upload,ParameterValue=1,ApplyMethod=immediate"
```

Untuk Windows:

```
aws rds modify-db-cluster-parameter-group ^
    --db-cluster-parameter-group-name mydbclusterparametergroup ^
    --parameters
 "ParameterName=server_audit_logging,ParameterValue=1,ApplyMethod=immediate" ^

 "ParameterName=server_audit_logs_upload,ParameterValue=1,ApplyMethod=immediate"
```

Perintah menghasilkan output seperti berikut:

```
DBCLUSTERPARAMETERGROUP  mydbclusterparametergroup
```

API RDS

Untuk memodifikasi grup parameter klaster DB, gunakan perintah `ModifyDBClusterParameterGroup` API RDS dengan parameter wajib berikut:

- `DBClusterParameterGroupName`
- `Parameters`

Mengatur ulang parameter dalam grup parameter klaster DB

Anda dapat mengatur ulang parameter ke nilai defaultnya dalam grup parameter klaster DB buatan pelanggan. Perubahan pada parameter dalam grup parameter klaster DB buatan pelanggan diterapkan ke semua klaster DB yang dikaitkan dengan grup parameter klaster DB.

> **Note**
>
> Dalam grup parameter klaster DB default, parameter selalu ditetapkan ke nilai defaultnya.

Konsol

Untuk mengatur ulang parameter dalam grup parameter klaster DB ke nilai defaultnya

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup parameter.
3. Dalam daftar, pilih grup parameter.
4. Untuk Tindakan grup parameter, pilih Edit.
5. Pilih parameter yang ingin Anda atur ulang ke nilai defaultnya. Anda dapat menggulir parameter menggunakan tombol panah di bagian kanan atas kotak dialog.

   Anda tidak dapat mengatur ulang nilai dalam grup parameter default.
6. Pilih Atur ulang lalu konfirmasikan dengan memilih Atur ulang parameter.

7. Reboot instans DB primer dalam klaster DB untuk menerapkan perubahan pada semua instans DB dalam klaster DB.

AWS CLI

Untuk mengatur ulang parameter dalam grup parameter cluster DB ke nilai defaultnya, gunakan AWS CLI `reset-db-cluster-parameter-group`perintah dengan opsi wajib berikut:`--db-cluster-parameter-group-name`.

Untuk mengatur ulang semua parameter dalam grup parameter klaster DB, tentukan opsi `--reset-all-parameters`. Untuk mengatur ulang parameter tertentu, tentukan opsi `--parameters`.

Contoh berikut mengatur ulang semua parameter dalam grup parameter DB yang bernama mydbparametergroup ke nilai defaultnya.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds reset-db-cluster-parameter-group \
    --db-cluster-parameter-group-name mydbparametergroup \
    --reset-all-parameters
```

Untuk Windows:

```
aws rds reset-db-cluster-parameter-group ^
    --db-cluster-parameter-group-name mydbparametergroup ^
    --reset-all-parameters
```

Contoh berikut mengatur ulang `server_audit_logging` dan `server_audit_logs_upload` ke nilai defaultnya dalam grup parameter klaster DB yang bernama mybdclusterparametergroup.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds reset-db-cluster-parameter-group \
    --db-cluster-parameter-group-name mydbclusterparametergroup \
    --parameters "ParameterName=server_audit_logging,ApplyMethod=immediate" \
```

```
                "ParameterName=server_audit_logs_upload,ApplyMethod=immediate"
```

Untuk Windows:

```
aws rds reset-db-cluster-parameter-group ^
    --db-cluster-parameter-group-name mydbclusterparametergroup ^
    --parameters
 "ParameterName=server_audit_logging,ParameterValue=1,ApplyMethod=immediate" ^

 "ParameterName=server_audit_logs_upload,ParameterValue=1,ApplyMethod=immediate"
```

Perintah menghasilkan output seperti berikut:

```
DBClusterParameterGroupName  mydbclusterparametergroup
```

API RDS

Untuk mengatur ulang parameter dalam grup parameter klaster DB ke nilai default, gunakan perintah `ResetDBClusterParameterGroup` API RDS dengan parameter wajib berikut: `DBClusterParameterGroupName`.

Untuk mengatur ulang semua parameter dalam grup parameter klaster DB, atur parameter `ResetAllParameters` ke `true`. Untuk mengatur ulang parameter tertentu, tentukan parameter `Parameters`.

Menyalin grup parameter klaster DB

Anda dapat menyalin grup parameter klaster DB kustom yang Anda buat. Menyalin grup parameter adalah solusi yang mudah jika Anda sudah membuat grup parameter klaster DB dan Anda ingin menyertakan sebagian besar parameter dan nilai kustom dari grup tersebut dalam grup parameter klaster DB yang baru. Anda dapat menyalin grup parameter cluster DB dengan menggunakan perintah AWS CLI copy-db-cluster-parameter-group atau operasi ClusterParameterGroupCopyDB API RDS.

Setelah Anda membuat grup parameter klaster DB, tunggu minimal 5 menit sebelum membuat klaster DB yang menggunakan grup parameter klaster DB tersebut. Dengan demikian, Amazon RDS dapat sepenuhnya menyalin grup parameter sebelum digunakan oleh klaster DB baru. Anda dapat menggunakan halaman grup Parameter di konsol Amazon RDS atau describe-db-cluster-parametersperintah untuk memverifikasi bahwa grup parameter cluster DB Anda dibuat.

> **Note**
>
> Anda tidak dapat menyalin grup parameter default. Namun, Anda dapat membuat grup parameter baru yang didasarkan pada grup parameter default.
> Anda tidak dapat menyalin grup parameter cluster DB ke grup yang berbeda Akun AWS atau Wilayah AWS.

Konsol

Untuk menyalin grup parameter klaster DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup parameter.
3. Dalam daftar, pilih grup parameter kustom yang ingin Anda salin.
4. Untuk Tindakan grup parameter, pilih Salin.
5. Di Pengidentifikasi grup parameter DB baru, masukkan nama untuk grup parameter baru.
6. Di Deskripsi, masukkan deskripsi untuk grup parameter DB yang baru.
7. Pilih Salin.

AWS CLI

Untuk menyalin grup parameter cluster DB, gunakan AWS CLI `copy-db-cluster-parameter-group`perintah dengan parameter yang diperlukan berikut:

- `--source-db-cluster-parameter-group-identifier`
- `--target-db-cluster-parameter-group-identifier`
- `--target-db-cluster-parameter-group-description`

Contoh berikut membuat grup parameter klaster DB baru yang bernama `mygroup2` yang merupakan salinan dari grup parameter klaster DB `mygroup1`.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds copy-db-cluster-parameter-group \
    --source-db-cluster-parameter-group-identifier mygroup1 \
    --target-db-cluster-parameter-group-identifier mygroup2 \
    --target-db-cluster-parameter-group-description "DB parameter group 2"
```

Untuk Windows:

```
aws rds copy-db-cluster-parameter-group ^
    --source-db-cluster-parameter-group-identifier mygroup1 ^
    --target-db-cluster-parameter-group-identifier mygroup2 ^
    --target-db-cluster-parameter-group-description "DB parameter group 2"
```

API RDS

Untuk menyalin grup parameter klaster DB, gunakan operasi `CopyDBClusterParameterGroup` API RDS dengan parameter wajib berikut:

- `SourceDBClusterParameterGroupIdentifier`
- `TargetDBClusterParameterGroupIdentifier`
- `TargetDBClusterParameterGroupDescription`

Mencantumkan grup parameter klaster DB

Anda dapat mencantumkan grup parameter cluster DB yang telah Anda buat untuk AWS akun Anda.

> **Note**
>
> Grup parameter default secara otomatis dibuat dari template parameter default ketika Anda membuat klaster DB untuk mesin dan versi DB tertentu. Grup parameter default ini berisi pengaturan parameter pilihan dan tidak dapat dimodifikasi. Saat membuat grup parameter kustom, Anda dapat memodifikasi pengaturan parameter.

Konsol

Untuk mencantumkan semua grup parameter cluster DB untuk AWS akun

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.

2. Di panel navigasi, pilih Grup parameter.

   Grup parameter klaster DB muncul dalam daftar dengan Grup parameter klaster DB untuk Jenis.

AWS CLI

Untuk mencantumkan semua grup parameter cluster DB untuk AWS akun, gunakan AWS CLI `describe-db-cluster-parameter-groups`perintah.

Example

Contoh berikut mencantumkan semua grup parameter klaster DB yang tersedia untuk akun AWS .

```
aws rds describe-db-cluster-parameter-groups
```

Contoh berikut menjelaskan grup parameter mydbclusterparametergroup.

Untuk Linux, macOS, atau Unix:

```
aws rds describe-db-cluster-parameter-groups \
    --db-cluster-parameter-group-name mydbclusterparametergroup
```

Untuk Windows:

```
aws rds describe-db-cluster-parameter-groups ^
    --db-cluster-parameter-group-name mydbclusterparametergroup
```

Perintah memberikan respons seperti berikut:

```
{
    "DBClusterParameterGroups": [
        {
            "DBClusterParameterGroupName": "mydbclusterparametergroup2",
            "DBParameterGroupFamily": "mysql8.0",
            "Description": "My new cluster parameter group",
            "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:cluster-
pg:mydbclusterparametergroup"
        }
    ]
}
```

API RDS

Untuk mencantumkan semua grup parameter cluster DB untuk AWS akun, gunakan `DescribeDBClusterParameterGroups`tindakan RDS API.

Melihat nilai parameter untuk grup parameter klaster DB

Anda dapat memperoleh daftar semua parameter dalam grup parameter klaster DB beserta nilainya.

Konsol

Untuk melihat nilai parameter untuk grup parameter klaster DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup parameter.

   Grup parameter klaster DB muncul dalam daftar dengan Grup parameter klaster DB untuk Jenis.
3. Pilih nama grup parameter klaster DB untuk melihat daftar parameternya.

AWS CLI

Untuk melihat nilai parameter untuk grup parameter cluster DB, gunakan AWS CLI `describe-db-cluster-parameters`perintah dengan parameter yang diperlukan berikut.

- `--db-cluster-parameter-group-name`

Example

Contoh berikut mencantumkan parameter dan nilai parameter untuk grup parameter klaster DB yang bernama mydbclusterparametergroup, dalam format JSON.

Perintah memberikan respons seperti berikut:

```
aws rds describe-db-cluster-parameters --db-cluster-parameter-group-
name mydbclusterparametergroup
```

```
{
    "Parameters": [
        {
```

```
            "ParameterName": "activate_all_roles_on_login",
            "ParameterValue": "0",
            "Description": "Automatically set all granted roles as active after the
 user has authenticated successfully.",
            "Source": "engine-default",
            "ApplyType": "dynamic",
            "DataType": "boolean",
            "AllowedValues": "0,1",
            "IsModifiable": true,
            "ApplyMethod": "pending-reboot",
            "SupportedEngineModes": [
                "provisioned"
            ]
        },
        {
            "ParameterName": "allow-suspicious-udfs",
            "Description": "Controls whether user-defined functions that have only an
 xxx symbol for the main function can be loaded",
            "Source": "engine-default",
            "ApplyType": "static",
            "DataType": "boolean",
            "AllowedValues": "0,1",
            "IsModifiable": false,
            "ApplyMethod": "pending-reboot",
            "SupportedEngineModes": [
                "provisioned"
            ]
        },
...
```

API RDS

Untuk melihat nilai parameter untuk grup parameter klaster DB, gunakan perintah `DescribeDBClusterParameters` API RDS dengan parameter wajib berikut.

- `DBClusterParameterGroupName`

Dalam beberapa kasus, nilai yang diizinkan untuk parameter tidak ditampilkan. Nilai ini selalu merupakan parameter yang sumbernya adalah mesin basis data default.

Untuk melihat nilai parameter ini, Anda dapat menjalankan pernyataan SQL berikut:

- MySQL:

```
-- Show the value of a particular parameter
mysql$ SHOW VARIABLES LIKE '%parameter_name%';

-- Show the values of all parameters
mysql$ SHOW VARIABLES;
```

- PostgreSQL:

```
-- Show the value of a particular parameter
postgresql=> SHOW parameter_name;

-- Show the values of all parameters
postgresql=> SHOW ALL;
```

Menghapus grup parameter cluster DB

Anda dapat menghapus grup parameter cluster DB menggunakan AWS Management Console, AWS CLI, atau RDS API. Grup parameter grup parameter cluster DB memenuhi syarat untuk dihapus hanya jika tidak terkait dengan cluster DB.

Konsol

Untuk menghapus grup parameter

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup parameter.

   Grup parameter muncul dalam daftar.
3. Pilih nama grup parameter cluster DB yang akan dihapus.
4. Pilih Tindakan dan kemudian Hapus.
5. Tinjau nama grup parameter dan kemudian pilih Hapus.

AWS CLI

Untuk menghapus grup parameter cluster DB, gunakan AWS CLI `delete-db-cluster-parameter-group`perintah dengan parameter yang diperlukan berikut.

- `--db-parameter-group-name`

Example

Contoh berikut menghapus kelompok parameter cluster DB bernama mydbparametergroup.

```
aws rds delete-db-cluster-parameter-group --db-parameter-group-name mydbparametergroup
```

API RDS

Untuk menghapus grup parameter cluster DB, gunakan [DeleteDBClusterParameterGroup](#)perintah RDS API dengan parameter wajib berikut.

- `DBParameterGroupName`

Membandingkan grup parameter DB

Anda dapat menggunakan AWS Management Console untuk melihat perbedaan antara dua kelompok parameter DB.

Kedua grup parameter yang ditentukan harus merupakan grup parameter DB atau grup parameter klaster DB. Hal ini berlaku meskipun mesin dan versi DB sama. Misalnya, Anda tidak dapat membandingkan grup parameter DB `aurora-mysql8.0` (Aurora MySQL versi 3) dan grup parameter klaster DB `aurora-mysql8.0`.

Anda dapat membandingkan grup parameter DB Aurora MySQL dan RDS for MySQL, bahkan untuk versi yang berbeda, tetapi Anda tidak dapat membandingkan grup parameter DB Aurora PostgreSQL dan RDS for PostgreSQL.

Untuk membandingkan dua kelompok parameter DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup Parameter.
3. Dalam daftar, pilih grup parameter yang ingin Anda bandingkan.

> **Note**
>
> Untuk membandingkan grup parameter default dengan grup parameter kustom, pilih grup parameter default pada tab Default terlebih dahulu, lalu pilih grup parameter kustom pada tab Kustom.

4. Dari Tindakan, pilih Bandingkan.

Menentukan parameter DB

Jenis parameter DB mencakup yang berikut ini:

- Bilangan Bulat
- Boolean
- String
- Panjang
- Ganda
- Stempel Waktu
- Objek dari tipe data lain yang ditentukan
- Rangkaian nilai tipe bilangan bulat, Boolean, string, panjang, ganda, stempel waktu, atau objek

Anda juga dapat menentukan parameter bilangan bulat dan Boolean menggunakan ekspresi, formula, dan fungsi.

Untuk mesin Oracle, Anda dapat menggunakan variabel formula `DBInstanceClassHugePagesDefault` untuk menentukan parameter DB Boolean. Lihat Variabel formula parameter DB.

Untuk mesin PostgreSQL, Anda dapat menggunakan ekspresi untuk menentukan parameter DB Boolean. Lihat Ekspresi parameter DB Boolean.

Daftar Isi

- Fungsi parameter DB
- Ekspresi parameter DB Boolean
- Ekspresi log parameter DB
- Contoh nilai parameter DB

Formula parameter DB

Formula parameter DB adalah ekspresi yang menghasilkan nilai bilangan bulat atau nilai Boolean. Anda mengapit ekspresi dalam kurung: {}. Anda dapat menentukan formula untuk nilai parameter DB atau sebagai argumen untuk fungsi parameter DB.

Sintaks

```
{FormulaVariable}
{FormulaVariable*Integer}
{FormulaVariable*Integer/Integer}
{FormulaVariable/Integer}
```

Variabel formula parameter DB

Setiap variabel formula menghasilkan nilai bilangan bulat atau Boolean. Nama variabel bersifat peka huruf besar-kecil.

AllocatedStorage

Mengembalikan bilangan bulat yang mewakili ukuran, dalam byte, dari volume data.

DB InstanceClassHugePagesDefault

Mengembalikan nilai Boolean. Saat ini, hanya didukung untuk mesin Oracle.

Untuk informasi selengkapnya, lihat Mengaktifkan HugePages untuk instans RDS for Oracle.

DB InstanceClassMemory

Mengembalikan bilangan bulat untuk jumlah byte memori yang tersedia untuk proses basis data. Angka ini dihitung secara internal dengan memulai dengan jumlah total memori untuk kelas instans DB. Dari sini, perhitungan mengurangi memori yang disediakan untuk sistem operasi dan proses RDS yang mengelola instans. Oleh karena itu, angkanya selalu agak lebih rendah dari

angka memori yang ditunjukkan pada tabel kelas instans di Kelas instans DB . Nilai yang pasti bergantung pada kombinasi faktor. Faktor ini termasuk kelas instans, mesin DB, dan apakah hal tersebut berlaku untuk instans RDS atau instans yang merupakan bagian dari klaster Aurora.

DBInstanceVCPU

Mengembalikan bilangan bulat yang mewakili jumlah unit pengolahan pusat virtual (vCPU) yang digunakan oleh Amazon RDS untuk mengelola instans. Saat ini, itu hanya didukung untuk RDS untuk mesin PostgreSQL.

EndPointPort

Mengembalikan bilangan bulat yang mewakili port yang digunakan saat terhubung ke instans DB.

TrueIfReplica

Mengembalikan `1` jika instans DB adalah replika baca dan `0` jika tidak. Nilai ini adalah nilai default untuk parameter `read_only` di MySQL.

Operator formula parameter DB

Formula parameter DB mendukung dua operator: pembagian dan perkalian.

Operator pembagian: /

Membagi dividen dengan pembagi, mengembalikan hasil bagi bilangan bulat. Desimal dalam hasil bagi dipotong, tidak dibulatkan.

Sintaks

```
dividend / divisor
```

Argumen terbagi dan pembagi harus merupakan ekspresi bilangan bulat.

Operator perkalian: *

Mengalikan ekspresi, mengembalikan hasil ekspresi. Desimal dalam ekspresi dipotong, tidak dibulatkan.

Sintaks

```
expression * expression
```

Kedua ekspresi harus berupa bilangan bulat.

Fungsi parameter DB

Anda menentukan argumen fungsi parameter DB sebagai bilangan bulat atau formula. Setiap fungsi harus memiliki setidaknya satu argumen. Tentukan beberapa argumen sebagai daftar yang dipisahkan koma. Daftar ini tidak dapat memiliki anggota yang kosong, seperti argument1,,argument3. Nama fungsi bersifat peka huruf besar-kecil.

JIKA

Mengembalikan argumen.

Saat ini, hanya didukung untuk mesin Oracle, dan satu-satunya argumen pertama yang didukung adalah `{DBInstanceClassHugePagesDefault}`. Untuk informasi selengkapnya, lihat Mengaktifkan HugePages untuk instans RDS for Oracle.

Sintaks

```
IF(argument1, argument2, argument3)
```

Mengembalikan argumen kedua jika argumen pertama ternyata benar setelah dievaluasi. Mengembalikan argumen ketiga.

TERBESAR

Mengembalikan nilai terbesar dari daftar bilangan bulat atau formula parameter.

Sintaks

```
GREATEST(argument1, argument2,...argumentn)
```

Mengembalikan bilangan bulat.

TERKECIL

Mengembalikan nilai terkecil dari daftar bilangan bulat atau formula parameter.

Sintaks

```
LEAST(argument1, argument2,...argumentn)
```

Mengembalikan bilangan bulat.

JUMLAH

Menambahkan nilai bilangan bulat atau formula parameter yang ditentukan.

Sintaks

```
SUM(argument1, argument2,...argumentn)
```

Mengembalikan bilangan bulat.

Ekspresi parameter DB Boolean

Ekspresi parameter DB Boolean menghasilkan nilai Boolean 1 atau 0. Ekspresi diapit dengan tanda petik.

> **Note**
>
> Ekspresi parameter DB Boolean hanya didukung untuk mesin PostgreSQL.

Sintaks

```
"expression operator expression"
```

Kedua ekspresi harus menghasilkan bilangan bulat. Ekspresi dapat berupa hal-hal berikut:

- Konstanta bilangan bulat
- Formula parameter DB
- Fungsi parameter DB
- Variabel parameter DB

Ekspresi parameter DB Boolean mendukung operator ketimpangan berikut:

Lebih besar dari operator: **>**

Sintaks

```
"expression > expression"
```

Kurang dari operator: **<**

Sintaks

```
"expression < expression"
```

Lebih besar atau sama dengan operator: **>=**, **=>**

Sintaks

```
"expression >= expression"
"expression => expression"
```

Lebih kecil dari atau sama dengan operator: **>=**, **=>**

Sintaks

```
"expression <= expression"
"expression =< expression"
```

Example menggunakan ekspresi parameter DB Boolean

Contoh ekspresi parameter DB Boolean berikut ini membandingkan hasil dari formula parameter dengan bilangan bulat. Hal ini dilakukan untuk memodifikasi parameter Boolean DB `wal_compression` untuk instans DB PostgreSQL. Ekspresi parameter membandingkan jumlah vCPU dengan nilai 2. Jika jumlah vCPU lebih besar dari 2, maka parameter DB `wal_compression` diatur ke benar.

```
aws rds modify-db-parameter-group --db-parameter-group-name group-name \
--parameters "ParameterName=wal_compression,ParameterValue=\"{DBInstanceVCPU} > 2\" "
```

Ekspresi log parameter DB

Anda dapat mengatur nilai parameter DB bilangan bulat ke ekspresi log. Anda mengapit ekspresi dalam kurung: {}. Sebagai contoh:

```
{log(DBInstanceClassMemory/8187281418)*1000}
```

Fungsi `log` mewakili dasar log 2. Contoh ini juga menggunakan variabel formula `DBInstanceClassMemory`. Lihat Variabel formula parameter DB.

> **Note**
>
> Saat ini, Anda tidak dapat menentukan parameter `innodb_log_file_size` MySQL dengan nilai selain bilangan bulat.

Contoh nilai parameter DB

Contoh-contoh ini menunjukkan penggunaan formula, fungsi, dan ekspresi untuk nilai parameter DB.

> **Warning**
>
> Pengaturan parameter yang tidak tepat dalam grup parameter DB dapat memiliki efek merugikan yang tidak diinginkan. Efek tersebut termasuk performa terdegradasi dan ketidakstabilan sistem. Selalu berhati-hati saat memodifikasi parameter basis data dan cadangkan data Anda sebelum memodifikasi grup parameter DB Anda. Cobalah perubahan grup parameter pada instance DB pengujian, yang dibuat menggunakan point-in-time-restores, sebelum menerapkan perubahan grup parameter tersebut ke instans DB produksi Anda.

Example Menggunakan fungsi parameter DB TERBESAR

Anda dapat menentukan fungsi `GREATEST` dalam parameter proses Oracle. Gunakan untuk mengatur jumlah proses pengguna menjadi yang lebih besar antara 80 atau `DBInstanceClassMemory` dibagi dengan 9.868.951.

```
GREATEST({DBInstanceClassMemory/9868951},80)
```

Example Menggunakan fungsi parameter DB TERKECIL

Anda dapat menentukan fungsi `LEAST` dalam nilai parameter `max_binlog_cache_size` MySQL. Gunakan untuk mengatur ukuran cache maksimum yang dapat digunakan oleh transaksi dalam instans MySQL menjadi yang lebih kecil antara 1 MB atau `DBInstanceClass`/256.

```
LEAST({DBInstanceClassMemory/256},10485760)
```

Membuat ElastiCache cache Amazon menggunakan pengaturan instans

ElastiCache adalah layanan caching dalam memori yang dikelola sepenuhnya yang menyediakan latensi baca dan tulis mikrodetik yang mendukung kasus penggunaan real-time yang fleksibel. ElastiCache dapat membantu Anda mempercepat kinerja aplikasi dan database. Anda dapat menggunakan ElastiCache sebagai penyimpanan data utama untuk kasus penggunaan yang tidak memerlukan daya tahan data, seperti papan peringkat game, streaming, dan analitik data. ElastiCache membantu menghilangkan kompleksitas yang terkait dengan penyebaran dan pengelolaan lingkungan komputasi terdistribusi. Untuk informasi selengkapnya, lihat Kasus ElastiCache Penggunaan Umum dan Bagaimana ElastiCache Dapat Membantu untuk Memcached dan Kasus ElastiCache Penggunaan Umum dan Bagaimana ElastiCache Dapat Membantu Redis. Anda dapat menggunakan konsol Amazon RDS untuk membuat ElastiCache cache.

Anda dapat mengoperasikan Amazon ElastiCache dalam dua format. Anda dapat memulai dengan cache nirserver atau memilih untuk merancang klaster cache Anda sendiri. Jika Anda memilih untuk mendesain cluster cache Anda sendiri, ElastiCache bekerja dengan mesin Redis dan Memcached. Jika Anda tidak yakin mesin mana yang ingin Anda gunakan, lihat Membandingkan Memcached dan Redis. Untuk informasi selengkapnya tentang Amazon ElastiCache, lihat Panduan ElastiCache Pengguna Amazon.

Topik

- Ikhtisar pembuatan ElastiCache cache dengan pengaturan
- Membuat ElastiCache cache dengan pengaturan dari instance

Ikhtisar pembuatan ElastiCache cache dengan pengaturan

Anda dapat membuat ElastiCache cache dari Amazon RDS menggunakan pengaturan konfigurasi yang sama dengan instans RDS DB cluster yang baru dibuat atau yang sudah ada.

Beberapa kasus penggunaan untuk mengaitkan ElastiCache cache dengan instans DB Anda:

- Anda dapat menghemat biaya dan meningkatkan kinerja Anda ElastiCache dengan menggunakan RDS versus berjalan pada RDS saja.

  Misalnya, Anda dapat menghemat biaya hingga 55% dan mendapatkan kinerja baca hingga 80x lebih cepat ElastiCache dengan menggunakan RDS untuk MySQL versus RDS untuk MySQL saja.

- Anda dapat menggunakan ElastiCache cache sebagai penyimpanan data utama untuk aplikasi yang tidak memerlukan daya tahan data. Aplikasi Anda yang menggunakan Redis atau Memcached dapat digunakan ElastiCache dengan hampir tidak ada modifikasi.

Saat Anda membuat ElastiCache cache dari RDS, ElastiCache cache mewarisi pengaturan berikut dari instance RDS :

- ElastiCache pengaturan konektivitas
- ElastiCache pengaturan keamanan

Anda juga dapat mengatur pengaturan konfigurasi cache sesuai dengan kebutuhan Anda.

Menyiapkan ElastiCache di aplikasi Anda

Aplikasi Anda harus diatur untuk memanfaatkan ElastiCache cache. Anda juga dapat mengoptimalkan dan meningkatkan kinerja cache dengan menyiapkan aplikasi Anda untuk menggunakan strategi caching tergantung pada kebutuhan Anda.

- Untuk mengakses ElastiCache cache dan memulai, lihat Memulai Amazon ElastiCache untuk Redis dan Memulai Amazon ElastiCache untuk Memcached.
- Untuk informasi lebih lanjut tentang strategi caching, lihat Strategi caching dan praktik terbaik untuk Memcached dan Strategi caching dan praktik terbaik untuk Redis.
- Untuk informasi selengkapnya tentang ketersediaan tinggi di ElastiCache klaster Redis, lihat Ketersediaan tinggi menggunakan grup replikasi.
- Anda mungkin dikenakan biaya yang terkait dengan penyimpanan cadangan, transfer data di dalam atau di seluruh wilayah, atau penggunaan. AWS Outposts Untuk detail harga, lihat ElastiCache harga Amazon.

Membuat ElastiCache cache dengan pengaturan dari instance

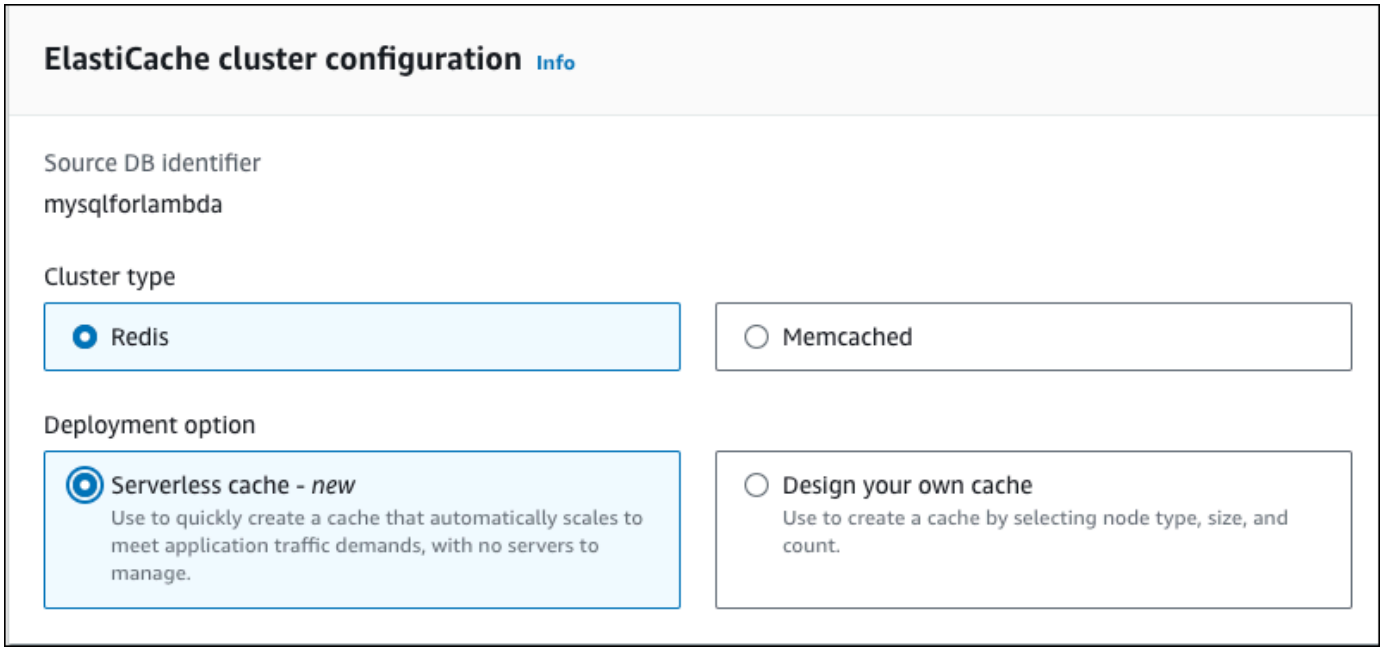
Buat ElastiCache cache dengan pengaturan dari instance DB

1. Untuk membuat instans DB, ikuti petunjuk di Membuat instans DB Amazon RDS.
2. Setelah membuat instance RDS, konsol menampilkan jendela add-on yang Disarankan. Pilih Buat ElastiCache cluster dari RDS menggunakan pengaturan DB Anda.

Untuk database yang ada, di halaman Databases, pilih instance DB yang diperlukan. Di menu dropdown Actions, pilih Create ElastiCache cluster untuk membuat ElastiCache cache di RDS yang memiliki pengaturan yang sama dengan instans DB cluster yang ada.

Di bagian ElastiCache konfigurasi, pengidentifikasi Source DB menampilkan instance DB mana yang mewarisi pengaturan ElastiCache cache.

3. Pilih apakah Anda ingin membuat klaster Redis atau Memcached. Untuk informasi selengkapnya, lihat Membandingkan Memcached dan Redis.



4. Setelah ini, pilih apakah Anda ingin membuat cache Tanpa Server atau Desain cache Anda sendiri. Untuk informasi selengkapnya, lihat Memilih antara opsi penerapan.

   Jika Anda memilih Cache tanpa server:

   a. Dalam pengaturan Cache, masukkan nilai untuk Nama dan Deskripsi.

   b. Di bawah Lihat pengaturan default, biarkan pengaturan default untuk membuat koneksi antara cache Anda dan instance DB.

   c. Anda juga dapat mengedit pengaturan default dengan memilih Sesuaikan pengaturan default. Pilih pengaturan ElastiCache konektivitas, pengaturan ElastiCache keamanan, dan Batas penggunaan maksimum.

5. Jika Anda memilih Desain cache Anda sendiri:

a. Jika Anda memilih klaster Redis, pilih apakah Anda ingin mempertahankan mode cluster Diaktifkan atau Dinonaktifkan. Untuk informasi selengkapnya, lihat Replikasi: Redis (Mode Klaster Dinonaktifkan) vs Redis (Mode Klaster Diaktifkan).

b. Masukkan nilai untuk Nama, Deskripsi, dan Versi mesin.

   Untuk Versi mesin, nilai default yang disarankan adalah versi mesin terbaru. Anda juga dapat memilih versi Engine untuk ElastiCache cache yang paling sesuai dengan kebutuhan Anda.

c. Pilih jenis simpul dalam opsi Jenis simpul. Untuk informasi selengkapnya, lihat Mengelola simpul.

   Jika Anda memilih untuk membuat klaster Redis dengan Mode klaster disetel ke Diaktifkan, maka masukkan jumlah serpihan (partisi/grup simpul) dalam opsi Jumlah serpihan.

   Masukkan jumlah replika setiap serpihan di Jumlah replika.

   > **Note**
   >
   > Jenis node yang dipilih, jumlah pecahan, dan jumlah replika semuanya memengaruhi kinerja cache dan biaya sumber daya Anda. Pastikan pengaturan ini sesuai dengan kebutuhan basis data Anda. Untuk informasi harga, lihat ElastiCache harga Amazon.

d. Pilih pengaturan ElastiCache konektivitas dan pengaturan ElastiCache keamanan. Anda dapat menyimpan pengaturan default atau menyesuaikan pengaturan ini sesuai kebutuhan Anda.

6. Verifikasi pengaturan default dan warisan ElastiCache cache Anda. Beberapa pengaturan tidak dapat diubah setelah pembuatan.

   > **Note**
   >
   > RDS mungkin menyesuaikan jendela cadangan ElastiCache cache Anda untuk memenuhi persyaratan jendela minimum 60 menit. Periode pencadangan basis data sumber Anda tetap sama.

7. Saat Anda siap, pilih Buat ElastiCache cache.

Konsol menampilkan spanduk konfirmasi untuk pembuatan ElastiCache cache. Ikuti tautan di spanduk ke ElastiCache konsol untuk melihat detail cache. ElastiCache Konsol menampilkan ElastiCache cache yang baru dibuat.

Mengelola instans DB Amazon RDS

Setelah itu, Anda dapat menemukan petunjuk untuk mengelola dan mempertahankan instans DB Amazon RDS Anda.

Topik

Menghentikan sementara instans DB Amazon RDS

Anda dapat menghentikan instans DB sebentar-sebentar untuk pengujian sementara atau untuk aktivitas pengembangan harian. Kasus penggunaan yang paling umum adalah pengoptimalan biaya.

> **Note**
>
> Dalam beberapa kasus, waktu yang lama diperlukan untuk menghentikan instans DB. Untuk menghentikan instans DB Anda dan memulai ulang segera, reboot instans DB. Untuk informasi selengkapnya, lihat Mem-boot ulang instans DB.

Topik

Gunakan kasus untuk menghentikan instans DB Anda

Menghentikan dan memulai instans DB lebih cepat daripada membuat snapshot DB, menghapus instans DB Anda, dan kemudian memulihkan snapshot saat Anda ingin mengakses instance. Kasus penggunaan umum untuk menghentikan instance termasuk yang berikut:

- Optimalisasi biaya — Untuk database non-produksi, Anda dapat menghentikan instans Amazon RDS DB sementara untuk menghemat uang. Saat instans dihentikan, Anda tidak dikenakan biaya untuk jam instans DB.

  > **Important**
  >
  > Saat instans DB Anda dihentikan, Anda dikenai biaya untuk penyimpanan terprovisi (termasuk IOPS yang Tersedia). Anda juga dikenai biaya untuk retensi cadangan,

termasuk snapshot manual dan cadangan otomatis dalam periode retensi yang Anda tentukan. Namun, Anda tidak dikenai biaya untuk jam instans DB. Untuk informasi selengkapnya, lihat FAQ penagihan.

- Pengembangan harian — Jika Anda memelihara instans DB untuk tujuan pengembangan, Anda dapat memulai instance saat diperlukan dan kemudian mematikan instance saat tidak diperlukan.
- Pengujian — Anda mungkin memerlukan instans DB sementara untuk menguji prosedur pencadangan dan pemulihan, migrasi, peningkatan aplikasi, atau aktivitas terkait. Dalam kasus penggunaan ini, Anda dapat menghentikan instans DB saat tidak diperlukan.
- Pelatihan — Jika Anda melakukan pelatihan di RDS, Anda mungkin perlu memulai instans DB selama sesi pelatihan dan mematikannya sesudahnya.

Mesin DB, kelas instans, dan Wilayah yang didukung

Anda dapat menghentikan dan memulai proses DB Amazon RDS yang menjalankan mesin DB berikut:

- Db2
- MariaDB
- Microsoft SQL Server, termasuk RDS Kustom untuk SQL Server
- MySQL
- Oracle
- PostgreSQL

Penghentian dan pengaktifan instans DB didukung untuk semua kelas instans DB, dan di semua Wilayah AWS .

Menghentikan instans DB di deployment Multi-AZ

Anda dapat menghentikan dan memulai instans DB dalam penerapan Multi-AZ. Perhatikan batasan berikut:

- Anda hanya dapat membuat penyebaran Multi-AZ jika mesin database Anda mendukungnya. Untuk informasi selengkapnya tentang dukungan mesin, lihatKlaster DB Multi-AZ.

- RDS untuk SQL Server tidak mendukung penghentian instans DB dalam penerapan Multi-AZ. Untuk informasi selengkapnya, lihat Batasan, catatan dan rekomendasi deployment Multi-AZ Microsoft SQL Server.
- Waktu yang lama mungkin diperlukan untuk menghentikan instans DB. Jika Anda memiliki setidaknya satu cadangan setelah failover sebelumnya, maka Anda dapat mempercepat operasi berhenti dengan melakukan reboot dengan operasi failover. Untuk informasi selengkapnya, lihat Mem-boot ulang instans DB.

Cara menghentikan instans DB

Operasi penghentian terjadi pada tahap-tahap berikut:

1. Instans DB memulai proses penonaktifan normal.

   Status instans DB berubah menjadi `stopping`.
2. Instans berhenti berjalan, hingga maksimal 7 hari berturut-turut.

   Status instans DB berubah menjadi `stopped`.

Karakteristik instans DB yang dihentikan

Ketika dalam keadaan berhenti, instans DB Anda memiliki karakteristik sebagai berikut:

- Instans DB Anda yang dihentikan mempertahankan yang berikut:
  - ID Instans
  - Titik akhir Server Nama Domain (DNS)
  - Grup parameter
  - Grup keamanan
  - Grup opsi
  - Log transaksi Amazon S3 (diperlukan untuk pemulihan) point-in-time

  Saat Anda memulai ulang instans DB, instans ini akan memiliki konfigurasi yang sama seperti saat Anda menghentikannya.
- Setiap volume penyimpanan tetap terlampir pada instans DB, dan datanya dipertahankan. RDS menghapus data apa pun yang disimpan dalam RAM instans DB.

Saat instans DB Anda dihentikan, Anda dikenai biaya untuk penyimpanan terprovisi (termasuk IOPS yang Tersedia). Anda juga dikenai biaya untuk retensi cadangan, termasuk snapshot manual dan cadangan otomatis dalam periode retensi yang Anda tentukan.

- RDS menghapus tindakan tertunda, kecuali untuk tindakan tertunda untuk grup opsi atau grup parameter DB.

> **Note**
>
> Terkadang, instans DB RDS for PostgreSQL tidak dinonaktifkan dengan normal. Jika ini terjadi, Anda akan melihat bahwa instans melewati proses pemulihan ketika Anda mengaktifkan ulang instans ini nanti. Ini adalah perilaku yang diharapkan dari mesin basis data, yang dimaksudkan untuk melindungi integritas basis data. Beberapa statistik dan penghitung berbasis memori tidak menyimpan riwayat dan diinisialisasi ulang setelah pengaktifan ulang, untuk mengambil data beban kerja operasional ke depan.

Restart otomatis dari instans DB yang dihentikan

Jika Anda tidak memulai instans DB Anda secara manual setelah dihentikan selama tujuh hari berturut-turut, RDS secara otomatis memulai instans DB Anda. Dengan cara ini, instans Anda tidak ketinggalan pembaruan pemeliharaan yang diperlukan. Untuk mempelajari cara menghentikan dan memulai instans sesuai jadwal, lihat Bagaimana cara menggunakan Step Functions untuk menghentikan instans Amazon RDS selama lebih dari 7 hari?.

Batasan untuk menghentikan instans DB Anda

Berikut ini adalah beberapa batasan untuk menghentikan dan memulai instans DB:

- Menghentikan instans DB RDS for SQL Server dalam deployment Multi-AZ tidak didukung.
- Anda tidak dapat menghentikan instans DB yang memiliki replika baca, atau yang merupakan replika baca.
- Anda tidak dapat memodifikasi instans DB yang dihentikan.
- Anda tidak dapat menghapus grup opsi yang terkait dengan instans DB yang dihentikan.
- Anda tidak dapat menghapus grup parameter DB yang terkait dengan instans DB yang dihentikan.

Batasan tambahan berlaku untuk RDS Custom for SQL Server. Untuk informasi selengkapnya, lihat Memulai dan menghentikan instans DB RDS Custom for SQL Server.

Pertimbangan grup opsi dan grup parameter

Anda tidak dapat menghapus opsi persisten (termasuk opsi permanen) dari grup opsi jika ada instans DB yang terkait dengan grup opsi tersebut. Fungsionalitas ini juga berlaku untuk semua instans DB dengan status `stopping`, `stopped`, atau `starting`.

Anda dapat mengubah grup opsi atau grup parameter DB yang terkait dengan instans DB yang dihentikan. Namun, perubahan tidak terjadi sampai Anda memulai instans DB di lain waktu. Jika Anda memilih untuk segera menerapkan perubahan, perubahan akan terjadi saat Anda memulai instans DB. Jika tidak, perubahan terjadi selama periode pemeliharaan berikutnya setelah Anda memulai instans DB.

Pertimbangan alamat IP publik

Saat Anda menghentikan instans DB, instans ini akan mempertahankan titik akhir DNS-nya. Jika Anda menghentikan instans DB yang memiliki alamat IP publik, Amazon RDS akan melepas alamat IP publiknya. Ketika instans DB diaktifkan ulang, instans ini akan memiliki alamat IP publik yang berbeda.

> **Note**
>
> Anda harus selalu terhubung ke instans DB menggunakan titik akhir DNS, bukan alamat IP.

Menghentikan instans DB sementara: langkah-langkah dasar

Anda dapat menghentikan DB menggunakan AWS Management Console, AWS CLI, atau RDS API.

Konsol

Untuk menghentikan instans DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data, lalu pilih instans DB yang ingin Anda hentikan.
3. Untuk Tindakan, pilih Berhenti sementara.

4. Di jendela Hentikan instans DB sementara, pilih konfirmasi bahwa instans DB akan diaktifkan ulang secara otomatis setelah 7 hari.
5. (Opsional) Pilih Simpan instans DB dalam snapshot dan masukkan nama snapshot untuk Nama snapshot. Pilih opsi ini jika Anda ingin membuat snapshot instans DB sebelum menghentikannya.
6. Pilih Berhenti sementara untuk menghentikan instans DB, atau pilih Batalkan untuk membatalkan operasi.

AWS CLI

Untuk menghentikan instance DB dengan menggunakan AWS CLI, panggil stop-db-instanceperintah dengan opsi berikut:

- `--db-instance-identifier` – nama instans DB.

Example

```
aws rds stop-db-instance --db-instance-identifier mydbinstance
```

API RDS

Untuk menghentikan instans DB menggunakan API Amazon RDS, panggil operasi StopDBInstance dengan parameter berikut ini:

- `DBInstanceIdentifier` – nama instans DB.

Memulai instans DB Amazon RDS yang sebelumnya dihentikan

Anda dapat menghentikan instans DB Amazon RDS untuk menghemat biaya sementara. Setelah Anda menghentikan instans DB Anda, Anda dapat memulainya kembali untuk menggunakannya lagi. Untuk detail selengkapnya tentang menghentikan dan memulai instans DB, lihat Menghentikan sementara instans DB Amazon RDS.

Ketika Anda memulai instans DB yang sebelumnya Anda hentikan, instans DB ini mempertahankan informasi tertentu. Informasi ini adalah titik akhir ID, Server Nama Domain (DNS), grup parameter, grup keamanan, dan grup opsi. Ketika Anda memulai instans yang terhenti, Anda akan dikenai biaya jam instans penuh.

Konsol

Untuk memulai instans DB

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data, lalu pilih instans DB yang ingin Anda mulai.
3. Untuk Tindakan, pilih Mulai.

AWS CLI

Untuk memulai instans DB dengan menggunakan AWS CLI, panggil perintah start-db-instance dengan opsi berikut:

- `--db-instance-identifier` – Nama instans DB.

Example

```
aws rds start-db-instance --db-instance-identifier mydbinstance
```

API RDS

Untuk memulai instans DB dengan menggunakan API Amazon RDS, panggil operasi StartDBInstance dengan parameter berikut:

- `DBInstanceIdentifier` – Nama instans DB.

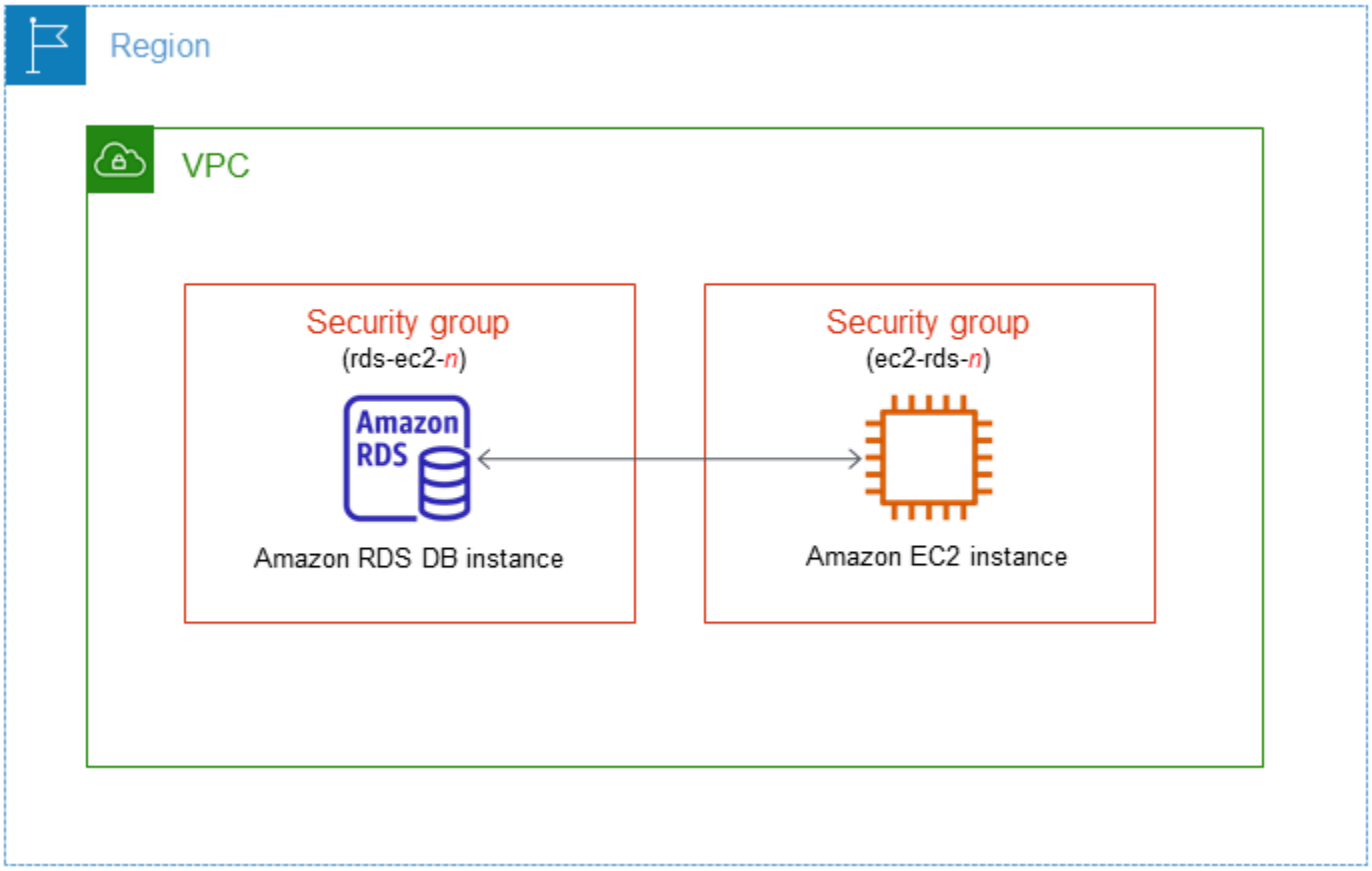
Otomatis menghubungkan sumber daya komputas AWS dan instans DB

Anda dapat menghubungkan instans DB dan AWS sumber daya komputasi seperti instans Amazon Elastic Compute Cloud (Amazon EC2) dan fungsi AWS Lambda secara otomatis.

Topik

Menghubungkan secara otomatis instans EC2 dan instans basis data

Anda dapat menggunakan konsol Amazon RDS untuk menyederhanakan penyiapan koneksi antara instans Amazon Elastic Compute Cloud (Amazon EC2) dan instans basis data. Sering kali, instans basis data Anda berada di subnet privat dan instans EC2 Anda berada dalam VPC di subnet publik. Anda dapat menggunakan klien SQL pada instans EC2 Anda untuk menghubungi instans basis data Anda. Instans EC2 juga dapat menjalankan server atau aplikasi web yang mengakses instans basis data privat Anda. Lihat petunjuk tentang pengaturan koneksi antara instans EC2 dan klaster basis data Multi-AZ di the section called “Menghubungkan instans EC2 dan klaster basis data Multi-AZ”.



Jika Anda ingin menghubungkan dengan instans EC2 yang tidak berada di VPC yang sama dengan instans DB, lihat skenario di Skenario untuk mengakses instans DB di VPC.

Topik

- Ikhtisar konektivitas otomatis dengan instans EC2
- Menghubungkan secara otomatis instans EC2 dan basis data RDS
- Melihat sumber daya komputasi terhubung
- Menghubungi instans basis data yang menjalankan mesin basis data tertentu

Ikhtisar konektivitas otomatis dengan instans EC2

Saat Anda menyiapkan koneksi antara instans EC2 dan basis data RDS, Amazon RDS mengonfigurasikan secara otomatis grup keamanan VPC untuk instans EC2 Anda dan untuk basis data RDS data Anda.

Berikut adalah persyaratan untuk menghubungkan instans EC2 dengan basis data RDS:

- Instans EC2 harus ada di VPC yang sama dengan basis data RDS.

  Jika tidak ada instans EC2 di VPC yang sama, maka konsol menyediakan tautan untuk membuatnya.
- Pengguna yang menyiapkan konektivitas harus memiliki izin untuk melakukan operasi-operasi Amazon EC2 berikut:
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:AuthorizeSecurityGroupIngress`
  - `ec2:CreateSecurityGroup`
  - `ec2:DescribeInstances`
  - `ec2:DescribeNetworkInterfaces`
  - `ec2:DescribeSecurityGroups`
  - `ec2:ModifyNetworkInterfaceAttribute`
  - `ec2:RevokeSecurityGroupEgress`

Jika instans basis data dan instans EC2 berada di Zona Ketersediaan yang berbeda, akun Anda mungkin dikenakan biaya lintas Zona Ketersediaan.

Saat Anda menyiapkan koneksi dengan instans EC2, Amazon RDS bertindak sesuai dengan konfigurasi grup keamanan saat ini yang terkait dengan basis data RDS dan instans EC2, seperti dijelaskan dalam tabel berikut.

Konfigurasi grup keamanan RDS saat ini	Konfigurasi grup keamanan EC2 saat ini	Tindakan RDS
Ada satu atau beberapa grup keamanan yang terkait dengan basis data RDS dengan nama yang cocok dengan pola `rds-ec2-`*n* (dengan *n* berupa angka). Grup keamanan yang cocok dengan pola belum diubah. Grup keamanan ini memiliki hanya satu aturan masuk	Ada satu atau beberapa grup keamanan yang terkait dengan instans EC2 dengan nama yang cocok dengan pola `ec2-rds-`*n* (dengan *n* berupa angka). Grup keamanan yang cocok dengan pola belum diubah. Grup keamanan ini memiliki hanya satu aturan keluar dengan grup keamanan	RDS tidak mengambil tindakan. Koneksi sudah dikonfigu rasikan secara otomatis antara instans EC2 dan basis data RDS. Karena koneksi sudah ada antara instans EC2 dan basis data RDS, grup keamanan tidak diubah.

Konfigurasi grup keamanan RDS saat ini	Konfigurasi grup keamanan EC2 saat ini	Tindakan RDS
dengan grup keamanan VPC instans EC2 sebagai sumbernya.	VPC basis data RDS sebagai sumbernya.	

Konfigurasi grup keamanan RDS saat ini	Konfigurasi grup keamanan EC2 saat ini	Tindakan RDS
Salah satu syarat berikut dipenuhi: • Tidak ada grup keamanan yang terkait dengan basis data RDS dengan nama yang cocok dengan pola `rds-ec2-`*n*. • Ada satu atau beberapa grup keamanan yang terkait dengan basis data RDS dengan nama yang cocok dengan pola `rds-ec2-`*n*. Namun, Amazon RDS tidak dapat menggunakan satu pun grup keamanan ini untuk koneksi dengan instans EC2. Amazon RDS tidak dapat menggunakan grup keamanan yang tidak memiliki satu aturan masuk dengan grup keamanan VPC instans EC2 sebagai sumbernya. Amazon RDS juga tidak dapat menggunakan grup keamanan yang telah diubah. Contoh-contoh perubahan meliputi penambahan aturan atau pengubahan port aturan yang ada.	Salah satu syarat berikut dipenuhi: • Tidak ada grup keamanan yang terkait dengan instans EC2 dengan nama yang cocok dengan pola `ec2-rds-`*n*. • Ada satu atau beberapa grup keamanan yang terkait dengan instans EC2 dengan nama yang cocok dengan pola `ec2-rds-`*n*. Namun, Amazon RDS tidak dapat menggunakan satu pun grup keamanan ini untuk koneksi dengan basis data RDS. Amazon RDS tidak dapat menggunakan grup keamanan yang tidak memiliki satu aturan keluar dengan grup keamanan VPC basis data RDS sebagai sumbernya. Amazon RDS juga tidak dapat menggunakan grup keamanan yang telah diubah.	RDS action: create new security groups

Konfigurasi grup keamanan RDS saat ini	Konfigurasi grup keamanan EC2 saat ini	Tindakan RDS
Ada satu atau beberapa grup keamanan yang terkait dengan basis data RDS dengan nama yang cocok dengan pola `rds-ec2-`*n*. Grup keamanan yang cocok dengan pola belum diubah. Grup keamanan ini memiliki hanya satu aturan masuk dengan grup keamanan VPC instans EC2 sebagai sumbernya.	Ada satu atau beberapa grup keamanan yang terkait dengan instans EC2 dengan nama yang cocok dengan pola `ec2-rds-`*n*. Namun, Amazon RDS tidak dapat menggunak an satu pun grup keamanan ini untuk koneksi dengan basis data RDS. Amazon RDS tidak dapat menggunakan grup keamanan yang tidak memiliki satu aturan keluar dengan grup keamanan VPC basis data RDS sebagai sumbernya. Amazon RDS juga tidak dapat menggunakan grup keamanan yang telah diubah.	RDS action: create new security groups
Ada satu atau beberapa grup keamanan yang terkait dengan basis data RDS dengan nama yang cocok dengan pola `rds-ec2-`*n*. Grup keamanan yang cocok dengan pola belum diubah. Grup keamanan ini memiliki hanya satu aturan masuk dengan grup keamanan VPC instans EC2 sebagai sumbernya.	Ada grup keamanan EC2 yang valid untuk koneksi, tetapi tidak terkait dengan instans EC2. Grup keamanan ini memiliki nama yang cocok dengan pola `ec2-rds-`*n*. Grup itu belum diubah. Grup ini memiliki hanya satu aturan keluar dengan grup keamanan VPC basis data RDS sebagai sumbernya.	RDS action: associate EC2 security group

Konfigurasi grup keamanan RDS saat ini	Konfigurasi grup keamanan EC2 saat ini	Tindakan RDS
Salah satu syarat berikut dipenuhi: • Tidak ada grup keamanan yang terkait dengan basis data RDS dengan nama yang cocok dengan pola `rds-ec2-`*n*. • Ada satu atau beberapa grup keamanan yang terkait dengan basis data RDS dengan nama yang cocok dengan pola `rds-ec2-`*n*. Namun, Amazon RDS tidak dapat menggunakan satu pun grup keamanan ini untuk koneksi dengan instans EC2. Amazon RDS tidak dapat menggunakan grup keamanan yang tidak memiliki satu aturan masuk dengan grup keamanan VPC instans EC2 sebagai sumbernya. Amazon RDS juga tidak dapat menggunak an grup keamanan yang telah diubah.	Ada satu atau beberapa grup keamanan yang terkait dengan instans EC2 dengan nama yang cocok dengan pola `ec2-rds-`*n*. Grup keamanan yang cocok dengan pola belum diubah. Grup keamanan ini memiliki hanya satu aturan keluar dengan grup keamanan VPC basis data RDS sebagai sumbernya.	RDS action: create new security groups

Tindakan RDS : membuat grup keamanan baru

Amazon RDS melakukan tindakan-tindakan berikut:

- Membuat grup keamanan baru yang cocok dengan pola `rds-ec2-`*n*. Grup keamanan ini memiliki aturan masuk dengan grup keamanan VPC instans EC2 sebagai sumbernya. Grup keamanan ini dikaitkan dengan basis data RDS dan memungkinkan instans EC2 mengakses basis data RDS.
- Membuat grup keamanan baru yang cocok dengan pola `ec2-rds-`*n*. Grup keamanan ini memiliki aturan keluar dengan grup keamanan VPC dari cluster RDS sebagai target. Grup keamanan ini dikaitkan dengan instans EC2 dan memungkinkan instans EC2 mengirim lalu lintas ke basis data RDS.

Tindakan RDS : mengaitkan grup keamanan EC2

Amazon RDS mengaitkan grup keamanan EC2 yang valid dan sudah ada dengan instans EC2. Grup keamanan ini memungkinkan instans EC2 mengirim lalu lintas ke basis data RDS.

Menghubungkan secara otomatis instans EC2 dan basis data RDS

Sebelum menyiapkan koneksi antara instans EC2 dan basis data RDS, pastikan untuk memenuhi persyaratan yang dijelaskan di Ikhtisar konektivitas otomatis dengan instans EC2.

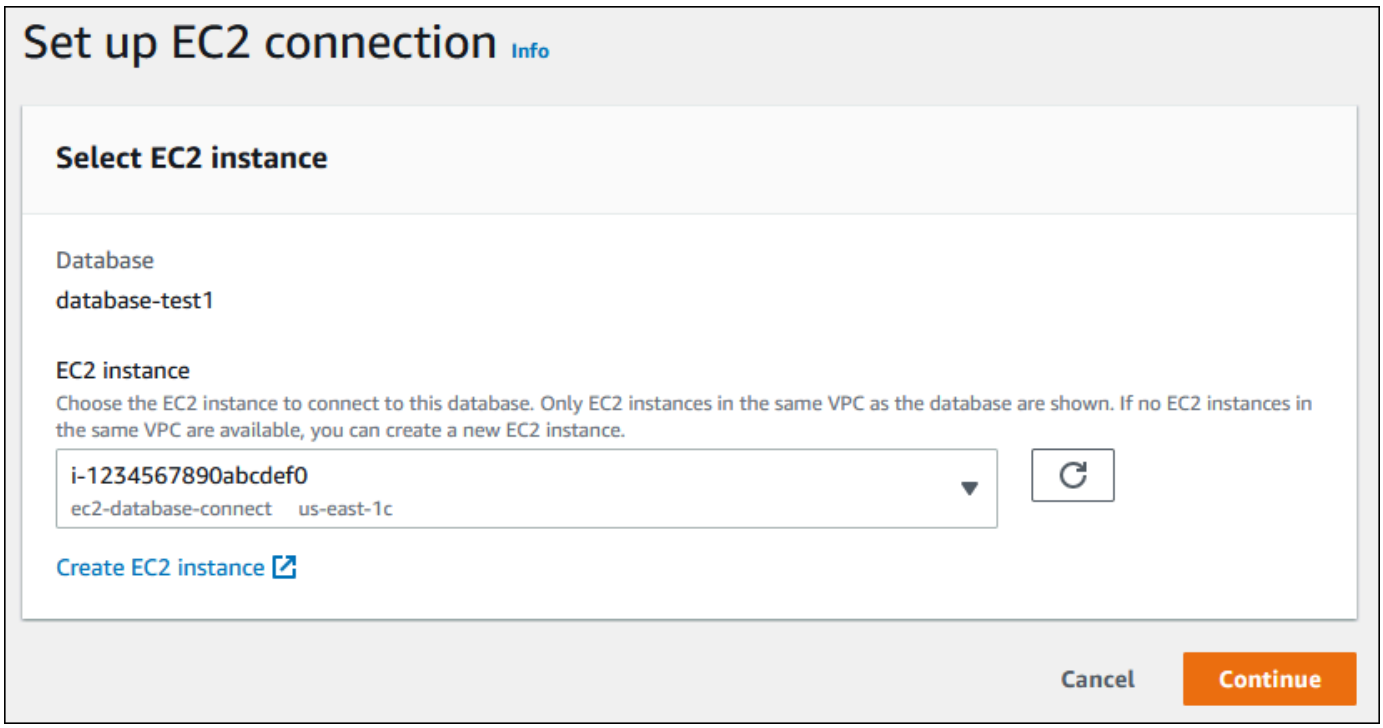
Jika Anda membuat perubahan pada grup keamanan setelah mengonfigurasikan konektivitas, perubahan itu dapat memengaruhi koneksi antara instans EC2 dan basis data RDS.

> **Note**
>
> Anda hanya dapat menyiapkan koneksi antara instans EC2 dan basis data RDS secara otomatis dengan menggunakan AWS Management Console. Anda tidak dapat mengatur koneksi secara otomatis dengan AWS CLI atau RDS API.

Untuk menghubungkan secara otomatis instans EC2 dan basis data RDS

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data, lalu pilih basis data RDS.
3. Untuk Tindakan, pilih Siapkan koneksi EC2.

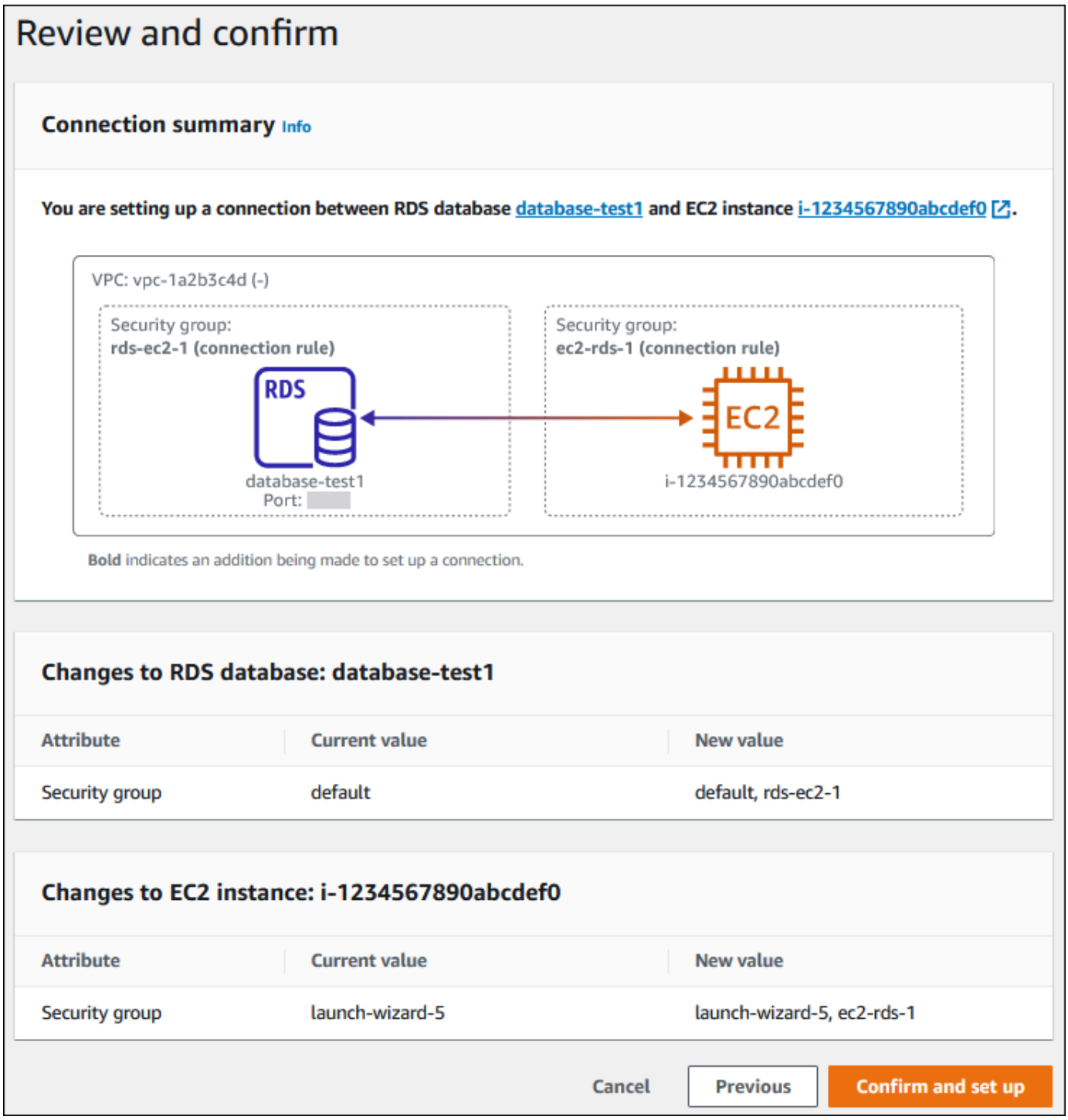
   Halaman Siapkan koneksi EC2 muncul.
4. Pada halaman Siapkan koneksi EC2, pilih instans EC2.



Jika tidak ada instans EC2 di VPC yang sama, pilih Buat instans EC2 untuk membuatnya. Dalam hal ini, pastikan bahwa instans EC2 baru berada di VPC yang sama dengan basis data RDS.

5. Pilih Lanjutkan.

   Halaman Tinjau dan tegaskan muncul.



6. Pada halaman Tinjau dan tegaskan, tinjau perubahan yang akan dilakukan RDS untuk menyiapkan konektivitas dengan instans EC2.

   Jika perubahan sudah benar, pilih Tegaskan dan siapkan.

   Jika perubahan masih salah, pilih Sebelumnya atau Batalkan.

Melihat sumber daya komputasi terhubung

Anda dapat menggunakan AWS Management Console untuk melihat sumber daya komputasi yang terhubung ke database RDS DB cluster. Sumber daya yang ditampilkan meliputi koneksi sumber daya komputasi yang disiapkan secara otomatis. Anda dapat menyiapkan secara otomatis konektivitas dengan sumber daya komputasi dengan cara berikut:

- Anda dapat memilih sumber daya komputasi saat membuat basis data.

  Lihat informasi yang lebih lengkap di Membuat instans DB Amazon RDS dan Membuat klaster DB Multi-AZ.
- Anda dapat menyiapkan konektivitas antara basis data yang ada dan sumber daya komputasi.

  Untuk informasi selengkapnya, lihat Menghubungkan secara otomatis instans EC2 dan basis data RDS.

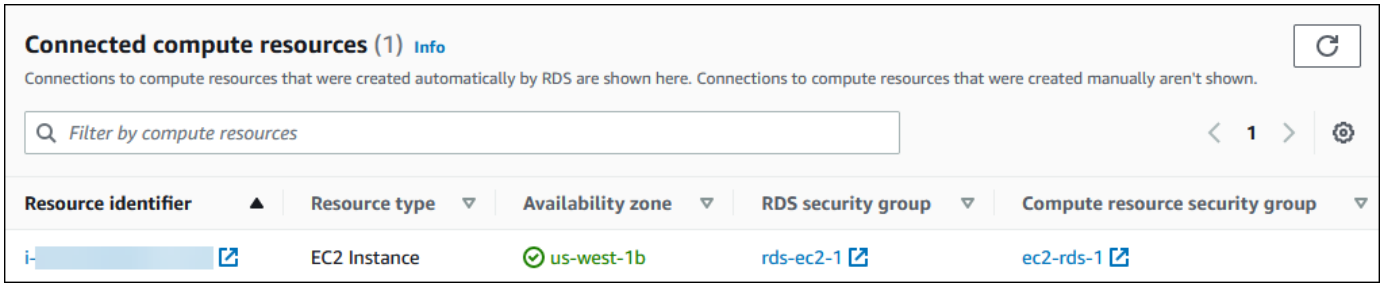
Sumber daya komputasi yang tercantum tidak menyertakan sumber daya yang dihubungkan secara manual dengan basis data. Misalnya, Anda dapat mengizinkan sumber daya komputasi untuk mengakses basis data secara manual dengan menambahkan aturan ke grup keamanan VPC yang terkait dengan basis data.

Agar sumber daya komputasi tercantum, syarat-syarat berikut harus dipenuhi:

- Nama grup keamanan yang terkait dengan sumber daya komputasi cocok dengan pola `ec2-rds-`*n* (dengan *n* berupa angka).
- Grup keamanan yang terkait dengan sumber daya komputasi memiliki aturan keluar dengan rentang port diatur ke port yang digunakan basis data RDS.
- Grup keamanan yang terkait dengan sumber daya komputasi memiliki aturan keluar dengan sumber yang diatur ke grup keamanan yang terkait dengan basis data RDS.
- Nama grup keamanan yang terkait dengan basis data RDS cocok dengan pola `rds-ec2-`*n* (dengan *n* berupa angka).
- Grup keamanan yang terkait dengan basis data RDS memiliki aturan masuk dengan rentang port yang diatur ke port yang digunakan basis data RDS.
- Grup keamanan yang terkait dengan basis data RDS memiliki aturan masuk dengan sumber yang diatur ke grup keamanan yang terkait dengan sumber daya komputasi.

Untuk melihat sumber daya komputasi yang menghubungi basis data RDS

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data, lalu pilih nama basis data RDS.
3. Pada tab Konektivitas dan keamanan, lihat sumber daya komputasi di Sumber daya komputasi terhubung.



Menghubungi instans basis data yang menjalankan mesin basis data tertentu

Untuk informasi tentang cara menghubungi instans basis data yang menjalankan mesin basis data tertentu, ikuti petunjuk untuk mesin basis data Anda:

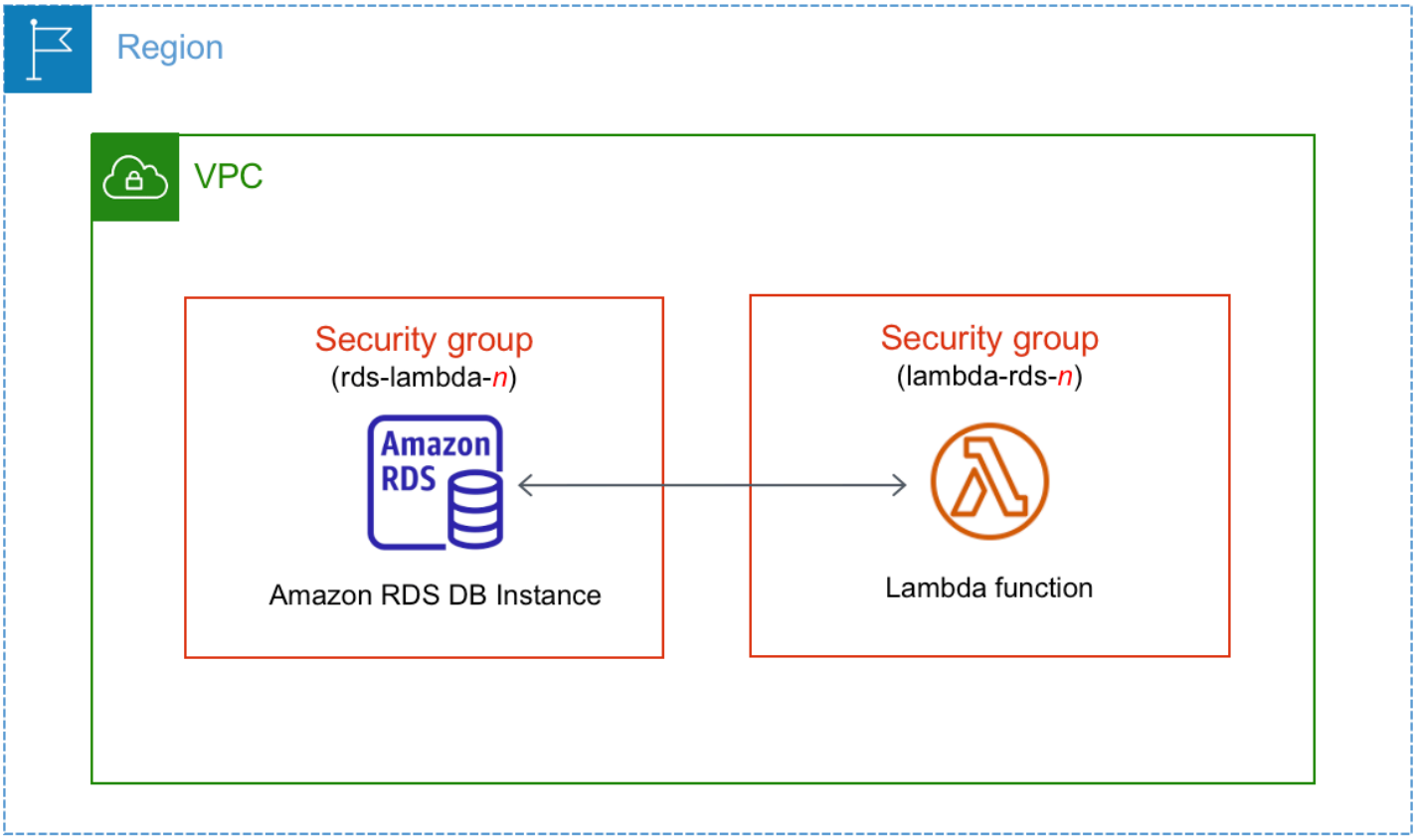
- Menghubungkan ke instans DB yang menjalankan mesin basis data MariaDB
- Menghubungkan ke instans DB yang menjalankan mesin basis data Microsoft SQL Server
- Menghubungkan ke instans DB yang menjalankan mesin basis data MySQL
- Menghubungkan ke instans RDS for Oracle DB
- Menghubungkan ke instans DB yang menjalankan mesin basis data PostgreSQL

Menghubungkan secara otomatis fungsi Lambda dan instans basis data

Anda dapat menggunakan konsol Amazon RDS untuk menyederhanakan penyiapan koneksi antara fungsi Lambda dan instans basis data. Sering kali, instans basis data Anda berada di subnet privat dalam VPC. Fungsi Lambda dapat digunakan oleh aplikasi untuk mengakses instans basis data privat Anda.

Lihat petunjuk tentang cara menyiapkan koneksi antara fungsi Lambda dan klaster basis data Multi-AZ di the section called “Menghubungkan fungsi Lambda dan klaster basis data Multi-AZ”.

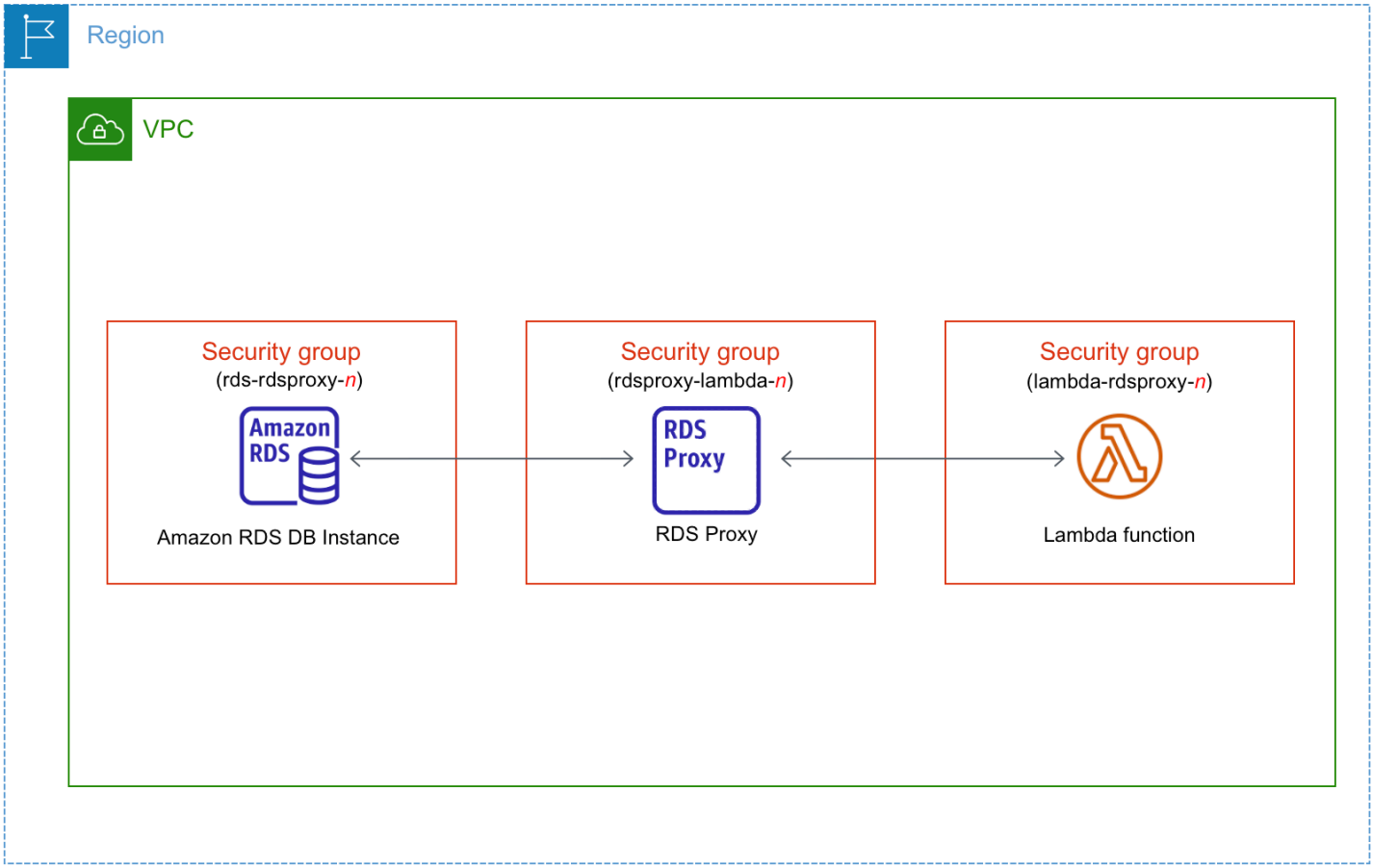
Gambar berikut menunjukkan koneksi langsung antara instans basis data dan fungsi Lambda.



Anda dapat menyiapkan koneksi antara fungsi Lambda dan instans basis data Anda melalui Proksi RDS untuk meningkatkan kinerja dan ketangguhan basis data Anda. Sering kali, fungsi Lambda membuat koneksi basis data yang singkat tetapi sering yang menarik manfaat dari penghimpunan koneksi yang ditawarkan Proksi RDS. Anda dapat memanfaatkan autentikasi AWS Identity and Access Management (IAM) apa pun yang Anda miliki untuk fungsi Lambda, alih-alih mengelola kredensial basis data dalam kode aplikasi Lambda. Lihat informasi yang lebih lengkap di Menggunakan Proksi Amazon RDS.

Saat Anda menggunakan konsol untuk menghubungi proksi yang ada, Amazon RDS memperbarui grup keamanan proksi untuk mengizinkan koneksi dari instans basis data dan fungsi Lambda Anda.

Anda juga dapat membuat proksi baru dari halaman konsol yang sama. Saat Anda membuat proksi di konsol, untuk mengakses instans basis data, Anda harus memasukkan kredensial basis data Anda atau memilih rahasia AWS Secrets Manager.



Topik

- Ikhtisar konektivitas otomatis dengan fungsi Lambda
- Menghubungkan secara otomatis fungsi Lambda dan basis data RDS
- Melihat sumber daya komputasi terhubung

Ikhtisar konektivitas otomatis dengan fungsi Lambda

Berikut adalah persyaratan untuk menghubungkan fungsi Lambda dengan instans basis data RDS:

- Fungsi Lambda harus ada di VPC yang sama dengan instans basis data.
- Pengguna yang menyiapkan konektivitas harus memiliki izin untuk melakukan operasi-operasi Amazon RDS, Amazon EC2, Lambda, Secrets Manager, dan IAM berikut:
  - Amazon RDS
    - `rds:CreateDBProxies`
    - `rds:DescribeDBInstances`

  - `rds:DescribeDBProxies`
  - `rds:ModifyDBInstance`
  - `rds:ModifyDBProxy`
  - `rds:RegisterProxyTargets`
- Amazon EC2
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:AuthorizeSecurityGroupIngress`
  - `ec2:CreateSecurityGroup`
  - `ec2:DeleteSecurityGroup`
  - `ec2:DescribeSecurityGroups`
  - `ec2:RevokeSecurityGroupEgress`
  - `ec2:RevokeSecurityGroupIngress`
- Lambda
  - `lambda:CreateFunctions`
  - `lambda:ListFunctions`
  - `lambda:UpdateFunctionConfiguration`
- Secrets Manager
  - `secretsmanager:CreateSecret`
  - `secretsmanager:DescribeSecret`
- IAM
  - `iam:AttachPolicy`
  - `iam:CreateRole`
  - `iam:CreatePolicy`
- AWS KMS
  - `kms:describeKey`

> **Note**
>
> Jika instans basis data dan fungsi Lambda berada di Zona Ketersediaan yang berbeda, akun Anda mungkin dikenakan biaya lintas Zona Ketersediaan.

Saat Anda menyiapkan koneksi antara fungsi Lambda dan basis data RDS, Amazon RDS mengonfigurasikan grup keamanan VPC untuk fungsi Anda dan untuk instans basis data Anda. Jika Anda menggunakan Proksi RDS, maka Amazon RDS juga mengonfigurasikan grup keamanan VPC untuk proksi itu. Amazon RDS bertindak sesuai dengan konfigurasi grup keamanan saat ini yang terkait dengan instans basis, fungsi Lambda, dan proksi, seperti dijelaskan dalam tabel berikut.

Konfigurasi grup keamanan RDS saat ini	Konfigurasi grup keamanan Lambda saat ini	Konfigurasi grup keamanan proksi saat ini	Tindakan RDS
Ada satu atau beberapa grup keamanan yang terkait dengan instans basis data dengan nama yang cocok dengan pola `rds-lambda-` *n* atau jika proksi sudah terhubung dengan instans basis data Anda, RDS memeriksa apakah `TargetHealth` proksi terkait adalah `AVAILABLE` . Grup keamanan yang cocok dengan pola belum diubah. Grup keamanan ini memiliki hanya satu aturan masuk dengan grup keamanan VPC proksi atau fungsi Lambda sebagai sumbernya.	Ada satu atau beberapa grup keamanan yang terkait dengan fungsi Lambda dengan nama yang cocok dengan pola `lambda-rds-` *n* atau `lambda-rdsproxy-` *n* (dengan *n* berupa angka). Grup keamanan yang cocok dengan pola belum diubah. Grup keamanan ini hanya memiliki satu aturan keluar dengan grup keamanan VPC instans basis data atau proksi sebagai tujuannya.	Ada satu atau beberapa grup keamanan yang terkait dengan proksi dengan nama yang cocok dengan pola `rdsproxy-lambda-` *n* (dengan *n* berupa angka). Grup keamanan yang cocok dengan pola belum diubah. Grup keamanan ini memiliki aturan masuk dan aturan keluar dengan grup keamanan VPC fungsi Lambda dan instans basis data.	Amazon RDS tidak mengambil tindakan. Koneksi sudah dikonfigurasikan secara otomatis antara fungsi Lambda, proksi (opsional), dan instans basis data. Karena koneksi sudah ada antara fungsi, proksi, dan basis data, grup keamanan tidak diubah.

Konfigurasi grup keamanan RDS saat ini	Konfigurasi grup keamanan Lambda saat ini	Konfigurasi grup keamanan proksi saat ini	Tindakan RDS
Salah satu syarat berikut dipenuhi: • Tidak ada grup keamanan yang terkait dengan instans basis data dengan nama yang cocok dengan pola `rds-lambda-` *n* atau jika `TargetHealth` proksi terkait adalah `AVAILABLE`. • Ada satu atau beberapa grup keamanan yang terkait dengan instans basis data dengan nama yang cocok dengan pola `rds-lambda-` *n* atau jika `TargetHealth` proksi terkait adalah `AVAILABLE`. Namun, tidak satu pun grup keamanan ini dapat digunakan untuk koneksi	Salah satu syarat berikut dipenuhi: • Tidak ada grup keamanan yang terkait dengan fungsi Lambda dengan nama yang cocok dengan pola `lambda-rds-` *n* atau `lambda-rdsproxy-` *n*. • Ada satu atau beberapa grup keamanan yang terkait dengan fungsi Lambda dengan nama yang cocok dengan pola `lambda-rds-` *n* atau `lambda-rdsproxy-` *n*. Namun, Amazon RDS tidak dapat menggunakan satu pun grup keamanan ini untuk koneksi dengan instans basis data. Amazon RDS tidak dapat menggunakan	Salah satu syarat berikut dipenuhi: • Tidak ada grup keamanan yang terkait dengan proksi dengan nama yang cocok dengan pola `rdsproxy-lambda-` *n*. • Ada satu atau beberapa grup keamanan yang terkait dengan proksi dengan nama yang cocok dengan `rdsproxy-lambda-` *n*. Namun, Amazon RDS tidak dapat menggunakan satu pun grup keamanan ini untuk koneksi dengan instans basis data atau fungsi Lambda. Amazon RDS tidak dapat menggunakan grup keamanan	RDS action: create new security groups

Konfigurasi grup keamanan RDS saat ini	Konfigurasi grup keamanan Lambda saat ini	Konfigurasi grup keamanan proksi saat ini	Tindakan RDS
dengan fungsi Lambda. Amazon RDS tidak dapat menggunakan grup keamanan yang tidak memiliki satu aturan masuk dengan grup keamanan VPC proksi atau fungsi Lambda sebagai sumbernya. Amazon RDS juga tidak dapat menggunak an grup keamanan yang telah diubah. Contoh-contoh perubahan meliputi penambahan aturan atau pengubahan porta aturan yang ada.	grup keamanan yang tidak memiliki satu aturan keluar dengan grup keamanan VPC instans basis data atau proksi sebagai tujuannya. Amazon RDS juga tidak dapat menggunakan grup keamanan yang telah diubah.	yang tidak memiliki aturan masuk dan keluar dengan grup keamanan VPC instans basis data dan fungsi Lambda. Amazon RDS juga tidak dapat menggunakan grup keamanan yang telah diubah.	

Konfigurasi grup keamanan RDS saat ini	Konfigurasi grup keamanan Lambda saat ini	Konfigurasi grup keamanan proksi saat ini	Tindakan RDS
Ada satu atau beberapa grup keamanan yang terkait dengan instans basis data dengan nama yang cocok dengan pola `rds-lambda-` *n* atau jika `TargetHealth` proksi terkait adalah `AVAILABLE` . Grup keamanan yang cocok dengan pola belum diubah. Grup keamanan ini memiliki hanya satu aturan masuk dengan grup keamanan VPC proksi atau fungsi Lambda sebagai sumbernya.	Ada satu atau beberapa grup keamanan yang terkait dengan fungsi Lambda dengan nama yang cocok dengan pola `lambda-rds-` *n* atau `lambda-rdsproxy-` *n*. Namun, Amazon RDS tidak dapat menggunakan satu pun grup keamanan ini untuk koneksi dengan instans basis data. Amazon RDS tidak dapat menggunakan grup keamanan yang tidak memiliki satu aturan keluar dengan grup keamanan VPC instans basis data atau proksi sebagai tujuannya. Amazon RDS juga tidak dapat menggunakan grup keamanan yang telah diubah.	Ada satu atau beberapa grup keamanan yang terkait dengan proksi dengan nama yang cocok dengan pola `rdsproxy-lambda-` *n*. Namun, Amazon RDS tidak dapat menggunakan satu pun grup keamanan ini untuk koneksi dengan instans basis data atau fungsi Lambda. Amazon RDS tidak dapat menggunakan grup keamanan yang tidak memiliki aturan masuk dan keluar dengan grup keamanan VPC instans basis data dan fungsi Lambda. Amazon RDS juga tidak dapat menggunakan grup keamanan yang telah diubah.	RDS action: create new security groups

Konfigurasi grup keamanan RDS saat ini	Konfigurasi grup keamanan Lambda saat ini	Konfigurasi grup keamanan proksi saat ini	Tindakan RDS
Ada satu atau beberapa grup keamanan yang terkait dengan instans basis data dengan nama yang cocok dengan pola `rds-lambda-` *n* atau jika `TargetHealth` proksi terkait adalah `AVAILABLE` . Grup keamanan yang cocok dengan pola belum diubah. Grup keamanan ini memiliki hanya satu aturan masuk dengan grup keamanan VPC proksi atau fungsi Lambda sebagai sumbernya.	Ada grup keamanan Lambda yang valid untuk koneksi, tetapi tidak dikaitkan dengan fungsi Lambda. Grup keamanan ini memiliki nama yang cocok dengan pola `lambda-rds-` *n* atau `lambda-rdsproxy-` *n*. Grup itu belum diubah. Grup hanya memiliki satu aturan keluar dengan grup keamanan VPCinstans basis data atau proksi sebagai tujuannya.	Ada grup keamanan proksi yang valid untuk koneksi, tetapi tidak dikaitkan dengan proksi. Grup keamanan ini memiliki nama yang cocok dengan pola `rdsproxy-lambda-` *n*. Grup itu belum diubah. Grup ini memiliki aturan masuk dan aturan keluar dengan grup keamanan VPC instans basis data dan fungsi Lambda.	RDS action: associate Lambda security group

Konfigurasi grup keamanan RDS saat ini	Konfigurasi grup keamanan Lambda saat ini	Konfigurasi grup keamanan proksi saat ini	Tindakan RDS
Salah satu syarat berikut dipenuhi: • Tidak ada grup keamanan yang terkait dengan instans basis data dengan nama yang cocok dengan pola `rds-lambda-` *n* atau jika `TargetHealth` proksi terkait adalah `AVAILABLE`. • Ada satu atau beberapa grup keamanan yang terkait dengan instans basis data dengan nama yang cocok dengan pola `rds-lambda-` *n* atau jika `TargetHealth` proksi terkait adalah `AVAILABLE`. Namun, Amazon RDS tidak dapat menggunakan satu pun grup keamanan ini untuk koneksi	Ada satu atau beberapa grup keamanan yang terkait dengan fungsi Lambda dengan nama yang cocok dengan pola `lambda-rds-` *n* atau `lambda-rdsproxy-` *n*. Grup keamanan yang cocok dengan pola belum diubah. Grup keamanan ini hanya memiliki satu aturan keluar dengan grup keamanan VPC instans basis data atau proksi sebagai tujuannya.	Ada satu atau beberapa grup keamanan yang terkait dengan proksi dengan nama yang cocok dengan pola `rdsproxy-lambda-` *n*. Grup keamanan yang cocok dengan pola belum diubah. Grup keamanan ini memiliki aturan masuk dan keluar dengan grup keamanan VPC instans basis data dan fungsi Lambda.	RDS action: create new security groups

Konfigurasi grup keamanan RDS saat ini	Konfigurasi grup keamanan Lambda saat ini	Konfigurasi grup keamanan proksi saat ini	Tindakan RDS
dengan proksi atau fungsi Lambda. Amazon RDS tidak dapat menggunakan grup keamanan yang tidak memiliki satu aturan masuk dengan grup keamanan VPC proksi atau fungsi Lambda sebagai sumbernya. Amazon RDS juga tidak dapat menggunakan grup keamanan yang telah diubah.			

Konfigurasi grup keamanan RDS saat ini	Konfigurasi grup keamanan Lambda saat ini	Konfigurasi grup keamanan proksi saat ini	Tindakan RDS
Salah satu syarat berikut dipenuhi: • Tidak ada grup keamanan yang terkait dengan instans basis data dengan nama yang cocok dengan pola `rds-lambda-` *n* atau jika `TargetHealth` proksi terkait adalah `AVAILABLE`. • Ada satu atau beberapa grup keamanan yang terkait dengan instans basis data dengan nama yang cocok dengan pola `rds-lambda-` *n* atau jika `TargetHealth` proksi terkait adalah `AVAILABLE`. Namun, Amazon RDS tidak dapat menggunakan satu pun grup keamanan ini untuk koneksi	Salah satu syarat berikut dipenuhi: • Tidak ada grup keamanan yang terkait dengan fungsi Lambda dengan nama yang cocok dengan pola `lambda-rds-` *n* atau `lambda-rdsproxy-` *n*. • Ada satu atau beberapa grup keamanan yang terkait dengan fungsi Lambda dengan nama yang cocok dengan pola `lambda-rds-` *n* atau `lambda-rdsproxy-` *n*. Namun, Amazon RDS tidak dapat menggunakan satu pun grup keamanan ini untuk koneksi dengan instans basis data. Amazon RDS tidak dapat menggunakan	Salah satu syarat berikut dipenuhi: • Tidak ada grup keamanan yang terkait dengan proksi dengan nama yang cocok dengan pola `rdsproxy-lambda-` *n*. • Ada satu atau beberapa grup keamanan yang terkait dengan proksi dengan nama yang cocok dengan `rdsproxy-lambda-` *n*. Namun, Amazon RDS tidak dapat menggunakan satu pun grup keamanan ini untuk koneksi dengan instans basis data atau fungsi Lambda. Amazon RDS tidak dapat menggunak an grup keamanan	RDS action: create new security groups

Konfigurasi grup keamanan RDS saat ini	Konfigurasi grup keamanan Lambda saat ini	Konfigurasi grup keamanan proksi saat ini	Tindakan RDS
dengan proksi atau fungsi Lambda. Amazon RDS tidak dapat menggunakan grup keamanan yang tidak memiliki satu aturan masuk dengan grup keamanan VPC proksi atau fungsi Lambda sebagai sumbernya. Amazon RDS juga tidak dapat menggunakan grup keamanan yang telah diubah.	grup keamanan yang tidak memiliki satu aturan keluar dengan grup keamanan VPC instans basis data atau proksi sebagai sumbernya. Amazon RDS juga tidak dapat menggunakan grup keamanan yang telah diubah.	yang tidak memiliki aturan masuk dan keluar dengan grup keamanan VPC instans basis data dan fungsi Lambda. Amazon RDS juga tidak dapat menggunakan grup keamanan yang telah diubah.	

Tindakan RDS : membuat grup keamanan baru

Amazon RDS melakukan tindakan-tindakan berikut:

- Membuat grup keamanan baru yang cocok dengan pola `rds-lambda-`*n* atau `rds-rdsproxy-`*n* (jika Anda memilih untuk menggunakan Proksi RDS). Grup keamanan ini memiliki aturan masuk dengan grup keamanan VPC fungsi Lambda atau proksi sebagai sumbernya. Grup keamanan ini dikaitkan dengan instans basis data dan memungkinkan fungsi atau proksi mengakses instans basis data.
- Membuat grup keamanan baru yang cocok dengan pola `lambda-rds-`*n* atau `lambda-rdsproxy-`*n*. Grup keamanan ini memiliki aturan keluar dengan grup keamanan VPC instans basis data atau proksi sebagai tujuannya. Grup keamanan ini dikaitkan dengan fungsi Lambda dan memungkinkan fungsi mengirim lalu lintas ke instans basis data atau mengirim lalu lintas melalui proksi.

- Membuat grup keamanan baru yang cocok dengan pola `rdsproxy-lambda-`*n*. Grup keamanan ini memiliki aturan masuk dan keluar dengan grup keamanan VPC instans basis data dan fungsi Lambda.

Tindakan RDS : mengaitkan grup keamanan Lambda

Amazon RDS mengaitkan grup keamanan Lambda yang valid dan sudah ada dengan fungsi Lambda. Grup keamanan ini memungkinkan fungsi mengirim lalu lintas ke instans basis data atau mengirim lalu lintas melalui proksi.

Menghubungkan secara otomatis fungsi Lambda dan basis data RDS

Anda dapat menggunakan konsol Amazon RDS untuk menghubungkan secara otomatis fungsi Lambda dengan instans basis dataklaster basis data Anda. Ini menyederhanakan proses penyiapan koneksi di antara sumber daya-sumber daya ini.

Anda juga dapat menggunakan Proksi RDS untuk menyertakan proksi dalam koneksi Anda. Fungsi Lambda membuat koneksi basis data yang singkat tetapi sering yang menarik manfaat dari penghimpunan koneksi yang ditawarkan Proksi RDS. Anda juga dapat menggunakan sebarang autentikasi IAM yang Anda miliki untuk fungsi Lambda, alih-alih mengelola kredensial basis data dalam kode aplikasi Lambda.

Anda dapat menghubungkan instans basis data yang ada dengan fungsi Lambda baru dan lama dengan menggunakan halaman Siapkan koneksi Lambda. Proses penyiapan menyiapkan secara otomatis grup keamanan yang diperlukan untuk Anda.

Sebelum menyiapkan koneksi antara fungsi Lambda dan instans basis data, pastikan bahwa:

- Fungsi Lambda dan instans basis data Anda berada di VPC yang sama.
- Anda memiliki izin-izin yang tepat untuk akun pengguna Anda. Lihat informasi lebih lanjut tentang persyaratan di Ikhtisar konektivitas otomatis dengan fungsi Lambda.

Jika Anda mengubah grup keamanan setelah mengonfigurasikan konektivitas, perubahan itu dapat memengaruhi koneksi antara fungsi Lambda dan instans basis data.

> **Note**
>
> Anda dapat menyiapkan secara otomatis koneksi antara instans basis data dan fungsi Lambda hanya di AWS Management Console. Untuk menghubungkan fungsi Lambda, instans basis data harus dalam keadaan Tersedia.

Untuk menghubungkan secara otomatis fungsi Lambda dan instans basis data

<result>

Setelah memastikan penyiapan, Amazon RDS memulai proses menghubungkan fungsi Lambda, Proksi RDS (jika Anda menggunakan proksi), dan instans basis data. Konsol menampilkan kotak dialog Detail koneksi, yang mencantumkan perubahan grup keamanan yang memungkinkan koneksi di antara sumber daya-sumber daya Anda.
</result>

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data, lalu pilih instans basis data yang ingin Anda hubungkan dengan fungsi Lambda.
3. Untuk Tindakan, pilih Siapkan koneksi Lambda.
4. Pada halaman Siapkan koneksi Lambda, di bawah Pilih fungsi Lambda, lakukan salah satu hal berikut:
   - Jika Anda memiliki fungsi Lambda yang ada di VPC yang sama dengan instans basis data Anda, pilih Pilih fungsi yang ada, lalu pilih fungsi itu.
   - Jika Anda tidak memiliki fungsi Lambda di VPC yang sama, pilih Buat fungsi baru, lalu masukkan Nama fungsi. Runtime bawaan diatur ke Nodejs.18. Anda dapat mengubah setelan untuk fungsi Lambda baru di konsol Lambda setelah menyelesaikan penyiapan koneksi.
5. (Opsional) Di bawah Proksi RDS, pilih Hubungkan lewat Proksi RDS, lalu lakukan salah satu hal berikut:
   - Jika Anda sudah memiliki proksi yang ingin Anda gunakan, pilih Pilih proksi yang ada, lalu pilih proksi itu.
   - Jika Anda tidak memiliki proksi, dan Anda ingin Amazon RDS membuatnya secara otomatis untuk Anda, pilih Buat proksi baru. Lalu, untuk Kredensial basis data, lakukan salah satu langkah berikut:

a. Pilih Nama pengguna dan kata sandi basis data, lalu masukkan Nama pengguna dan Kata sandi untuk instans basis data Anda.

b. Pilih Rahasia Secrets Manager. Kemudian, untuk Pilih rahasia, pilih rahasia AWS Secrets Manager. Jika Anda tidak memiliki rahasia Secrets Manager, pilih Buat rahasia Secrets Manager baru untuk membuat rahasia baru. Setelah Anda membuat rahasia, untuk Pilih rahasia, pilih rahasia baru.

Setelah Anda membuat proksi baru, pilih Pilih proksi yang ada, lalu pilih proksi. Perhatikan bahwa mungkin perlu beberapa waktu sebelum proksi Anda tersedia untuk koneksi.

6. (Opsional) Perluas Ringkasan koneksi dan periksa pembaruan yang disorot untuk sumber daya Anda.
7. Pilih Siapkan.

Melihat sumber daya komputasi terhubung

Anda dapat menggunakan AWS Management Console untuk melihat fungsi Lambda yang terhubung dengan instans basis data Anda. Sumber daya yang ditampilkan meliputi koneksi sumber daya komputasi yang disiapkan secara otomatis oleh Amazon RDS.

Sumber daya komputasi tercantum yang tidak menyertakan sumber daya yang dihubungkan secara manual dengan instans basis data. Misalnya, Anda dapat mengizinkan sumber daya komputasi untuk mengakses instans basis data Anda secara manual dengan menambahkan aturan ke grup keamanan VPC yang terkait dengan basis data.

Agar konsol menampilkan suatu fungsi Lambda, kondisi-kondisi berikut harus terpenuhi:

- Nama grup keamanan yang terkait dengan sumber daya komputasi cocok dengan pola `lambda-rds-`*n* atau `lambda-rdsproxy-`*n* (dengan *n* berupa angka).
- Grup keamanan yang terkait dengan sumber daya komputasi memiliki aturan keluar dengan rentang porta yang diatur ke porta instans basis data atau proksi terkait. Tujuan untuk aturan keluar harus diatur ke grup keamanan yang terkait dengan instans basis data atau proksi terkait.
- Jika konfigurasi menyertakan proksi, nama grup keamanan yang dilampirkan pada proksi yang terkait dengan basis data Anda cocok dengan pola `rdsproxy-lambda-`*n* (dengan *n* berupa angka).

- Grup keamanan yang terkait dengan fungsi memiliki aturan keluar dengan porta yang diatur ke porta yang digunakan oleh instans basis data atau proksi terkait. Tujuan harus diatur ke grup keamanan yang terkait dengan instans basis data atau proksi terkait.

Untuk melihat sumber daya komputasi yang dihubungkan secara otomatis dengan instans basis data

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data, lalu pilih instans basis data.
3. Pada tab Konektivitas dan keamanan, lihat sumber daya komputasi di bawah Sumber daya komputasi terhubung.

Memodifikasi instans DB Amazon RDS

Anda dapat mengubah pengaturan instans DB untuk menyelesaikan tugas seperti menambahkan penyimpanan tambahan atau mengubah kelas instans DB. Dalam topik ini, Anda dapat mengetahui cara memodifikasi instans DB Amazon RDS dan mempelajari pengaturan untuk instans DB.

Sebaiknya uji setiap perubahan pada instans pengujian sebelum memodifikasi instans produksi. Hal ini membantu Anda untuk sepenuhnya memahami dampak dari setiap perubahan. Pengujian sangat penting, khususnya ketika meningkatkan versi basis data.

Sebagian besar modifikasi pada instans DB dapat langsung diterapkan atau ditangguhkan hingga waktu pemeliharaan berikutnya. Beberapa modifikasi, seperti perubahan grup parameter, memerlukan boot ulang instans DB secara manual agar dapat diterapkan.

> **Important**
>
> Beberapa modifikasi menyebabkan waktu henti karena Amazon RDS harus mem-boot ulang instans DB Anda agar perubahan dapat diterapkan. Tinjau dampaknya terhadap basis data dan aplikasi Anda sebelum memodifikasi pengaturan instans DB Anda.

Konsol

Untuk memodifikasi instans DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data, kemudian pilih instans DB yang ingin diubah.
3. Pilih Ubah. Halaman Modifikasi instans DB akan muncul.
4. Ubah pengaturan apa pun yang Anda inginkan. Lihat informasi tentang setiap setelan di Pengaturan untuk instans DB.
5. Ketika semua perubahan sudah sesuai dengan keinginan Anda, pilih Lanjutkan dan periksa ringkasan modifikasi.
6. (Opsional) Pilih Terapkan seketika untuk menerapkan perubahan dengan serta-merta. Memilih opsi ini dapat menyebabkan waktu henti dalam beberapa kasus. Untuk informasi selengkapnya, lihat Menggunakan pengaturan Terapkan Segera.

7. Di halaman konfirmasi, tinjau perubahan Anda. Jika sudah benar, pilih Modifikasi instans DB untuk menyimpan perubahan Anda.

   Atau pilih Kembali untuk mengedit perubahan atau Batalkan untuk membatalkan perubahan Anda.

AWS CLI

Untuk memodifikasi instance DB dengan menggunakan AWS CLI, panggil modify-db-instanceperintah. Tentukan pengidentifikasi instans DB dan nilai untuk opsi yang ingin Anda modifikasi. Untuk informasi tentang setiap opsi, lihat Pengaturan untuk instans DB.

Example

Kode berikut mengubah `mydbinstance` dengan mengatur periode retensi cadangan ke 1 minggu (7 hari). Kode ini mengaktifkan perlindungan penghapusan menggunakan `--deletion-protection`. Untuk menonaktifkan perlindungan penghapusan, gunakan `--no-deletion-protection`. Perubahan diterapkan selama jendela pemeliharaan berikutnya menggunakan `--no-apply-immediately`. Gunakan `--apply-immediately` untuk segera menerapkan perubahan. Untuk informasi selengkapnya, lihat Menggunakan pengaturan Terapkan Segera.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --backup-retention-period 7 \
    --deletion-protection \
    --no-apply-immediately
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --backup-retention-period 7 ^
    --deletion-protection ^
    --no-apply-immediately
```

RDS API

Untuk memodifikasi instans DB menggunakan Amazon RDS API, panggil operasi ModifyDBInstance. Tentukan pengidentifikasi instans DB dan parameter untuk pengaturan yang ingin Anda modifikasi. Untuk informasi tentang setiap parameter, lihat Pengaturan untuk instans DB.

Menggunakan pengaturan Terapkan Segera

Saat Anda memodifikasi instans DB, Anda dapat langsung menerapkan perubahan. Untuk langsung menerapkan perubahan, pilih opsi Langsung terapkan pada AWS Management Console. Atau Anda menggunakan `--apply-immediately` parameter saat memanggil AWS CLI atau mengatur `ApplyImmediately` parameter `true` saat menggunakan Amazon RDS API.

Jika Anda tidak memilih untuk menerapkan perubahan dengan serta-merta, perubahan akan dimasukkan ke dalam antrean perubahan yang tertunda. Selama jendela pemeliharaan berikutnya, perubahan yang tertunda di antrean akan diterapkan. Jika Anda memilih untuk menerapkan perubahan dengan serta-merta, perubahan baru dan segala perubahan di antrean pengubahan yang tertunda akan diterapkan.

Untuk melihat modifikasi yang tertunda untuk jendela pemeliharaan berikutnya, gunakan describe-db-instances AWS CLI perintah dan periksa `PendingModifiedValues` bidangnya.

> **Important**
>
> Jika salah satu modifikasi yang tertunda mengharuskan penghentian ketersediaan instans DB untuk sementara waktu (waktu henti), memilih opsi terapkan langsung dapat menyebabkan waktu henti yang tidak terduga.
> Ketika Anda memilih untuk langsung menerapkan perubahan, setiap modifikasi yang tertunda juga akan langsung diterapkan, bukan diterapkan selama jendela pemeliharaan berikutnya. Jika Anda tidak ingin perubahan tertunda diterapkan pada jendela pemeliharaan berikutnya, Anda dapat memodifikasi instans DB untuk membatalkan perubahan. Anda dapat melakukan ini dengan menggunakan AWS CLI dan menentukan `--apply-immediately` opsi.

Perubahan pada beberapa pengaturan basis data langsung diterapkan, meski Anda memilih untuk menunda perubahan. Untuk melihat reaksi berbagai pengaturan basis data dengan pengaturan terapkan langsung, lihat Pengaturan untuk instans DB.

Pengaturan untuk instans DB

Pada tabel berikut, Anda dapat menemukan detail tentang pengaturan mana yang dapat dan tidak dapat Anda modifikasi. Anda juga dapat menemukan kapan perubahan dapat diterapkan dan apakah perubahan menyebabkan waktu henti untuk instans DB Anda. Dengan fitur Amazon RDS seperti Multi-AZ, Anda dapat meminimalkan waktu henti saat memodifikasi instans DB. Untuk informasi selengkapnya, lihat Mengonfigurasi dan mengelola deployment Multi-AZ.

Anda dapat memodifikasi instans DB menggunakan konsol, perintah CLI `modify-db-instance`, atau operasi RDS API `ModifyDBInstance`.

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung

Penyimpanan yang dialokasikan
 	
Penyimpanan, dalam gibibyte, yang ingin Anda alokasikan untuk instans DB Anda. Anda hanya dapat meningkatkan penyimpanan yang dialokasikan. Anda tidak dapat mengurangi penyimpanan yang dialokasikan.
 	
Anda tidak dapat memodifikasi penyimpanan beberapa instans DB lama, atau instans DB yang dipulihkan dari snapshot DB lama. Pengaturan Penyimpanan yang dialokasikan dinonaktifkan di konsol jika instans DB Anda tidak memenuhi syarat. Anda dapat memeriksa apakah Anda dapat mengalokasikan lebih banyak penyimpanan dengan menggunakan
	Opsi CLI:
	
	`--allocated-storage`
	
	Parameter API RDS:
	
	`AllocatedStorage`
		Jika Anda memilih untuk langsung menerapkan perubahan, perubahan akan langsung diterapkan.
		
		Jika Anda memilih untuk tidak langsung menerapkan perubahan, perubahan akan diterapkan pada jendela pemeliharaan berikutnya.
			Tidak akan ada waktu henti selama perubahan ini. Selama perubahan mungkin akan terjadi penurunan performa.	Semua mesin DB

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung
perintah CLI describe-valid-db-instance -modifikasi. Perintah ini mengembalikan opsi penyimpanan valid untuk instans DB Anda. Anda tidak dapat memodifikasi penyimpanan yang dialokasikan jika status instans DB adalah storage-optimization. Anda juga tidak dapat memodifikasi penyimpanan yang dialokasikan untuk instans DB jika penyimpanan telah dimodifikasi dalam enam jam terakhir. Penyimpanan maksimum yang diizinkan bergantung pada mesin DB dan jenis penyimpanan. Untuk informasi selengkapnya, lihat Penyimpanan instans DB Amazon RDS.				

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung
Konfigurasi arsitektur Konfigurasi yang memungkinkan beberapa basis data penghuni berada di instans DB Anda. Saat ini, hanya basis data kontainer (CDB) RDS for Oracle yang mendukung pengaturan ini. Jika CDB Anda dalam konfigurasi satu penghuni, Anda dapat memodifikasinya untuk menggunakan Konfigurasi multi-penghuni. Dalam konfigurasi ini, Anda dapat menggunakan RDS API untuk membuat 1-30 basis data penghuni, tergantung edisi basis data dan lisensi opsi yang diperlukan. PDB aplikasi dan PDB proksi tidak didukung. Konfigurasi multi-penghuni bersifat permanen, artinya CDB tidak dapat dikembalikan ke konfigurasi satu penghuni. **Note** Fitur Amazon RDS disebut "multi-penghuni" dan bukan "multi-penghuni" karena fitur ini merupakan kemampuan	Opsi CLI: `--multi-tenant` (konfigurasi multi-penghuni pada arsitektur CDB) `--no-multi-tenant` (konfigurasi satu penghuni pada arsitektur CDB) Parameter API: `MultiTenant`	Perubahan langsung diterapkan.	Tidak ada waktu henti selama perubahan ini.	Oracle

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung
dari platform RDS, bukan hanya mesin Oracle DB. Istilah "Oracle multi-penghuni" secara eksklusif merujuk pada arsitektur basis data Oracle, yang kompatibel dengan deployment on-premise dan RDS. Untuk informasi selengkapnya, lihat Ikhtisar CDB RDS for Oracle.				

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung
Pengaturan arsitektur Arsitektur basis data Oracle: CDB atau non-CDB. Jika Anda memilih Arsitektur multi-penghuni Oracle, RDS for Oracle akan mengubah non-CDB Anda menjadi CDB yang menggunakan konfigurasi satu penghuni. Pengaturan ini hanya didukung jika basis data Anda adalah non-CDB yang menjalankan Oracle Database 19c dengan RU April 2021 atau yang lebih tinggi. Setelah konversi, CDB Anda berisi satu basis data pluggable (PDB) awal. Perubahan arsitektur ini bersifat permanen, yang berarti CDB tidak dapat diubah kembali menjadi non-CDB. **Note** Untuk mengonversi CDB dalam konfigurasi penghuni tunggal ke konfigurasi multi-penghuni, modifikasi kembali instans CDB Anda dan pilih Konfigurasi multi-pen	Opsi CLI: `--engine oracle-ee-cdb` (Multi-penghuni Oracle) `--engine oracle-se2-cdb` (Multi-penghuni Oracle) Parameter API: `Engine`	Jika Anda memilih untuk langsung menerapkan perubahan, perubahan akan langsung diterapkan. Jika Anda memilih untuk tidak langsung menerapkan perubahan, perubahan akan diterapkan pada jendela pemeliharaan berikutnya.	Akan ada waktu henti selama perubahan ini.	Oracle

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung
ghuni untuk Konfigurasi arsitektur. Untuk informasi selengkapnya, lihat Konfigurasi satu penghuni pada arsitektur CDB.				

Peningkatan versi minor otomatis
 	
Pilih Aktifkan peningkatan versi minor otomatis untuk mengaktifkan instans DB Anda agar menerima peningkatan versi mesin DB minor pilihan secara otomatis saat tersedia. Ini adalah perilaku default. Amazon RDS melakukan peningkatan versi minor otomatis selama jendela pemeliharaan. Jika Anda tidak memilih Aktifkan peningkatan versi minor otomatis, instans DB Anda tidak ditingkatkan secara otomatis saat versi minor baru tersedia.
 	
Untuk informasi selengkapnya, lihat Meng-upgrade versi mesin minor secara otomatis.
	Opsi CLI:
	
	`--auto-minor-version-upgrade | --no-auto-minor-version-upgrade`
	
	Parameter API RDS:
	
	`AutoMinorVersionUpgrade`
		Perubahan langsung diterapkan. Pengaturan ini mengabaikan pengaturan terapkan langsung.	Tidak ada waktu henti selama perubahan ini.	Semua mesin DB

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung
Replikasi cadangan Pilih Aktifkan replikasi ke AWS Wilayah lain untuk membuat cadangan di Wilayah tambahan untuk pemulihan bencana. Lalu pilih Wilayah Tujuan untuk cadangan tambahan.	Tidak tersedia saat memodifikasi instans DB. Untuk informasi tentang mengaktifkan pencadangan Lintas wilayah menggunakan AWS CLI atau RDS API, lihat. Mengaktifkan pencadangan otomatis lintas Wilayah	Perubahan diterapkan secara asinkron, sesegera mungkin.	Tidak ada waktu henti selama perubahan ini.	Oracle, PostgreSQL, SQL Server

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung

Periode retensi cadangan
 	
Jumlah hari penyimpanan cadangan otomatis. Untuk menonaktifkan pencadangan otomatis, atur periode retensi pencadangan ke 0.
 	
Untuk informasi selengkapnya, lihat Pengantar cadangan.
 	
Note
Jika Anda menggunakan AWS Backup untuk mengelola cadangan Anda, opsi ini tidak berlaku. Untuk selengkapnya AWS Backup, lihat Panduan Pengembang AWS Cadangan.
	Opsi CLI:
	
	`--backup-retention-period`
	
	Parameter API RDS:
	
	`BackupRetentionPeriod`
		Jika Anda memilih untuk langsung menerapkan perubahan, perubahan akan langsung diterapkan.
		
		Jika Anda memilih untuk tidak langsung menerapkan perubahan, dan Anda mengubah pengaturan dari nilai bukan nol ke nilai bukan nol lainnya, perubahan akan diterapkan secara asinkron, sesegera mungkin. Jika tidak, perubahan akan terjadi selama jendela
			Waktu henti terjadi jika Anda mengubah dari 0 ke nilai bukan nol, atau dari nilai bukan nol ke 0.
			
			Hal ini berlaku baik untuk instans DB AZ tunggal dan Multi-AZ.
				Semua mesin DB

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung
		pemeliharaan berikutnya.		

Jendela pencadangan
 	
Rentang waktu saat pencadangan otomatis basis data Anda terjadi. Jendela pencadangan adalah waktu mulai dalam Waktu Terkoordinasi Universal (UTC), dan durasi dalam jam.
 	
Untuk informasi selengkapnya, lihat Pengantar cadangan.
 	
Note
Jika Anda menggunakan AWS Backup untuk mengelola cadangan Anda, opsi ini tidak muncul. Untuk selengkapnya AWS Backup, lihat Panduan AWS Backup Pengembang.
	Opsi CLI:
	
	`--preferred-backup-window`
	
	Parameter API RDS:
	
	`PreferredBackupWindow`
		Perubahan diterapkan secara asinkron, sesegera mungkin.	Tidak ada waktu henti selama perubahan ini.	Semua mesin DB

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung

Otoritas Sertifikat
 	
Otoritas sertifikat (CA) untuk sertifikat server yang digunakan oleh instans DB.
 	
Untuk informasi selengkapnya, lihat .
	Opsi CLI:
	
	`--ca-certificate-identifier`
	
	Parameter API RDS:
	
	`CACertificateIdentifier`
		Jika Anda memilih untuk langsung menerapkan perubahan, perubahan akan langsung diterapkan.
		
		Jika Anda memilih untuk tidak langsung menerapkan perubahan, perubahan akan diterapkan pada jendela pemeliharaan berikutnya.
			Waktu henti hanya terjadi jika mesin DB tidak mendukung rotasi tanpa mulai ulang. Anda dapat menggunakan describe-db-engine-versions AWS CLI perintah untuk menentukan apakah mesin DB mendukung rotasi tanpa restart.	Semua mesin DB

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung

Salin tag ke snapshot
 	
Jika Anda memiliki tag instans DB, aktifkan opsi ini untuk menyalinnya saat Anda membuat snapshot DB.
 	
Untuk informasi selengkapnya, lihat Memberi tag pada sumber daya Amazon RDS.
	Opsi CLI:
	
	`--copy-tags-to-snapshot` atau `--no-copy-tags-to-snapshot`
	
	Parameter API RDS:
	
	`CopyTagsToSnapshot`
		Perubahan langsung diterapkan. Pengaturan ini mengabaikan pengaturan terapkan langsung.	Tidak ada waktu henti selama perubahan ini.	Semua mesin DB

Port basis data
 	
Port yang ingin Anda gunakan untuk mengakses instans DB.
 	
Nilai port tidak boleh sama dengan nilai port mana pun yang ditentukan untuk opsi dalam grup opsi yang terkait dengan instans DB.
 	
Untuk informasi selengkapnya, lihat Menghubungkan ke instans DB Amazon RDS.
	Opsi CLI:
	
	`--db-port-number`
	
	Parameter API RDS:
	
	`DBPortNumber`
		Perubahan langsung diterapkan. Pengaturan ini mengabaikan pengaturan langsung terapkan.	Instans DB langsung di-boot ulang.	Semua mesin DB

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung

Versi mesin DB
 	
Versi mesin DB yang ingin Anda gunakan. Sebelum meningkatkan instans DB produksi, sebaiknya uji proses peningkatan pada instans DB pengujian. Hal ini membantu memverifikasi durasi dan memvalidasi aplikasi Anda.
 	
Untuk informasi selengkapnya, lihat Meng-upgrade versi mesin instans DB.
	Opsi CLI:
	
	`--engine-version`
	
	Parameter API RDS:
	
	`EngineVersion`
		Jika Anda memilih untuk langsung menerapkan perubahan, perubahan akan langsung diterapkan.
		
		Jika Anda memilih untuk tidak langsung menerapkan perubahan, perubahan akan diterapkan pada jendela pemeliharaan berikutnya.
			Akan ada waktu henti selama perubahan ini.	Semua mesin DB

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung

Kelas instans DB
 	
Kelas instans DB yang ingin Anda gunakan.
 	
Untuk informasi selengkapnya, lihat Kelas instans DB .
	Opsi CLI:
	
	`--db-instance-class`
	
	Parameter API RDS:
	
	`DBInstanceClass`
		Jika Anda memilih untuk langsung menerapkan perubahan, perubahan akan langsung diterapkan.
		
		Jika Anda memilih untuk tidak langsung menerapkan perubahan, perubahan akan diterapkan pada jendela pemeliharaan berikutnya.
			Akan ada waktu henti selama perubahan ini.	Semua mesin DB

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung

Pengidentifikasi instans DB
 	
Pengidentifikasi instans DB baru. Nilai ini disimpan sebagai string huruf kecil.
 	
Untuk informasi selengkapnya tentang dampak penggantian nama instans DB, lihat Mengganti nama instans DB.
	Opsi CLI:
	
	`--new-db-instance-identifier`
	
	Parameter API RDS:
	
	`NewDBInstanceIdentifier`
		Jika Anda memilih untuk langsung menerapkan perubahan, perubahan akan langsung diterapkan.
		
		Jika Anda memilih untuk tidak langsung menerapkan perubahan, perubahan akan diterapkan pada jendela pemeliharaan berikutnya.
			Akan ada waktu henti selama perubahan ini, kecuali versi mesin DB Anda mendukung pengunggahan SSL dinamis. Untuk menentukan apakah versi Anda memerlukan restart, jalankan AWS CLI perintah berikut:
			
			`aws rds describe-db-engine-versions \ --default-only \ --engine your-db-engine \ --query 'DBEngineVersions[*].Suppor`
				Semua mesin DB

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung
			`tsCertificateRotationWithoutRestart'`	

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung

Grup parameter DB
 	
Grup parameter DB yang ingin Anda kaitkan dengan instans DB.
 	
Untuk informasi selengkap nya, lihat Bekerja dengan grup parameter.
	Opsi CLI:
	
	`--db-parameter-group-name`
	
	Parameter API RDS:
	
	`DBParameterGroupName`
		Asosiasi grup parameter DB baru dengan instans DB terjadi segera.	Waktu henti tidak terjadi saat Anda mengaitkan grup parameter DB baru dengan instans DB Anda.
			
			Asosiasi grup parameter DB berbeda dari penerapan perubahan parameter dalam grup parameter . RDS menerapka n pengatura n parameter statis dan dinamis yang dimodifikasi dalam grup yang baru terkait hanya setelah Anda me-reboot instans DB secara manual. Namun,
				Semua mesin DB

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung
			jika Anda memodifikasi parameter dinamis dalam grup parameter DB setelah Anda mengaitkannya dengan instans DB, pengaturan parameter ini diterapkan segera tanpa memerlukan reboot. Untuk informasi lebih lanjut, lihat Bekerja dengan grup parameter dan Mem-boot ulang instans DB.	

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung

Volume Log Khusus
 	
Gunakan volume log khusus (DLV) untuk menyimpan log transaksi basis data pada volume penyimpanan yang terpisah dari volume yang berisi tabel basis data.
 	
Untuk informasi selengkapnya, lihat Menggunakan volume log khusus (DLV).
	Opsi CLI:
	
	`--dedicated-log-volume`
	
	Parameter API RDS:
	
	`DedicatedLogVolume`
		Perubahan diterapkan ketika instans DB di-boot ulang.	Akan terjadi waktu henti saat instans DB di-boot ulang.	MariaDB, MySQL, PostgreSQL

Perlindungan penghapusan
 	
Aktifkan perlindungan penghapusan agar instans DB Anda tidak terhapus.
 	
Untuk informasi selengkapnya, lihat Menghapus instans DB.
	Opsi CLI:
	
	`--deletion-protection|--no-deletion-protection`
	
	Parameter API RDS:
	
	`DeletionProtection`
		Perubahan langsung diterapkan. Pengaturan ini mengabaikan pengaturan terapkan langsung.	Tidak ada waktu henti selama perubahan ini.	Semua mesin DB

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung

Pemantauan yang Ditingkatkan
 	
Aktifkan Pemantauan yang Ditingkatkan untuk mengumpulkan metrik waktu nyata sistem operasi yang menjalankan instans DB Anda.
 	
Untuk informasi selengkapnya, lihat Memantau metrik OS dengan Pemantauan yang Disempurnakan.
	Opsi CLI:
	
	`--monitoring-interval` dan `--monitoring-role-arn`
	
	Parameter API RDS:
	
	`MonitoringInterval` dan `MonitoringRoleArn`
		Perubahan langsung diterapkan. Pengaturan ini mengabaikan pengaturan terapkan langsung.	Tidak ada waktu henti selama perubahan ini.	Semua mesin DB

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung

Autentikasi IAM DB
 	
Aktifkan autentikasi IAM DB untuk mengautentikasi pengguna basis data melalui pengguna dan peran.
 	
Untuk informasi selengkapnya, lihat Autentikasi basis data IAM untuk MariaDB, MySQL, dan PostgreSQL.
	Opsi CLI:
	
	`--enable-iam-database-authentication |--no-enable-iam-database-authentication`
	
	Parameter API RDS:
	
	`EnableIAMDatabaseAuthentication`
		Jika Anda memilih untuk langsung menerapkan perubahan, perubahan akan langsung diterapkan.
		
		Jika Anda memilih untuk tidak langsung menerapkan perubahan, perubahan akan diterapkan pada jendela pemeliharaan berikutnya.
			Tidak ada waktu henti selama perubahan ini.	Hanya MariaDB, MySQL, dan PostgreSQL

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung

Autentikasi Kerberos
 	
Pilih Active Directory sebagai tempat pemindahan instans DB. Direktori harus ada sebelum operasi ini. Jika direktori sudah dipilih, Anda dapat memilih Tidak ada untuk menghapus instans DB dari direktorinya saat ini.
 	
Untuk informasi selengkapnya, lihat Autentikasi Kerberos.
	Opsi CLI:
	
	`--domain` dan `--domain-iam-role-name`
	
	Parameter API RDS:
	
	`Domain` dan `DomainIAMRoleName`
		Jika Anda memilih untuk langsung menerapkan perubahan, perubahan akan langsung diterapkan.
		
		Jika Anda memilih untuk tidak langsung menerapkan perubahan, perubahan akan diterapkan pada jendela pemeliharaan berikutnya.
			Akan terjadi waktu henti singkat selama perubahan ini.	Hanya Microsoft SQL Server, MySQL, Oracle, dan PostgreSQL

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung

Model lisensi
 	
Pilih bring-your-own-licenseuntuk menggunakan lisensi Anda untuk Db2 dan Oracle.
 	
Pilih license-included untuk menggunakan perjanjian lisensi umum untuk Microsoft SQL Server atau Oracle.
 	
Untuk informasi selengkapnya, lihat Opsi-opsi pelisensian RDS for Db2, Melisensikan Microsoft SQL Server di Amazon RDS, dan Opsi lisensi RDS for Oracle.
	Opsi CLI:
	
	`--license-model`
	
	Parameter API RDS:
	
	`LicenseModel`
		Jika Anda memilih untuk langsung menerapkan perubahan, perubahan akan langsung diterapkan.
		
		Jika Anda memilih untuk tidak langsung menerapkan perubahan, perubahan akan diterapkan pada jendela pemeliharaan berikutnya.
			Akan ada waktu henti selama perubahan ini.	Hanya Microsoft SQL Server dan Oracle

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung

Ekspor log
 	
Jenis file log database untuk dipublikasikan ke Amazon CloudWatch Logs.
 	
Untuk informasi selengkapnya, lihat Menerbitkan log basis data ke Log Amazon CloudWatch.
	Opsi CLI:
	
	`--cloudwatch-logs-export-configuration`
	
	Parameter API RDS:
	
	`CloudwatchLogsExportConfiguration`
		Perubahan langsung diterapkan. Pengaturan ini mengabaikan pengaturan terapkan langsung.	Tidak ada waktu henti selama perubahan ini.	Semua mesin DB

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung

Jendela pemeliharaan
 	
Rentang waktu saat pemeliharaan sistem berlangsung. Pemeliharaan sistem mencakup peningkatan, jika ada. Jendela pemeliharaan adalah waktu mulai dalam Waktu Terkoordinasi Universal (UTC), dan durasi dalam jam.
 	
Jika jendela diatur ke waktu saat ini, harus ada jeda waktu minimal 30 menit antara waktu saat ini dan waktu akhir jendela. Waktu ini membantu memastikan bahwa setiap perubahan yang tertunda telah diterapkan.
 	
Untuk informasi selengkapnya, lihat Periode pemeliharaan Amazon RDS.
	Opsi CLI:
	
	`--preferred-maintenance-window`
	
	Parameter API RDS:
	
	`PreferredMaintenanceWindow`
		Perubahan langsung diterapkan. Pengaturan ini mengabaikan pengaturan langsung terapkan.	Jika ada satu atau beberapa tindakan tertunda yang menyebabkan waktu henti, dan jendela pemeliharaan diubah untuk menyertakan waktu saat ini, tindakan tertunda tersebut akan langsung diterapkan dan terjadi waktu henti.	Semua mesin DB

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung

Kelola kredensi master di AWS Secrets Manager
 	
Pilih Kelola kredensial master di AWS Secrets Manager untuk mengelola kata sandi pengguna master dalam rahasia di Secrets Manager.
 	
Anda juga dapat memilih kunci KMS yang akan digunakan untuk melindungi rahasia. Pilih dari kunci KMS di akun Anda, atau masukkan kunci dari akun lain.
 	
Jika RDS sudah mengelola kata sandi pengguna utama untuk instans DB, Anda dapat merotasi kata sandi pengguna utama dengan memilih Langsung rotasikan rahasia.
 	
Untuk informasi selengkapnya, lihat Manajemen kata sandi dengan Amazon RDS Aurora dan AWS Secrets Manager.
	Opsi CLI:
	
	`--manage-master-user-password | --no-manage-master-user-password`
	
	`--master-user-secret-kms-key-id`
	
	`--rotate-master-user-password | --no-rotate-master-user-password`
	
	Parameter API RDS:
	
	`ManageMasterUserPassword`
		Jika Anda mengaktifkan atau menonaktifkan manajemen kata sandi pengguna utama otomatis, perubahan akan langsung diterapkan. Perubahan ini mengabaikan pengaturan terapkan langsung.
		
		Saat merotasi kata sandi pengguna utama, Anda harus menentukan bahwa perubahan tersebut akan langsung diterapkan.
			Tidak ada waktu henti selama perubahan ini.	Semua mesin DB

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung

	`MasterUserSecretKmsKeyId`
	
	`RotateMasterUserPassword`			

Deployment multi-AZ
 	
Ya untuk melakukan deployment instans DB di beberapa Zona Ketersediaan. Pilih Tidak jika sebaliknya.
 	
Untuk informasi selengkapnya, lihat Mengonfigurasi dan mengelola deployment Multi-AZ.
	Opsi CLI:
	
	`--multi-az|--no-multi-az`
	
	Parameter API RDS:
	
	`MultiAZ`
		Jika Anda memilih untuk langsung menerapkan perubahan, perubahan akan langsung diterapkan.
		
		Jika Anda memilih untuk tidak langsung menerapkan perubahan, perubahan akan diterapkan pada jendela pemeliharaan berikutnya.
			Tidak ada waktu henti selama perubahan ini. Namun, ada kemungkinan dampak pada performa. Untuk informasi selengkapnya, lihat Mengubah instans DB menjadi deployment instans DB Multi-AZ.	Semua mesin DB

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung

Jenis jaringan
 	
Protokol pengalamatan IP didukung oleh instans DB.
 	
IPv4 untuk menentukan bahwa sumber daya dapat berkomunikasi dengan instans DB hanya melalui protokol pengalamatan Internet Protocol versi 4 (IPv4).
 	
Mode tumpukan ganda untuk menentukan bahwa sumber daya dapat berkomunikasi dengan instans DB melalui IPv4, Internet Protocol versi 6 (IPv6), atau keduanya. Gunakan mode tumpukan ganda jika Anda memiliki sumber daya yang harus berkomunikasi dengan instans DB melalui protokol pengalamatan IPv6. Selain itu, pastikan Anda mengaitkan blok CIDR IPv6 dengan semua subnet dalam grup subnet DB yang Anda tentukan.
 	
Untuk informasi selengkapnya, lihat Penentuan alamat IP Amazon RDS.
	Opsi CLI:
	
	`--network-type`
	
	Parameter API RDS:
	
	`NetworkType`
		Jika Anda memilih untuk langsung menerapkan perubahan, perubahan akan langsung diterapkan.
		
		Jika Anda memilih untuk tidak langsung menerapkan perubahan, perubahan akan diterapkan pada jendela pemeliharaan berikutnya.
			Kemungkinan akan ada waktu henti selama perubahan ini.	Semua mesin DB

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung

Kata sandi utama baru
 	
Kata sandi untuk pengguna utama Anda. Kata sandi harus berisi 8–41 karakter alfanumerik.
	Opsi CLI:
	
	`--master-user-password`
	
	Parameter API RDS:
	
	`MasterUserPassword`
		Perubahan diterapkan secara asinkron, sesegera mungkin. Pengaturan ini mengabaikan pengaturan langsung terapkan.	Tidak ada waktu henti selama perubahan ini.	Semua mesin DB

Grup opsi
 	
Grup opsi yang Anda inginkan terkait dengan instans DB.
 	
Untuk informasi selengkapnya, lihat Menggunakan grup opsi.
	Opsi CLI:
	
	`--option-group-name`
	
	Parameter API RDS:
	
	`OptionGroupName`
		Jika Anda memilih untuk langsung menerapkan perubahan, perubahan akan langsung diterapkan.
		
		Jika Anda memilih untuk tidak langsung menerapkan perubahan, perubahan akan diterapkan pada jendela pemeliharaan berikutnya.
			Tidak ada waktu henti selama perubahan ini. Kecuali Anda menambahkan Plugin Audit MariaDB ke instans DB RDS for MariaDB atau RDS for MySQL, yang dapat menyebabkan pemadaman.	Semua mesin DB

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung

Wawasan Performa
 	
Aktifkan Wawasan Performa untuk memantau beban instans DB sehingga Anda dapat menganalisis dan memecahkan masalah performa pada basis data Anda.
 	
Wawasan Performa tidak tersedia untuk beberapa versi mesin DB dan kelas instans DB. Bagian Wawasan Performa tidak muncul di konsol jika tidak tersedia untuk instans DB Anda.
 	
Untuk informasi lebih lanjut, lihat Memantau muatan DB dengan Wawasan Performa di Amazon RDS dan Dukungan kelas instans, Wilayah, dan mesin DB Amazon RDS untuk Wawasan Performa.
	Opsi CLI:
	
	`--enable-performance-insights| --no-enable-performance-insights`
	
	Parameter API RDS:
	
	`EnablePerformanceInsights`
		Perubahan langsung diterapkan. Pengaturan ini mengabaikan pengaturan terapkan langsung.	Tidak ada waktu henti selama perubahan ini.	Semua kecuali Db2

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung

Wawasan Performa AWS KMS key
 	
Pengenal AWS KMS kunci untuk enkripsi data AWS KMS key Performance Insights. Pengidentifikasi kunci adalah Nama Sumber Daya Amazon (ARN) AWS KMS , pengenal kunci, atau alias kunci untuk kunci KMS.
 	
Untuk informasi selengkapnya, lihat Mengaktifkan dan menonaktifkan Wawasan Performa.
	Opsi CLI:
	
	`--performance-insights-kms-key-id`
	
	Parameter API RDS:
	
	`PerformanceInsightsKMSKeyId`
		Perubahan langsung diterapkan. Pengaturan ini mengabaikan pengaturan terapkan langsung.	Tidak ada waktu henti selama perubahan ini.	Semua kecuali Db2

Periode Retensi Wawasan Performa
 	
Jumlah waktu, dalam hari, untuk mempertahankan data Wawasan Performa. Pengaturan retensi di tingkat gratis adalah Default (7 hari). Untuk mempertahankan data performa Anda lebih lama, tetapkan 1–24 bulan. Untuk informasi selengkapnya tentang periode retensi, lihat Harga dan retensi data untuk Wawasan Performa.
 	
Untuk informasi selengkapnya, lihat Mengaktifkan dan menonaktifkan Wawasan Performa.
	Opsi CLI:
	
	`--performance-insights-retention-period`
	
	Parameter API RDS:
	
	`PerformanceInsightsRetentionPeriod`
		Perubahan langsung diterapkan. Pengaturan ini mengabaikan pengaturan terapkan langsung.	Tidak ada waktu henti selama perubahan ini.	Semua kecuali Db2

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung

Fitur prosesor
 	
Jumlah inti CPU dan jumlah utas per inti untuk kelas instans DB pada instans DB tersebut.
 	
Untuk informasi selengkapnya, lihat Mengonfigurasi prosesor untuk kelas instans DB di RDS for Oracle.
	Opsi CLI:
	
	`--processor-features` dan `--use-default-processor-features` | `--no-use-default-processor-features`
	
	Parameter API RDS:
	
	`ProcessorFeatures` dan `UseDefaultProcessorFeatures`
		Jika Anda memilih untuk langsung menerapkan perubahan, perubahan akan langsung diterapkan.
		
		Jika Anda memilih untuk tidak langsung menerapkan perubahan, perubahan akan diterapkan pada jendela pemeliharaan berikutnya.
			Akan ada waktu henti selama perubahan ini.	Hanya Oracle

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung

IOPS yang Tersedia
 	
Nilai IOPS (operasi I/O per detik) yang Tersedia untuk instans DB. Pengaturan ini hanya tersedia jika Anda memilih salah satu Jenis penyimpanan berikut:
 	
• SSD tujuan umum (gp3)
• SSD IOPS yang tersedia (io1)
• IOPS SSD yang disediakan (io2)
 	
Untuk informasi lebih lanjut, lihat the section called “Penyimpanan IOPS yang Tersedia” dan the section called “Penyimpanan gp3”.
	Opsi CLI:
	
	`--iops`
	
	Parameter API RDS:
	
	`Iops`
		Jika Anda memilih untuk langsung menerapkan perubahan, perubahan akan langsung diterapkan.
		
		Jika Anda memilih untuk tidak langsung menerapkan perubahan, perubahan akan diterapkan pada jendela pemeliharaan berikutnya.
			Tidak ada waktu henti selama perubahan ini.	Semua mesin DB

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung

Akses publik
 	
Dapat diakses publik untuk memberikan alamat IP publik pada instans DB, yang berarti bahwa instans DB ini dapat diakses di luar VPC. Agar dapat diakses oleh publik, instans DB juga harus berada di subnet publik di VPC.
 	
Tidak dapat diakses publik agar instans DB hanya dapat diakses dari dalam VPC.
 	
Untuk informasi selengkapnya, lihat Menyembunyikan klaster DB dalam VPC dari internet.
 	
Untuk terhubung ke instans DB dari luar VPC, instans DB harus dapat diakses publik. Selain itu, akses harus diberikan menggunakan aturan masuk dari grup keamanan instans DB. Selain itu, persyaratan lain harus dipenuhi. Untuk informasi selengkapnya, lihat Tidak dapat terhubung ke instans DB Amazon RDS.
 	
Jika instans DB Anda tidak dapat diakses publik, Anda juga dapat
	Opsi CLI:
	
	`--publicly-accessible | --no-publicly-accessible`
	
	Parameter API RDS:
	
	`PubliclyAccessible`
		Perubahan langsung diterapkan. Pengaturan ini mengabaikan pengaturan terapkan langsung.	Tidak ada waktu henti selama perubahan ini.	Semua mesin DB

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung
menggunakan koneksi VPN AWS Site-to-Site AWS Direct Connect atau koneksi untuk mengakses nya dari jaringan pribadi. Untuk informasi selengkapnya, lihat Privasi lalu lintas jaringan internet.				

Grup keamanan
 	
Grup keamanan VPC yang Anda inginkan terkait dengan instans DB.
 	
Untuk informasi selengkapnya, lihat Mengontrol akses dengan grup keamanan.
	Opsi CLI:
	
	`--vpc-security-group-ids`
	
	Parameter API RDS:
	
	`VpcSecurityGroupIds`
		Perubahan diterapkan secara asinkron, sesegera mungkin. Pengaturan ini mengabaikan pengaturan langsung terapkan.	Tidak ada waktu henti selama perubahan ini.	Semua mesin DB

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung

Penskalaan otomatis penyimpanan
 	
Aktifkan penskalaan otomatis penyimpanan agar Amazon RDS dapat secara otomatis menambah penyimpanan saat diperlukan untuk menghindari kehabisan ruang penyimpanan pada instans DB Anda.
 	
Gunakan Ambang batas penyimpanan maksimum untuk mengatur batas maksimum penambahan peningkatan penyimpanan otomatis untuk instans DB Anda oleh Amazon RDS. Defaultnya adalah 1.000 GiB.
 	
Untuk informasi selengkapnya, lihat Mengelola kapasitas secara otomatis dengan penskalaan otomatis penyimpanan Amazon RDS.
	Opsi CLI:
	
	`--max-allocated-storage`
	
	Parameter API RDS:
	
	`MaxAllocatedStorage`
		Perubahan langsung diterapkan. Pengaturan ini mengabaikan pengaturan terapkan langsung.	Tidak ada waktu henti selama perubahan ini.	Semua mesin DB

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung

Throughput penyimpanan
 	
Nilai throughput penyimpanan baru untuk instans DB. Pengaturan ini hanya tersedia jika Anda memilih Jenis penyimpanan SSD tujuan umum (gp3).
 	
Untuk informasi selengkapnya, lihat the section called “Penyimpanan gp3”.
	Opsi CLI:
	
	`--storage-throughput`
	
	Parameter API RDS:
	
	`StorageThroughput`
		Jika Anda memilih untuk langsung menerapkan perubahan, perubahan akan langsung diterapkan.
		
		Jika Anda memilih untuk tidak langsung menerapkan perubahan, perubahan akan diterapkan pada jendela pemeliharaan berikutnya.
			Tidak ada waktu henti selama perubahan ini.	Semua mesin DB

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung

Jenis penyimpanan
 	
Jenis penyimpanan yang ingin Anda gunakan.
 	
Jika Anda memilih SSD Tujuan Umum (gp3), Anda dapat menyediakan tambahan IOPS yang Tersedia dan Throughput penyimpanan di bagian Pengaturan lanjutan.
 	
Jika Anda memilih Provisioned IOPS SSD (io1) atau Provisioned IOPS SSD (io2), masukkan nilai IOPS Provisioned.
 	
Setelah Amazon RDS mulai memodifikasi instans DB Anda untuk mengubah ukuran atau jenis penyimpanan, Anda tidak dapat mengirimkan permintaan lain untuk mengubah ukuran, performa, atau jenis penyimpanan selama enam jam.
 	
Untuk informasi selengkapnya, lihat Jenis penyimpanan Amazon RDS.
	Opsi CLI:
	
	`--storage-type`
	
	Parameter API RDS:
	
	`StorageType`
		Jika Anda memilih untuk langsung menerapkan perubahan, perubahan akan langsung diterapkan.
		
		Jika Anda memilih untuk tidak langsung menerapkan perubahan, perubahan akan diterapkan pada jendela pemeliharaan berikutnya.
			Semua perubahan berikut menyebabkan terjadinya waktu henti singkat saat proses dimulai. Setelah itu, Anda dapat menggunakan basis data secara normal saat perubahan terjadi.
			
			• Dari Tujuan Umum (SSD) atau IOPS yang Tersedia (SSD) menjadi Magnetik.
			• Dari Magnetik menjadi Tujuan Umum (SSD) atau IOPS yang
				Semua mesin DB

Pengaturan dan deskripsi konsol	Opsi CLI dan parameter RDS API	Kapan perubahan terjadi	Catatan waktu henti	Mesin DB yang didukung
			Tersedia (SSD).	

Grup subnet DB
 	
Grup subnet DB untuk instans DB. Anda dapat menggunakan pengaturan ini untuk memindahkan instans DB ke VPC yang berbeda.
 	
Untuk informasi selengkapnya, lihat Amazon VPC dan Amazon RDS.
	Opsi CLI:
	
	`--db-subnet-group-name`
	
	Parameter API RDS:
	
	`DBSubnetGroupName`
		Jika Anda memilih untuk langsung menerapkan perubahan, perubahan akan langsung diterapkan.
		
		Jika Anda memilih untuk tidak langsung menerapkan perubahan, perubahan akan diterapkan pada jendela pemeliharaan berikutnya.
			Akan ada waktu henti selama perubahan ini.	Semua mesin DB

Memelihara instans DB

Amazon RDS melakukan pemeliharaan secara berkala pada sumber daya Amazon RDS. Pemeliharaan sering kali melibatkan pembaruan ke sumber daya berikut di instans DB:

- Perangkat keras yang mendasarinya
- Sistem operasi yang mendasarinya (OS)
- Versi mesin basis data

Pembaruan pada sistem operasi paling sering terjadi untuk masalah keamanan. Anda harus melakukannya sesegera mungkin.

Beberapa item pemeliharaan mengharuskan Amazon RDS membuat instans DB Anda offline selama waktu yang singkat. Item pemeliharaan yang mengharuskan sumber daya untuk offline mencakup patching sistem operasi atau basis data yang diperlukan. Patching yang diperlukan secara otomatis dijadwalkan hanya untuk patch yang terkait dengan keamanan dan keandalan instans. Patching tersebut jarang terjadi, biasanya sekali setiap beberapa bulan. Ini jarang membutuhkan lebih dari periode pemeliharaan Anda.

Modifikasi instans DB tertunda yang Anda pilih untuk tidak segera diterapkan juga diterapkan selama periode pemeliharaan. Misalnya, Anda dapat memilih untuk mengubah kelas instans DB atau grup parameter selama periode pemeliharaan. Modifikasi seperti yang Anda tentukan menggunakan pengaturan boot ulang tertunda tidak muncul dalam daftar Pemeliharaan tertunda. Untuk informasi tentang cara mengubah instans DB, lihat Memodifikasi instans DB Amazon RDS.

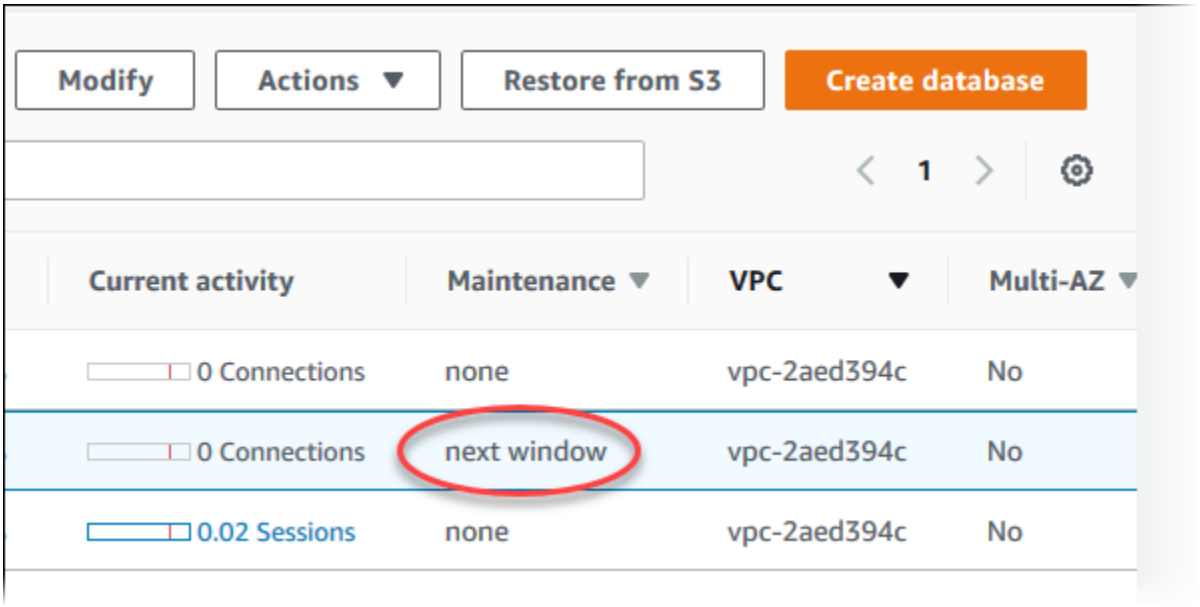
Untuk melihat modifikasi yang tertunda untuk jendela pemeliharaan berikutnya, gunakan describe-db-instances AWS CLI perintah dan periksa `PendingModifiedValues` bidangnya.

Topik

- Melihat pemeliharaan Tertunda
- Menerapkan pembaruan untuk instans DB
- Pemeliharaan untuk deployment multi-AZ
- Periode pemeliharaan Amazon RDS
- Menyesuaikan periode pemeliharaan instans DB yang dinginkan
- Bekerja dengan pembaruan sistem operasi

Melihat pemeliharaan Tertunda

Lihat apakah pembaruan pemeliharaan tersedia untuk instans DB Anda dengan menggunakan konsol RDS, API AWS CLI, atau RDS. Jika tersedia, pembaruan akan dicantumkan dalam kolom Pemeliharaan untuk instans DB di konsol Amazon RDS, seperti yang ditunjukkan berikut.



Jika pembaruan pemeliharaan tidak tersedia untuk instans DB, nilai kolomnya adalah tidak ada.

Jika pembaruan pemeliharaan tersedia untuk instans DB, kemungkinan nilai kolomnya adalah sebagai berikut:

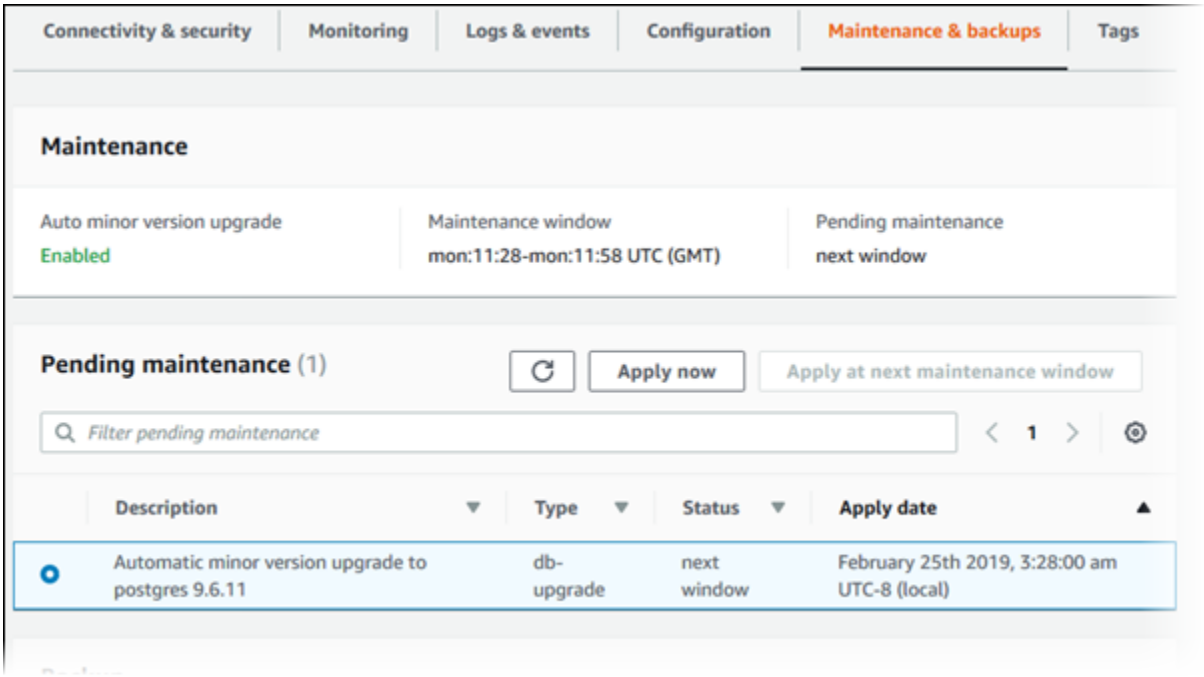
- diperlukan – Tindakan pemeliharaan akan diterapkan ke sumber daya dan tidak dapat ditunda tanpa batas waktu.
- tersedia – Tindakan pemeliharaan tersedia, tetapi tidak akan diterapkan ke sumber daya secara otomatis. Anda dapat menerapkannya secara manual.
- periode berikutnya – Tindakan pemeliharaan akan diterapkan ke sumber daya pada periode pemeliharaan berikutnya.
- Sedang berlangsung – Tindakan pemeliharaan sedang dalam proses penerapan ke sumber daya.

Jika pembaruan tersedia, Anda dapat melakukan salah satu tindakan berikut:

- Jika nilai pemeliharaannya periode berikutnya, tunda item pemeliharaan dengan memilih Tunda peningkatan dari Tindakan. Anda tidak dapat menunda tindakan pemeliharaan jika sudah dimulai.
- Segera terapkan item pemeliharaan.

- Jadwalkan item pemeliharaan untuk dimulai pada periode pemeliharaan berikutnya.
- Tidak melakukan tindakan apa pun.

Untuk melakukan tindakan, pilih instans DB untuk menampilkan detailnya, kemudian pilih Pemeliharaan & pencadangan. Item pemeliharaan yang tertunda muncul.



Periode pemeliharaan menentukan kapan operasi yang tertunda dimulai, tetapi tidak membatasi total waktu eksekusi operasi ini. Operasi pemeliharaan tidak dijamin selesai sebelum periode pemeliharaan berakhir, dan dapat berlanjut melebihi waktu akhir yang ditentukan. Untuk informasi selengkapnya, lihat Periode pemeliharaan Amazon RDS.

Anda juga dapat melihat apakah pembaruan pemeliharaan tersedia untuk instans DB Anda dengan menjalankan `describe-pending-maintenance-actions` AWS CLI perintah.

Menerapkan pembaruan untuk instans DB

Dengan Amazon RDS, Anda dapat memilih waktu untuk menerapkan operasi pemeliharaan. Anda dapat memutuskan kapan Amazon RDS menerapkan pembaruan dengan menggunakan konsol RDS, AWS Command Line Interface (AWS CLI), atau RDS API.

> **Note**
>
> Untuk RDS for SQL Server, pembaruan ke sistem operasi yang mendasarinya dapat diterapkan dengan menghentikan dan memulai instans DB Anda, atau dengan menaikkan skala kelas instans DB dan kemudian menurunkannya lagi.

Konsol

Untuk mengelola pembaruan untuk klaster DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data.
3. Pilih instans DB yang memiliki pembaruan yang diperlukan.
4. Untuk Tindakan, pilih salah satu opsi berikut:
   - Tingkatkan sekarang
   - Tingkatkan pada periode berikutnya

   > **Note**
   >
   > Jika memilih Tingkatkan pada periode berikutnya dan ingin menunda pembaruan di lain waktu, Anda dapat memilih Tunda peningkatan. Anda tidak dapat menunda tindakan pemeliharaan jika sudah dimulai.
   > Untuk membatalkan tindakan pemeliharaan, ubah instans DB dan nonaktifkan Peningkatan versi minor otomatis.

AWS CLI

Untuk menerapkan pembaruan yang tertunda ke instans DB, gunakan apply-pending-maintenance-action AWS CLI perintah.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds apply-pending-maintenance-action \
    --resource-identifier arn:aws:rds:us-west-2:001234567890:db:mysql-db \
    --apply-action system-update \
    --opt-in-type immediate
```

Untuk Windows:

```
aws rds apply-pending-maintenance-action ^
    --resource-identifier arn:aws:rds:us-west-2:001234567890:db:mysql-db ^
    --apply-action system-update ^
    --opt-in-type immediate
```

> **Note**
>
> Untuk menunda tindakan pemeliharaan, tentukan `undo-opt-in` untuk `--opt-in-type`. Anda tidak dapat menentukan `undo-opt-in` untuk `--opt-in-type` jika tindakan pemeliharaan sudah dimulai.
> Untuk membatalkan tindakan pemeliharaan, jalankan modify-db-instance AWS CLI perintah dan tentukan`--no-auto-minor-version-upgrade`.

Untuk mengembalikan daftar sumber daya yang memiliki setidaknya satu pembaruan yang tertunda, gunakan describe-pending-maintenance-actions AWS CLI perintah.

Example

Untuk Linux, macOS, atau Unix:

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

Untuk Windows:

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

Anda juga dapat mengembalikan daftar sumber daya untuk instans DB dengan menentukan `--filters` parameter `describe-pending-maintenance-actions` AWS CLI perintah. Format untuk perintah `--filters` adalah `Name=`*`filter-name`*`,Value=`*`resource-id`*`,...`.

Berikut adalah nilai yang diterima untuk parameter `Name` dari filter:

- `db-instance-id` – Menerima daftar pengidentifikasi instans DB atau Amazon Resource Name (ARN). Daftar yang ditampilkan hanya mencakup tindakan pemeliharaan yang tertunda untuk instans DB yang diidentifikasi oleh pengidentifikasi atau ARN tersebut.
- `db-cluster-id` – Menerima daftar pengidentifikasi klaster DB atau ARN untuk Amazon Aurora. Daftar yang ditampilkan hanya mencakup tindakan pemeliharaan yang tertunda untuk klaster DB yang diidentifikasi oleh pengidentifikasi atau ARN tersebut.

Misalnya, contoh berikut menampilkan tindakan pemeliharaan yang tertunda untuk instans DB `sample-instance1` dan `sample-instance2`.

Example

Untuk Linux, macOS, atau Unix:

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

Untuk Windows:

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

RDS API

Untuk menerapkan pembaruan ke instans DB, panggil operasi `ApplyPendingMaintenanceAction` Amazon RDS API.

Untuk menampilkan daftar sumber daya yang memiliki setidaknya satu pembaruan tertunda, panggil operasi `DescribePendingMaintenanceActions` Amazon RDS API.

Pemeliharaan untuk deployment multi-AZ

Menjalankan instans DB sebagai deployment multi-AZ dapat lebih mengurangi dampak dari peristiwa pemeliharaan. Hasil ini karena Amazon RDS menerapkan pembaruan sistem operasi dengan mengikuti langkah-langkah berikut:

1. Lakukan pemeliharaan pada waktu siaga.
2. Naikkan siaga ke primer.
3. Lakukan pemeliharaan pada primer yang lama, yang menjadi siaga baru.

Jika Anda meningkatkan meningkatkan mesin basis data untuk instans DB dalam deployment multi-AZ, Amazon RDS akan memodifikasi instans DB primer dan sekunder secara bersamaan. Dalam hal ini, instans DB primer dan sekunder dalam deployment multi-AZ tidak tersedia selama peningkatan. Operasi ini menyebabkan waktu henti hingga peningkatan selesai. Durasi waktu henti bervariasi berdasarkan ukuran instans DB Anda.

Jika ada tambalan sistem operasi yang mendasari yang perlu diterapkan, failover multi-AZ singkat diperlukan untuk menerapkan tambalan ke instans DB utama. Failover ini biasanya berlangsung kurang dari satu menit.

Jika instans DB Anda menjalankan RDS untuk MySQL, RDS untuk PostgreSQL, atau RDS untuk MariaDB, Anda dapat meminimalkan waktu henti yang diperlukan untuk peningkatan dengan menggunakan penerapan biru/hijau. Untuk informasi selengkapnya, lihat Menggunakan Deployment Blue/Green Amazon RDS untuk pembaruan basis data. Jika Anda meningkatkan instans DB RDS for SQL Server dalam deployment multi-AZ, Amazon RDS akan melakukan peningkatan bergulir, sehingga Anda mengalami pemadaman hanya selama durasi failover. Untuk informasi selengkapnya, lihat Pertimbangan Multi-AZ dan optimisasi dalam memori.

Jika instans DB Anda menjalankan RDS for SQL Server dalam deployment multi-AZ, Anda dapat menerapkan pembaruan ke sistem operasi yang mendasarinya dengan menggunakan salah satu metode berikut:

- Ubah kelas instans DB ke ukuran yang berbeda, lalu ubah lagi ke ukuran awalnya.
- Naikkan skala ukuran instans DB, turunkan lagi skalanya ke ukuran awalnya.
- Ubah instans DB dari Multi-AZ ke AZ Tunggal, hentikan dan mulai instans DB, lalu ubah kembali instans ke Multi-AZ.

Untuk informasi selengkapnya tentang deployment Multi-AZ, lihat Mengonfigurasi dan mengelola deployment Multi-AZ.

Periode pemeliharaan Amazon RDS

Setiap instans DB memiliki periode pemeliharaan mingguan di mana setiap perubahan sistem diterapkan. Anggap periode pemeliharaan sebagai peluang untuk mengontrol ketika modifikasi dan patching perangkat lunak terjadi. Jika peristiwa pemeliharaan dijadwalkan selama satu minggu, ini akan dimulai selama periode pemeliharaan 30 menit yang Anda identifikasi. Sebagian besar peristiwa pemeliharaan juga selesai selama periode pemeliharaan 30 menit, meskipun peristiwa pemeliharaan yang lebih besar bisa memakan waktu lebih dari 30 menit.

Periode pemeliharaan 30 menit dipilih secara acak dari blok waktu 8 jam per wilayah. Jika Anda tidak menentukan periode pemeliharaan saat membuat instans DB, RDS akan menetapkan periode pemeliharaan 30 menit pada hari yang dipilih secara acak dalam seminggu.

RDS menggunakan beberapa sumber daya di instans DB Anda saat pemeliharaan diterapkan. Anda mungkin mendapati efek minimal pada performa. Untuk instans DB, dalam situasi yang jarang terjadi, failover Multi-AZ mungkin diperlukan untuk menyelesaikan pembaruan pemeliharaan.

Setelah itu, Anda dapat menemukan blok waktu untuk setiap wilayah tempat asal periode pemeliharaan default ditetapkan.

Nama Wilayah	Wilayah	Blok Waktu
AS Timur (Ohio)	us-east-2	03.00–11.00 UTC
AS Timur (Virginia Utara)	us-east-1	03.00–11.00 UTC
AS Barat (California Utara)	us-west-1	06.00–14.00 UTC
AS Barat (Oregon)	us-west-2	06.00–14.00 UTC
Afrika (Cape Town)	af-south-1	03.00–11.00 UTC
Asia Pasifik (Hong Kong)	ap-east-1	06.00–14.00 UTC

Nama Wilayah	Wilayah	Blok Waktu
Asia Pasifik (Hyderabad)	ap-south-2	06.30–14.30 UTC
Asia Pasifik (Jakarta)	ap-southeast-3	08.00–16.00 UTC
Asia Pasifik (Melbourne)	ap-southeast-4	11.00–19.00 UTC
Asia Pasifik (Mumbai)	ap-south-1	06.00–14.00 UTC
Asia Pasifik (Osaka)	ap-northeast-3	22.00–23.59 UTC
Asia Pasifik (Seoul)	ap-northeast-2	13.00–21.00 UTC
Asia Pasifik (Singapura)	ap-southeast-1	14.00–22.00 UTC
Asia Pasifik (Sydney)	ap-southeast-2	12.00–20.00 UTC
Asia Pasifik (Tokyo)	ap-northeast-1	13.00–21.00 UTC
Kanada (Pusat)	ca-central-1	03.00–11.00 UTC
Kanada Barat (Calgary)	ca-west-1	18:00 — 02:00 UTC
Tiongkok (Beijing)	cn-north-1	06.00–14.00 UTC
Tiongkok (Ningxia)	cn-northwest-1	06.00–14.00 UTC
Eropa (Frankfurt)	eu-central-1	21.00–05.00 UTC
Eropa (Irlandia)	eu-west-1	22.00–06.00 UTC
Eropa (London)	eu-west-2	22.00–06.00 UTC
Eropa (Milan)	eu-south-1	02.00–10.00 UTC
Eropa (Paris)	eu-west-3	23.59–07.29 UTC

Nama Wilayah	Wilayah	Blok Waktu
Eropa (Spanyol)	eu-south-2	02.00–10.00 UTC
Eropa (Stockholm)	eu-north-1	23.00–07.00 UTC
Eropa (Zürich)	eu-central-2	02.00–10.00 UTC
Israel (Tel Aviv)	il-central-1	03.00–11.00 UTC
Timur Tengah (Bahrain)	me-south-1	06.00–14.00 UTC
Timur Tengah (UEA)	me-central-1	05.00–13.00 UTC
Amerika Selatan (Sao Paulo)	sa-east-1	00.00–08.00 UTC
AWS GovCloud (AS-Timur)	us-gov-east-1	17.00–01.00 UTC
AWS GovCloud (AS-Barat)	us-gov-west-1	06.00–14.00 UTC

Menyesuaikan periode pemeliharaan instans DB yang dinginkan

Periode pemeliharaan harus berada dalam waktu penggunaan terendah, sehingga kemungkinan memerlukan perubahan dari waktu ke waktu. Instans DB Anda tidak akan tersedia selama waktu ini hanya jika perubahan sistem, seperti perubahan kelas instans DB, diterapkan dan memerlukan pemadaman. Instans DB Anda tidak tersedia hanya untuk jumlah waktu minimum yang diperlukan untuk melakukan perubahan yang diperlukan.

Dalam contoh berikut, Anda menyesuaikan periode pemeliharaan yang dinginkan untuk instans DB.

Untuk contoh ini, kami mengasumsikan bahwa instans DB bernama mydbinstance ada dan memiliki pemeliharaan pilihan “Sun:05:00-Sun:06:00” UTC.

Konsol

Untuk menyesuaikan periode pemeliharaan yang diinginkan

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data, lalu pilih instans DB yang ingin diubah.
3. Pilih Ubah. Halaman Mengubah instans DB akan muncul.
4. Di bagian Pemeliharaan, perbarui periode pemeliharaan.

   > **Note**
   >
   > Periode pemeliharaan dan periode pencadangan untuk instans DB tidak boleh tumpang tindih. Jika Anda memasukkan nilai untuk periode waktu pemeliharaan yang tumpang tindih dengan waktu pencadangan, pesan kesalahan akan muncul.

5. Pilih Lanjutkan.

   Di halaman konfirmasi, tinjau perubahan Anda.
6. Untuk menerapkan perubahan ke periode pemeliharaan secara langsung, pilih Terapkan langsung.
7. Pilih Ubah instans DB untuk menyimpan perubahan Anda.

   Atau, pilih Kembali untuk mengedit perubahan, atau pilih Batal untuk membatalkan perubahan.

AWS CLI

Untuk menyesuaikan jendela pemeliharaan yang disukai, gunakan AWS CLI `modify-db-instance`perintah dengan parameter berikut:

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

Example

Contoh kode berikut mengatur periode pemeliharaan ke Selasa mulai pukul 04.00-04.30 UTC.

Untuk Linux, macOS, atau Unix:

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

Untuk Windows:

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

RDS API

Untuk menyesuaikan periode pemeliharaan yang diinginkan, gunakan operasi Amazon RDS API `ModifyDBInstance` dengan parameter berikut:

- `DBInstanceIdentifier`
- `PreferredMaintenanceWindow`

Bekerja dengan pembaruan sistem operasi

Instans DB RDS for Db2, RDS for MariaDB, RDS for MySQL, RDS for PostgreSQL, dan RDS for Oracle terkadang memerlukan pembaruan sistem operasi. Amazon RDS meningkatkan sistem operasi ke versi yang lebih baru untuk meningkatkan performa basis data dan postur keamanan pelanggan secara keseluruhan. Pembaruan biasanya memerlukan waktu sekitar 10 menit. Pembaruan sistem operasi tidak akan mengubah versi mesin DB atau kelas instans DB dari instans DB.

Pembaruan sistem operasi bisa opsional atau wajib:

- Pembaruan opsional dapat diterapkan kapan saja. Meskipun pembaruan ini bersifat opsional, sebaiknya Anda menerapkannya secara berkala agar armada RDS Anda tetap diperbarui. RDS tidak menerapkan pembaruan ini secara otomatis.

  Untuk menerima pemberitahuan saat patch sistem operasi opsional yang baru tersedia, Anda dapat berlangganan RDS-EVENT-0230 dalam kategori peristiwa patching keamanan. Untuk informasi tentang berlangganan peristiwa RDS, lihat Berlangganan pemberitahuan peristiwa Amazon RDS.

> **Note**
>
> `RDS-EVENT-0230` tidak berlaku untuk peningkatan distribusi sistem operasi.

> **Note**
>
> Jika Anda menerima `RDS-EVENT-0230` untuk instans DB RDS for SQL Server, pembaruan OS tidak dapat diterapkan melalui tindakan `apply-pending-maintenance`. Untuk informasi selengkapnya, lihat Menerapkan pembaruan untuk instans DB.

- Pembaruan wajib diperlukan dan memiliki tanggal penerapan. Rencanakan untuk menjadwalkan pembaruan sebelum tanggal penerapan ini. Setelah tanggal penerapan yang ditentukan, Amazon RDS secara otomatis meningkatkan sistem operasi untuk instans DB Anda ke versi terbaru selama salah satu periode pemeliharaan yang ditetapkan.

> **Note**
>
> Tetap mengikuti semua pembaruan opsional dan wajib mungkin diperlukan untuk memenuhi berbagai kewajiban kepatuhan. Sebaiknya Anda menerapkan semua pembaruan yang disediakan oleh RDS secara rutin selama periode pemeliharaan Anda.

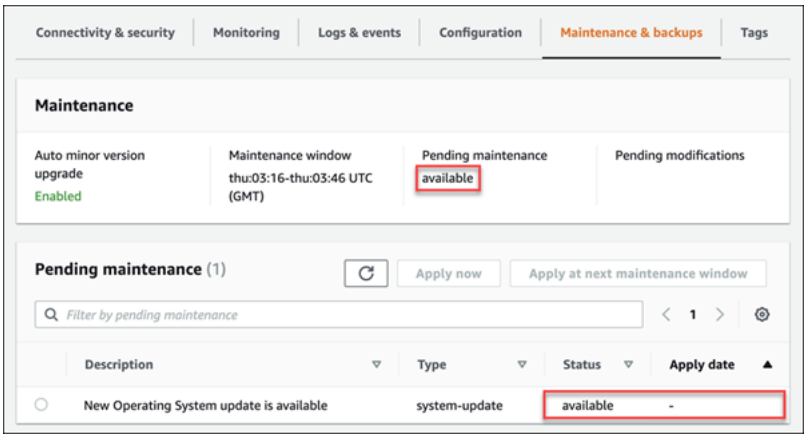
Anda dapat menggunakan AWS Management Console atau AWS CLI untuk mendapatkan informasi tentang jenis upgrade sistem operasi.

Konsol

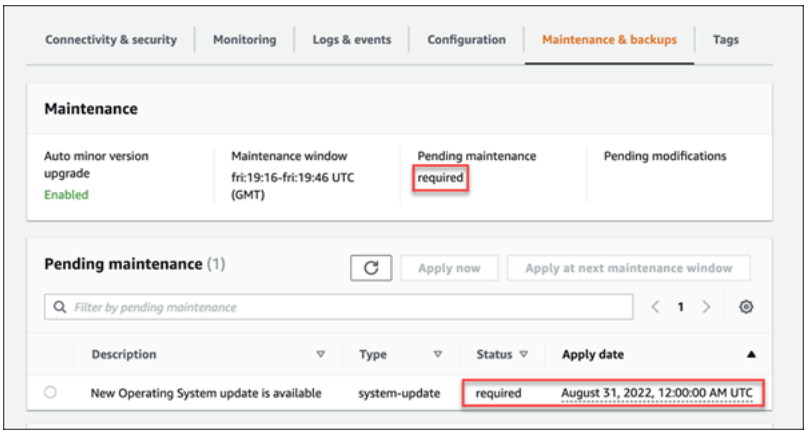
Untuk mendapatkan informasi pembaruan menggunakan AWS Management Console

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data, lalu pilih instans DB.
3. Pilih Pemeliharaan & pencadangan.
4. Di bagian Pemeliharaan tertunda, cari pembaruan sistem operasi, dan periksa nilai Status.

Di dalam AWS Management Console, pembaruan opsional memiliki Status pemeliharaan disetel ke tersedia dan tidak memiliki tanggal Terapkan, seperti yang ditunjukkan pada gambar berikut.



Nilai Status pemeliharaan pembaruan wajib diatur ke wajib dan memiliki Tanggal penerapan, seperti yang ditunjukkan pada gambar berikut.



AWS CLI

Untuk mendapatkan informasi pembaruan dari AWS CLI, gunakan describe-pending-maintenance-actionsperintah.

```
aws rds describe-pending-maintenance-actions
```

Pembaruan sistem operasi wajib mencakup nilai `AutoAppliedAfterDate` dan nilai `CurrentApplyDate`. Pembaruan sistem operasi opsional tidak mencakup nilai-nilai ini.

Output berikut menunjukkan pembaruan sistem operasi wajib.

```
{
```

```
  "ResourceIdentifier": "arn:aws:rds:us-east-1:123456789012:db:mydb1",
  "PendingMaintenanceActionDetails": [
    {
      "Action": "system-update",
      "AutoAppliedAfterDate": "2022-08-31T00:00:00+00:00",
      "CurrentApplyDate": "2022-08-31T00:00:00+00:00",
      "Description": "New Operating System update is available"
    }
  ]
}
```

Output berikut menunjukkan pembaruan sistem operasi opsional.

```
{
  "ResourceIdentifier": "arn:aws:rds:us-east-1:123456789012:db:mydb2",
  "PendingMaintenanceActionDetails": [
    {
      "Action": "system-update",
      "Description": "New Operating System update is available"
    }
  ]
}
```

Ketersediaan pembaruan sistem operasi

Pembaruan sistem operasi khusus untuk versi mesin DB dan kelas instans DB. Oleh karena itu, instans DB menerima atau memerlukan pembaruan di waktu yang berbeda. Ketika pembaruan sistem operasi tersedia untuk instans DB Anda berdasarkan versi mesin dan kelas instansnya, pembaruan akan muncul di konsol. Hal ini juga dapat dilihat dengan menjalankan AWS CLI describe-pending-maintenance-actionsperintah atau dengan memanggil operasi RDS DescribePendingMaintenanceActionsAPI. Jika pembaruan tersedia untuk instans Anda, Anda dapat memperbarui sistem operasi dengan mengikuti petunjuk di Menerapkan pembaruan untuk instans DB.

Jadwal pembaruan sistem operasi wajib

Kami berencana untuk menggunakan jadwal berikut untuk pembaruan sistem operasi wajib. Tanggal penerapan mengacu pada waktu saat Amazon RDS mulai menerapkan pembaruan wajib. Untuk setiap tanggal dalam tabel, waktu mulainya adalah 00.00 Waktu Universal Terkoordinasi (UTC).

Mesin DB	Tanggal penerapan
RDS for MySQL	30 Januari 2023
RDS for MariaDB	30 Januari 2023
RDS for PostgreSQL	31 Maret 2023

> **Note**
>
> Tanggal dalam tabel berlaku untuk pelanggan yang tidak mengalami pembaruan sistem operasi wajib pada tahun 2022. Untuk mengonfirmasi apakah pembaruan sistem operasi wajib pada tahun 2023 berdampak pada Anda, periksa bagian Pemeliharaan tertunda di konsol untuk pembaruan sistem operasi. Untuk informasi selengkapnya, lihat bagian Konsol di Bekerja dengan pembaruan sistem operasi.

Setelah tanggal penerapan, Amazon RDS secara otomatis meningkatkan sistem operasi untuk instans DB Anda ke versi terbaru pada periode pemeliharaan berikutnya. Untuk menghindari peningkatan otomatis, sebaiknya Anda menjadwalkan pembaruan sebelum tanggal penerapan.

Meng-upgrade versi mesin instans DB

Amazon RDS up-to-date Versi yang lebih baru dapat mencakup perbaikan bug, peningkatan keamanan, dan peningkatan lain untuk mesin basis data. Ketika Amazon RDS mendukung versi baru mesin basis data, Anda dapat memilih cara dan waktu upgrade instans DB basis data Anda.

Ada dua jenis upgrade: upgrade versi mayor dan upgrade versi minor. Secara umum, upgrade versi mesin mayor dapat menyebabkan perubahan yang tidak kompatibel dengan aplikasi yang ada. Sebaliknya, upgrade versi minor hanya mencakup perubahan yang memiliki kompatibilitas mundur dengan aplikasi yang ada.

Untuk klaster DB Multi-AZ, upgrade versi mayor hanya didukung untuk RDS for PostgreSQL. Upgrade versi minor didukung untuk semua mesin yang mendukung klaster DB Multi-AZ. Untuk informasi selengkapnya, lihat the section called “Memutakhirkan versi mesin klaster basis data Multi-AZ”.

Urutan penomoran versi bersifat khusus untuk setiap mesin basis data. Misalnya, RDS for MySQL 5.7 dan 8.0 adalah versi mesin mayor dan upgrade dari versi 5.7 ke versi 8.0 merupakan upgrade versi mayor. RDS for MySQL versi 5.7.22 dan 5.7.23 adalah versi minor dan upgrade dari 5.7.22 ke 5.7.23 merupakan upgrade versi minor.

> **Important**
>
> Anda tidak dapat memodifikasi instans DB saat sedang di-upgrade. Selama upgrade, status instans DB adalah `upgrading`.

Untuk informasi selengkapnya tentang upgrade versi mayor dan minor untuk mesin DB tertentu, lihat dokumentasi berikut untuk mesin DB Anda:

- Meningkatkan mesin DB MariaDB
- Meng-upgrade mesin DB Microsoft SQL Server
- Meng-upgrade mesin DB MySQL
- Meng-upgrade mesin DB Oracle
- Meningkatkan mesin DB PostgreSQL untuk Amazon RDS

Untuk upgrade versi mayor, Anda harus mengubah versi mesin DB secara manual melalui AWS Management Console, AWS CLI, atau API RDS. Untuk upgrade versi minor, Anda dapat mengubah versi mesin secara manual, atau memilih untuk mengaktifkan opsi Peningkatan versi minor otomatis.

> **Note**
>
> Upgrade mesin basis data memerlukan waktu henti. Anda dapat meminimalkan waktu henti yang diperlukan untuk upgrade instans DB dengan menggunakan deployment blue/green. Untuk informasi selengkapnya, lihat Menggunakan Deployment Blue/Green Amazon RDS untuk pembaruan basis data.

Topik

- Meng-upgrade versi mesin secara manual
- Meng-upgrade versi mesin minor secara otomatis

Meng-upgrade versi mesin secara manual

Untuk meng-upgrade versi mesin instans DB secara manual, Anda dapat menggunakan AWS Management Console, AWS CLI, atau API RDS.

Konsol

Untuk meng-upgrade versi mesin instans DB dengan menggunakan konsol

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data, lalu pilih instans DB yang ingin Anda upgrade.
3. Pilih Ubah. Halaman Modifikasi instans DB akan muncul.
4. Untuk Versi mesin DB, pilih versi baru.
5. Pilih Lanjutkan dan periksa ringkasan modifikasi.
6. Untuk segera menerapkan perubahan, pilih Terapkan segera. Dalam beberapa kasus, memilih opsi ini dapat menyebabkan pemadaman. Untuk informasi selengkapnya, lihat Menggunakan pengaturan Terapkan Segera.
7. Di halaman konfirmasi, tinjau perubahan Anda. Jika sudah benar, pilih Modifikasi instans DB untuk menyimpan perubahan Anda.

Alternatifnya, pilih Kembali untuk mengedit perubahan, atau pilih Batal untuk membatalkan perubahan Anda.

AWS CLI

Untuk meng-upgrade versi mesin dari instans DB, gunakan perintah CLI modify-db-instance. Tentukan parameter berikut:

- `--db-instance-identifier` – nama instans DB.
- `--engine-version` – nomor versi mesin basis data yang akan menjadi target upgrade.

  Untuk informasi tentang versi mesin yang valid, gunakan AWS CLI describe-db-engine-versionsperintah.
- `--allow-major-version-upgrade` – untuk meng-upgrade versi mayor.
- `--no-apply-immediately` – untuk menerapkan perubahan selama periode pemeliharaan berikutnya. Untuk segera menerapkan perubahan, gunakan `--apply-immediately`.

Example

Untuk Linux, macOS, atau Unix:

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

Untuk Windows:

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

API RDS

Untuk meng-upgrade versi mesin instans DB, gunakan tindakan ModifyDBInstance. Tentukan parameter berikut:

- `DBInstanceIdentifier` – nama instans DB, misalnya *`mydbinstance`*.
- `EngineVersion` – nomor versi mesin basis data yang akan menjadi target upgrade. Untuk informasi tentang versi mesin yang valid, gunakan operasi DescribeDB EngineVersions.
- `AllowMajorVersionUpgrade` – apakah mengizinkan upgrade versi mayor atau tidak. Untuk melakukannya, tetapkan nilainya ke `true`.
- `ApplyImmediately` – apakah akan segera menerapkan perubahan atau selama periode pemeliharaan berikutnya. Untuk segera menerapkan perubahan, tetapkan nilai ke `true`. Untuk menerapkan perubahan selama periode pemeliharaan berikutnya, tetapkan nilai ke `false`.

Meng-upgrade versi mesin minor secara otomatis

Versi mesin minor adalah pembaruan ke versi mesin DB dalam versi mesin mayor. Misalnya, versi mesin mayor adalah 9.6 dengan versi mesin minor 9.6.11 dan 9.6.12 di dalamnya.

Jika Anda ingin Amazon RDS meng-upgrade versi mesin DB dari basis data secara otomatis, Anda dapat mengaktifkan upgrade otomatis versi minor untuk basis data.

Topik

- Cara kerja upgrade versi minor otomatis
- Mengaktifkan upgrade versi minor otomatis
- Menentukan ketersediaan pembaruan pemeliharaan
- Menemukan target upgrade versi minor otomatis

Cara kerja upgrade versi minor otomatis

Amazon RDS menetapkan versi mesin minor sebagai versi mesin minor pilihan ketika kondisi berikut terpenuhi:

- Basis data menjalankan versi minor mesin DB yang lebih rendah daripada versi minor mesin pilihan.

Anda dapat menemukan versi mesin Anda saat ini untuk instans DB Anda dengan melihat tab Konfigurasi di halaman detail basis data atau menjalankan perintah CLI `describe-db-instances`.

- Basis data memiliki upgrade otomatis versi minor yang diaktifkan.

RDS menjadwalkan upgrade untuk berjalan secara otomatis pada periode pemeliharaan. Selama upgrade, RDS melakukan langkah-langkah dasar berikut:

1. Menjalankan pra-pemeriksaan untuk memastikan basis data berkondisi baik dan siap untuk di-upgrade
2. Meng-upgrade mesin DB
3. Menjalankan pemeriksaan pasca-upgrade
4. Menandai upgrade basis data sebagai selesai

Upgrade otomatis menimbulkan waktu henti. Durasi waktu henti tergantung pada berbagai faktor, termasuk jenis mesin DB dan ukuran basis data.

Mengaktifkan upgrade versi minor otomatis

Anda dapat mengontrol apakah akan mengaktifkan upgrade otomatis versi minor untuk instans DB ketika Anda melakukan tugas-tugas berikut:

- Membuat instans DB
- Memodifikasi instans DB
- Membuat replika baca
- Memulihkan instans DB dari snapshot
- Memulihkan instans DB ke waktu tertentu
- Mengimpor instans DB dari Amazon S3 (untuk cadangan MySQL di Amazon S3)

Saat melakukan tugas ini, Anda dapat mengontrol apakah akan mengaktifkan upgrade otomatis versi minor untuk instans DB dengan cara berikut:

- Dengan konsol, atur opsi Peningkatan versi minor otomatis.
- Dengan AWS CLI, atur opsi `--auto-minor-version-upgrade|--no-auto-minor-version-upgrade`.

- Dengan API RDS, atur parameter `AutoMinorVersionUpgrade`.

Menentukan ketersediaan pembaruan pemeliharaan

Untuk mengetahui apakah pembaruan pemeliharaan, seperti upgrade versi mesin DB, tersedia untuk instans DB Anda, gunakan konsol, AWS CLI, atau API RDS. Anda juga dapat meng-upgrade versi mesin DB secara manual dan menyesuaikan periode pemeliharaan. Untuk informasi selengkapnya, lihat Memelihara instans DB.

Menemukan target upgrade versi minor otomatis

Anda dapat menggunakan perintah AWS CLI berikut untuk menentukan versi target upgrade minor otomatis saat ini untuk versi mesin DB minor yang ditentukan di Wilayah AWS spesifik. Anda dapat menemukan nilai `--engine` yang mungkin untuk perintah ini dalam deskripsi untuk parameter `Engine` di CreateDBInstance.

Untuk Linux, macOS, atau Unix:

```
aws rds describe-db-engine-versions \
--engine engine \
--engine-version minor-version \
--region region \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
--output text
```

Untuk Windows:

```
aws rds describe-db-engine-versions ^
--engine engine ^
--engine-version minor-version ^
--region region ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output text
```

Misalnya, perintah AWS CLI berikut menentukan target upgrade minor otomatis untuk MySQL versi minor 8.0.11 di Wilayah AWS AS Timur (Ohio) (us-east-2).

Untuk Linux, macOS, atau Unix:

```
aws rds describe-db-engine-versions \
--engine mysql \
--engine-version 8.0.11 \
--region us-east-2 \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
--output table
```

Untuk Windows:

```
aws rds describe-db-engine-versions ^
--engine mysql ^
--engine-version 8.0.11 ^
--region us-east-2 ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output table
```

Output Anda akan seperti yang berikut ini.

```
---------------------------------
|    DescribeDBEngineVersions    |
+--------------+-----------------+
|  AutoUpgrade |  EngineVersion  |
+--------------+-----------------+
|  False       |  8.0.15         |
|  False       |  8.0.16         |
|  False       |  8.0.17         |
|  False       |  8.0.19         |
|  False       |  8.0.20         |
|  False       |  8.0.21         |
|  True        |  8.0.23         |
|  False       |  8.0.25         |
+--------------+-----------------+
```

Dalam contoh ini, nilai `AutoUpgrade` adalah `True` untuk MySQL versi 8.0.23. Jadi, target upgrade minor otomatis adalah MySQL versi 8.0.23, yang disorot dalam output.

Important

Jika Anda berencana untuk memigrasikan instans DB RDS for PostgreSQL ke klaster DB Aurora PostgreSQL, kami sangat menyarankan Anda menonaktifkan upgrade versi minor otomatis untuk instans DB di awal selama perencanaan. Migrasi ke Aurora PostgreSQL mungkin tertunda jika versi RDS for PostgreSQL belum didukung oleh Aurora PostgreSQL. Untuk informasi tentang versi Aurora PostgreSQL, lihat Versi mesin untuk Amazon Aurora PostgreSQL.

Mengganti nama instans DB

Anda dapat mengubah nama instans DB dengan menggunakan perintah AWS Management Console, AWS CLI `modify-db-instance`, atau tindakan API Amazon RDS `ModifyDBInstance`. Mengganti nama instans DB dapat memiliki efek yang luas. Berikut ini adalah daftar pertimbangan sebelum Anda mengubah nama instans DB.

- Saat Anda mengubah nama instans DB, titik akhir untuk instans DB berubah karena URL-nya menyertakan nama yang Anda tetapkan ke instans DB tersebut. Anda harus selalu mengalihkan lalu lintas dari URL lama ke yang baru.
- Saat Anda mengubah nama instans DB, nama DNS lama yang digunakan oleh instans DB akan segera dihapus, meskipun dapat tetap di-caching selama beberapa menit. Nama DNS baru untuk instans DB yang diubah namanya akan menjadi efektif dalam waktu sekitar 10 menit. Instans DB yang diubah namanya tidak tersedia hingga nama baru menjadi efektif.
- Anda tidak dapat menggunakan nama instans DB yang sudah ada saat melakukan penggantian nama suatu instans.
- Semua replika baca yang terkait dengan sebuah instans DB akan tetap terkait dengan instans tersebut setelah namanya diubah. Misalnya, anggaplah Anda memiliki instans DB yang melayani basis data produksi Anda dan instans tersebut memiliki beberapa replika baca terkait. Jika Anda mengubah nama instans DB lalu menggantinya di lingkungan produksi dengan snapshot DB, instans DB yang Anda ubah namanya masih akan memiliki replika baca yang terkait dengannya.
- Metrik dan peristiwa yang terkait dengan nama instans DB akan dipertahankan jika Anda menggunakan ulang nama instans DB tersebut. Misalnya, jika Anda mempromosikan replika baca dan mengubah namanya menjadi nama instans DB primer sebelumnya, peristiwa dan metrik yang terkait dengan instans DB primer ini akan dikaitkan dengan instans yang diubah namanya.
- Tag instans DB akan dipertahankan dengan instans DB, terlepas dari perubahan namanya.
- Snapshot DB dipertahankan untuk instans DB yang diubah namanya.

> **Note**
>
> Instans DB adalah lingkungan basis data terisolasi yang berjalan di cloud. Instans DB dapat meng-host banyak basis data, atau satu basis data Oracle dengan beberapa skema. Untuk informasi tentang mengganti nama basis data, lihat dokumentasi untuk mesin DB Anda.

Mengubah nama untuk mengganti instans DB yang ada

Alasan paling umum untuk mengganti nama instans DB adalah bahwa Anda mempromosikan replika baca atau Anda memulihkan data dari snapshot atau point-in-time pemulihan DB (PITR). Dengan mengubah nama basis data, Anda dapat mengganti instans DB tanpa harus mengubah kode aplikasi apa pun yang mengacu pada instans DB ini. Dalam kasus ini, Anda akan melakukan hal berikut:

1. Hentikan semua lalu lintas ke instans DB primer. Hal ini dapat dilakukan dengan pengalihan lalu lintas dari mengakses basis data di instans DB atau cara lain yang ingin Anda gunakan untuk mencegah lalu lintas mengakses basis data Anda di instans DB.
2. Ubah nama instans DB primer dengan nama yang menunjukkan bahwa instans ini bukan lagi instans DB primer seperti yang dijelaskan nanti dalam topik ini.
3. Buat instans DB primer baru dengan memulihkan dari snapshot DB atau dengan mempromosikan replika baca, lalu beri instans baru ini nama yang sama dengan nama instans DB primer sebelumnya.
4. Kaitkan replika baca dengan instans DB primer baru.

Jika Anda menghapus instans DB primer lama, Anda bertanggung jawab untuk menghapus setiap snapshot DB yang tidak diinginkan dari instans DB primer lama.

Untuk informasi tentang mempromosikan replika baca, lihat Mempromosikan replika baca menjadi instans DB mandiri.

> **Important**
>
> Instans DB di-boot ulang saat diubah namanya.

Konsol

Untuk mengubah nama instans DB

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data.
3. Pilih instans DB yang ingin Anda ubah namanya.
4. Pilih Ubah.

5. Di Pengaturan, masukkan nama baru untuk Pengidentifikasi instans DB.
6. Pilih Lanjutkan.
7. Untuk menerapkan perubahan dengan serta-merta, pilih Terapkan seketika. Dalam beberapa kasus, memilih opsi ini dapat menyebabkan pemadaman. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.
8. Di halaman konfirmasi, tinjau perubahan Anda. Jika sudah benar, pilih Modifikasi Instans DB untuk menyimpan perubahan Anda.

   Alternatifnya, pilih Kembali untuk mengedit perubahan, atau pilih Batal untuk membatalkan perubahan Anda.

AWS CLI

Untuk mengubah nama instans DB, gunakan perintah AWS CLI`modify-db-instance`. Berikan nama baru instans DB untuk nilai `--db-instance-identifier` dan parameter `--new-db-instance-identifier` saat ini.

Example

Untuk Linux, macOS, atau Unix:

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

Untuk Windows:

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

API RDS

Untuk mengubah nama instans DB, panggil fungsi API Amazon RDS `ModifyDBInstance` dengan parameter berikut ini:

- `DBInstanceIdentifier` — nama yang ada untuk instans
- `NewDBInstanceIdentifier` — nama yang baru untuk instans

Mem-boot ulang instans DB

Anda dapat menghentikan dan memulai layanan database pada instans RDS DB Anda dalam satu operasi, yang disebut reboot.

Topik

Gunakan kasus untuk me-reboot instans DB cluster DB

Biasanya, Anda me-reboot instans DB Anda untuk alasan pemeliharaan sehingga perubahan Anda berlaku. Kasus penggunaan berikut adalah umum:

- Mengaitkan grup parameter DB baru - Saat Anda mengaitkan grup parameter DB baru dengan instans DB, RDS menerapkan parameter statis dan dinamis yang dimodifikasi hanya setelah instans DB di-boot ulang. Namun, jika Anda memodifikasi parameter dinamis dalam grup parameter DB setelah Anda mengaitkannya dengan instans DB, perubahan ini diterapkan segera tanpa reboot. Untuk informasi selengkapnya, lihat Bekerja dengan grup parameter.
- Menerapkan perubahan ke parameter statis dalam grup parameter DB yang ada — Saat Anda mengubah parameter statis dan menyimpan grup parameter DB, status instance DB yang terkait dengan grup parameter ini di konsol berubah menjadi pending-reboot. Perubahan parameter berlaku hanya setelah instance DB terkait di-boot ulang. Saat Anda mengubah parameter dinamis dalam grup parameter yang ada, perubahan akan segera berlaku secara default, tanpa memerlukan reboot.

> **Note**
>
> Status pending-reboot tidak menghasilkan reboot otomatis selama jendela pemeliharaan berikutnya. Untuk menerapkan perubahan parameter terbaru ke instans DB Anda, reboot

instans DB secara manual. Lihat informasi lebih lanjut tentang grup parameter di Bekerja dengan grup parameter.

- Menguji failover Multi-AZ — Strategi pengujian Anda untuk klaster DB multi-AZ mungkin melibatkan reboot instans DB utama Anda untuk memulai failover ke AZ yang berbeda.
- Pemecahan masalah — Anda mungkin mengalami kinerja atau masalah operasional lainnya yang memerlukan reboot. Misalnya, instans DB Anda mungkin tidak responsif.

Cara me-reboot instans DB

Saat Amazon RDS me-reboot instans DB Anda, instans DB melakukan tugas berurutan berikut:

1. Menghentikan layanan database pada instans DB Anda
2. Memulai layanan database pada instans DB Anda

Proses reboot menyebabkan pemadaman singkat. Selama pemadaman ini, status instans DB sedang reboot. Pemadaman akan terjadi untuk deployment AZ Tunggal dan deployment instans DB Multi-AZ, bahkan saat Anda melakukan boot ulang dengan failover.

Cara me-reboot instans DB dalam penerapan Multi-AZ bekerja

Jika instans Amazon RDS DB berada dalam penerapan Multi-AZ, Anda dapat melakukan reboot dengan failover. Operasi ini berguna untuk mensimulasikan kegagalan instans DB atau mengembalikan operasi ke Availability Zone asli setelah failover.

Selama reboot dengan failover, Amazon RDS melakukan hal berikut

- Menginterupsi database secara tiba-tiba. Instans DB dan sesi kliennya mungkin tidak memiliki waktu untuk dinonaktifkan dengan normal.

> **Warning**
>
> Untuk menghindari kemungkinan kehilangan data, sebaiknya hentikan transaksi pada instans DB Anda sebelum mem-boot ulang dengan failover.

- Beralih ke replika siaga di AZ lain secara otomatis. Perubahan AZ mungkin tidak tercermin dalam AWS Management Console, dan dalam panggilan ke AWS CLI dan RDS API, selama beberapa menit.

- Memperbarui catatan DNS untuk instans DB untuk menunjuk ke instans DB siaga. Oleh karena itu, Anda perlu membersihkan dan membuat kembali koneksi yang sudah ada ke instans DB Anda. Untuk informasi selengkapnya, lihat Mengonfigurasi dan mengelola deployment Multi-AZ.
- Membuat acara Amazon RDS setelah reboot.

Pada RDS untuk Microsoft SQL Server, failover hanya me-reboot instans DB utama. Setelah failover, instans DB primer akan menjadi instans DB sekunder baru. Parameter mungkin tidak diperbarui untuk instans Multi-AZ. Untuk mem-boot ulang tanpa failover, instans DB primer dan sekunder, serta parameter diperbarui setelah boot ulang. Jika instans DB tidak responsif, kami sarankan boot ulang tanpa failover.

Pertimbangan saat me-reboot instans DB cluster DB

Sebelum Anda me-reboot instance Anda, pertimbangkan hal berikut:

- Untuk replika baca dengan instans DB, Anda dapat mem-boot ulang instans DB sumber dan replika bacanya secara terpisah. Setelah boot ulang selesai, replikasi akan berlanjut secara otomatis.
- Waktu reboot tergantung pada proses pemulihan kerusakan, aktivitas database pada saat reboot, dan perilaku mesin DB spesifik Anda. Untuk meningkatkan waktu reboot, kami sarankan Anda mengurangi aktivitas database sebanyak mungkin selama reboot. Teknik ini mengurangi aktivitas rollback untuk transaksi dalam perjalanan.

Pastikan Anda memenuhi prasyarat berikut:

- Instans DB Anda harus berada dalam status `available`. Database Anda mungkin tidak tersedia karena beberapa alasan, seperti pencadangan yang sedang berlangsung, modifikasi yang diminta sebelumnya, atau operasi jendela pemeliharaan.
- Jika Anda memaksakan failover ke AZ yang berbeda, instans DB Anda harus dikonfigurasi untuk Multi-AZ.
- Jika Anda memaksakan failover ke AZ yang berbeda, sebaiknya hentikan transaksi pada instans DB Anda terlebih dahulu untuk mencegah kemungkinan kehilangan data.

Mem-boot ulang instans DB dasar

Anda dapat me-reboot instans DB Anda menggunakan AWS Management Console, AWS CLI, atau RDS API.

Konsol

Untuk mem-boot ulang instans DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data, lalu pilih instans DB yang ingin Anda boot ulang.
3. Untuk Tindakan, pilih Boot ulang.

   Halaman instans Reboot DB muncul.
4. (Opsional) Pilih Boot ulang dengan failover? untuk memaksa failover dari satu AZ ke AZ lainnya.
5. Pilih Boot ulang untuk mem-boot ulang instans DB Anda.

   Alternatifnya, pilih Batal.

AWS CLI

Untuk me-reboot instance DB dengan menggunakan AWS CLI, panggil `reboot-db-instance`perintah.

Example Boot ulang sederhana

Untuk Linux, macOS, atau Unix:

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

Untuk Windows:

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

Example Boot ulang dengan failover

Untuk memaksa failover dari satu AZ ke yang lain dalam cluster DB multi-AZ, gunakan parameter. `--force-failover`

Untuk Linux, macOS, atau Unix:

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

Untuk Windows:

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

API RDS

Untuk mem-boot ulang instans DB menggunakan API Amazon RDS, panggil operasi `RebootDBInstance`.

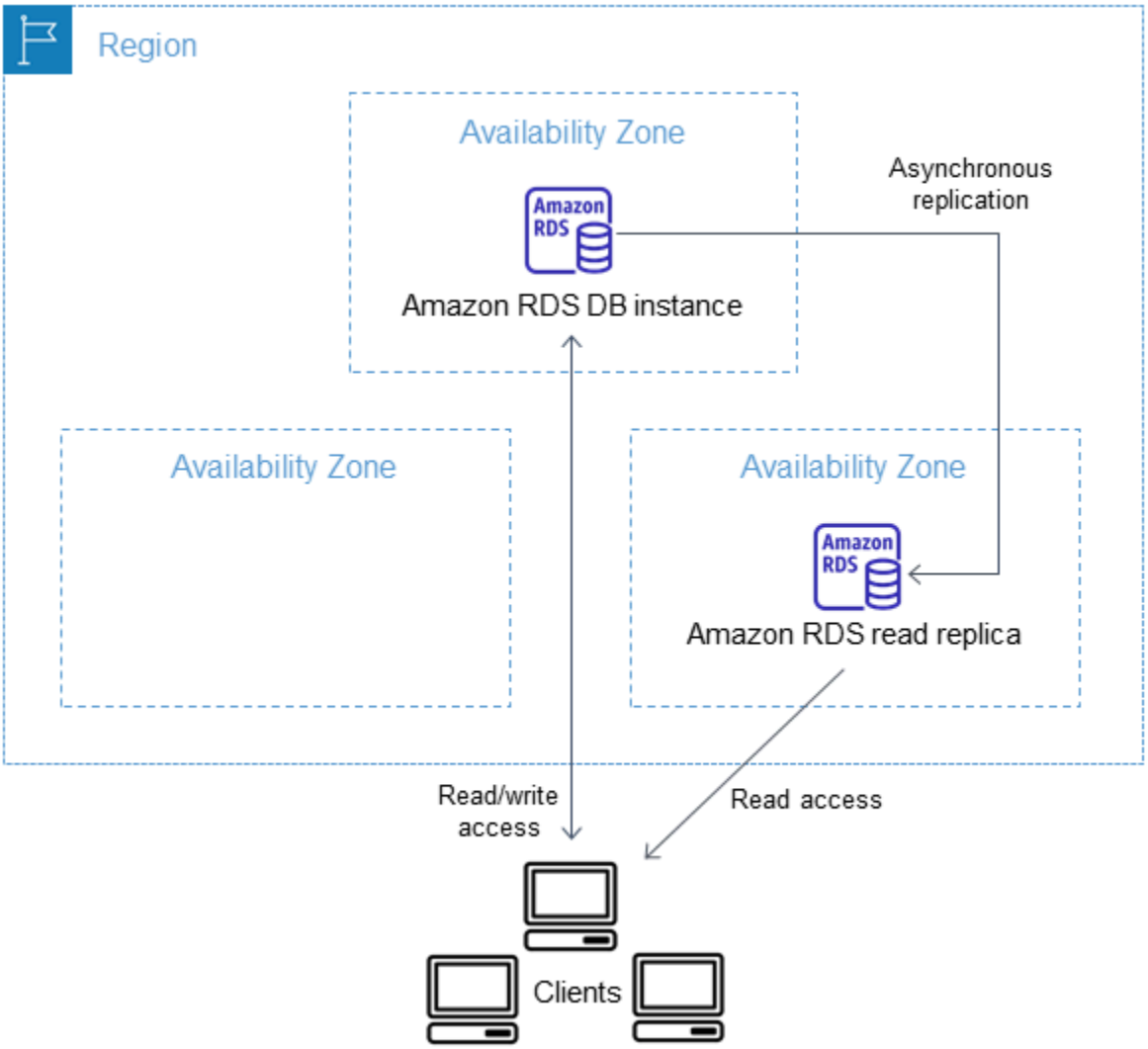
Menggunakan replika baca instans DB

Replika baca adalah salinan hanya baca dari instans DB. Anda dapat mengurangi beban pada instans DB primer dengan merutekan kueri baca dari aplikasi Anda ke replika baca. Dengan cara ini, Anda dapat secara elastis melakukan penskalaan ke luar melampaui batasan kapasitas instans DB tunggal untuk beban kerja basis data yang sarat pembacaan.

Untuk membuat replika baca dari instans DB sumber, Amazon RDS menggunakan fitur replikasi bawaan mesin DB. Untuk informasi tentang menggunakan replika baca dengan mesin tertentu, lihat bagian berikut:

- Menggunakan replika baca MariaDB
- Menggunakan replika baca untuk Microsoft SQL Server di Amazon RDS
- Menggunakan replika baca MySQL
- Menggunakan replika baca untuk Amazon RDS for Oracle
- Menggunakan replika baca untuk Amazon RDS for PostgreSQL

Setelah Anda membuat replika baca dari instans DB sumber, instans DB sumber ini menjadi instans DB primer. Saat Anda membuat pembaruan instans DB primer, Amazon RDS menyalinnya secara asinkron ke replika baca. Diagram berikut menunjukkan instans DB sumber yang mereplikasi ke replika baca di Zona Ketersediaan (AZ) yang berbeda. Klien memiliki akses baca/tulis ke instans DB primer dan akses hanya baca ke replika.



Topik

- Gambaran umum replika baca Amazon RDS
- Membuat replika baca
- Mempromosikan replika baca menjadi instans DB mandiri
- Memantau replikasi baca
- Membuat replika baca di tempat yang berbeda Wilayah AWS

Gambaran umum replika baca Amazon RDS

Bagian berikut membahas replika baca instans DB. Untuk informasi tentang replika baca klaster DB Multi-AZ, lihat the section called “Bekerja dengan replika baca klaster DB multi-AZ”.

Topik

Kasus penggunaan untuk replika baca

Melakukan deployment satu atau beberapa replika baca untuk instans DB sumber tertentu mungkin masuk akal dalam berbagai skenario, termasuk skenario berikut ini:

- Penskalaan di luar kapasitas komputasi atau I/O dari instans DB tunggal untuk beban kerja basis data yang sarat pembacaan. Anda dapat mengarahkan kelebihan lalu lintas baca ini ke satu atau beberapa replika baca.
- Melayani lalu lintas baca saat instans DB sumber tidak tersedia. Dalam beberapa kasus, instans DB sumber Anda mungkin tidak dapat menerima permintaan I/O, misalnya karena penangguhan I/O untuk cadangan atau pemeliharaan terjadwal. Dalam hal ini, Anda dapat mengarahkan lalu lintas baca ke replika baca Anda. Untuk kasus penggunaan ini, perlu diingat bahwa data di replika baca mungkin "usang" karena instans DB sumber tidak tersedia.
- Skenario pelaporan bisnis atau pergudangan data saat Anda mungkin ingin kueri pelaporan bisnis dijalankan terhadap replika baca, bukan instans DB produksi Anda.
- Menerapkan pemulihan bencana. Anda dapat mempromosikan replika baca menjadi instans mandiri sebagai solusi pemulihan bencana jika instans DB primer mengalami kegagalan.

Cara kerja replika baca

Saat Anda membuat replika baca, tentukan terlebih dahulu instans DB yang ada sebagai sumber. Kemudian, Amazon akan RDS mengambil snapshot dari instans sumber dan membuat Instans hanya baca dari snapshot. Amazon RDS selanjutnya menggunakan metode replikasi asinkron untuk mesin DB untuk memperbarui replika baca setiap kali ada perubahan pada instans DB primer.

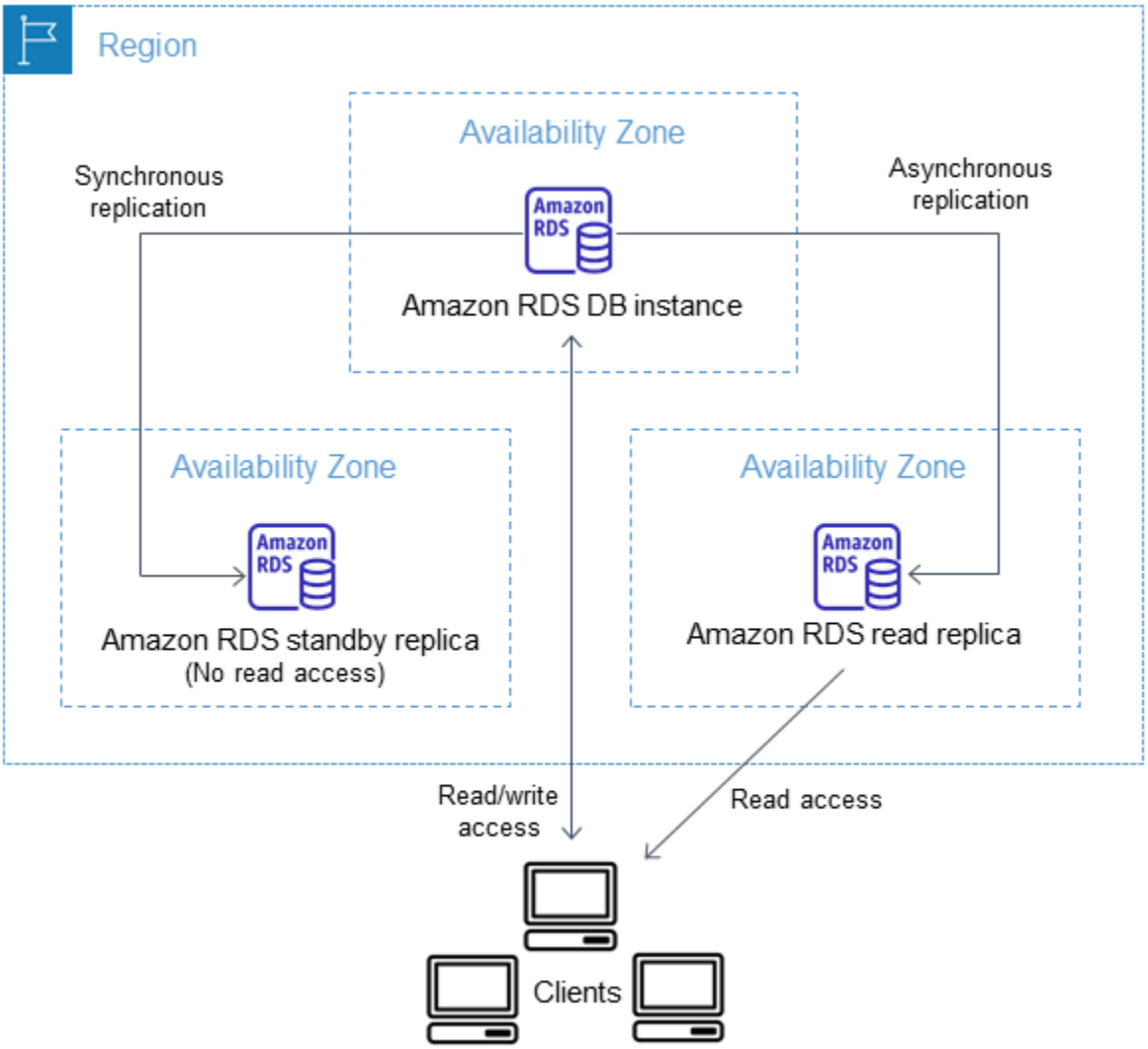
Replika baca beroperasi sebagai instans DB yang hanya memungkinkan koneksi hanya baca. Pengecualiannya adalah mesin DB RDS for Oracle, yang mendukung basis data replika dalam mode terpasang. Replika terpasang tidak menerima koneksi pengguna sehingga tidak dapat melayani beban kerja hanya baca. Penggunaan primer untuk replika yang terpasang adalah pemulihan bencana lintas Wilayah. Untuk informasi selengkapnya, lihat Menggunakan replika baca untuk Amazon RDS for Oracle.

Aplikasi terhubung ke replika baca seperti terhubung ke instans DB apa pun. Amazon RDS mereplikasi semua basis data dari instans DB sumber.

Replika baca dalam deployment Multi-AZ

Anda dapat mengonfigurasi replika baca untuk instans DB yang juga memiliki replika siaga yang dikonfigurasi untuk ketersediaan tinggi dalam deployment Multi-AZ. Replikasi dengan replika siaga bersifat sinkron. Tidak seperti replika baca, replika siaga tidak dapat melayani lalu lintas baca.

Dalam skenario berikut, klien memiliki akses baca/tulis ke instans DB primer dalam satu AZ. Instans primer menyalin pembaruan secara asinkron ke replika baca di AZ kedua dan juga menyalinnya secara sinkron ke replika siaga di AZ ketiga. Klien hanya memiliki akses baca ke replika baca.



Untuk informasi selengkapnya tentang replika ketersediaan tinggi dan siaga, lihat Mengonfigurasi dan mengelola deployment Multi-AZ.

Replika baca lintas wilayah

Dalam beberapa kasus, replika baca berada di tempat yang berbeda Wilayah AWS dari instance DB utamanya. Dalam kasus ini, Amazon RDS menyiapkan saluran komunikasi yang aman antara instans DB primer dan replika baca. Amazon RDS menetapkan konfigurasi AWS keamanan apa pun yang diperlukan untuk mengaktifkan saluran aman, seperti menambahkan entri grup keamanan. Untuk informasi tentang replika baca lintas Wilayah, lihat Membuat replika baca di tempat yang berbeda Wilayah AWS.

Informasi dalam Bab ini berlaku untuk membuat replika baca Amazon RDS baik dalam hal yang Wilayah AWS sama dengan instans DB sumber, atau terpisah. Wilayah AWS Informasi berikut tidak

berlaku untuk mengatur replikasi dengan instans yang berjalan pada instans Amazon EC2 atau yang berada di on-premise.

Perbedaan di antara beberapa replika baca untuk mesin DB

Karena mesin DB Amazon RDS menerapkan replikasi secara berbeda, ada beberapa perbedaan signifikan yang harus Anda ketahui, seperti yang diperlihatkan dalam tabel berikut.

Fitur atau perilaku	MySQL dan MariaDB	Oracle	PostgreSQL	SQL Server
Apa itu metode replikasi?	Replikasi logis.	Replikasi fisik.	Replikasi fisik.	Replikasi fisik.
Bagaimana log transaksi di-purging?	RDS for MySQL dan RDS for MariaDB mempertahankan log biner yang belum diterapkan.	Jika instans DB primer tidak memiliki replika baca lintas Wilayah, Amazon RDS for Oracle mempertahankan minimal dua jam log transaksi pada instans DB sumber. Log dihapus dari instans DB sumber setelah dua jam atau setelah pengaturan jam retensi log arsip berlalu, mana saja yang lebih lama. Log dihapus dari replika baca setelah pengaturan jam retensi log arsip telah berlalu hanya	PostgreSQL memiliki parameter `wal_keep_segments` yang menentukan berapa banyak file write ahead log (WAL) yang dipertahankan untuk menyediakan data ke replika baca. Nilai parameter menentukan jumlah log yang akan dipertahankan.	File Log Virtual (VLF) dari file log transaksi di replika primer dapat dipotong setelah tidak lagi diperlukan untuk replika sekunder. VLF hanya dapat ditandai sebagai tidak aktif jika catatan log telah di-hardening di replika.

Fitur atau perilaku	MySQL dan MariaDB	Oracle	PostgreSQL	SQL Server
		jika telah berhasil diterapkan ke basis data. Dalam beberapa kasus, instans DB primer mungkin memiliki satu atau beberapa replika baca lintas Wilayah. Jika demikian, Amazon RDS for Oracle akan mempertahankan log transaksi pada instans DB sumber sampai log transaksi tersebut dikirim dan diterapkan ke semua replika baca lintas Wilayah. Untuk informasi tentang mengatur jam retensi log arsip, lihat Mempertahankan log pengulangan yang diarsipkan.		Terlepas dari seberapa cepat subsistem disk berada di replika primer, log transaksi akan mempertahankan VLF hingga replika yang paling lambat telah melakukan hardening terhadap VLF ini.

Fitur atau perilaku	MySQL dan MariaDB	Oracle	PostgreSQL	SQL Server
Bisakah replika dijadikan dapat ditulis?	Ya. Anda dapat memungkinkan replika baca MySQL atau MariaDB menjadi dapat ditulis.	Tidak. Replika baca Oracle adalah salinan fisik, dan Oracle tidak mengizinkan penulisan di replika baca. Anda dapat mempromosikan replika baca agar menjadikannya dapat ditulis. Replika baca yang dipromosikan memiliki data yang direplikasi hingga titik saat ada permintaan untuk mempromosikannya.	Tidak. Replika baca PostgreSQL adalah salinan fisik, dan PostgreSQL tidak mengizinkan replika baca dijadikan dapat ditulis.	Tidak. Replika baca SQL Server adalah salinan fisik, dan SQL Server tidak mengizinkan penulisan di replika baca. Anda dapat mempromosikan replika baca agar menjadikannya dapat ditulis. Replika baca yang dipromosikan memiliki data yang direplikasi hingga titik saat ada permintaan untuk mempromosikannya.

Fitur atau perilaku	MySQL dan MariaDB	Oracle	PostgreSQL	SQL Server
Dapatkah pencadangan dilakukan pada replika?	Ya. Pencadangan otomatis dan snapshot manual didukung di replika baca RDS for MySQL atau RDS for MariaDB.	Ya. Pencadangan otomatis dan snapshot manual didukung di replika baca RDS for Oracle.	Ya, Anda dapat membuat snapshot manual replika baca RDS for PostgreSQL. Pencadangan otomatis untuk replika baca didukung untuk RDS for PostgreSQL 14.1 dan versi yang lebih tinggi saja. Anda tidak dapat mengaktifkan pencadangan otomatis untuk replika baca PostgreSQL untuk versi RDS for PostgreSQL yang lebih lama dari 14.1. Untuk RDS for PostgreSQL 13 dan versi yang lebih lama, buat snapshot dari replika baca jika Anda menginginkan cadangannya.	Tidak. Pencadangan otomatis dan snapshot manual tidak didukung di replika baca RDS for SQL Server.

Fitur atau perilaku	MySQL dan MariaDB	Oracle	PostgreSQL	SQL Server
Bisakah Anda menggunakan replikasi paralel?	Ya. Semua versi MariaDB dan MySQL yang didukung memungkinkan thread replikasi paralel.	Ya. Data log redo selalu dikirim secara paralel dari basis data primer ke semua replika bacanya.	Tidak. PostgreSQL memiliki satu replikasi penanganan proses.	Ya. Data log redo selalu dikirim secara paralel dari basis data primer ke semua replika bacanya.
Dapatkah Anda mempertahankan replika dalam mode terpasang alih-alih mode hanya baca?	Tidak.	Ya. Penggunaan primer untuk replika yang terpasang adalah pemulihan bencana lintas Wilayah. Lisensi Active Data Guard tidak diperlukan untuk replika yang terpasang. Untuk informasi selengkapnya, lihat Menggunakan replika baca untuk Amazon RDS for Oracle.	Tidak.	Tidak.

Jenis penyimpanan replika baca

Secara default, replika baca dibuat dengan jenis penyimpanan yang sama dengan instans DB sumber. Namun, Anda dapat membuat replika baca yang memiliki jenis penyimpanan berbeda dari instans DB sumber berdasarkan opsi yang tercantum di tabel berikut.

Jenis penyimpanan instans DB sumber	Alokasi penyimpanan instans DB sumber	Opsi jenis penyimpanan replika baca
IOPS yang Tersedia	100 GiB–64 TiB	IOPS yang Tersedia, Tujuan Umum, Magnetik
Tujuan Umum	100 GiB–64 TiB	IOPS yang Tersedia, Tujuan Umum, Magnetik
Tujuan Umum	<100 GiB	Tujuan Umum, Magnetik
Magnetik	100 GiB–6 TiB	IOPS yang Tersedia, Tujuan Umum, Magnetik
Magnetik	<100 GiB	Tujuan Umum, Magnetik

> **Note**
>
> Saat Anda meningkatkan alokasi penyimpanan replika baca, jumlahnya minimal harus 10 persen. Jika Anda mencoba meningkatkan nilai sebesar kurang dari 10 persen, Anda akan mendapat kesalahan.

Batasan untuk membuat replika dari replika

Amazon RDS tidak mendukung replikasi sirkular. Anda tidak dapat mengonfigurasi instans DB agar berfungsi sebagai sumber replikasi untuk instans DB yang ada. Anda hanya dapat membuat replika baca baru dari instans DB yang ada. Misalnya, jika **`MySourceDBInstance`** mereplikasi ke **`ReadReplica1`**, Anda tidak dapat mengonfigurasi **`ReadReplica1`** untuk mereplikasi kembali ke **`MySourceDBInstance`**.

Untuk RDS for MariaDB dan RDS for MySQL, dan untuk versi RDS for PostgreSQL tertentu, Anda dapat membuat replika baca dari replika baca yang sudah ada. Misalnya, Anda dapat membuat replika baca baru **`ReadReplica2`** dari replika **`ReadReplica1`** yang sudah ada. Untuk RDS for Oracle dan RDS for SQL Server, Anda tidak dapat membuat replika baca dari replika baca yang sudah ada.

Pertimbangan saat menghapus replika

Jika tidak lagi membutuhkan replika baca, Anda dapat secara eksplisit menghapusnya menggunakan mekanisme yang sama untuk menghapus instans DB. Jika Anda menghapus instans DB sumber tanpa menghapus replika bacanya dalam hal yang sama Wilayah AWS, setiap replika baca dipromosikan ke instans DB mandiri. Untuk informasi tentang menghapus instans DB, lihat Menghapus instans DB. Untuk informasi tentang promosi replika baca, lihat Mempromosikan replika baca menjadi instans DB mandiri.

Jika Anda memiliki replika baca lintas Wilayah, lihat Pertimbangan replikasi lintas Wilayah untuk informasi terkait penghapusan instans DB sumber untuk replika baca lintas Wilayah.

Membuat replika baca

Anda dapat membuat replika baca dari instans DB yang ada menggunakan AWS Management Console, AWS CLI, atau RDS API. Anda membuat replika baca dengan menentukan `SourceDBInstanceIdentifier`, yang merupakan pengidentifikasi instans DB dari instans DB sumber yang ingin Anda replikasi.

Saat membuat replika baca, Amazon RDS mengambil snapshot DB dari instans DB sumber Anda dan memulai replikasi. Akibatnya, Anda mengalami penangguhan I/O singkat pada instans DB sumber Anda saat snapshot DB terjadi.

> **Note**
>
> Penangguhan I/O biasanya berlangsung sekitar satu menit. Anda dapat menghindari penangguhan I/O jika instans DB sumber adalah deployment Multi-AZ, karena dalam hal ini, snapshot diambil dari instans DB sekunder.

Transaksi aktif yang berjalan lama dapat memperlambat proses pembuatan replika baca. Kami menyarankan Anda menunggu transaksi yang berjalan lama selesai sebelum membuat replika baca. Jika Anda membuat beberapa replika baca secara paralel dari instans DB sumber yang sama, Amazon RDS hanya mengambil satu snapshot di awal tindakan pembuatan pertama.

Saat membuat replika baca, ada beberapa hal yang perlu dipertimbangkan. Pertama, Anda harus mengaktifkan pencadangan otomatis pada instans DB sumber dengan mengatur periode retensi cadangan ke nilai selain 0. Persyaratan ini juga berlaku untuk replika baca yang merupakan instans

DB sumber untuk replika baca lain. Untuk mengaktifkan pencadangan otomatis pada replika baca RDS for MySQL, pertama-tama buat replika baca, lalu ubah replika baca tersebut untuk mengaktifkan pencadangan otomatis.

> **Note**
>
> Dalam sebuah Wilayah AWS, kami sangat menyarankan agar Anda membuat semua replika baca di virtual private cloud (VPC) yang sama berdasarkan Amazon VPC sebagai instans DB sumber. Jika Anda membuat replika baca di VPC yang berbeda dari instans DB sumber, rentang Perutean Antar Domain Tanpa Kelas (CIDR) dapat tumpang-tindih antara replika dan sistem RDS. CIDR yang tumpang-tindih membuat replika tidak stabil, yang dapat berdampak negatif pada aplikasi yang terhubung dengannya. Jika Anda terjadi kesalahan saat membuat replika baca, pilih grup subnet DB tujuan yang berbeda. Untuk informasi selengkapnya, lihat Bekerja dengan klaster DB dalam VPC.
>
> Tidak ada cara langsung untuk membuat replika baca di tempat lain Akun AWS menggunakan konsol atau AWS CLI.

Konsol

Untuk membuat replika baca dari instans DB sumber

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data.
3. Pilih instans DB yang ingin Anda gunakan sebagai sumber untuk replika baca.
4. Untuk Tindakan, pilih Buat replika baca.
5. Untuk Pengidentifikasi instans DB, masukkan nama replika baca.
6. Pilih konfigurasi instans Anda. Kami menyarankan Anda menggunakan kelas dan jenis penyimpanan instans DB yang sama atau lebih besar sebagai instans DB sumber untuk replika baca.
7. Untuk Wilayah AWS, tentukan Wilayah tujuan untuk replika baca.
8. Untuk Penyimpanan, tentukan ukuran penyimpanan yang dialokasikan dan apakah Anda ingin menggunakan penskalaan otomatis penyimpanan.

   Jika instans DB sumber Anda tidak menggunakan konfigurasi penyimpanan terbaru, opsi Tingkatkan konfigurasi sistem file penyimpanan tersedia. Anda dapat mengaktifkan pengaturan

ini untuk meng-upgrade sistem file penyimpanan replika baca ke konfigurasi yang diinginkan. Untuk informasi selengkapnya, lihat the section called “Meningkatkan sistem file penyimpanan”.

9. Untuk Ketersediaan, pilih apakah akan membuat replika siaga dari replika Anda di Zona Ketersediaan lain guna menyediakan dukungan failover untuk replika tersebut.

> **Note**
>
> Pembuatan replika baca Anda sebagai instans DB Multi-AZ tidak tergantung pada apakah basis data sumber merupakan instans DB Multi-AZ.

10. Tentukan pengaturan instans DB lainnya. Untuk informasi tentang setiap pengaturan yang tersedia, lihat Pengaturan untuk instans DB.
11. Untuk membuat replika baca terenkripsi, perluas Konfigurasi tambahan dan tentukan pengaturan berikut:

    a. Pilih Aktifkan enkripsi.

    b. Untuk AWS KMS key, pilih AWS KMS key pengenal kunci KMS.

> **Note**
>
> Instans DB sumber harus dienkripsi. Untuk mempelajari selengkapnya tentang cara mengenkripsi instans DB sumber, lihat Mengenkripsi sumber daya Amazon RDS.

12. Pilih Buat replika baca.

Setelah replika baca dibuat, Anda dapat melihatnya di halaman Basis data di konsol RDS. Halaman tersebut menunjukkan Replika di kolom Peran.

AWS CLI

Untuk membuat replika baca dari instance DB sumber, gunakan AWS CLI perintah create-db-instance-read-replica. Contoh ini juga menetapkan ukuran penyimpanan yang dialokasikan, memungkinkan penskalaan otomatis penyimpanan, dan meng-upgrade sistem file ke konfigurasi yang diinginkan.

Anda dapat menentukan pengaturan lain. Untuk informasi tentang setiap pengaturan, lihat Pengaturan untuk instans DB.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-instance-read-replica \
    --db-instance-identifier myreadreplica \
    --source-db-instance-identifier mydbinstance \
    --allocated-storage 100 \
    --max-allocated-storage 1000 \
    --upgrade-storage-config
```

Untuk Windows:

```
aws rds create-db-instance-read-replica ^
    --db-instance-identifier myreadreplica ^
    --source-db-instance-identifier mydbinstance ^
    --allocated-storage 100 ^
    --max-allocated-storage 1000 ^
    --upgrade-storage-config
```

API RDS

Untuk membuat replika baca dari instans DB MySQL, MariaDB, Oracle, PostgreSQL, atau SQL Server sumber, panggil operasi API Amazon RDS `CreateDBInstanceReadReplica` dengan parameter wajib berikut:

- `DBInstanceIdentifier`
- `SourceDBInstanceIdentifier`

Mempromosikan replika baca menjadi instans DB mandiri

Anda dapat mempromosikan replika baca menjadi instans DB mandiri. Jika instans DB sumber memiliki beberapa replika baca, mempromosikan salah satu replika baca menjadi instans DB tidak akan memengaruhi replika lainnya.

Saat Anda mempromosikan replika baca, RDS me-reboot instans DB sebelum membuatnya tersedia. Proses promosi dapat memakan waktu beberapa menit atau lebih lama, tergantung dari ukuran replika baca.

Kasus penggunaan untuk mempromosikan replika baca

Anda mungkin ingin mempromosikan replika baca ke instans DB mandiri karena salah satu alasan berikut:

- Menerapkan pemulihan kegagalan – Anda dapat menggunakan promosi replika baca sebagai skema pemulihan data jika instans DB primer mengalami kegagalan. Pendekatan ini akan melengkapi replikasi sinkron, deteksi kegagalan otomatis, dan failover.

Jika Anda mengetahui konsekuensi dan batasan replikasi asinkron dan Anda masih ingin menggunakan promosi replika baca untuk pemulihan data, Anda dapat melakukannya. Untuk melakukannya, pertama-tama buat replika baca lalu pantau instans DB primer untuk mengetahui adanya kegagalan. Jika terjadi kegagalan, lakukan hal berikut:

1. Promosikan replika baca.
2. Arahkan lalu lintas basis data ke instans DB yang dipromosikan.
3. Buat replika baca pengganti dengan instans DB yang dipromosikan sebagai sumbernya.

- Meng-upgrade konfigurasi penyimpanan – Jika instans DB sumber Anda tidak menggunakan konfigurasi penyimpanan yang diinginkan, Anda dapat membuat replika baca instans dan meng-upgrade konfigurasi sistem file penyimpanan. Opsi ini memigrasikan sistem file replika baca ke konfigurasi yang diinginkan. Anda kemudian dapat mempromosikan replika baca menjadi instans mandiri.

  Anda dapat menggunakan opsi ini untuk mengatasi batasan penskalaan pada penyimpanan dan ukuran file untuk sistem file 32-bit yang lebih lama. Untuk informasi selengkapnya, lihat the section called “Meningkatkan sistem file penyimpanan”.

  Opsi ini hanya tersedia jika instans DB sumber Anda tidak menggunakan konfigurasi penyimpanan terbaru, atau jika Anda memodifikasi kelas instans DB dalam permintaan yang sama.
- Sharding – Sharding (pembuatan serpihan) mewujudkan arsitektur "share-nothing" dan pada dasarnya merupakan pemecahan basis data besar menjadi beberapa basis data yang lebih kecil. Salah satu cara umum untuk memisahkan basis data adalah memisahkan tabel yang tidak digabungkan dalam kueri yang sama ke host yang berbeda. Metode lain adalah menduplikasi tabel di beberapa host lalu menggunakan algoritma hashing untuk menentukan host mana yang menerima pembaruan tertentu. Anda dapat membuat replika baca yang sesuai dengan setiap serpihan (basis data yang lebih kecil) dan mempromosikannya saat Anda memutuskan untuk mengubahnya menjadi shard mandiri. Anda kemudian dapat mengambil ruang kunci (jika Anda memisahkan baris) atau distribusi tabel untuk setiap serpihan tergantung kebutuhan Anda.
- Melakukan operasi DDL (MySQL dan MariaDB saja) – Operasi DDL, seperti membuat atau membuat kembali indeks, dapat memakan waktu dan memberikan dampak performa yang signifikan pada instans DB Anda. Anda dapat melakukan operasi ini pada replika baca MySQL atau MariaDB setelah replika baca disinkronkan dengan instans DB primernya. Kemudian, Anda dapat mempromosikan replika baca dan mengarahkan aplikasi Anda untuk menggunakan Instans yang dipromosikan.

> **Note**
>
> Jika replika baca Anda adalah RDS untuk instans Oracle DB, Anda dapat melakukan peralihan alih-alih promosi. Dalam peralihan, instans DB sumber menjadi replika baru, dan replika menjadi instance DB sumber baru. Untuk informasi selengkapnya, lihat Melakukan switchover Oracle Data Guard.

Karakteristik replika baca yang dipromosikan

Setelah Anda mempromosikan replika baca, itu berhenti berfungsi sebagai replika baca dan menjadi instance DB mandiri. Instans DB mandiri baru memiliki karakteristik sebagai berikut:

- Instans DB mandiri mempertahankan grup opsi dan grup parameter dari replika baca pra-promosi.
- Anda dapat membuat replika baca dari instans DB mandiri dan melakukan operasi point-in-time pemulihan.
- Anda tidak dapat menggunakan instans DB sebagai target replikasi karena ini bukan lagi replika baca.

Prasyarat untuk mempromosikan replika baca

Sebelum Anda mempromosikan replika baca, lakukan hal berikut:

- Tinjau strategi pencadangan Anda:
  - Kami menyarankan Anda mengaktifkan cadangan dan menyelesaikan setidaknya satu cadangan. Durasi pencadangan adalah fungsi jumlah perubahan basis data sejak pencadangan sebelumnya.
  - Jika Anda telah mengaktifkan pencadangan pada replika baca, konfigurasikan periode pencadangan otomatis sehingga pencadangan harian tidak akan mengganggu promosi replika baca.
  - Pastikan replika baca Anda tidak memiliki `backing-up` status. Anda tidak dapat mempromosikan replika baca saat berada dalam keadaan ini.
- Hentikan transaksi apa pun agar tidak ditulis ke instans DB utama, lalu tunggu RDS menerapkan semua pembaruan ke replika baca.

Pembaruan basis data terjadi pada replika baca setelah pembaruan terjadi pada instans DB utama. Kelambatan replikasi dapat sangat bervariasi. Gunakan metrik `Replica Lag` untuk menentukan saat semua pembaruan sudah dilakukan pada replika baca.

- (Hanya MySQL dan MariaDB) Untuk membuat perubahan pada replika baca MySQL atau MariaDB sebelum Anda mempromosikannya, atur parameter ke dalam grup parameter DB untuk replika baca. `read_only 0` Anda kemudian dapat melakukan semua operasi DDL yang diperlukan, seperti membuat indeks pada replika baca. Tindakan yang dilakukan pada replika baca tidak memengaruhi performa instans DB primer.

Mempromosikan replika baca: langkah-langkah dasar

Langkah-langkah berikut ini menunjukkan proses umum untuk mempromosikan replika baca menjadi instans DB:

1. Promosikan replika baca dengan menggunakan opsi Promosikan di konsol Amazon RDS, AWS CLI perintah `promote-read-replica`, atau operasi `PromoteReadReplica`Amazon RDS API.

   > **Note**
   >
   > Proses promosi memakan waktu beberapa menit. Saat Anda mempromosikan replika baca, RDS menghentikan replikasi dan me-reboot replika baca. Saat boot ulang selesai, replika baca tersedia sebagai instans DB baru.

2. (Opsional) Ubah instans DB baru menjadi deployment Multi-AZ. Untuk informasi lebih lanjut, lihat Memodifikasi instans DB Amazon RDS dan Mengonfigurasi dan mengelola deployment Multi-AZ.

Konsol

Untuk mempromosikan replika baca menjadi instans DB mandiri

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di konsol Amazon RDS, pilih Database.

   Panel Database muncul. Setiap replika baca menampilkan Replika di kolom Peran.
3. Pilih replika baca yang ingin Anda promosikan.

4. Untuk Tindakan, pilih Promosikan.
5. Di halaman Promosikan Replika Baca, masukkan periode retensi cadangan dan periode pencadangan untuk instans DB yang baru dipromosikan.
6. Saat pengaturan sudah sesuai keinginan Anda, pilih Lanjutkan.
7. Di halaman konfirmasi, pilih Promosikan Replika Baca.

AWS CLI

Untuk mempromosikan replika baca ke instans DB mandiri, gunakan perintah. AWS CLI `promote-read-replica`

Example

Untuk Linux, macOS, atau Unix:

```
aws rds promote-read-replica \
    --db-instance-identifier myreadreplica
```

Untuk Windows:

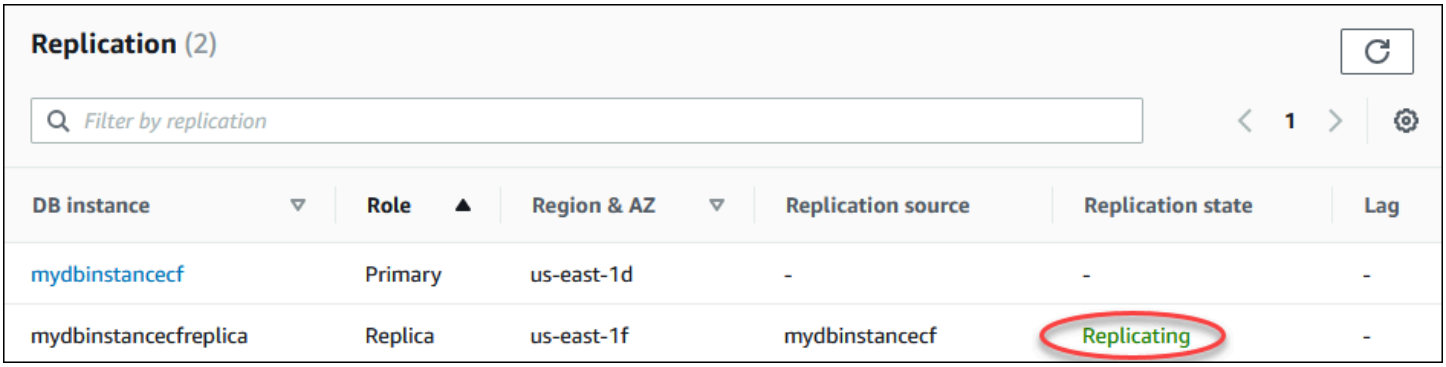
```
aws rds promote-read-replica ^
    --db-instance-identifier myreadreplica
```

API RDS

Untuk mempromosikan replika baca menjadi instans DB mandiri, panggil operasi API Amazon RDS `PromoteReadReplica` dengan parameter wajib `DBInstanceIdentifier`.

Memantau replikasi baca

Anda dapat memantau status replika baca dengan beberapa cara. Konsol Amazon RDS menampilkan status replika baca di bagian Replikasi pada tab Konektivitas & keamanan di detail replika baca. Untuk melihat detail replika baca, pilih nama replika baca di daftar instans di konsol Amazon RDS.



Anda juga dapat melihat status replika baca menggunakan AWS CLI `describe-db-instances` perintah atau operasi Amazon RDS API`DescribeDBInstances`.

Status replika baca dapat berupa salah satu dari berikut ini:

- mereplikasi – Replika baca berhasil direplikasi.
- replikasi terdegradasi (hanya SQL Server dan PostgreSQL) – Replika menerima data dari instans primer, tetapi satu atau beberapa basis data mungkin tidak mendapatkan pembaruan. Hal ini bisa terjadi, misalnya, ketika replika sedang dalam proses menyiapkan basis data yang baru dibuat. Hal ini juga dapat terjadi ketika DDL yang tidak didukung atau perubahan objek besar dibuat di lingkungan blue dalam deployment blue/green.

  Statusnya tidak beralih dari `replication degraded` ke `error`, kecuali jika terjadi kesalahan selama status terdegradasi.
- kesalahan – Telah terjadi kesalahan pada replikasi. Periksa kolom Kesalahan Replikasi di konsol Amazon RDS atau log peristiwa untuk menentukan kesalahan sebenarnya. Untuk informasi selengkapnya tentang pemecahan masalah kesalahan replikasi, lihat Pemecahan Masalah batasan replika baca MySQL.
- diakhiri (MariaDB, MySQL, atau PostgreSQL saja) – Replikasi diakhiri. Hal ini terjadi jika replikasi dihentikan selama lebih dari 30 hari berturut-turut, baik secara manual atau karena kesalahan replikasi. Jika ini terjadi, Amazon RDS menghentikan replikasi antara instans DB primer dan semua replika baca. Amazon RDS melakukannya untuk mencegah peningkatan kebutuhan penyimpanan pada instans DB sumber dan waktu failover yang lama.

  Replikasi yang rusak dapat memengaruhi penyimpanan karena ukuran dan jumlah log bertambah akibat tingginya volume pesan kesalahan yang ditulis ke log. Replikasi yang rusak juga dapat memengaruhi pemulihan kegagalan karena waktu yang diperlukan Amazon RDS untuk memelihara dan memproses log dalam jumlah besar selama pemulihan.

- diakhiri (Oracle saja) – Replikasi diakhiri. Hal ini terjadi jika replikasi dihentikan selama lebih dari 8 jam karena penyimpanan yang tersisa tidak memadai di replika baca. Jika ini terjadi, Amazon RDS menghentikan replikasi antara instans DB primer dan replika baca yang terpengaruh. Status ini adalah status akhir, dan replika baca harus dibuat ulang.
- dihentikan (MariaDB atau MySQL saja) – Replikasi telah dihentikan karena permintaan yang diajukan oleh pelanggan.
- titik henti replikasi ditetapkan (MySQL saja) – Titik henti yang diajukan oleh pelanggan ditetapkan menggunakan prosedur tersimpan mysql.rds_start_replication_until dan replikasi sedang berlangsung.
- titik henti replikasi tercapai (MySQL saja) – Titik henti yang diajukan oleh pelanggan ditetapkan menggunakan prosedur tersimpan mysql.rds_start_replication_until dan replikasi dihentikan karena titik henti tercapai.

Anda dapat melihat tempat instans DB sedang direplikasi dan jika demikian, memeriksa status replikasinya. Pada halaman Basis data di konsol RDS, Primer akan ditampilkan dalam kolom Peran. Pilih nama instans DB nya. Pada halaman detailnya, pada tab Konektivitas & keamanan, status replikasinya ada di bawah Replikasi.

Memantau lag replikasi

Anda dapat memantau kelambatan replikasi di Amazon CloudWatch dengan melihat metrik Amazon RDS. `ReplicaLag`

Untuk MariaDB dan MySQL, metrik `ReplicaLag` melaporkan nilai bidang `Seconds_Behind_Master` dari perintah `SHOW REPLICA STATUS`. Penyebab umum lag replikasi untuk MySQL dan MariaDB adalah sebagai berikut:

- Pemadaman jaringan.
- Menulis ke tabel dengan indeks pada replika baca. Jika parameter `read_only` tidak diatur ke 0 pada replika baca, hal ini dapat merusak replikasi.
- Menggunakan mesin penyimpanan non-transaksional seperti MyISAM. Replikasi hanya didukung untuk mesin penyimpanan InnoDB pada MySQL dan mesin penyimpanan XtraDB pada MariaDB.

> **Note**
>
> Versi sebelumnya dari MariaDB dan MySQL menggunakan `SHOW SLAVE STATUS`, bukan `SHOW REPLICA STATUS`. Jika Anda menggunakan versi MariaDB sebelum 10.5 atau versi MySQL sebelum 8.0.23, gunakan `SHOW SLAVE STATUS`.

Saat metrik `ReplicaLag` mencapai 0, replika telah menjadi instans DB primer. Jika metrik `ReplicaLag` menampilkan `-1`, maka replikasi saat ini tidak aktif. `ReplicaLag = -1` setara dengan `Seconds_Behind_Master = NULL`.

Untuk Oracle, metrik `ReplicaLag` adalah jumlah dari nilai `Apply Lag` dan perbedaan antara waktu saat ini dan nilai `DATUM_TIME` untuk apply lag. Nilai `DATUM_TIME` adalah terakhir kali replika baca menerima data dari instans DB sumbernya. Untuk informasi selengkapnya, lihat V$DATAGUARD_STATS dalam dokumentasi Oracle.

Untuk SQL Server, metrik `ReplicaLag` adalah lag maksimum dari basis data yang tertinggal, dalam hitungan detik. Misalnya, jika Anda memiliki dua basis data yang masing-masing tertinggal 5 detik dan 10 detik, maka `ReplicaLag` adalah 10 detik. Metrik `ReplicaLag` menampilkan nilai dari kueri berikut.

```
SELECT MAX(secondary_lag_seconds) max_lag FROM sys.dm_hadr_database_replica_states;
```

Untuk informasi selengkapnya, lihat secondary_lag_seconds dalam dokumentasi Microsoft.

`ReplicaLag` menampilkan `-1` jika RDS tidak dapat menentukan lag, seperti selama penyiapan replika, atau saat replika baca berada dalam status `error`.

> **Note**
>
> Basis data baru tidak disertakan dalam penghitungan lag sampai basis data tersebut dapat diakses di replika baca.

Untuk PostgreSQL, metrik `ReplicaLag` menampilkan nilai dari kueri berikut.

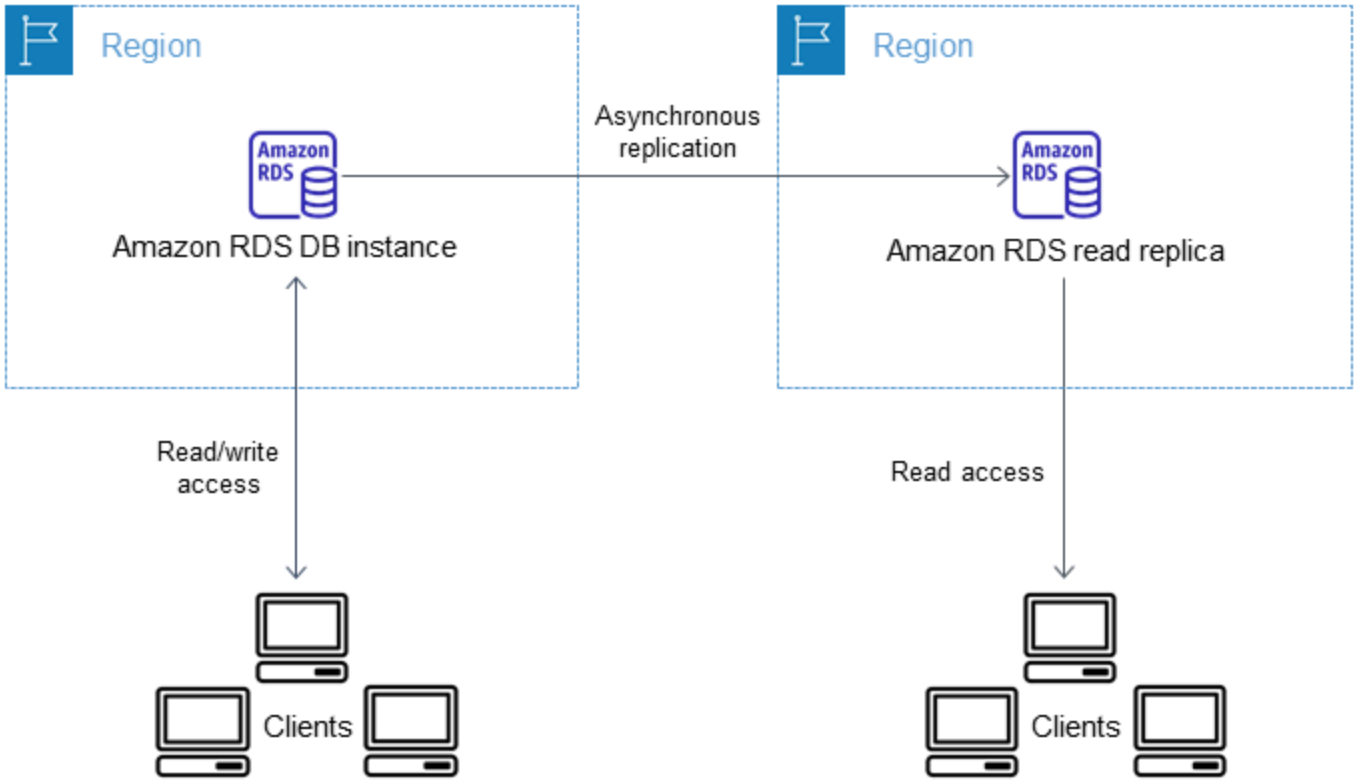
```
SELECT extract(epoch from now() - pg_last_xact_replay_timestamp()) AS reader_lag
```

PostgreSQL versi 9.5.2 dan yang lebih baru menggunakan slot replikasi fisik untuk mengelola retensi write ahead log (WAL) pada instans sumber. Untuk setiap instans replika baca lintas Wilayah, Amazon RDS membuat slot replikasi fisik dan mengaitkannya dengan instans. Dua CloudWatch metrik Amazon, `Oldest Replication Slot Lag` dan`Transaction Logs Disk Usage`, menunjukkan seberapa jauh di belakang replika yang paling tertinggal dalam hal data WAL yang diterima dan berapa banyak penyimpanan yang digunakan untuk data WAL. Nilai `Transaction Logs Disk Usage` dapat meningkat secara substansial ketika replika baca lintas Wilayah tertinggal secara signifikan.

Untuk informasi selengkapnya tentang memantau instans DB dengan CloudWatch, lihatMemantau metrik Amazon RDS dengan Amazon CloudWatch.

Membuat replika baca di tempat yang berbeda Wilayah AWS

Dengan Amazon RDS, Anda dapat membuat replika baca yang berbeda Wilayah AWS dari instans DB sumber.



Anda membuat replika baca di tempat yang berbeda Wilayah AWS untuk melakukan hal berikut:

- Mengoptimalkan kemampuan pemulihan bencana Anda.
- Skalakan operasi baca agar Wilayah AWS lebih dekat dengan pengguna Anda.

- Buat lebih mudah untuk bermigrasi dari pusat data di satu Wilayah AWS ke pusat data di pusat data lainnya Wilayah AWS.

Membuat replika baca di instance yang berbeda Wilayah AWS dari sumber mirip dengan membuat replika dalam hal yang sama. Wilayah AWS Anda dapat menggunakan AWS Management Console, menjalankan `create-db-instance-read-replica`perintah, atau memanggil operasi `CreateDBInstanceReadReplica`API.

> **Note**
>
> Untuk membuat replika baca terenkripsi yang berbeda Wilayah AWS dari instans DB sumber, instans DB sumber harus dienkripsi.

Ketersediaan Wilayah dan versi

Ketersediaan dan dukungan fitur bervariasi di seluruh versi spesifik dari setiap mesin basis data, dan di seluruh Wilayah AWS. Untuk informasi selengkapnya tentang versi dan ketersediaan Wilayah dengan replikasi lintas Wilayah, lihat Replika baca lintas Wilayah.

Membuat replika baca lintas Wilayah

Prosedur berikut menunjukkan cara membuat replika baca dari instans DB MariaDB, Microsoft SQL Server, MySQL, Oracle, atau PostgreSQL sumber di Wilayah AWS yang berbeda.

Konsol

Anda dapat membuat replika baca Wilayah AWS menggunakan. AWS Management Console

Untuk membuat replika baca Wilayah AWS dengan konsol

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data.
3. Pilih instans DB MariaDB, Microsoft SQL, MySQL, Oracle, atau PostgreSQL yang ingin Anda gunakan sebagai sumber replika baca.
4. Untuk Tindakan, pilih Buat replika baca.
5. Untuk Pengidentifikasi instans DB, masukkan nama replika baca.

6. Pilih Wilayah Tujuan.
7. Pilih spesifikasi instans yang ingin Anda gunakan. Kami menyarankan Anda menggunakan kelas dan jenis penyimpanan instans DB yang sama atau lebih besar untuk replika baca.
8. Untuk membuat replika baca terenkripsi di tempat lain: Wilayah AWS

   a. Pilih Aktifkan enkripsi.

   b. Untuk AWS KMS key, pilih AWS KMS key pengenal kunci KMS di tujuan. Wilayah AWS

> **Note**
>
> Untuk membuat replika baca terenkripsi, instans DB sumber harus dienkripsi. Untuk mempelajari selengkapnya tentang cara mengenkripsi instans DB sumber, lihat Mengenkripsi sumber daya Amazon RDS.

9. Pilih opsi lainnya, seperti penskalaan otomatis penyimpanan.
10. Pilih Buat replika baca.

AWS CLI

Untuk membuat replika baca dari instans DB MySQL, Microsoft SQL Server, MariaDB, Oracle, atau PostgreSQL sumber di Wilayah AWS yang berbeda, Anda dapat menggunakan perintah `create-db-instance-read-replica`. Dalam hal ini, Anda menggunakan `create-db-instance-read-replica`dari Wilayah AWS tempat yang Anda inginkan replika baca (Wilayah tujuan) dan tentukan Nama Sumber Daya Amazon (ARN) untuk instans DB sumber. ARN secara unik mengidentifikasi sumber daya yang dibuat di Amazon Web Services.

Misalnya, jika instans DB sumber Anda berada di Wilayah AS Timur (Virginia Utara), ARN terlihat seperti contoh ini:

```
arn:aws:rds:us-east-1:123456789012:db:mydbinstance
```

Untuk informasi tentang ARN, lihat Bekerja dengan Amazon Resource Name (ARN) di Amazon RDS.

Untuk membuat replika baca di instans DB sumber yang berbeda Wilayah AWS , Anda dapat menggunakan AWS CLI `create-db-instance-read-replica`perintah dari tujuan Wilayah AWS. Parameter berikut diperlukan untuk membuat replika baca di Wilayah AWS lain:

- `--region`— Tujuan Wilayah AWS di mana replika baca dibuat.
- `--source-db-instance-identifier` – Pengidentifikasi instans DB untuk instans DB sumber. Pengidentifikasi ini harus dalam format ARN untuk Wilayah AWS sumber.
- `--db-instance-identifier` – Pengidentifikasi replika baca di Wilayah AWS tujuan.

Example replika baca lintas Wilayah

Kode berikut membuat replika baca di Wilayah AS Barat (Oregon) dari instans DB sumber di Wilayah AS Timur (Virginia Utara).

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-instance-read-replica \
    --db-instance-identifier myreadreplica \
    --region us-west-2 \
    --source-db-instance-identifier arn:aws:rds:us-east-1:123456789012:db:mydbinstance
```

Untuk Windows:

```
aws rds create-db-instance-read-replica ^
    --db-instance-identifier myreadreplica ^
    --region us-west-2 ^
    --source-db-instance-identifier arn:aws:rds:us-east-1:123456789012:db:mydbinstance
```

Parameter berikut juga diperlukan untuk membuat replika baca terenkripsi di Wilayah AWS lain:

- `--kms-key-id`— AWS KMS key Pengidentifikasi kunci KMS yang digunakan untuk mengenkripsi replika baca di tujuan. Wilayah AWS

Example replika baca lintas Wilayah terenkripsi

Kode berikut membuat replika baca terenkripsi di Wilayah AS Barat (Oregon) dari instans DB sumber di Wilayah AS Timur (Virginia Utara).

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-instance-read-replica \
    --db-instance-identifier myreadreplica \
```

```
    --region us-west-2 \
    --source-db-instance-identifier arn:aws:rds:us-east-1:123456789012:db:mydbinstance
 \
    --kms-key-id my-us-west-2-key
```

Untuk Windows:

```
aws rds create-db-instance-read-replica ^
    --db-instance-identifier myreadreplica ^
    --region us-west-2 ^
    --source-db-instance-identifier arn:aws:rds:us-east-1:123456789012:db:mydbinstance
 ^
    --kms-key-id my-us-west-2-key
```

`--source-region`Opsi ini diperlukan saat Anda membuat replika baca terenkripsi antara Wilayah AWS GovCloud (AS-Timur) dan AWS GovCloud (AS-Barat). Untuk `--source-region`, tentukan Wilayah AWS instans DB sumber.

Jika `--source-region` tidak ditentukan, tentukan nilai `--pre-signed-url`. URL yang telah ditandatangani adalah URL yang berisi permintaan bertanda tangan Signature Versi 4 untuk perintah `create-db-instance-read-replica` yang dipanggil di Wilayah AWS sumber. Untuk mempelajari lebih lanjut tentang `pre-signed-url` opsi, lihat create-db-instance-read-replica di AWS CLI Command Reference.

API RDS

Untuk membuat replika baca dari sumber MySQL, Microsoft SQL Server, MariaDB, Oracle, atau PostgreSQL DB instans yang berbeda, Anda dapat memanggil operasi Amazon RDS API CreateDB. Wilayah AWS InstanceReadReplica Dalam kasus ini, Anda memanggil createDB InstanceReadReplica dari Wilayah AWS tempat yang Anda inginkan replika baca (Wilayah tujuan) dan tentukan Nama Sumber Daya Amazon (ARN) untuk instans DB sumber. ARN secara unik mengidentifikasi sumber daya yang dibuat di Amazon Web Services.

Untuk membuat replika baca terenkripsi di instans DB sumber yang Wilayah AWS berbeda, Anda dapat menggunakan `CreateDBInstanceReadReplica`operasi Amazon RDS API dari tujuan. Wilayah AWS Untuk membuat replika baca terenkripsi di tempat lain Wilayah AWS, Anda harus menentukan nilai untuk. `PreSignedURL` `PreSignedURL`harus berisi permintaan `CreateDBInstanceReadReplica`operasi untuk memanggil sumber Wilayah

AWS tempat replika baca dibuat. Untuk mempelajari lebih lanjut tentang `PreSignedUrl`, lihat `CreateDBInstanceReadReplica`.

Misalnya, jika instans DB sumber Anda berada di Wilayah AS Timur (Virginia Utara), ARN terlihat seperti yang berikut ini.

```
arn:aws:rds:us-east-1:123456789012:db:mydbinstance
```

Untuk informasi tentang ARN, lihat Bekerja dengan Amazon Resource Name (ARN) di Amazon RDS.

Example

```
https://us-west-2.rds.amazonaws.com/
    ?Action=CreateDBInstanceReadReplica
    &KmsKeyId=my-us-east-1-key
    &PreSignedUrl=https%253A%252F%252Frds.us-west-2.amazonaws.com%252F
         %253FAction%253DCreateDBInstanceReadReplica
         %2526DestinationRegion%253Dus-east-1
         %2526KmsKeyId%253Dmy-us-east-1-key
         %2526SourceDBInstanceIdentifier%253Darn%25253Aaws%25253Ards%25253Aus-
west-2%123456789012%25253Adb%25253Amydbinstance
         %2526SignatureMethod%253DHmacSHA256
         %2526SignatureVersion%253D4%2526SourceDBInstanceIdentifier%253Darn%25253Aaws
%25253Ards%25253Aus-west-2%25253A123456789012%25253Ainstance%25253Amydbinstance
         %2526Version%253D2014-10-31
         %2526X-Amz-Algorithm%253DAWS4-HMAC-SHA256
         %2526X-Amz-Credential%253DAKIADQKE4SARGYLE%252F20161117%252Fus-west-2%252Frds
%252Faws4_request
         %2526X-Amz-Date%253D20161117T215409Z
         %2526X-Amz-Expires%253D3600
         %2526X-Amz-SignedHeaders%253Dcontent-type%253Bhost%253Buser-agent%253Bx-amz-
content-sha256%253Bx-amz-date
         %2526X-Amz-Signature
%253D255a0f17b4e717d3b67fad163c3ec26573b882c03a65523522cf890a67fca613
    &DBInstanceIdentifier=myreadreplica
    &SourceDBInstanceIdentifier=&region-arn;rds:us-east-1:123456789012:db:mydbinstance
    &Version=2012-01-15
    &SignatureVersion=2
    &SignatureMethod=HmacSHA256
    &Timestamp=2012-01-20T22%3A06%3A23.624Z
    &AWSAccessKeyId=<&AWS; Access Key ID>
    &Signature=<Signature>
```

Cara Amazon RDS melakukan replikasi lintas Wilayah

Amazon RDS menggunakan proses berikut untuk membuat replika baca lintas Wilayah. Bergantung pada yang Wilayah AWS terlibat dan jumlah data dalam database, proses ini dapat memakan waktu berjam-jam untuk diselesaikan. Anda dapat menggunakan informasi ini untuk menentukan sejauh mana proses telah berjalan saat Anda membuat replika baca lintas Wilayah:

1. Amazon RDS mulai mengonfigurasi instans DB sumber sebagai sumber replikasi dan mengatur statusnya menjadi memodifikasi.
2. Amazon RDS mulai menyiapkan replika baca yang ditentukan di tujuan Wilayah AWS dan menetapkan status untuk dibuat.
3. Amazon RDS membuat snapshot DB otomatis dari instans DB sumber di Wilayah AWS sumber. Format nama snapshot DB adalah `rds:<InstanceID>-<timestamp>`, dengan `<InstanceID>` adalah pengidentifikasi instans sumber, dan `<timestamp>` adalah tanggal dan waktu penyalinan dimulai. Misalnya, `rds:mysourceinstance-2013-11-14-09-24` dibuat dari instans `mysourceinstance` pada `2013-11-14-09-24`. Selama pembuatan snapshot DB otomatis, status instans DB sumber tetap memodifikasi, status replika baca tetap membuat, dan status snapshot DB adalah membuat. Kolom progres pada halaman snapshot DB di konsol melaporkan seberapa jauh pembuatan snapshot DB telah berlangsung. Ketika snapshot DB selesai, status snapshot DB dan instans DB sumber diatur ke tersedia.
4. Amazon RDS memulai penyalinan snapshot lintas Wilayah untuk transfer data awal. Salinan snapshot terdaftar sebagai snapshot otomatis di tujuan Wilayah AWS dengan status pembuatan. Salinan ini memiliki nama yang sama dengan snapshot DB sumber. Kolom progres tampilan snapshot DB menunjukkan sejauh mana progres penyalinan. Ketika penyalinan selesai, status salinan snapshot DB diatur ke tersedia.
5. Amazon RDS kemudian menggunakan snapshot DB yang disalin untuk pemuatan data awal pada replika baca. Selama fase ini, replika baca ada dalam daftar instans DB di Wilayah tujuan, dengan status membuat. Ketika pemuatan selesai, status replika baca diatur ke tersedia, dan salinan snapshot DB dihapus.
6. Saat replika baca mencapai status tersedia, Amazon RDS memulai dengan mereplikasi perubahan yang dibuat ke instans sumber sejak dimulainya operasi buat replika baca. Selama fase ini, waktu lag replikasi untuk replika baca lebih besar dari 0.

   Untuk informasi tentang waktu lag replikasi, lihat Memantau replikasi baca.

Pertimbangan replikasi lintas Wilayah

Semua pertimbangan untuk melakukan replikasi dalam Wilayah AWS berlaku untuk replikasi lintas wilayah. Pertimbangan tambahan berikut berlaku saat mereplikasi di antara Wilayah AWS:

- Instans DB sumber dapat memiliki replika baca lintas Wilayah di beberapa Wilayah AWS.
- Anda dapat mereplikasi antara Wilayah GovCloud (AS-Timur) dan GovCloud (AS-Barat), tetapi tidak masuk atau keluar dari GovCloud (AS).
- Untuk instans DB Microsoft SQL Server, Oracle, dan PostgreSQL, Anda hanya dapat membuat replika baca Amazon RDS lintas Wilayah dari instans Amazon RDS sumber yang bukan replika baca dari instans DB Amazon RDS lainnya. Batasan ini tidak berlaku untuk instans DB MariaDB dan MySQL.
- Anda dapat mengharapkan untuk melihat tingkat jeda waktu yang lebih tinggi untuk replika baca apa pun yang berbeda Wilayah AWS dari contoh sumber. Waktu lag ini berasal dari saluran jaringan yang lebih panjang antar-pusat data regional.
- Untuk replika baca lintas Wilayah, setiap perintah buat replika baca yang menentukan parameter `--db-subnet-group-name` harus menentukan grup subnet DB dari VPC yang sama.
- Karena batas jumlah entri daftar kontrol akses (ACL) untuk VPC sumber, kami tidak dapat menjamin lebih dari lima instans replika baca lintas Wilayah.
- Dalam kebanyakan kasus, replika baca menggunakan grup parameter DB dan grup opsi DB default untuk mesin DB yang ditentukan.

  Untuk mesin MySQL dan Oracle DB, Anda dapat menentukan grup parameter khusus untuk replika baca dalam `--db-parameter-group-name` opsi perintah. AWS CLI `create-db-instance-read-replica` Anda tidak dapat menentukan grup parameter kustom saat menggunakan AWS Management Console.
- Replika baca menggunakan grup keamanan default.
- Untuk instans DB MariaDB, Microsoft SQL Server, MySQL, dan Oracle, ketika instans DB sumber untuk replika baca lintas Wilayah dihapus, replika baca tersebut akan dipromosikan.
- Untuk instans DB PostgreSQL, ketika instans DB sumber untuk replika baca lintas Wilayah dihapus, status replikasi replika baca diatur ke `terminated`. Replika baca tidak dipromosikan.

  Anda harus mempromosikan replika baca secara manual atau menghapusnya.

Meminta replika baca lintas Wilayah

Untuk berkomunikasi dengan Wilayah sumber untuk meminta pembuatan replika baca lintas Wilayah, pemohon (peran IAM atau pengguna IAM) harus memiliki akses ke instans DB sumber dan Wilayah sumber.

Kondisi tertentu dalam kebijakan IAM pemohon dapat menyebabkan permintaan ini gagal. Contoh berikut berasumsi bahwa instans DB sumber berada di AS Timur (Ohio) dan replika baca dibuat di AS Timur (Virginia Utara). Contoh ini menunjukkan kondisi dalam kebijakan IAM pemohon yang menyebabkan permintaan gagal:

- Kebijakan milik pemohon memiliki kondisi untuk `aws:RequestedRegion`.

```
...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": "*",
"Condition": {
    "StringEquals": {
        "aws:RequestedRegion": "us-east-1"
    }
}
```

  Permintaan gagal karena kebijakan tidak mengizinkan akses ke Wilayah sumber. Agar permintaan berhasil, tentukan Wilayah sumber dan tujuan.

```
...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": "*",
"Condition": {
    "StringEquals": {
        "aws:RequestedRegion": [
            "us-east-1",
            "us-east-2"
        ]
    }
}
```

- Kebijakan milik pemohon tidak mengizinkan akses ke instans DB sumber.

```
...
```

```
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": "arn:aws:rds:us-east-1:123456789012:db:myreadreplica"
...
```

Agar permintaan berhasil, tentukan instans sumber dan replikanya.

```
...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": [
    "arn:aws:rds:us-east-1:123456789012:db:myreadreplica",
    "arn:aws:rds:us-east-2:123456789012:db:mydbinstance"
]
...
```

- Kebijakan milik pemohon menolak `aws:ViaAWSService`.

```
...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": "*",
"Condition": {
    "Bool": {"aws:ViaAWSService": "false"}
}
```

Komunikasi dengan Wilayah sumber dibuat oleh RDS atas nama pemohon. Untuk permintaan yang berhasil, jangan menolak panggilan yang dilakukan oleh AWS layanan.

- Kebijakan milik pemohon memiliki kondisi untuk `aws:SourceVpc` atau `aws:SourceVpce`.

Permintaan ini mungkin gagal karena ketika RDS membuat panggilan ke Wilayah jarak jauh, panggilan tersebut bukan dari VPC atau titik akhir VPC yang ditentukan.

Jika Anda perlu menggunakan salah satu kondisi sebelumnya yang akan menyebabkan permintaan gagal, Anda dapat menyertakan pernyataan kedua dengan `aws:CalledVia` dalam kebijakan Anda untuk membuat permintaan berhasil. Misalnya, Anda dapat menggunakan `aws:CalledVia` dengan `aws:SourceVpce` seperti yang ditunjukkan di sini:

```
...
"Effect": "Allow",
```

```
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": "*",
"Condition": {
    "Condition" : {
        "ForAnyValue:StringEquals" : {
          "aws:SourceVpce": "vpce-1a2b3c4d"
        }
     }
},
{
    "Effect": "Allow",
    "Action": [
        "rds:CreateDBInstanceReadReplica"
    ],
    "Resource": "*",
    "Condition": {
        "ForAnyValue:StringEquals": {
            "aws:CalledVia": [
                "rds.amazonaws.com"
            ]
        }
    }
}
```

Untuk informasi selengkapnya, lihat Kebijakan dan izin di IAM dalam Panduan Pengguna IAM.

Mengotorisasi replika baca

Setelah permintaan pembuatan replika baca DB lintas Wilayah menampilkan `success`, RDS memulai pembuatan replika di latar belakang. Otorisasi agar RDS dapat mengakses instans DB sumber telah dibuat. Otorisasi ini menghubungkan instans DB sumber ke replika baca, dan mengizinkan RDS untuk menyalin hanya ke replika baca yang ditentukan.

Otorisasi ini diverifikasi oleh RDS menggunakan izin `rds:CrossRegionCommunication` dalam peran IAM terkait layanan. Jika replika sudah diotorisasi, RDS berkomunikasi dengan Wilayah sumber dan menyelesaikan pembuatan replika.

RDS tidak memiliki akses ke instans DB yang tidak diotorisasi sebelumnya oleh permintaan `CreateDBInstanceReadReplica`. Otorisasi dicabut saat pembuatan replika baca selesai.

RDS menggunakan peran terkait layanan untuk memverifikasi otorisasi di Wilayah sumber. Jika Anda menghapus peran terkait layanan selama proses pembuatan replika, maka pembuatan tersebut gagal.

Untuk informasi selengkapnya, lihat Menggunakan peran terkait layanan dalam Panduan Pengguna IAM.

Menggunakan AWS Security Token Service kredensyal

Token sesi dari titik akhir global AWS Security Token Service (AWS STS) hanya valid di Wilayah AWS yang diaktifkan secara default (Wilayah komersial). Jika Anda menggunakan kredensil dari operasi `assumeRole` API di AWS STS, gunakan titik akhir regional jika Wilayah sumber adalah Region keikutsertaan. Jika tidak, permintaan akan gagal. Hal ini terjadi karena kredensil Anda harus valid di kedua Wilayah, yang berlaku untuk Wilayah keikutsertaan hanya jika titik AWS STS akhir regional digunakan.

Untuk menggunakan titik akhir global, pastikan bahwa titik akhir tersebut diaktifkan untuk kedua Wilayah dalam operasi. Setel titik akhir global ke `Valid in all Wilayah AWS` dalam pengaturan AWS STS akun.

Aturan yang sama berlaku untuk kredensial dalam parameter URL yang telah ditandatangani sebelumnya.

Untuk informasi selengkapnya, lihat Mengelola AWS STSWilayah AWS dalam Panduan Pengguna IAM.

Biaya replikasi lintas Wilayah

Data yang ditransfer untuk replikasi lintas Wilayah akan menimbulkan biaya transfer data Amazon RDS. Tindakan replikasi lintas Wilayah ini menimbulkan biaya untuk data yang ditransfer keluar dari Wilayah AWS sumber:

- Saat Anda membuat replika baca, Amazon RDS mengambil snapshot dari instans sumber dan mentransfer snapshot tersebut ke Wilayah AWS replika baca.
- Untuk setiap modifikasi data yang dibuat dalam database sumber, Amazon RDS mentransfer data dari sumber Wilayah AWS ke replika baca. Wilayah AWS

Untuk informasi selengkapnya tentang biaya transfer data, lihat Harga Amazon RDS.

Untuk instans MySQL dan MariaDB, Anda dapat mengurangi biaya transfer data dengan mengurangi jumlah replika baca lintas Wilayah yang Anda buat. Misalnya, misalkan Anda memiliki instance DB sumber di satu Wilayah AWS dan ingin memiliki tiga replika baca di yang lain Wilayah AWS. Dalam kasus ini, Anda perlu membuat hanya satu replika baca dari instans DB sumber. Anda membuat dua replika lainnya dari replika baca pertama, bukan instans DB sumber.

Misalnya, jika Anda memilikinya `source-instance-1` Wilayah AWS, Anda dapat melakukan hal berikut:

- Buat `read-replica-1` yang baru Wilayah AWS, tentukan `source-instance-1` sebagai sumber.
- Buat `read-replica-2` dari `read-replica-1`.
- Buat `read-replica-3` dari `read-replica-1`.

Dalam contoh ini, Anda hanya dikenai biaya untuk data yang ditransfer dari `source-instance-1` ke `read-replica-1`. Anda tidak dikenai biaya untuk data yang ditransfer dari `read-replica-1` ke dua replika lainnya karena semuanya berada di Wilayah AWS yang sama. Jika Anda membuat ketiga replika langsung dari `source-instance-1`, Anda akan dikenai biaya transfer data ke ketiga replika tersebut.

Memberi tag pada sumber daya Amazon RDS

Anda dapat menggunakan tag Amazon RDS untuk menambahkan metadata ke sumber daya Amazon RDS Anda. Anda dapat menggunakan tag untuk menambahkan notasi Anda sendiri tentang instans basis data, snapshot, klaster Aurora, dan sebagainya. Tindakan ini dapat membantu Anda mendokumentasikan sumber daya Amazon RDS Anda. Anda juga dapat menggunakan tag dengan prosedur pemeliharaan otomatis.

Khususnya, Anda dapat menggunakan tag ini dengan kebijakan IAM. Anda dapat menggunakannya untuk mengelola akses ke sumber daya RDS dan mengontrol tindakan yang dapat diterapkan pada sumber daya RDS. Anda dapat menggunakan tag ini untuk melacak biaya dengan mengelompokkan pengeluaran untuk sumber daya serupa yang diberi tag.

Anda dapat memberi tag pada sumber daya Amazon RDS berikut:

- Instans DB
- Klaster DB
- Titik akhir cluster DB
- Replika baca
- Snapshot DB
- Snapshot klaster DB
- Instans DB terpesan
- Langganan peristiwa
- Grup opsi DB
- Grup parameter DB
- Grup parameter klaster DB
- Grup subnet DB
- Proksi RDS
- Titik akhir Proksi RDS
- Deployment blue/green
- Integrasi nol-ETL (pratinjau)

> **Note**
>
> Saat ini, Anda tidak dapat menandai RDS Proxies dan RDS Proxy endpoint dengan menggunakan. AWS Management Console

Topik

Gambaran umum tag sumber daya Amazon RDS

Tag Amazon RDS adalah pasangan nama-nilai yang Anda tentukan dan tautkan dengan sumber daya Amazon RDS. Nama ini disebut sebagai kunci. Memberikan nilai untuk kunci bersifat opsional. Anda dapat menggunakan tag untuk memberikan informasi tambahan ke sumber daya Amazon RDS. Anda dapat menggunakan kunci tag, misalnya, untuk menentukan kategori, dan nilai tag mungkin merupakan item dalam kategori tersebut. Misalnya, Anda dapat menentukan kunci tag “project” dan nilai tag “Salix”. Hal ini menunjukkan bahwa sumber daya Amazon RDS ditetapkan ke proyek Salix. Anda juga dapat menggunakan tag untuk menunjukkan bahwa sumber daya Amazon RDS sedang digunakan untuk pengujian atau produksi dengan menggunakan kunci seperti `environment=test` atau `environment=production`. Sebaiknya Anda menggunakan kumpulan kunci tag yang konsisten guna mempermudah palacakan metadata yang terkait dengan sumber daya Amazon RDS.

Selain itu, Anda dapat menggunakan kondisi dalam kebijakan IAM Anda untuk mengontrol akses ke AWS sumber daya berdasarkan tag pada sumber daya tersebut. Anda dapat melakukan ini dengan menggunakan kunci kondisi `aws:ResourceTag/`*`tag-key`* global. Untuk informasi selengkapnya, lihat Mengontrol akses ke AWS sumber daya di Panduan Pengguna AWS Identity and Access Management.

Setiap sumber daya Amazon RDS memiliki serangkaian tag, yang berisi semua tag yang ditetapkan ke sumber daya Amazon RDS tersebut. Rangkaian tag dapat berisi 50 tag atau kosong. Jika Anda

menambahkan tag ke sumber daya RDS dengan kunci yang sama dengan tag sumber daya yang ada, nilai yang baru akan menimpa yang lama.

AWS tidak menerapkan makna semantik apa pun pada tag Anda; tag ditafsirkan secara ketat sebagai string karakter. RDS dapat mengatur tag pada instans DB atau sumber daya RDS lainnya. Pengaturan tag bergantung pada opsi yang Anda gunakan saat membuat sumber daya. Misalnya, Amazon RDS dapat menambahkan tag yang menunjukkan bahwa instans DB digunakan untuk produksi atau pengujian.

- Kunci tag adalah nama wajib tag. Nilai string dapat terdiri dari 1 hingga 128 karakter Unicode dan tidak boleh diawali dengan `aws:` atau `rds:`. String hanya dapat berisi kumpulan huruf Unicode, angka, spasi, '_', '.', ':', '/', '=', '+', '-', '@' (regex Java: "^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-@]*)$").

- Nilai tag adalah nilai string opsional dari tag. Nilai string dapat terdiri dari 1 hingga 256 karakter Unicode. String hanya dapat berisi kumpulan huruf Unicode, angka, spasi, '_', '.', ':', '/', '=', '+', '-', '@' (regex Java: "^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-@]*)$").

  Nilai rangkaian tag tidak harus unik dan bisa nol. Misalnya, Anda dapat menggunakan pasangan kunci-nilai dalam satu rangkaian tag `project=Trinity` dan `cost-center=Trinity`.

Anda dapat menggunakan, API AWS Management Console AWS CLI, atau Amazon RDS untuk menambahkan, membuat daftar, dan menghapus tag di sumber daya Amazon RDS. Saat menggunakan CLI atau API, pastikan untuk menyediakan Amazon Resource Name (ARN) milik sumber daya RDS yang akan ditangani. Untuk informasi selengkapnya tentang cara menyusun ARN, lihat Membuat konsep ARN untuk Amazon RDS.

Tag disimpan di cache untuk diotorisasi. Oleh karena itu, penambahan dan pembaruan tag di sumber daya Amazon RDS dapat memakan waktu beberapa menit sebelum tersedia.

Menggunakan tag untuk kontrol akses dengan IAM

Anda dapat menggunakan tag dengan kebijakan IAM untuk mengelola akses ke sumber daya Amazon RDS. Anda juga dapat menggunakan tag untuk mengontrol tindakan yang dapat diterapkan ke sumber daya Amazon RDS.

Untuk informasi tentang pengelolaan akses ke sumber daya yang diberi tag dengan kebijakan IAM, lihat Manajemen identitas dan akses untuk Amazon RDS.

Menggunakan tag untuk menghasilkan laporan penagihan mendetail

Anda dapat menggunakan tag ini untuk melacak biaya dengan mengelompokkan pengeluaran untuk sumber daya serupa yang diberi tag.

Gunakan tag untuk mengatur AWS tagihan Anda untuk mencerminkan struktur biaya Anda sendiri. Untuk melakukan ini, daftar untuk mendapatkan Akun AWS tagihan Anda dengan nilai kunci tag disertakan. Kemudian, untuk melihat biaya sumber daya gabungan, atur informasi penagihan Anda sesuai dengan sumber daya Anda dengan nilai kunci tag yang sama. Misalnya, Anda dapat memberi tag beberapa sumber daya dengan nama aplikasi tertentu, kemudian susun informasi penagihan Anda untuk melihat biaya total aplikasi tersebut pada beberapa layanan. Untuk informasi selengkapnya, lihat Menggunakan Tag Alokasi Biaya dalam Panduan Pengguna AWS Billing .

> **Note**
>
> Anda dapat menambahkan tag ke snapshot DB; namun, tagihan Anda tidak akan mencerminkan pengelompokan ini.
> Agar tag alokasi biaya diterapkan ke snapshot DB, tag harus dilampirkan ke instans DB induk, dan instance induk harus ada Wilayah AWS sama dengan snapshot. Biaya untuk snapshot yatim piatu digabungkan dalam satu item yang tidak ditandai.

Menambahkan, menampilkan daftar, dan menghapus tag

Prosedur berikut menunjukkan cara melakukan operasi pemberian tag standar pada sumber daya yang terkait dengan instans DB.

Konsol

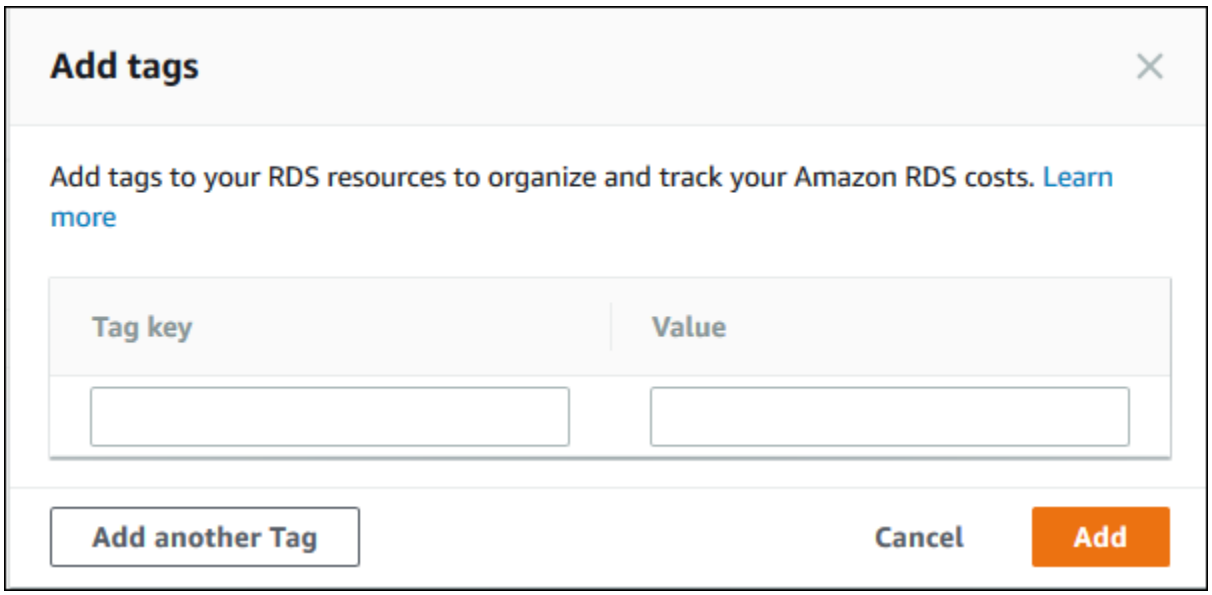
Proses pemberian tag pada sumber daya Amazon RDS untuk semua sumber daya dilakukan secara sama. Prosedur berikut menunjukkan cara memberi tag pada instans DB Amazon RDS.

Untuk menambahkan tag ke instans DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data.

> **Note**
>
> Untuk memfilter daftar instans DB dalam panel Basis data, masukkan string teks untuk Filter basis data. Hanya instans DB yang berisi string yang muncul.

3. Pilih nama instans DB yang ingin Anda beri tag untuk menampilkan detailnya.
4. Di bagian detail, gulir ke bawah hingga bagian Tag.
5. Pilih Tambahkan. Jendela Tambahkan tag akan muncul.



6. Masukkan nilai untuk Kunci tag dan nilai.
7. Untuk menambahkan tag lain, Anda dapat memilih Tambahkan Tag lain dan memasukkan nilai untuk Kunci tag dan Nilai-nya.

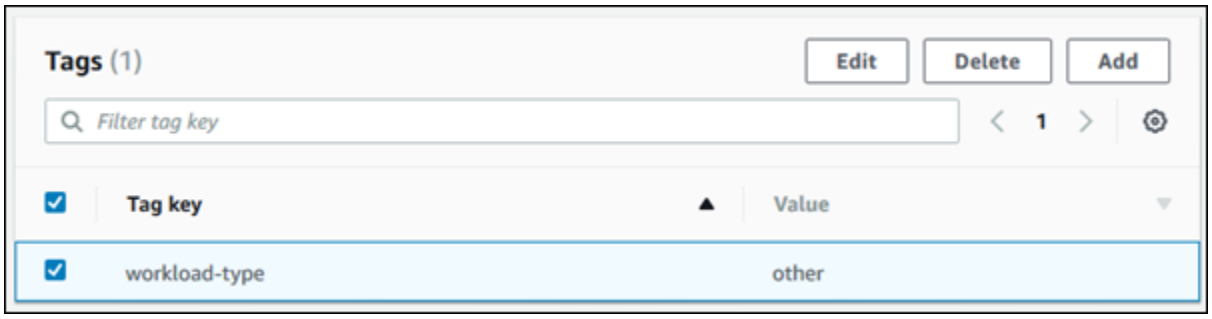
   Ulangi langkah ini seperlunya.
8. Pilih Tambahkan.

Untuk menghapus tag dari instans DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data.

> **Note**
>
> Untuk memfilter daftar instans DB dalam panel Basis data, masukkan string teks dalam kotak Filter basis data. Hanya instans DB yang berisi string yang muncul.

3. Pilih nama instans DB untuk menampilkan detailnya.
4. Di bagian detail, gulir ke bawah hingga bagian Tag.
5. Pilih tag yang ingin Anda hapus.



6. Pilih Hapus, lalu pilih Hapus di jendela Hapus tag.

AWS CLI

Anda dapat menambahkan, menampilkan daftar, atau menghapus tag untuk instans DB menggunakan AWS CLI.

- Untuk menambahkan satu atau beberapa tag ke sumber daya Amazon RDS, gunakan AWS CLI perintah `add-tags-to-resource`.
- Untuk membuat daftar tag pada sumber daya Amazon RDS, gunakan AWS CLI perintah `list-tags-for-resource`.
- Untuk menghapus satu atau beberapa tag dari sumber daya Amazon RDS, gunakan AWS CLI perintah `remove-tags-from-resource`.

Untuk mempelajari lebih lanjut tentang cara membuat ARN yang diperlukan, lihat Membuat konsep ARN untuk Amazon RDS.

API RDS

Anda dapat menambahkan, menampilkan daftar, atau menghapus tag untuk instans DB menggunakan API Amazon RDS.

- Untuk menambahkan tag ke sumber daya Amazon RDS, gunakan operasi `AddTagsToResource`.
- Untuk menampilkan daftar tag yang ditetapkan ke sumber daya Amazon RDS, gunakan `ListTagsForResource`.
- Untuk menghapus tag dari sumber daya Amazon RDS, gunakan operasi `RemoveTagsFromResource`.

Untuk mempelajari selengkapnya tentang cara menyusun ARN yang diperlukan, lihat Membuat konsep ARN untuk Amazon RDS.

Ketika menggunakan XML dengan API Amazon RDS, tag menggunakan skema berikut:

```
<Tagging>
    <TagSet>
        <Tag>
            <Key>Project</Key>
            <Value>Trinity</Value>
        </Tag>
        <Tag>
            <Key>User</Key>
            <Value>Jones</Value>
        </Tag>
    </TagSet>
</Tagging>
```

Tabel berikut menyediakan daftar tag XML yang diizinkan beserta karakteristiknya. Nilai untuk Kunci dan Nilai bersifat peka huruf besar/kecil. Misalnya, project=Trinity dan PROJECT=Trinity adalah dua tag yang berbeda.

Elemen tag	Deskripsi
TagSet	Rangkaian tag adalah wadah untuk semua tag yang ditetapkan ke sumber daya Amazon RDS. Hanya ada satu rangkaian tag per sumber daya. Anda bekerja dengan TagSet hanya melalui Amazon RDS API.
Tag	Tag adalah pasangan kunci-nilai yang ditentukan pengguna. Satu rangkaian tag bisa berisi 1 hingga 50 tag.
Kunci	Kunci adalah nama wajib tag. Nilai string dapat terdiri dari 1 hingga 128 karakter Unicode dan tidak boleh diawali dengan `aws:` atau `rds:`. String

Elemen tag	Deskripsi
	hanya dapat berisi kumpulan huruf Unicode, angka, spasi, '_', '.', '/', '=', '+', '-' (regex Java: "^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-]*)$"). Kunci dalam rangkaian tag harus unik. Misalnya, Anda tidak dapat memiliki pasangan kunci dalam satu rangkaian tag dengan kunci yang sama tetapi dengan nilai yang berbeda, seperti project/Trinity dan project/Xanadu.
Nilai	Nilai adalah nilai opsional tag. Nilai string dapat terdiri dari 1 hingga 256 karakter Unicode dan tidak boleh diawali dengan `aws:` atau `rds:`. String hanya dapat berisi kumpulan huruf Unicode, angka, spasi, '_', '.', '/', '=', '+', '-' (regex Java: "^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-]*)$"). Nilai rangkaian tag tidak harus unik dan bisa nol. Misalnya, Anda dapat menggunakan pasangan kunci-nilai dalam satu rangkaian tag project/Trinity dan cost-center/Trinity.

Menggunakan Editor AWS Tag

Anda dapat menelusuri dan mengedit tag pada sumber daya RDS Anda AWS Management Console dengan menggunakan editor AWS Tag. Untuk informasi selengkapnya, lihat Editor Tag dalam Panduan Pengguna AWS .

Menyalin tag ke snapshot instans DB

Saat membuat atau memulihkan instans DB, Anda dapat menentukan bahwa tag dari instans DB disalin ke snapshot instans DB. Penyalinan tag memastikan bahwa metadata untuk snapshot DB cocok dengan metadata instans DB sumber. Ini juga memastikan bahwa kebijakan akses apa pun untuk snapshot DB juga cocok dengan kebijakan akses instans DB sumber.

Anda dapat menentukan bahwa tag disalin ke snapshot DB untuk tindakan berikut:

- Membuat instans DB.
- Memulihkan instans DB.
- Membuat replika baca.
- Menyalin snapshot DB.

Dalam kebanyakan kasus, tag tidak disalin secara default. Namun, saat Anda memulihkan instans DB dari snapshot DB, RDS memeriksa apakah Anda menentukan tag baru. Jika ya, tag baru akan ditambahkan ke instans DB yang dipulihkan. Jika tidak ada tag baru, RDS akan menambahkan tag dari instans DB sumber pada saat pembuatan snapshot ke instans DB yang dipulihkan.

Untuk mencegah tag dari instans DB sumber ditambahkan ke instans DB yang dipulihkan, sebaiknya Anda menentukan tag baru saat memulihkan instans DB.

> **Note**
>
> Dalam beberapa kasus, Anda mungkin menyertakan nilai untuk `--tags` parameter create-db-snapshot AWS CLI perintah. Atau Anda mungkin perlu memasukkan setidaknya satu tag ke operasi CreateDBSnapshot API. Dalam kasus ini, RDS tidak menyalin tag dari instans DB sumber ke snapshot DB baru. Fungsi ini berlaku meskipun opsi `--copy-tags-to-snapshot` (`CopyTagsToSnapshot`) di instans DB sumber diaktifkan.
> Jika menggunakan pendekatan ini, Anda dapat membuat salinan instans DB dari snapshot DB. Pendekatan ini menghindari penambahan tag yang tidak berlaku untuk instans DB baru. Anda membuat snapshot DB menggunakan AWS CLI `create-db-snapshot` perintah (atau operasi `CreateDBSnapshot` RDS API). Setelah membuat snapshot DB, Anda dapat menambahkan tag seperti yang dijelaskan nanti dalam topik ini.

Tutorial: Menggunakan tag untuk menentukan instans DB yang akan dihentikan

Misalkan Anda membuat sejumlah instans DB dalam lingkungan pengembangan atau pengujian. Anda perlu mempertahankan semua instans DB ini selama beberapa hari. Beberapa instans DB menjalankan pengujian di malam hari. Instans DB lainnya dapat dihentikan di malam hari dan dimulai lagi keesokan harinya. Contoh berikut menunjukkan cara menetapkan tag untuk instans DB yang cocok untuk dihentikan di malam hari. Kemudian, contoh tersebut menunjukkan bagaimana skrip dapat mendeteksi instans DB yang memiliki tag lalu menghentikan instans DB tersebut. Dalam contoh ini, porsi nilai dari pasangan kunci-nilai tidaklah penting. Keberadaan tag `stoppable` menandakan bahwa instans DB memiliki properti yang ditetapkan pengguna ini.

Untuk menentukan instans DB yang akan dihentikan

1. Tentukan ARN instans DB yang ingin ditentukan sebagai dapat dihentikan.

Perintah dan API untuk pemberian tag berfungsi dengan ARN. Dengan begitu, mereka dapat bekerja dengan lancar di seluruh AWS Wilayah, AWS akun, dan berbagai jenis sumber daya yang mungkin memiliki nama pendek yang identik. Anda dapat menentukan ARN, bukan ID instans DB, dalam perintah CLI yang beroperasi pada instans DB. Gantikan nama instans DB Anda sendiri untuk *dev-test-db-instance*. Dalam perintah berikutnya yang menggunakan parameter ARN, ganti ARN instans DB Anda sendiri. ARN menyertakan ID AWS akun Anda sendiri dan nama AWS Wilayah tempat instans DB Anda berada.

```
$ aws rds describe-db-instances --db-instance-identifier dev-test-db-instance \
  --query "*[].{DBInstance:DBInstanceArn}" --output text
arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance
```

2. Tambahkan tag `stoppable` ke instans DB ini.

   Pilih nama untuk tag ini. Dengan pendekatan ini, berarti Anda tidak perlu merancang konvensi penamaan yang mengenkode semua informasi yang relevan dalam nama. Dalam konvensi tersebut, Anda dapat mengenkode informasi dalam nama instans DB atau nama-nama sumber daya lainnya. Karena memperlakukan tag sebagai atribut yang ada atau tidak ada, contoh ini menghilangkan bagian `Value=` dari parameter `--tags`.

   ```
   $ aws rds add-tags-to-resource \
     --resource-name arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance \
     --tags Key=stoppable
   ```

3. Konfirmasi bahwa tag tersebut ada dalam instans DB.

   Perintah ini mengambil informasi tag untuk instans DB dalam format JSON dan dalam bentuk teks biasa yang dipisahkan tab.

   ```
   $ aws rds list-tags-for-resource \
     --resource-name arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance
   {
       "TagList": [
           {
               "Key": "stoppable",
               "Value": ""

           }
       ]
   }
   ```

```
aws rds list-tags-for-resource \
  --resource-name arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance --
output text
TAGLIST stoppable
```

4. Untuk menghentikan semua instans DB yang ditetapkan sebagai `stoppable`, siapkan daftar semua instans DB Anda. Lihat daftar dan periksa apakah setiap instans DB diberi tag dengan atribut yang relevan.

   Contoh Linux ini menggunakan pembuatan skrip shell. Skrip ini menyimpan daftar ARN instans DB ke file sementara, lalu menjalankan perintah CLI untuk setiap instans DB.

```
$ aws rds describe-db-instances --query "*[].[DBInstanceArn]" --output text >/tmp/
db_instance_arns.lst
$ for arn in $(cat /tmp/db_instance_arns.lst)
do
  match="$(aws rds list-tags-for-resource --resource-name $arn --output text | grep
 stoppable)"
  if [[ ! -z "$match" ]]
  then
      echo "DB instance $arn is tagged as stoppable. Stopping it now."
# Note that you need to get the DB instance identifier from the ARN.
      dbid=$(echo $arn | sed -e 's/.*://')
      aws rds stop-db-instance --db-instance-identifier $dbid
  fi
done

DB instance arn:arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance is
 tagged as stoppable. Stopping it now.
{
    "DBInstance": {
        "DBInstanceIdentifier": "dev-test-db-instance",
        "DBInstanceClass": "db.t3.medium",
        ...
```

Anda dapat menjalankan skrip seperti ini pada akhir setiap hari untuk memastikan bahwa instans DB yang tidak penting dihentikan. Anda juga dapat menjadwalkan pekerjaan menggunakan utilitas seperti `cron` untuk melakukan pemeriksaan setiap malam. Misalnya, Anda mungkin melakukan ini jika beberapa instans DB dibiarkan berjalan secara tidak sengaja. Di sini, Anda dapat menyesuaikan perintah yang mempersiapkan daftar instans DB yang akan diperiksa.

Perintah berikut menghasilkan daftar instans DB, tetapi hanya yang berada dalam status `available`. Skrip ini dapat mengabaikan instans DB yang sudah dihentikan karena instans tersebut akan memiliki nilai status yang berbeda seperti `stopped` atau `stopping`.

```
$ aws rds describe-db-instances \
  --query '*[].{DBInstanceArn:DBInstanceArn,DBInstanceStatus:DBInstanceStatus}|[?
DBInstanceStatus == `available`]|[].{DBInstanceArn:DBInstanceArn}' \
  --output text
arn:aws:rds:us-east-1:123456789102:db:db-instance-2447
arn:aws:rds:us-east-1:123456789102:db:db-instance-3395
arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance
arn:aws:rds:us-east-1:123456789102:db:pg2-db-instance
```

> **Tip**
>
> Anda dapat menggunakan penetapan tag dan pencarian instans DB dengan tag tersebut untuk mengurangi biaya dengan cara lain. Misalnya, ikuti skenario ini dengan instans DB yang digunakan untuk pengembangan dan pengujian. Dalam hal ini, Anda dapat menetapkan beberapa instans DB yang akan dihapus pada akhir setiap hari. Atau Anda dapat menetapkan instans tersebut agar instans DB-nya diubah menjadi kelas instans DB kecil selama waktu penggunaan rendah yang telah diperkirakan.

Bekerja dengan Amazon Resource Name (ARN) di Amazon RDS

Sumber daya yang dibuat di Amazon Web Services masing-masing diidentifikasi secara unik dengan Amazon Resource Name (ARN). Untuk operasi Amazon RDS tertentu, Anda harus mengidentifikasi sumber daya Amazon RDS secara unik dengan menentukan ARN-nya. Misalnya, saat membuat replika baca instans DB RDS, Anda harus menyediakan ARN untuk instans DB sumber.

Membuat konsep ARN untuk Amazon RDS

Sumber daya yang dibuat di Amazon Web Services masing-masing diidentifikasi secara unik dengan Amazon Resource Name (ARN). Anda dapat membuat konsep ARN untuk sumber daya Amazon RDS menggunakan sintaks berikut.

`arn:aws:rds:<region>:<account number>:<resourcetype>:<name>`

Nama Wilayah	Wilayah	Titik Akhir	Protokol
AS Timur (Ohio)	us–east-2	rds.us-east-2.amazonaws.com	HTTPS
		rds-fips.us-east-2.api.aws	HTTPS
		rds.us-east-2.api.aws	HTTPS
		rds-fips.us-east-2.amazonaws.com	HTTPS
AS Timur (Virginia Utara)	us-east-1	rds.us-east-1.amazonaws.com	HTTPS
		rds-fips.us-east-1.api.aws	HTTPS
		rds-fips.us-east-1.amazonaws.com	HTTPS
		rds.us-east-1.api.aws	HTTPS
AS Barat (California Utara)	us-west-1	rds.us-west-1.amazonaws.com	HTTPS
		rds.us-west-1.api.aws	HTTPS
		rds-fips.us-west-1.amazonaws.com	HTTPS
		rds-fips.us-west-1.api.aws	HTTPS

Nama Wilayah	Wilayah	Titik Akhir	Protokol
AS Barat (Oregon)	us-west-2	rds.us-west-2.amazonaws.com rds-fips.us-west-2.amazonaws.com rds.us-west-2.api.aws rds-fips.us-west-2.api.aws	HTTPS HTTPS HTTPS HTTPS
Afrika (Cape Town)	af-south-1	rds.af-south-1.amazonaws.com rds.af-south-1.api.aws	HTTPS HTTPS
Asia Pasifik (Hong Kong)	ap-east-1	rds.ap-east-1.amazonaws.com rds.ap-east-1.api.aws	HTTPS HTTPS
Asia Pasifik (Hyderabad)	ap-south-2	rds.ap-south-2.amazonaws.com rds.ap-south-2.api.aws	HTTPS HTTPS
Asia Pasifik (Jakarta)	ap-southeast-3	rds.ap-southeast-3.amazonaws.com rds.ap-southeast-3.api.aws	HTTPS HTTPS
Asia Pasifik (Melbourne)	ap-southeast-4	rds.ap-southeast-4.amazonaws.com rds.ap-southeast-4.api.aws	HTTPS HTTPS
Asia Pasifik (Mumbai)	ap-south-1	rds.ap-south-1.amazonaws.com rds.ap-south-1.api.aws	HTTPS HTTPS

Nama Wilayah	Wilayah	Titik Akhir	Protokol
Asia Pasifik (Osaka)	ap-northeast-3	rds.ap-northeast-3.amazonaws.com	HTTPS
		rds.ap-northeast-3.api.aws	HTTPS
Asia Pasifik (Seoul)	ap-northeast-2	rds.ap-northeast-2.amazonaws.com	HTTPS
		rds.ap-northeast-2.api.aws	HTTPS
Asia Pasifik (Singapura)	ap-southeast-1	rds.ap-southeast-1.amazonaws.com	HTTPS
		rds.ap-southeast-1.api.aws	HTTPS
Asia Pasifik (Sydney)	ap-southeast-2	rds.ap-southeast-2.amazonaws.com	HTTPS
		rds.ap-southeast-2.api.aws	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	rds.ap-northeast-1.amazonaws.com	HTTPS
		rds.ap-northeast-1.api.aws	HTTPS
Kanada (Pusat)	ca-central-1	rds.ca-central-1.amazonaws.com	HTTPS
		rds.ca-central-1.api.aws	HTTPS
		rds-fips.ca-central-1.api.aws	HTTPS
		rds-fips.ca-central-1.amazonaws.com	HTTPS
Kanada Barat (Calgary)	ca-west-1	rds.ca-west-1.amazonaws.com	HTTPS
		rds-fips.ca-west-1.amazonaws.com	HTTPS
Eropa (Frankfurt)	eu-central-1	rds.eu-central-1.amazonaws.com	HTTPS
		rds.eu-central-1.api.aws	HTTPS

Nama Wilayah	Wilayah	Titik Akhir	Protokol
Eropa (Irlandia)	eu-west-1	rds.eu-west-1.amazonaws.com	HTTPS
		rds.eu-west-1.api.aws	HTTPS
Eropa (London)	eu-west-2	rds.eu-west-2.amazonaws.com	HTTPS
		rds.eu-west-2.api.aws	HTTPS
Eropa (Milan)	eu-south-1	rds.eu-south-1.amazonaws.com	HTTPS
		rds.eu-south-1.api.aws	HTTPS
Eropa (Paris)	eu-west-3	rds.eu-west-3.amazonaws.com	HTTPS
		rds.eu-west-3.api.aws	HTTPS
Eropa (Spanyol)	eu-south-2	rds.eu-south-2.amazonaws.com	HTTPS
		rds.eu-south-2.api.aws	HTTPS
Eropa (Stockholm)	eu-north-1	rds.eu-north-1.amazonaws.com	HTTPS
		rds.eu-north-1.api.aws	HTTPS
Eropa (Zürich)	eu-central-2	rds.eu-central-2.amazonaws.com	HTTPS
		rds.eu-central-2.api.aws	HTTPS
Israel (Tel Aviv)	il-central-1	rds.il-central-1.amazonaws.com	HTTPS
		rds.il-central-1.api.aws	HTTPS
Timur Tengah (Bahrain)	me-south-1	rds.me-south-1.amazonaws.com	HTTPS
		rds.me-south-1.api.aws	HTTPS

Nama Wilayah	Wilayah	Titik Akhir	Protokol
Timur Tengah (UAE)	me-central-1	rds.me-central-1.amazonaws.com rds.me-central-1.api.aws	HTTPS HTTPS
Amerika Selatan (Sao Paulo)	sa-east-1	rds.sa-east-1.amazonaws.com rds.sa-east-1.api.aws	HTTPS HTTPS
AWS GovCloud (AS-Timur)	us-gov-east-1	rds.us-gov-east-1.amazonaws.com rds.us-gov-east-1.api.aws	HTTPS HTTPS
AWS GovCloud (AS-Barat)	us-gov-west-1	rds.us-gov-west-1.amazonaws.com rds.us-gov-west-1.api.aws	HTTPS HTTPS

Tabel berikut menunjukkan format yang harus Anda gunakan saat membuat konsep ARN untuk jenis sumber daya Amazon RDS tertentu.

Jenis sumber daya	Format ARN
Instans DB	arn:aws:rds:*<region>*:*<account>* :db:*<name>* Misalnya: `arn:aws:rds: us-east-2 :123456789012 :db:my-mysql-instance-1`
Klaster DB	arn:aws:rds:*<region>*:*<account>* :cluster:*<name>* Misalnya:

Jenis sumber daya	Format ARN
	`arn:aws:rds: us-east-2 :123456789012 :cluster: my-aurora-cluster-1`
Langganan peristiwa	arn:aws:rds:*<region>*:*<account>* :es:*<name>* Misalnya: `arn:aws:rds: us-east-2 :123456789012 :es:my-subscription`
Grup opsi DB	arn:aws:rds:*<region>*:*<account>* :og:*<name>* Misalnya: `arn:aws:rds: us-east-2 :123456789012 :og:my-og`
Grup parameter DB	arn:aws:rds:*<region>*:*<account>* :pg:*<name>* Misalnya: `arn:aws:rds: us-east-2 :123456789012 :pg:my-param-enable-logs`
Grup parameter klaster DB	arn:aws:rds:*<region>*:*<account>* :cluster-pg:*<name>* Misalnya: `arn:aws:rds: us-east-2 :123456789012 :cluster-pg: my-cluster-param-timezone`
Instans DB yang dicadangkan	arn:aws:rds:*<region>*:*<account>* :ri:*<name>* Misalnya: `arn:aws:rds: us-east-2 :123456789012 :ri:my-reserved-postgresql`

Jenis sumber daya	Format ARN
Grup keamanan DB	arn:aws:rds:*<region>*:*<account>* :secgrp:*<name>* Misalnya: `arn:aws:rds: us-east-2 :123456789012 :secgrp:my-public`
Snapshot DB otomatis	arn:aws:rds:*<region>*:*<account>* :snapshot:rds:*<name>* Misalnya: `arn:aws:rds: us-east-2 :123456789012 :snapshot:rds: my-mysql-db-2019-07-22-07-23`
Snapshot klaster DB otomatis	arn:aws:rds:*<region>*:*<account>* :cluster-snapshot:rds:*<name>* Misalnya: `arn:aws:rds: us-east-2 :123456789012 :cluster-snapshot:rds: my-aurora-cluster-2019-07-22-16-16`
Snapshot DB manual	arn:aws:rds:*<region>*:*<account>* :snapshot:*<name>* Misalnya: `arn:aws:rds: us-east-2 :123456789012 :snapshot: my-mysql-db-snap`
Snapshot klaster DB manual	arn:aws:rds:*<region>*:*<account>* :cluster-snapshot:*<name>* Misalnya: `arn:aws:rds: us-east-2 :123456789012 :cluster-snapshot: my-aurora-cluster-snap`

Jenis sumber daya	Format ARN
Grup subnet DB	arn:aws:rds:*<region>*:*<account>* :subgrp:*<name>* Misalnya: `arn:aws:rds: us-east-2 :123456789012 :subgrp:my-subnet -10`

Mendapatkan ARN yang sudah ada

Anda bisa mendapatkan ARN dari sumber daya RDS dengan menggunakan AWS Management Console, AWS Command Line Interface (AWS CLI), atau RDS API.

Konsol

Untuk mendapatkan ARN dari AWS Management Console, navigasikan ke sumber daya yang Anda inginkan untuk ARN, dan lihat detail untuk sumber daya itu.

Misalnya, Anda bisa mendapatkan ARN untuk instans DB dari tab Konfigurasi detail instans DB.

AWS CLI

Untuk mendapatkan ARN dari sumber daya RDS tertentu, Anda menggunakan `describe` perintah untuk sumber daya itu. AWS CLI Tabel berikut menunjukkan setiap AWS CLI perintah, dan properti ARN yang digunakan dengan perintah untuk mendapatkan ARN.

AWS CLI perintah	Properti ARN
describe-event-subscriptions	EventSubscriptionArn
describe-certificates	CertificateArn
describe-db-parameter-groups	DB ParameterGroupArn
describe-db-cluster-parameter-kelompok	DB ClusterParameterGroupArn
describe-db-instances	DB InstanceArn

AWS CLI perintah	Properti ARN
describe-db-security-groups	DB SecurityGroupArn
describe-db-snapshots	DB SnapshotArn
describe-events	SourceArn
describe-reserved-db-instances	ReservedDB InstanceArn
describe-db-subnet-groups	DB SubnetGroupArn
describe-option-groups	OptionGroupArn
describe-db-clusters	DB ClusterArn
describe-db-cluster-snapshots	DB ClusterSnapshotArn

Misalnya, AWS CLI perintah berikut mendapatkan ARN untuk instance DB.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds describe-db-instances \
--db-instance-identifier DBInstanceIdentifier \
--region us-west-2 \
--query "*[].{DBInstanceIdentifier:DBInstanceIdentifier,DBInstanceArn:DBInstanceArn}"
```

Untuk Windows:

```
aws rds describe-db-instances ^
--db-instance-identifier DBInstanceIdentifier ^
--region us-west-2 ^
--query "*[].{DBInstanceIdentifier:DBInstanceIdentifier,DBInstanceArn:DBInstanceArn}"
```

Output dari perintah tersebut adalah sebagai berikut:

```
[
    {
```

```
        "DBInstanceArn": "arn:aws:rds:us-west-2:account_id:db:instance_id",
        "DBInstanceIdentifier": "instance_id"
    }
]
```

RDS API

Untuk mendapatkan ARN untuk sumber daya RDS tertentu, Anda dapat memanggil operasi API RDS berikut dan menggunakan properti ARN yang ditunjukkan sebagai berikut.

Operasi API RDS	Properti ARN
DescribeEventSubscriptions	EventSubscriptionArn
DescribeCertificates	CertificateArn
DijelaskanB ParameterGroups	DB ParameterGroupArn
DijelaskanB ClusterParameterGroups	DB ClusterParameterGroupArn
DescribeDBInstances	DB InstanceArn
DijelaskanB SecurityGroups	DB SecurityGroupArn
DescribeDBSnapshots	DB SnapshotArn
DescribeEvents	SourceArn
DescribeReservedDBInstances	ReservedDB InstanceArn
DijelaskanB SubnetGroups	DB SubnetGroupArn
DescribeOptionGroups	OptionGroupArn
DescribeDBClusters	DB ClusterArn
DijelaskanB ClusterSnapshots	DB ClusterSnapshotArn

Menggunakan penyimpanan untuk instans DB Amazon RDS

Untuk menentukan cara bagaimana data Anda disimpan di Amazon RDS, pilih jenis penyimpanan dan berikan ukuran penyimpanan saat Anda membuat atau memodifikasi instans DB. Kemudian, Anda dapat meningkatkan jumlah atau mengubah jenis penyimpanan dengan memodifikasi instans DB. Untuk informasi selengkapnya tentang jenis penyimpanan yang akan digunakan untuk beban kerja Anda, lihat Jenis penyimpanan Amazon RDS.

Topik

Meningkatkan kapasitas penyimpanan instans DB

Jika Anda memerlukan ruang untuk data tambahan, Anda dapat menaikkan skala penyimpanan instans DB yang sudah ada. Untuk melakukannya, Anda dapat menggunakan Amazon RDS Management Console, API Amazon RDS, atau AWS Command Line Interface (AWS CLI). Untuk informasi tentang batas penyimpanan, lihat Penyimpanan instans DB Amazon RDS.

> **Note**
>
> Menskalakan penyimpanan untuk instans DB Amazon RDS for Microsoft SQL Server hanya didukung untuk jenis penyimpanan SSD Tujuan Umum atau SSD IOPS yang Tersedia.

Untuk memantau jumlah penyimpanan gratis untuk instans DB Anda sehingga Anda dapat merespons bila perlu, kami sarankan Anda membuat CloudWatch alarm Amazon. Untuk informasi selengkapnya tentang pengaturan CloudWatch alarm, lihat Menggunakan CloudWatch alarm.

Menskalakan penyimpanan biasanya tidak menyebabkan pemadaman atau penurunan performa instans DB. Setelah Anda mengubah ukuran penyimpanan untuk instans DB, status instans DB adalah storage-optimization.

> **Note**
>
> Optimalisasi penyimpanan dapat membutuhkan waktu beberapa jam. Anda tidak dapat melakukan modifikasi penyimpanan lebih lanjut selama enam (6) jam atau hingga pengoptimalan penyimpanan pada instans selesai, mana pun yang lebih lama. Anda dapat melihat kemajuan pengoptimalan penyimpanan di AWS Management Console atau dengan menggunakan describe-db-instances AWS CLI perintah.

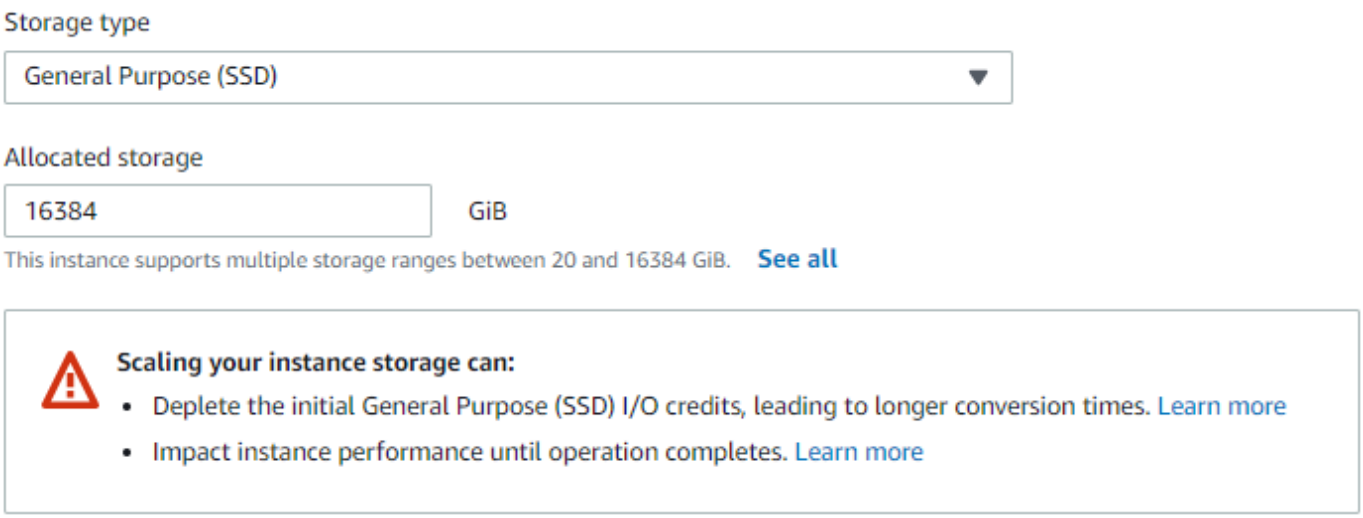
Namun, kasus khusus adalah jika Anda memiliki instans DB SQL Server dan belum memodifikasi konfigurasi penyimpanan sejak November 2017. Dalam hal ini, Anda mungkin mengalami pemadaman singkat beberapa menit ketika Anda memodifikasi instans DB untuk meningkatkan alokasi penyimpanan. Setelah pemadaman, instans DB sedang online tetapi dalam status `storage-optimization`. Performanya mungkin menurun selama pengoptimalan penyimpanan.

> **Note**
>
> Anda tidak dapat mengurangi jumlah penyimpanan untuk instans DB setelah penyimpanan dialokasikan. Saat Anda meningkatkan alokasi penyimpanan, peningkatannya setidaknya harus 10 persen. Jika Anda mencoba meningkatkan nilai sebesar kurang dari 10 persen, Anda akan mendapat kesalahan.

Konsol

Untuk meningkatkan penyimpanan untuk instans DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data.
3. Pilih instans DB yang ingin Anda ubah.
4. Pilih Modifikasi.
5. Masukkan nilai baru untuk Penyimpanan yang dialokasikan. Nilai ini harus lebih besar dari nilai saat ini.



6. Pilih Lanjutkan untuk beralih ke layar berikutnya.
7. Pilih Langsung terapkan dalam bagian Penjadwalan modifikasi untuk menerapkan perubahan penyimpanan ke instans DB dengan segera.

   Atau pilih Terapkan pada jendela pemeliharaan terjadwal berikutnya untuk menerapkan perubahan pada jendela pemeliharaan berikutnya.
8. Jika pengaturan sesuai keinginan Anda, pilih Modifikasi instans DB.

AWS CLI

Untuk meningkatkan penyimpanan untuk instans DB, gunakan AWS CLI perintah `modify-db-instance`. Atur parameter berikut:

- `--allocated-storage` – Jumlah penyimpanan yang akan dialokasikan untuk instans DB, dalam gibibyte.
- `--apply-immediately` – Gunakan `--apply-immediately` untuk langsung menerapkan perubahan penyimpanan.

  Gunakan `--no-apply-immediately` (default) untuk menerapkan perubahan saat jendela pemeliharaan berikutnya. Pemadaman langsung terjadi saat perubahan diterapkan.

Untuk informasi selengkapnya tentang penyimpanan, lihat Penyimpanan instans DB Amazon RDS.

API RDS

Untuk meningkatkan penyimpanan instans DB, gunakan perintah operasi API Amazon RDS `ModifyDBInstance`. Atur parameter berikut:

- `AllocatedStorage` – Jumlah penyimpanan yang akan dialokasikan untuk instans DB, dalam gibibyte.
- `ApplyImmediately` – Atur opsi ini `True` untuk segera menerapkan perubahan penyimpanan. Atur opsi ini ke `False` (default) untuk menerapkan perubahan pada jendela pemeliharaan berikutnya. Pemadaman langsung terjadi saat perubahan diterapkan.

Untuk informasi selengkapnya tentang penyimpanan, lihat Penyimpanan instans DB Amazon RDS.

Mengelola kapasitas secara otomatis dengan penskalaan otomatis penyimpanan Amazon RDS

Jika beban kerja Anda tidak dapat diprediksi, Anda dapat mengaktifkan penskalaan otomatis penyimpanan untuk instans DB Amazon RDS. Untuk melakukannya, Anda dapat menggunakan konsol Amazon RDS, API Amazon RDS, atau AWS CLI.

Misalnya, Anda mungkin menggunakan fitur ini untuk aplikasi game seluler baru yang diadopsi pengguna dengan cepat. Dalam hal ini, peningkatan beban kerja yang cepat dapat melebihi penyimpanan basis data yang tersedia. Agar tidak perlu meningkatkan penyimpanan basis data secara manual, Anda dapat menggunakan penskalaan otomatis penyimpanan Amazon RDS.

Dengan penskalaan otomatis penyimpanan diaktifkan, saat Amazon RDS mendeteksi bahwa Anda kehabisan ruang basis data, akan secara otomatis meningkatkan penyimpanan Anda. Amazon RDS memulai modifikasi penyimpanan untuk instans DB yang diaktifkan dengan penskalaan otomatis ketika faktor ini berlaku:

- Ruang kosong yang tersedia kurang dari atau sama dengan 10 persen dari alokasi penyimpanan.
- Kondisi penyimpanan rendah berlangsung setidaknya lima menit.
- Setidaknya enam jam telah berlalu sejak modifikasi penyimpanan terakhir, atau pengoptimalan penyimpanan telah selesai instans, mana pun yang lebih lama.

Penyimpanan tambahan berada dalam kelipatan mana pun dari penyimpanan berikut ini yang lebih besar:

- 10 GiB
- 10 persen alokasi penyimpanan saat ini
- Pertumbuhan penyimpanan yang diperkirakan melebihi ukuran penyimpanan yang dialokasikan saat ini dalam 7 jam ke depan berdasarkan metrik `FreeStorageSpace` dari satu jam terakhir. Untuk informasi selengkapnya tentang metrik, lihat Memantau dengan Amazon CloudWatch.

Ambang batas penyimpanan maksimum adalah batas yang Anda tetapkan untuk penskalaan otomatis instans DB. Hal ini memiliki batasan berikut:

- Anda harus menetapkan ambang batas penyimpanan maksimum setidaknya 10% lebih banyak dari penyimpanan yang dialokasikan saat ini. Sebaiknya atur ke setidaknya 26% lebih banyak untuk menghindari penerimaan pemberitahuan peristiwa bahwa ukuran penyimpanan mendekati ambang batas penyimpanan maksimum.

  Misalnya, jika Anda memiliki instans DB dengan 1.000 GiB penyimpanan yang dialokasikan, atur ambang batas penyimpanan maksimum setidaknya 1.100 GiB. Jika tidak, Anda mendapatkan kesalahan seperti Ukuran penyimpanan maks tidak valid untuk *`engine_name`*. Namun, sebaiknya Anda mengatur ambang batas penyimpanan maksimum setidaknya 1.260 GiB untuk menghindari pemberitahuan peristiwa.
- Untuk instance DB yang menggunakan penyimpanan IOPS Tertentu (io1 atau io2 Block Express), rasio IOPS terhadap ambang penyimpanan maksimum (dalam GiB) harus berada dalam kisaran tertentu. Untuk informasi selengkapnya, lihat Penyimpanan SSD IOPS yang Tersedia.
- Anda tidak dapat mengatur ambang batas penyimpanan maksimum untuk instans yang mendukung penskalaan otomatis ke nilai yang lebih besar dari alokasi penyimpanan maksimum untuk mesin basis data dan kelas instans DB.

  Sebagai contoh, SQL Server Standard Edition pada db.m5.xlarge memiliki penyimpanan default yang dialokasikan untuk instans sebesar 20 GiB (minimum) dan penyimpanan maksimum yang dialokasikan sebesar 16.384 GiB. Ambang batas penyimpanan maksimum default untuk penskalaan otomatis adalah 1.000 GiB. Jika Anda menggunakan batas default ini, instans tidak otomatis diskalakan otomatis melebihi 1.000 GiB. Hal ini berlaku meskipun penyimpanan maksimum yang dialokasikan untuk instans tersebut adalah 16.384 GiB.

> **Note**
>
> Sebaiknya Anda memilih dengan cermat ambang batas penyimpanan maksimum berdasarkan pola penggunaan dan kebutuhan pelanggan. Jika ada penyimpangan dalam pola penggunaan, ambang batas penyimpanan maksimum dapat mencegah penskalaan penyimpanan ke nilai tinggi yang tidak terduga ketika penskalaan otomatis memperkirakan ambang batas yang sangat tinggi. Setelah instans DB telah diskalakan otomatis, alokasi penyimpanannya tidak dapat dikurangi.

Topik

Batasan

Batasan berikut berlaku untuk penskalaan otomatis penyimpanan:

- Penskalaan otomatis tidak terjadi jika ambang batas penyimpanan maksimum akan disamakan atau dilampaui oleh peningkatan penyimpanan.
- Saat menskalakan otomatis, RDS memprediksi ukuran penyimpanan untuk operasi penskalaan otomatis berikutnya. Jika operasi selanjutnya diperkirakan melebihi ambang batas penyimpanan maksimum, skala otomatis RDS ke ambang batas penyimpanan maksimum.
- Penskalaan otomatis tidak dapat sepenuhnya mencegah situasi penyimpanan penuh untuk muatan data yang besar. Hal ini karena modifikasi penyimpanan lebih lanjut selama enam (6) jam atau hingga pengoptimalan penyimpanan pada instans selesai, mana pun yang lebih lama.

  Jika Anda melakukan pemuatan data besar, dan penskalaan otomatis tidak memberikan ruang yang cukup, basis data mungkin tetap berada dalam status penyimpanan penuh selama beberapa jam. Tindakan ini dapat membahayakan basis data.
- Jika Anda memulai operasi penskalaan penyimpanan pada saat yang sama ketika Amazon RDS memulai operasi penskalaan otomatis, modifikasi penyimpanan Anda lebih diutamakan. Operasi penskalaan otomatis dibatalkan.

- Penskalaan otomatis tidak dapat mengurangi penyimpanan yang dialokasikan. Anda tidak dapat mengurangi jumlah penyimpanan untuk instans DB setelah penyimpanan dialokasikan.
- Penskalaan otomatis tidak dapat digunakan dengan penyimpanan magnetik.
- Penskalaan otomatis tidak dapat digunakan dengan kelas instans generasi sebelumnya berikut ini yang memiliki kurang dari 6 TiB penyimpanan yang dapat dipesan: db.m3.large, db.m3.xlarge, and db.m3.2xlarge.
- Operasi penskalaan otomatis tidak dicatat oleh. AWS CloudTrail Untuk informasi lebih lanjut tentang CloudTrail, lihatMemantau panggilan API Amazon RDS di AWS CloudTrail.

Meskipun penskalaan otomatis membantu Anda meningkatkan penyimpanan di instans DB Amazon RDS secara dinamis, Anda masih harus mengonfigurasi penyimpanan awal untuk instans DB ke ukuran yang sesuai dengan beban kerja tipikal Anda.

Mengaktifkan penskalaan otomatis penyimpanan untuk instans DB baru

Saat Anda membuat instans DB Amazon RDS baru, Anda dapat memilih apakah akan mengaktifkan penskalaan otomatis penyimpanan. Anda juga dapat mengatur batas maksimal di penyimpanan yang dapat dialokasikan Amazon RDS untuk instans DB.

> **Note**
>
> Saat Anda mengkloning instans DB Amazon RDS yang memiliki penskalaan otomatis penyimpanan aktif, pengaturan tersebut tidak secara otomatis diwariskan kepada instans yang dikloning. Instans DB baru memiliki jumlah alokasi penyimpanan yang sama dengan instans asli. Anda dapat mengaktifkan kembali penskalaan otomatis penyimpanan untuk instans baru jika instans yang digandakan terus meningkatkan kebutuhan penyimpanannya.

Konsol

Untuk mengaktifkan penskalaan otomatis penyimpanan untuk instans DB baru

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di sudut kanan atas konsol Amazon RDS, pilih AWS Wilayah tempat Anda ingin membuat instans DB.
3. Di panel navigasi, pilih Basis Data.

4. Pilih Buat basis data. Di halaman Pilih mesin, pilih mesin basis data Anda dan tentukan informasi instans DB Anda seperti yang dijelaskan di Mulai menggunakan Amazon RDS.
5. Di bagian Penskalaan otomatis penyimpanan, atur nilai Ambang batas maksimum penyimpanan untuk instans DB.
6. Tentukan sisa informasi instans DB Anda seperti yang dijelaskan di Mulai menggunakan Amazon RDS.

AWS CLI

Untuk mengaktifkan penskalaan otomatis penyimpanan untuk instans DB baru, gunakan perintah. AWS CLI `create-db-instance` Atur parameter berikut:

- `--max-allocated-storage` – Mengaktifkan penskalaan otomatis penyimpanan dan mengatur batas maksimal pada ukuran penyimpanan, dalam gibibyte.

Untuk memverifikasi bahwa penskalaan otomatis penyimpanan Amazon RDS tersedia untuk instans DB Anda, gunakan perintah. AWS CLI `describe-valid-db-instance-modifications` Untuk memeriksa berdasarkan kelas instans sebelum membuat instans, gunakan perintah `describe-orderable-db-instance-options`. Periksa kolom berikut dalam nilai hasil:

- `SupportsStorageAutoscaling` – Mengindikasikan apakah instans DB atau kelas instans mendukung penskalaan otomatis penyimpanan.

Untuk informasi selengkapnya tentang penyimpanan, lihat Penyimpanan instans DB Amazon RDS.

API RDS

Untuk mengaktifkan penskalaan otomatis penyimpanan untuk instans DB baru, gunakan operasi API Amazon RDS `CreateDBInstance`. Atur parameter berikut:

- `MaxAllocatedStorage` – Mengaktifkan penskalaan otomatis penyimpanan Amazon RDS dan mengatur batas maksimal pada ukuran penyimpanan, dalam gibibyte.

Untuk memverifikasi bahwa penskalaan otomatis penyimpanan Amazon RDS tersedia untuk instans DB Anda, gunakan operasi API Amazon RDS `DescribeValidDbInstanceModifications` untuk instans yang ada, atau operasi `DescribeOrderableDBInstanceOptions` sebelum membuat instans. Periksa kolom berikut dalam nilai hasil:

- `SupportsStorageAutoscaling` – Mengindikasikan apakah instans DB mendukung penskalaan otomatis penyimpanan.

Untuk informasi selengkapnya tentang penyimpanan, lihat Penyimpanan instans DB Amazon RDS.

Mengubah pengaturan penskalaan otomatis penyimpanan untuk instans DB

Anda dapat mengaktifkan penskalaan otomatis penyimpanan untuk instans DB Amazon RDS yang sudah ada. Anda juga dapat mengubah batas maksimal di penyimpanan yang dapat dialokasikan Amazon RDS untuk instans DB.

Konsol

Untuk mengubah pengaturan penskalaan otomatis penyimpanan untuk instans DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data.
3. Pilih instans DB yang ingin dimodifikasi, lalu pilih Modifikasi. Halaman Modifikasi instans DB akan muncul.
4. Ubah batas penyimpanan di bagian Penskalaan otomatis. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.
5. Jika semua perubahan sudah sesuai dengan keinginan Anda, pilih Lanjutkan dan periksa modifikasi Anda.
6. Di halaman konfirmasi, tinjau perubahan Anda. Jika sudah benar, pilih Modifikasi instans DB untuk menyimpan perubahan Anda. Jika tidak benar, pilih Kembali untuk mengedit perubahan Anda atau Batalkan untuk membatalkan perubahan Anda.

   Mengubah batas penskalaan otomatis penyimpanan langsung terjadi. Pengaturan ini mengabaikan pengaturan Langsung terapkan.

AWS CLI

Untuk mengubah pengaturan penskalaan otomatis penyimpanan untuk instans DB, gunakan perintah. AWS CLI `modify-db-instance` Atur parameter berikut:

- `--max-allocated-storage` – Mengatur batas maksimal ukuran penyimpanan, dalam gibibyte. Jika nilainya lebih besar dari parameter `--allocated-storage`, penskalaan otomatis

penyimpanan diaktifkan. Jika nilainya sama dengan parameter `--allocated-storage`, penskalaan otomatis penyimpanan dinonaktifkan.

Untuk memverifikasi bahwa penskalaan otomatis penyimpanan Amazon RDS tersedia untuk instans DB Anda, gunakan perintah. AWS CLI `describe-valid-db-instance-modifications` Untuk memeriksa berdasarkan kelas instans sebelum membuat instans, gunakan perintah `describe-orderable-db-instance-options`. Periksa kolom berikut dalam nilai hasil:

- `SupportsStorageAutoscaling` – Mengindikasikan apakah instans DB mendukung penskalaan otomatis penyimpanan.

Untuk informasi selengkapnya tentang penyimpanan, lihat Penyimpanan instans DB Amazon RDS.

API RDS

Untuk mengubah pengaturan penskalaan otomatis penyimpanan untuk instans DB, gunakan operasi API Amazon RDS `ModifyDBInstance`. Atur parameter berikut:

- `MaxAllocatedStorage` – Mengatur batas maksimal ukuran penyimpanan, dalam gibibyte.

Untuk memverifikasi bahwa penskalaan otomatis penyimpanan Amazon RDS tersedia untuk instans DB Anda, gunakan operasi API Amazon RDS `DescribeValidDbInstanceModifications` untuk instans yang ada, atau operasi `DescribeOrderableDBInstanceOptions` sebelum membuat instans. Periksa kolom berikut dalam nilai hasil:

- `SupportsStorageAutoscaling` – Mengindikasikan apakah instans DB mendukung penskalaan otomatis penyimpanan.

Untuk informasi selengkapnya tentang penyimpanan, lihat Penyimpanan instans DB Amazon RDS.

Menonaktifkan penskalaan otomatis penyimpanan untuk instans DB

Jika Anda tidak perlu lagi Amazon RDS untuk meningkatkan penyimpanan secara otomatis untuk instans DB Amazon RDS, Anda dapat menonaktifkan penskalaan otomatis penyimpanan. Setelah itu, Anda masih dapat meningkatkan jumlah penyimpanan secara manual untuk instans DB Anda.

Konsol

Untuk menonaktifkan penskalaan otomatis penyimpanan instans DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data.
3. Pilih instans DB yang ingin dimodifikasi, lalu pilih Modifikasi. Halaman Modifikasi instans DB akan muncul.
4. Hapus kontak centang Aktifkan penskalaan otomatis penyimpanan di bagian Penskalaan otomatis penyimpanan. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.
5. Jika semua perubahan sudah sesuai dengan keinginan Anda, pilih Lanjutkan dan periksa modifikasi tersebut.
6. Di halaman konfirmasi, tinjau perubahan Anda. Jika sudah benar, pilih Modifikasi instans DB untuk menyimpan perubahan Anda. Jika tidak benar, pilih Kembali untuk mengedit perubahan Anda atau Batalkan untuk membatalkan perubahan Anda.

Mengubah batas penskalaan otomatis penyimpanan langsung terjadi. Pengaturan ini mengabaikan pengaturan Langsung terapkan.

AWS CLI

Untuk mematikan penskalaan otomatis penyimpanan untuk instans DB, gunakan AWS CLI perintah `modify-db-instance`dan parameter berikut:

- `--max-allocated-storage` – Tentukan nilai yang sama dengan pengaturan `--allocated-storage` untuk mencegah penskalaan otomatis penyimpanan Amazon RDS lebih lanjut untuk instans DB yang ditentukan.

Untuk informasi selengkapnya tentang penyimpanan, lihat Penyimpanan instans DB Amazon RDS.

API RDS

Untuk menonaktifkan penskalaan otomatis penyimpanan untuk instans DB, gunakan operasi API Amazon RDS `ModifyDBInstance`. Atur parameter berikut:

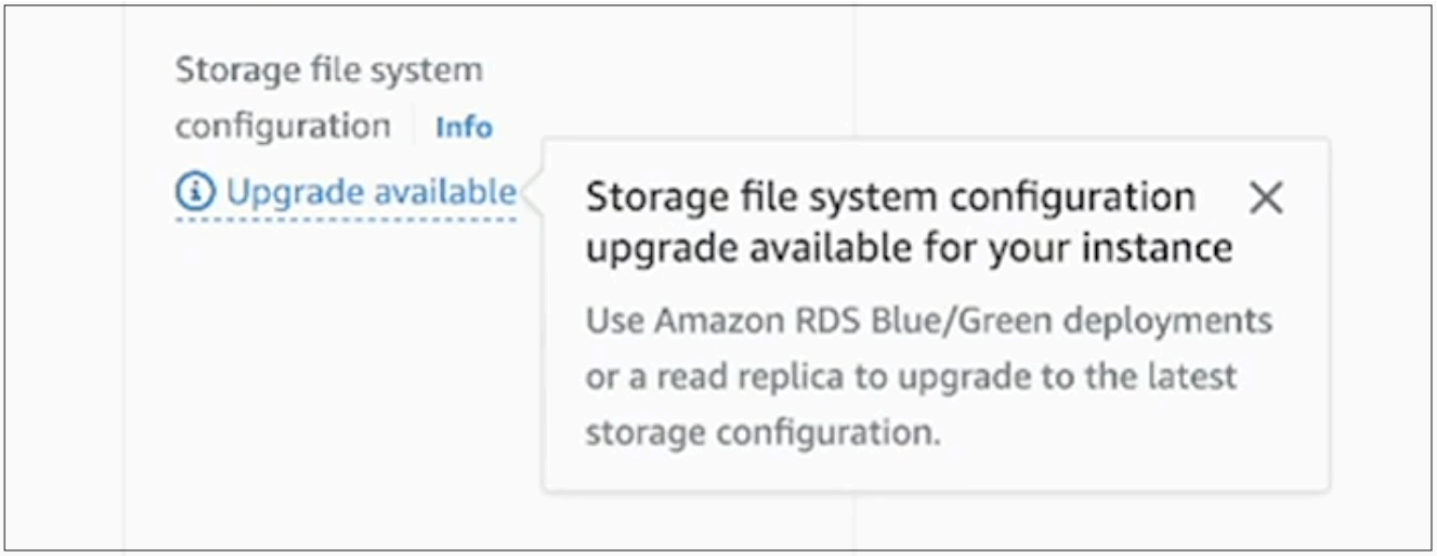
- `MaxAllocatedStorage` – Tentukan nilai yang sama dengan pengaturan `AllocatedStorage` untuk mencegah penskalaan otomatis penyimpanan Amazon RDS lebih lanjut untuk instans DB yang ditentukan.

Untuk informasi selengkapnya tentang penyimpanan, lihat Penyimpanan instans DB Amazon RDS.

Meningkatkan sistem file penyimpanan untuk instans DB

Sebagian besar instans DB RDS menawarkan ukuran penyimpanan maksimum 64 TiB untuk basis data MySQL dan MariaDB. Namun, beberapa sistem file 32-bit yang lebih lama memiliki kapasitas penyimpanan yang lebih rendah. Untuk menentukan kapasitas penyimpanan instans DB Anda, Anda dapat menggunakan AWS CLI perintah describe-valid-db-instance-modifikasi.

Jika RDS mendeteksi bahwa salah satu instans DB Anda menjalankan sistem file yang lebih lama (yang memiliki ukuran penyimpanan 16 TiB, batas ukuran file 2 TiB, atau penulisan yang tidak dioptimalkan), konsol RDS memberi tahu Anda bahwa konfigurasi sistem file Anda memenuhi syarat untuk peningkatan. Anda dapat memeriksa kelayakan peningkatan instans DB Anda di panel Penyimpanan pada halaman detail instans DB.



Jika instans DB Anda memenuhi syarat untuk peningkatan sistem file, Anda dapat melakukan peningkatan dengan salah satu dari dua cara:

- Buat deployment blue/green dan tentukan Tingkatkan konfigurasi sistem file penyimpanan. Opsi ini meningkatkan sistem file di lingkungan green ke konfigurasi yang disukai. Anda kemudian dapat beralih antara deployment blue/green, yang mendukung lingkungan hijau sebagai lingkungan

produksi yang baru. Untuk petunjuk mendetail, lihat the section called “Membuat deployment blue/green”.

- Buat replika baca instans DB dan tentukan Tingkatkan konfigurasi sistem file penyimpanan. Opsi ini meningkatkan sistem file replika baca ke konfigurasi pilihan. Anda kemudian dapat mempromosikan replika baca menjadi instans mandiri. Untuk petunjuk mendetail, lihat the section called “Membuat replika baca”.

Meningkatkan konfigurasi penyimpanan adalah operasi intensif I/O dan menyebabkan waktu pembuatan yang lebih lama untuk deployment blue/green dan replika baca. Proses peningkatan penyimpanan lebih cepat jika instans DB sumber menggunakan penyimpanan Provisioned IOPS SSD (io1 atau io2 Block Express) dan Anda menyediakan lingkungan hijau atau membaca replika dengan ukuran instans 4xlarge atau lebih besar. Peningkatan penyimpanan yang melibatkan penyimpanan SSD Tujuan Umum (gp2) dapat mengurangi saldo kredit I/O, sehingga menyebabkan waktu peningkatan yang lebih lama. Untuk informasi selengkapnya, lihat the section called “Penyimpanan instans DB”.

Selama proses peningkatan penyimpanan, mesin basis data tidak tersedia. Jika penggunaan penyimpanan pada instans DB sumber Anda lebih besar dari atau sama dengan 90% dari ukuran penyimpanan yang dialokasikan, proses peningkatan penyimpanan akan meningkatkan ukuran penyimpanan yang dialokasikan sebesar 10% untuk instans green atau replika baca.

Memodifikasi pengaturan penyimpanan SSD IOPS yang Tersedia

Anda dapat memodifikasi pengaturan untuk instans DB yang menggunakan penyimpanan SSD IOPS yang Tersedia menggunakan konsol Amazon RDS, AWS CLI, atau API Amazon RDS. Tentukan jenis penyimpanan, alokasi penyimpanan, dan jumlah IOPS yang Tersedia yang Anda butuhkan. Rentang ini bergantung pada mesin basis data dan jenis instans Anda.

Meskipun Anda dapat mengurangi jumlah IOPS yang tersedia untuk instans, Anda tidak dapat mengurangi ukuran penyimpanan.

Dalam banyak kasus, menskalakan penyimpanan tidak memerlukan pemadaman dan tidak menurunkan performa server. Setelah Anda mengubah IOPS penyimpanan untuk instans DB, status instans DB adalah storage-optimization.

> **Note**
>
> Pengoptimalan penyimpanan dapat membutuhkan waktu beberapa jam. Anda tidak dapat melakukan modifikasi penyimpanan lebih lanjut selama enam (6) jam atau hingga pengoptimalan penyimpanan pada instans selesai, mana pun yang lebih lama.

Untuk informasi tentang rentang penyimpanan yang dialokasikan dan IOPS yang Tersedia yang ada untuk setiap mesin basis data, lihat Penyimpanan SSD IOPS yang Tersedia.

Konsol

Untuk mengubah pengaturan IOPS yang Tersedia untuk instans DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data.

   Untuk memfilter daftar instans DB, untuk Filter basis data masukkan string teks untuk Amazon RDS yang digunakan untuk memfilter hasil. Hanya instans DB yang namanya berisi string yang muncul.
3. Pilih instans DB dengan IOPS yang Tersedia yang ingin Anda modifikasi.
4. Pilih Modifikasi.
5. Pada halaman Modify DB instans, pilih Provisioned IOPS SSD (io1) atau Provisioned IOPS SSD (io2) untuk tipe Storage.
6. Untuk IOPS yang Tersedia, masukkan nilai.

   Jika nilai yang Anda tetapkan untuk Alokasi penyimpanan atau IOPS yang Tersedia berada di luar batas yang didukung oleh parameter lain, pesan peringatan akan ditampilkan. Pesan ini memberikan rentang nilai yang diperlukan untuk parameter lainnya.
7. Pilih Lanjutkan.
8. Pilih Langsung terapkan dalam bagian Penjadwalan modifikasi untuk menerapkan perubahan ke instans DB dengan segera. Atau pilih Terapkan pada jendela pemeliharaan terjadwal berikutnya untuk menerapkan perubahan pada jendela pemeliharaan berikutnya.
9. Tinjau parameter yang akan diubah, dan pilih Modifikasi instans DB untuk menyelesaikan modifikasi.

Nilai baru untuk alokasi penyimpanan atau untuk IOPS yang Tersedia muncul di kolom Status.

AWS CLI

Untuk mengubah pengaturan IOPS yang Disediakan untuk instans DB, gunakan perintah. AWS CLI `modify-db-instance` Atur parameter berikut:

- `--storage-type`— Setel ke `io1` atau `io2` untuk IOPS yang Disediakan.
- `--allocated-storage` – Jumlah penyimpanan yang akan dialokasikan untuk instans DB, dalam gibibyte.
- `--iops` – Jumlah baru IOPS yang Tersedia untuk instans DB, dinyatakan dalam operasi I/O per detik.
- `--apply-immediately` – Gunakan `--apply-immediately` untuk segera menerapkan perubahan. Gunakan `--no-apply-immediately` (default) untuk menerapkan perubahan selama jendela pemeliharaan berikutnya.

API RDS

Untuk mengubah pengaturan IOPS yang Tersedia untuk instans DB, gunakan operasi API Amazon RDS `ModifyDBInstance`. Atur parameter berikut:

- `StorageType`— Setel ke `io1` atau `io2` untuk IOPS yang Disediakan.
- `AllocatedStorage` – Jumlah penyimpanan yang akan dialokasikan untuk instans DB, dalam gibibyte.
- `Iops` – Rasio IOPS baru untuk instans DB, dinyatakan dalam operasi I/O per detik.
- `ApplyImmediately` – Atur opsi ini ke `True` untuk segera menerapkan perubahan. Atur opsi ini ke `False` (default) untuk menerapkan perubahan pada jendela pemeliharaan berikutnya.

Modifikasi penyimpanan intensif I/O

Instans DB Amazon RDS menggunakan volume Amazon Elastic Block Store (EBS) untuk penyimpanan basis data dan log. Tergantung pada jumlah penyimpanan yang diminta, RDS (kecuali RDS for SQL Server) secara otomatis melakukan striping beberapa volume Amazon EBS untuk meningkatkan performa. Instans DB RDS dengan jenis penyimpanan SSD didukung oleh satu atau empat volume Amazon EBS yang di-striping dalam konfigurasi RAID 0. Secara desain, operasi

modifikasi penyimpanan untuk instans DB RDS memiliki dampak minimal pada operasi basis data yang sedang berlangsung.

Dalam kebanyakan kasus, modifikasi penskalaan penyimpanan sepenuhnya diturunkan ke lapisan Amazon EBS dan transparan ke basis data. Proses ini biasanya selesai dalam beberapa menit. Namun, beberapa volume penyimpanan RDS yang lebih lama memerlukan proses yang berbeda untuk memodifikasi ukuran, IOPS yang Tersedia, atau jenis penyimpanan. Hal ini melibatkan pembuatan salinan lengkap data menggunakan operasi intensif I/O yang berpotensi.

Modifikasi penyimpanan menggunakan operasi intensif I/O jika salah satu faktor berikut berlaku:

- Jenis penyimpanan sumber bersifat magnetik. Penyimpanan magnetik tidak mendukung modifikasi volume elastis.
- Instans DB RDS tidak menggunakan tata letak Amazon EBS satu atau empat volume. Anda dapat melihat jumlah volume Amazon EBS yang digunakan pada instans DB RDS Anda dengan menggunakan metrik Pemantauan yang Ditingkatkan. Untuk informasi selengkapnya, lihat Melihat metrik OS di konsol RDS.
- Ukuran target permintaan modifikasi meningkatkan penyimpanan yang dialokasikan di atas 400 GiB untuk instans RDS for MariaDB, MySQL, dan PostgreSQL, serta 200 GiB untuk RDS for Oracle. Operasi penskalaan otomatis penyimpanan memiliki efek yang sama ketika meningkatkan ukuran penyimpanan yang dialokasikan dari instans DB Anda di atas ambang batas ini.

Jika modifikasi penyimpanan Anda melibatkan operasi intensif I/O, operasi tersebut mengonsumsi sumber daya I/O dan meningkatkan beban pada instans DB Anda. Modifikasi penyimpanan dengan operasi intensif I/O yang melibatkan penyimpanan SSD Tujuan Umum (gp2) dapat mengurangi saldo kredit I/O, sehingga menghasilkan waktu konversi yang lebih lama.

Kami merekomendasikan sebagai praktik terbaik untuk menjadwalkan permintaan modifikasi penyimpanan ini di luar jam sibuk untuk membantu mengurangi waktu yang diperlukan guna menyelesaikan operasi modifikasi penyimpanan. Atau, Anda dapat membuat replika baca instans DB dan melakukan modifikasi penyimpanan pada replika baca. Kemudian, promosikan replika baca menjadi instans DB primer. Untuk informasi selengkapnya, lihat Menggunakan replika baca instans DB.

Untuk informasi selengkapnya, lihat Mengapa instans DB Amazon RDS tetap dalam status modifikasi ketika saya mencoba meningkatkan penyimpanan yang dialokasikan?

Memodifikasi pengaturan untuk penyimpanan SSD Tujuan Umum (gp3)

Anda dapat mengubah pengaturan untuk instans DB yang menggunakan penyimpanan General Purpose SSD (gp3) dengan menggunakan konsol Amazon RDS, AWS CLI atau Amazon RDS API. Tentukan jenis penyimpanan, alokasi penyimpanan, jumlah IOPS yang Tersedia, dan throughput penyimpanan yang Anda butuhkan. Meskipun Anda dapat mengurangi jumlah IOPS yang tersedia untuk instans, Anda tidak dapat mengurangi ukuran penyimpanan.

Dalam banyak kasus, penyimpanan penskalaan tidak memerlukan pemadaman. Setelah Anda mengubah IOPS penyimpanan untuk instans DB, status instans DB adalah storage-optimization. Anda dapat mengharapkan latensi yang meningkat, tetapi masih dalam kisaran milidetik satu digit, selama pengoptimalan penyimpanan. Instans DB beroperasi penuh setelah modifikasi penyimpanan.

> **Note**
>
> Anda tidak dapat melakukan modifikasi penyimpanan penuh hingga enam (6) jam setelah pengoptimalan penyimpanan selesai instans.

Untuk informasi tentang rentang penyimpanan yang dialokasikan, IOPS yang Tersedia, dan throughput penyimpanan yang tersedia untuk setiap mesin basis data, lihat Penyimpanan gp3.

Konsol

Untuk mengubah pengaturan penyimpanan untuk instans DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data.

   Untuk memfilter daftar instans DB, untuk Filter basis data masukkan string teks untuk Amazon RDS yang digunakan untuk memfilter hasil. Hanya instans DB yang namanya berisi string yang muncul.
3. Pilih instans DB dengan penyimpanan gp3 yang ingin Anda modifikasi.
4. Pilih Modifikasi.
5. Pada halaman Modifikasi Instans DB, pilih SSD Tujuan Umum (gp3) untuk Jenis penyimpanan, lalu lakukan hal berikut:

a. Untuk IOPS yang Tersedia, masukkan nilai.

   Jika nilai yang Anda tetapkan untuk Alokasi penyimpanan atau IOPS yang Tersedia berada di luar batas yang didukung oleh parameter lain, pesan peringatan akan muncul. Pesan ini memberikan rentang nilai yang diperlukan untuk parameter lainnya.

b. Untuk Throughput penyimpanan, pilih nilai.

   Jika nilai yang Anda tetapkan untuk IOPS yang Tersedia atau Throughput penyimpanan berada di luar batas yang didukung oleh parameter lain, pesan peringatan akan muncul. Pesan ini memberikan rentang nilai yang diperlukan untuk parameter lainnya.

6. Pilih Lanjutkan.
7. Pilih Langsung terapkan dalam bagian Penjadwalan modifikasi untuk menerapkan perubahan ke instans DB dengan segera. Atau pilih Terapkan pada jendela pemeliharaan terjadwal berikutnya untuk menerapkan perubahan pada jendela pemeliharaan berikutnya.
8. Tinjau parameter yang akan diubah, dan pilih Modifikasi instans DB untuk menyelesaikan modifikasi.

   Nilai baru untuk IOPS yang Tersedia muncul di kolom Status.

AWS CLI

Untuk mengubah pengaturan kinerja penyimpanan untuk instans DB, gunakan AWS CLI perintah `modify-db-instance`. Atur parameter berikut:

- `--storage-type` – Atur ke `gp3` SSD Serba Guna (gp3).
- `--allocated-storage` – Jumlah penyimpanan yang akan dialokasikan untuk instans DB, dalam gibibyte.
- `--iops` – Jumlah baru IOPS yang Tersedia untuk instans DB, dinyatakan dalam operasi I/O per detik.
- `--storage-throughput`— Throughput penyimpanan baru untuk instans DB, dinyatakan dalam MiBps.
- `--apply-immediately` – Gunakan `--apply-immediately` untuk segera menerapkan perubahan. Gunakan `--no-apply-immediately` (default) untuk menerapkan perubahan selama jendela pemeliharaan berikutnya.

API RDS

Untuk mengubah pengaturan performa penyimpanan untuk instans DB, gunakan operasi API Amazon RDS `ModifyDBInstance`. Atur parameter berikut:

- `StorageType` – Atur ke `gp3` SSD Serba Guna (gp3).
- `AllocatedStorage` – Jumlah penyimpanan yang akan dialokasikan untuk instans DB, dalam gibibyte.
- `Iops` – Rasio IOPS baru untuk instans DB, dinyatakan dalam operasi I/O per detik.
- `StorageThroughput`— Throughput penyimpanan baru untuk instans DB, dinyatakan dalam MiBps.
- `ApplyImmediately` – Atur opsi ini ke `True` untuk segera menerapkan perubahan. Atur opsi ini ke `False` (default) untuk menerapkan perubahan pada jendela pemeliharaan berikutnya.

Menggunakan volume log khusus (DLV)

Anda dapat menggunakan volume log khusus (DLV) untuk instans DB yang menggunakan penyimpanan IOPS Tertentu (PIOPS). DLV memindahkan log transaksi database PostgreSQL dan log redo MySQL/MariaDB dan log biner ke volume penyimpanan yang terpisah dari volume yang berisi tabel database. DLV membuat pencatatan log penulisan transaksi menjadi lebih efisien dan konsisten. DLV ideal untuk basis data dengan penyimpanan besar yang dialokasikan, kebutuhan I/O per detik (IOPS) tinggi, atau beban kerja yang sensitif terhadap latensi.

DLV didukung untuk penyimpanan PIOPS (io1 dan io2 Block Express) dan dibuat dengan ukuran tetap 1.000 GiB dan 3.000 IOPS yang Disediakan.

Amazon RDS mendukung DLV secara keseluruhan Wilayah AWS untuk versi berikut:

- MariaDB 10.6.7 dan versi 10 yang lebih tinggi
- MySQL 8.0.28 dan versi 8 yang lebih tinggi
- PostgreSQL 13.10 dan versi 13 yang lebih tinggi, 14.7 dan versi 14 yang lebih tinggi, serta 15.2 dan versi 15 yang lebih tinggi

RDS mendukung DLV dengan deployment Multi-AZ. Saat Anda memodifikasi atau membuat instance Multi-AZ, DLV dibuat untuk primer dan sekunder.

RDS mendukung DLV dengan replika baca. Jika instans DB primer memiliki DLV yang aktif, semua replika baca yang dibuat setelah mengaktifkan DLV juga akan memiliki DLV. Setiap replika baca yang dibuat sebelum beralih ke DLV tidak akan mengaktifkan DLV kecuali diubah secara eksplisit untuk mengaktifkannya. Sebaiknya semua replika baca yang dilampirkan ke instans primer sebelum DLV diaktifkan juga diubah secara manual untuk memiliki DLV.

> **Note**
>
> Volume log khusus direkomendasikan untuk konfigurasi basis data 5 TiB atau lebih besar.

Untuk informasi tentang rentang penyimpanan yang dialokasikan, IOPS yang Tersedia, dan throughput penyimpanan yang tersedia untuk setiap mesin basis data, lihat Penyimpanan SSD IOPS yang Tersedia.

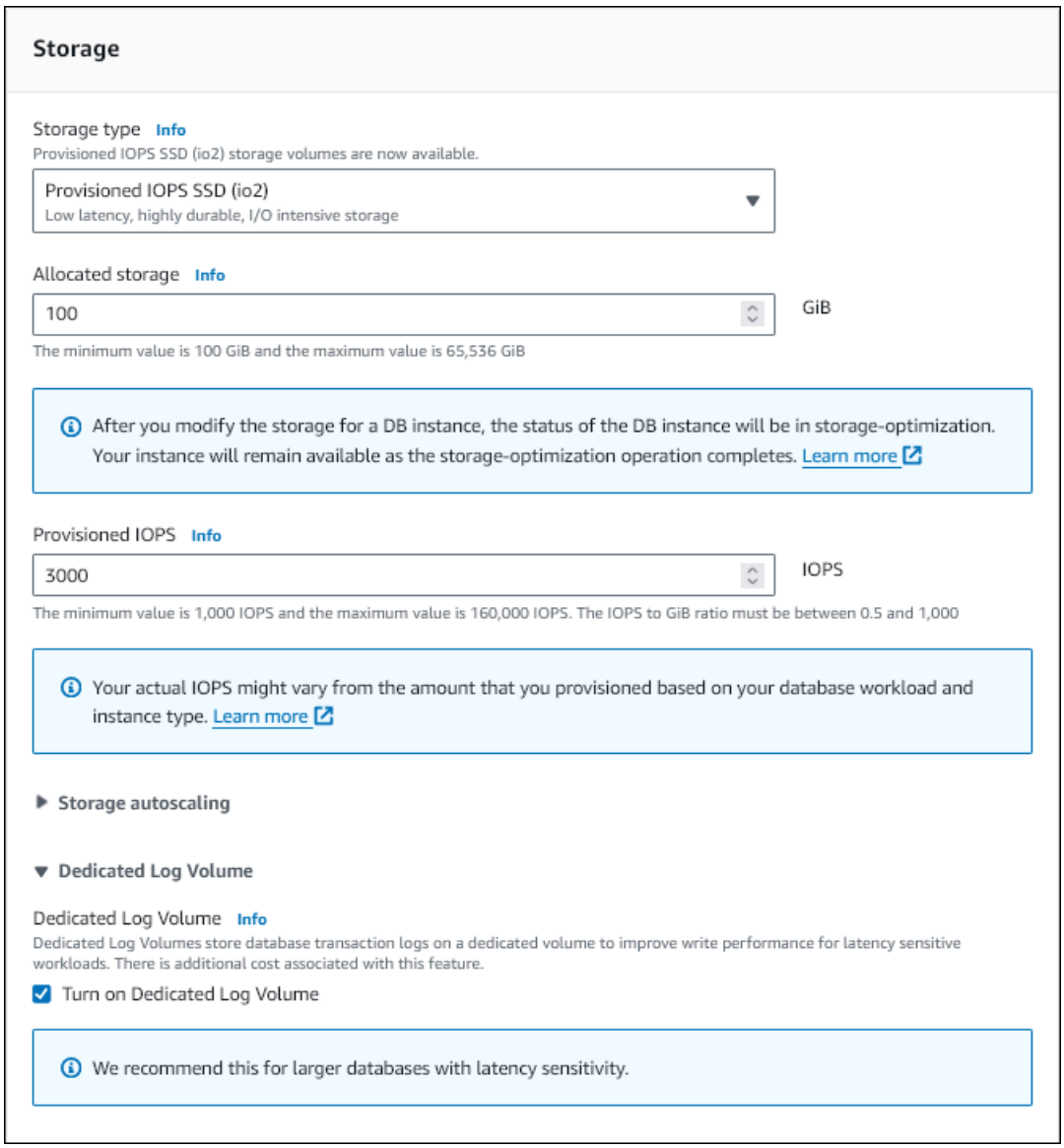
Mengaktifkan DLV saat Anda membuat instans DB

Anda dapat menggunakan AWS Management Console, AWS CLI, atau RDS API untuk membuat instans DB dengan DLV diaktifkan.

Konsol

Untuk mengaktifkan DLV pada instans DB baru

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Pilih Buat basis data.
3. Pada halaman Instans Create DB, pilih mesin DB yang mendukung DLV.
4. Untuk Penyimpanan:
    a. Pilih Provisioned IOPS SSD (io1) atau Provisioned IOPS SSD (io2).
    b. Masukkan penyimpanan yang dialokasikan dan IOPS yang disediakan yang Anda inginkan.
    c. Perluas Volume Log Khusus, lalu pilih Aktifkan Volume Log Khusus.



5. Pilih pengaturan lain sesuai kebutuhan.
6. Pilih Buat basis data.

Setelah database dibuat, nilai untuk Volume Log Khusus muncul di tab Konfigurasi halaman detail database.

CLI

Untuk mengaktifkan DLV saat Anda membuat instans DB menggunakan penyimpanan IOPS yang Disediakan, gunakan perintah. AWS CLI create-db-instance Atur parameter berikut:

- `--storage-type`— Setel ke `io1` atau `io2` untuk IOPS yang Disediakan.
- `--allocated-storage` – Jumlah penyimpanan yang akan dialokasikan untuk instans DB, dalam gibibyte.
- `--iops`— Jumlah IOPS yang Disediakan untuk instans DB, dinyatakan dalam operasi I/O per detik.
- `--dedicated-log-volume` – Atur ke `enabled` untuk menggunakan volume log khusus.

API RDS

Untuk mengaktifkan DLV saat Anda membuat instans DB menggunakan penyimpanan IOPS Terprovisi, gunakan operasi Amazon RDS API CreateDBInstance. Atur parameter berikut:

- `StorageType`— Setel ke `io1` atau `io2` untuk IOPS yang Disediakan.
- `AllocatedStorage` – Jumlah penyimpanan yang akan dialokasikan untuk instans DB, dalam gibibyte.
- `Iops`— Tingkat IOPS untuk instans DB, dinyatakan dalam operasi I/O per detik.
- `DedicatedLogVolume` – Atur ke `enabled` untuk menggunakan volume log khusus.

Mengaktifkan DLV pada instans DB yang ada

Anda dapat menggunakan AWS Management Console, AWS CLI, atau RDS API untuk memodifikasi instans DB untuk mengaktifkan DLV.

Setelah Anda memodifikasi pengaturan DLV untuk instans DB, Anda harus me-reboot instans DB.

Konsol

Untuk mengaktifkan DLV pada instans DB yang ada

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data.

   Untuk memfilter daftar instans DB, untuk Filter basis data masukkan string teks untuk Amazon RDS yang digunakan untuk memfilter hasil. Hanya instans DB yang namanya berisi string yang muncul.
3. Pilih instans DB dengan penyimpanan IOPS Tertentu yang ingin Anda modifikasi.

4. Pilih Modifikasi.
5. Pada halaman instans Modify DB:
   - Untuk Penyimpanan, perluas Volume Log Khusus, lalu pilih Aktifkan Volume Log Khusus.
6. Pilih Lanjutkan.
7. Pilih Terapkan segera untuk menerapkan perubahan ke instans DB segera. Atau pilih Terapkan pada jendela pemeliharaan terjadwal berikutnya untuk menerapkan perubahan pada jendela pemeliharaan berikutnya.
8. Tinjau parameter yang akan diubah, dan pilih Modifikasi instans DB untuk menyelesaikan modifikasi.

Nilai baru untuk Volume Log Khusus muncul di tab Konfigurasi halaman detail database.

CLI

Untuk mengaktifkan atau menonaktifkan DLV pada instans DB yang ada menggunakan penyimpanan IOPS Terketentuan, gunakan perintah. AWS CLI `modify-db-instance` Atur parameter berikut:

- `--dedicated-log-volume` – Atur ke `enabled` untuk menggunakan volume log khusus.
- `--apply-immediately` – Gunakan `--apply-immediately` untuk segera menerapkan perubahan. Gunakan `--no-apply-immediately` (default) untuk menerapkan perubahan selama jendela pemeliharaan berikutnya.

API RDS

Untuk mengaktifkan atau menonaktifkan DLV pada instans DB yang ada menggunakan penyimpanan IOPS yang Tersedia, gunakan operasi API Amazon RDS `ModifyDBInstance`. Atur parameter berikut:

- `DedicatedLogVolume` – Atur ke `enabled` untuk menggunakan volume log khusus.
- `ApplyImmediately` – Atur opsi ini ke `True` untuk segera menerapkan perubahan. Atur opsi ini ke `False` (default) untuk menerapkan perubahan pada jendela pemeliharaan berikutnya.

Menghapus instans DB

Anda dapat menghapus instans DB menggunakan AWS Management Console, AWS CLI, atau RDS API. Jika Anda ingin menghapus instans DB di klaster DB Aurora, lihat Menghapus klaster DB Aurora dan instans DB.

Topik

- Prasyarat untuk menghapus instans DB
- Pertimbangan saat menghapus instans DB
- Menghapus instans DB

Prasyarat untuk menghapus instans DB

Sebelum Anda mencoba menghapus instans DB, pastikan perlindungan penghapusan dinonaktifkan. Secara default, perlindungan penghapusan diaktifkan untuk instans DB yang dibuat dengan konsol.

Jika instans DB mengaktifkan perlindungan penghapusan, Anda dapat menonaktifkannya dengan memodifikasi pengaturan instans Anda. Pilih Ubah di halaman detail database atau panggil modify-db-instanceperintah. Operasi ini tidak menyebabkan penghentian. Untuk informasi selengkapnya, lihat Pengaturan untuk instans DB.

Pertimbangan saat menghapus instans DB

Menghapus instans DB berpengaruh pada pemulihan instans, ketersediaan cadangan, dan status replika baca. Pertimbangkan masalah berikut:

- Anda dapat memilih apakah akan membuat snapshot DB akhir. Anda memiliki opsi berikut:
  - Jika mengambil snapshot akhir, Anda dapat menggunakannya untuk memulihkan instans DB Anda yang dihapus. RDS mempertahankan snapshot akhir dan snapshot manual apa pun yang Anda ambil sebelumnya. Anda tidak dapat membuat snapshot DB akhir dari instans DB Anda jika tidak berada dalam status `Available`. Untuk informasi selengkapnya, lihat Melihat status instans DB Amazon RDS.
  - Jika Anda tidak mengambil snapshot akhir, penghapusan akan berjalan lebih cepat. Namun, Anda tidak dapat menggunakan snapshot akhir untuk memulihkan instans DB Anda. Jika nanti Anda memutuskan untuk memulihkan instans DB yang dihapus, pertahankan pencadangan otomatis atau gunakan snapshot manual sebelumnya untuk memulihkan instans DB Anda ke titik waktu snapshot.

- Anda dapat memilih apakah akan mempertahankan pencadangan otomatis. Anda memiliki opsi berikut:
  - Jika Anda mempertahankan pencadangan otomatis, RDS menyimpannya selama periode retensi yang berlaku untuk instans DB saat Anda menghapusnya. Anda dapat menggunakan pencadangan otomatis untuk memulihkan instans DB Anda selama tetapi tidak setelah periode retensi. Periode retensi berlaku terlepas dari apakah Anda membuat snapshot DB akhir. Untuk menghapus pencadangan otomatis yang dipertahankan, lihat Menghapus cadangan otomatis yang dipertahankan.
  - Pencadangan otomatis yang dipertahankan dan snapshot manual dikenakan biaya penagihan hingga dihapus. Untuk informasi selengkapnya, lihat Biaya retensi.
  - Jika Anda tidak menyimpan pencadangan otomatis, RDS menghapus pencadangan otomatis yang berada di Wilayah AWS yang sama dengan instans DB Anda. Anda tidak dapat memulihkan pencadangan ini. Jika pencadangan otomatis Anda telah direplikasi ke Wilayah AWS lainnya, RDS menyimpannya meskipun Anda tidak memilih untuk mempertahankan pencadangan otomatis. Untuk informasi selengkapnya, lihat Mereplikasi backup otomatis ke yang lain Wilayah AWS.

    > **Note**
    >
    > Biasanya, jika Anda membuat snapshot DB akhir, Anda tidak perlu menyimpan pencadangan otomatis.

- Saat Anda menghapus instans DB Anda, RDS tidak menghapus snapshot DB manual. Untuk informasi selengkapnya, lihat Membuat snapshot DB untuk instans DB Single-AZ.
- Jika Anda ingin menghapus semua sumber daya RDS, perhatikan bahwa sumber daya berikut dikenakan biaya penagihan:
  - Instans DB
  - Snapshot DB
  - Klaster DB

  Jika Anda membeli instans cadangan, maka mereka ditagih sesuai dengan kontrak yang Anda setujui ketika Anda membeli instans. Untuk informasi selengkapnya, lihat Instans DB terpesan untuk Amazon RDS. Anda bisa mendapatkan informasi penagihan untuk semua sumber daya AWS Anda menggunakan AWS Cost Explorer. Untuk mengetahui informasi selengkapnya, lihat Menganalisis biaya Anda dengan AWS Cost Explorer.

- Jika Anda menghapus instans DB yang memiliki replika baca di Wilayah AWS yang sama, setiap replika baca otomatis dipromosikan menjadi instans DB mandiri. Untuk informasi selengkapnya, lihat Mempromosikan replika baca menjadi instans DB mandiri. Jika instans DB Anda memiliki replika baca di Wilayah AWS yang berbeda, lihat Pertimbangan replikasi lintas Wilayah untuk mengetahui informasi terkait penghapusan instans DB sumber untuk replika baca lintas Wilayah.
- Jika status untuk instans DB adalah `deleting`, nilai sertifikat CA-nya tidak muncul di konsol RDS atau di output untuk perintah AWS CLI atau operasi RDS API. Untuk informasi selengkapnya tentang sertifikat CA, lihat .
- Waktu yang diperlukan untuk menghapus instans DB dapat bervariasi bergantung pada periode retensi cadangan (yaitu, berapa banyak cadangan yang harus dihapus), berapa banyak data yang dihapus, dan apakah snapshot akhir diambil.

Menghapus instans DB

Anda dapat menghapus instans DB menggunakan AWS Management Console, AWS CLI, atau RDS API. Anda harus melakukan tindakan berikut:

- Memberikan nama instans DB
- Mengaktifkan atau menonaktifkan opsi untuk mengambil snapshot DB akhir dari instans
- Mengaktifkan atau menonaktifkan opsi untuk mempertahankan pencadangan otomatis

> **Note**
>
> Anda tidak dapat menghapus klaster DB Multi-AZ saat perlindungan penghapusan diaktifkan. Untuk informasi selengkapnya, lihat Prasyarat untuk menghapus instans DB.

Konsol

Untuk menghapus instans DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data, lalu pilih instans DB yang ingin dihapus.
3. Untuk Tindakan, pilih Hapus.
4. Untuk membuat snapshot DB akhir untuk instans DB, pilih Buat snapshot akhir?.

5. Jika Anda memilih untuk membuat snapshot akhir, masukkan Nama snapshot akhir.
6. Untuk menyimpan pencadangan otomatis, pilih Pertahankan cadangan otomatis.
7. Masukkan **`delete me`** di kotak teks.
8. Pilih Hapus.

AWS CLI

Untuk menemukan ID instans dari instans DB di akun Anda, panggil describe-db-instancesperintah:

```
aws rds describe-db-instances --query 'DBInstances[*].[DBInstanceIdentifier]' --output
 text
```

Untuk menghapus instans DB dengan menggunakanAWS CLI, panggil delete-db-instanceperintah dengan opsi berikut:

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

Example Dengan snapshot akhir dan tanpa pencadangan otomatis yang dipertahankan

Untuk Linux, macOS, atau Unix:

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

Untuk Windows:

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

Example Dengan pencadangan otomatis yang dipertahankan dan tanpa snapshot akhir

Untuk Linux, macOS, atau Unix:

```
aws rds delete-db-instance \
```

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

Untuk Windows:

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

RDS API

Untuk menghapus instans DB dengan menggunakan Amazon RDS API, panggil operasi `DeleteDBInstance` dengan parameter berikut ini:

- `DBInstanceIdentifier`
- `FinalDBSnapshotIdentifier` atau `SkipFinalSnapshot`

Mengonfigurasi dan mengelola deployment Multi-AZ

Deployment Multi-AZ dapat memiliki satu atau dua instans DB siaga. Ketika deployment memiliki satu instans DB siaga, itu disebut deployment instans DB Multi-AZ. Deployment instans DB Multi-AZ memiliki satu instans DB siaga yang menyediakan dukungan failover, tetapi tidak melayani lalu lintas baca. Ketika deployment memiliki dua instans DB siaga, itu disebut deployment klaster DB Multi-AZ. Deployment klaster DB Multi-AZ memiliki instans DB siaga yang menyediakan dukungan failover dan juga dapat melayani lalu lintas baca.

Anda dapat menggunakan AWS Management Console untuk menentukan apakah deployment Multi-AZ merupakan deployment instans DB Multi-AZ atau deployment klaster DB Multi-AZ. Di panel navigasi, pilih Basis Data, lalu pilih Pengidentifikasi DB.

- Deployment instans DB Multi-AZ memiliki karakteristik sebagai berikut:
  - Hanya ada satu baris untuk instans DB.
  - Nilai Peran adalah Instans atau Primer.
  - Nilai Multi-AZ adalah Ya.
- Deployment klaster DB Multi-AZ memiliki karakteristik sebagai berikut:
  - Ada baris tingkat cluster dengan tiga baris instasn DB di bawahnya.
  - Untuk baris tingkat klaster, nilai Peran adalah klaster DB Multi-AZ.
  - Untuk setiap baris tingkat instans, nilai Peran adalah Instans penulis atau Instas pembaca.
  - Untuk setiap baris tingkat instans, nilai Multi-AZ adalah 3 Zona.

Topik

Selain itu, topik berikut berlaku untuk instans DB dan klaster DB multi-AZ:

- the section called “Memberi tag pada sumber daya RDS”
- the section called “Bekerja dengan ARN”
- the section called “Menggunakan penyimpanan”
- the section called “Memelihara instans DB”

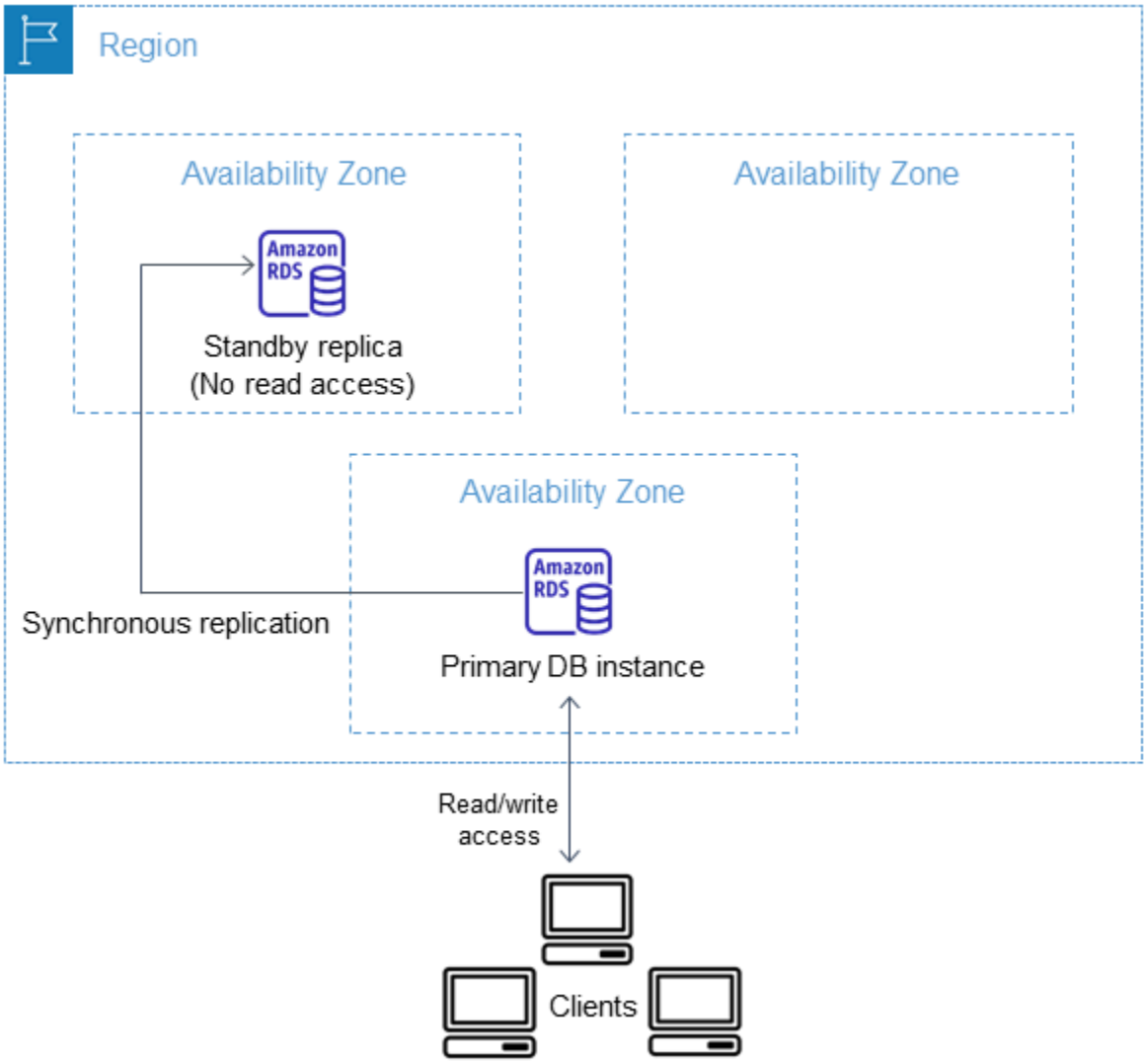
- the section called “Meng-upgrade versi mesin”

Deployment instans DB Multi-AZ

Amazon RDS menyediakan ketersediaan tinggi dan dukungan failover untuk instans DB menggunakan deployment Multi-AZ dengan instans DB siaga tunggal. Jenis deployment ini disebut deployment instans DB Multi-AZ. Amazon RDS menggunakan berbagai teknologi untuk memberikan dukungan failover ini. Deployment Multi-AZ untuk instans DB MariaDB, MySQL, Oracle, PostgreSQL, dan RDS Custom for SQL Server menggunakan teknologi failover Amazon. Instans DB Microsoft SQL Server menggunakan SQL Server Database Mirroring (DBM) atau Always On Availability Groups (AG). Untuk informasi tentang dukungan versi SQL Server untuk Multi-AZ, lihat Deployment Multi-AZ untuk Amazon RDS for Microsoft SQL Server. Untuk informasi tentang penggunaan RDS Custom for SQL Server untuk Multi-AZ, lihat Mengelola deployment Multi-AZ untuk RDS Custom for SQL Server.

Dalam deployment instans DB Multi-AZ, Amazon RDS akan otomatis menyediakan dan mempertahankan replika siaga yang sinkron di Zona Ketersediaan yang berbeda. Instans DB primer direplikasi secara sinkron di seluruh Zona Ketersediaan ke replika siaga untuk memberikan redundansi data dan meminimalkan lonjakan latensi selama pencadangan sistem. Menjalankan instans DB dengan ketersediaan tinggi dapat meningkatkan ketersediaan selama pemeliharaan sistem terencana. Hal ini juga dapat membantu melindungi basis data Anda terhadap kegagalan instans DB dan gangguan Zona Ketersediaan. Untuk informasi selengkapnya tentang Zona Ketersediaan, lihat Wilayah, Zona Ketersediaan, dan Zona Lokal.

> **Note**
>
> Opsi ketersediaan tinggi bukanlah solusi penskalaan untuk skenario baca-saja. Anda tidak dapat menggunakan replika siaga untuk menyajikan lalu lintas baca. Untuk menyajikan lalu lintas baca-saja, gunakan klaster DB Multi-AZ atau replika baca. Untuk informasi selengkapnya tentang klaster DB Multi-AZ, lihat Deployment klaster basis data Multi-AZ. Untuk informasi selengkapnya tentang replika baca, lihat Menggunakan replika baca instans DB.



Dengan menggunakan konsol RDS, Anda dapat membuat deployment instans DB Multi-AZ hanya dengan menentukan Multi-AZ saat membuat instans DB. Anda dapat menggunakan konsol untuk mengonversi instans DB yang ada ke deployment instans DB Multi-AZ dengan mengubah instans DB dan menentukan opsi Multi-AZ. Anda juga dapat menentukan penerapan instans DB multi-AZ dengan atau AWS CLI Amazon RDS API. Gunakan perintah create-db-instanceatau modify-db-instanceCLI, atau operasi CreateDBInstance atau ModifydBInstance API.

Konsol RDS menunjukkan Zona Ketersediaan replika siaga (disebut AZ sekunder). Anda juga dapat menggunakan perintah describe-db-instancesCLI atau operasi API DescribedBInstances untuk menemukan AZ sekunder.

Instans DB yang menggunakan deployment instans DB Multi-AZ dapat meningkatkan latensi tulis dan commit dibandingkan dengan deployment Single-AZ. Hal ini karena replikasi data sinkron yang terjadi. Anda mungkin mengalami perubahan latensi jika penerapan Anda gagal ke replika siaga,

meskipun direkayasa dengan konektivitas jaringan latensi rendah AWS antara Availability Zones. Untuk beban kerja produksi, sebaiknya Anda menggunakan IOPS yang Tersedia (operasi input/output per detik) untuk performa yang cepat dan konsisten. Untuk informasi selengkapnya tentang kelas instans DB, lihat Kelas instans DB .

Mengubah instans DB menjadi deployment instans DB Multi-AZ

Jika Anda memiliki instans DB dalam deployment Single-AZ dan mengubahnya menjadi deployment instans DB Multi-AZ (untuk mesin selain Amazon Aurora), Amazon RDS akan melakukan beberapa tindakan:

1. Mengambil snapshot volume Amazon Elastic Block Store (EBS) instans DB utama.
2. Membuat volume baru untuk replika siaga dari snapshot. Volume tersebut diinisialisasi di latar belakang, dan performa volume maksimum akan tercapai setelah data sepenuhnya diinisialisasi.
3. Mengaktifkan replikasi tingkat blok sinkron antara volume replika utama dan siaga.

> **Important**
>
> Penggunaan snapshot untuk membuat instans siaga dapat menghindari waktu henti saat mengonversi dari Single-AZ ke Multi-AZ, tetapi Anda dapat merasakan dampak performa selama dan setelah mengonversi ke Multi-AZ. Hal ini dapat memberikan dampak yang signifikan terhadap beban kerja yang sensitif terhadap latensi tulis.
> Meskipun kemampuan ini memungkinkan volume besar dipulihkan dengan cepat dari snapshot, peningkatan latensi operasi I/O yang signifikan dapat terjadi karena adanya replikasi yang sinkron. Latensi ini dapat memengaruhi performa basis data Anda. Praktik terbaik yang sangat direkomendasikan adalah tidak melakukan konversi multi-AZ pada instans DB produksi.
> Untuk menghindari dampak performa pada instans DB yang saat ini melayani beban kerja sensitif, buat replika baca dan aktifkan pencadangan pada replika baca. Konversikan replika baca ke Multi-AZ, dan jalankan kueri yang memuat data ke dalam volume replika baca (pada kedua AZ). Kemudian, promosikan replika baca menjadi instans DB primer. Untuk informasi selengkapnya, lihat Menggunakan replika baca instans DB.

Ada dua cara untuk mengubah instans DB menjadi deployment instans DB Multi-AZ:

Topik

- Mengonversi menjadi deployment instans DB Multi-AZ dengan konsol RDS
- Mengubah instans DB menjadi deployment instans DB Multi-AZ

Mengonversi menjadi deployment instans DB Multi-AZ dengan konsol RDS

Anda dapat menggunakan konsol RDS untuk mengonversi instans DB menjadi deployment instans DB Multi-AZ.

Anda hanya dapat menggunakan konsol tersebut untuk menyelesaikan konversi. Untuk menggunakan AWS CLI atau RDS API, ikuti petunjuk diMengubah instans DB menjadi deployment instans DB Multi-AZ.

Untuk mengonversi menjadi deployment instans DB Multi-AZ dengan konsol RDS

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data, lalu pilih instans DB yang ingin diubah.
3. Dari Tindakan, pilih Konversikan menjadi deployment Multi-AZ.
4. Di halaman konfirmasi, pilih Terapkan langsung untuk langsung menerapkan perubahan. Memilih opsi ini tidak akan menyebabkan waktu henti, tetapi ada kemungkinan dampak performa. Atau, Anda dapat memilih untuk menerapkan pembaruan pada periode pemeliharaan berikutnya. Untuk informasi selengkapnya, lihat Menggunakan pengaturan Terapkan Segera.
5. Pilih Konversikan menjadi Multi-AZ.

Mengubah instans DB menjadi deployment instans DB Multi-AZ

Anda dapat mengubah instans DB menjadi deployment instans DB MultiAZ dengan cara berikut:

- Dengan menggunakan konsol RDS, ubah instans DB, dan tetapkan Deployment Multi-AZ ke Ya.
- Menggunakan AWS CLI, panggil modify-db-instanceperintah, dan atur `--multi-az` opsi.
- Dengan menggunakan API RDS, panggil operasi ModifyDBInstance, dan tetapkan parameter `MultiAZ` ke `true`.

Untuk informasi tentang cara mengubah instans DB, lihat Memodifikasi instans DB Amazon RDS. Setelah perubahan selesai, Amazon RDS memicu peristiwa (RDS-EVENT-0025) yang menunjukkan

bahwa proses telah selesai. Anda dapat memantau peristiwa Amazon RDS. Untuk informasi selengkapnya tentang peristiwa, lihat Menggunakan pemberitahuan peristiwa Amazon RDS.

Proses failover untuk Amazon RDS

Jika penghentian instans DB terencana atau tidak terencana terjadi karena cacat infrastruktur, Amazon RDS akan otomatis beralih ke replika siaga di Zona Ketersediaan lain jika Anda telah mengaktifkan Multi-AZ. Durasi penyelesaian failover bergantung pada aktivitas basis data dan kondisi lain pada saat instans DB primer tidak tersedia. Durasi failover biasanya antara 60–120 detik. Namun, transaksi besar atau proses pemulihan yang panjang dapat meningkatkan waktu failover. Setelah failover selesai, perlu waktu tambahan agar konsol RDS dapat mencerminkan Zona Ketersediaan baru.

> **Note**
>
> Anda dapat memaksakan failover secara manual saat me-reboot instans DB. Untuk informasi selengkapnya, lihat Mem-boot ulang instans DB.

Amazon RDS menangani failover secara otomatis sehingga Anda dapat melanjutkan operasi basis data secepat mungkin tanpa intervensi administratif. Instans DB primer otomatis beralih ke replika siaga jika salah satu dari kondisi yang dijelaskan dalam tabel berikut terjadi. Anda dapat melihat alasan failover ini di log peristiwa.

Alasan failover	Deskripsi
Sistem operasi yang mendasari instans basis data RDS sedang di-patch dalam operasi offline.	Failover dipicu selama periode pemeliharaan untuk patch OS atau pembaruan keamanan. Untuk informasi selengkapnya, lihat Memelihara instans DB.
Host primer instans Multi-AZ RDS tidak sehat.	Deployment instans DB Multi-AZ mendeteksi instans DB primer mengalami gangguan dan failover.

Alasan failover	Deskripsi
Host primer instans Multi-AZ RDS tidak terjangkau karena kehilangan konektivitas jaringan.	Pemantauan RDS mendeteksi kegagalan menjangkau jaringan ke instans DB primer dan telah memicu failover.
Instans RDS diubah oleh pelanggan.	Perubahan instans DB RDS memicu failover. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Alasan failover	Deskripsi
Instans primer Multi-AZ RDS sibuk dan tidak responsif.	Instans DB primer tidak responsif. Sebaiknya Anda melakukan tindakan berikut: • Periksa peristiwa dan CloudWatch log untuk penggunaan CPU, memori, atau ruang swap yang berlebihan. Untuk informasi lebih lanjut, lihat Menggunakan pemberitahuan peristiwa Amazon RDS dan Membuat aturan yang memicu peristiwa Amazon RDS. • Evaluasi beban kerja Anda untuk menentukan apakah Anda menggunakan kelas instans DB yang sesuai. Untuk informasi selengkapnya, lihat Kelas instans DB . • Gunakan Pemantauan yang Disempurnakan untuk metrik sistem operasi waktu nyata. Untuk informasi selengkapnya, lihat Memantau metrik OS dengan Pemantauan yang Disempurnakan. • Gunakan Wawasan Performa untuk membantu menganalisis masalah yang memengaruhi performa instans DB Anda. Untuk informasi selengkapnya, lihat Memantau muatan DB dengan Wawasan Performa di Amazon RDS. Untuk informasi selengkapnya terkait rekomendasi ini, lihat Ikhtisar metrik pemantauan di Amazon RDS dan Praktik terbaik untuk Amazon RDS.

Alasan failover	Deskripsi
Volume penyimpanan yang mendasari host primer instans Multi-AZ RDS mengalami kegagalan.	Deployment instans DB Multi-AZ mendeteksi masalah penyimpanan pada instans DB primer dan mengalami failover.
Pengguna meminta failover instans DB.	Anda me-reboot instans DB dan memilih Boot ulang dengan failover. Untuk informasi selengkapnya, lihat Mem-boot ulang instans DB.

Untuk mengetahui apakah instans DB Multi-AZ mengalami failover, Anda dapat melakukan tindakan berikut:

- Siapkan langganan peristiwa DB untuk memberi tahu Anda melalui email atau SMS bahwa failover telah dimulai. Untuk informasi selengkapnya tentang peristiwa, lihat Menggunakan pemberitahuan peristiwa Amazon RDS.
- Lihat peristiwa DB Anda menggunakan operasi API konsol RDS.
- Lihat kondisi deployment instans DB Multi-AZ menggunakan operasi API konsol RDS.

Untuk informasi tentang cara merespons failover, mengurangi waktu pemulihan, dan praktik terbaik lainnya untuk Amazon RDS, lihat Praktik terbaik untuk Amazon RDS.

Mengatur TTL JVM untuk pencarian nama DNS

Mekanisme failover otomatis mengubah catatan Sistem Nama Domain (DNS) instans DB menjadi titik ke instans DB siaga. Oleh karena itu, Anda perlu membuat kembali koneksi yang ada ke instans DB Anda. Di lingkungan mesin virtual Java (JVM), karena cara kerja mekanisme pengisian cache DNS Java, Anda mungkin perlu mengonfigurasikan ulang setelan JVM.

JVM menyimpan pencarian nama DNS di cache. Ketika JVM menyelesaikan nama host ke alamat IP, itu cache alamat IP untuk jangka waktu tertentu, yang dikenal sebagai (TTL). time-to-live

Karena AWS sumber daya menggunakan entri nama DNS yang terkadang berubah, kami sarankan Anda mengonfigurasi JVM Anda dengan nilai TTL tidak lebih dari 60 detik. Hal ini memastikan bahwa ketika alamat IP sumber daya berubah, aplikasi Anda dapat menerima dan menggunakan alamat IP baru sumber daya itu dengan mengueri ulang DNS.

Pada beberapa konfigurasi Java, TTL bawaan JVM diatur sedemikian rupa sehingga tidak pernah menyegarkan entri DNS hingga JVM dimulai ulang. Jadi, jika alamat IP untuk AWS sumber daya berubah saat aplikasi Anda masih berjalan, itu tidak dapat menggunakan sumber daya itu sampai Anda secara manual me-restart JVM dan informasi IP cache di-refresh. Dalam kasus ini, penting untuk mengatur TTL JVM untuk menyegarkan informasi IP yang tersimpan dalam cache secara berkala.

Anda bisa mendapatkan TTL default JVM dengan mengambil nilai properti: `networkaddress.cache.ttl`

```
String ttl = java.security.Security.getProperty("networkaddress.cache.ttl");
```

> **Note**
>
> TTL default dapat bervariasi berdasarkan versi JVM Anda dan apakah manajer keamanan telah diinstal. Banyak JVM menyediakan TTL bawaan yang kurang dari 60 detik. Jika Anda menggunakan JVM seperti itu dan tidak menggunakan manajer keamanan, Anda dapat mengabaikan bagian selebihnya topik ini. Lihat informasi yang lebih lengkap tentang manajer keamanan di Oracle di Manajer keamanan dalam dokumentasi Oracle.

Untuk mengubah TTL JVM, atur nilai properti `networkaddress.cache.ttl`. Gunakan salah satu metode berikut, sesuai dengan kebutuhan Anda:

- Untuk mengatur nilai properti secara global bagi semua aplikasi yang menggunakan JVM, atur `networkaddress.cache.ttl` dalam file `$JAVA_HOME/jre/lib/security/java.security`.

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

- Untuk menetapkan properti secara lokal hanya bagi aplikasi Anda, tetapkan `networkaddress.cache.ttl` dalam kode inisialisasi aplikasi Anda sebelum koneksi jaringan dibuat.

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

Deployment klaster basis data Multi-AZ

Penerapan klaster DB multi-AZ adalah mode penyebaran Amazon RDS semisinkron dan ketersediaan tinggi dengan dua instans replika DB yang dapat dibaca. Klaster basis data Multi-AZ memiliki instans basis data penulis dan dua instans basis data pembaca di tiga Zona Ketersediaan terpisah dengan Wilayah AWS yang sama. Klaster basis data Multi-AZ menyediakan ketersediaan tinggi, kapasitas yang meningkat untuk beban kerja baca, dan latensi tulis yang lebih rendah jika dibandingkan dengan deployment instans basis data Multi-AZ.

Anda dapat mengimpor data dari basis data on-premise ke klaster basis data Multi-AZ dengan mengikuti petunjuk di Mengimpor data ke basis data Amazon RDS MariaDB atau MySQL dengan lebih sedikit waktu henti.

Anda dapat membeli instans basis data cadangan untuk klaster basis data Multi-AZ. Untuk informasi selengkapnya, lihat Instans DB terpesan untuk klaster DB Multi-AZ.

Ketersediaan dan dukungan fitur bervariasi di seluruh versi spesifik dari setiap mesin basis data, dan di seluruh Wilayah AWS. Lihat informasi yang lebih lengkap tentang versi dan ketersediaan Wilayah Amazon RDS dengan klaster basis data Multi-AZ di Klaster DB Multi-AZ.

Topik

- Membut ulang klaster basis data Multi-AZ dan instans basis data pembaca
- Bekerja dengan replika baca klaster DB multi-AZ
- Menggunakan replikasi logis PostgreSQL dengan klaster DB Multi-AZ
- Menghapus klaster DB Multi-AZ
- Keterbatasan cluster DB multi-AZ

> **Important**
>
> Klaster basis data Multi-AZ tidak sama dengan klaster basis data Aurora. Lihat informasi tentang klaster basis data Aurora di Panduan Pengguna Amazon Aurora.

Ketersediaan kelas instans untuk cluster DB multi-AZ

Penerapan cluster DB multi-AZ didukung untuk kelas instans DB berikut:`db.m5d`,`db.m6gd`,`m6id`,,`db.m6idn`, `db.r5ddb.r6gd`, dan `db.x2iedndb.r6id`, dan`db.r6idn`. `db.c6gd`

> **Note**
>
> Kelas instance c6gd adalah satu-satunya yang mendukung ukuran instance. `medium`

Lihat informasi lebih lanjut tentang kelas instans basis data di the section called “Kelas instans DB”.

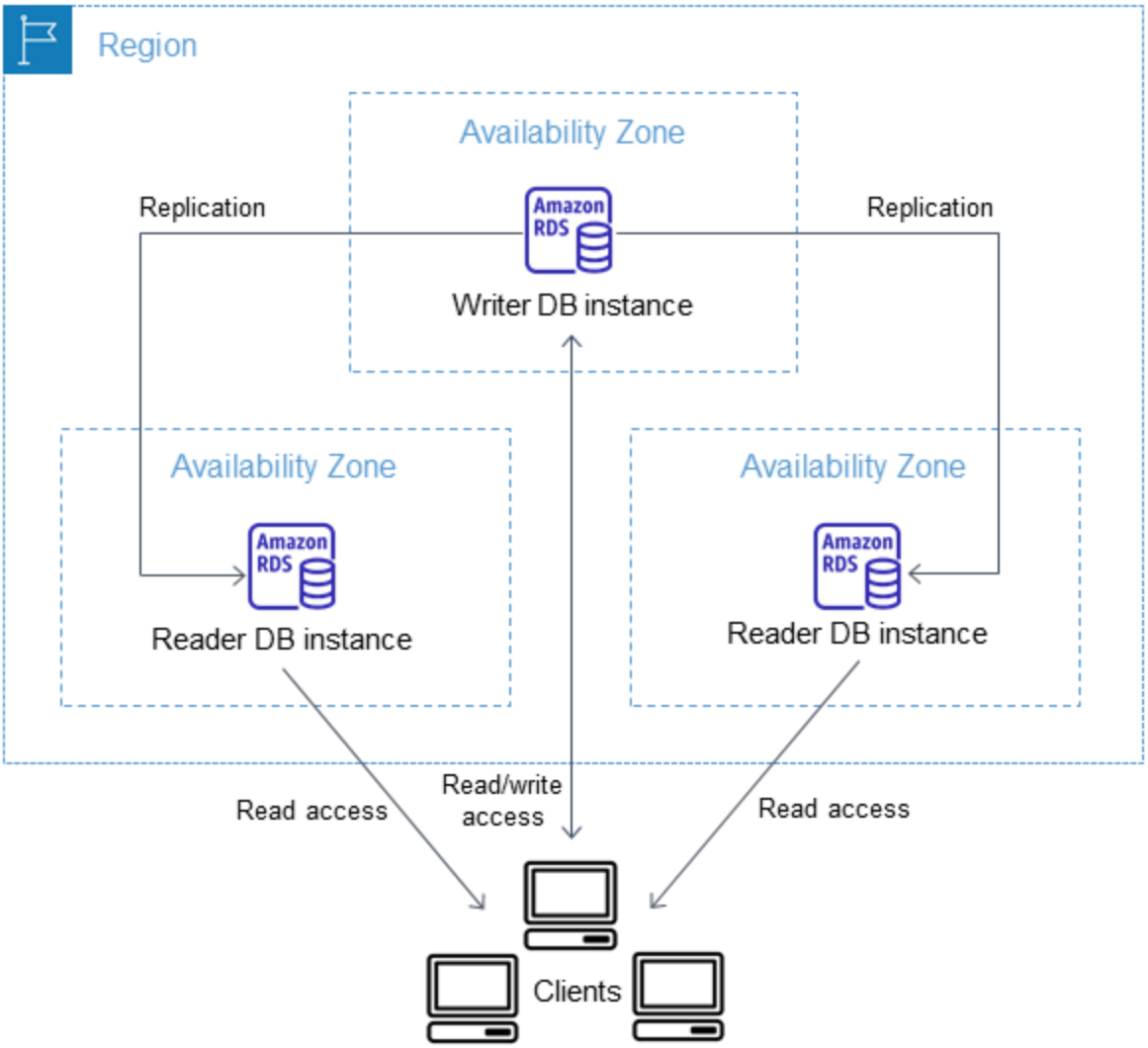
Ikhtisar klaster basis data Multi-AZ

Dengan klaster basis data Multi-AZ, Amazon RDS mereplikasi data dari instans basis data penulis ke kedua instans basis data pembaca dengan menggunakan kemampuan replikasi asli mesin basis data. Ketika perubahan dibuat pada instans basis data penulis, perubahan itu dikirim ke setiap instans basis data pembaca.

Deployment klaster basis data Multi-AZ menggunakan replikasi semisinkron, yang memerlukan pengakuan dari setidaknya satu instans basis data pembaca agar perubahan dapat di-commit. Deployment tidak memerlukan pengakuan bahwa peristiwa telah dieksekusi dan di-commit sepenuhnya pada semua replika.

Instans basis data pembaca bertindak sebagai target failover otomatis dan juga melayani lalu lintas baca untuk meningkatkan throughput baca aplikasi. Jika pemadaman terjadi pada instans basis data penulis, RDS mengelola failover ke salah satu instans basis data pembaca. RDS melakukan pemindahan berdasarkan instans basis data pembaca yang memiliki catatan perubahan terbaru.

Diagram berikut menunjukkan sebuah klaster basis data Multi-AZ.



Klaster basis data Multi-AZ biasanya memiliki latensi penulisan yang lebih rendah jika dibandingkan dengan deployment instans basis data Multi-AZ. Klaster itu juga memungkinkan beban kerja hanya baca berjalan pada instans basis data pembaca. Konsol RDS menunjukkan Zona Ketersediaan instans basis data penulis dan Zona Ketersediaan instans basis data pembaca. Anda juga dapat menggunakan perintah describe-db-clustersCLI atau operasi API DescribedBClusters untuk menemukan informasi ini.

> **Important**
>
> Untuk mencegah kesalahan replikasi di RDS bagi klaster basis data Multi-AZ MySQL, kami sangat menyarankan agar semua tabel memiliki kunci primer.

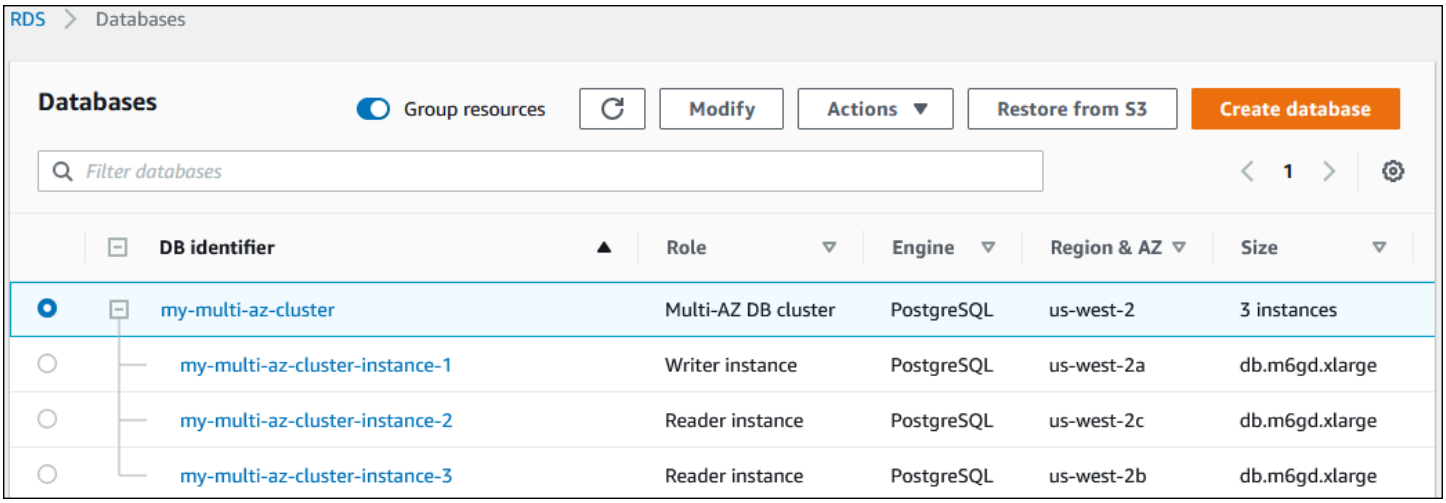
Mengelola cluster DB multi-AZ dengan AWS Management Console

Anda dapat mengelola klaster basis data Multi-AZ dengan konsol.

Untuk mengelola klaster basis data Multi-AZ dengan konsol

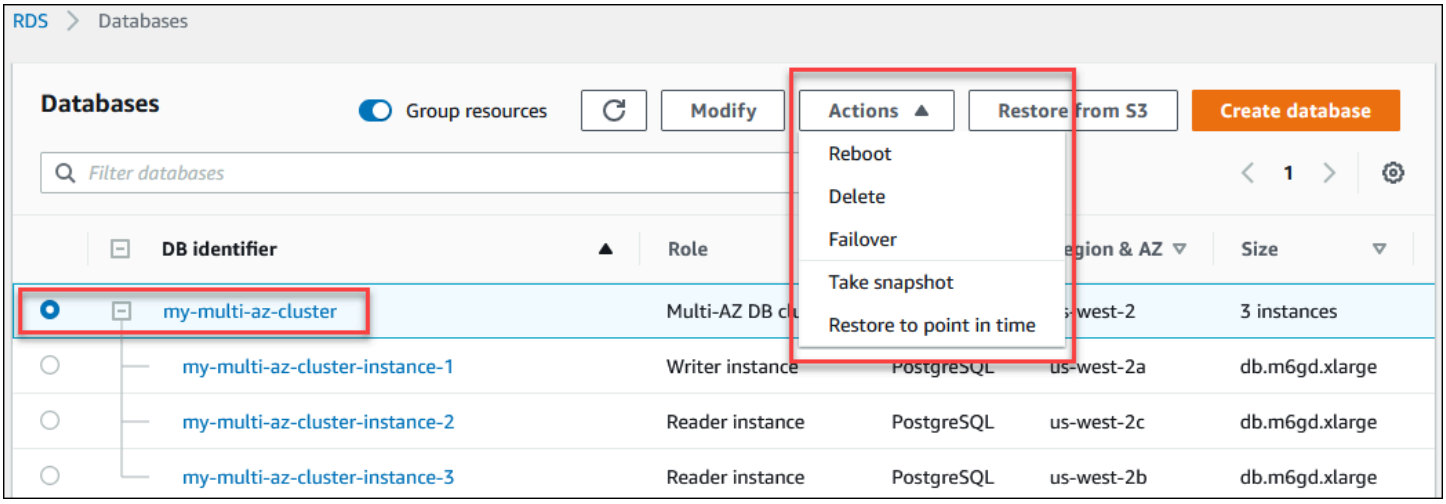
1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data, lalu pilih klaster basis data Multi-AZ yang ingin Anda kelola.

Gambar berikut menunjukkan klaster basis data Multi-AZ di konsol.

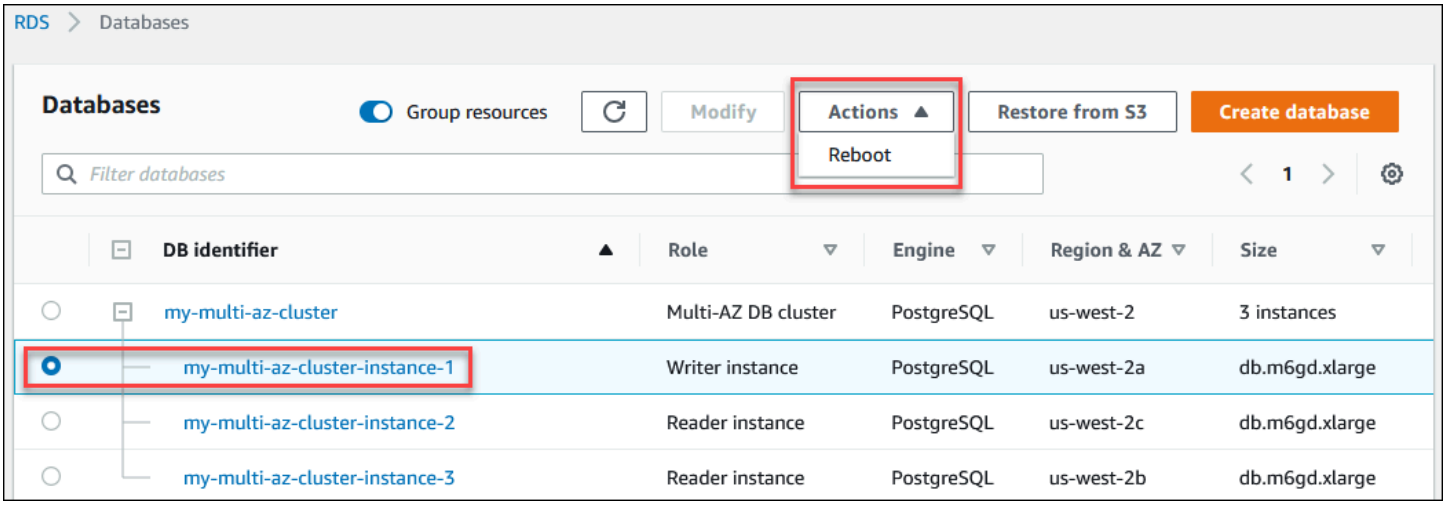


Tindakan-tindakan yang tersedia di menu Tindakan bergantung pada apakah yang dipilih klaster basis data Multi-AZ atau instans basis data di klaster itu.

Pilih klaster basis data Multi-AZ untuk melihat detail klaster dan melakukan tindakan di tingkat klaster.



Pilih instans basis data di klaster basis data Multi-AZ untuk melihat detail instans basis data dan melakukan tindakan pada tingkat instans basis data.



Bekerja dengan grup parameter untuk klaster basis data Multi-AZ

Di klaster basis data Multi-AZ, sebuah grup parameter klaster basis data bertindak sebagai kontainer untuk nilai-nilai konfigurasi mesin yang diterapkan pada setiap instans basis data di klaster basis data Multi-AZ.

Di klaster basis data Multi-AZ, grup parameter basis data diatur ke grup parameter basis data default untuk mesin basis data dan versi mesin basis datanya. Setelan dalam grup parameter klaster basis data digunakan untuk semua instans basis data di klaster.

Lihat informasi tentang grup parameter di Bekerja dengan grup parameter.

Memutakhirkan versi mesin klaster basis data Multi-AZ

Amazon RDS menyediakan versi-versi baru untuk setiap mesin basis data yang didukung sehingga Anda dapat selalu memperbarui klaster basis data Multi-AZ Anda. Ketika Amazon RDS mendukung versi baru mesin basis data, Anda dapat memilih cara dan waktu harus memutakhirkan klaster basis data Multi-AZ Anda.

Ada dua jenis pemutakhiran yang dapat Anda lakukan:

Pemutakhiran versi utama

Pemutakhiran versi mesin utama dapat membawa perubahan yang tidak kompatibel dengan aplikasi yang ada. Saat Anda memulai pemutakhiran versi utama, Amazon RDS turut memutakhirkan instans pembaca dan penulis. Oleh karena itu, klaster basis data Anda dapat tidak tersedia hingga pemutakhiran selesai.

Pemutakhiran versi kecil

Pemutakhiran versi kecil hanya mencakup perubahan yang kompatibel surut dengan aplikasi yang ada. Saat Anda memulai pemutakhiran versi kecil, Amazon RDS memutakhirkan dahulu instans-instans basis data pembaca satu per satu. Kemudian, salah satu instans basis data pembaca beralih menjadi instans basis data penulis baru. Amazon RDS lalu memutakhirkan instans penulis lama (yang kini menjadi instans pembaca).

Waktu henti selama pemutakhiran dibatasi pada waktu yang dibutuhkan salah satu instans basis data pembaca untuk menjadi instans basis data penulis baru. Waktu henti ini bertindak seperti failover otomatis. Untuk informasi selengkapnya, lihat the section called “Proses failover untuk klaster basis data Multi-AZ”. Perhatikan bahwa kelambatan replika klaster basis data Multi-AZ Anda dapat memengaruhi waktu henti. Untuk informasi selengkapnya, lihat the section called “Kelambatan replika dan klaster basis data Multi-AZ”.

Untuk replika baca klaster basis data Multi-AZ RDS for PostgreSQL, Amazon RDS memutakhirkan instans-instans anggota klaster satu per satu. Peran-peran klaster pembaca dan penulis tidak bertukar selama pemutakhiran. Oleh karena itu, klaster basis data Anda mungkin mengalami waktu henti saat Amazon RDS memutakhirkan instans penulis klaster.

> **Note**
>
> Waktu henti untuk pemutakhiran versi kecil klaster basis data Multi-AZ biasanya 35 detik. Saat digunakan dengan Proksi RDS, Anda dapat mengurangi waktu henti ke satu detik atau kurang. Untuk informasi selengkapnya, lihat Menggunakan Proksi RDS. Sebagai

alternatif, Anda dapat menggunakan proxy database open source seperti ProxySQL, PgBouncer, atau Driver AWS JDBC untuk MySQL.

Saat ini, Amazon RDS mendukung pemutakhiran versi utama hanya untuk klaster basis data Multi-AZ RDS for PostgreSQL. Amazon RDS mendukung peningkatan versi kecil untuk semua mesin basis data yang mendukung klaster basis data Multi-AZ.

Amazon RDS tidak memutakhirkan secara otomatis replika baca klaster basis data Multi-AZ. Untuk pemutakhiran versi kecil, Anda harus memutakhirkan dahulu semua replika baca secara manual, lalu memutakhirkan klaster. Jika tidak, pemutakhiran diblokir. Saat Anda melakukan pemutakhiran versi utama sebuah klaster, keadaan replikasi semua replika baca berubah ke dihentikan. Anda harus menghapus dan membuat ulang replika baca setelah pemutakhiran selesai. Untuk informasi selengkapnya, lihat the section called “Memantau replikasi baca”.

Proses untuk memutakhirkan versi mesin klaster basis data Multi-AZ sama dengan proses untuk memutakhirkan versi mesin instans basis data. Untuk petunjuk, lihat the section called “Meng-upgrade versi mesin”. Satu-satunya perbedaan adalah bahwa ketika menggunakan AWS Command Line Interface (AWS CLI), Anda menggunakan modify-db-clusterperintah dan menentukan `--db-cluster-identifier` parameter (bersama dengan `--allow-major-version-upgrade` parameter).

Lihat informasi yang lebih lengkap tentang pemutakhiran versi utama dan kecil dalam dokumentasi berikut untuk mesin basis data Anda:

- the section called “Meningkatkan mesin DB PostgreSQL”
- the section called “Meng-upgrade mesin DB MySQL”

Menggunakan Proksi RDS dengan klaster basis data Multi-AZ

Anda dapat menggunakan Proksi Amazon RDS untuk membuat proksi bagi klaster basis data Multi-AZ. Dengan menggunakan Proksi RDS, aplikasi Anda dapat mengumpulkan dan berbagi koneksi basis data untuk meningkatkan kemampuan menskalakan. Setiap proksi melakukan pembentukan multipleks koneksi, yang juga disebut dengan penggunaan ulang koneksi. Dengan multipleks, Proksi RDS melakukan semua operasi untuk sebuah transaksi dengan menggunakan satu koneksi basis data yang mendasari. Proksi RDS juga dapat mengurangi waktu henti untuk pemutakhiran versi kecil suatu klaster basis data Multi-AZ ke satu detik atau kurang. Lihat informasi yang lebih lengkap tentang manfaat-manfaat Proksi RDS di Menggunakan Proksi RDS.

Untuk menyiapkan proksi bagi klaster basis data Multi-AZ, pilih Buat Proksi RDS saat membuat klaster. Untuk petunjuk membuat dan mengelola titik akhir Proksi RDS, lihat the section called "Bekerja dengan titik akhir Proksi RDS".

Kelambatan replika dan klaster basis data Multi-AZ

Kelambatan replika adalah selisih waktu antara transaksi terbaru pada instans basis data penulis dan transaksi terbaru yang diterapkan pada instans basis data pembaca. CloudWatch Metrik Amazon `ReplicaLag` mewakili perbedaan waktu ini. Untuk informasi selengkapnya tentang CloudWatch metrik, lihatMemantau metrik Amazon RDS dengan Amazon CloudWatch.

Meskipun klaster basis data Multi-AZ memungkinkan performa tulis yang tinggi, kelambatan replika masih dapat terjadi karena sifat replikasi berbasis mesin. Karena setiap failover harus menyelesaikan dahulu kelambatan replika sebelum mempromosikan instans basis data penulis baru, memantau dan mengelola kelambatan replika ini menjadi sebuah pertimbangan.

Untuk klaster basis data Multi-AZ RDS for MySQL, waktu failover bergantung pada kelambatan replika kedua instans basis data pembaca yang tersisa. Kedua instans basis data pembaca harus menerapkan transaksi yang belum diterapkan sebelum salah satunya dipromosikan menjadi instans basis data penulis baru.

Untuk klaster basis data Multi-AZ RDS for PostgreSQL, waktu failover bergantung pada kelambatan replika terendah dari dua instans basis data pembaca yang tersisa. Instans basis data pembaca dengan kelambatan replika terendah harus menerapkan transaksi yang belum diterapkan sebelum dipromosikan menjadi instans basis data penulis baru.

Untuk tutorial yang menunjukkan cara membuat CloudWatch alarm saat lag replika melebihi jumlah waktu yang ditentukan, lihatTutorial: Membuat alarm Amazon CloudWatch untuk kelambatan replika klaster basis data Multi-AZ.

Penyebab umum kelambatan replika

Secara umum, kelambatan replika terjadi ketika beban kerja tulis terlalu tinggi bagi instans basis data pembaca untuk menerapkan transaksi dengan efisien. Berbagai beban kerja dapat menimbulkan kelambatan replika sementara atau sinambung. Berikut beberapa contoh penyebab umum:

- Konkurensi tulis tinggi atau pembaruan tumpak/batch berat pada instans basis data penulis, menyebabkan proses penerapan pada instans basis data pembaca tertinggal.

- Beban kerja baca berat yang menggunakan sumber daya pada satu atau beberapa instans basis data pembaca. Menjalankan kueri yang lambat atau besar dapat memengaruhi proses penerapan dan dapat menyebabkan kelambatan replika.
- Transaksi yang mengubah sejumlah besar data atau pernyataan DDL terkadang dapat menyebabkan kenaikan sementara kelambatan replika karena basis data harus menjaga urutan commit.

Mengurangi kelambatan replika

Untuk klaster-klaster basis data Multi-AZ RDS for MySQL dan RDS for PostgreSQL, Anda dapat mengurangi kelambatan replika dengan mengurangi beban pada instans basis data penulis. Anda juga dapat menggunakan kontrol aliran untuk mengurangi kelambatan replika. Kontrol aliran bekerja dengan melakukan throttling operasi tulis pada instans basis data penulis, yang memastikan bahwa kelambatan replika tidak terus tumbuh tanpa batas. Throttling tulis dilakukan dengan menambahkan penundaan ke akhir transaksi, yang mengurangi throughput tulis pada instans basis data penulis. Meskipun tidak menjamin hilangnya kelambatan, kontrol aliran dapat membantu mengurangi kelambatan keseluruhan dalam banyak beban kerja. Bagian-bagian berikut memberikan informasi tentang cara menggunakan kontrol aliran dengan RDS for MySQL dan RDS for PostgreSQL.

Mengurangi kelambatan replika dengan kontrol aliran untuk RDS for MySQL

Bila Anda menggunakan klaster basis data Multi-AZ RDS for MySQL, kontrol aliran diaktifkan secara default dengan menggunakan parameter dinamis `rpl_semi_sync_master_target_apply_lag`. Parameter ini menentukan batas atas yang Anda inginkan untuk kelambatan replika. Saat kelambatan replika mendekati batas yang dikonfigurasikan ini, kontrol aliran membatasi transaksi tulis pada instans basis data penulis untuk mencoba mempertahankan kelambatan replika di bawah nilai yang ditentukan. Dalam beberapa kasus, kelambatan replika dapat melebihi batas yang ditentukan. Secara default, parameter ini diatur ke 120 detik. Untuk mematikan kontrol aliran, atur parameter ini ke nilai maksimumnya 86.400 detik (satu hari).

Untuk melihat penundaan saat ini yang disuntikkan oleh kontrol aliran, tampilkan parameter `Rpl_semi_sync_master_flow_control_current_delay` dengan menjalankan kueri berikut.

```
SHOW GLOBAL STATUS like '%flow_control%';
```

Output-nya semestinya mirip dengan yang berikut.

```
+-----------------------------------------------+-------+
```

```
| Variable_name                                    | Value |
+--------------------------------------------------+-------+
| Rpl_semi_sync_master_flow_control_current_delay | 2010  |
+--------------------------------------------------+-------+
1 row in set (0.00 sec)
```

> **Note**
>
> Penundaan ditampilkan dalam mikrodetik.

Jika Wawasan Performa diaktifkan untuk klaster basis data Multi-AZ RDS for MySQL, Anda dapat memantau peristiwa tunggu yang bersangkutan dengan suatu pernyataan SQL yang menunjukkan bahwa kueri ditunda oleh kontrol aliran. Saat penundaan dikenakan oleh kontrol aliran, Anda dapat melihat peristiwa tunggu `/wait/synch/cond/semisync/semi_sync_flow_control_delay_cond` yang bersangkutan dengan pernyataan SQL itu di dasbor Wawasan Performa. Untuk melihat semua metrik ini, pastikan bahwa Skema Performa diaktifkan. Lihat informasi tentang Wawasan Performa di Memantau muatan DB dengan Wawasan Performa di Amazon RDS.

Mengurangi kelambatan replika dengan kontrol aliran untuk RDS for PostgreSQL

Saat Anda menggunakan klaster basis data Multi-AZ RDS for PostgreSQL, kontrol aliran digunakan sebagai ekstensi. Kontrol mengaktifkan pekerja latar belakang untuk semua instans basis data dalam klaster basis data. Secara default, pekerja latar belakang pada instans basis data pembaca mengomunikasikan kelambatan replika saat ini dengan pekerja latar belakang pada instans basis data penulis. Jika kelambatan melebihi dua menit pada sebarang instans basis data pembaca, pekerja latar belakang pada instans basis data penulis menambahkan penundaan di akhir transaksi. Untuk mengendalikan ambang batas kelambatan, gunakan parameter `flow_control.target_standby_apply_lag`.

Saat kontrol aliran membatasi proses PostgreSQL, peristiwa tunggu `Extension` di `pg_stat_activity` dan Wawasan Performa menunjukkan hal itu. Fungsi `get_flow_control_stats` menampilkan detail lama penundaan yang saat ini ditambahkan.

Kontrol aliran dapat menguntungkan sebagian besar beban kerja pemrosesan transaksi online (OLTP) yang memiliki transaksi singkat tetapi sangat konkuren. Jika kelambatan disebabkan oleh transaksi yang berjalan lama, seperti operasi tumpak/batch, kontrol aliran tidak memberikan manfaat yang sama besarnya.

Anda dapat mematikan kontrol aliran dengan menghapus ekstensi dari `preload_shared_libraries` dan mem-boot ulang instans basis data Anda.

Proses failover untuk klaster basis data Multi-AZ

Jika ada pemadaman terencana atau tidak terencana instans basis data dalam klaster basis data Multi-AZ, Amazon RDS akan melakukan failover ke instans basis data pembaca di Zona Ketersediaan yang berbeda. Waktu yang diperlukan untuk menyelesaikan failover bergantung pada aktivitas basis data dan kondisi-kondisi lain ketika instans basis data penulis tidak tersedia. Durasi failover biasanya kurang dari 35 detik. Failover selesai ketika kedua instans basis data pembaca telah menerapkan transaksi tertunggak dari penulis yang gagal. Setelah failover selesai, mungkin perlu beberapa waktu sebelum konsol RDS dapat mencerminkan Zona Ketersediaan baru.

Topik

Failover otomatis

Amazon RDS menangani secara otomatis failover sehingga Anda dapat melanjutkan operasi basis data secepat mungkin tanpa campur tangan administratif. Agar dapat failover, instans basis data penulis beralih secara otomatis ke instans basis data pembaca.

Melakukan failover secara manual klaster basis data Multi-AZ

Jika Anda gagal secara manual melalui cluster DB multi-AZ, RDS pertama-tama menghentikan instans DB utama. Kemudian, sistem pemantauan internal mendeteksi bahwa instans DB primer tidak sehat dan mempromosikan instance DB replika yang dapat dibaca. Durasi failover biasanya kurang dari 35 detik.

Anda dapat gagal melalui cluster DB multi-AZ secara manual menggunakan AWS Management Console, AWS CLI, atau RDS API.

Konsol

Untuk melakukan secara manual failover klaster basis data Multi-AZ

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data.
3. Pilih klaster basis data Multi-AZ yang ingin Anda lakukan failover.
4. Untuk Tindakan, pilih Failover.

   Halaman Failover klaster basis data muncul.
5. Pilih Failover untuk menegaskan failover manual.

AWS CLI

Untuk gagal melalui cluster DB Multi-AZ secara manual, gunakan AWS CLI perintah failover-db-cluster.

Example

```
aws rds failover-db-cluster --db-cluster-identifier mymultiazdbcluster
```

API RDS

Untuk melakukan failover secara manual klaster basis data Multi-AZ, panggil API Amazon RDS FailoverDBCluster dan tentukan file `DBClusterIdentifier`.

Menentukan apakah klaster basis data Multi-AZ sudah failover

Untuk mengetahui apakah klaster basis data Multi-AZ Anda sudah failover, Anda dapat melakukan hal-hal berikut:

- Menyiapkan pelangganan peristiwa basis data untuk memberi tahu Anda melalui email atau SMS bahwa failover telah dimulai. Lihat informasi yang lebih lengkap tentang peristiwa di Menggunakan pemberitahuan peristiwa Amazon RDS.
- Lihat peristiwa basis data Anda dengan menggunakan konsol Amazon RDS atau operasi API.
- Lihat status klaster DB multi-AZ Anda saat ini dengan menggunakan konsol Amazon RDS, API AWS CLI, dan RDS.

Lihat informasi tentang cara merespons failover, mengurangi waktu pemulihan, dan praktik-praktik terbaik lain untuk Amazon RDS di Praktik terbaik untuk Amazon RDS.

Mengatur TTL JVM untuk pencarian nama DNS

Mekanisme failover mengubah secara otomatis catatan Sistem Nama Domain (DNS) dari instans basis data untuk menunjuk ke instans basis data pembaca. Alhasil, Anda perlu membentuk kembali koneksi yang ada dengan instans basis data Anda. Di lingkungan mesin virtual Java (JVM), karena cara kerja mekanisme caching DNS Java, Anda mungkin perlu mengonfigurasi ulang pengaturan JVM.

JVM menyimpan cache pencarian nama DNS. Ketika JVM menyelesaikan nama host ke alamat IP, itu cache alamat IP untuk jangka waktu tertentu, yang dikenal sebagai (TTL). time-to-live

Karena AWS sumber daya menggunakan entri nama DNS yang terkadang berubah, kami sarankan Anda mengonfigurasi JVM Anda dengan nilai TTL tidak lebih dari 60 detik. Hal ini dapat memastikan bahwa ketika alamat IP sumber daya berubah, aplikasi Anda dapat menerima dan menggunakan alamat IP baru sumber daya dengan mengueri ulang DNS.

Pada beberapa konfigurasi Java, TTL default JVM diatur untuk tidak pernah menyegarkan entri DNS hingga JVM dimulai ulang. Jadi, jika alamat IP untuk AWS sumber daya berubah saat aplikasi Anda masih berjalan, itu tidak dapat menggunakan sumber daya itu sampai Anda secara manual me-restart JVM dan informasi IP cache di-refresh. Dalam kasus ini, mengatur TTL JVM adalah penting agar fungsi itu menyegarkan secara berkala informasi IP yang tersimpan dalam cache.

> **Note**
>
> TTL default dapat bervariasi menurut versi JVM dan apakah manajer keamanan terinstal. Banyak JVM memberikan TTL default kurang dari 60 detik. Jika Anda menggunakan JVM seperti itu dan tidak menggunakan manajer keamanan, Anda dapat mengabaikan topik selanjutnya. Untuk informasi selengkapnya tentang manajer keamanan di Oracle, lihat The security manager dalam dokumentasi Oracle.

Untuk mengubah TTL JVM, atur nilai properti `networkaddress.cache.ttl`. Gunakan salah satu metode berikut, tergantung pada kebutuhan Anda:

- Untuk menetapkan nilai properti secara global untuk semua aplikasi yang menggunakan JVM, tetapkan `networkaddress.cache.ttl` dalam file `$JAVA_HOME/jre/lib/security/java.security`.

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

- Untuk menetapkan properti secara lokal hanya untuk aplikasi Anda, tetapkan `networkaddress.cache.ttl` dalam kode inisialisasi aplikasi Anda sebelum koneksi jaringan dibuat.

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

Membuat klaster DB Multi-AZ

Klaster DB Multi-AZ memiliki instans DB penulis dan dua instans DB pembaca di tiga Zona Ketersediaan terpisah. Klaster DB Multi-AZ menyediakan ketersediaan tinggi, peningkatan kapasitas untuk beban kerja baca, dan latensi yang lebih rendah jika dibandingkan dengan deployment Multi-AZ. Untuk informasi selengkapnya tentang klaster DB Multi-AZ, lihat Deployment klaster basis data Multi-AZ.

> **Note**
>
> Klaster DB Multi-AZ hanya didukung untuk mesin MySQL dan PostgreSQL DB.

Prasyarat klaster DB

> **Important**
>
> Sebelum dapat membuat klaster DB Multi-AZ, Anda harus menyelesaikan tugas-tugas di Menyiapkan Amazon RDS.

Berikut adalah prasyarat yang harus diselesaikan sebelum membuat klaster DB Multi-AZ.

Topik

- Konfigurasikan jaringan untuk klaster DB
- Prasyarat tambahan

Konfigurasikan jaringan untuk klaster DB

Anda hanya dapat membuat klaster DB Multi-AZ di Cloud Privat Virtual (VPC) berdasarkan layanan Amazon VPC. Itu harus dalam Wilayah AWS yang memiliki setidaknya tiga Availability Zone. Grup subnet DB yang Anda pilih untuk klaster DB harus mencakup setidaknya tiga Zona Ketersediaan. Konfigurasi ini memastikan bahwa setiap instans DB di klaster DB berada di Zona Ketersediaan yang berbeda.

Untuk mengatur konektivitas antara klaster DB baru Anda dan instans Amazon EC2 di VPC yang sama, lakukan saat Anda membuat klaster DB. Untuk terhubung ke klaster DB Anda dari sumber daya selain instans EC2 di VPC yang sama, konfigurasikan koneksi jaringan secara manual.

Topik

- Mengonfigurasi konektivitas jaringan otomatis dengan instans EC2
- Mengonfigurasi jaringan secara manual

Mengonfigurasi konektivitas jaringan otomatis dengan instans EC2

Saat Anda membuat cluster DB multi-AZ, Anda dapat menggunakan AWS Management Console untuk mengatur konektivitas antara instans EC2 dan cluster DB baru. Ketika Anda melakukannya, RDS mengonfigurasi VPC dan pengaturan jaringan Anda secara otomatis. Klaster DB tersebut dibuat dalam VPC yang sama dengan instans EC2, sehingga instans EC2 dapat mengakses klaster DB.

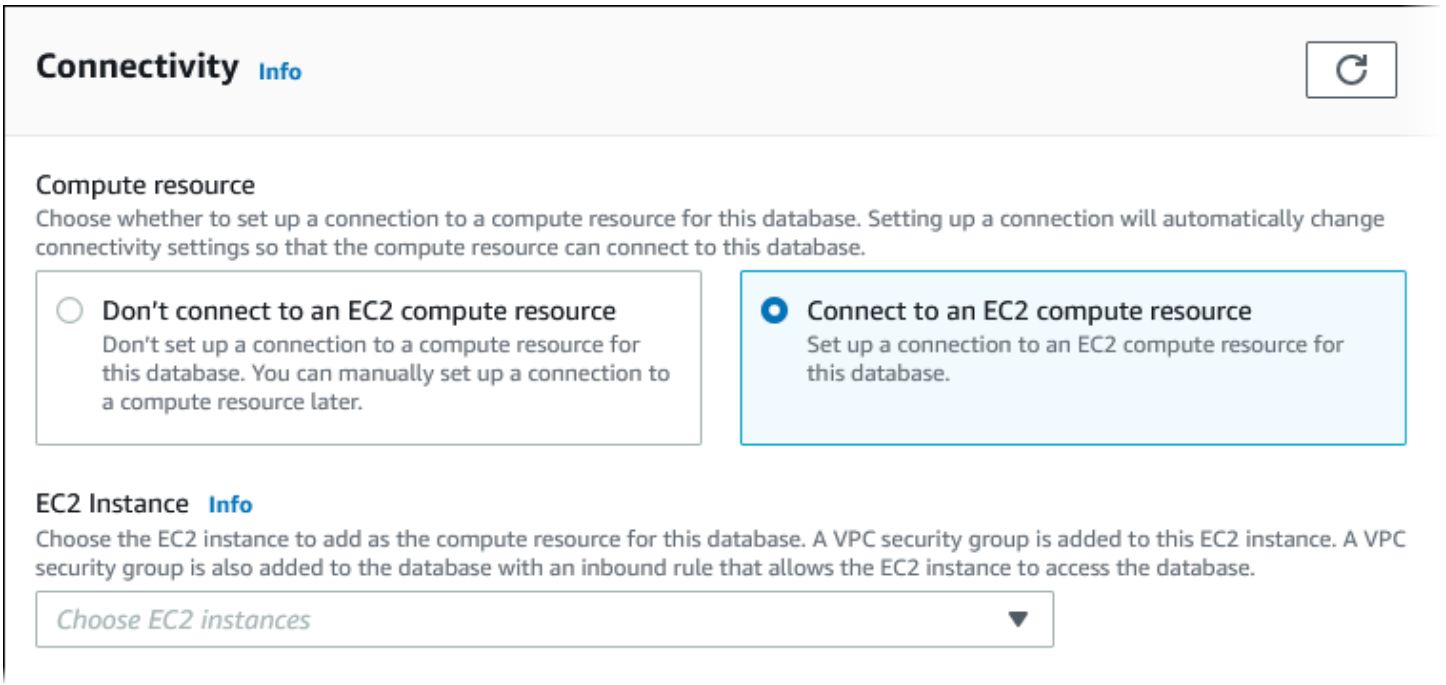
Berikut ini adalah persyaratan untuk menghubungkan instans EC2 dengan klaster DB:

- Instans EC2 harus ada Wilayah AWS sebelum Anda membuat cluster DB.

  Jika tidak ada instans EC2 di Wilayah AWS, konsol menyediakan tautan untuk membuatnya.
- Pengguna yang membuat klaster DB harus memiliki izin untuk melakukan operasi berikut:
  - `ec2:AssociateRouteTable`
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:AuthorizeSecurityGroupIngress`
  - `ec2:CreateRouteTable`
  - `ec2:CreateSubnet`
  - `ec2:CreateSecurityGroup`
  - `ec2:DescribeInstances`
  - `ec2:DescribeNetworkInterfaces`
  - `ec2:DescribeRouteTables`
  - `ec2:DescribeSecurityGroups`
  - `ec2:DescribeSubnets`
  - `ec2:ModifyNetworkInterfaceAttribute`
  - `ec2:RevokeSecurityGroupEgress`

Menggunakan opsi ini membuat klaster DB pribadi. Klaster DB menggunakan grup subnet DB hanya dengan subnet privat untuk membatasi akses ke sumber daya dalam VPC.

Untuk menghubungkan instans EC2 ke klaster DB, pilih Hubungkan ke sumber daya komputasi EC2 di bagian Konektivitas pada halaman Buat basis data.



Saat Anda memilih Hubungkan ke sumber daya komputasi EC2, RDS menetapkan opsi berikut secara otomatis. Anda tidak dapat mengubah pengaturan ini kecuali Anda memilih untuk tidak mengatur konektivitas dengan instans EC2 dengan memilih Jangan hubungkan ke sumber daya komputasi EC2.

Opsi konsol	Pengaturan otomatis
Cloud Privat Virtual (VPC)	RDS mengatur VPC ke yang terkait dengan instans EC2.
Grup subnet DB	RDS membutuhkan grup subnet DB dengan subnet privat di Zona Ketersediaan yang sama dengan instans EC2. Jika grup subnet DB yang memenuhi persyaratan ini ada, RDS menggunakan grup subnet DB yang ada. Secara default, opsi ini diatur ke Pengaturan otomatis. Ketika Anda memilih Pengaturan otomatis dan tidak ada grup subnet DB yang memenuhi persyaratan ini, tindakan berikut terjadi. RDS menggunakan tiga subnet privat yang tersedia di tiga Zona Ketersediaan dengan salah satu Zona Ketersediaan

Opsi konsol	Pengaturan otomatis
	sama dengan instans EC2. Jika subnet privat tidak tersedia di Zona Ketersediaan, RDS membuat subnet privat di Zona Ketersediaan. Kemudian RDS membuat grup subnet DB. Ketika subnet privat tersedia, RDS menggunakan tabel rute yang terkait dengan subnet dan menambahkan subnet apa pun yang dibuatnya ke tabel rute ini. Ketika tidak ada subnet privat yang tersedia, RDS membuat tabel rute tanpa akses gateway internet dan menambahkan subnet yang dibuatnya ke tabel rute. RDS juga memungkinkan Anda untuk menggunakan grup subnet DB yang ada. Pilih Pilih yang ada jika Anda ingin menggunakan grup subnet DB pilihan Anda yang sudah ada.
Akses publik	RDS memilih Tidak sehingga klaster DB tidak dapat diakses publik. Untuk keamanan, ini adalah praktik terbaik untuk menjaga basis data tetap privat dan memastikannya tidak dapat diakses dari internet.

Opsi konsol	Pengaturan otomatis
Grup keamanan VPC (firewall)	RDS membuat grup keamanan baru yang terkait dengan klaster DB. Grup keamanan diberi nama `rds-ec2-`*n*, dengan *n* berupa angka. Grup keamanan ini mencakup aturan masuk dengan grup keamanan VPC EC2 (firewall) sebagai sumbernya. Grup keamanan yang terkait dengan klaster DB ini memungkinkan instans EC2 untuk mengakses klaster DB. RDS juga membuat grup keamanan baru yang terkait dengan instans EC2. Grup keamanan diberi nama `ec2-rds-`*n*, dengan *n* berupa angka. Grup keamanan ini mencakup aturan keluar dengan grup keamanan VPC dari klaster DB sebagai sumbernya. Grup keamanan ini memungkinkan instans EC2 untuk mengirim lalu lintas ke klaster DB. Anda dapat menambahkan grup keamanan baru lainnya dengan memilih Buat baru dan mengetik nama grup keamanan baru. Anda dapat menambahkan grup keamanan yang ada dengan memilih Pilih yang ada dan memilih grup keamanan untuk ditambahkan.
Zona Ketersediaan	RDS memilih Zona Ketersediaan dari instans EC2 untuk satu instans DB dalam deployment klaster DB Multi-AZ. RDS secara acak memilih Zona Ketersediaan yang berbeda untuk kedua instans DB lainnya. Instans DB penulis dibuat di Zona Ketersedi aan yang sama dengan instans EC2. Kemungkinan akan ada biaya lintas Zona Ketersediaan jika terjadi failover dan instans DB penulis berada di Zona Ketersediaan yang berbeda.

Untuk informasi selengkapnya tentang pengaturan ini, lihat Pengaturan untuk membuat klaster DB Multi-AZ.

Jika Anda mengubah pengaturan ini setelah klaster DB dibuat, perubahan dapat memengaruhi koneksi antara instans EC2 dan klaster DB.

Mengonfigurasi jaringan secara manual

Untuk terhubung ke klaster DB Anda dari sumber daya selain instans EC2 di VPC yang sama, konfigurasikan koneksi jaringan secara manual. Jika Anda menggunakan AWS Management Console untuk membuat cluster DB multi-AZ Anda, Anda dapat meminta Amazon RDS secara otomatis membuat VPC untuk Anda. Anda juga dapat menggunakan VPC yang ada atau membuat VPC baru untuk klaster DB Multi-AZ Anda. VPC Anda harus memiliki setidaknya satu subnet di masing-masing dari setidaknya tiga Zona Ketersediaan untuk digunakan dengan klaster DB Multi-AZ. Untuk informasi tentang VPC, lihat Amazon VPC dan Amazon RDS.

Jika Anda tidak memiliki VPC default atau belum membuat VPC, dan Anda tidak berencana menggunakan konsol, lakukan hal berikut:

- Buat VPC dengan setidaknya satu subnet di masing-masing setidaknya tiga Availability Zone di AWS Wilayah tempat Anda ingin menerapkan cluster DB Anda. Untuk informasi selengkapnya, lihat Bekerja dengan klaster DB dalam VPC.
- Tentukan grup keamanan VPC yang mengizinkan koneksi ke klaster DB Anda. Lihat informasi yang lebih lengkap di Memberikan akses ke instans DB di VPC Anda dengan membuat grup keamanan dan Mengontrol akses dengan grup keamanan.
- Tentukan grup subnet DB RDS yang menentukan setidaknya tiga subnet di VPC yang dapat digunakan oleh klaster DB Multi-AZ. Untuk informasi selengkapnya, lihat Bekerja dengan grup subnet DB.

Untuk informasi tentang batasan yang berlaku pada klaster DB Multi-AZ, lihat Keterbatasan cluster DB multi-AZ.

Jika Anda ingin terhubung ke sumber daya yang tidak berada di VPC yang sama dengan klaster DB Multi-AZ, lihat skenario yang sesuai di Skenario untuk mengakses instans DB di VPC.

Prasyarat tambahan

Sebelum Anda membuat klaster DB Multi-AZ, pertimbangkan prasyarat tambahan berikut:

- Untuk terhubung AWS menggunakan kredensi AWS Identity and Access Management (IAM), AWS akun Anda harus memiliki kebijakan IAM tertentu. Ini memberikan izin yang diperlukan untuk melakukan operasi Amazon RDS. Untuk informasi selengkapnya, lihat Manajemen identitas dan akses untuk Amazon RDS.

Jika Anda menggunakan IAM untuk mengakses konsol RDS, pertama-tama masuk ke AWS Management Console dengan kredenal pengguna IAM Anda. Kemudian, buka konsol RDS di https://console.aws.amazon.com/rds/.

- Untuk menyesuaikan parameter konfigurasi klaster DB Anda, tentukan grup parameter klaster DB dengan pengaturan parameter yang diperlukan. Untuk informasi tentang membuat atau memodifikasi grup parameter klaster DB, lihat Bekerja dengan grup parameter untuk klaster basis data Multi-AZ.
- Tentukan nomor port TCP/IP untuk menentukan klaster DB Anda. Firewall di beberapa perusahaan memblokir koneksi ke port default. Jika firewall perusahaan Anda memblokir port default, pilih port lain untuk klaster DB Anda. Semua instans DB dalam klaster DB menggunakan port yang sama.
- Jika versi mesin utama untuk database Anda mencapai akhir RDS pada tanggal dukungan standar, Anda harus menggunakan opsi Extended Support CLI atau parameter RDS API. Untuk informasi selengkapnya, lihat RDS Extended Support diPengaturan untuk membuat klaster DB Multi-AZ.

Membuat klaster DB

Anda dapat membuat cluster DB multi-AZ menggunakan AWS Management Console, AWS CLI, atau RDS API.

Konsol

Anda dapat membuat klaster DB Multi-AZ dengan memilih klaster DB Multi-AZ di bagian Ketersediaan dan daya tahan.

Cara membuat klaster DB Multi-AZ menggunakan konsol

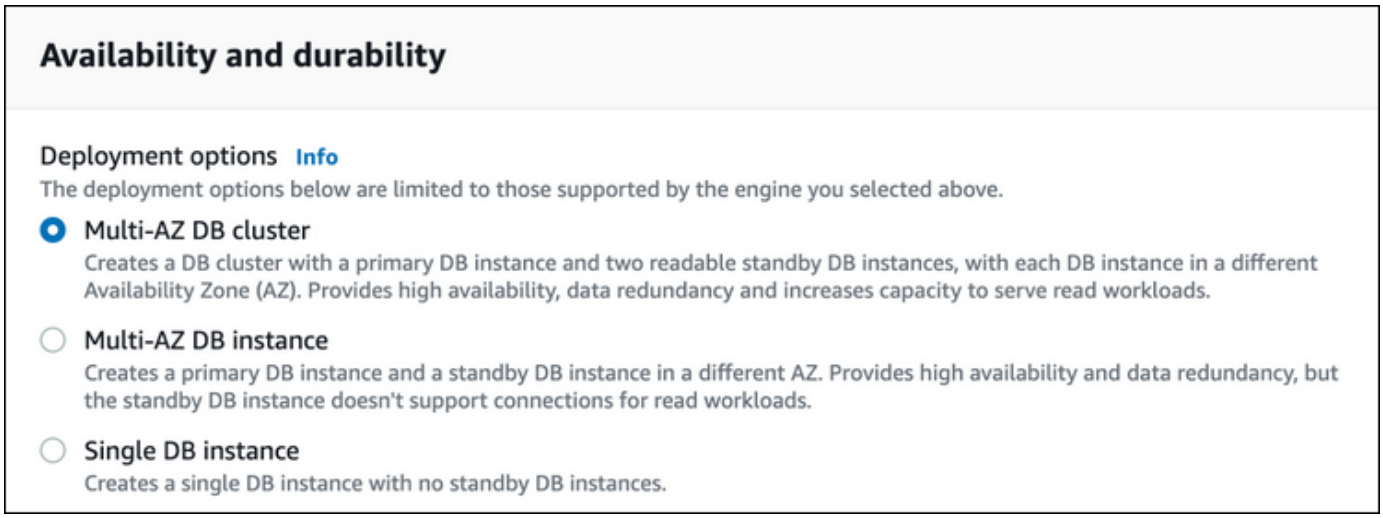
1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di sudut kanan atas AWS Management Console, pilih Wilayah AWS di mana Anda ingin membuat cluster DB.

   Untuk informasi tentang Wilayah AWS yang mendukung klaster DB multi-AZ, lihat. Keterbatasan cluster DB multi-AZ
3. Di panel navigasi, pilih Basis Data.
4. Pilih Buat basis data.

Untuk membuat klaster DB Multi-AZ, pastikan Pembuatan Standar dipilih dan Mudah Dibuat tidak.

5. Pada Jenis mesin, pilih MySQL atau PostgreSQL.
6. Untuk Versi, pilih versi mesin DB.

   Untuk informasi tentang versi mesin DB yang mendukung klaster DB Multi-AZ, lihat Keterbatasan cluster DB multi-AZ.
7. Di bagian Templat, pilih templat yang sesuai untuk deployment Anda.
8. Dalam Ketersediaan dan daya tahan, pilih Klaster DB Multi-AZ.



9. Di Pengidentifikasi klaster DB, masukkan pengidentifikasi untuk klaster DB Anda.
10. Di Nama pengguna master, masukkan nama pengguna master Anda atau biarkan pengaturan default.
11. Masukkan kata sandi master Anda:

    a. Di bagian Pengaturan, buka Pengaturan Kredensial.

    b. Jika Anda ingin menentukan kata sandi, hapus kotak Buat kata sandi secara otomatis jika dipilih.

    c. (Opsional) Ubah nilai Nama pengguna master.

    d. Masukkan kata sandi yang sama di Kata sandi master dan Konfirmasikan kata sandi.
12. Untuk Kelas instans DB, pilih kelas instans DB. Untuk daftar kelas instans DB yang didukung, lihat the section called “Ketersediaan kelas instans untuk cluster DB multi-AZ”.
13. (Opsional) Atur koneksi ke sumber daya komputasi untuk klaster DB ini.

Anda dapat mengonfigurasi konektivitas antara instans Amazon EC2 dan klaster DB baru selama pembuatan klaster DB. Untuk informasi selengkapnya, lihat Mengonfigurasi konektivitas jaringan otomatis dengan instans EC2.

14. Di bagian Konektivitas di bawah Grup keamanan VPC (firewall), jika Anda memilih Buat baru, grup keamanan VPC dibuat dengan aturan masuk yang memungkinkan alamat IP komputer lokal Anda mengakses basis data.
15. Untuk bagian yang lainnya, tentukan pengaturan klaster DB Anda. Untuk informasi tentang setiap pengaturan, lihat Pengaturan untuk membuat klaster DB Multi-AZ.
16. Pilih Buat basis data.

    Jika Anda memilih untuk menggunakan kata sandi yang dibuat otomatis, tombol Lihat detail kredensial muncul pada halaman Basis data.

    Untuk melihat nama pengguna dan kata sandi master klaster DB, pilih Lihat detail kredensial.

    Untuk terhubung ke klaster DB sebagai pengguna master, gunakan nama pengguna dan kata sandi yang muncul.

    > **Important**
    >
    > Anda tidak dapat melihat kata sandi pengguna master lagi.

17. Untuk Basis data, pilih nama klaster DB baru.

Pada konsol RDS, detail untuk klaster DB baru muncul. Klaster DB memiliki status Membuat hingga klaster DB telah dibuat dan siap digunakan. Saat statusnya berubah menjadi Tersedia, Anda dapat terhubung ke klaster DB. Bergantung pada kelas klaster DB dan penyimpanan yang dialokasikan, perlu waktu beberapa menit agar klaster DB baru tersedia.

AWS CLI

Sebelum Anda membuat cluster DB multi-AZ menggunakan AWS CLI, pastikan untuk memenuhi prasyarat yang diperlukan. Prasyarat tersebut termasuk membuat VPC dan grup subnet DB RDS. Untuk informasi selengkapnya, lihat Prasyarat klaster DB.

Untuk membuat cluster DB multi-AZ dengan menggunakan AWS CLI, panggil create-db-clusterperintah. Tentukan `--db-cluster-identifier`. Untuk opsi `--engine`, tentukan salah `mysql` atau `postgres`.

Lihat informasi tentang setiap opsi di Pengaturan untuk membuat klaster DB Multi-AZ.

Untuk informasi tentang, mesin DB Wilayah AWS, dan versi mesin DB yang mendukung cluster DB multi-AZ, lihat. Keterbatasan cluster DB multi-AZ

Perintah `create-db-cluster` membuat instans DB penulis dan dua instans DB pembaca untuk klaster DB Anda. Setiap instans DB berada di Zona Ketersediaan yang berbeda.

Misalnya, perintah berikut membuat klaster DB Multi-AZ MySQL 8.0 bernama `mysql-multi-az-db-cluster`.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-cluster \
   --db-cluster-identifier mysql-multi-az-db-cluster \
   --engine mysql \
   --engine-version 8.0.28  \
   --master-username admin \
   --manage-master-user-password  \
   --port 3306 \
   --backup-retention-period 1  \
   --db-subnet-group-name default \
   --allocated-storage 4000 \
   --storage-type io1 \
   --iops 10000 \
   --db-cluster-instance-class db.m5d.xlarge
```

Untuk Windows:

```
aws rds create-db-cluster ^
   --db-cluster-identifier mysql-multi-az-db-cluster ^
   --engine mysql ^
   --engine-version 8.0.28 ^
   --manage-master-user-password ^
   --master-username admin ^
   --port 3306 ^
   --backup-retention-period 1 ^
   --db-subnet-group-name default ^
   --allocated-storage 4000 ^
   --storage-type io1 ^
```

```
    --iops 10000 ^
    --db-cluster-instance-class db.m5d.xlarge
```

Perintah berikut membuat klaster DB Multi-AZ PostgreSQL 13.4 bernama `postgresql-multi-az-db-cluster`.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-cluster \
    --db-cluster-identifier postgresql-multi-az-db-cluster \
    --engine postgres \
    --engine-version 13.4 \
    --manage-master-user-password \
    --master-username postgres \
    --port 5432 \
    --backup-retention-period 1  \
    --db-subnet-group-name default \
    --allocated-storage 4000 \
    --storage-type io1 \
    --iops 10000 \
    --db-cluster-instance-class db.m5d.xlarge
```

Untuk Windows:

```
aws rds create-db-cluster ^
    --db-cluster-identifier postgresql-multi-az-db-cluster ^
    --engine postgres ^
    --engine-version 13.4 ^
    --manage-master-user-password ^
    --master-username postgres ^
    --port 5432 ^
    --backup-retention-period 1 ^
    --db-subnet-group-name default ^
    --allocated-storage 4000 ^
    --storage-type io1 ^
    --iops 10000 ^
    --db-cluster-instance-class db.m5d.xlarge
```

API RDS

Sebelum dapat membuat klaster DB Multi-AZ menggunakan API RDS, pastikan untuk memenuhi prasyarat yang diperlukan, seperti membuat VPC dan grup subnet DB RDS. Untuk informasi selengkapnya, lihat Prasyarat klaster DB.

Untuk membuat klaster DB Multi-AZ dengan menggunakan API RDS, panggil operasi CreateDBCluster. Tentukan `DBClusterIdentifier`. Untuk parameter `Engine`, tentukan `mysql` atau `postgresql`.

Lihat informasi tentang setiap opsi, lihat Pengaturan untuk membuat klaster DB Multi-AZ.

Operasi `CreateDBCluster` membuat instans DB penulis dan dua instans DB pembaca untuk klaster DB Anda. Setiap instans DB berada di Zona Ketersediaan yang berbeda.

Pengaturan untuk membuat klaster DB Multi-AZ

Untuk detail pengaturan yang Anda pilih ketika membuat klaster DB Multi-AZ, lihat tabel berikut. Untuk informasi selengkapnya tentang AWS CLI opsi, lihat create-db-cluster. Untuk informasi selengkapnya tentang parameter API RDS, lihat CreateDBCluster.

Pengaturan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS

Penyimpanan yang dialokasikan	Jumlah penyimpanan yang dialokasikan untuk setiap instans DB dalam klaster DB Anda (dalam gibibyte). Untuk informasi selengkapnya, lihat Penyimpanan instans DB Amazon RDS.	Opsi CLI:
		`--allocated-storage`
		Parameter API:
		`AllocatedStorage`

Peningkatan versi minor otomatis	Aktifkan peningkatan versi minor otomatis agar klaster basis data Anda otomatis menerima peningkatan versi mesin DB minor pilihan Anda saat tersedia. Amazon RDS melakukan peningkatan versi minor otomatis selama jendela pemeliharaan.	Opsi CLI:
		`--auto-minor-version-upgrade`
		`--no-auto-minor-version-upgrade`
		Parameter API:

Pengaturan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS
		`AutoMinorVersionUpgrade`

Periode retensi cadangan	Jumlah hari yang Anda inginkan untuk menyimpan cadangan otomatis klaster DB Anda. Untuk klaster DB Multi-AZ, nilai ini harus ditetapkan ke **1** atau lebih besar.
	
	Untuk informasi selengkapnya, lihat Pengantar cadangan.
		Opsi CLI:
		
		`--backup-retention-period`
		
		Parameter API:
		
		`BackupRetentionPeriod`

Jendela pencadangan	Periode waktu ketika Amazon RDS membuat cadangan klaster basis data Anda secara otomatis. Kecuali jika Anda ingin basis data dicadangkan pada waktu tertentu, gunakan nilai default Tidak ada preferensi.
	
	Untuk informasi selengkapnya, lihat Pengantar cadangan.
		Opsi CLI:
		
		`--preferred-backup-window`
		
		Parameter API:
		
		`PreferredBackupWindow`

Otoritas sertifikat	Otoritas sertifikat (CA) untuk sertifikat server yang digunakan oleh cluster DB.
	
	Untuk informasi selengkapnya, lihat .
		Opsi CLI:
		
		`--ca-certificate-identifier`
		
		Parameter API RDS:
		
		`CACertificateIdentifier`

Pengaturan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS

Salin tag ke cuplikan	Opsi ini menyalin semua tag klaster basis data ke snapshot DB saat Anda membuat snapshot.
	
	Untuk informasi selengkapnya, lihat Memberi tag pada sumber daya Amazon RDS.
		Opsi CLI:
		
		`-copy-tags-to-snapshot`
		
		`-no-copy-tags-to-snapshot`
		
		Parameter API RDS:
		
		`CopyTagsToSnapshot`

Autentikasi basis data	Untuk klaster DB Multi-AZ, hanya Autentikasi kata sandi yang didukung.	Tidak ada karena autentikasi kata sandi adalah default.

Port basis data	Port yang ingin Anda gunakan untuk mengakses klaster DB. Port default akan ditampilkan.
	
	Port tidak dapat diubah setelah klaster DB dibuat.
	
	Firewall di beberapa perusahaan memblokir koneksi ke port default. Jika firewall perusahaan Anda memblokir port default, masukkan port lain untuk klaster DB Anda.
		Opsi CLI:
		
		`--port`
		
		Parameter API RDS:
		
		`Port`

Pengidentifikasi klaster DB	Nama untuk klaster DB Anda. Beri nama klaster DB Anda dengan cara yang sama seperti cara Anda menamai server on-premise Anda. Pengidentifikasi cluster DB Anda dapat berisi hingga 63 karakter alfanumerik, dan harus unik untuk akun Anda di Wilayah yang Anda pilih. AWS	Opsi CLI:
		
		`--db-cluster-identifier`
		
		Parameter API RDS:
		
		`DBClusterIdentifier`

Pengaturan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS

Kelas instans DB	Kapasitas komputasi dan memori setiap instans DB di klaster DB Multi-AZ, misalnya `db.m5d.xlarge` .
	
	Jika memungkinkan, pilih kelas instans DB yang cukup besar sehingga set kerja kueri yang umum dapat disimpan di memori. Ketika set kerja disimpan di memori, sistem dapat menghindari penulisan pada disk, yang akan meningkatkan performa.
	
	Untuk daftar kelas instans DB yang didukung, lihat the section called “Ketersediaan kelas instans untuk cluster DB multi-AZ”.
		Opsi CLI:
		
		`--db-cluster-instance-class`
		
		Parameter API RDS:
		
		`DBClusterInstanceClass`

Grup parameter klaster DB	Grup parameter klaster DB yang ingin Anda kaitkan dengan klaster DB.
	
	Untuk informasi selengkapnya, lihat Bekerja dengan grup parameter untuk klaster basis data Multi-AZ.
		Opsi CLI:
		
		`--db-cluster-parameter-group-name`
		
		Parameter API RDS:
		
		`DBClusterParameterGroupName`

Versi mesin DB	Versi mesin basis data yang ingin Anda gunakan.	Opsi CLI:
		
		`--engine-version`
		
		Parameter API RDS:
		
		`EngineVersion`

Pengaturan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS
Grup parameter klaster DB	Grup parameter instans DB untuk mengasosiasikan dengan cluster DB. Untuk informasi selengkapnya, lihat Bekerja dengan grup parameter untuk klaster basis data Multi-AZ.	Opsi CLI: `--db-cluster-parameter-group-name` Parameter API RDS: **`DBClusterParameterGroupName`**

Grup subnet DB	Grup subnet DB yang Anda ingin gunakan untuk klaster DB.
	Pilih Pilih yang ada untuk menggunakan grup subnet DB yang ada. Kemudian pilih grup subnet yang diperlukan dari daftar dropdown Grup subnet DB yang ada.
	
	Pilih Pengaturan otomatis untuk membiarkan RDS memilih grup subnet DB yang kompatibel. Jika tidak ada, RDS membuat grup subnet baru untuk klaster Anda.
	
	Untuk informasi selengkapnya, lihat Bekerja dengan grup subnet DB.
		Opsi CLI:
		
		`--db-subnet-group-name`
		
		Parameter API RDS:
		
		`DBSubnetGroupName`

Pengaturan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS

Perlindungan penghapusan	Aktifkan perlindungan penghapusan agar klaster DB tidak terhapus. Jika Anda membuat klaster DB produksi dengan konsol, perlindungan penghapusan diaktifkan secara default.
	
	Untuk informasi selengkapnya, lihat Menghapus instans DB.
		Opsi CLI:
		
		`--deletion-protection`
		
		`--no-deletion-protection`
		
		Parameter API RDS:
		
		`DeletionProtection`

Enkripsi	Aktifkan Enkripsi guna mengaktifkan enkripsi saat diam untuk klaster DB ini.
	
	Enkripsi diaktifkan secara default untuk klaster DB Multi-AZ.
	
	Untuk informasi selengkapnya, lihat Mengenkripsi sumber daya Amazon RDS.
		Opsi CLI:
		
		`--kms-key-id`
		
		`--storage-encrypted`
		
		`--no-storage-encrypted`
		
		Parameter API RDS:
		
		`KmsKeyId`
		
		`StorageEncrypted`

Pemantauan yang Ditingkatkan	Aktifkan pemantauan yang ditingkatkan guna mengaktifkan pengumpulan metrik secara waktu nyata untuk sistem operasi tempat klaster DB Anda berjalan.
	
	Untuk informasi selengkapnya, lihat Memantau metrik OS dengan Pemantauan yang Disempurnakan.
		Opsi CLI:
		
		`--monitoring-interval`
		
		`--monitoring-role-arn`
		
		Parameter API RDS:
		
		`MonitoringInterval`
		
		`MonitoringRoleArn`

Pengaturan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS

Nama basis data awal	Nama untuk basis data di klaster DB Anda. Jika Anda tidak memberikan nama, Amazon RDS tidak membuat basis data di klaster DB untuk MySQL. Namun, Amazon RDS membuat basis data pada klaster DB untuk PostgreSQL. Nama tidak dapat berupa kata yang disimpan oleh mesin basis data. Terdapat batasan lain bergantung pada mesin DB.
	
	MySQL:
	
	• Harus berisi antara 1–64 karakter alfanumerik.
	
	PostgreSQL:
	
	• Harus berisi antara 1–63 karakter alfanumerik.
	• Harus dimulai dengan sebuah huruf atau garis bawah. Karakter selanjutnya dapat berupa huruf, garis bawah, atau digit (0-9).
	• Nama basis data awal adalah `postgres`.
		Opsi CLI:
		
		`--database-name`
		
		Parameter API RDS:
		
		`DatabaseName`

Pengaturan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS

Ekspor log	Jenis file log database untuk dipublikasikan ke Amazon CloudWatch Logs.
	
	Untuk informasi selengkapnya, lihat Menerbitkan log basis data ke Log Amazon CloudWatch.
		Opsi CLI:
		
		`-enable-cloudwatch-logs-exports`
		
		Parameter API RDS:
		
		`EnableCloudwatchLogsExports`

Jendela pemeliharaan	Jendela 30 menit ketika perubahan yang tertunda untuk klaster DB diterapkan. Jika jangka waktu bukan masalah, pilih Tidak ada preferensi.
	
	Untuk informasi selengkapnya, lihat Periode pemeliharaan Amazon RDS.
		Opsi CLI:
		
		`--preferred-maintenance-window`
		
		Parameter API RDS:
		
		`PreferredMaintenanceWindow`

Kelola kredensi master di AWS Secrets Manager	Pilih Kelola kredensial master di AWS Secrets Manager untuk mengelola kata sandi pengguna master dalam rahasia di Secrets Manager.
	
	Anda juga dapat memilih kunci KMS yang akan digunakan untuk melindungi rahasia. Pilih dari kunci KMS di akun Anda, atau masukkan kunci dari akun yang berbeda.
	
	Untuk informasi selengkapnya, lihat Manajemen kata sandi dengan Amazon RDS Aurora dan AWS Secrets Manager.
		Opsi CLI:
		
		`--manage-master-user-password | --no-manage-master-user-password`
		
		`--master-user-secret-kms-key-id`
		
		Parameter API RDS:
		
		`ManageMasterUserPassword`
		
		`MasterUserSecretKmsKeyId`

Pengaturan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS

Kata sandi master	Kata sandi untuk akun pengguna master Anda.	Opsi CLI:
		`--master-user-password`
		Parameter API RDS:
		`MasterUserPassword`

Nama pengguna master	Nama yang Anda gunakan sebagai nama pengguna master untuk masuk ke klaster DB dengan semua hak istimewa basis data.
	• Nama dapat berisi 1–16 karakter alfanumerik dan garis bawah.
	• Karakter pertamanya harus berupa huruf.
	• Nama tidak dapat berupa kata yang disimpan oleh mesin basis data.
	Anda tidak dapat mengubah nama pengguna master setelah klaster DB Multi-AZ dibuat.
	Untuk informasi selengkapnya tentang hak istimewa yang diberikan kepada pengguna master, lihat Hak akses akun pengguna master.
		Opsi CLI:
		`--master-username`
		Parameter API RDS:
		`MasterUsername`

Pengaturan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS

Wawasan Performa	Aktifkan Wawasan Performa untuk memantau beban klaster DB, sehingga Anda dapat menganalisis dan memecahkan masalah performa basis data Anda.
	
	Pilih periode retensi untuk menentukan berapa banyak riwayat data Wawasan Performa yang akan disimpan. Pengaturan retensi di tingkat gratis adalah Default (7 hari). Untuk mempertahankan data performa Anda lebih lama, tetapkan 1–24 bulan. Untuk informasi selengkapnya tentang periode retensi, lihat Harga dan retensi data untuk Wawasan Performa.
	
	Pilih kunci master yang akan digunakan untuk melindungi kunci enkripsi volume basis data ini. Pilih dari kunci master di akun Anda atau masukkan kunci dari akun yang lain.
	
	Untuk informasi selengkapnya, lihat Memantau muatan DB dengan Wawasan Performa di Amazon RDS.
		Opsi CLI:
		
		`--enable-performance-insights`
		
		`--no-enable-performance-insights`
		
		`--performance-insights-retention-period`
		
		`--performance-insights-kms-key-id`
		
		Parameter API RDS:
		
		`EnablePerformanceInsights`
		
		`PerformanceInsightsRetentionPeriod`
		
		`PerformanceInsightsKMSKeyId`

Pengaturan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS

IOPS yang Tersedia	Jumlah IOPS yang Tersedia (operasi input/output per detik) yang akan dialokasikan di awal untuk klaster basis data.	Opsi CLI:
		`--iops`
		Parameter API RDS:
		`Iops`

Pengaturan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS

Akses publik	Dapat diakses publik untuk memberikan alamat IP publik ke klaster DB, yang berarti bahwa klaster dapat diakses di luar VPC. Agar dapat diakses publik, klaster DB juga harus berada di subnet publik di VPC.
	
	Tidak dapat diakses publik agar klaster DB hanya dapat diakses dari dalam VPC.
	
	Untuk informasi selengkapnya, lihat Menyembunyikan klaster DB dalam VPC dari internet.
	
	Untuk terhubung ke klaster DB dari luar VPC, klaster DB harus dapat diakses publik. Selain itu, akses harus diberikan menggunakan aturan masuk grup keamanan klaster DB, dan persyaratan lain harus terpenuhi. Untuk informasi selengkapnya, lihat Tidak dapat terhubung ke instans DB Amazon RDS.
	
	Jika kluster DB Anda tidak dapat diakses publik, Anda dapat menggunakan koneksi VPN AWS Site-to-Site AWS Direct Connect atau koneksi untuk mengakses nya dari jaringan pribadi. Untuk
		Opsi CLI:
		
		`--publicly-accessible`
		
		`--no-publicly-accessible`
		
		Parameter API RDS:
		
		`PubliclyAccessible`

Pengaturan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS
	informasi selengkapnya, lihat Privasi lalu lintas jaringan internet.	

RDS Extended Support	Pilih Aktifkan RDS Extended Support untuk memungkinkan versi mesin utama yang didukung untuk terus berjalan melewati akhir RDS dari tanggal dukungan standar.
	
	Saat Anda membuat cluster DB, Amazon RDS default ke RDS Extended Support. Untuk mencegah pembuatan cluster DB baru setelah RDS berakhir pada tanggal dukungan standar dan untuk menghindari biaya untuk RDS Extended Support, nonaktifkan pengaturan ini. Cluster DB Anda yang ada tidak akan dikenakan biaya hingga tanggal mulai penetapan harga RDS Extended Support.
	
	Untuk informasi selengkapnya, lihat Menggunakan Dukungan Diperpanjang Amazon RDS.
		Opsi CLI:
		
		`--engine-lifecycle-support`
		
		Parameter API RDS:
		
		`EngineLifecycleSupport`

Pengaturan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS
Throughput penyimpanan	Nilai throughput penyimpanan untuk cluster DB. Pengaturan ini hanya terlihat jika Anda memilih General Purpose SSD (gp3) untuk jenis penyimpanan. Pengaturan ini tidak dapat dikonfigurasi dan diatur secara otomatis berdasarkan IOPS yang Anda tentukan. Untuk informasi selengkapnya, lihat Penyimpanan gp3.	Nilai ini dihitung secara otomatis dan tidak memiliki opsi CLI.
Proksi RDS	Pilih Buat Proksi RDS guna membuat proksi untuk klaster DB Anda. Amazon RDS secara otomatis membuat peran IAM dan rahasia Secrets Manager untuk proksi.	Tidak tersedia saat membuat klaster DB.

Jenis penyimpanan	Jenis penyimpanan untuk klaster DB Anda.
	
	Hanya penyimpanan General Purpose SSD (gp3), Provisioned IOPS (io1), dan Provisioned IOPS SSD (io2) yang didukung.
	
	Untuk informasi selengkapnya, lihat Jenis penyimpanan Amazon RDS.
		Opsi CLI:
		
		`--storage-type`
		
		Parameter API RDS:
		
		`StorageType`

Pengaturan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS
Cloud Privat Virtual (VPC)	VPC berbasis layanan Amazon VPC yang akan dihubungkan dengan klaster DB ini. Untuk informasi selengkapnya, lihat Amazon VPC dan Amazon RDS.	Untuk CLI dan API, Anda menentukan ID grup keamanan VPC.

Grup keamanan VPC (firewall)	Grup keamanan untuk dikaitkan dengan klaster DB.
	
	Untuk informasi selengkapnya, lihat Ikhtisar grup keamanan VPC.
		Opsi CLI:
		
		`--vpc-security-group-ids`
		
		Parameter API RDS:
		
		`VpcSecurityGroupIds`

Pengaturan yang tidak berlaku saat membuat klaster DB Multi-AZ

Pengaturan berikut dalam AWS CLI perintah `create-db-cluster`dan operasi RDS API `CreateDBCluster`tidak berlaku untuk cluster DB multi-AZ.

Anda juga tidak dapat menentukan pengaturan ini untuk klaster DB Multi-AZ di konsol.

AWS CLI pengaturan	Pengaturan API RDS
`--availability-zones`	`AvailabilityZones`
`--backtrack-window`	`BacktrackWindow`
`--character-set-name`	`CharacterSetName`
`--domain`	`Domain`
`--domain-iam-role-name`	`DomainIAMRoleName`

AWS CLI pengaturan	Pengaturan API RDS
`--enable-global-write-forwarding | --no-enable-global-write-forwarding`	`EnableGlobalWriteForwarding`
`--enable-http-endpoint | --no-enable-http-endpoint`	`EnableHttpEndpoint`
`--enable-iam-database-authentication | --no-enable-iam-database-authentication`	`EnableIAMDatabaseAuthentication`
`--global-cluster-identifier`	`GlobalClusterIdentifier`
`--option-group-name`	`OptionGroupName`
`--pre-signed-url`	`PreSignedUrl`
`--replication-source-identifier`	`ReplicationSourceIdentifier`
`--scaling-configuration`	`ScalingConfiguration`

Menghubungi klaster basis data Multi-AZ

Sebuah klaster basis data Multi-AZ memiliki tiga instans basis data, bukan instans basis data tunggal. Setiap koneksi ditangani oleh instans basis data tertentu. Saat Anda menghubungi klaster basis data Multi-AZ, nama host dan porta yang Anda tentukan menunjuk ke nama domain berkualifikasi penuh yang disebut dengan titik akhir. Klaster basis data Multi-AZ menggunakan mekanisme titik akhir untuk mengabstrakkan koneksi ini sehingga Anda tidak perlu menentukan instans basis data di klaster basis data yang akan dihubungi. Dengan demikian, Anda tidak perlu mencantumkan nama persis/hardcode semua nama host atau menulis logika Anda sendiri untuk merutekan ulang koneksi saat beberapa instans basis data tidak tersedia.

Titik akhir penulis menghubungi instans basis data penulis klaster basis data, yang mendukung operasi-operasi baca dan tulis. Titik akhir pembaca menghubungi salah satu dari dua instans basis data pembaca, yang hanya mendukung operasi baca.

Dengan menggunakan titik akhir, Anda dapat memetakan setiap koneksi ke instans basis data atau grup instans basis data yang layak berdasarkan kasus penggunaan Anda. Misalnya, untuk melakukan pernyataan-pernyataan DDL dan DML, Anda dapat menghubungi instans basis data mana pun yang merupakan instans basis data penulis. Untuk melakukan kueri, Anda dapat menghubungi titik akhir pembaca, dengan klaster basis data Multi-AZ mengelola secara otomatis koneksi di antara instans-instans basis data pembaca. Untuk diagnosis atau penyetelan, Anda dapat menghubungi titik akhir instans basis data tertentu untuk memeriksa detail instans basis data itu.

Lihat informasi yang lebih lengkap tentang cara menghubungi instans basis data di Menghubungkan ke instans DB Amazon RDS.

Topik

- Jenis-jenis titik akhir klaster basis data Multi-AZ
- Melihat titik akhir untuk klaster basis data Multi-AZ
- Menggunakan titik akhir klaster
- Menggunakan titik akhir pembaca
- Menggunakan titik akhir instans
- Cara titik akhir basis data Multi-AZ bekerja dengan ketersediaan tinggi

Jenis-jenis titik akhir klaster basis data Multi-AZ

Titik akhir diwakili oleh sebuah pengidentifikasi unik yang berisi alamat host. Jenis-jenis titik akhir berikut tersedia dari klaster basis data Multi-AZ:

Titik akhir klaster

Titik akhir klaster (atau titik akhir penulis) untuk klaster basis data Multi-AZ menghubungi instans basis data penulis saat ini untuk klaster basis data itu. Titik akhir ini adalah satu-satunya titik akhir yang dapat melakukan operasi tulis seperti pernyataan-pernyataan DDL dan DML. Titik akhir ini juga dapat melakukan operasi baca.

Setiap klaster basis data Multi-AZ memiliki satu titik akhir klaster dan satu instans basis data penulis.

Anda menggunakan titik akhir klaster untuk semua operasi tulis pada klaster basis data itu, yang meliputi penyisipan, pembaruan, penghapusan, dan perubahan DDL. Anda juga dapat menggunakan titik akhir klaster untuk operasi baca, seperti kueri.

Jika instans basis data penulis saat ini suatu klaster basis data gagal, klaster basis data Multi-AZ melakukan secara otomatis pindah saat gagal/failover ke instans basis data penulis baru. Selama pindah saat gagal/failover, klaster basis data itu terus melayani permintaan koneksi ke titik akhir klaster penulis baru instans basis data itu, dengan pemutusan layanan yang minimal.

Contoh berikut mengilustrasikan titik akhir klaster untuk suatu klaster basis data Multi-AZ.

```
mydbcluster.cluster-123456789012.us-east-1.rds.amazonaws.com
```

Titik akhir pembaca

Titik akhir pembaca untuk suatu klaster basis data Multi-AZ memberikan dukungan untuk koneksi hanya baca ke klaster basis data. Gunakan titik akhir pembaca untuk operasi-operasi baca, seperti kueri `SELECT`. Dengan memproses pernyataan-pernyataan itu pada instans basis data pembaca, titik akhir ini mengurangi sisihan umum/overhead pada instans basis data penulis. Titik akhir juga membantu klaster menskalakan kapasitas untuk menangani kueri `SELECT` simultan. Setiap klaster basis data Multi-AZ memiliki satu titik akhir pembaca.

Titik akhir pembaca mengirimkan setiap permintaan koneksi ke salah satu instans basis data pembaca. Saat menggunakan titik akhir pembaca untuk suatu sesi, Anda hanya dapat melakukan pernyataan hanya baca seperti `SELECT` dalam sesi itu.

Contoh berikut mengilustrasikan titik akhir pembaca untuk suatu klaster basis data Multi-AZ. Maksud hanya baca suatu titik akhir pembaca dilambangkan oleh `-ro` di dalam nama titik akhir klaster.

```
mydbcluster.cluster-ro-123456789012.us-east-1.rds.amazonaws.com
```

Titik akhir instans

Titik akhir instans menghubungi instans basis data tertentu dalam klaster basis data Multi-AZ. Setiap instans basis data dalam sebuah klaster basis data memiliki titik akhir yang unik. Jadi, ada satu titik akhir instans untuk instans basis data penulis saat ini di klaster basis data, dan ada satu titik akhir instans untuk setiap instans basis data pembaca di klaster basis data.

Titik akhir instans menyediakan kendali langsung atas koneksi ke klaster basis data. Kendali ini dapat membantu Anda menangani skenario-skenario ketika penggunaan titik akhir klaster atau titik akhir pembaca mungkin tidak layak. Misalnya, aplikasi klien Anda mungkin meminta penyeimbangan beban yang lebih terurai halus berdasarkan jenis beban kerja. Dalam hal ini, Anda dapat mengonfigurasikan beberapa klien untuk menghubungi instans basis data pembaca yang berbeda dalam sebuah klaster basis data untuk menyebarkan beban kerja baca.

Contoh berikut mengilustrasikan titik akhir instans untuk sebuah instans basis data di dalam klaster basis data Multi-AZ.

```
mydbinstance.123456789012.us-east-1.rds.amazonaws.com
```

Melihat titik akhir untuk klaster basis data Multi-AZ

Di AWS Management Console, Anda melihat titik akhir klaster dan titik akhir pembaca di halaman detail untuk masing-masing klaster basis data Multi-AZ. Anda melihat titik akhir instans di halaman detail untuk masing-masing instans basis data.

Dengan AWS CLI, Anda melihat titik akhir penulis dan pembaca dalam output perintah describe-db-clusters. Misalnya, perintah berikut menampilkan atribut-atribut titik akhir untuk semua klaster di Kawasan AWS Anda saat ini.

```
aws rds describe-db-cluster-endpoints
```

Dengan API Amazon RDS, Anda mengambil informasi titik akhir dengan memanggil tindakan DescribeDBClusterEndpoints. Outputnya juga menampilkan titik akhir klaster basis data Amazon Aurora, jika ada.

Menggunakan titik akhir klaster

Setiap klaster basis data Multi-AZ memiliki satu titik akhir klaster bawaan, yang nama dan atribut-atribut lainnya dikelola oleh Amazon RDS. Anda tidak dapat membuat, menghapus, atau mengubah titik akhir jenis ini.

Anda menggunakan titik akhir klaster saat mengelola klaster basis data, melakukan ekstraksi, mengubah, operasi pemuatan (ETL), atau mengembangkan dan menguji aplikasi. Titik akhir klaster menghubungi instans basis data penulis klaster itu'. Instans basis data penulis adalah satu-satunya instans basis data tempat Anda dapat membuat tabel dan indeks, menjalankan pernyataan `INSERT`, dan melakukan operasi DDL dan DML yang lain.

Alamat IP fisik yang ditunjuk oleh titik akhir klaster berubah saat mekanisme pindah saat gagal/failover mempromosikan instans basis data baru menjadi instans basis data penulis bagi klaster. Jika Anda menggunakan sebentuk penghimpunan koneksi atau multipleks lain, bersiaplah untuk menggelontor/flush atau mengurangi waktu untuk hidup (time-to-live) bagi informasi DNS yang tersimpan di cache. Melakukan hal itu memastikan bahwa Anda tidak mencoba membentuk koneksi baca/tulis ke instans basis data yang menjadi tidak tersedia atau kini hanya baca setelah terjadi pindah saat gagal/failover.

Menggunakan titik akhir pembaca

Anda menggunakan titik akhir pembaca untuk koneksi hanya baca dengan klaster basis data Multi-AZ. Titik akhir ini membantu klaster basis data Anda menangani beban kerja padat kueri. Titik akhir pembaca adalah titik akhir yang Anda berikan ke aplikasi yang melakukan pelaporan atau operasi hanya baca lain pada klaster. Titik akhir pembaca mengirimkan koneksi ke instans basis data pembaca yang tersedia di klaster basis data Multi-AZ.

Setiap klaster Multi-AZ memiliki satu titik akhir pembaca bawaan, yang nama dan atribut-atribut lainnya dikelola oleh Amazon RDS. Anda tidak dapat membuat, menghapus, atau mengubah titik akhir jenis ini.

Menggunakan titik akhir instans

Setiap instans basis data dalam klaster basis data Multi-AZ memiliki titik akhir instans bawaan, yang nama dan atribut-atribut lainnya dikelola oleh Amazon RDS. Anda tidak dapat membuat, menghapus, atau mengubah titik akhir jenis ini. Dengan klaster basis data Multi-AZ, Anda biasanya lebih sering menggunakan titik akhir penulis dan pembaca daripada titik akhir instans.

Dalam operasi sehari-hari, cara utama Anda menggunakan titik akhir instans adalah untuk mendiagnosis permasalahan kapasitas atau kinerja yang memengaruhi satu instans basis data tertentu dalam klaster basis data Multi-AZ. Selagi terhubung dengan instans basis data tertentu, Anda dapat memeriksa variabel status, metrik, dan sebagainya instans itu. Melakukan hal itu membantu Anda menentukan apa yang terjadi untuk instans basis data itu yang berbeda dengan apa yang terjadi untuk instans-instans basis data yang lain di klaster.

Cara titik akhir basis data Multi-AZ bekerja dengan ketersediaan tinggi

Untuk klaster basis data Multi-AZ dengan ketersediaan tinggi adalah penting, gunakan titik akhir penulis untuk koneksi baca/tulis atau koneksi umum dan titik akhir pembaca untuk koneksi hanya baca. Titik akhir penulis dan pembaca mengelola pindah saat gagal/failover instans basis data dengan lebih baik daripada titik akhir instans. Tidak seperti titik akhir instans, titik akhir penulis dan titik akhir pembaca mengubah secara otomatis instans basis data yang dihubungi jika instans basis data di klaster menjadi tidak tersedia.

Jika instans basis data penulis sebuah klaster basis data gagal, Amazon RDS melakukan secara otomatis pindah saat gagal/failover ke instans basis data penulis baru. Sistem melakukannya dengan mempromosikan instans basis data pembaca menjadi instans basis data penulis baru. Jika terjadi pindah saat gagal/failover, Anda dapat menggunakan titik akhir penulis untuk menghubungi kembali instans basis data penulis yang baru dipromosikan. Atau Anda dapat menggunakan titik akhir pembaca untuk menghubungi kembali salah satu instans basis data pembaca di klaster basis data. Selama pindah saat gagal/failover, Titik akhir pembaca mungkin akan mengarahkan sejenak koneksi ke instans basis data penulis baru klaster basis data setelah instans basis data pembaca dipromosikan menjadi instans basis data penulis baru. Jika Anda merancang logika aplikasi Anda sendiri untuk mengelola koneksi titik akhir instans, Anda dapat menemukan secara manual atau programatis set instans basis data yang dihasilkan yang tersedia di klaster basis data.

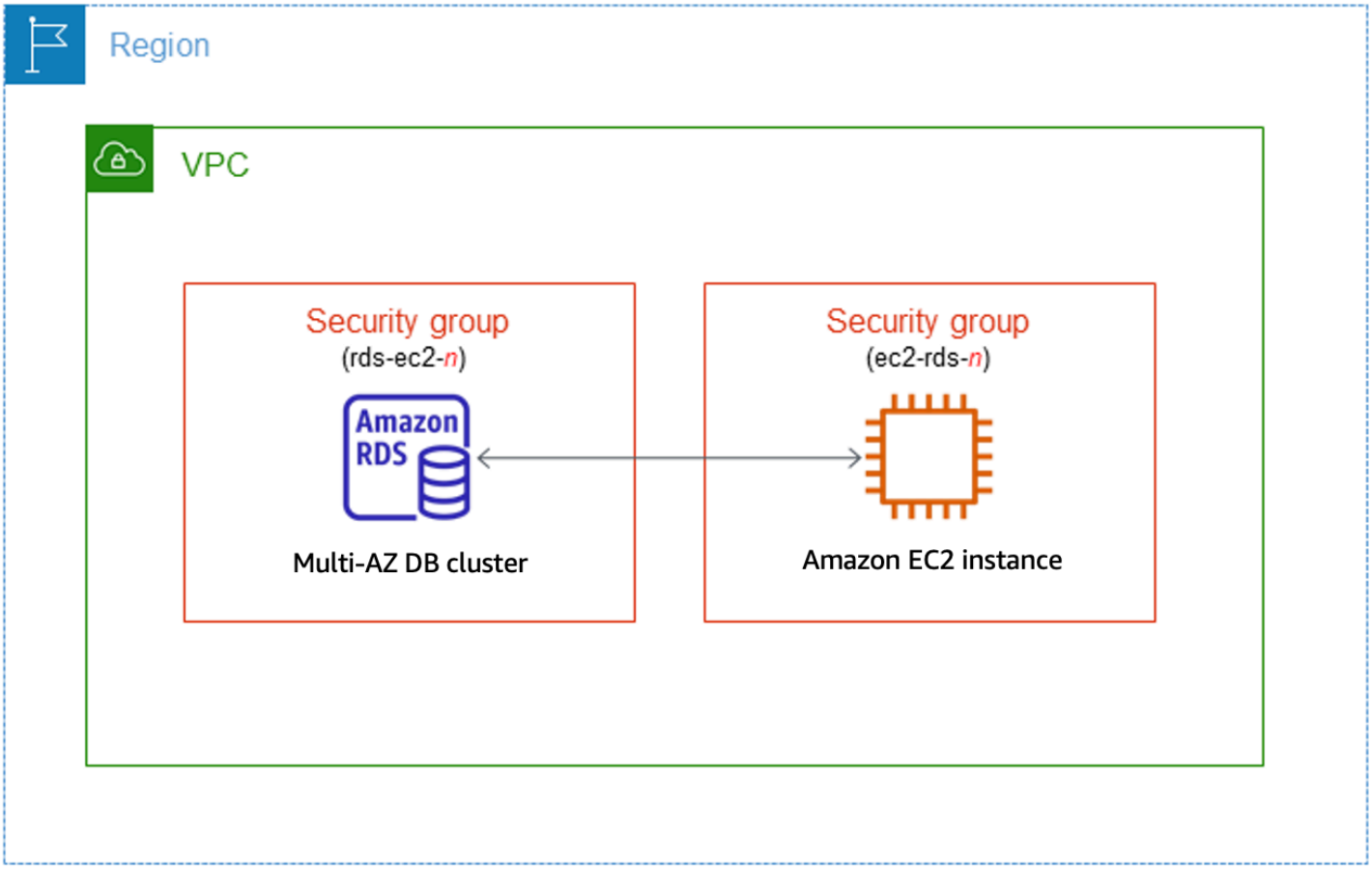
Menghubungkan secara otomatis sumber daya komputasi AWS dan klaster basis data Multi-AZ

Anda dapat menghubungkan secara otomatis klaster basis data Multi-AZ dan sumber daya komputasi AWS seperti instans dan fungsi AWS Lambda Amazon Elastic Compute Cloud (Amazon EC2).

Topik

Menghubungkan secara otomatis instans EC2 dan klaster basis data Multi-AZ

Anda dapat menggunakan konsol Amazon RDS untuk menyederhanakan penyiapan koneksi antara instans Amazon Elastic Compute Cloud (Amazon EC2) dan klaster basis data Multi-AZ. Sering kali, klaster basis data Multi-AZ Anda berada di subnet privat dan instans EC2 Anda berada dalam VPC di subnet publik. Anda dapat menggunakan klien SQL pada instans EC2 untuk menghubungi klaster basis data Multi-AZ. Instans EC2 juga dapat menjalankan server atau aplikasi web yang mengakses klaster basis data Multi-AZ privat Anda.



Jika Anda ingin menghubungkan instans EC2 yang tidak berada di VPC yang sama dengan klaster basis data Multi-AZ, lihat skenario-skenario di the section called “Skenario untuk mengakses instans DB di VPC”.

Topik

- Ikhtisar konektivitas otomatis dengan instans EC2
- Menghubungkan secara otomatis instans EC2 dan klaster basis data Multi-AZ
- Melihat sumber daya komputasi terhubung

Ikhtisar konektivitas otomatis dengan instans EC2

Saat Anda menyiapkan koneksi antara instans EC2 dan klaster basis data Multi-AZ secara otomatis, Amazon RDS mengonfigurasikan grup keamanan VPC untuk instans EC2 Anda dan untuk klaster basis data Anda.

Berikut adalah persyaratan untuk menghubungkan instans EC2 dengan klaster basis data Multi-AZ:

- Instans EC2 harus ada di VPC yang sama dengan klaster basis data Multi-AZ.

  Jika tidak ada instans EC2 di VPC yang sama, maka konsol menyediakan tautan untuk membuatnya.
- Pengguna yang menyiapkan konektivitas harus memiliki izin untuk melakukan operasi-operasi EC2 berikut:
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:AuthorizeSecurityGroupIngress`
  - `ec2:CreateSecurityGroup`
  - `ec2:DescribeInstances`
  - `ec2:DescribeNetworkInterfaces`
  - `ec2:DescribeSecurityGroups`
  - `ec2:ModifyNetworkInterfaceAttribute`
  - `ec2:RevokeSecurityGroupEgress`

Saat Anda menyiapkan koneksi dengan instans EC2, Amazon RDS bertindak sesuai dengan konfigurasi grup keamanan saat ini yang terkait dengan klaster basis data Multi-AZ dan instans EC2, seperti dijelaskan dalam tabel berikut.

Konfigurasi grup keamanan RDS saat ini	Konfigurasi grup keamanan EC2 saat ini	Tindakan RDS
Ada satu atau beberapa grup keamanan yang terkait dengan klaster basis data Multi-AZ dengan nama yang cocok dengan pola `rds-ec2-`*n* (dengan *n* berupa angka). Grup keamanan yang cocok dengan pola belum diubah. Grup keamanan ini memiliki hanya satu aturan masuk dengan grup	Ada satu atau beberapa grup keamanan yang terkait dengan instans EC2 dengan nama yang cocok dengan pola `rds-ec2-`*n* (dengan *n* berupa angka). Grup keamanan yang cocok dengan pola belum diubah. Grup keamanan ini hanya memiliki satu aturan keluar dengan grup keamanan VPC klaster basis data Multi-AZ sebagai sumbernya.	Amazon RDS tidak mengambil tindakan. Koneksi sudah dikonfigu rasikan secara otomatis antara instans EC2 dan klaster basis data Multi-AZ. Karena koneksi sudah ada antara instans EC2 dan basis data RDS, grup keamanan tidak diubah.

Konfigurasi grup keamanan RDS saat ini	Konfigurasi grup keamanan EC2 saat ini	Tindakan RDS
keamanan VPC instans EC2 sebagai sumbernya.		

Konfigurasi grup keamanan RDS saat ini	Konfigurasi grup keamanan EC2 saat ini	Tindakan RDS
Salah satu syarat berikut dipenuhi: • Tidak ada grup keamanan yang terkait dengan klaster basis data Multi-AZ dengan nama yang cocok dengan pola `rds-ec2-`*n*. • Ada satu atau beberapa grup keamanan yang terkait dengan klaster basis data multi-AZ dengan nama yang cocok dengan pola `rds-ec2-`*n*. Namun, tidak satu pun grup keamanan ini dapat digunakan untuk koneksi dengan instans EC2. Grup keamanan tidak dapat digunakan jika tidak memiliki satu aturan masuk dengan grup keamanan VPC instans EC2 sebagai sumbernya. Grup keamanan juga tidak dapat digunakan jika telah diubah. Contoh-contoh perubahan meliputi penambahan aturan atau pengubahan port aturan yang ada.	Salah satu syarat berikut dipenuhi: • Tidak ada grup keamanan yang terkait dengan instans EC2 dengan nama yang cocok dengan pola `ec2-rds-`*n*. • Ada satu atau beberapa grup keamanan yang terkait dengan instans EC2 dengan nama yang cocok dengan pola `ec2-rds-`*n*. Namun, tidak satu pun grup keamanan ini dapat digunakan untuk koneksi dengan klaster basis data Multi-AZ. Grup keamanan tidak dapat digunakan jika tidak memiliki satu aturan keluar dengan grup keamanan VPC klaster basis data Multi-AZ sebagai sumbernya. Grup keamanan juga tidak dapat digunakan jika telah diubah.	RDS action: create new security groups

Konfigurasi grup keamanan RDS saat ini	Konfigurasi grup keamanan EC2 saat ini	Tindakan RDS
Ada satu atau beberapa grup keamanan yang terkait dengan klaster basis data multi-AZ dengan nama yang cocok dengan pola `rds-ec2-`*n*. Grup keamanan yang cocok dengan pola belum diubah. Grup keamanan ini memiliki hanya satu aturan masuk dengan grup keamanan VPC instans EC2 sebagai sumbernya.	Ada satu atau beberapa grup keamanan yang terkait dengan instans EC2 dengan nama yang cocok dengan pola `ec2-rds-`*n*. Namun, tidak satu pun grup keamanan ini dapat digunakan untuk koneksi dengan klaster basis data Multi-AZ. Grup keamanan tidak dapat digunakan jika tidak memiliki satu aturan keluar dengan grup keamanan VPC klaster basis data Multi-AZ sebagai sumbernya. Grup keamanan juga tidak dapat digunakan jika telah diubah.	RDS action: create new security groups
Ada satu atau beberapa grup keamanan yang terkait dengan klaster basis data multi-AZ dengan nama yang cocok dengan pola `rds-ec2-`*n*. Grup keamanan yang cocok dengan pola belum diubah. Grup keamanan ini memiliki hanya satu aturan masuk dengan grup keamanan VPC instans EC2 sebagai sumbernya.	Ada grup keamanan EC2 yang valid untuk koneksi, tetapi tidak terkait dengan instans EC2. Grup keamanan ini memiliki nama yang cocok dengan pola `rds-ec2-`*n*. Grup itu belum diubah. Grup ini hanya memiliki satu aturan keluar dengan grup keamanan VPC klaster basis data Multi-AZ sebagai sumbernya.	RDS action: associate EC2 security group

Konfigurasi grup keamanan RDS saat ini	Konfigurasi grup keamanan EC2 saat ini	Tindakan RDS
Salah satu syarat berikut dipenuhi: • Tidak ada grup keamanan yang terkait dengan klaster basis data Multi-AZ dengan nama yang cocok dengan pola `rds-ec2-`*n*. • Ada satu atau beberapa grup keamanan yang terkait dengan klaster basis data multi-AZ dengan nama yang cocok dengan pola `rds-ec2-`*n*. Namun, tidak satu pun grup keamanan ini dapat digunakan untuk koneksi dengan instans EC2. Grup keamanan tidak dapat digunakan jika tidak memiliki satu aturan masuk dengan grup keamanan VPC instans EC2 sebagai sumbernya. Grup keamanan juga tidak dapat digunakan jika telah diubah.	Ada satu atau beberapa grup keamanan yang terkait dengan instans EC2 dengan nama yang cocok dengan pola `rds-ec2-`*n*. Grup keamanan yang cocok dengan pola belum diubah. Grup keamanan ini hanya memiliki satu aturan keluar dengan grup keamanan VPC klaster basis data Multi-AZ sebagai sumbernya.	RDS action: create new security groups

Tindakan RDS : membuat grup keamanan baru

Amazon RDS melakukan tindakan-tindakan berikut:

- Membuat grup keamanan baru yang cocok dengan pola `rds-ec2-`*n*. Grup keamanan ini memiliki aturan masuk dengan grup keamanan VPC instans EC2 sebagai sumbernya. Grup keamanan ini dikaitkan dengan klaster basis data Multi-AZ dan memungkinkan instans EC2 mengakses klaster basis data Multi-AZ.
- Membuat grup keamanan baru yang cocok dengan pola `ec2-rds-`*n*. Grup keamanan ini memiliki aturan keluar dengan grup keamanan VPC klaster basis data Multi-AZ sebagai sumbernya. Grup keamanan ini dikaitkan dengan instans EC2 dan memungkinkan instans EC2 mengirim lalu lintas ke klaster basis data Multi-AZ.

Tindakan RDS : mengaitkan grup keamanan EC2

Amazon RDS mengaitkan grup keamanan EC2 yang valid dan sudah ada dengan instans EC2. Grup keamanan ini memungkinkan instans EC2 mengirim lalu lintas ke klaster basis data Multi-AZ.

Menghubungkan secara otomatis instans EC2 dan klaster basis data Multi-AZ

Sebelum menyiapkan koneksi antara instans EC2 dan basis data RDS, pastikan untuk memenuhi persyaratan yang dijelaskan di Ikhtisar konektivitas otomatis dengan instans EC2.

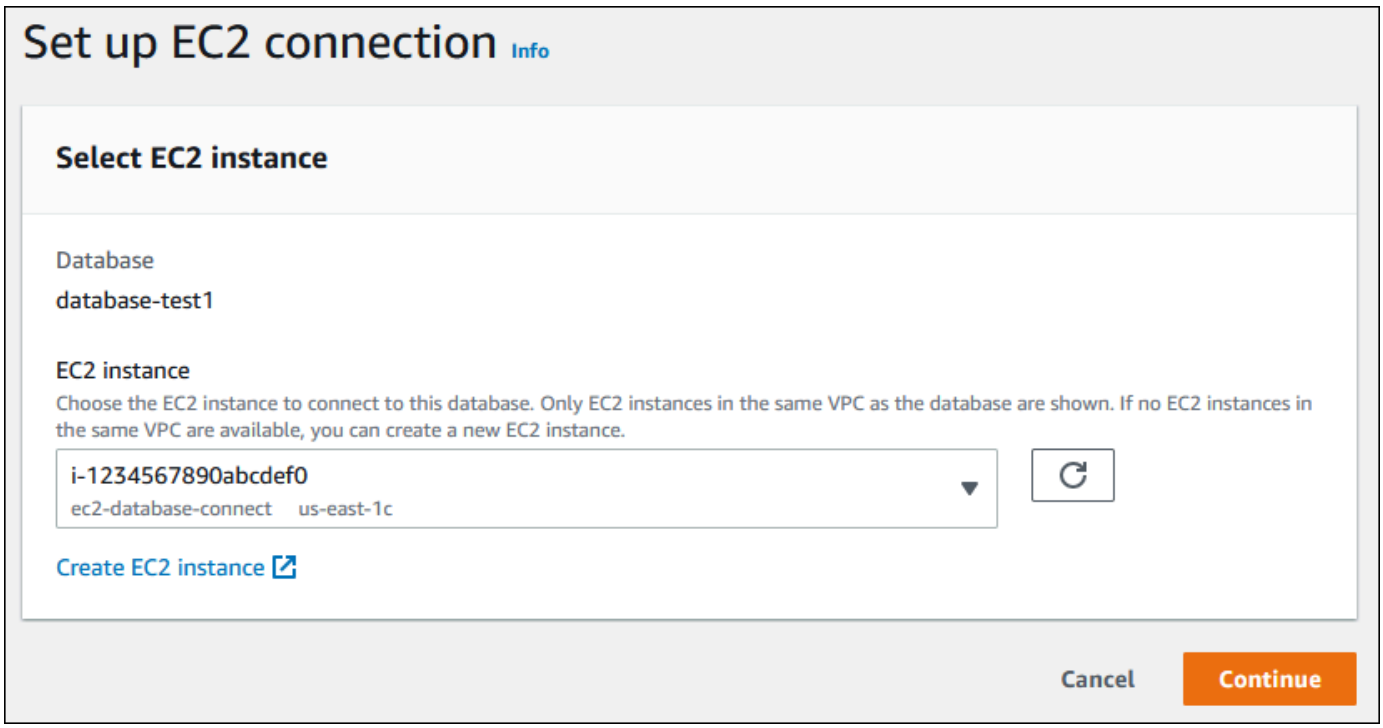
Jika Anda membuat perubahan pada grup keamanan setelah mengonfigurasikan konektivitas, perubahan itu dapat memengaruhi koneksi antara instans EC2 dan basis data RDS.

> **Note**
>
> Anda hanya dapat menyiapkan koneksi antara instans EC2 dan basis data RDS secara otomatis dengan menggunakan AWS Management Console. Anda tidak dapat mengatur koneksi secara otomatis dengan AWS CLI atau RDS API.

Untuk menghubungkan secara otomatis instans EC2 dan basis data RDS

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data, lalu pilih basis data RDS.
3. Untuk Tindakan, pilih Siapkan koneksi EC2.

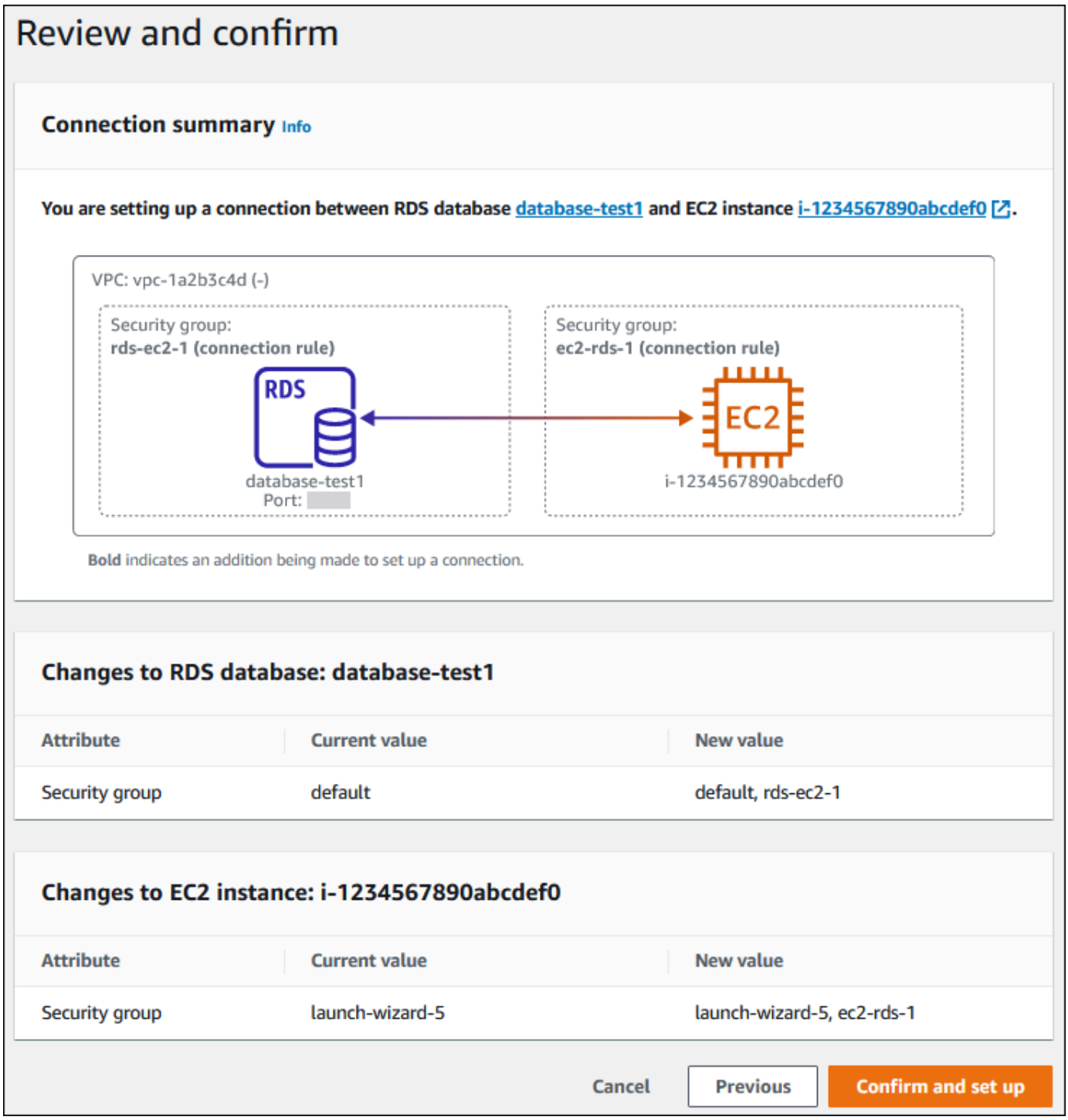
   Halaman Siapkan koneksi EC2 muncul.
4. Pada halaman Siapkan koneksi EC2, pilih instans EC2.



Jika tidak ada instans EC2 di VPC yang sama, pilih Buat instans EC2 untuk membuatnya. Dalam hal ini, pastikan bahwa instans EC2 baru berada di VPC yang sama dengan basis data RDS.

5. Pilih Lanjutkan.

   Halaman Tinjau dan tegaskan muncul.



6. Pada halaman Tinjau dan tegaskan, tinjau perubahan yang akan dilakukan RDS untuk menyiapkan konektivitas dengan instans EC2.

   Jika perubahan sudah benar, pilih Tegaskan dan siapkan.

   Jika perubahan masih salah, pilih Sebelumnya atau Batalkan.

Melihat sumber daya komputasi terhubung

Anda dapat menggunakan AWS Management Console untuk melihat sumber daya komputasi yang terhubung ke database RDS DB cluster. Sumber daya yang ditampilkan meliputi koneksi sumber daya komputasi yang disiapkan secara otomatis. Anda dapat menyiapkan secara otomatis konektivitas dengan sumber daya komputasi dengan cara berikut:

- Anda dapat memilih sumber daya komputasi saat membuat basis data.

  Lihat informasi yang lebih lengkap di Membuat instans DB Amazon RDS dan Membuat klaster DB Multi-AZ.
- Anda dapat menyiapkan konektivitas antara basis data yang ada dan sumber daya komputasi.

  Untuk informasi selengkapnya, lihat Menghubungkan secara otomatis instans EC2 dan basis data RDS.

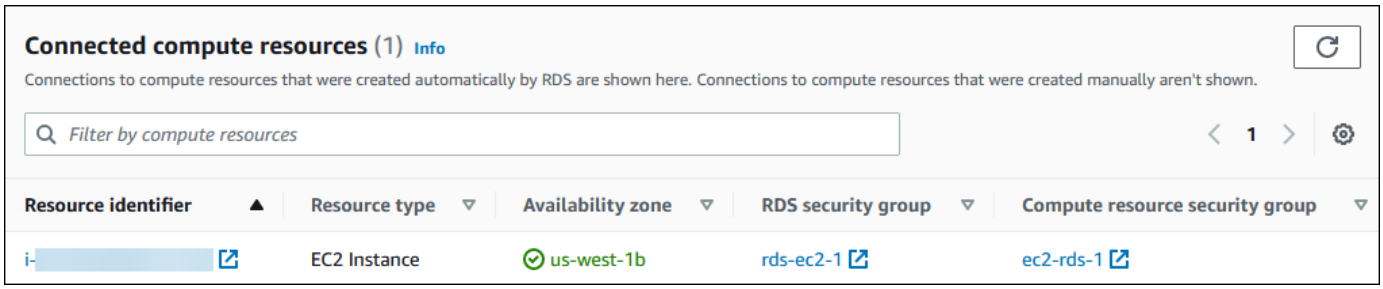
Sumber daya komputasi yang tercantum tidak menyertakan sumber daya yang dihubungkan secara manual dengan basis data. Misalnya, Anda dapat mengizinkan sumber daya komputasi untuk mengakses basis data secara manual dengan menambahkan aturan ke grup keamanan VPC yang terkait dengan basis data.

Agar sumber daya komputasi tercantum, syarat-syarat berikut harus dipenuhi:

- Nama grup keamanan yang terkait dengan sumber daya komputasi cocok dengan pola `ec2-rds-`*n* (dengan *n* berupa angka).
- Grup keamanan yang terkait dengan sumber daya komputasi memiliki aturan keluar dengan rentang port diatur ke port yang digunakan basis data RDS.
- Grup keamanan yang terkait dengan sumber daya komputasi memiliki aturan keluar dengan sumber yang diatur ke grup keamanan yang terkait dengan basis data RDS.
- Nama grup keamanan yang terkait dengan basis data RDS cocok dengan pola `rds-ec2-`*n* (dengan *n* berupa angka).
- Grup keamanan yang terkait dengan basis data RDS memiliki aturan masuk dengan rentang port yang diatur ke port yang digunakan basis data RDS.
- Grup keamanan yang terkait dengan basis data RDS memiliki aturan masuk dengan sumber yang diatur ke grup keamanan yang terkait dengan sumber daya komputasi.

Untuk melihat sumber daya komputasi yang menghubungi basis data RDS

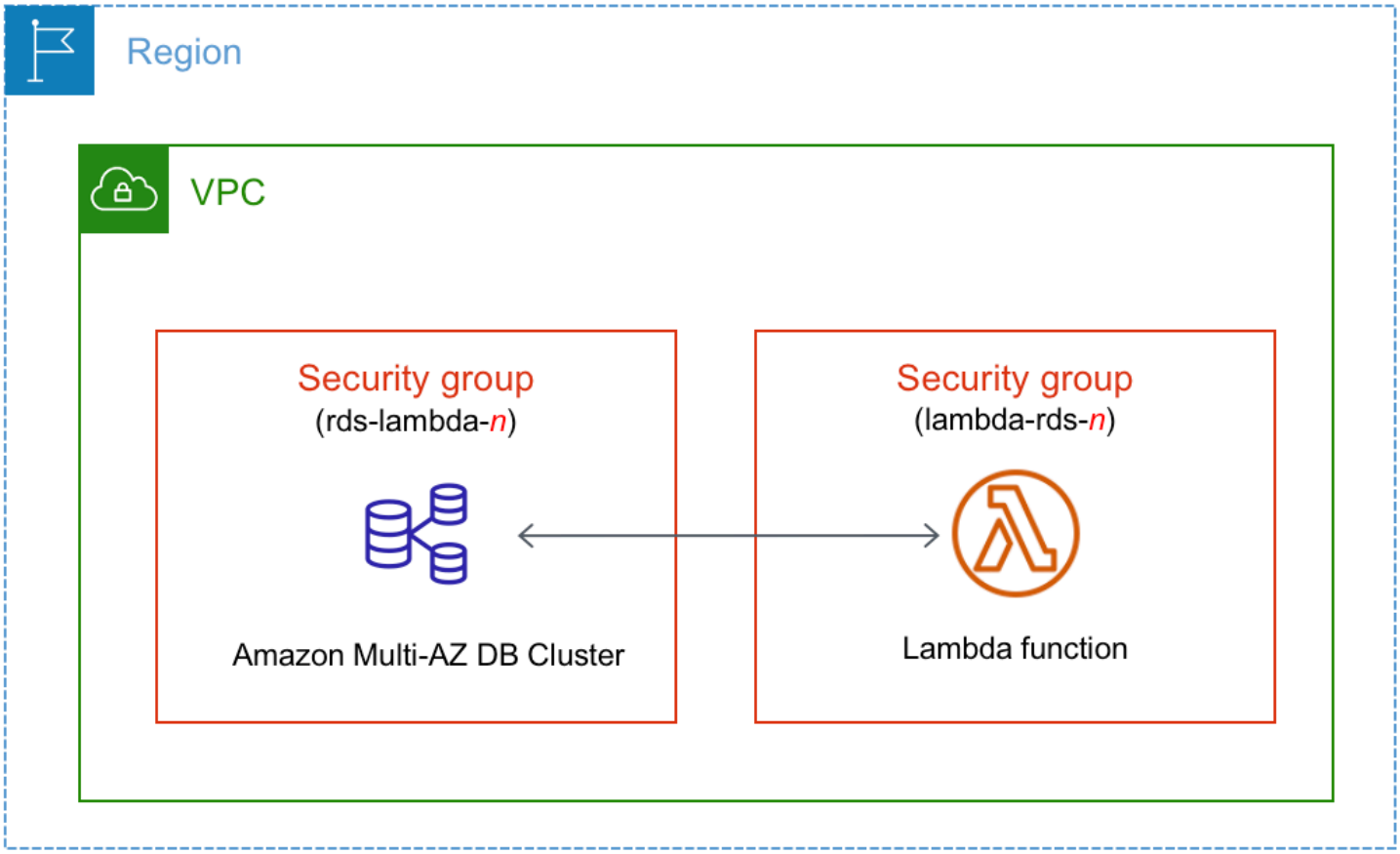
1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data, lalu pilih nama basis data RDS.
3. Pada tab Konektivitas dan keamanan, lihat sumber daya komputasi di Sumber daya komputasi terhubung.



Menghubungkan secara otomatis fungsi Lambda dan klaster basis data Multi-AZ

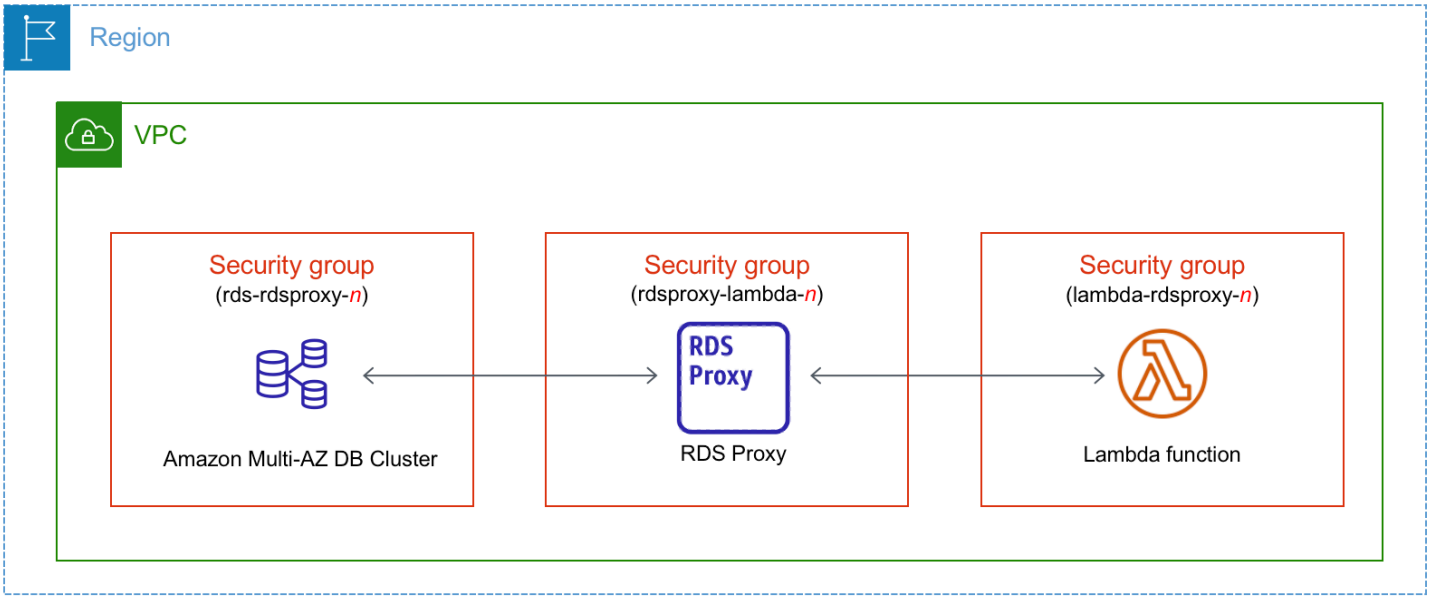
Anda dapat menggunakan konsol RDS untuk menyederhanakan penyiapan koneksi antara fungsi Lambda dan klaster basis data Multi-AZ. Anda dapat menggunakan konsol RDS untuk menyederhanakan penyiapan koneksi antara fungsi Lambda dan klaster basis data Multi-AZ. Sering kali, klaster basis data Multi-AZ Anda berada dalam VPC di subnet privat. Fungsi Lambda dapat digunakan oleh aplikasi untuk mengakses klaster basis data Multi-AZ privat Anda.

Gambar berikut menunjukkan koneksi langsung antara klaster basis data Multi-AZ dan fungsi Lambda.



Anda dapat menyiapkan koneksi antara fungsi Lambda dan basis data Anda melalui Proksi RDS untuk meningkatkan kinerja dan ketangguhan basis data Anda. Sering kali, fungsi Lambda membuat koneksi basis data yang singkat tetapi sering yang menarik manfaat dari penghimpunan koneksi yang ditawarkan Proksi RDS. Anda dapat memanfaatkan sebarang autentikasi IAM yang Anda miliki untuk fungsi Lambda, alih-alih mengelola kredensial basis data dalam kode aplikasi Lambda. Lihat informasi yang lebih lengkap di Menggunakan Proksi Amazon RDS.

Anda dapat menggunakan konsol untuk membuat secara otomatis proksi bagi koneksi Anda. Anda juga dapat memilih proksi yang ada. Konsol memperbarui grup keamanan proksi untuk memungkinkan koneksi dari basis data dan fungsi Lambda. Anda dapat memasukkan kredensial basis data Anda atau memilih rahasia Secrets Manager yang Anda perlukan untuk mengakses basis data.



Topik

- Ikhtisar konektivitas otomatis dengan fungsi Lambda
- Menghubungkan secara otomatis fungsi Lambda dan klaster basis data Multi-AZ
- Melihat sumber daya komputasi terhubung

Ikhtisar konektivitas otomatis dengan fungsi Lambda

Saat Anda menyiapkan secara otomatis koneksi antara fungsi Lambda dan klaster basis data Multi-AZ, Amazon RDS mengonfigurasikan grup keamanan VPC untuk fungsi Lambda Anda dan untuk klaster basis data Anda.

Berikut adalah persyaratan untuk menghubungkan fungsi Lambda dengan klaster basis data Multi-AZ:

- Fungsi Lambda harus berada di VPC yang sama dengan klaster basis data Multi-AZ.

  Jika tidak ada fungsi Lambda di VPC yang sama, konsol menyediakan penaut untuk membuatnya.
- Pengguna yang menyiapkan konektivitas harus memiliki izin untuk melakukan operasi-operasi Amazon RDS, Amazon EC2, Lambda, Secrets Manager, dan IAM berikut:
  - Amazon RDS
    - `rds:CreateDBProxies`
    - `rds:DescribeDBInstances`

    - `rds:DescribeDBProxies`
    - `rds:ModifyDBInstance`
    - `rds:ModifyDBProxy`
    - `rds:RegisterProxyTargets`
- Amazon EC2
    - `ec2:AuthorizeSecurityGroupEgress`
    - `ec2:AuthorizeSecurityGroupIngress`
    - `ec2:CreateSecurityGroup`
    - `ec2:DeleteSecurityGroup`
    - `ec2:DescribeSecurityGroups`
    - `ec2:RevokeSecurityGroupEgress`
    - `ec2:RevokeSecurityGroupIngress`
- Lambda
    - `lambda:CreateFunctions`
    - `lambda:ListFunctions`
    - `lambda:UpdateFunctionConfiguration`
- Secrets Manager
    - `sercetsmanager:CreateSecret`
    - `secretsmanager:DescribeSecret`
- IAM
    - `iam:AttachPolicy`
    - `iam:CreateRole`
    - `iam:CreatePolicy`
- AWS KMS
    - `kms:describeKey`

Saat Anda menyiapkan koneksi antara fungsi Lambda dan klaster basis data Multi-AZ, Amazon RDS mengonfigurasikan grup keamanan VPC untuk fungsi Anda dan untuk klaster basis data Multi-AZ Anda. Jika Anda menggunakan Proksi RDS, maka Amazon RDS juga mengonfigurasikan grup keamanan VPC untuk proksi itu. Amazon RDS bertindak sesuai dengan konfigurasi grup keamanan

saat ini yang terkait dengan klaster basis data Multi-AZ dan fungsi Lambda, dan proksi, seperti dijelaskan dalam tabel berikut.

Konfigurasi grup keamanan RDS saat ini	Konfigurasi grup keamanan Lambda saat ini	Konfigurasi grup keamanan proksi saat ini	Tindakan RDS
Amazon RDS tidak mengambil tindakan karena grup keamanan semua sumber daya mengikuti pola penamaan yang benar dan memiliki aturan masuk dan aturan keluar yang tepat.	Ada satu atau beberapa grup keamanan yang terkait dengan klaster basis data Multi-AZ dengan nama yang cocok dengan pola `rds-lambda-` *n* (dengan *n* berupa angka) atau jika `TargetHealth` proksi terkait adalah `AVAILABLE` . Grup keamanan yang cocok dengan pola belum diubah. Grup keamanan ini memiliki hanya satu aturan masuk dengan grup keamanan VPC proksi atau fungsi Lambda sebagai sumbernya.	Ada satu atau beberapa grup keamanan yang terkait dengan fungsi Lambda dengan nama yang cocok dengan pola `lambda-rds-` *n* atau `lambda-rdsproxy-` *n* (dengan *n* berupa angka). Grup keamanan yang cocok dengan pola belum diubah. Grup keamanan ini hanya memiliki satu aturan keluar dengan grup keamanan VPC klaster basis data Multi-AZ atau proksi sebagai tujuannya.	Ada satu atau beberapa grup keamanan yang terkait dengan proksi dengan nama yang cocok dengan pola `rdsproxy-lambda-` *n* (dengan *n* berupa angka). Grup keamanan yang cocok dengan pola belum diubah. Grup keamanan ini memiliki aturan masuk dan aturan keluar dengan grup keamanan VPC fungsi Lambda dan klaster basis data Multi-AZ.
Salah satu syarat berikut dipenuhi: • Tidak ada grup keamanan yang	Salah satu syarat berikut dipenuhi: • Tidak ada grup keamanan yang	Salah satu syarat berikut dipenuhi: • Tidak ada grup keamanan yang	RDS action: create new security groups

Konfigurasi grup keamanan RDS saat ini	Konfigurasi grup keamanan Lambda saat ini	Konfigurasi grup keamanan proksi saat ini	Tindakan RDS
terkait dengan klaster basis data Multi-AZ dengan nama yang cocok dengan pola `rds-lambda-` *n* atau jika `TargetHealth` proksi terkaitnya adalah `AVAILABLE` . • Ada satu atau beberapa grup keamanan yang terkait dengan klaster basis data Multi-AZ dengan nama yang cocok dengan pola `rds-lambda-` *n* atau jika `TargetHealth` proksi terkaitnya adalah `AVAILABLE` . Namun, Amazon RDS tidak dapat menggunakan satu pun grup keamanan ini untuk koneksi dengan fungsi Lambda.	terkait dengan fungsi Lambda dengan nama yang cocok dengan pola `lambda-rds-` *n* atau `lambda-rdsproxy-` *n*. • Ada satu atau beberapa grup keamanan yang terkait dengan fungsi Lambda dengan nama yang cocok dengan pola `lambda-rds-` *n* atau `lambda-rdsproxy-` *n*. Namun, Amazon RDS tidak dapat menggunakan satu pun grup keamanan ini untuk koneksi dengan klaster basis data Multi-AZ. Amazon RDS tidak dapat menggunakan grup keamanan jika tidak memiliki satu aturan keluar dengan grup keamanan VPC	terkait dengan proksi dengan nama yang cocok dengan pola `rdsproxy-lambda-` *n*. • Ada satu atau beberapa grup keamanan yang terkait dengan proksi dengan nama yang cocok dengan `rdsproxy-lambda-` *n*. Namun, Amazon RDS tidak dapat menggunakan satu pun grup keamanan ini untuk koneksi dengan klaster basis data Multi-AZ atau fungsi Lambda. Amazon RDS tidak dapat menggunak an grup keamanan yang tidak memiliki aturan masuk dan keluar dengan grup keamanan VPC	

Konfigurasi grup keamanan RDS saat ini	Konfigurasi grup keamanan Lambda saat ini	Konfigurasi grup keamanan proksi saat ini	Tindakan RDS
Amazon RDS tidak dapat menggunakan grup keamanan yang tidak memiliki satu aturan masuk dengan grup keamanan VPC proksi atau fungsi Lambda sebagai sumbernya. Amazon RDS juga tidak dapat menggunak an grup keamanan yang telah diubah. Contoh-contoh perubahan meliputi penambahan aturan atau pengubahan porta aturan yang ada.	klaster basis data Multi-AZ atau proksi sebagai sumbernya . Amazon RDS juga tidak dapat menggunakan grup keamanan yang telah diubah.	klaster basis data Multi-AZ dan fungsi Lambda. Amazon RDS juga tidak dapat menggunakan grup keamanan yang telah diubah.	

Konfigurasi grup keamanan RDS saat ini	Konfigurasi grup keamanan Lambda saat ini	Konfigurasi grup keamanan proksi saat ini	Tindakan RDS
Ada satu atau beberapa grup keamanan yang terkait dengan klaster basis data Multi-AZ dengan nama yang cocok dengan pola `rds-lambda-` *n* atau jika `TargetHealth` proksi terkaitnya adalah `AVAILABLE`. Grup keamanan yang cocok dengan pola belum diubah. Grup keamanan ini memiliki hanya satu aturan masuk dengan grup keamanan VPC proksi atau fungsi Lambda sebagai sumbernya.	Ada satu atau beberapa grup keamanan yang terkait dengan fungsi Lambda dengan nama yang cocok dengan pola `lambda-rds-` *n* atau `lambda-rdsproxy-` *n*. Namun, Amazon RDS tidak dapat menggunakan satu pun grup keamanan ini untuk koneksi dengan klaster basis data Multi-AZ. Amazon RDS tidak dapat menggunakan grup keamanan yang tidak memiliki satu aturan keluar dengan grup keamanan VPC klaster basis data Multi-AZ atau proksi sebagai tujuannya. Amazon RDS juga tidak dapat menggunakan grup keamanan yang telah diubah.	Ada satu atau beberapa grup keamanan yang terkait dengan proksi dengan nama yang cocok dengan pola `rdsproxy-lambda-` *n*. Namun, Amazon RDS tidak dapat menggunakan satu pun grup keamanan ini untuk koneksi dengan klaster basis data Multi-AZ atau fungsi Lambda. Amazon RDS tidak dapat menggunakan grup keamanan yang tidak memiliki aturan masuk dan keluar dengan grup keamanan VPC klaster basis data Multi-AZ dan fungsi Lambda. Amazon RDS juga tidak dapat menggunakan grup keamanan yang telah diubah.	RDS action: create new security groups

Konfigurasi grup keamanan RDS saat ini	Konfigurasi grup keamanan Lambda saat ini	Konfigurasi grup keamanan proksi saat ini	Tindakan RDS
Ada satu atau beberapa grup keamanan yang terkait dengan klaster basis data Multi-AZ dengan nama yang cocok dengan pola `rds-lambda-` *n* atau jika `TargetHealth` proksi terkaitnya adalah `AVAILABLE`. Grup keamanan yang cocok dengan pola belum diubah. Grup keamanan ini memiliki hanya satu aturan masuk dengan grup keamanan VPC proksi atau fungsi Lambda sebagai sumbernya.	Ada grup keamanan Lambda yang valid untuk koneksi, tetapi tidak dikaitkan dengan fungsi Lambda. Grup keamanan ini memiliki nama yang cocok dengan pola `lambda-rds-` *n* atau `lambda-rdsproxy-` *n*. Grup itu belum diubah. Grup hanya memiliki satu aturan keluar dengan grup keamanan VPC klaster basis data Multi-AZ atau proksi sebagai tujuannya.	Ada grup keamanan proksi yang valid untuk koneksi, tetapi tidak dikaitkan dengan proksi. Grup keamanan ini memiliki nama yang cocok dengan pola `rdsproxy-lambda-` *n*. Grup itu belum diubah. Grup memiliki aturan masuk dan aturan keluar dengan grup keamanan VPC klaster basis data Multi-AZ dan fungsi Lambda.	RDS action: associate Lambda security group

Konfigurasi grup keamanan RDS saat ini	Konfigurasi grup keamanan Lambda saat ini	Konfigurasi grup keamanan proksi saat ini	Tindakan RDS
Salah satu syarat berikut dipenuhi: • Tidak ada grup keamanan yang terkait dengan klaster basis data Multi-AZ dengan nama yang cocok dengan pola `rds-lambda-` *n* atau jika `TargetHealth` proksi terkaitnya adalah `AVAILABLE` . • Ada satu atau beberapa grup keamanan yang terkait dengan klaster basis data Multi-AZ dengan nama yang cocok dengan pola `rds-lambda-` *n* atau jika `TargetHealth` proksi terkaitnya adalah `AVAILABLE` . Namun, Amazon RDS tidak dapat	Ada satu atau beberapa grup keamanan yang terkait dengan fungsi Lambda dengan nama yang cocok dengan pola `lambda-rds-` *n* atau `lambda-rdsproxy-` *n*. Grup keamanan yang cocok dengan pola belum diubah. Grup keamanan ini hanya memiliki satu aturan keluar dengan grup keamanan VPC klaster basis data Multi-AZ atau proksi sebagai tujuannya.	Ada satu atau beberapa grup keamanan yang terkait dengan proksi dengan nama yang cocok dengan pola `rdsproxy-lambda-` *n*. Grup keamanan yang cocok dengan pola belum diubah. Grup keamanan ini memiliki aturan masuk dan aturan keluar dengan grup keamanan VPC klaster basis data Multi-AZ dan fungsi Lambda.	RDS action: create new security groups

Konfigurasi grup keamanan RDS saat ini	Konfigurasi grup keamanan Lambda saat ini	Konfigurasi grup keamanan proksi saat ini	Tindakan RDS
menggunakan satu pun grup keamanan ini untuk koneksi dengan proksi atau fungsi Lambda. Amazon RDS tidak dapat menggunakan grup keamanan yang tidak memiliki satu aturan masuk dengan grup keamanan VPC proksi atau fungsi Lambda sebagai sumbernya. Amazon RDS juga tidak dapat menggunakan grup keamanan yang telah diubah.			

Konfigurasi grup keamanan RDS saat ini	Konfigurasi grup keamanan Lambda saat ini	Konfigurasi grup keamanan proksi saat ini	Tindakan RDS
Ada satu atau beberapa grup keamanan yang terkait dengan klaster basis data Multi-AZ dengan nama yang cocok dengan pola `rds-rdsproxy-` *n* (dengan *n* berupa angka).	Salah satu syarat berikut dipenuhi: • Tidak ada grup keamanan yang terkait dengan fungsi Lambda dengan nama yang cocok dengan pola `lambda-rds-` *n* atau `lambda-rdsproxy-` *n*. • Ada satu atau beberapa grup keamanan yang terkait dengan fungsi Lambda dengan nama yang cocok dengan pola `lambda-rds-` *n* atau `lambda-rdsproxy-` *n*. Namun, Amazon RDS tidak dapat menggunakan satu pun grup keamanan ini untuk koneksi dengan klaster basis data Multi-AZ.	Salah satu syarat berikut dipenuhi: • Tidak ada grup keamanan yang terkait dengan proksi dengan nama yang cocok dengan pola `rdsproxy-lambda-` *n*. • Ada satu atau beberapa grup keamanan yang terkait dengan proksi dengan nama yang cocok dengan `rdsproxy-lambda-` *n*. Namun, Amazon RDS tidak dapat menggunakan satu pun grup keamanan ini untuk koneksi dengan klaster basis data Multi-AZ atau fungsi Lambda.	RDS action: create new security groups

Konfigurasi grup keamanan RDS saat ini	Konfigurasi grup keamanan Lambda saat ini	Konfigurasi grup keamanan proksi saat ini	Tindakan RDS
	Amazon RDS tidak dapat menggunakan grup keamanan yang tidak memiliki satu aturan keluar dengan grup keamanan VPC klaster basis data Multi-AZ atau proksi sebagai tujuannya . Amazon RDS juga tidak dapat menggunakan grup keamanan yang telah diubah.	Amazon RDS tidak dapat menggunak an grup keamanan yang tidak memiliki aturan masuk dan keluar dengan grup keamanan VPC klaster basis data Multi-AZ dan fungsi Lambda. Amazon RDS juga tidak dapat menggunakan grup keamanan yang telah diubah.	

Tindakan RDS : membuat grup keamanan baru

Amazon RDS melakukan tindakan-tindakan berikut:

- Membuat grup keamanan baru yang cocok dengan pola `rds-lambda-`*n*. Grup keamanan ini memiliki aturan masuk dengan grup keamanan VPC fungsi Lambda atau proksi sebagai sumbernya. Grup keamanan ini dikaitkan dengan klaster basis data Multi-AZ dan memungkinkan fungsi atau proksi mengakses klaster basis data Multi-AZ.
- Membuat grup keamanan baru yang cocok dengan pola `lambda-rds-`*n*. Grup keamanan ini memiliki aturan keluar dengan grup keamanan VPC klaster basis data Multi-AZ atau proksi sebagai tujuannya. Grup keamanan ini dikaitkan dengan fungsi Lambda dan memungkinkan fungsi Lambda mengirim lalu lintas ke klaster basis data Multi-AZ atau mengirim lalu lintas melalui proksi.
- Membuat grup keamanan baru yang cocok dengan pola `rdsproxy-lambda-`*n*. Grup keamanan ini memiliki aturan masuk dan aturan keluar dengan grup keamanan VPC klaster basis data Multi-AZ dan fungsi Lambda.

Tindakan RDS : mengaitkan grup keamanan Lambda

Amazon RDS mengaitkan grup keamanan Lambda yang valid dan sudah ada dengan fungsi Lambda. Grup keamanan ini memungkinkan fungsi mengirim lalu lintas ke klaster basis data Multi-AZ atau mengirim lalu lintas melalui proksi.

Menghubungkan secara otomatis fungsi Lambda dan klaster basis data Multi-AZ

Anda dapat menggunakan konsol Amazon RDS untuk menghubungkan secara otomatis fungsi Lambda dengan klaster basis data Multi-AZ Anda. Ini menyederhanakan proses penyiapan koneksi di antara sumber daya-sumber daya ini.

Anda juga dapat menggunakan Proksi RDS untuk menyertakan proksi dalam koneksi Anda. Fungsi Lambda membuat koneksi basis data yang singkat tetapi sering yang menarik manfaat dari penghimpunan koneksi yang ditawarkan Proksi RDS. Anda juga dapat menggunakan sebarang autentikasi IAM yang Anda miliki untuk fungsi Lambda, alih-alih mengelola kredensial basis data dalam kode aplikasi Lambda.

Anda dapat menghubungkan klaster basis data Multi-AZ yang ada dengan fungsi Lambda baru dan lama dengan menggunakan halaman Siapkan koneksi Lambda. Proses penyiapan menyiapkan secara otomatis grup keamanan yang diperlukan untuk Anda.

Sebelum menyiapkan koneksi antara fungsi Lambda dan klaster basis data Multi-AZ, pastikan bahwa:

- Fungsi Lambda dan klaster basis data Multi-AZ Anda berada di VPC yang sama.
- Anda memiliki izin-izin yang tepat untuk akun pengguna Anda. Lihat informasi lebih lanjut tentang persyaratan di Ikhtisar konektivitas otomatis dengan fungsi Lambda.

Jika Anda mengubah grup keamanan setelah mengonfigurasikan konektivitas, perubahan itu dapat memengaruhi koneksi antara fungsi Lambda dan klaster basis data Multi-AZ.

> **Note**
>
> Anda dapat menyiapkan secara otomatis koneksi antara klaster basis data Multi-AZ dan fungsi Lambda hanya di AWS Management Console. Untuk menghubungkan fungsi Lambda, semua instans di klaster basis data Multi-AZ harus dalam keadaan Tersedia.

Untuk menghubungkan secara otomatis fungsi Lambda dan klaster basis data Multi-AZ

<result>

Setelah Anda memastikan penyiapan, Amazon RDS memulai proses menghubungkan fungsi Lambda, Proksi RDS (jika Anda menggunakan proksi), dan klaster basis data Multi-AZ. Konsol menampilkan kotak dialog Detail koneksi, yang mencantumkan perubahan grup keamanan yang memungkinkan koneksi di antara sumber daya-sumber daya Anda.
</result>

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data, lalu pilih klaster basis data Multi-AZ yang ingin Anda hubungkan dengan fungsi Lambda.
3. Untuk Tindakan, pilih Siapkan koneksi Lambda.
4. Pada halaman Siapkan koneksi Lambda, di bawah Pilih fungsi Lambda, lakukan salah satu hal berikut:
   - Jika Anda sudah memiliki fungsi Lambda di VPC yang sama dengan klaster basis data Multi-AZ, pilih Pilih fungsi yang ada, lalu pilih fungsi itu.
   - Jika Anda tidak memiliki fungsi Lambda di VPC yang sama, pilih Buat fungsi baru, lalu masukkan Nama fungsi. Runtime bawaan diatur ke Nodejs.18. Anda dapat mengubah setelan untuk fungsi Lambda baru di konsol Lambda setelah menyelesaikan penyiapan koneksi.
5. (Opsional) Di bawah Proksi RDS, pilih Hubungkan lewat Proksi RDS, lalu lakukan salah satu hal berikut:
   - Jika Anda sudah memiliki proksi yang ingin Anda gunakan, pilih Pilih proksi yang ada, lalu pilih proksi itu.
   - Jika Anda tidak memiliki proksi, dan Anda ingin Amazon RDS membuatnya secara otomatis untuk Anda, pilih Buat proksi baru. Lalu, untuk Kredensial basis data, lakukan salah satu langkah berikut:
     a. Pilih Nama pengguna dan kata sandi basis data, lalu masukkan Nama pengguna dan Kata sandi untuk klaster basis data Multi-AZ Anda.
     b. Pilih Rahasia Secrets Manager. Kemudian, untuk Pilih rahasia, pilih rahasia AWS Secrets Manager. Jika Anda tidak memiliki rahasia Secrets Manager, pilih Buat rahasia Secrets Manager baru untuk membuat rahasia baru. Setelah Anda membuat rahasia, untuk Pilih rahasia, pilih rahasia baru.

Setelah Anda membuat proksi baru, pilih Pilih proksi yang ada, lalu pilih proksi. Perhatikan bahwa mungkin perlu beberapa waktu sebelum proksi Anda tersedia untuk koneksi.

6. (Opsional) Perluas Ringkasan koneksi dan periksa pembaruan yang disorot untuk sumber daya Anda.
7. Pilih Siapkan.

Melihat sumber daya komputasi terhubung

Anda dapat menggunakan AWS Management Console untuk melihat sumber daya komputasi yang terhubung dengan klaster basis data Multi-AZ Anda. Sumber daya yang ditampilkan meliputi koneksi sumber daya komputasi yang disiapkan secara otomatis oleh Amazon RDS.

Sumber daya komputasi yang tercantum tidak menyertakan sumber daya yang dihubungkan secara manual dengan klaster basis data Multi-AZ. Misalnya, Anda dapat mengizinkan sumber daya komputasi untuk mengakses klaster basis data Multi-AZ Anda secara manual dengan menambahkan aturan ke grup keamanan VPC yang terkait dengan klaster.

Agar konsol menampilkan suatu fungsi Lambda, kondisi-kondisi berikut harus terpenuhi:

- Nama grup keamanan yang terkait dengan sumber daya komputasi cocok dengan pola `lambda-rds-`*n* atau `lambda-rdsproxy-`*n* (dengan *n* berupa angka).
- Grup keamanan yang terkait dengan sumber daya komputasi memiliki aturan keluar dengan rentang porta yang diatur ke porta klaster basis data Multi-AZ atau proksi terkait. Tujuan untuk aturan keluar harus diatur ke grup keamanan yang terkait dengan klaster basis data Multi-AZ atau proksi terkait.
- Nama grup keamanan yang dilampirkan pada proksi yang terkait dengan basis data Anda cocok dengan pola `rds-rdsproxy-`*n* (dengan *n* berupa angka).
- Grup keamanan yang terkait dengan fungsi memiliki aturan keluar dengan porta yang diatur ke porta yang digunakan klaster basis data Multi-AZ atau proksi terkait. Tujuan harus diatur ke grup keamanan yang terkait dengan klaster basis data Multi-AZ atau proksi terkait.

Untuk melihat sumber daya komputasi yang terhubung secara otomatis dengan klaster basis data Multi-AZ

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.

2. Di panel navigasi, pilih Basis Data, lalu pilih klaster basis data Multi-AZ.
3. Pada tab Konektivitas dan keamanan, lihat sumber daya komputasi di bawah Sumber daya komputasi terhubung.

Mengubah klaster basis data Multi-AZ

Klaster basis data Multi-AZ memiliki satu instans basis data penulis dan dua instans basis data pembaca dalam tiga Zona Ketersediaan terpisah. Klaster DB Multi-AZ menyediakan ketersediaan tinggi, peningkatan kapasitas untuk beban kerja baca, dan latensi yang lebih rendah jika dibandingkan dengan deployment Multi-AZ. Lihat informasi yang lebih lengkap tentang klaster basis data Multi-AZ di Deployment klaster basis data Multi-AZ.

Anda dapat mengubah klaster basis data Multi-AZ untuk mengubah setelannya. Anda juga dapat melakukan operasi-operasi pada klaster basis data Multi-AZ, seperti mengambil cuplikannya.

> **Important**
>
> Anda tidak dapat memodifikasi instans DB dalam cluster DB multi-AZ. Semua modifikasi harus dilakukan pada tingkat cluster DB. Satu-satunya operasi yang dapat Anda lakukan pada instans DB dalam cluster DB multi-AZ adalah me-reboot.

Anda dapat memodifikasi cluster DB multi-AZ menggunakan AWS Management Console, AWS CLI, atau RDS API.

Konsol

Untuk mengubah klaster basis data Multi-AZ

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data, lalu pilih klaster basis data Multi-AZ yang ingin Anda ubah.
3. Pilih Ubah. Halaman Modifikasi klaster DB akan muncul.
4. Ubah pengaturan apa pun yang Anda inginkan. Lihat informasi tentang setiap setelan di Setelan untuk mengubah klaster basis data Multi-AZ.
5. Ketika semua perubahan sudah sesuai dengan keinginan Anda, pilih Lanjutkan dan periksa ringkasan modifikasi.
6. (Opsional) Pilih Terapkan seketika untuk menerapkan perubahan dengan serta-merta. Memilih opsi ini dapat menyebabkan waktu henti dalam beberapa kasus. Untuk informasi selengkapnya, lihat Menerapkan perubahan dengan serta--merta.
7. Di halaman penegasan, tinjau perubahan Anda. Jika benar, pilih Ubah klaster basis data untuk menyimpan perubahan Anda.

Atau pilih Kembali untuk mengedit perubahan atau Batalkan untuk membatalkan perubahan.

AWS CLI

Untuk memodifikasi cluster DB Multi-AZ dengan menggunakan AWS CLI, panggil modify-db-clusterperintah. Tentukan pengidentifikasi klaster basis data dan nilai untuk opsi-opsi yang ingin Anda ubah. Lihat informasi tentang setiap opsi di Setelan untuk mengubah klaster basis data Multi-AZ.

Example

Kode berikut mengubah `my-multi-az-dbcluster` dengan mengatur periode retensi cadangan ke 1 minggu (7 hari). Kode ini mengaktifkan perlindungan penghapusan dengan menggunakan `--deletion-protection`. Untuk mematikan perlindungan penghapusan, gunakan `--no-deletion-protection`. Perubahan diterapkan selama jendela pemeliharaan berikutnya dengan menggunakan `--no-apply-immediately`. Gunakan `--apply-immediately` untuk segera menerapkan perubahan. Untuk informasi selengkapnya, lihat Menerapkan perubahan dengan serta-merta.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-cluster \
    --db-cluster-identifier my-multi-az-dbcluster \
    --backup-retention-period 7 \
    --deletion-protection \
    --no-apply-immediately
```

Untuk Windows:

```
aws rds modify-db-cluster ^
    --db-cluster-identifier my-multi-az-dbcluster ^
    --backup-retention-period 7 ^
    --deletion-protection ^
    --no-apply-immediately
```

API RDS

Untuk mengubah klaster basis data Multi-AZ dengan menggunakan API Amazon RDS, panggil operasi ModifyDBCluster. Tentukan pengidentifikasi klaster basis data, dan parameter-parameter untuk setelan yang ingin Anda ubah. Lihat informasi tentang tiap parameter di Setelan untuk mengubah klaster basis data Multi-AZ.

Menerapkan perubahan dengan serta--merta

Saat Anda mengubah klaster basis data Multi-AZ, Anda dapat menerapkan perubahan itu dengan serta-merta. Untuk menerapkan perubahan dengan serta-merta, pilih opsi Terapkan Segera di AWS Management Console. Atau Anda menggunakan `--apply-immediately` opsi saat memanggil AWS CLI atau mengatur `ApplyImmediately` parameter `true` saat menggunakan Amazon RDS API.

Jika Anda tidak memilih untuk menerapkan perubahan dengan serta-merta, perubahan akan dimasukkan ke dalam antrean perubahan yang tertunda. Selama jendela pemeliharaan berikutnya, perubahan yang tertunda di antrean akan diterapkan. Jika Anda memilih untuk menerapkan perubahan dengan serta-merta, perubahan baru dan segala perubahan di antrean pengubahan yang tertunda akan diterapkan.

> **Important**
>
> Jika salah satu pengubahan yang tertunda mengharuskan klaster basis data tidak tersedia untuk sementara (waktu henti), memilih opsi Terapkan seketika dapat menyebabkan waktu henti yang tidak terduga.
> Ketika Anda memilih untuk menerapkan perubahan dengan serta merta, setiap pengubahan yang tertunda juga akan diterapkan dengan serta-merta alih-alih selama jendela pemeliharaan berikutnya.
> Jika Anda tidak ingin perubahan tertunda diterapkan pada jendela pemeliharaan berikutnya, Anda dapat memodifikasi instans DB untuk membatalkan perubahan. Anda dapat melakukan ini dengan menggunakan AWS CLI dan menentukan `--apply-immediately` opsi.

Perubahan pada beberapa pengaturan basis data langsung diterapkan, meski Anda memilih untuk menunda perubahan. Lihat cara berbagai setelan basis data berinteraksi dengan setelan Terapkan seketika di Setelan untuk mengubah klaster basis data Multi-AZ.

Setelan untuk mengubah klaster basis data Multi-AZ

Lihat detail setelan yang dapat Anda gunakan untuk mengubah klaster basis data Multi-AZ di tabel berikut. Untuk informasi selengkapnya tentang AWS CLI opsi, lihat modify-db-cluster. Lihat informasi yang lebih lengkap tentang parameter API RDS di ModifyDBCluster.

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Kapan perubahan terjadi	Catatan waktu henti

Penyimpa an dialokasi kan	Jumlah penyimpanan yang dialokasi kan untuk setiap instans DB dalam klaster DB Anda (dalam gibibyte) . Untuk informasi selengkapnya, lihat Penyimpan an instans DB Amazon RDS.	Opsi CLI:
		`--allocated-storage`
		Parameter API RDS:
		`Allocated Storage`
			Jika Anda memilih untuk langsung menerapka n perubahan, perubahan akan langsung diterapka n.
			Jika Anda memilih untuk tidak langsung menerapka n perubahan, perubahan akan diterapkan pada jendela pemelihar aan berikutnya.
				Tidak terjadi waktu henti selama perubahan ini.

Pemutakl ran versi kecil otomatis	Aktifkan peningkatan versi minor otomatis agar klaster basis data Anda otomatis menerima peningkatan versi mesin DB minor pilihan Anda saat tersedia. Amazon RDS melakukan peningkatan versi minor otomatis	Opsi CLI:
		`--auto-minor-version-upgrade`
		`--no-auto-minor-version-upgrade`
		Parameter API RDS:
		`AutoMinor VersionUp grade`
			Perubahan langsung diterapka n. Pengaturan ini mengabaik an pengaturan terapkan langsung.	Tidak terjadi waktu henti selama perubahan ini.

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Kapan perubahan terjadi	Catatan waktu henti
	selama jendela pemeliharaan.			

Periode retensi cadangan	Jumlah hari yang Anda inginkan untuk menyimpan cadangan otomatis klaster basis data Anda. Untuk setiap klaster basis data nontrivial, atur nilai ini ke **1** atau lebih besar.
	
	Untuk informasi selengkapnya, lihat Pengantar cadangan.
		Opsi CLI:
		
		`--backup-retention-period`
		
		Parameter API RDS:
		
		`BackupRetentionPeriod`
			Jika Anda memilih untuk langsung menerapkan perubahan, perubahan akan langsung diterapkan.
			
			Jika Anda memilih untuk tidak langsung menerapkan perubahan, dan Anda mengubah pengaturan dari nilai bukan nol ke nilai bukan nol lainnya, perubahan akan diterapkan secara asinkron, sesegera mungkin. Jika tidak, perubahan akan terjadi selama jendela pemeliharaan berikutnya.
				Waktu henti terjadi jika Anda mengubah dari 0 ke nilai bukan nol, atau dari nilai bukan nol ke 0.

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Kapan perubahan terjadi	Catatan waktu henti

Jendela pencadangan	Periode waktu ketika Amazon RDS membuat cadangan klaster basis data Anda secara otomatis. Kecuali jika Anda ingin basis data dicadangkan pada waktu tertentu, gunakan nilai default Tidak ada preferensi.
	
	Untuk informasi selengkapnya, lihat Pengantar cadangan.
		Opsi CLI:
		
		`--preferred-backup-window`
		
		Parameter API RDS:
		
		`PreferredBackupWindow`
			Perubahan diterapkan secara asinkron, sesegera mungkin.	Tidak terjadi waktu henti selama perubahan ini.

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Kapan perubahan terjadi	Catatan waktu henti

Otoritas sertifikat	Otoritas sertifikat (CA) untuk sertifikat server yang digunakan oleh cluster DB.
	
	Untuk informasi selengkapnya, lihat .
		Opsi CLI:
		
		`--ca-certificate-identifier`
		
		Parameter API RDS:
		
		`CACertificateIdentifier`
			Jika Anda memilih untuk langsung menerapkan perubahan, perubahan akan langsung diterapkan.
			
			Jika Anda memilih untuk tidak langsung menerapkan perubahan, perubahan akan diterapkan pada jendela pemeliharaan berikutnya.
				Waktu henti hanya terjadi jika mesin DB tidak mendukung rotasi tanpa mulai ulang. Anda dapat menggunakan describe-db-engine-versions AWS CLI perintah untuk menentukan apakah mesin DB mendukung rotasi tanpa restart.

Salin tag ke cuplikan	Opsi ini menyalin semua tag klaster basis data ke snapshot DB saat Anda membuat snapshot.
	
	Untuk informasi selengkapnya, lihat Memberi tag pada sumber daya Amazon RDS.
		Opsi CLI:
		
		`-copy-tags-to-snapshot`
		
		`-no-copy-tags-to-snapshot`
		
		Parameter API RDS:
		
		`CopyTagsToSnapshot`
			Perubahan langsung diterapkan. Pengaturan ini mengabaikan pengaturan terapkan langsung.	Tidak terjadi waktu henti selama perubahan ini.

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Kapan perubahan terjadi	Catatan waktu henti
Autentikasi basis data	Untuk klaster DB Multi-AZ, hanya Autentikasi kata sandi yang didukung.	Tidak ada karena autentikasi kata sandi bersifat default.	Jika Anda memilih untuk menerapkan perubahan dengan serta-merta, perubahan terjadi dengan serta-merta. Jika Anda memilih untuk tidak langsung menerapkan perubahan, perubahan akan diterapkan pada jendela pemeliharaan berikutnya.	Tidak terjadi waktu henti selama perubahan ini.

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Kapan perubahan terjadi	Catatan waktu henti

Pengidentifikasi klaster basis data	Pengidentifikasi klaster DB. Nilai ini disimpan sebagai string huruf kecil.
	
	Saat Anda mengubah pengidentifikasi klaster basis data, titik akhir klaster basis data berubah. Pengidentifikasi dan titik akhir instans basis data dalam klaster basis data juga berubah. Nama klaster basis data yang baru harus unik. Panjang maksimalnya 63 karakter.
	
	Nama instans basis data dalam klaster basis data diubah agar sesuai dengan nama baru klaster basis data itu. Nama instans basis data baru
		Opsi CLI:
		
		`--new-db-cluster-identifier`
		
		Parameter API RDS:
		
		`NewDBClusterIdentifier`
			Jika Anda memilih untuk langsung menerapkan perubahan, perubahan akan langsung diterapkan.
			
			Jika Anda tidak memilih untuk menerapkan perubahan dengan serta-merta, perubahan akan terjadi selama jendela pemeliharaan berikutnya.
				Pemadaman tidak akan terjadi selama perubahan ini.

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Kapan perubahan terjadi	Catatan waktu henti
	tidak boleh sama dengan nama instans basis data yang ada. Misalnya, jika Anda mengganti nama klaster basis data menjadi `maz`, nama instans basis data dapat diubah menjadi `maz-instance-1` . Dalam hal ini, tidak boleh ada instans basis data bernama `maz-instance-1` . Untuk informasi selengkapnya, lihat Mengganti nama klaster basis data Multi-AZ.			

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Kapan perubahan terjadi	Catatan waktu henti

Kelas instans klaster basis data	Kapasitas komputasi dan memori setiap instans basis data dalam klaster basis data Multi-AZ, misalnya `db.r6gd.xlarge` .
	
	Jika mungkin, pilih kelas instans basis data yang cukup besar sehingga set kerja kueri yang lazim dapat disimpan di memori. Ketika set kerja disimpan di memori, sistem dapat menghindari menulis pada disk, yang akan meningkatkan performa.
	
	Untuk informasi selengkapnya, lihat the section called “Ketersediaan kelas instans untuk cluster DB multi-AZ”.
		Opsi CLI:
		
		`--db-cluster-instance-class`
		
		Parameter API RDS:
		
		`DBClusterInstanceClass`
			Jika Anda memilih untuk langsung menerapkan perubahan, perubahan akan langsung diterapkan.
			
			Jika Anda memilih untuk tidak langsung menerapkan perubahan, perubahan akan diterapkan pada jendela pemeliharaan berikutnya.
				Waktu henti terjadi selama perubahan ini.

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Kapan perubahan terjadi	Catatan waktu henti

Grup paramete klaster basis data	Grup parameter klaster DB yang ingin Anda kaitkan dengan klaster DB.
	
	Untuk informasi selengkapnya, lihat Bekerja dengan grup parameter untuk klaster basis data Multi-AZ.
		Opsi CLI:
		
		`--db-cluster-parameter-group-name`
		
		Parameter API RDS:
		
		`DBCluster Parameter GroupName`
			Perubahan grup parameter terjadi dengan serta-merta.	Pemadaman tidak akan terjadi selama perubahan ini. Saat Anda mengubah grup parameter , perubahan untuk beberapa parameter diterapkan pada instans basis data di dalam klaster basis data Multi-AZ dengan serta-merta tanpa boot ulang. Perubahan pada parameter -parameter lain diterapkan hanya setelah instans basis data di-boot ulang.

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Kapan perubahan terjadi	Catatan waktu henti

Versi mesin basis data	Versi mesin basis data yang ingin Anda gunakan.	Opsi CLI:
		`--engine-version`
		Parameter API RDS:
		`EngineVersion`
			Jika Anda memilih untuk langsung menerapkan perubahan, perubahan akan langsung diterapkan.
			Jika Anda tidak memilih untuk menerapkan perubahan dengan serta-merta, perubahan akan terjadi selama jendela pemeliharaan berikutnya.
				Pemadaman terjadi selama perubahan ini.

Perlindungan penghapusan	Aktifkan perlindungan penghapusan agar klaster DB tidak terhapus.
	Untuk informasi selengkapnya, lihat Menghapus instans DB.
		Opsi CLI:
		`--deletion-protection`
		`--no-deletion-protection`
		Parameter API RDS:
		`DeletionProtection`
			Perubahan langsung diterapkan. Setelan ini mengabaikan setelan Terapkan seketika.	Pemadaman tidak akan terjadi selama perubahan ini.

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Kapan perubahan terjadi	Catatan waktu henti

Jendela pemeliharaan	Jendela 30 menit ketika perubahan yang tertunda untuk klaster DB diterapkan. Jika jangka waktu bukan masalah, pilih Tidak ada preferensi.
	
	Untuk informasi selengkapnya, lihat Periode pemeliharaan Amazon RDS.
		Opsi CLI:
		
		`--preferred-maintenance-window`
		
		Parameter API RDS:
		
		`PreferredMaintenanceWindow`
			Perubahan langsung diterapkan. Pengaturan ini mengabaikan pengaturan langsung terapkan.	Jika ada satu atau beberapa tindakan tertunda yang menyebabkan waktu henti, dan jendela pemeliharaan diubah untuk menyertakan waktu saat ini, tindakan tertunda itu akan diterapkan dengan serta-merta dan terjadi waktu henti.

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Kapan perubahan terjadi	Catatan waktu henti

Kelola kredensi master di AWS Secrets Manager	Pilih Kelola kredensial master di AWS Secrets Manager untuk mengelola kata sandi pengguna master dalam rahasia di Secrets Manager.
	
	Anda juga dapat memilih kunci KMS yang akan digunakan untuk melindungi rahasia. Pilih dari kunci-kunci KMS di akun Anda, atau masukkan kunci dari akun yang lain.
	
	Jika RDS sudah mengelola kata sandi pengguna master untuk klaster basis data, Anda dapat merotasi kata sandi pengguna master dengan memilih Lakukan seketika rotasi rahasia.
		Opsi CLI:
		
		`--manage-master-user-password | --no-manage-master-user-password`
		
		`--master-user-secret-kms-key-id`
		
		`--rotate-master-user-password | --no-rotate-master-user-password`
		
		Parameter API RDS:
		
		`ManageMasterUserPassword`
		
		`MasterUserSecretKmsKeyId`
		
		`RotateMasterUserPassword`
			Jika Anda mengaktifkan atau menonaktifkan manajemen kata sandi pengguna utama otomatis, perubahan akan langsung diterapkan. Perubahan ini mengabaikan pengaturan terapkan langsung.
			
			Saat merotasi kata sandi pengguna utama, Anda harus menentukan bahwa perubahan tersebut akan langsung diterapkan.
				Tidak terjadi waktu henti selama perubahan ini.

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Kapan perubahan terjadi	Catatan waktu henti
	Untuk informasi selengkapnya, lihat Manajemen kata sandi dengan Amazon RDS Aurora dan AWS Secrets Manager.			

Kata sandi master baru	Kata sandi untuk akun pengguna master Anda.	Opsi CLI:
		`--master-user-password`
		Parameter API RDS:
		`MasterUserPassword`
			Perubahan diterapkan secara asinkron, sesegera mungkin. Pengaturan ini mengabaikan pengaturan langsung terapkan.	Tidak terjadi waktu henti selama perubahan ini.

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Kapan perubahan terjadi	Catatan waktu henti

IOPS yang Tersedia	Jumlah IOPS yang Tersedia (operasi input/output per detik) yang akan dialokasikan di awal untuk klaster basis data.	Opsi CLI:
		`--iops`
		Parameter API RDS:
		`Iops`
			Jika Anda memilih untuk langsung menerapkan perubahan, perubahan akan langsung diterapkan.
			Jika Anda memilih untuk tidak langsung menerapkan perubahan, perubahan akan diterapkan pada jendela pemeliharaan berikutnya.
				Tidak terjadi waktu henti selama perubahan ini.

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Kapan perubahan terjadi	Catatan waktu henti
Akses publik	Dapat diakses publik untuk memberikan alamat IP publik ke klaster basis data, yang berarti bahwa klaster dapat diakses di luar cloud privat virtual (VPC). Agar dapat diakses publik, klaster basis data juga harus berada di subnet publik di VPC. Tidak dapat diakses publik agar klaster DB hanya dapat diakses dari dalam VPC. Untuk informasi selengkapnya, lihat Menyembunyikan klaster DB dalam VPC dari internet. Untuk terhubung ke klaster DB dari luar VPC, klaster	Tidak tersedia saat mengubah klaster basis data.	Perubahan terjadi dengan serta-merta. Setelan ini mengabaikan setelan Terapkan seketika.	Pemadaman tidak akan terjadi selama perubahan ini.

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Kapan perubahan terjadi	Catatan waktu henti
	DB harus dapat diakses publik. Selain itu, akses harus diberikan menggunakan aturan masuk grup keamanan klaster DB, dan persyaratan lain harus terpenuhi . Untuk informasi selengkapnya, lihat Tidak dapat terhubung ke instans DB Amazon RDS. Jika kluster DB Anda tidak dapat diakses publik, Anda dapat menggunakan koneksi VPN AWS Site-to-S ite AWS Direct Connect atau koneksi untuk mengaksesnya dari jaringan pribadi. Untuk informasi selengkapnya, lihat Privasi lalu			

Setelan konsol	Deskripsi pengaturan	Opsi CLI dan parameter API RDS	Kapan perubahan terjadi	Catatan waktu henti
	lintas jaringan internet.			

Jenis penyimpanan	Jenis penyimpanan untuk klaster DB Anda.
	
	Hanya penyimpanan General Purpose SSD (gp3), Provisioned IOPS (io1), dan Provisioned IOPS SSD (io2) yang didukung.
	
	Untuk informasi selengkapnya, lihat Jenis penyimpanan Amazon RDS.
		Opsi CLI:
		
		`--storage-type`
		
		Parameter API RDS:
		
		`StorageType`
			Jika Anda memilih untuk langsung menerapkan perubahan, perubahan akan langsung diterapkan.
			
			Jika Anda memilih untuk tidak langsung menerapkan perubahan, perubahan akan diterapkan pada jendela pemeliharaan berikutnya.
				Tidak terjadi waktu henti selama perubahan ini.

Grup keamanan VPC	Grup keamanan untuk dikaitkan dengan klaster basis data.
	
	Untuk informasi selengkapnya, lihat Ikhtisar grup keamanan VPC.
		Opsi CLI:
		
		`--vpc-security-group-ids`
		
		Parameter API RDS:
		
		`VpcSecurityGroupIds`
			Perubahan diterapkan secara asinkron, sesegera mungkin. Setelan ini mengabaikan setelan Terapkan seketika.	Pemadaman tidak akan terjadi selama perubahan ini.

Setelan yang tidak berlaku saat mengubah klaster basis data Multi-AZ

Pengaturan berikut dalam AWS CLI perintah modify-db-clusterdan operasi RDS API ModifyDBCluster tidak berlaku untuk cluster DB multi-AZ.

Anda juga tidak dapat mengubah semua setelan ini untuk klaster basis data Multi-AZ di konsol.

AWS CLI pengaturan	Pengaturan API RDS
`--backtrack-window`	`BacktrackWindow`
`--cloudwatch-logs-export-configuration`	`CloudwatchLogsExportConfiguration`
`--copy-tags-to-snapshot | --no-copy-tags-to-snapshot`	`CopyTagsToSnapshot`
`--db-instance-parameter-group-name`	`DBInstanceParameterGroupName`
`--domain`	`Domain`
`--domain-iam-role-name`	`DomainIAMRoleName`
`--enable-global-write-forwarding | --no-enable-global-write-forwarding`	`EnableGlobalWriteForwarding`
`--enable-http-endpoint | --no-enable-http-endpoint`	`EnableHttpEndpoint`
`--enable-iam-database-authentication | --no-enable-iam-database-authentication`	`EnableIAMDatabaseAuthentication`
`--option-group-name`	`OptionGroupName`
`--port`	`Port`
`--scaling-configuration`	`ScalingConfiguration`

AWS CLI pengaturan	Pengaturan API RDS
`--storage-type`	`StorageType`

Mengganti nama klaster basis data Multi-AZ

Anda dapat mengganti nama klaster basis data Multi-AZ dengan menggunakanAWS Management Console, perintah AWS CLI `modify-db-cluster`, atau operasi API Amazon RDS `ModifyDBCluster`. Mengganti nama klaster basis data Multi-AZ dapat berefek besar. Berikut adalah daftar pertimbangan sebelum Anda mengganti nama klaster basis data Multi-AZ.

- Saat Anda mengganti nama klaster basis data Multi-AZ, titik akhir klaster untuk klaster basis data Multi-AZ berubah. Titik akhir ini berubah karena menyertakan nama yang Anda tetapkan untuk klaster basis data Multi-AZ. Anda dapat mengarahkan lalu lintas dari titik akhir lama ke yang baru. Lihat informasi yang lebih lengkap tentang titik akhir klaster basis data Multi-AZ di Menghubungi klaster basis data Multi-AZ.
- Saat Anda mengganti nama klaster basis data Multi-AZ, nama DNS lama yang digunakan oleh klaster itu akan dihapus, meskipun tetap disimpan dalam cache selama beberapa menit. Nama DNS baru untuk klaster basis data Multi-AZ menjadi berlaku dalam waktu sekitar dua menit. Klaster basis data Multi-AZ tidak tersedia hingga nama baru berlaku.
- Anda tidak dapat menggunakan nama klaster basis data Multi-AZ ketika nama klaster sedang diganti.
- Metrik dan peristiwa yang terkait dengan nama klaster basis data Multi-AZ dipertahankan jika Anda menggunakan ulang nama klaster basis data.
- Tag klaster basis data Multi-AZ tetap bersama klaster, terlepas dari penggantian nama.
- Cuplikan klaster basis data dipertahankan untuk klaster basis data Multi-AZ yang namanya diganti.

> **Note**
>
> Klaster basis data Multi-AZ adalah lingkungan basis data terisolasi yang berjalan di cloud. Klaster basis data Multi-AZ dapat menampung beberapa basis data. Lihat informasi tentang penggantian nama basis data dalam dokumentasi untuk mesin basis data Anda.

Mengganti nama untuk mengganti klaster basis data Multi-AZ yang ada

Skenario paling umum untuk mengganti nama cluster DB multi-AZ termasuk memulihkan data dari snapshot cluster DB atau melakukan point-in-time pemulihan (PITR). Dengan mengganti nama klaster basis data Multi-AZ, Anda dapat mengganti klaster basis data Multi-AZ tanpa mengubah sama

sekali kode aplikasi yang merujuk ke klaster basis data Multi-AZ. Dalam kasus ini, lakukan langkah-langkah berikut:

1. Hentikan semua lalu lintas ke klaster basis data Multi-AZ. Anda dapat mengarahkan lalu lintas dari berupaya mengakses basis data di klaster basis data Multi-AZ, atau memilih cara lain untuk mencegah lalu lintas yang mengakses basis data Anda di klaster basis data Multi-AZ.
2. Ganti nama klaster basis data Multi-AZ yang ada.
3. Buat klaster basis data Multi-AZ baru dengan memulihkan dari cuplikan klaster basis data atau memulihkan ke suatu titik waktu. Kemudian, beri klaster basis data Multi-AZ baru itu nama klaster basis data Multi-AZ lama.

Jika Anda menghapus klaster basis data Multi-AZ lama, Anda bertanggung jawab untuk menghapus setiap cuplikan klaster basis data Multi-AZ lama yang tidak diinginkan.

Konsol

Untuk mengganti nama klaster basis data Multi-AZ

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data.
3. Pilih klaster basis data Multi-AZ yang ingin Anda ganti namanya.
4. Pilih Ubah.
5. Di Pengaturan, masukkan nama baru untuk pengidentifikasi klaster basis data.
6. Pilih Lanjutkan.
7. Untuk menerapkan perubahan dengan serta-merta, pilih Terapkan seketika. Dalam beberapa kasus, memilih opsi ini dapat menyebabkan pemadaman. Untuk informasi selengkapnya, lihat Menerapkan perubahan dengan serta--merta.
8. Di halaman penegasan, tinjau perubahan Anda. Jika sudah benar, pilih Ubah klaster untuk menyimpan perubahan Anda.

   Atau, pilih Kembali untuk mengedit perubahan, atau pilih Batalkan untuk membuang perubahan.

AWS CLI

Untuk mengganti nama cluster DB Multi-AZ, gunakan perintah. AWS CLI modify-db-cluster Berikan nilai `--db-cluster-identifier` saat ini dan parameter `--new-db-cluster-identifier` dengan nama baru klaster basis data Multi-AZ.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-cluster \
    --db-cluster-identifier DBClusterIdentifier \
    --new-db-cluster-identifier NewDBClusterIdentifier
```

Untuk Windows:

```
aws rds modify-db-cluster ^
    --db-cluster-identifier DBClusterIdentifier ^
    --new-db-cluster-identifier NewDBClusterIdentifier
```

API RDS

Untuk mengganti nama klaster basis data Multi-AZ, panggil fungsi API Amazon RDS ModifyDBCluster dengan parameter-parameter berikut:

- `DBClusterIdentifier` – Nama klaster basis data yang ada.
- `NewDBClusterIdentifier` – Nama baru klaster basis data.

Membut ulang klaster basis data Multi-AZ dan instans basis data pembaca

Anda mungkin perlu membut ulang klaster basis data Multi-AZ, biasanya karena alasan pemeliharaan. Misalnya, jika Anda membuat pengubahan tertentu atau mengubah grup parameter klaster basis data yang terkait dengan suatu klaster basis data, Anda membut ulang klaster basis data itu. Melakukan hal itu menyebabkan perubahan berlaku.

Jika klaster basis data tidak menggunakan perubahan terbaru pada grup parameter klaster basis data terkaitnya, AWS Management Console menunjukkan grup parameter klaster basis data dengan status menunggu but ulang. Status grup parameter menunggu but ulang tidak menyebabkan pembutan ulang otomatis selama periode pemeliharaan berikutnya. Untuk menerapkan perubahan parameter terbaru pada klaster basis data itu, but ulang klaster basis data secara manual. Lihat informasi lebih lanjut tentang grup parameter di Bekerja dengan grup parameter untuk klaster basis data Multi-AZ.

Membut ulang klaster basis data akan memulai ulang layanan mesin basis data. Membut ulang klaster basis data akan menyebabkan pemadaman sementara, selama status klaster basis data ditetapkan ke membut ulang.

Anda tidak dapat membut ulang klaster basis data yang tidak berada dalam keadaan Tersedia. Basis data Anda dapat tidak tersedia karena beberapa alasan, seperti ada pencadangan sedang berlangsung, pengubahan yang telah diminta, atau tindakan jendela pemeliharaan.

Waktu yang diperlukan untuk membut ulang klaster Anda bergantung pada proses pemulihan kemacetan, aktivitas basis data pada saat dibut ulang, dan perilaku klaster basis data itu sendiri. Untuk mengurangi waktu but ulang, sebaiknya kurangi aktivitas basis data sebanyak mungkin selama proses itu. Mengurangi aktivitas basis data akan mengurangi aktivitas gulir balik/rollback untuk transaksi dalam transit.

> **Important**
>
> Klaster basis data Multi-AZ tidak mendukung but ulang dengan pindah saat gagal/failover. Saat Anda membut ulang instans penulis sebuah klaster basis data Multi-AZ, langkah itu tidak memengaruhi instans basis data pembaca di klaster basis data itu dan tidak ada pindah saat gagal/failover yang terjadi. Saat Anda membut ulang instans basis data, tidak terjadi pindah saat gagal/failover. Untuk membuat pindah saat gagal/failover sebuah klaster basis data Multi-AZ, pilih Failover di konsol, panggil perintah AWS CLI `failover-db-cluster`, atau panggil operasi API `FailoverDBCluster`.

Konsol

Untuk membut ulang klaster basis data

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data, lalu pilih klaster basis data Multi-AZ yang ingin Anda but ulang.
3. Untuk Tindakan, pilih But ulang.

   Halaman But ulang klaster basis data muncul.
4. Pilih But ulang untuk membut ulang klaster basis data Anda.

   Atau pilih Batalkan.

AWS CLI

Untuk membut ulang klaster basis data Multi-AZ dengan menggunakan AWS CLI, panggil perintah reboot-db-cluster.

```
aws rds reboot-db-cluster --db-cluster-identifier mymultiazdbcluster
```

API RDS

Untuk membut ulang klaster basis data Multi-AZ dengan menggunakan API Amazon RDS, panggil operasi RebootDBCluster.

Bekerja dengan replika baca klaster DB multi-AZ

Replika baca klaster DB adalah jenis klaster khusus yang Anda buat dari instans DB sumber. Setelah Anda membuat replika baca, pembaruan apa pun yang dilakukan pada instans DB utama disalin secara asinkron ke replika baca klaster DB Multi-AZ. Anda dapat mengurangi beban pada instans DB utama dengan merutekan kueri baca dari aplikasi Anda ke replika baca. Dengan replika baca, Anda dapat dengan mudah menskalakan ke luar dari batasan kapasitas instans DB tunggal untuk beban kerja database yang berstatus read-heavy.

Anda juga dapat membuat satu atau beberapa replika baca instans DB dari klaster DB Multi-AZ. Replika baca instans DB memungkinkan Anda menskalakan di luar kapasitas komputasi atau I/O dari klaster DB Multi-AZ sumber dengan mengarahkan lalu lintas baca berlebih ke replika baca. Saat ini, Anda tidak dapat membuat replika baca klaster DB Multi-AZ dari klaster DB Multi-AZ yang ada.

Topik

Migrasi ke klaster DB Multi-AZ menggunakan replika baca

Untuk memigrasikan deployment Satu AZ atau deployment instans DB Multi-AZ ke deployment klaster DB Multi-AZ dengan waktu henti yang dikurangi, Anda dapat membuat replika baca klaster DB Multi-AZ. Untuk sumbernya, Anda menentukan instans DB dalam deployment Satu AZ atau instans DB utama dalam deployment instans DB Multi-AZ. Instans DB dapat memproses transaksi tulis selama migrasi ke klaster DB Multi-AZ.

Pertimbangkan hal berikut sebelum Anda membuat replika baca klaster Multi-AZ:

- Instans DB sumber harus ada pada versi yang mendukung klaster DB Multi-AZ. Untuk informasi selengkapnya, lihat Klaster DB Multi-AZ.
- Replika baca klaster multi-AZ DB harus pada versi utama yang sama dengan sumbernya, dan versi minor yang sama atau lebih tinggi.
- Anda harus mengaktifkan pencadangan otomatis pada instans DB sumber dengan mengatur periode penyimpanan cadangan ke nilai selain 0.
- Penyimpanan yang dialokasikan dari instans DB sumber harus 100 GiB atau lebih tinggi.
- Untuk RDS for MySQL, parameter `gtid-mode` dan `enforce_gtid_consistency` harus diatur ke `ON` untuk instans DB sumber. Anda harus menggunakan grup parameter khusus, bukan grup

parameter default. Untuk informasi selengkapnya, lihat the section called “Bekerja dengan grup parameter DB”.

- Transaksi aktif yang berjangka panjang dapat memperlambat proses pembuatan replika baca. Kami menyarankan Anda menunggu transaksi yang berjangka panjang selesai sebelum membuat replika baca.
- Jika Anda menghapus instans DB sumber untuk replika baca klaster DB Multi-AZ, replika baca akan dipromosikan menjadi klaster DB Multi-AZ mandiri.

Membuat dan mempromosikan replika baca klaster DB Multi-AZ

Anda dapat membuat dan mempromosikan replika baca cluster DB multi-AZ menggunakan AWS Management Console, AWS CLI, atau RDS API.

> **Note**
>
> Kami sangat menyarankan Anda untuk membuat semua replika baca di cloud privat virtual (VPC) yang sama berdasarkan Amazon VPC instans DB sumber.
> Jika Anda membuat replika baca di VPC yang berbeda dari instans DB sumber, rentang Perutean Antar Domain Tanpa Kelas (CIDR) dapat tumpang tindih antara replika dan sistem Amazon RDS. CIDR yang tumpang tindih membuat replika tidak stabil, yang dapat berdampak negatif pada aplikasi yang terhubung dengannya. Jika Anda terjadi kesalahan saat membuat replika baca, pilih grup subnet DB tujuan yang berbeda. Untuk informasi selengkapnya, lihat Bekerja dengan klaster DB dalam VPC.

Konsol

Untuk memigrasikan deployment Satu AZ atau deployment instans DB Multi-AZ ke klaster DB Multi-AZ menggunakan replika baca, selesaikan langkah-langkah berikut menggunakan AWS Management Console.

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Buat replika baca klaster DB Multi-AZ.
   a. Di panel navigasi, pilih Basis Data.
   b. Pilih instans DB yang ingin Anda gunakan sebagai sumber untuk replika baca.

c. Untuk Tindakan, pilih Buat replika baca.

d. Untuk Ketersediaan dan daya tahan, pilih Klaster DB Multi-AZ.

e. Untuk Pengidentifikasi instans DB, masukkan nama replika baca.

f. Untuk bagian yang tersisa, tentukan pengaturan klaster DB Anda. Untuk informasi tentang sebuah pengaturan, lihat Pengaturan untuk membuat klaster DB Multi-AZ.

g. Pilih Buat replika baca.

3. Saat Anda siap, tingkatkan replika baca menjadi klaster DB Multi-AZ mandiri:

a. Hentikan transaksi apa pun agar tidak ditulis ke instans DB sumber, lalu tunggu semua pembaruan yang akan dilakukan ke replika baca.

Pembaruan basis data terjadi pada replika baca setelah pembaruan terjadi pada instans DB utama. Kelambatan replikasi ini dapat sangat bervariasi. Gunakan metrik `ReplicaLag` untuk menentukan saat semua pembaruan sudah dilakukan pada replika baca. Untuk informasi selengkapnya tentang kelambatan replika, lihat Memantau replikasi baca.

b. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.

c. Di konsol Amazon RDS, pilih Database.

Panel Database muncul. Setiap replika baca menampilkan Replika di kolom Peran.

d. Pilih replika baca klaster DB Multi-AZ yang ingin Anda promosikan.

e. Untuk Tindakan, pilih Promosikan.

f. Pada halaman Tingkatkan replika baca, masukkan periode penyimpanan cadangan dan jendela cadangan untuk klaster DB Multi-AZ yang baru ditingkatkan.

g. Jika pengaturan sudah sesuai keinginan Anda, pilih Tingkatkan replika baca.

h. Tunggu status klaster DB Multi-AZ yang dipromosikan menjadi `Available`.

i. Arahkan aplikasi Anda untuk menggunakan klaster DB Multi-AZ yang dipromosikan.

Secara opsional, hapus deployment Satu AZ atau deployment instans DB Multi-AZ jika tidak lagi diperlukan. Untuk petunjuk, lihat Menghapus instans DB.

AWS CLI

Untuk memigrasikan deployment Satu AZ atau deployment instans DB Multi-AZ ke klaster DB Multi-AZ menggunakan replika baca, selesaikan langkah-langkah berikut menggunakan AWS CLI.

1. Buat replika baca klaster DB Multi-AZ.

   Untuk membuat replika baca dari instance DB sumber, gunakan AWS CLI perintah `create-db-cluster`. Untuk `--replication-source-identifier`, tentukan Amazon Resource Name (ARN) instans DB sumber.

   Untuk Linux, macOS, atau Unix:

   ```
   aws rds create-db-cluster \
     --db-cluster-identifier mymultiazdbcluster \
     --replication-source-identifier arn:aws:rds:us-
   east-2:123456789012:db:mydbinstance
     --engine postgres \
     --db-cluster-instance-class db.m5d.large \
     --storage-type io1 \
     --iops 1000 \
     --db-subnet-group-name defaultvpc \
     --backup-retention-period 1
   ```

   Untuk Windows:

   ```
   aws rds create-db-cluster ^
     --db-cluster-identifier mymultiazdbcluster ^
     --replication-source-identifier arn:aws:rds:us-
   east-2:123456789012:db:mydbinstance
     --engine postgres ^
     --db-cluster-instance-class db.m5d.large ^
     --storage-type io1 ^
     --iops 1000 ^
     --db-subnet-group-name defaultvpc ^
     --backup-retention-period 1
   ```

2. Hentikan transaksi apa pun agar tidak ditulis ke instans DB sumber, lalu tunggu semua pembaruan yang akan dilakukan ke replika baca.

   Pembaruan basis data terjadi pada replika baca setelah pembaruan terjadi pada instans DB utama. Kelambatan replikasi ini dapat sangat bervariasi. Gunakan metrik `Replica Lag` untuk

menentukan saat semua pembaruan sudah dilakukan pada replika baca. Untuk informasi selengkapnya tentang kelambatan replika, lihat Memantau replikasi baca.

3. Saat Anda siap, tingkatkan replika baca menjadi klaster DB Multi-AZ mandiri.

   Untuk mempromosikan replika baca klaster Multi-AZ DB, gunakan perintah. AWS CLI `promote-read-replica-db-cluster` Untuk `--db-cluster-identifier`, tentukan pengidentifikasi replika baca klaster DB Multi-AZ.

   ```
   aws rds promote-read-replica-db-cluster --db-cluster-identifier mymultiazdbcluster
   ```

4. Tunggu status klaster DB Multi-AZ yang dipromosikan menjadi `Available`.
5. Arahkan aplikasi Anda untuk menggunakan klaster DB Multi-AZ yang dipromosikan.

Secara opsional, hapus deployment Satu AZ atau deployment instans DB Multi-AZ jika tidak lagi diperlukan. Untuk petunjuk, lihat Menghapus instans DB.

API RDS

Untuk memigrasikan deployment Satu AZ atau deployment instans DB Multi-AZ ke klaster DB Multi-AZ menggunakan replika baca, selesaikan langkah-langkah berikut menggunakan API RDS.

1. Buat replika baca klaster DB Multi-AZ.

   Untuk membuat replika baca klaster DB Multi-AZ, gunakan `CreateDBCluster` operasi dengan parameter yang diperlukan `DBClusterIdentifier`. Untuk `ReplicationSourceIdentifier`, tentukan Amazon Resource Name (ARN) instans DB sumber.

2. Hentikan transaksi apa pun agar tidak ditulis ke instans DB sumber, lalu tunggu semua pembaruan yang akan dilakukan ke replika baca.

   Pembaruan basis data terjadi pada replika baca setelah pembaruan terjadi pada instans DB utama. Kelambatan replikasi ini dapat sangat bervariasi. Gunakan metrik `Replica Lag` untuk menentukan saat semua pembaruan sudah dilakukan pada replika baca. Untuk informasi selengkapnya tentang kelambatan replika, lihat Memantau replikasi baca.

3. Saat Anda siap, tingkatkan replika baca menjadi klaster DB Multi-AZ mandiri.

Untuk membuat replika baca klaster DB Multi-AZ, gunakan `PromoteReadReplicaDBCluster` operasi dengan parameter yang diperlukan `DBClusterIdentifier`. Tentukan pengidentifikasi replika baca klaster DB Multi-AZ.

4. Tunggu status klaster DB Multi-AZ yang dipromosikan menjadi `Available`.
5. Arahkan aplikasi Anda untuk menggunakan klaster DB Multi-AZ yang dipromosikan.

Secara opsional, hapus deployment Satu AZ atau deployment instans DB Multi-AZ jika tidak lagi diperlukan. Untuk petunjuk, lihat Menghapus instans DB.

Keterbatasan untuk membuat replika baca klaster DB Multi-AZ

Batasan berikut berlaku untuk membuat replika baca klaster DB Multi-AZ dari deployment Satu AZ atau deployment instans DB Multi-AZ.

- Anda tidak dapat membuat replika baca cluster DB multi-AZ dalam replika Akun AWS yang berbeda dari Akun AWS yang memiliki instans DB sumber.
- Anda tidak dapat membuat replika baca cluster DB multi-AZ di instans DB sumber yang Wilayah AWS berbeda.
- Anda tidak dapat memulihkan replika baca klaster DB Multi-AZ ke suatu titik waktu.
- Enkripsi penyimpanan harus memiliki pengaturan yang sama pada instans DB sumber dan klaster DB Multi-AZ.
- Jika instans DB sumber dienkripsi, replika baca klaster DB Multi-AZ harus dienkripsi menggunakan tombol KMS yang sama.
- Jika instans DB sumber menggunakan penyimpanan General Purpose SSD (gp3) dan memiliki kurang dari 400 GiB penyimpanan yang dialokasikan, Anda tidak dapat memodifikasi IOPS yang disediakan untuk replika baca cluster DB multi-AZ.
- Untuk melakukan upgrade versi minor pada instans DB sumber, Anda harus terlebih dahulu melakukan upgrade versi minor pada replika baca klaster DB Multi-AZ DB.
- Ketika Anda melakukan peningkatan versi minor pada RDS untuk PostgreSQL replika baca klaster DB Multi-AZ, instans DB pembaca tidak beralih ke instans DB penulis setelah peningkatan. Oleh karena itu, klaster DB Anda mungkin mengalami waktu henti saat Amazon RDS memutakhirkan instans penulis.
- Anda tidak dapat melakukan upgrade versi utama pada replika baca cluster multi-AZ DB.
- Anda dapat melakukan peningkatan versi utama pada instans DB sumber dari replika baca klaster DB Multi-AZ, tetapi replikasi ke replika baca berhenti dan tidak dapat dimulai ulang.

- Replika baca klaster DB Multi-AZ tidak mendukung replika baca berjenjang.
- Untuk RDS untuk PostgreSQL, replika baca klaster Multi-AZ DB tidak dapat gagal.

Membuat replika baca instans DB Multi-AZ dari klaster DB Multi-AZ

Anda dapat membuat replika baca instans DB Multi-AZ agar dapat menskalakan di luar kapasitas komputasi atau I/O klaster untuk beban kerja database yang berstatus read-heavy. Anda dapat mengarahkan kelebihan lalu lintas baca ini ke satu atau beberapa replika baca instans DB. Anda juga dapat menggunakan replika baca untuk bermigrasi dari klaster DB Multi-AZ ke instans DB.

Untuk membuat replika baca, tentukan klaster DB Multi-AZ sebagai sumber replikasi. Salah satu instans pembaca dari klaster DB Multi-AZ selalu menjadi sumber replikasi, bukan instans penulis. Kondisi ini memastikan bahwa replika selalu sinkron dengan klaster sumber, bahkan dalam kasus failover.

Topik

Membandingkan instans DB pembaca dan replika baca instans DB

Replika baca instans DB dari klaster DB Multi-AZ berbeda dari instans DB pembaca dari klaster DB Multi-AZ dengan cara berikut:

- Instans DB pembaca bertindak sebagai target failover otomatis, sedangkan replika baca instans DB tidak.
- Instans DB pembaca harus mengakui perubahan dari instans DB penulis sebelum perubahan dapat dilakukan. Untuk replika baca instans DB, pembaruan disalin secara asinkron ke replika baca tanpa memerlukan pengakuan.
- Instans DB pembaca selalu berbagi kelas instans, tipe penyimpanan, dan versi mesin yang sama dengan instans DB penulis dari klaster DB Multi-AZ. Replika baca instans DB, bagaimanapun, tidak harus berbagi konfigurasi yang sama dengan klaster sumber.

- Anda dapat meningkatkan replika baca instans DB ke instans DB mandiri. Anda tidak dapat mempromosikan instans DB pembaca dari klaster DB Multi-AZ ke instans mandiri.
- Titik akhir pembaca hanya merutekan permintaan ke instans DB pembaca dari klaster DB Multi-AZ. Titik akhir tidak pernah merutekan permintaan ke replika baca instans DB.

Untuk informasi selengkapnya tentang cara membuat instans DB, lihat the section called “Ikhtisar klaster basis data Multi-AZ”.

Pertimbangan

Pertimbangkan hal berikut sebelum Anda membuat replika baca instans DB dari klaster DB Multi-AZ:

- Ketika Anda membuat replika baca instans DB, replika tersebut harus pada versi utama yang sama dengan klaster sumbernya, dan versi minor yang sama atau lebih tinggi. Setelah Anda membuatnya, Anda dapat secara opsional meningkatkan replika baca ke versi minor yang lebih tinggi daripada klaster sumber.
- Saat Anda membuat replika baca instans DB, penyimpanan yang dialokasikan harus sama dengan penyimpanan yang dialokasikan dari klaster DB Multi-AZ sumber. Anda dapat mengubah penyimpanan yang dialokasikan setelah replika baca dibuat.
- Untuk RDS untuk MySQL, parameter `gtid-mode` harus diatur ke `ON` untuk sumber klaster DB Multi-AZ. Untuk informasi selengkapnya, lihat the section called “Bekerja dengan grup parameter klaster DB”.
- Transaksi aktif yang berjangka panjang dapat memperlambat proses pembuatan replika baca. Kami menyarankan Anda menunggu transaksi yang berjangka panjang selesai sebelum membuat replika baca.
- Jika Anda menghapus klaster DB Multi-AZ sumber untuk replika baca instans DB, replika baca apa pun yang ditulisnya dipromosikan ke instans DB mandiri.

Membuat replika baca instans DB

Anda dapat membuat replika baca instans DB dari cluster DB multi-AZ menggunakan AWS Management Console, AWS CLI, atau RDS API.

> **Note**
>
> Kami sangat menyarankan Anda untuk membuat semua replika baca di cloud privat virtual (VPC) yang sama berdasarkan Amazon VPC klaster DB Multi-AZ sumber.

Jika Anda membuat replika baca di VPC yang berbeda dari klaster DB sumber, rentang Perutean Antar Domain Tanpa Kelas (CIDR) dapat tumpang tindih antara replika dan sistem Amazon RDS. CIDR yang tumpang tindih membuat replika tidak stabil, yang dapat berdampak negatif pada aplikasi yang terhubung dengannya. Jika Anda terjadi kesalahan saat membuat replika baca, pilih grup subnet DB tujuan yang berbeda. Untuk informasi selengkapnya, lihat the section called “Bekerja dengan klaster DB dalam VPC”.

Konsol

Untuk membuat replika baca instans DB dari klaster DB Multi-AZ, selesaikan langkah-langkah berikut menggunakan AWS Management Console.

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data.
3. Pilih klaster DB Multi-AZ yang ingin Anda gunakan sebagai sumber untuk replika baca.
4. Untuk Tindakan, pilih Buat replika baca.
5. Untuk sumber Replica, pastikan bahwa klaster DB Multi-AZ yang benar dipilih.
6. Untuk Pengenal DB, masukkan nama replika baca.
7. Untuk bagian yang tersisa, tentukan pengaturan instans DB Anda. Untuk informasi tentang sebuah pengaturan, lihat the section called “Pengaturan tersedia”.

   > **Note**
   >
   > Penyimpanan yang dialokasikan untuk instans DB harus sama dengan penyimpanan yang dialokasikan untuk klaster DB Multi-AZ sumber.

8. Pilih Buat replika baca.

AWS CLI

Untuk membuat replika baca instans DB dari cluster DB multi-AZ, gunakan perintah. AWS CLI `create-db-instance-read-replica` Untuk `--source-db-cluster-identifier`, tentukan pengidentifikasi replika baca klaster DB Multi-AZ.

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-instance-read-replica \
  --db-instance-identifier myreadreplica \
  --source-db-cluster-identifier mymultiazdbcluster
```

Untuk Windows:

```
aws rds create-db-instance-read-replica ^
  --db-instance-identifier myreadreplica ^
  --source-db-cluster-identifier mymultiazdbcluster
```

API RDS

Untuk membuat replika baca instans DB dari klaster DB Multi-AZ, gunakan operasi `CreateDBInstanceReadReplica`.

Mempromosikan replika baca instans DB

Jika Anda tidak lagi membutuhkan replika baca instans DB, Anda dapat mempromosikannya menjadi instans DB mandiri. Saat Anda meningkatkan replika baca, instans DB akan di-boot ulang sebelum replika baca itu tersedia. Untuk petunjuk, lihat the section called “Mempromosikan replika baca”.

Jika Anda menggunakan replika baca untuk memigrasikan deployment klaster DB Multi-AZ ke deployment instans DB Satu AZ atau Multi-AZ, pastikan untuk menghentikan transaksi apa pun yang sedang ditulis ke klaster DB sumber. Kemudian, tunggu semua pembaruan dilakukan untuk replika baca. Pembaruan database terjadi pada replika baca setelah terjadi pada salah satu instans DB pembaca klaster DB Multi-AZ. Kelambatan replikasi ini dapat sangat bervariasi. Gunakan metrik `ReplicaLag` untuk menentukan saat semua pembaruan sudah dilakukan pada replika baca. Untuk informasi selengkapnya tentang kelambatan replika, lihat the section called “Memantau replikasi baca”.

Setelah Anda mempromosikan replika baca, tunggu status instans DB yang dipromosikan menjadi `Available` sebelum Anda mengarahkan aplikasi Anda untuk menggunakan instans DB yang dipromosikan. Opsional, hapus deployment klaster DB Multi-AZ jika Anda tidak lagi membutuhkannya. Untuk petunjuk, lihat the section called “Menghapus klaster DB Multi-AZ”.

Batasan untuk membuat replika baca instans dari klaster DB Multi-AZ

Batasan berikut berlaku untuk membuat replika baca instans DB dari deployment klaster DB Multi-AZ.

- Anda tidak dapat membuat replika baca instans DB dalam Akun AWS yang berbeda dari Akun AWS yang memiliki cluster DB multi-AZ sumber.

- Anda tidak dapat membuat replika baca instans DB di cluster DB multi-AZ sumber yang Wilayah AWS berbeda.
- Anda tidak dapat memulihkan replika baca instans DB ke suatu titik waktu.
- Enkripsi penyimpanan harus memiliki pengaturan yang sama pada klaster DB Multi-AZ sumber dan replika baca instans DB.
- Jika klaster DB Multi-AZ sumber dienkripsi, replika baca instans DB harus dienkripsi menggunakan tombol KMS yang sama.
- Untuk melakukan upgrade versi minor pada klaster DB Multi-AZ sumber, Anda harus terlebih dahulu melakukan upgrade versi minor pada replika baca instans DB.
- Replika baca instans DB tidak mendukung replika baca berjenjang.
- Untuk RDS for PostgreSQL, klaster DB Multi-AZ sumber harus menjalankan PostgreSQL versi 13.11, 14.8, atau 15.2.R2 atau lebih tinggi untuk membuat replika baca instans DB.
- Anda dapat melakukan peningkatan versi utama pada klaster DB Multi-AZ sumber dari replika baca instans DB, tetapi replikasi ke replika baca berhenti dan tidak dapat dimulai ulang.

Menggunakan replikasi logis PostgreSQL dengan klaster DB Multi-AZ

Dengan menggunakan replikasi logis PostgreSQL dengan klaster DB Multi-AZ Anda, Anda dapat mereplikasi dan menyinkronkan tabel individual daripada seluruh instans basis data. Replikasi logis menggunakan model terbit dan langganan untuk mereplikasi perubahan dari sumber ke satu atau lebih penerima. Hal ini bekerja dengan menggunakan catatan perubahan dari log write-ahead PostgreSQL (WAL). Untuk informasi selengkapnya, lihat the section called “Replikasi logis”.

Saat Anda membuat slot replikasi logis baru pada instans DB penulis dari klaster DB Multi-AZ, slot tersebut disalin secara asinkron ke setiap instans DB pembaca di cluster. Slot pada instans DB pembaca terus disinkronkan dengan yang ada di instans DB penulis.

Replikasi logis didukung untuk klaster DB Multi-AZ yang menjalankan RDS untuk PostgreSQL versi 14.8-R2 dan lebih tinggi, dan 15.3-R2 dan lebih tinggi.

> **Note**
>
> Selain fitur replikasi logis PostgreSQL asli, klaster DB Multi-AZ yang menjalankan RDS untuk PostgreSQL juga mendukung ekstensi `pglogical`.

Untuk informasi selengkapnya tentang fitur replikasi logis PostgreSQL, lihat Replikasi logis dalam dokumentasi PostgreSQL.

Topik

Prasyarat

Untuk mengonfigurasi replikasi logis PostgreSQL untuk klaster DB Multi-AZ, Anda harus memenuhi prasyarat berikut.

- Akun pengguna Anda harus menjadi anggota `rds_superuser` grup dan memiliki `rds_superuser` hak istimewa. Untuk informasi selengkapnya, lihat the section called “Memahami peran dan izin PostgreSQL”.

- Klaster DB Multi-AZ Anda harus dikaitkan dengan grup parameter klaster DB kustom sehingga Anda dapat mengonfigurasi nilai parameter yang dijelaskan dalam prosedur berikut. Untuk informasi selengkapnya, lihat the section called “Bekerja dengan grup parameter klaster DB”.

Menyiapkan replikasi logis

Untuk mengatur replikasi logis untuk klaster DB Multi-AZ, Anda mengaktifkan parameter tertentu dalam grup parameter klaster DB terkait, lalu membuat slot replikasi logis.

> **Note**
>
> Dimulai dengan PostgreSQL versi 16, Anda dapat menggunakan instance DB pembaca dari cluster DB multi-AZ untuk replikasi logis.

Untuk mengatur replikasi logis untuk sebuah RDS untuk klaster DB Multi-AZ PostgreSQL

1. Buka grup parameter klaster DB kustom yang terkait dengan RDS Anda untuk klaster DB Multi-AZ PostgreSQL.
2. Di bidang pencarian Parameter, cari parameter `rds.logical_replication` statis dan atur nilainya ke 1. Perubahan parameter ini dapat meningkatkan pembuatan WAL, jadi aktifkan hanya saat Anda menggunakan slot logis.
3. Sebagai bagian dari perubahan ini, konfigurasi parameter klaster DB berikut.

   - `max_wal_senders`
   - `max_replication_slots`
   - `max_connections`

   Tergantung penggunaan yang Anda harapkan, Anda mungkin juga perlu mengubah nilai parameter berikut. Namun, dalam banyak kasus, nilai default sudah cukup.

   - `max_logical_replication_workers`
   - `max_sync_workers_per_subscription`
4. Reboot klaster DB Multi-AZ agar nilai parameter diterapkan. Untuk petunjuknya, lihat the section called “Membut ulang klaster basis data Multi-AZ”.

5. Buat slot replikasi logis pada instans DB penulis dari klaster DB Multi-AZ seperti yang dijelaskan dalam the section called “Menggunakan slot replikasi logis”. Proses ini mengharuskan Anda menentukan plugin decoding. Saat ini, RDS untuk PostgreSQL mendukung plugin `test_decoding`, `wal2json`, dan `pgoutput` yang dikirimkan dengan PostgreSQL.

   Slot disalin secara asinkron ke setiap instans DB pembaca di klaster.

6. Verifikasi status slot pada semua instans DB pembaca dari klaster DB Multi-AZ. Untuk melakukannya, periksa `pg_replication_slots` tampilan pada semua instans DB pembaca dan pastikan bahwa `confirmed_flush_lsn` status membuat kemajuan saat aplikasi secara aktif mengkonsumsi perubahan logis.

   Perintah berikut menunjukkan cara memeriksa status replikasi pada instans DB pembaca.

```
% psql -h test-postgres-instance-2.abcdefabcdef.us-west-2.rds.amazonaws.com

postgres=> select slot_name, slot_type, confirmed_flush_lsn from
 pg_replication_slots;
  slot_name   | slot_type | confirmed_flush_lsn
--------------+-----------+---------------------
 logical_slot | logical   | 32/D0001700
(1 row)

postgres=> select slot_name, slot_type, confirmed_flush_lsn from
 pg_replication_slots;
  slot_name   | slot_type | confirmed_flush_lsn
--------------+-----------+---------------------
 logical_slot | logical   | 32/D8003628
(1 row)

% psql -h test-postgres-instance-3.abcdefabcdef.us-west-2.rds.amazonaws.com

postgres=> select slot_name, slot_type, confirmed_flush_lsn from
 pg_replication_slots;
  slot_name   | slot_type | confirmed_flush_lsn
--------------+-----------+---------------------
 logical_slot | logical   | 32/D0001700
(1 row)

postgres=> select slot_name, slot_type, confirmed_flush_lsn from
 pg_replication_slots;
  slot_name   | slot_type | confirmed_flush_lsn
--------------+-----------+---------------------
```

```
 logical_slot | logical   | 32/D8003628
(1 row)
```

Setelah Anda menyelesaikan tugas replikasi Anda, hentikan proses replikasi, jatuhkan slot replikasi, dan matikan replikasi logis. Untuk mematikan replikasi logis, ubah grup parameter klaster DB Anda dan atur nilai `rds.logical_replication` kembali ke `0`. Reboot klaster agar perubahan parameter diterapkan.

Batasan dan rekomendasi

Keterbatasan dan rekomendasi berikut berlaku untuk menggunakan replikasi logis dengan cluster DB multi-AZ yang menjalankan PostgreSQL versi 16:

- Anda hanya dapat menggunakan instance DB penulis untuk membuat atau menjatuhkan slot replikasi logis. Misalnya, `CREATE SUBSCRIPTION` perintah harus menggunakan titik akhir penulis cluster dalam string koneksi host.
- Anda harus menggunakan titik akhir penulis cluster selama sinkronisasi tabel atau sinkronisasi ulang. Misalnya, Anda dapat menggunakan perintah berikut untuk menyinkronkan ulang tabel yang baru ditambahkan:

  ```
  Postgres=>ALTER SUBSCRIPTION subscription-name CONNECTION host=writer-endpoint
  Postgres=>ALTER SUBSCRIPTION subscription-name REFRESH PUBLICATION
  ```
- Anda harus menunggu sinkronisasi tabel selesai sebelum menggunakan instance DB pembaca untuk replikasi logis. Anda dapat menggunakan tabel [pg_subscription_rel] katalog untuk memantau sinkronisasi tabel. Sinkronisasi tabel selesai ketika `srsubstate` kolom diatur ke ready (`r`).
- Sebaiknya gunakan titik akhir instance untuk koneksi replikasi logis setelah sinkronisasi tabel awal selesai. Perintah berikut mengurangi beban pada instance DB penulis dengan membongkar replikasi ke salah satu instance DB pembaca:

  ```
  Postgres=>ALTER SUBSCRITPION subscription-name CONNECTION host=reader-instance-
  endpoint
  ```

  Anda tidak dapat menggunakan slot yang sama pada lebih dari satu instans DB sekaligus. Ketika dua atau lebih aplikasi mereplikasi perubahan logis dari instance DB yang berbeda di cluster, beberapa perubahan mungkin hilang karena failover cluster atau masalah jaringan. Dalam situasi ini, Anda dapat menggunakan titik akhir instance untuk replikasi logis dalam string koneksi host.

Aplikasi lain yang menggunakan konfigurasi yang sama akan menampilkan pesan kesalahan berikut:

```
replication slot slot_name is already active for PID x providing immediate feedback.
```

- Cluster DB multi-AZ yang menjalankan PostgreSQL versi 16.1 tidak dapat menghidupkan kembali slot replikasi logis yang hilang pada instance DB. Penyebab umum adalah failover yang diprakarsai oleh `failover-db-cluster` API, atau peristiwa infrastruktur.
- Saat menggunakan `pglogical` ekstensi, Anda hanya dapat menggunakan titik akhir penulis cluster. Ekstensi memiliki keterbatasan yang diketahui yang dapat membuat slot replikasi logis yang tidak digunakan selama sinkronisasi tabel. Slot replikasi basi menyimpan file write-ahead log (WAL) dan dapat menyebabkan masalah ruang disk.

Menghapus klaster DB Multi-AZ

Anda dapat menghapus cluster DB Multi-AZ DB menggunakan AWS Management Console, AWS CLI, atau RDS API. Untuk menghapus cluster DB multi-AZ, Anda harus terlebih dahulu menghapus semua instans DB itu.

Waktu yang diperlukan untuk menghapus cluster DB multi-AZ dapat bervariasi tergantung pada faktor-faktor berikut:

- Anda periode retensi cadangan (yaitu, berapa banyak cadangan yang harus dihapus).
- Berapa banyak data yang dihapus.
- Apakah snapshot terakhir diambil.

Perlindungan penghapusan harus dinonaktifkan pada cluster DB multi-AZ sebelum Anda dapat menghapusnya. Untuk informasi selengkapnya, lihat the section called “Prasyarat untuk menghapus instans DB”. Anda dapat menonaktifkan perlindungan penghapusan dengan memodifikasi cluster DB multi-AZ. Untuk informasi selengkapnya, lihat the section called “Mengubah klaster basis data Multi-AZ”.

Konsol

Untuk menghapus klaster DB Multi-AZ

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data, lalu pilih klaster DB Multi-AZ yang ingin dihapus.
3. Untuk Tindakan, pilih Hapus.
4. Pilih Buat cuplikan akhir? untuk membuat snapshot DB akhir untuk klaster DB Multi-AZ.

   Jika Anda membuat snapshot akhir, masukkan nama untuk Nama snapshot akhir.
5. Pilih Pertahankan cadangan otomatis untuk menyimpan cadangan otomatis.
6. Masukkan **`delete me`** di kotak.
7. Pilih Hapus.

AWS CLI

Untuk menghapus cluster DB Multi-AZ dengan menggunakan AWS CLI, panggil delete-db-clusterperintah dengan opsi berikut:

- `--db-cluster-identifier`
- `--final-db-snapshot-identifier` atau `--skip-final-snapshot`

Example Dengan snapshot akhir

Untuk Linux, macOS, atau Unix:

```
aws rds delete-db-cluster \
    --db-cluster-identifier mymultiazdbcluster \
    --final-db-snapshot-identifier mymultiazdbclusterfinalsnapshot
```

Untuk Windows:

```
aws rds delete-db-cluster ^
    --db-cluster-identifier mymultiazdbcluster ^
    --final-db-snapshot-identifier mymultiazdbclusterfinalsnapshot
```

Example Dengan snapshot akhir

Untuk Linux, macOS, atau Unix:

```
aws rds delete-db-cluster \
    --db-cluster-identifier mymultiazdbcluster \
    --skip-final-snapshot
```

Untuk Windows:

```
aws rds delete-db-cluster ^
    --db-cluster-identifier mymultiazdbcluster ^
    --skip-final-snapshot
```

RDS API

Untuk menghapus klaster DB Multi-AZ menggunakan Amazon RDS API, panggil operasi DeleteDBCluster dengan parameter berikut ini:

- `DBClusterIdentifier`
- `FinalDBSnapshotIdentifier` atau `SkipFinalSnapshot`

Keterbatasan cluster DB multi-AZ

Klaster basis data Multi-AZ memiliki satu instans basis data penulis dan dua instans basis data pembaca dalam tiga Zona Ketersediaan terpisah. Klaster DB Multi-AZ menyediakan ketersediaan tinggi, peningkatan kapasitas untuk beban kerja baca, dan latensi yang lebih rendah jika dibandingkan dengan deployment Multi-AZ. Lihat informasi yang lebih lengkap tentang klaster basis data Multi-AZ di Deployment klaster basis data Multi-AZ.

Batasan berikut berlaku untuk cluster DB multi-AZ.

- Klaster basis data Multi-AZ tidak mendukung fitur-fitur berikut:
  - Koneksi IPv6 (mode dual-stack)
  - Cadangan otomatis lintas Wilayah
  - Autentikasi IAM DB dan otentikasi Kerberos
  - Memodifikasi port. Sebagai gantinya, Anda dapat memulihkan klaster basis data Multi-AZ ke suatu titik waktu dan menentukan port yang berbeda.
  - Grup opsi
  - P oint-in-time-recovery (PITR) untuk cluster yang dihapus
  - Mengekspor data snapshot cluster DB multi-AZ ke bucket S3, atau memulihkan snapshot cluster DB multi-AZ dari bucket S3
  - Penyimpanan autoscaling dengan mengatur penyimpanan maksimum yang dialokasikan. Sebagai gantinya, Anda dapat menskalakan penyimpanan secara manual.
  - Menghentikan dan memulai cluster DB Multi-AZ
  - Menyalin cuplikan dari klaster basis data Multi-AZ
  - Mengenkripsi klaster basis data Multi-AZ yang tidak terenkripsi
- Klaster basis data Multi-AZ RDS for MySQL tidak mendukung replikasi ke basis data target eksternal.
- Klaster basis data Multi-AZ RDS for MySQL hanya mendukung prosedur-prosedur tersimpan sistem berikut:
  - `mysql.rds_rotate_general_log`
  - `mysql.rds_rotate_slow_log`
  - `mysql.rds_show_configuration`
  - `mysql.rds_set_external_master_with_auto_position`

- RDS untuk cluster DB PostgreSQL Multi-AZ tidak mendukung ekstensi berikut: dan. `aws_s3` `pg_transport`
- Klaster basis data Multi-AZ RDS for PostgreSQL tidak mendukung penggunaan server DNS kustom untuk akses jaringan keluar.

Menggunakan Dukungan Diperpanjang Amazon RDS

Dengan Dukungan Diperpanjang Amazon RDS, Anda dapat terus menjalankan basis data Anda pada suatu versi mesin utama melewati akhir tanggal dukungan standar RDS dengan biaya tambahan. Pada akhir tanggal dukungan standar RDS , Amazon RDS Amazon secara otomatis mendaftarkan database Anda di RDS Extended Support. Pendaftaran otomatis ke RDS Extended Support tidak mengubah mesin database dan tidak memengaruhi waktu aktif atau kinerja instans DB Anda.

Penawaran berbayar ini memberi Anda lebih banyak waktu untuk meningkatkan ke versi mesin utama yang didukung. Selama RDS Extended Support, Amazon RDS akan memasok patch untuk CVE Kritis dan Tinggi seperti yang didefinisikan oleh peringkat tingkat keparahan CVSS National Vulnerability Database (NVD). Lihat informasi yang lebih lengkap di Metrik Kerentanan.

Anda juga dapat membuat basis data baru dengan versi mesin utama yang telah mencapai akhir tanggal dukungan standar RDS. RDS secara otomatis mendaftarkan database baru ini di RDS Extended Support dan menagih Anda untuk penawaran ini.

Untuk melakukan ini, gunakan AWS CLI atau API RDS. Dalam AWS CLI, `open-source-rds-extended-support-disabled` tentukan `--engine-lifecycle-support` opsi. Di RDS API, tentukan `open-source-rds-extended-support-disabled` `LifeCycleSupport` parameternya.

Misalnya, akhir tanggal dukungan standar RDS untuk RDS for MySQL versi 5.7 adalah 29 Februari 2024. Namun, Anda tidak siap untuk secara manual meng-upgrade ke RDS untuk MySQL versi 8.0 sebelum tanggal tersebut. Pada tanggal 29 Februari 2024, Amazon RDS secara otomatis mendaftarkan database Anda di RDS Extended Support, dan Anda dapat terus menjalankan RDS untuk MySQL versi 5.7. Mulai 1 Maret 2024, Amazon RDS secara otomatis menagih Anda untuk RDS Extended Support.

RDS Extended Support tersedia hingga 3 tahun setelah akhir RDS dari tanggal dukungan standar untuk versi mesin utama MySQL versi 2). Setelah waktu ini, jika Anda belum memutakhirkan versi mesin utama Anda ke versi yang didukung, Amazon RDS Amazon akan secara otomatis meningkatkan versi mesin utama Anda. Kami menyarankan agar Anda meningkatkan ke versi mesin utama sesegera mungkin.

Topik

- Biaya Amazon RDS Extended Support

- Versi dengan Amazon RDS Extended Support
- Penamaan versi Amazon RDS Extended Support
- Membuat instans DB atau cluster DB multi-AZ, cluster Extended Support
- Melihat pendaftaran instans DB atau klaster DB multi-AZ Kluster Aurora DB atau
- Memulihkan instans DB atau cluster DB multi-AZ, cluster Support

Biaya Amazon RDS Extended Support

Biaya tambahan untuk RDS Extended Support secara otomatis berhenti segera setelah Anda meningkatkan ke versi engine yang tercakup dalam dukungan standar, atau Anda menghapus database yang menjalankan versi utama melewati akhir RDS dari tanggal dukungan standar. Namun, pengisian daya akan dimulai kembali jika versi mesin target Anda memasuki RDS Extended Support di masa mendatang.

Misalnya, RDS untuk PostgreSQL 11 memasuki Extended Support pada 1 Maret 2024, tetapi biaya tidak dimulai hingga 1 April 2024. Namun, jika Anda terus menjalankan RDS untuk PostgreSQL 12 pada instans DB ini melewati akhir RDS tanggal dukungan standar 28 Februari 2025, maka database Anda akan kembali dikenakan biaya RDS Extended Support mulai 1 Maret 2025.

Biaya RDS Extended Support berlaku untuk instans siaga di penerapan Multi-AZ.

Lihat informasi yang lebih lengkap di Struktur harga Amazon RDS for MySQL dan Struktur harga Amazon RDS for PostgreSQL.

Versi dengan Amazon RDS Extended Support

RDS Extended Support hanya tersedia untuk versi utama. Ini tidak tersedia untuk versi minor.

RDS Extended Support tersedia untuk RDS untuk MySQL 5.7 dan 8.0, dan untuk RDS untuk PostgreSQL 11 dan lebih tinggi. Lihat informasi yang lebih lengkap di Versi utama MySQL yang didukung dan Kalender rilis untuk Amazon RDS for PostgreSQL.

Amazon RDS secara otomatis memutakhirkan instans DB Anda ke versi minor terakhir yang dirilis sebelum tanggal dukungan standar berakhir RDS, jika Anda belum menjalankan versi tersebut. Amazon RDS tidak akan memutakhirkan versi minor Anda sampai setelah RDS berakhir tanggal dukungan standar untuk versi mesin utama Anda. Untuk informasi selengkapnya, lihat Versi kecil MySQL yang didukung di Amazon RDS dan Rilis kalender untuk Amazon RDS for PostgreSQL.

Penamaan versi Amazon RDS Extended Support

Amazon RDS akan merilis versi minor baru dengan perbaikan dan patch CVE untuk engine pada RDS Extended Support. Nama-nama rilis minor ini akan dalam bentuk major.minor-RDS.YYYYMMDD.patch.YYYYMMDD, misalnya, 5.7.44-RDS.20240208.R2.20240210 (untuk RDS untuk MySQL) 11.22-RDS.20240208.R2.20240210 atau (untuk RDS untuk PostgreSQL).

mayor

Untuk MySQL, nomor versi utama adalah bilangan bulat dan bagian fraksional pertama dari nomor versi, misalnya, 8.0. Peningkatan versi mayor akan meningkatkan bagian mayor dari nomor versi. Misalnya, upgrade dari 5.7.44 ke 8.0.33 adalah upgrade versi utama, di mana 5.7 dan 8.0 adalah nomor versi utama.

Untuk PostgreSQL, nomor versi utama adalah bilangan bulat, misalnya, 11.

Minor-RDS.YYYYMMDD

Untuk MySQL, nomor versi minor adalah bagian ketiga dari nomor versi, misalnya, in. 44-RDS.20240208 5.7.44-RDS.20240208

Untuk PostgeSQL, nomor versi minor adalah bagian kedua dari nomor versi, misalnya, in. 22-RDS.20240208 11.22-RDS.20240208

Tanggalnya adalah ketika Amazon RDS membuat versi minor Amazon RDS.

tambalan

Versi tambalan adalah yang mengikuti tanggal ketika Amazon RDS membuat versi minor Amazon RDS, misalnya, R2 di atau. 5.7.44-RDS.20240208.R2 11.22-RDS.20240208.R2

Versi patch Amazon RDS mencakup perbaikan bug penting yang ditambahkan ke versi minor Amazon RDS setelah dirilis.

YYYMMDD

Tanggalnya adalah ketika Amazon RDS membuat versi tambalan, misalnya, 20240210 di atau. 5.7.44-RDS.20240208.R2.20240210 11.22-RDS.20240208.R2.20240210

Versi tanggal Amazon RDS adalah patch keamanan yang mencakup perbaikan keamanan penting yang ditambahkan ke versi minor setelah dirilis. Itu tidak termasuk perbaikan apa pun yang mungkin mengubah perilaku mesin.

Membuat instans DB atau cluster DB multi-AZ, cluster Extended Support

Saat Anda membuat instans DB atau cluster DB multi-AZ, cluster , pilih Aktifkan RDS Extended Support di konsol, atau gunakan opsi Extended Support AWS CLI di atau parameter di RDS API.

> **Note**
>
> Jika Anda tidak menentukan pengaturan RDS Extended Support, RDS default ke RDS Extended Support. Perilaku default ini menjaga ketersediaan database Anda melewati akhir RDS dari tanggal dukungan standar.

Topik

Pertimbangan untuk RDS Extended Support

Sebelum membuat instans DB atau cluster DB multi-AZ, cluster , pertimbangkan item berikut:

- Setelah tanggal dukungan standar RDS berakhir, Anda dapat mencegah pembuatan instans DB baru atau cluster DB multi-AZ baru, cluster Aurora DB baru atau Support. Untuk melakukan ini, gunakan AWS CLI atau RDS API. Dalam AWS CLI, `open-source-rds-extended-support-disabled` tentukan `--engine-lifecycle-support` opsi. Di RDS API, tentukan `open-source-rds-extended-support-disabled` `LifeCycleSupport` parameternya. Jika Anda menentukan `open-source-rds-extended-support-disabled` dan akhir RDS dari tanggal dukungan standar telah berlalu, membuat instans DB atau cluster DB multi-AZ, cluster akan selalu gagal.
- RDS Extended Support diatur pada tingkat cluster. Anggota cluster akan selalu memiliki pengaturan yang sama untuk RDS Extended Support di konsol RDS, `--engine-lifecycle-support` di AWS CLI, dan `EngineLifecycleSupport` di RDS API.

Untuk informasi selengkapnya, lihat Versi MySQL dan Rilis kalender untuk Amazon RDS for PostgreSQL.

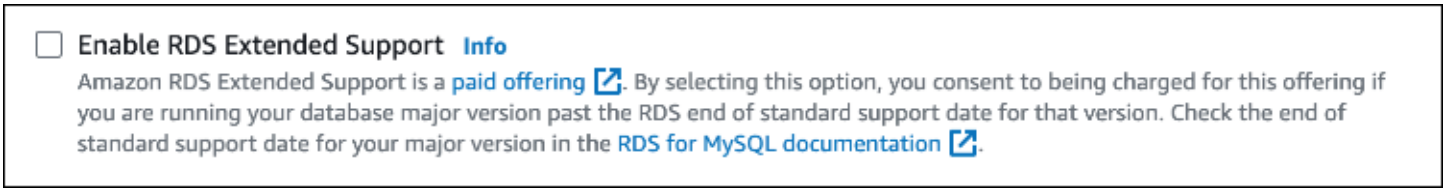
Buat instans DB atau cluster DB multi-AZ, cluster Extended Support

Anda dapat membuat instans DB atau cluster DB multi-AZ, cluster dengan versi RDS Extended Support AWS Management Console menggunakan AWS CLI,, atau RDS API.

Konsol

Saat Anda membuat cluster DB multi-AZ, di bagian opsi Engine, pilih Enable RDS Extended Support.

Gambar berikut menunjukkan pengaturan Enable RDS Extended Support:



AWS CLI

Saat Anda menggunakan AWS CLI perintah (Multi-AZ DB cluster), pilih RDS Extended Support dengan menentukan `open-source-rds-extended-support` opsi tersebut. `--engine-lifecycle-support` Secara default, opsi ini diatur ke`open-source-rds-extended-support`.

Untuk mencegah pembuatan instans DB baru atau cluster DB multi-AZ, cluster setelah RDS berakhir dari tanggal dukungan standar, tentukan opsi. `open-source-rds-extended-support-disabled` `--engine-lifecycle-support` Dengan demikian, Anda akan menghindari biaya RDS Extended Support terkait.

API RDS

Saat Anda menggunakan operasi) Amazon RDS API, pilih RDS Extended Support dengan menyetel parameter ke. `EngineLifecycleSupport` `open-source-rds-extended-support` Secara default, parameter ini diatur ke`open-source-rds-extended-support`.

Untuk mencegah pembuatan instans DB baru atau cluster DB multi-AZ, cluster setelah RDS berakhir dari tanggal dukungan standar, tentukan parameternya. `open-source-rds-extended-support-disabled` `EngineLifecycleSupport` Dengan demikian, Anda akan menghindari biaya RDS Extended Support terkait.

Untuk informasi selengkapnya, lihat topik berikut:

- Untuk membuat instans DB, ikuti petunjuk untuk mesin DB Anda di Membuat instans DB Amazon RDS.

- Untuk membuat klaster DB Multi-AZ, ikuti petunjuk untuk mesin DB Anda di Membuat klaster DB Multi-AZ.

Melihat pendaftaran instans DB atau klaster DB multi-AZ Kluster Aurora DB atau

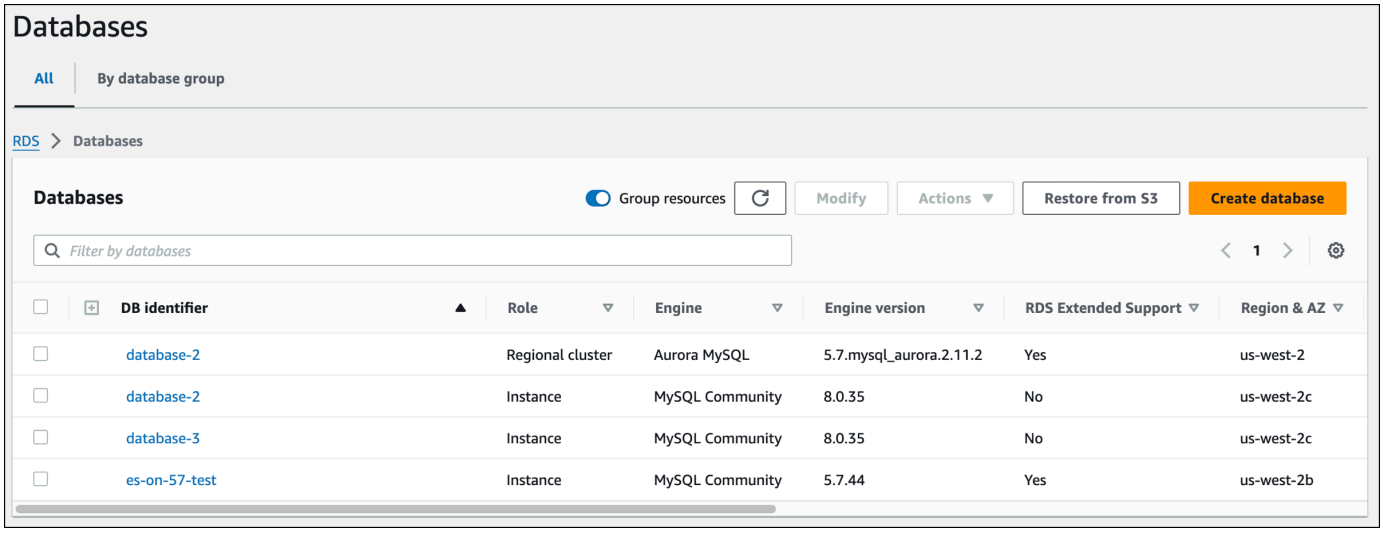
Anda dapat melihat pendaftaran instans DB atau cluster DB multi-AZ cluster Aurora DB atau cluster menggunakan file. AWS Management Console

Konsol

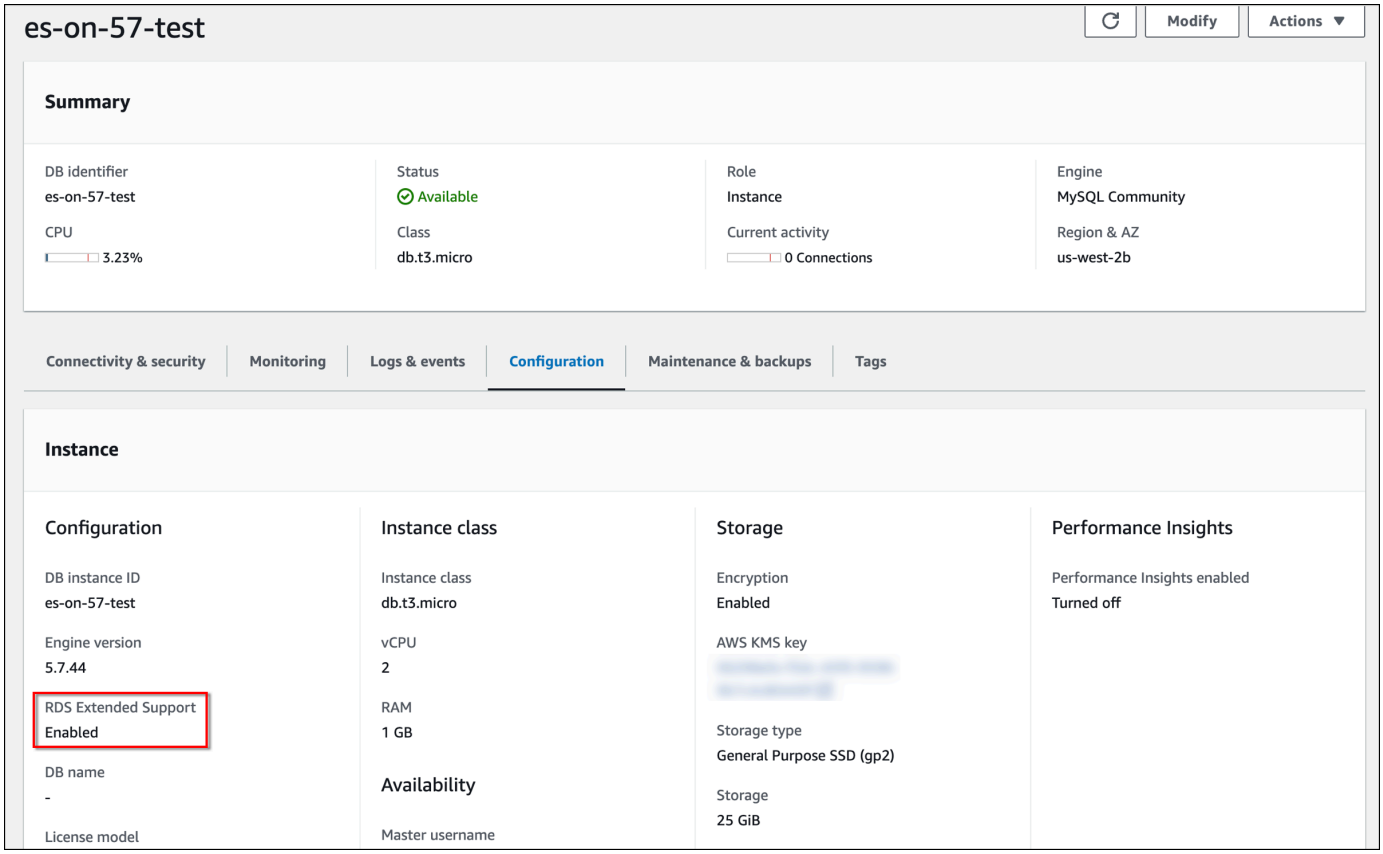
Untuk melihat pendaftaran instans DB atau cluster DB multi-AZ, klaster Aurora DB atau cluster global

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data. Nilai di bawah RDS Extended Support menunjukkan jika instans DB atau cluster DB multi-AZ cluster terdaftar di RDS Extended Support. Jika tidak ada nilai yang muncul, maka RDS Extended Support tidak tersedia untuk database Anda.

> **Tip**
>
> Jika kolom RDS Extended Support tidak muncul, pilih ikon Preferensi, lalu aktifkan RDS Extended Support.



3. Anda juga dapat melihat pendaftaran pada tab Konfigurasi untuk setiap database. Pilih database di bawah pengenal DB. Pada tab Configuration, lihat di bawah Extended Support untuk melihat apakah database terdaftar atau tidak.



Memulihkan instans DB atau cluster DB multi-AZ, cluster Support

Saat memulihkan instans DB atau cluster DB multi-AZ, cluster , pilih Aktifkan RDS Extended Support di konsol, atau gunakan opsi Extended Support AWS CLI di atau parameter di RDS API.

> **Note**
>
> Jika Anda tidak menentukan pengaturan RDS Extended Support, RDS default ke RDS Extended Support. Perilaku default ini menjaga ketersediaan database Anda melewati akhir RDS dari tanggal dukungan standar.

Topik

- Pertimbangan untuk RDS Extended Support

- Kembalikan instans DB atau cluster DB multi-AZ, cluster DB cluster Extended Support

Pertimbangan untuk RDS Extended Support

Sebelum memulihkan instans DB atau cluster DB multi-AZ, cluster , pertimbangkan item berikut:

- Setelah tanggal dukungan standar RDS berakhir, jika Anda ingin memulihkan instans DB atau cluster DB multi-AZ, cluster Amazon S3, Anda hanya dapat melakukannya dengan menggunakan atau RDS API. AWS CLI Gunakan `--engine-lifecycle-support` opsi dalam AWS CLI perintah restore-db-cluster-from-s3 atau `EngineLifecycleSupport` parameter dalam operasi API RDS RestoreDB ClusterFrom S3.
- Jika Anda ingin mencegah RDS memulihkan database Anda ke versi RDS Extended Support, `open-source-rds-extended-support-disabled` tentukan di AWS CLI atau RDS API. Dengan demikian, Anda akan menghindari biaya RDS Extended Support terkait.

  Jika Anda menentukan setelan ini, Amazon RDS Aurora akan secara otomatis memutakhirkan database Anda yang dipulihkan ke versi utama yang lebih baru dan didukung. Jika pemutakhiran gagal dalam pemeriksaan pra-pemutakhiran, Amazon RDS akan kembali dengan aman ke versi mesin RDS Extended Support. Basis data ini akan tetap dalam mode RDS Extended Support, dan Amazon RDS Amazon akan menagih Anda untuk RDS Extended Support hingga Anda memutakhirkan database secara manual.

  Misalnya, jika Anda memulihkan snapshot MySQL 5.7 tanpa menggunakan RDS Extended Support, Amazon RDS akan mencoba untuk secara otomatis meng-upgrade database Anda ke MySQL 8.0. Jika pemutakhiran ini gagal karena masalah yang perlu Anda selesaikan, Amazon RDS akan mengembalikan database ke MySQL 5.7. Amazon RDS akan menyimpan database pada RDS Extended Support sampai Anda dapat memperbaiki masalah. Misalnya, peningkatan mungkin gagal karena ruang penyimpanan yang tidak mencukupi. Setelah Anda memperbaiki masalah, Anda harus memulai upgrade. Setelah upaya pertama untuk memutakhirkan database Anda, Amazon RDS tidak akan mencoba memutakhirkannya lagi.
- RDS Extended Support diatur pada tingkat cluster. Anggota cluster akan selalu memiliki pengaturan yang sama untuk RDS Extended Support di konsol RDS, `--engine-lifecycle-support` di AWS CLI, dan `EngineLifecycleSupport` di RDS API.

Untuk informasi selengkapnya, lihat Versi MySQL dan Rilis kalender untuk Amazon RDS for PostgreSQL.

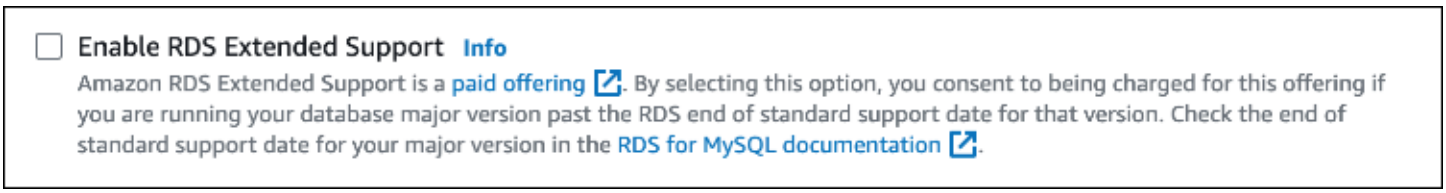
Kembalikan instans DB atau cluster DB multi-AZ, cluster DB cluster Extended Support

Anda dapat memulihkan instans DB atau cluster DB multi-AZ, cluster dengan versi RDS Extended Support AWS Management Console menggunakan AWS CLI,, atau RDS API.

Konsol

Saat Anda memulihkan cluster DB Multi-AZ, pilih Aktifkan RDS Extended Support di bagian opsi Engine.

Gambar berikut menunjukkan pengaturan Enable RDS Extended Support:



AWS CLI

Saat Anda menggunakan AWS CLI perintah atau restore-db-cluster-from -snapshot, pilih RDS Extended Support dengan menentukan opsi. `open-source-rds-extended-support` `--engine-lifecycle-support`

Jika Anda ingin menghindari biaya yang terkait dengan RDS Extended Support, atur `--engine-lifecycle-support` opsi ke`open-source-rds-extended-support-disabled`. Secara default, opsi ini diatur ke`open-source-rds-extended-support`.

Anda juga dapat menentukan nilai ini menggunakan AWS CLI perintah berikut:

- restore-db-cluster-from-s3
- restore-db-cluster-to-point-in-time
- restore-db-instance-from-s3
- restore-db-instance-to-point-in-time

API RDS

Saat Anda menggunakan operasi RestoreDB atau ClusterFromSnapshot RestoreDB ClusterFromSnapshot RDS API, pilih RDS Extended Support dengan menyetel parameter ke. `EngineLifecycleSupport` `open-source-rds-extended-support`

Jika Anda ingin menghindari biaya yang terkait dengan RDS Extended Support, atur `EngineLifecycleSupport` parameternya ke`open-source-rds-extended-support-disabled`. Secara default, parameter ini diatur ke`open-source-rds-extended-support`.

Anda juga dapat menentukan nilai ini menggunakan operasi API RDS berikut:

- DipulihkanB S3 ClusterFrom
- DipulihkanB ClusterToPointInTime
- DipulihkanB S3 InstanceFrom
- DipulihkanB InstanceToPointInTime

Untuk informasi selengkapnya tentang memulihkan instans DB atau cluster DB multi-AZ, ikuti petunjuk untuk mesin DB Anda. Memulihkan dari snapshot DB

Menggunakan Deployment Blue/Green Amazon RDS untuk pembaruan basis data

Deployment blue/green menyalin lingkungan basis data produksi ke lingkungan penahapan yang terpisah dan tersinkron. Dengan menggunakan Deployment Blue/Green Amazon RDS, Anda dapat membuat perubahan pada basis data di lingkungan penahapan tanpa memengaruhi lingkungan produksi. Misalnya, Anda dapat meningkatkan versi mesin DB besar atau kecil, mengubah parameter basis data, atau membuat perubahan skema di lingkungan penahapannya. Saat Anda siap, Anda dapat mempromosikan lingkungan pementasan menjadi lingkungan basis data produksi baru, dengan waktu henti biasanya di bawah satu menit.

> **Note**
>
> Saat ini, Penerapan Biru/Hijau didukung untuk RDS untuk MariaDB, RDS untuk MySQL, dan RDS untuk PostgreSQL saja. Untuk ketersediaan Amazon Aurora, lihat Menggunakan Deployment Blue/Green Amazon RDS untuk pembaruan basis data di Panduan Pengguna Amazon Aurora.

Topik

Gambaran Umum Deployment Blue/Green Amazon RDS

Dengan menggunakan Deployment Blue/Green Amazon RDS, Anda dapat membuat dan menguji perubahan basis data sebelum menerapkannya di lingkungan produksi. Deployment blue/green menciptakan lingkungan pementasan yang menyalin lingkungan produksi. Dalam deployment blue/green, lingkungan biru adalah lingkungan produksi saat ini. Lingkungan hijau adalah lingkungan pementasannya. Lingkungan pementasan tetap sinkron dengan lingkungan produksi saat ini menggunakan replikasi logis.

Anda dapat membuat perubahan pada instans DB RDS di lingkungan hijau tanpa memengaruhi beban kerja produksi. Misalnya, Anda dapat meningkatkan versi mesin DB mayor atau minor, meningkatkan konfigurasi sistem file yang mendasarinya, atau mengubah parameter basis data di lingkungan pementasannya. Anda dapat menguji perubahan di lingkungan hijau secara menyeluruh. Setelah siap, Anda dapat melakukan switchover lingkungan untuk mempromosikan lingkungan hijau menjadi lingkungan produksi baru. Switchover biasanya memakan waktu kurang dari satu menit tanpa kehilangan data dan tidak perlu mengubah aplikasi.

Karena lingkungan hijau adalah salinan dari topologi lingkungan produksi, lingkungan hijau mencakup fitur yang digunakan oleh instans DB. Fitur-fitur ini termasuk replika baca, konfigurasi penyimpanan, snapshot DB, pencadangan otomatis, Wawasan Performa, dan Pemantauan yang Ditingkatkan. Jika instans DB biru adalah deployment instans DB Multi-AZ, instans DB hijau juga merupakan deployment instans DB Multi-AZ.

> **Note**
>
> Saat ini, Deployment Blue/Green hanya didukung untuk RDS for MariaDB, RDS for MySQL, dan RDS for PostgreSQL. Untuk ketersediaan Amazon Aurora, lihat Menggunakan Amazon RDS Blue/Green Deployment untuk pembaruan database di Panduan Pengguna Amazon Aurora.

Topik

- Ketersediaan wilayah dan versi
- Batasan untuk deployment blue/green

Manfaat menggunakan Deployment Blue/Green Amazon RDS

Dengan menggunakan Deployment Blue/Green Amazon RDS, Anda dapat tetap mengikuti perkembangan patch keamanan, meningkatkan performa basis data, dan mengadopsi fitur basis data yang lebih baru dengan waktu henti yang singkat dan dapat diprediksi. Deployment blue/green mengurangi risiko dan waktu henti untuk pembaruan basis data, seperti peningkatan versi mesin mayor atau minor.

Deployment blue/green memberikan manfaat berikut:

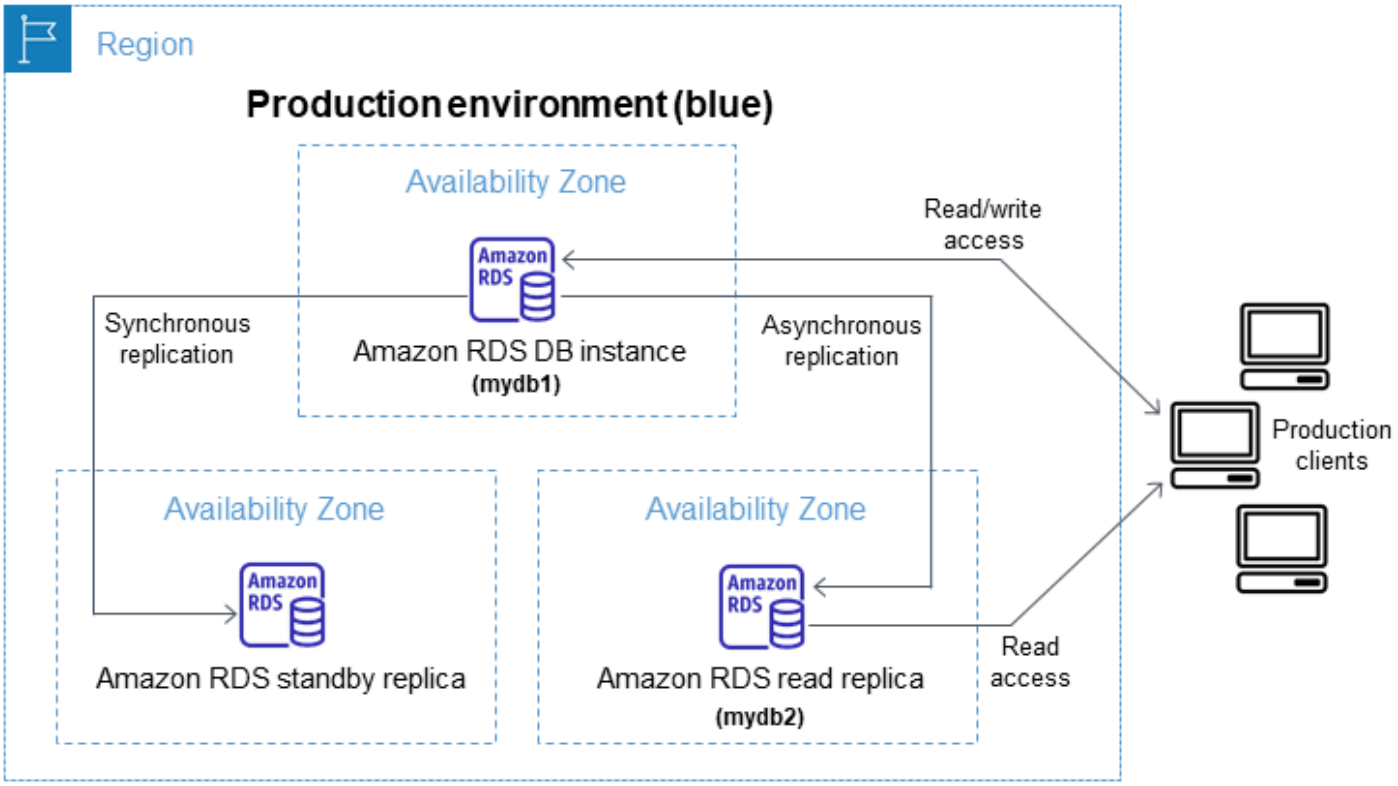
- Memudahkan pembuatan lingkungan pementasan siap produksi.
- Mereplikasi otomatis perubahan basis data dari lingkungan produksi ke lingkungan pementasan.
- Menguji perubahan basis data di lingkungan pementasan yang aman tanpa memengaruhi lingkungan produksi.
- Anda dapat mengikuti perkembangan terbaru dengan patch basis data dan pembaruan sistem.
- Menerapkan dan menguji fitur basis data yang lebih baru.
- Melakukan switchover pada lingkungan pementasan untuk menjadi lingkungan produksi baru tanpa perubahan pada aplikasi.
- Melakukan switchover dengan aman melalui penggunaan pagar pembatas switchover default.
- Tidak ada kehilangan data selama switchover.
- Melakukan switchover dengan cepat, biasanya kurang dari satu menit tergantung beban kerja Anda.

Alur kerja deployment blue/green

Selesaikan langkah-langkah utama berikut saat Anda menggunakan deployment blue/green untuk pembaruan basis data.

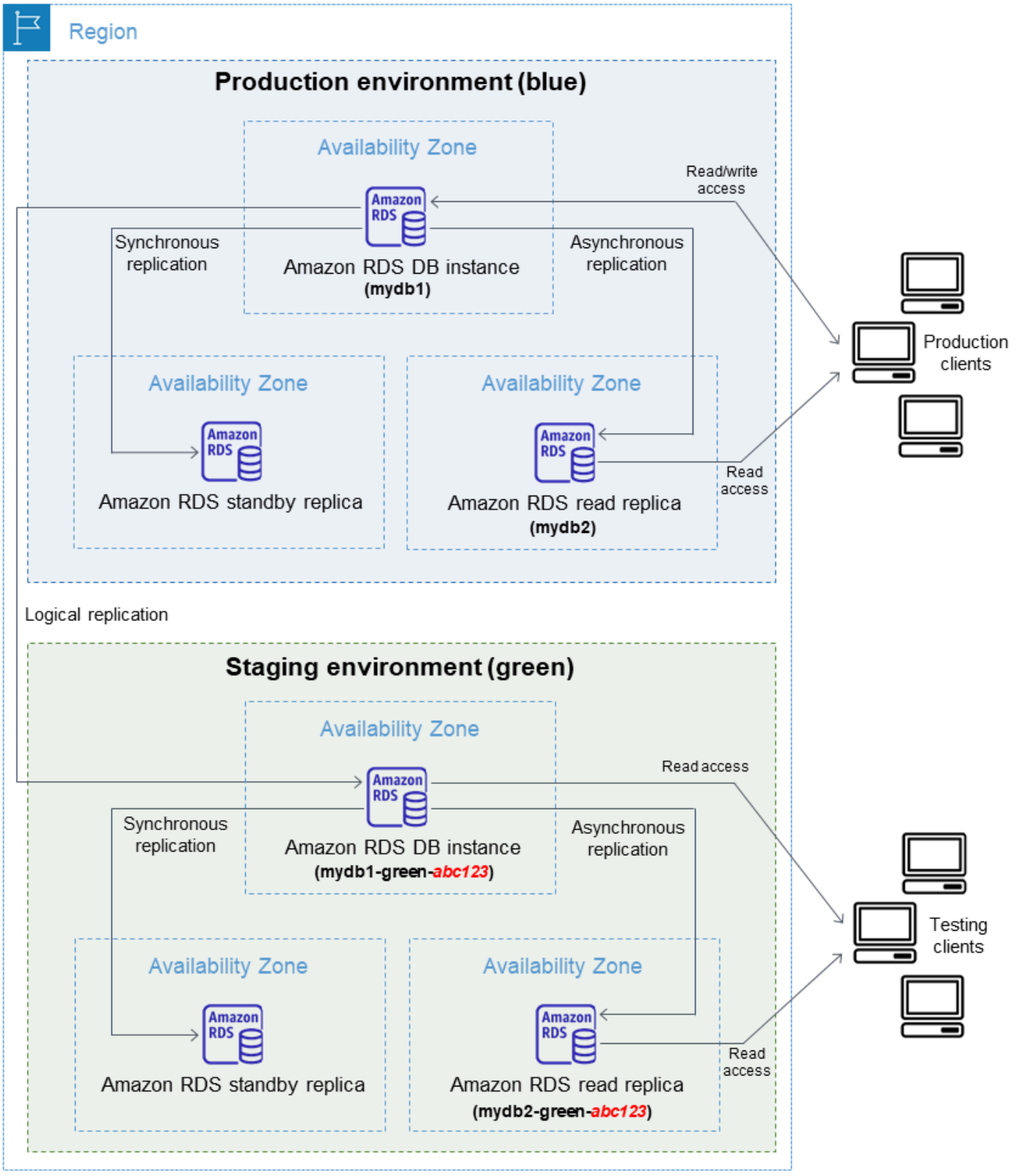
1. Identifikasi lingkungan produksi yang membutuhkan pembaruan.

   Misalnya, lingkungan produksi dalam gambar ini memiliki deployment instans DB multi-AZ (mydb1) dan replika baca (mydb2).



2. Buat deployment blue/green. Untuk petunjuknya, lihat Membuat deployment blue/green.

   Gambar berikut menunjukkan contoh deployment blue/green pada lingkungan produksi dari langkah 1. Saat membuat deployment blue/green, RDS menyalin topologi lengkap dan konfigurasi instans DB primer untuk menciptakan lingkungan hijau. Nama instans DB yang disalin ditambahkan dengan `-green-`*`random-characters`*. Lingkungan pementasan dalam gambar berisi deployment instans DB Multi-AZ (mydb1-green-*`abc123`*) dan replika baca (mydb2-green-*`abc123`*).



Saat membuat deployment blue/green, Anda dapat meningkatkan versi mesin DB Anda dan menentukan grup parameter DB yang berbeda untuk instans DB di lingkungan hijau. RDS juga

mengonfigurasi replikasi logis dari instans DB primer di lingkungan biru ke instans DB primer di lingkungan hijau.

Setelah Anda membuat deployment blue/green, instans DB di lingkungan hijau bersifat hanya baca secara default.

3. Buat perubahan tambahan pada lingkungan pementasan, jika diperlukan.

   Misalnya, Anda dapat membuat perubahan skema pada basis data Anda atau mengubah kelas instans DB yang digunakan oleh satu atau beberapa instans DB di lingkungan hijau.

   Untuk informasi tentang memodifikasi instans DB, lihat Memodifikasi instans DB Amazon RDS.

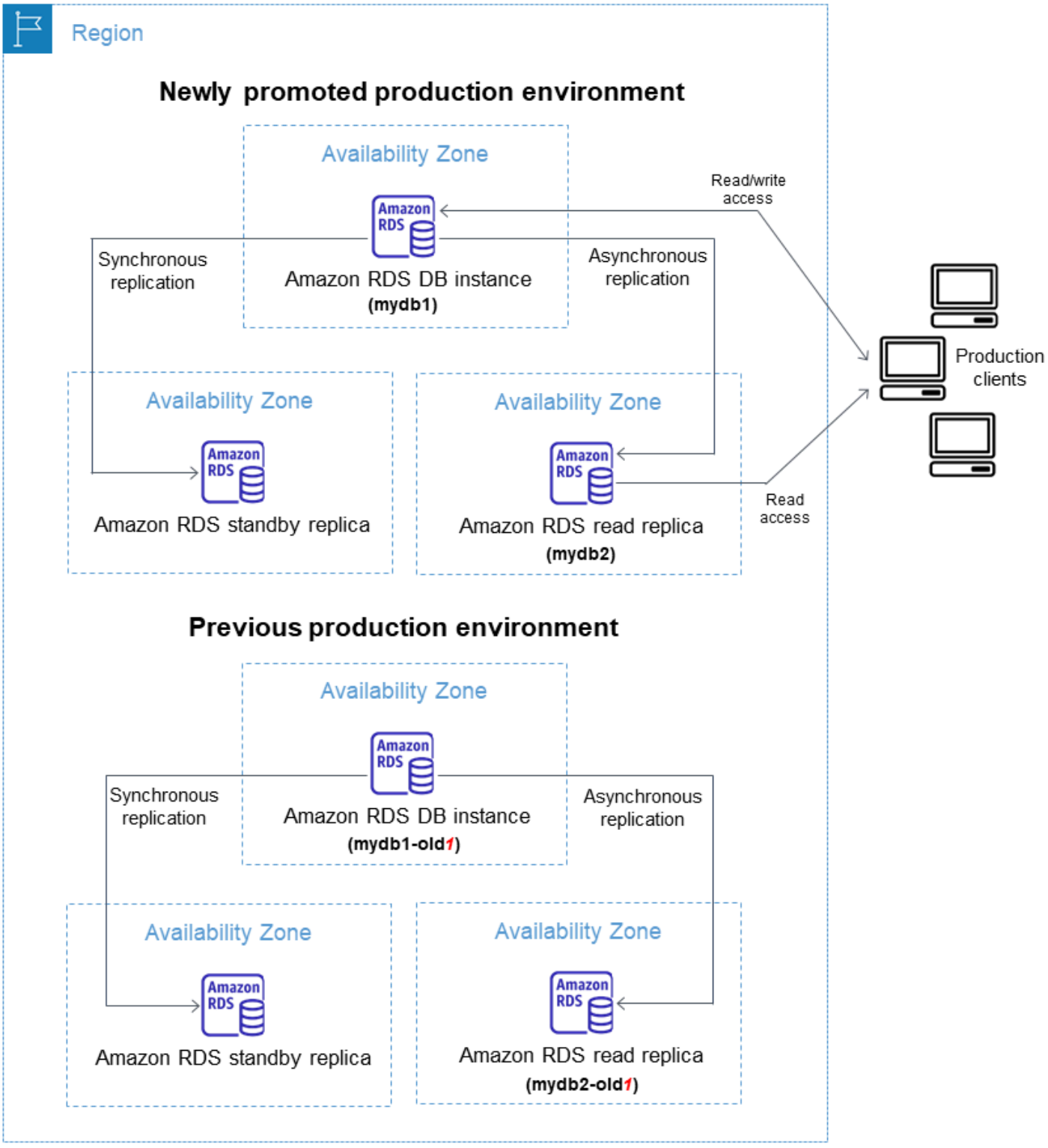
4. Uji lingkungan pementasan Anda.

   Selama pengujian, sebaiknya pertahankan basis data Anda di lingkungan hijau agar hanya dapat dibaca saja. Aktifkan operasi tulis di lingkungan hijau dengan hati-hati karena dapat mengakibatkan konflik replikasi. Hal ini juga dapat menghasilkan data yang tidak diinginkan dalam basis data produksi setelah switchover. Untuk mengaktifkan operasi tulis untuk RDS untuk MySQL, atur `read_only` parameternya `0` ke, lalu reboot instance DB. Untuk RDS untuk PostgreSQL, atur parameter `default_transaction_read_only` ke level sesi. `off`

5. Saat siap, lakukan switchover untuk mempromosikan lingkungan pementasan menjadi lingkungan produksi baru. Untuk petunjuknya, lihat Mengganti deployment blue/green.

   Switchover menyebabkan waktu henti. Waktu henti biasanya kurang dari satu menit, tetapi bisa lebih lama tergantung beban kerja Anda.

   Gambar berikut menunjukkan instans DB setelah switchover.



Setelah switchover, instans DB yang berada di lingkungan hijau menjadi instans DB produksi baru. Nama dan titik akhir di lingkungan produksi saat ini ditetapkan ke lingkungan produksi yang baru dipromosikan, sehingga Anda tidak perlu melakukan perubahan pada aplikasi. Akibatnya, lalu

lintas produksi Anda sekarang mengalir ke lingkungan produksi baru. Instans DB di lingkungan biru sebelumnya diganti namanya dengan menambahkan `-old`*n* ke nama saat ini, dengan *n* adalah angka. Misalnya, anggap nama instans DB di lingkungan biru adalah `mydb1`. Setelah switchover, nama instans DB bisa jadi `mydb1-old1`.

Dalam contoh pada gambar, perubahan berikut terjadi selama switchover:

- Deployment instans DB Multi-AZ lingkungan hijau bernama `mydb1-green-abc123` menjadi deployment instans DB Multi-AZ produksi bernama `mydb1`.
- Replika baca lingkungan hijau bernama `mydb2-green-abc123` menjadi replika baca produksi `mydb2`.
- Deployment instans DB Multi-AZ lingkungan biru bernama `mydb1` menjadi `mydb1-old1`.
- Replika baca lingkungan biru bernama `mydb2` menjadi `mydb2-old1`.

6. Jika Anda tidak lagi memerlukan deployment blue/green, Anda dapat menghapusnya. Untuk petunjuknya, lihat Menghapus deployment blue/green.

   Setelah switchover, lingkungan produksi sebelumnya tidak dihapus sehingga Anda dapat menggunakannya untuk pengujian regresi, jika perlu.

Mengizinkan akses ke operasi deployment blue/green

Pengguna harus memiliki izin yang diperlukan untuk melakukan operasi yang terkait dengan deployment blue/green. Anda dapat membuat kebijakan IAM yang memberi izin kepada pengguna dan peran untuk menjalankan operasi API tertentu pada sumber daya yang diperlukan. Anda kemudian dapat melampirkan kebijakan tersebut ke set izin IAM atau peran yang memerlukan izin tersebut. Untuk informasi selengkapnya, lihat Manajemen identitas dan akses untuk Amazon RDS.

Pengguna yang membuat deployment blue/green harus memiliki izin untuk menjalankan operasi RDS berikut:

- `rds:AddTagsToResource`
- `rds:CreateDBInstanceReadReplica`

Pengguna yang melakukan switchover pada deployment blue/green harus memiliki izin untuk menjalankan operasi RDS berikut:

- `rds:ModifyDBInstance`

- `rds:PromoteReadReplica`

Pengguna yang menghapus deployment blue/green harus memiliki izin untuk menjalankan operasi RDS berikut:

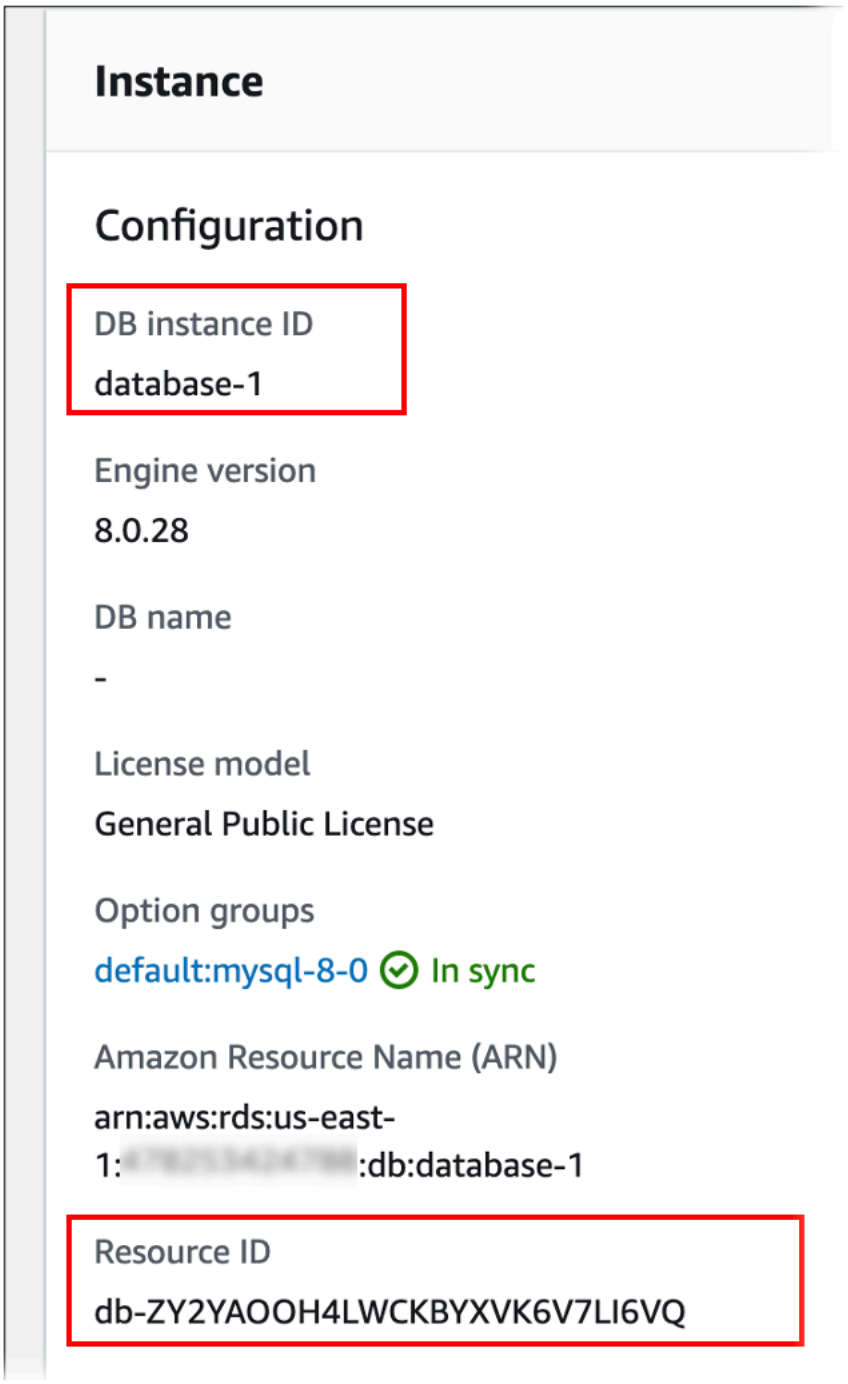
- `rds:DeleteDBInstance`

Amazon RDS menyediakan dan memodifikasi sumber daya di lingkungan pementasan atas nama Anda. Sumber daya ini mencakup instance DB yang menggunakan konvensi penamaan yang ditentukan secara internal. Oleh karena itu, kebijakan IAM terlampir tidak dapat berisi pola nama sumber daya sebagian sepertimy`-db-prefix-*`. Hanya wildcard (*) yang didukung. Secara umum, sebaiknya gunakan tag sumber daya dan atribut lain yang didukung untuk mengontrol akses ke sumber daya ini, bukan wildcard. Untuk informasi selengkapnya, lihat Kunci tindakan, sumber daya, dan kondisi untuk Amazon RDS.

Pertimbangan untuk deployment blue/green

Amazon RDS melacak sumber daya di deployment blue/green dengan `DbiResourceId` dari setiap sumber daya. ID sumber daya ini adalah pengenal Wilayah AWS-unik dan tidak dapat diubah untuk sumber daya.

ID sumber daya terpisah dari ID klaster DB:



Nama (ID instans) sumber daya berubah saat Anda switchover deployment blue/green, tetapi setiap sumber daya menyimpan ID sumber daya yang sama. Misalnya, pengidentifikasi instans DB mungkin adalah `mydb` di lingkungan biru. Setelah switchover, instans DB yang sama mungkin diganti namanya menjadi `mydb-old1`. Namun, ID sumber daya instans DB tidak berubah selama switchover. Jadi, ketika sumber daya hijau dipromosikan menjadi sumber daya produksi baru, ID sumber dayanya tidak cocok dengan ID sumber daya biru yang sebelumnya diproduksi.

Setelah switchover deployment blue/green, sebaiknya perbarui ID sumber daya ke sumber daya produksi yang baru dipromosikan untuk fitur dan layanan terintegrasi yang Anda gunakan dengan sumber daya produksi. Secara khusus, pertimbangkan pembaruan berikut:

- Jika Anda melakukan pemfilteran menggunakan API RDS dan ID sumber daya, sesuaikan ID sumber daya yang digunakan dalam pemfilteran setelah switchover.
- Jika Anda menggunakan CloudTrail untuk mengaudit sumber daya, sesuaikan konsumen CloudTrail untuk melacak ID sumber daya baru setelah peralihan. Untuk informasi selengkapnya, lihat Memantau panggilan API Amazon RDS di AWS CloudTrail.
- Jika Anda menggunakan API Wawasan Performa, sesuaikan ID sumber daya dalam panggilan ke API setelah switchover. Untuk informasi selengkapnya, lihat Memantau muatan DB dengan Wawasan Performa di Amazon RDS.

  Anda dapat memantau basis data dengan nama yang sama setelah switchover, tetapi basis data tersebut tidak berisi data sebelum switchover.
- Jika menggunakan ID sumber daya dalam kebijakan IAM, pastikan Anda menambahkan ID sumber daya dari sumber daya yang baru dipromosikan jika diperlukan. Untuk informasi selengkapnya, lihat Manajemen identitas dan akses untuk Amazon RDS.
- Jika Anda memiliki peran IAM yang terkait dengan instans Anda, pastikan untuk mengasosiasikannya kembali setelah peralihan. Peran terlampir tidak secara otomatis disalin ke lingkungan hijau.
- Jika Anda mengautentikasi instans DB menggunakan autentikasi basis data IAM, pastikan kebijakan IAM yang digunakan untuk akses basis data memiliki basis data biru dan hijau yang tercantum di elemen `Resource` kebijakan. Ini diperlukan agar dapat terhubung ke basis data hijau setelah switchover. Untuk informasi selengkapnya, lihat the section called “Membuat dan menggunakan kebijakan IAM untuk akses basis data IAM”.
- Jika Anda menggunakannya AWS Backup untuk mengelola pencadangan otomatis sumber daya dalam penerapan biru/hijau, sesuaikan ID sumber daya yang digunakan setelah peralihan. AWS Backup Untuk informasi selengkapnya, lihat Menggunakan AWS Backup untuk mengelola backup otomatis.
- Jika Anda ingin memulihkan snapshot DB manual atau otomatis untuk instans DB yang merupakan bagian dari deployment blue/green, pastikan Anda memulihkan snapshot DB yang benar dengan memeriksa waktu ketika snapshot diambil. Untuk informasi selengkapnya, lihat Memulihkan dari snapshot DB.
- Jika Anda ingin menjelaskan pencadangan otomatis instans DB lingkungan biru sebelumnya atau memulihkannya ke waktu tertentu, gunakan ID sumber daya untuk operasi tersebut.

Karena nama instans DB berubah selama switchover, Anda tidak dapat menggunakan nama sebelumnya untuk operasi `DescribeDBInstanceAutomatedBackups` atau `RestoreDBInstanceToPointInTime`.

Untuk informasi selengkapnya, lihat Memulihkan instans DB dengan waktu yang ditentukan.

- Saat Anda menambahkan replika baca ke instans DB di lingkungan hijau dari deployment blue/green, replika baca baru tidak akan menggantikan replika baca di lingkungan biru saat Anda switch over. Namun, replika baca baru dipertahankan di lingkungan produksi baru setelah switchover.
- Jika Anda menghapus instans DB di lingkungan hijau deployment blue/green, Anda tidak dapat membuat instans DB baru untuk menggantikannya dalam deployment blue/green.

  Jika Anda membuat instans DB baru dengan nama yang sama dan Amazon Resource Name (ARN) dengan instans DB yang dihapus, instans tersebut memiliki `DbiResourceId` yang berbeda, jadi instans tersebut bukan bagian dari lingkungan hijau.

  Perilaku berikut akan terjadi jika Anda menghapus instans DB di lingkungan hijau:

  - Jika ada instans DB di lingkungan biru dengan nama yang sama, instans tersebut tidak akan switchover ke instans DB di lingkungan hijau. Instans DB ini tidak akan diganti namanya dengan menambahkan `-old`*n* ke nama instans DB tersebut.
  - Aplikasi apa pun yang menunjuk ke instans DB di lingkungan biru terus menggunakan instans DB yang sama setelah switchover.

  Perilaku yang sama berlaku untuk instans DB dan replika baca.

Praktik terbaik untuk deployment blue/green

Berikut ini adalah praktik terbaik untuk deployment blue/green:

Praktik terbaik umum

- Uji instans DB secara menyeluruh di lingkungan hijau sebelum switchover.
- Simpan basis data Anda di lingkungan hijau dengan kondisi hanya baca. Sebaiknya Anda mengaktifkan operasi tulis di lingkungan hijau dengan hati-hati karena dapat mengakibatkan konflik replikasi. Hal ini juga dapat menghasilkan data yang tidak diinginkan dalam basis data produksi setelah switchover.

- Saat menggunakan deployment blue/green untuk mengimplementasikan perubahan skema, hanya buat perubahan yang kompatibel dengan replikasi.

  Misalnya, Anda dapat menambahkan kolom baru di akhir tabel, membuat indeks, atau menghapus indeks tanpa mengganggu replikasi dari deployment biru ke deployment hijau. Namun, perubahan skema, seperti penggantian nama kolom atau nama tabel, memecah replikasi ke deployment hijau.

  Untuk informasi selengkapnya tentang perubahan yang kompatibel dengan replikasi, lihat Replication with Differing Table Definitions on Source and Replica di dokumentasi MySQL dan Restrictions dalam dokumentasi replikasi logis PostgreSQL.
- Setelah Anda membuat deployment blue/green, tangani pemuatan lambat jika perlu. Pastikan pemuatan data selesai sebelum switchover. Untuk informasi selengkapnya, lihat Menangani pemuatan lambat saat Anda membuat deployment blue/green.
- Saat Anda mengalihkan deployment blue/green, ikuti praktik terbaik switchover. Untuk informasi selengkapnya, lihat the section called “Praktik terbaik switchover”.

Praktik terbaik RDS for MySQL

- Hindari menggunakan mesin penyimpanan non-transaksional, seperti MyISAM, yang tidak dioptimalkan untuk replikasi.
- Optimalkan replika baca untuk replikasi log biner.

  Misalnya, jika versi mesin DB Anda mendukungnya, sebaiknya gunakan replikasi GTID, replikasi paralel, dan replikasi aman dari crash di lingkungan produksi Anda sebelum men-deploy deployment blue/green. Opsi ini mendukung konsistensi dan daya tahan data Anda sebelum switchover deployment blue/green. Untuk informasi selengkapnya tentang replikasi GTID untuk replika baca, lihat Menggunakan replikasi berbasis GTID untuk Amazon RDS for MySQL.

Praktik terbaik RDS for PostgreSQL

- Jika database Anda memiliki memori freeable yang cukup, tingkatkan nilai parameter `logical_decoding_work_mem` DB di lingkungan biru. Tindakan ini memungkinkan lebih sedikit decoding pada disk, alih-alih menggunakan memori. Anda dapat memantau memori yang dapat dibebaskan dengan `FreeableMemory` CloudWatch metrik. Untuk informasi selengkapnya, lihat the section called “Metrik CloudWatch tingkat instans Amazon untuk Amazon RDS”.

- Perbarui semua ekstensi PostgreSQL Anda ke versi terbaru sebelum membuat deployment blue/green. Untuk informasi selengkapnya, lihat the section called “Meningkatkan ekstensi PostgreSQL”.
- Jika Anda menggunakan ekstensi `aws_s3`, pastikan Anda memberikan akses instans DB hijau ke Amazon S3 melalui peran IAM setelah lingkungan hijau dibuat. Hal ini memungkinkan perintah impor dan ekspor untuk terus berfungsi setelah switchover. Untuk petunjuknya, lihat the section called “Menyiapkan akses ke bucket Amazon S3”.

Ketersediaan wilayah dan versi

Ketersediaan fitur dan dukungan bervariasi di seluruh versi spesifik dari setiap mesin basis data, dan di seluruh bagian Wilayah AWS. Untuk informasi selengkapnya tentang versi dan ketersediaan Wilayah dengan Deployment Blue/Green Amazon RDS, lihat Deployment Blue/Green.

Batasan untuk deployment blue/green

Batasan berikut berlaku untuk deployment blue/green.

Topik

- Batasan umum untuk deployment blue/green
- Batasan ekstensi PostgreSQL untuk deployment blue/green
- Batasan untuk perubahan dalam deployment blue/green
- Batasan replikasi logis PostgreSQL untuk deployment blue/green

Batasan umum untuk deployment blue/green

Batasan umum berikut berlaku untuk deployment blue/green:

- MySQL versi 8.0.11 hingga 8.0.13 memiliki bug komunitas yang mencegahnya mendukung deployment blue/green.
- Versi berikut dari RDS for PostgreSQL didukung sebagai versi target dan sumber peningkatan: 11.21 dan yang lebih tinggi, 12.16 dan yang lebih tinggi, 13.12 dan yang lebih tinggi, 14.9 dan yang lebih tinggi, serta 15.4 dan yang lebih tinggi. Untuk versi yang lebih rendah, Anda dapat melakukan peningkatan versi minor ke versi yang didukung.
- Penerapan biru/hijau tidak mendukung pengelolaan kata sandi pengguna utama dengan. AWS Secrets Manager

- Untuk RDS for PostgreSQL, tabel yang tidak tercatat tidak direplikasi ke lingkungan hijau.
- Untuk , instans DB lingkungan biru tidak dapat berupa sumber logis (penerbit) atau replika (pelanggan) yang dikelola sendiri.
- Penjadwal Peristiwa (parameter `event_scheduler`) harus dinonaktifkan di lingkungan hijau saat Anda membuat deployment blue/green. Ini mencegah peristiwa dihasilkan di lingkungan hijau dan menyebabkan inkonsistensi.
- Penerapan biru/hijau tidak mendukung Driver AWS JDBC untuk MySQL. Untuk informasi selengkapnya, lihat Batasan yang Diketahui pada GitHub.
- Deployment blue/green tidak didukung untuk fitur berikut:
  - Proksi Amazon RDS
  - Replika baca kaskade
  - Replika baca Lintas Wilayah
  - AWS CloudFormation
  - Deployment klaster DB Multi-AZ

    Deployment blue/green didukung untuk deployment instans DB Multi-AZ. Untuk informasi selengkapnya tentang deployment Multi-AZ, lihat Mengonfigurasi dan mengelola deployment Multi-AZ.

Batasan ekstensi PostgreSQL untuk deployment blue/green

Batasan berikut berlaku untuk ekstensi PostgreSQL:

- Ekstensi `pg_partman` harus dinonaktifkan di lingkungan biru saat Anda membuat deployment blue/green. Ekstensi tersebut menjalankan operasi DDL seperti `CREATE TABLE`, yang memecah replikasi logis dari lingkungan biru ke lingkungan hijau.
- Ekstensi `pg_cron` harus tetap dinonaktifkan di semua basis data hijau setelah deployment blue/green dibuat. Ekstensi tersebut memiliki pekerja latar belakang yang berjalan sebagai superuser dan melewati pengaturan hanya baca di lingkungan hijau, yang dapat menyebabkan konflik replikasi.
- Jika instans DB biru dikonfigurasi sebagai server asing dari ekstensi pembungkus data asing (FDW), Anda harus menggunakan nama titik akhir instans, alih-alih alamat IP. Hal ini memungkinkan konfigurasi untuk tetap berfungsi setelah switchover.
- Ekstensi `pg_active` dan `pglogical` harus dinonaktifkan di lingkungan biru saat Anda membuat deployment blue/green. Setelah Anda mempromosikan lingkungan hijau menjadi lingkungan

produksi baru, Anda dapat mengaktifkan ekstensi lagi. Selain itu, basis data biru tidak bisa menjadi pelanggan logis dari instans eksternal.

- Jika Anda menggunakan `pgAudit` ekstensi, ekstensi harus tetap berada di pustaka bersama (`shared_preload_libraries`) pada grup parameter DB khusus untuk instance DB biru dan hijau. Untuk informasi selengkapnya, lihat the section called “Menyiapkan ekstensi pgAdit”.

Batasan untuk perubahan dalam deployment blue/green

Berikut ini adalah batasan untuk perubahan dalam deployment blue/green:

- Anda tidak dapat mengubah instans DB yang tidak terenkripsi menjadi instans DB yang terenkripsi.
- Anda tidak dapat mengubah instans DB yang terenkripsi menjadi instans DB yang tidak terenkripsi.
- Anda tidak dapat mengubah instans DB lingkungan biru ke versi mesin yang lebih tinggi daripada instans DB lingkungan hijau yang sesuai.
- Sumber daya di lingkungan biru dan lingkungan hijau harus berada dalam Akun AWS yang sama.
- Untuk RDS for MySQL, jika basis data sumber dikaitkan dengan grup opsi kustom, Anda tidak dapat menentukan peningkatan versi mayor saat Anda membuat deployment blue/green.

  Dalam hal ini, Anda dapat membuat deployment blue/green tanpa menentukan peningkatan versi mayor. Kemudian, Anda dapat meningkatkan basis data di lingkungan hijau. Untuk informasi selengkapnya, lihat Meng-upgrade versi mesin instans DB.

Batasan replikasi logis PostgreSQL untuk deployment blue/green

Deployment blue/green menggunakan replikasi logis agar lingkungan pementasan tetap sinkron dengan lingkungan produksi. PostgreSQL memiliki batasan tertentu yang terkait dengan replikasi logis, yang diterjemahkan ke batasan saat membuat deployment blue/green untuk instans DB RDS for PostgreSQL.

Tabel berikut menjelaskan batasan replikasi logis yang berlaku untuk deployment blue/green RDS for PostgreSQL.

Batasan	Penjelasan
Pernyataan bahasa	Jika Amazon RDS mendeteksi perubahan DDL di lingkungan biru, basis data hijau Anda memasukkan status Replikasi terdegradasi.

Batasan	Penjelasan
definisi data (DDL), seperti `CREATE TABLE` dan `CREATE SCHEMA`, tidak direplikasi dari lingkungan biru ke lingkungan hijau.	Anda menerima peristiwa yang memberi tahu bahwa perubahan DDL di lingkungan biru tidak dapat direplikasi ke lingkungan hijau. Anda harus menghapus deployment blue/green dan semua basis data hijau, lalu buat kembali semuanya. Jika tidak, Anda tidak dapat switchover deployment blue/green.
Operasi `NEXTVAL` pada objek urutan tidak disinkronkan antara lingkungan biru dan lingkungan hijau.	Selama switchover, Amazon RDS menambah nilai urutan di lingkungan hijau agar sesuai dengan yang ada di lingkungan biru. Jika Anda memiliki ribuan urutan, hal ini dapat menunda switchover.
Pembuatan atau perubahan objek besar di lingkungan biru tidak direplikasi ke lingkungan hijau.	Jika Amazon RDS mendeteksi pembuatan atau perubahan objek besar di lingkungan biru yang disimpan dalam tabel sistem `pg_largeobject` , basis data hijau Anda memasukkan status Replikasi terdegradasi. RDS menghasilkan peristiwa yang memberi tahu Anda bahwa perubahan objek besar di lingkungan biru tidak dapat direplikasi ke lingkungan hijau. Anda harus menghapus deployment blue/green dan semua basis data hijau, lalu buat kembali semuanya. Jika tidak, Anda tidak dapat switchover deployment blue/green.
Tampilan terwujud tidak secara otomatis disegarkan di lingkungan hijau.	Menyegarkan tampilan terwujud di lingkungan biru tidak akan menyegarkannya di lingkungan hijau. Setelah switchover, Anda dapat menjadwalkan penyegaran tampilan terwujud.

Batasan	Penjelasan
Operasi UPDATE dan DELETE tidak diizinkan pada tabel yang tidak memiliki kunci primer.	Sebelum Anda membuat deployment blue/green, pastikan semua tabel di instans DB memiliki kunci primer.

Untuk informasi selengkapnya, lihat Restrictions di dokumentasi replikasi logis PostgreSQL.

Membuat deployment blue/green

Saat Anda membuat deployment blue/green, Anda menentukan instans DB sumber yang akan disalin dalam deployment. Instans DB yang Anda pilih adalah instans DB produksi, dan menjadi instans DB primer di lingkungan biru. Instans DB ini disalin ke lingkungan hijau, dan RDS mengonfigurasi replikasi dari instans DB di lingkungan biru ke instans DB di lingkungan hijau.

RDS menyalin topologi lingkungan biru ke area pementasan, beserta fitur yang dikonfigurasinya. Setelah instans DB biru memiliki replika baca, replika baca disalin sebagai replika baca instans DB hijau dalam deployment. Jika instans DB biru adalah deployment instans DB Multi-AZ, berarti instans DB hijau dibuat sebagai deployment instans DB Multi-AZ.

Topik

- Mempersiapkan deployment blue/green
- Menentukan perubahan saat membuat deployment blue/green
- Menangani pemuatan lambat saat Anda membuat deployment blue/green
- Membuat deployment blue/green

Mempersiapkan deployment blue/green

Ada langkah-langkah tertentu yang harus Anda ambil sebelum Anda membuat penerapan biru/hijau, tergantung pada mesin yang menjalankan instans Anda.

Topik

- Mempersiapkan RDS untuk instance MySQL DB untuk penerapan biru/hijau
- Mempersiapkan instans DB RDS for PostgreSQL untuk deployment blue/green

Mempersiapkan RDS untuk instance MySQL DB untuk penerapan biru/hijau

Sebelum Anda membuat penerapan biru/hijau untuk RDS untuk instance MySQL DB, Anda harus mengaktifkan pencadangan otomatis. Untuk petunjuk, lihat the section called “Mengaktifkan pencadangan otomatis”.

Mempersiapkan instans DB RDS for PostgreSQL untuk deployment blue/green

Sebelum membuat deployment blue/green untuk instans DB RDS for PostgreSQL, pastikan untuk melakukan hal berikut:

- Kaitkan instans dengan grup parameter DB kustom, dengan replikasi logis (`rds.logical_replication`) diaktifkan. Replikasi logika diperlukan untuk replikasi dari lingkungan biru ke lingkungan hijau. Untuk petunjuknya, lihat the section called “Memodifikasi parameter dalam grup parameter DB”.

  Karena penerapan biru/hijau memerlukan setidaknya satu pekerja latar belakang per database, pastikan untuk menyetel pengaturan konfigurasi berikut sesuai dengan beban kerja Anda. Untuk petunjuk untuk menyetel setiap pengaturan, lihat Pengaturan Konfigurasi dalam dokumentasi PostgreSQL.

  - `max_replication_slots`
  - `max_wal_senders`
  - `max_logical_replication_workers`
  - `max_worker_processes`

  Setelah Anda mengaktifkan replikasi logis dan mengatur semua opsi konfigurasi, pastikan untuk mem-boot ulang instans DB agar perubahan Anda diterapkan. Deployment blue/green mengharuskan instans DB disinkronkan dengan grup parameter DB, jika tidak, pembuatan akan gagal. Untuk informasi selengkapnya, lihat the section called “Mem-boot ulang instans DB”.
- Pastikan instans DB Anda menjalankan versi RDS for PostgreSQL yang kompatibel dengan Deployment Blue/Green RDS. Untuk daftar versi yang kompatibel, lihat the section called “Deployment Blue/Green”.
- Konfirmasikan bahwa instans DB bukan sumber atau target replikasi eksternal. Untuk informasi selengkapnya, lihat the section called “Batasan umum”.

- Pastikan bahwa semua tabel dalam instans DB memiliki kunci primer. Replikasi logis PostgreSQL tidak mengizinkan operasi `UPDATE` atau `DELETE` pada tabel yang tidak memiliki kunci primer.

Menentukan perubahan saat membuat deployment blue/green

Anda dapat membuat perubahan berikut pada instans DB di lingkungan hijau saat membuat deployment blue/green.

Anda dapat membuat penyesuaian pada instans DB di lingkungan hijau setelah di-deploy. Misalnya, Anda dapat membuat perubahan skema pada basis data Anda atau mengubah kelas instans DB yang digunakan oleh satu atau beberapa instans DB di lingkungan hijau.

Untuk informasi tentang memodifikasi instans DB, lihat Memodifikasi instans DB Amazon RDS.

Menentukan versi mesin yang lebih tinggi

Anda dapat menentukan versi mesin yang lebih tinggi jika ingin menguji peningkatan mesin DB. Setelah switchover, basis data ditingkatkan ke versi mesin DB mayor atau minor yang Anda tentukan.

Menentukan grup parameter DB yang berbeda

Anda dapat menguji bagaimana perubahan parameter memengaruhi instans DB di lingkungan hijau atau menentukan grup parameter untuk versi mesin DB mayor baru jika terjadi peningkatan.

Jika Anda menentukan grup parameter DB yang berbeda, grup parameter DB yang ditentukan dikaitkan dengan semua instans DB di lingkungan hijau. Jika Anda tidak menentukan grup parameter yang berbeda, setiap instans DB di lingkungan hijau dikaitkan dengan grup parameter dari instans DB biru yang sesuai.

Mengaktifkan RDS Optimized Writes

Anda dapat menggunakan Deployment Blue/Green untuk meningkatkan ke kelas instans DB yang mendukung RDS Optimized Writes. Anda hanya dapat mengaktifkan RDS Optimized Writes pada basis data yang dibuat dengan kelas instans DB yang didukung. Oleh karena itu, opsi ini membuat basis data hijau yang menggunakan kelas instans DB yang didukung, yang memungkinkan Anda mengaktifkan RDS Optimized Writes pada instans DB hijau.

Jika Anda meningkatkan dari kelas instans DB yang tidak mendukung RDS Optimized Writes ke kelas yang mendukung, Anda juga harus meningkatkan konfigurasi penyimpanan instans DB hijau. Untuk informasi selengkapnya, lihat the section called “Meningkatkan konfigurasi penyimpanan”.

Anda hanya dapat meningkatkan kelas instans DB dari instans DB hijau primer. Secara default, replika baca di lingkungan hijau mewarisi pengaturan instans DB dari lingkungan biru. Setelah lingkungan hijau berhasil dibuat, Anda harus menyesuaikan kelas instans DB replika baca secara manual di lingkungan hijau.

Tergantung versi mesin dan kelas instans dari instans DB biru, beberapa peningkatan kelas instans tidak didukung. Untuk informasi selengkapnya tentang kelas instans DB, lihat the section called “Kelas instans DB”.

Meningkatkan konfigurasi penyimpanan

Jika basis data biru Anda tidak menggunakan konfigurasi penyimpanan terbaru, RDS dapat memigrasikan instans DB hijau dari konfigurasi penyimpanan yang lebih lama (sistem file 32-bit) ke konfigurasi yang diinginkan. Anda dapat menggunakan Deployment Blue/Green RDS untuk mengatasi batasan penskalaan pada penyimpanan dan ukuran file untuk sistem file 32-bit yang lebih lama. Selain itu, pengaturan ini mengubah konfigurasi penyimpanan agar kompatibel dengan RDS Optimized Writes jika kelas instans DB yang ditentukan mendukung Optimized Writes.

> **Note**
>
> Meningkatkan konfigurasi penyimpanan adalah operasi intensif I/O dan menyebabkan waktu pembuatan yang lebih lama untuk deployment blue/green. Proses peningkatan penyimpanan akan lebih cepat jika instans DB biru menggunakan penyimpanan SSD IOPS yang tersedia (io1), dan jika Anda menyediakan lingkungan hijau dengan ukuran instans 4xlarge atau lebih besar. Peningkatan penyimpanan yang melibatkan penyimpanan SSD Tujuan Umum (gp2) dapat mengurangi saldo kredit I/O, sehingga menyebabkan waktu peningkatan yang lebih lama. Untuk informasi selengkapnya, lihat the section called “Penyimpanan instans DB”. Selama proses peningkatan penyimpanan, mesin basis data tidak tersedia. Jika penggunaan penyimpanan pada instans DB biru Anda lebih besar dari atau sama dengan 90% dari ukuran penyimpanan yang dialokasikan, proses peningkatan penyimpanan akan meningkatkan ukuran penyimpanan yang dialokasikan sebesar 10% untuk instans hijau.

Opsi ini hanya tersedia jika basis data biru Anda tidak menggunakan konfigurasi penyimpanan terbaru, atau jika Anda mengubah kelas instans DB dalam permintaan yang sama.

Menangani pemuatan lambat saat Anda membuat deployment blue/green

Saat Anda membuat deployment blue/green, Amazon RDS membuat instans DB primer di lingkungan hijau dengan memulihkan dari snapshot DB. Setelah dibuat, instans DB hijau terus memuat data di latar belakang, yang dikenal sebagai pemuatan lambat. Jika instans DB memiliki replika baca, replika tersebut juga dibuat dari snapshot DB dan mengalami pemuatan lambat.

Jika Anda mengakses data yang belum dimuat, instans DB segera mengunduh data yang diminta dari Amazon S3, lalu lanjut memuat sisa data di latar belakang. Untuk informasi selengkapnya, lihat Snapshot Amazon EBS.

Untuk membantu mengurangi efek dari pemuatan lambat pada tabel yang harus diakses dengan cepat, Anda dapat melakukan operasi yang melibatkan pemindaian tabel lengkap, seperti `SELECT *`. Operasi ini memungkinkan Amazon RDS untuk mengunduh semua data tabel yang dicadangkan dari S3.

Jika aplikasi mencoba mengakses data yang tidak dimuat, aplikasi dapat mengalami latensi yang lebih tinggi dari biasanya saat data dimuat. Latensi yang lebih tinggi karena pemuatan lambat ini dapat menyebabkan performa yang buruk untuk beban kerja yang sensitif terhadap latensi.

> **Important**
>
> Jika Anda melakukan switchover deployment blue/green sebelum pemuatan data selesai, aplikasi Anda dapat mengalami masalah performa karena latensi tinggi.

Membuat deployment blue/green

Anda dapat membuat penerapan biru/hijau menggunakan AWS Management Console, API AWS CLI, atau RDS.

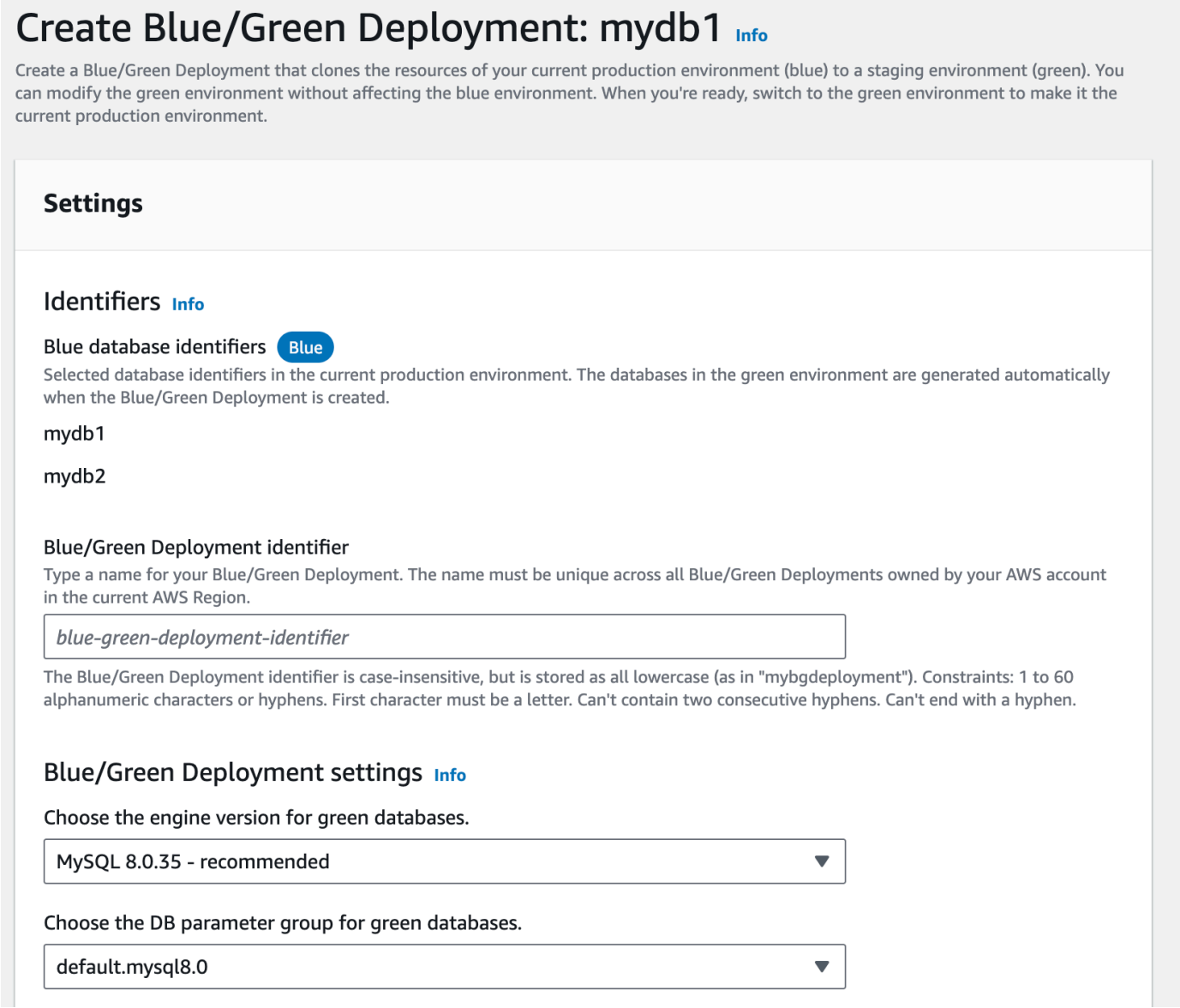
Konsol

Untuk membuat deployment blue/green

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data, lalu pilih instans DB yang ingin disalin ke lingkungan hijau.
3. Pilih Tindakan, Buat Penerapan Biru/Hijau.

Jika Anda memilih instans DB RDS for PostgreSQL, tinjau dan konfirmasi batasan replikasi logisnya. Untuk informasi selengkapnya, lihat the section called “Batasan replikasi logis PostgreSQL”.

Halaman Buat Deployment Blue/Green muncul.



4. Tinjau pengidentifikasi database biru. Pastikan bahwa mereka cocok dengan instans DB yang Anda harapkan di lingkungan biru. Jika tidak, pilih Batalkan.
5. Untuk pengidentifikasi Deployment Blue/Green, masukkan nama untuk deployment blue/green Anda.
6. (Opsional) Untuk pengaturan Deployment Blue/Green, tentukan pengaturan untuk lingkungan hijau:

- Pilih versi mesin DB jika Anda ingin menguji peningkatan versi mesin DB.
- Pilih grup parameter DB yang akan dikaitkan dengan instans DB di lingkungan hijau.

Anda dapat membuat penyesuaian lain pada basis data di lingkungan hijau setelah di-deploy.

7. (Opsional) Untuk RDS Optimized Writes, aktifkan RDS Optimized Writes dengan meningkatkan kelas instans DB dari instans DB hijau primer. Untuk informasi selengkapnya, lihat the section called “Mengaktifkan RDS Optimized Writes”.

   Jika Anda mengubah dari kelas instans DB yang tidak mendukung Optimized Writes ke kelas yang mendukung, Anda juga harus melakukan peningkatan konfigurasi penyimpanan. Lihat langkah berikutnya untuk detailnya.

8. (Opsional) Untuk Peningkatan konfigurasi penyimpanan, pilih apakah akan meningkatkan konfigurasi sistem file penyimpanan Anda. Jika Anda mengaktifkan opsi ini, RDS memigrasikan instans DB hijau dari sistem file penyimpanan lama ke konfigurasi yang diinginkan. Untuk informasi selengkapnya, lihat the section called “Meningkatkan sistem file penyimpanan”.

   Opsi ini hanya tersedia jika basis data biru Anda tidak pada konfigurasi penyimpanan terbaru, atau jika Anda mengaktifkan RDS Optimized Writes dalam permintaan yang sama.

9. Pilih Buat lingkungan pementasan.

AWS CLI

Untuk membuat penyebaran biru/hijau menggunakan AWS CLI, gunakan create-blue-green-deploymentperintah dengan opsi berikut:

- `--blue-green-deployment-name` – Tentukan nama deployment blue/green.
- `--source` – Tentukan ARN dari instans DB yang ingin Anda salin.
- `--target-engine-version` – Tentukan versi mesin jika Anda ingin menguji peningkatan versi mesin DB di lingkungan hijau. Opsi ini meningkatkan instans DB di lingkungan hijau ke versi mesin DB yang ditentukan.

  Jika tidak ditentukan, setiap instans DB di lingkungan hijau dibuat dengan versi mesin yang sama dengan instans DB yang sesuai di lingkungan biru.
- `--target-db-parameter-group-name` – Tentukan grup parameter DB yang akan dikaitkan dengan instans DB di lingkungan hijau.

- `--target-db-instance-class` – Tentukan kelas instans DB yang mendukung RDS Optimized Writes. Opsi ini memungkinkan RDS Optimized Writes pada instans DB primer hijau. Untuk informasi selengkapnya, lihat the section called “Mengaktifkan RDS Optimized Writes”.
- `--upgrade-target-storage-config` – Tentukan apakah akan meningkatkan konfigurasi sistem file penyimpanan pada basis data hijau. Anda hanya dapat mengaktifkan opsi ini jika nilai opsi `is-storage-config-upgrade-available` adalah `true` untuk instans DB, atau jika Anda memodifikasi nilai opsi `target-db-instance-class` dalam permintaan yang sama. Untuk informasi selengkapnya, lihat the section called “Meningkatkan sistem file penyimpanan”.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds create-blue-green-deployment \
    --blue-green-deployment-name my-blue-green-deployment \
    --source arn:aws:rds:us-east-2:123456789012:db:mydb1 \
    --target-engine-version 8.0.31 \
    --target-db-parameter-group-name mydbparametergroup \
    --target-db-instance-class db.m5.8xlarge \
    --upgrade-target-storage-config
```

Untuk Windows:

```
aws rds create-blue-green-deployment ^
    --blue-green-deployment-name my-blue-green-deployment ^
    --source arn:aws:rds:us-east-2:123456789012:db:mydb1 ^
    --target-engine-version 8.0.31 ^
    --target-db-parameter-group-name mydbparametergroup ^
    --target-db-instance-class db.m5.8xlarge ^
    --upgrade-target-storage-config
```

API RDS

Untuk membuat deployment blue/green menggunakan API Amazon RDS, gunakan operasi `CreateBlueGreenDeployment` dengan parameter berikut:

- `BlueGreenDeploymentName` – Tentukan nama deployment blue/green.
- `Source` – Tentukan ARN dari instans DB yang ingin Anda salin ke lingkungan hijau.

- `TargetEngineVersion` – Tentukan versi mesin jika Anda ingin menguji peningkatan versi mesin DB di lingkungan hijau. Opsi ini meningkatkan instans DB di lingkungan hijau ke versi mesin DB yang ditentukan.

  Jika tidak ditentukan, setiap instans DB di lingkungan hijau dibuat dengan versi mesin yang sama dengan instans DB yang sesuai di lingkungan biru.
- `TargetDBParameterGroupName` – Tentukan grup parameter DB yang akan dikaitkan dengan instans DB di lingkungan hijau.
- `TargetDBInstanceClass` – Tentukan kelas instans DB yang mendukung RDS Optimized Writes. Opsi ini memungkinkan RDS Optimized Writes pada instans DB primer hijau. Untuk informasi selengkapnya, lihat the section called “Mengaktifkan RDS Optimized Writes”.
- `UpgradeTargetStorageConfig` – Tentukan apakah akan meningkatkan konfigurasi sistem file penyimpanan pada basis data hijau. Anda hanya dapat mengaktifkan opsi ini jika nilai opsi `is-storage-config-upgrade-available` adalah `true` untuk instans DB, atau jika Anda memodifikasi nilai opsi `target-db-instance-class` dalam permintaan yang sama. Untuk informasi selengkapnya, lihat the section called “Meningkatkan sistem file penyimpanan”.

Melihat deployment blue/green

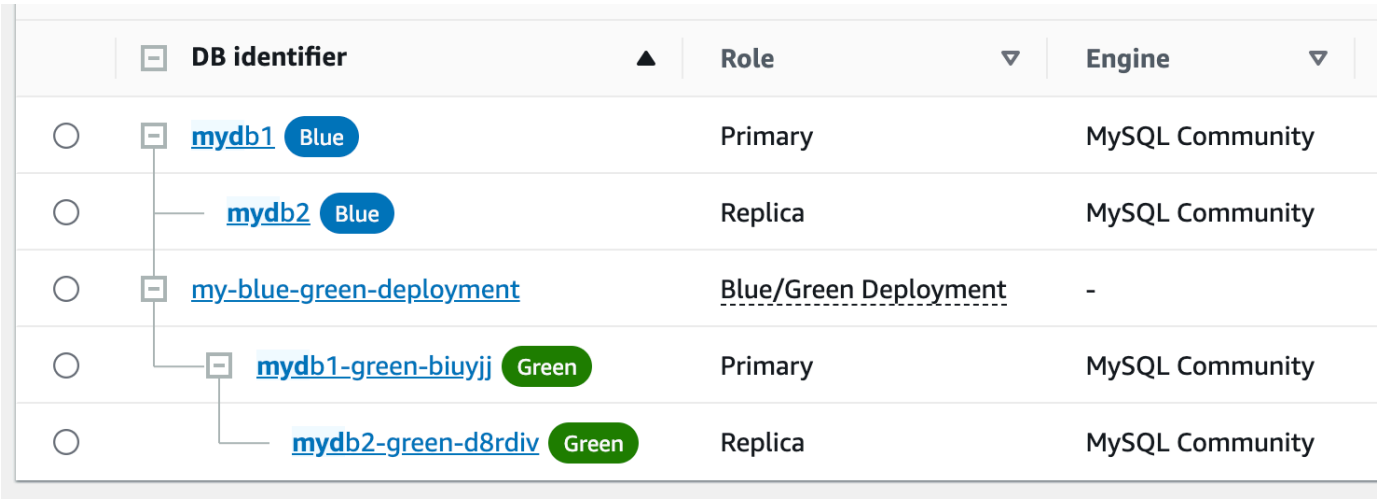
Anda dapat melihat detail tentang deployment blue/green menggunakan AWS Management Console, AWS CLI, atau API RDS.

Anda juga dapat melihat dan berlangganan peristiwa untuk mendapatkan informasi tentang deployment blue/green. Untuk informasi selengkapnya, lihat Peristiwa deployment blue/green.

Konsol

Untuk melihat detail tentang deployment blue/green

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data, lalu temukan deployment blue/green di dalam daftar.

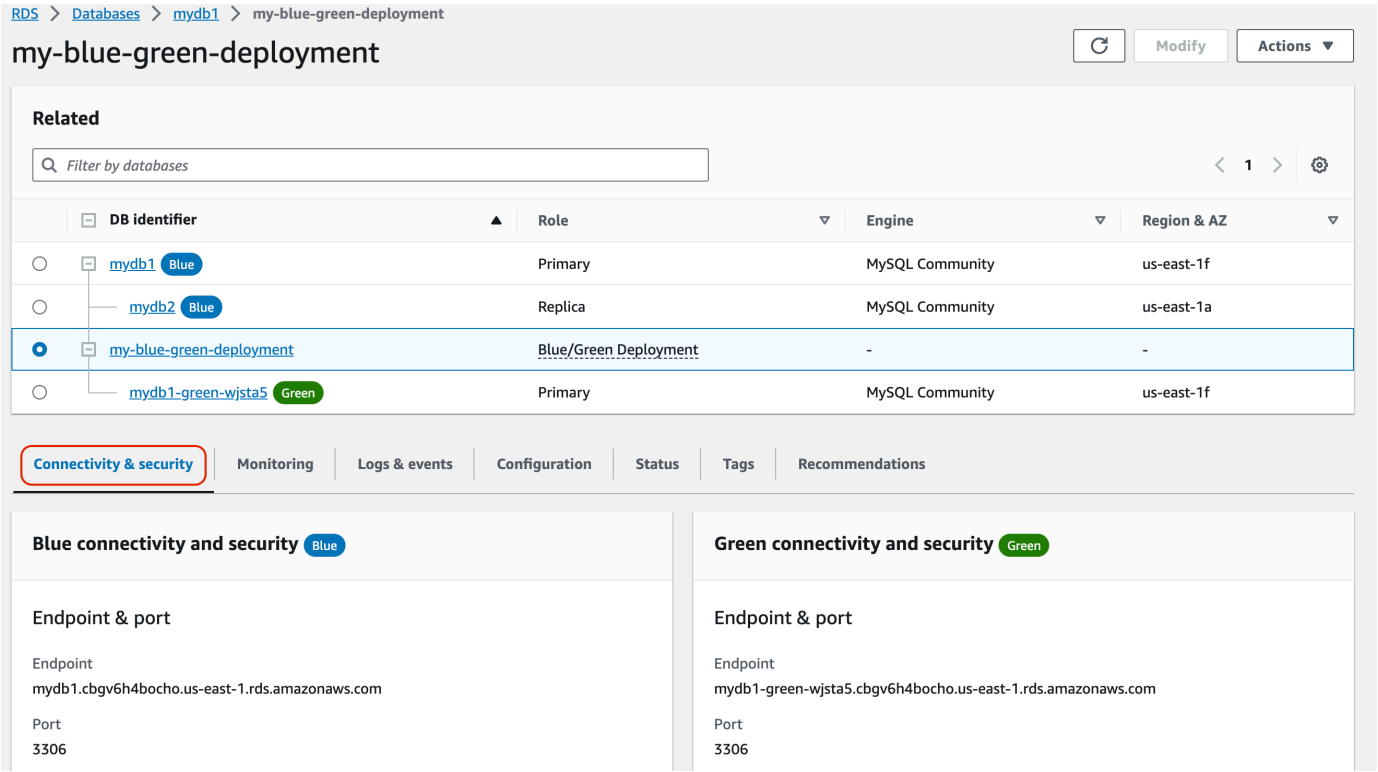


Nilai Peran untuk Deployment adalah Deployment Blue/Green.

3. Pilih nama deployment blue/green hijau yang ingin Anda lihat untuk menampilkan detailnya.

Setiap tab memiliki bagian untuk deployment biru dan bagian untuk deployment hijau. Misalnya, pada tab Konfigurasi, versi mesin DB mungkin berbeda di lingkungan biru dan di lingkungan hijau jika Anda memutakhirkan versi mesin DB di lingkungan hijau.

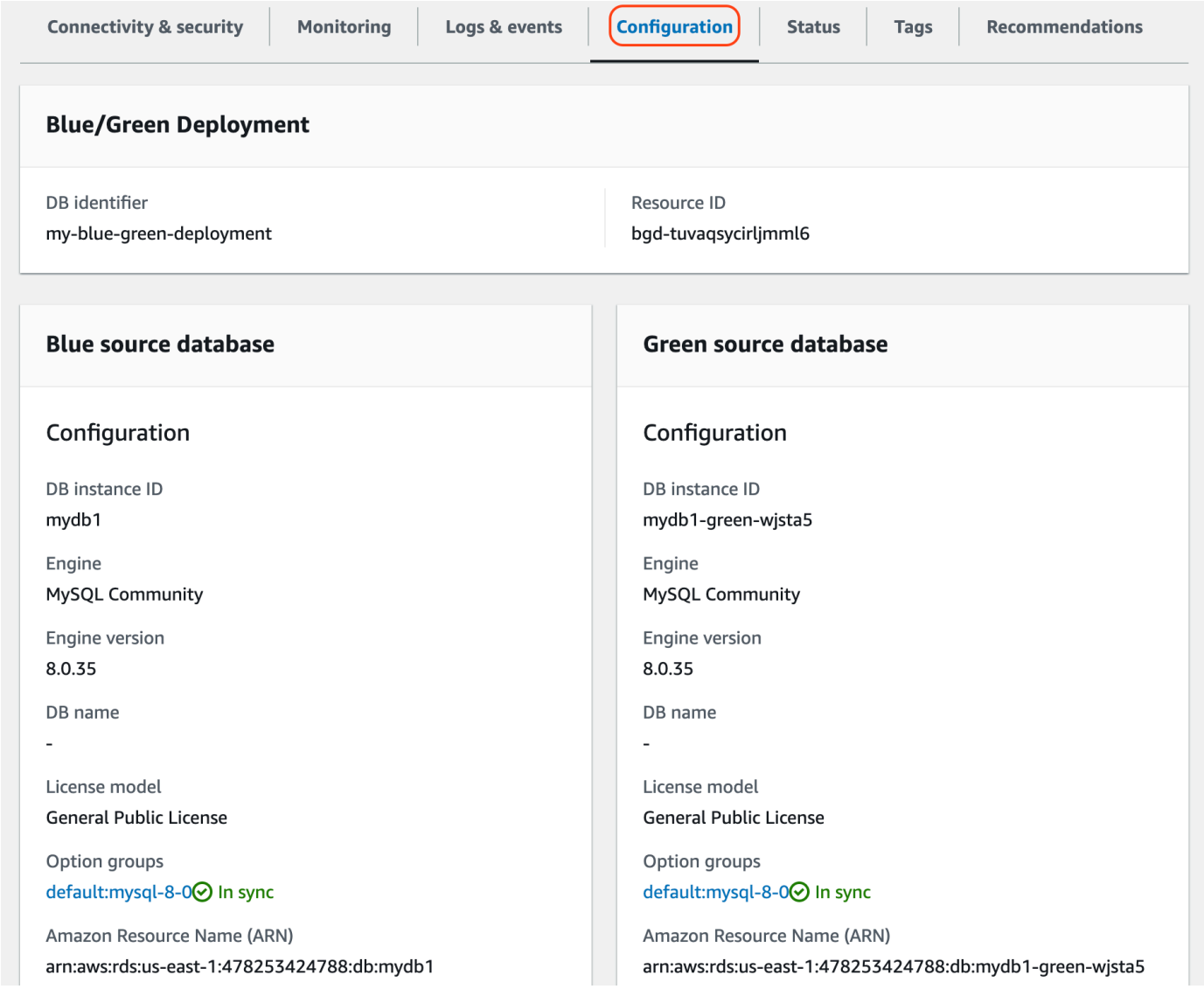
Gambar berikut menunjukkan contoh tab Konektivitas & keamanan:



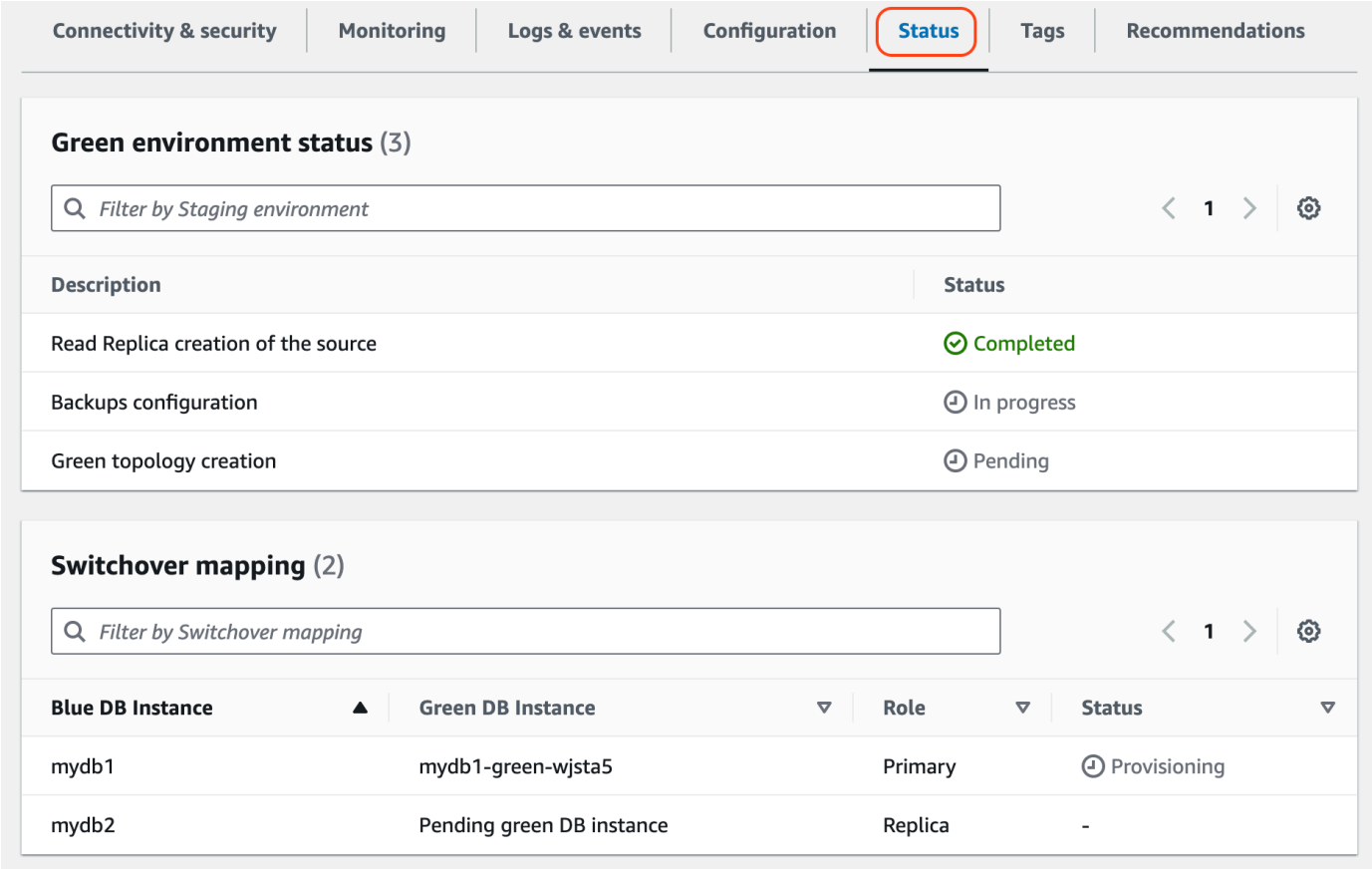
Tab Konektivitas & keamanan juga mencakup bagian yang disebut Replikasi, yang menunjukkan status replikasi logis saat ini dan jeda replika antara lingkungan biru dan hijau. Jika status replikasi adalah `Replicating`, deployment blue/green berhasil direplikasi.

Untuk deployment blue/green RDS for PostgreSQL, status replikasi dapat berubah menjadi `Replication degraded` jika Anda membuat perubahan objek besar atau DDL yang tidak didukung di lingkungan biru. Untuk informasi selengkapnya, lihat the section called “Batasan replikasi logis PostgreSQL”.

Gambar berikut menunjukkan contoh tab Konfigurasi:



Gambar berikut menunjukkan contoh tab Status:



AWS CLI

Untuk melihat detail tentang penyebaran biru/hijau dengan menggunakanAWS CLI, gunakan perintah. describe-blue-green-deployments

Example Melihat detail tentang deployment blue/green dengan memfilter namanya

Saat Anda menggunakan describe-blue-green-deploymentsperintah, Anda dapat memfilter pada file`--blue-green-deployment-name`. Contoh berikut menunjukkan detail untuk deployment blue/green bernama *`my-blue-green-deployment`*.

```
aws rds describe-blue-green-deployments --filters Name=blue-green-deployment-
name,Values=my-blue-green-deployment
```

Example Melihat detail tentang deployment blue/green dengan menentukan pengidentifikasinya

Bila Anda menggunakan describe-blue-green-deploymentsperintah, Anda dapat menentukan`--blue-green-deployment-identifier`. Contoh berikut menunjukkan detail untuk deployment blue/green dengan pengidentifikasi *`bgd-1234567890abcdef`*.

```
aws rds describe-blue-green-deployments --blue-green-deployment-
identifier bgd-1234567890abcdef
```

API RDS

Untuk melihat detail tentang deployment blue/green menggunakan API Amazon RDS, gunakan operasi `DescribeBlueGreenDeployments` dan tentukan `BlueGreenDeploymentIdentifier`.

Mengganti deployment blue/green

Switchover mempromosikan lingkungan hijau untuk menjadi lingkungan produksi baru. Setelah instans DB hijau memiliki replika baca, replika tersebut juga dipromosikan. Sebelum Anda switchover, lalu lintas produksi diarahkan ke instans DB dan replika baca di lingkungan biru. Setelah Anda switchover, lalu lintas produksi diarahkan ke instans DB dan replika baca di lingkungan hijau.

Topik

Waktu habis switchover

Anda dapat menentukan periode waktu habis switchover antara 30 detik dan 3.600 detik (satu jam). Jika switchover memakan waktu lebih lama dari durasi yang ditentukan, maka perubahan apa pun akan dikembalikan dan tidak ada perubahan pada lingkungan mana pun. Periode waktu habis default adalah 300 detik (lima menit).

Pagar pembatas switchover

Saat Anda memulai switchover, Amazon RDS menjalankan beberapa pemeriksaan dasar untuk menguji kesiapan lingkungan biru dan hijau untuk switchover. Pemeriksaan ini dikenal sebagai pagar pembatas switchover. Pagar pembatas switchover ini mencegah switchover jika lingkungan belum

siap. Oleh karena itu, pagar pembatas tersebut dapat mencegah waktu henti yang lebih lama dari yang diharapkan dan mencegah hilangnya data antara lingkungan biru dan hijau yang mungkin terjadi jika switchover dimulai.

Amazon RDS menjalankan pemeriksaan pagar pembatas berikut pada lingkungan hijau:

- Kondisi replikasi – Memeriksa apakah status replikasi instans DB primer hijau berkondisi baik. Instans DB primer hijau adalah replika dari instans DB primer biru.
- Jeda replikasi – Memeriksa apakah jeda replika instans DB primer hijau berada dalam batas yang diizinkan untuk switchover. Batas yang diizinkan didasarkan pada periode waktu habis yang ditentukan. Jeda replika menunjukkan seberapa jauh instans DB primer tertinggal dari instans DB primer biru. Untuk informasi selengkapnya, lihat the section called “Mendiagnosis dan mengatasi jeda di antara replika baca” untuk RDS for MySQL dan the section called “Memantau dan menyetel proses replikasi” untuk RDS for PostgreSQL.
- Penulisan aktif – Memastikan tidak ada penulisan aktif pada instans DB primer hijau.

Amazon RDS menjalankan pemeriksaan pagar pembatas berikut pada lingkungan biru:

- Replikasi eksternal — Untuk PostgreSQL RDS untuk PostgreSQL, pastikan bahwa lingkungan biru bukan sumber logis (penerbit) atau replika (pelanggan) yang dikelola sendiri. Jika ya, kami sarankan Anda melepaskan slot replikasi dan langganan yang dikelola sendiri di semua database di lingkungan biru, lanjutkan dengan peralihan, lalu buat ulang untuk melanjutkan replikasi. Untuk RDS untuk MySQL, pastikan bahwa database biru bukan replika binlog eksternal.
- Penulisan aktif yang berjalan lama – Memastikan tidak ada penulisan aktif yang berjalan lama pada instans DB primer biru karena dapat meningkatkan jeda replika.
- Pernyataan DDL yang berjalan lama – Memastikan tidak ada pernyataan DDL yang berjalan lama pada klaster DB biru karena dapat meningkatkan jeda replika.
- Perubahan PostgreSQL yang tidak didukung – Untuk instans DB RDS for PostgreSQL, memastikan tidak ada perubahan DDL dan tidak ada penambahan atau perubahan objek besar yang dilakukan pada lingkungan biru. Untuk informasi selengkapnya, lihat the section called “Batasan replikasi logis PostgreSQL”.

  Jika Amazon RDS mendeteksi perubahan PostgreSQL yang tidak didukung, status replikasi diubah menjadi `Replication degraded` dan memberi tahu Anda bahwa switchover tidak tersedia untuk deployment blue/green. Untuk melanjutkan switchover, sebaiknya Anda menghapus dan membuat ulang deployment blue/green dan semua basis data hijau. Untuk melakukannya, pilih Tindakan, Hapus dengan basis data hijau.

Tindakan switchover

Saat Anda melakukan switchover pada deployment blue/green, RDS melakukan tindakan berikut:

1. Menjalankan pemeriksaan pagar pembatas untuk memverifikasi apakah lingkungan biru dan hijau siap untuk switchover.
2. Menghentikan operasi tulis baru pada klaster DB di kedua lingkungan.
3. Memutuskan koneksi ke instans DB di kedua lingkungan dan tidak mengizinkan koneksi baru.
4. Menunggu replikasi untuk mengejar ketertinggalan di lingkungan hijau sehingga lingkungan hijau sinkron dengan lingkungan biru.
5. Mengganti nama instans DB dan di kedua lingkungan.

   RDS mengganti nama instans DB di lingkungan hijau agar cocok dengan instans DB di lingkungan biru. Misalnya, asumsikan nama instans DB di lingkungan biru adalah `mydb`. Asumsikan juga nama instans DB yang sesuai di lingkungan hijau adalah `mydb-green-abc123`. Selama switchover, nama instans DB di lingkungan hijau berubah menjadi `mydb`.

   RDS mengganti nama instans DB di lingkungan biru dengan menambahkan `-old`*n* ke nama saat ini, dengan *n* adalah angka. Misalnya, asumsikan nama instans DB di lingkungan biru adalah `mydb`. Setelah switchover, nama instans DB bisa jadi `mydb-old1`.

   RDS juga mengganti nama titik akhir di lingkungan hijau agar sinkron dengan titik akhir yang sesuai di lingkungan biru sehingga perubahan aplikasi tidak diperlukan.
6. Memungkinkan koneksi ke basis data di kedua lingkungan.
7. Memungkinkan operasi tulis pada klaster DB di lingkungan produksi baru.

   Setelah peralihan, DB instans DB primer produksi sebelumnya hanya mengizinkan operasi baca hingga di-boot ulang.

Anda dapat memantau status peralihan menggunakan Amazon. EventBridge Untuk informasi selengkapnya, lihat the section called “Peristiwa deployment blue/green”.

Jika tag sudah dikonfigurasi di lingkungan biru, tag tersebut dipindahkan ke lingkungan produksi baru selama switchover. Lingkungan produksi sebelumnya juga mempertahankan tag ini. Untuk informasi selengkapnya tentang tag, lihat Memberi tag pada sumber daya Amazon RDS.

Jika switchover dimulai lalu berhenti sebelum selesai karena alasan apa pun, maka perubahan apa pun akan dikembalikan, dan tidak ada perubahan yang diterapkan pada lingkungan mana pun.

Praktik terbaik switchover

Sebelum melakukan switchover, Anda sangat dianjurkan untuk mengikuti praktik terbaik dengan menyelesaikan tugas-tugas berikut:

- Uji sumber daya secara menyeluruh di lingkungan hijau. Pastikan sumber daya berfungsi dengan baik dan efisien.
- Pantau CloudWatch metrik Amazon yang relevan. Untuk informasi selengkapnya, lihat the section called “Memverifikasi CloudWatch metrik sebelum peralihan”.
- Identifikasi waktu terbaik untuk melakukan switchover.

  Selama switchover, penulisan terputus dari basis data di kedua lingkungan. Identifikasi waktu ketika lalu lintas berada pada titik terendah di lingkungan produksi Anda. Transaksi yang berjalan lama, seperti DDL aktif, dapat meningkatkan waktu switchover Anda, menghasilkan waktu henti yang lebih lama untuk beban kerja produksi.

  Jika terdapat banyak koneksi pada instans DB, pertimbangkan untuk menguranginya secara manual ke jumlah minimum yang diperlukan untuk aplikasi Anda sebelum Anda melakukan switchover pada deployment blue/green. Salah satu cara untuk melakukannya adalah dengan membuat skrip yang memantau status deployment blue/green dan mulai membersihkan koneksi ketika mendeteksi bahwa status telah berubah menjadi `SWITCHOVER_IN_PROGRESS`.
- Pastikan instans DB di kedua lingkungan berada dalam status `Available`.
- Pastikan instans DB primer di lingkungan hijau berkondisi baik dan bereplikasi.
- Pastikan konfigurasi jaringan dan klien Anda tidak meningkatkan cache DNS Time-To-Live (TTL) lebih dari lima detik, yang merupakan default untuk zona DNS RDS.
  Jika tidak, aplikasi akan terus mengirimkan lalu lintas tulis ke lingkungan biru setelah switchover.
- Anda tidak dapat memutar kembali penerapan biru/hijau setelah peralihan. Untuk beban kerja produksi yang kritis, pertimbangkan untuk menyediakan .
- Pastikan pemuatan data selesai sebelum switchover. Untuk informasi selengkapnya, lihat the section called “Menangani pemuatan lambat”.
- Untuk , lakukan hal berikut:
  - Tinjau batasan replikasi logis dan lakukan tindakan apa pun yang diperlukan sebelum peralihan. Untuk informasi selengkapnya, lihat the section called “Batasan replikasi logis PostgreSQL”.
  - Jalankan operasi `ANALYZE` untuk menyegarkan tabel `pg_statistics`. Ini mengurangi risiko masalah kinerja setelah peralihan.

> **Note**
>
> Selama switchover, Anda tidak dapat mengubah klaster DB apa pun yang disertakan dalam switchover.

Memverifikasi CloudWatch metrik sebelum peralihan

Sebelum Anda mengalihkan penerapan biru/hijau, kami sarankan Anda memeriksa nilai metrik berikut di Amazon. CloudWatch

- `ReplicaLag` – Gunakan metrik ini untuk mengidentifikasi jeda replikasi saat ini pada lingkungan hijau. Untuk mengurangi waktu henti, pastikan nilai ini mendekati nol sebelum Anda switchover.
- `DatabaseConnections` – Gunakan metrik ini untuk memperkirakan tingkat aktivitas pada deployment blue/green, dan pastikan nilainya berada pada tingkat yang dapat diterima untuk deployment Anda sebelum Anda switchover. Jika Wawasan Performa diaktifkan, `DBLoad` adalah metrik yang lebih akurat.

Untuk informasi selengkapnya tentang metrik ini, lihat the section called “CloudWatch metrik untuk RDS”.

Melakukan switchover pada deployment blue/green

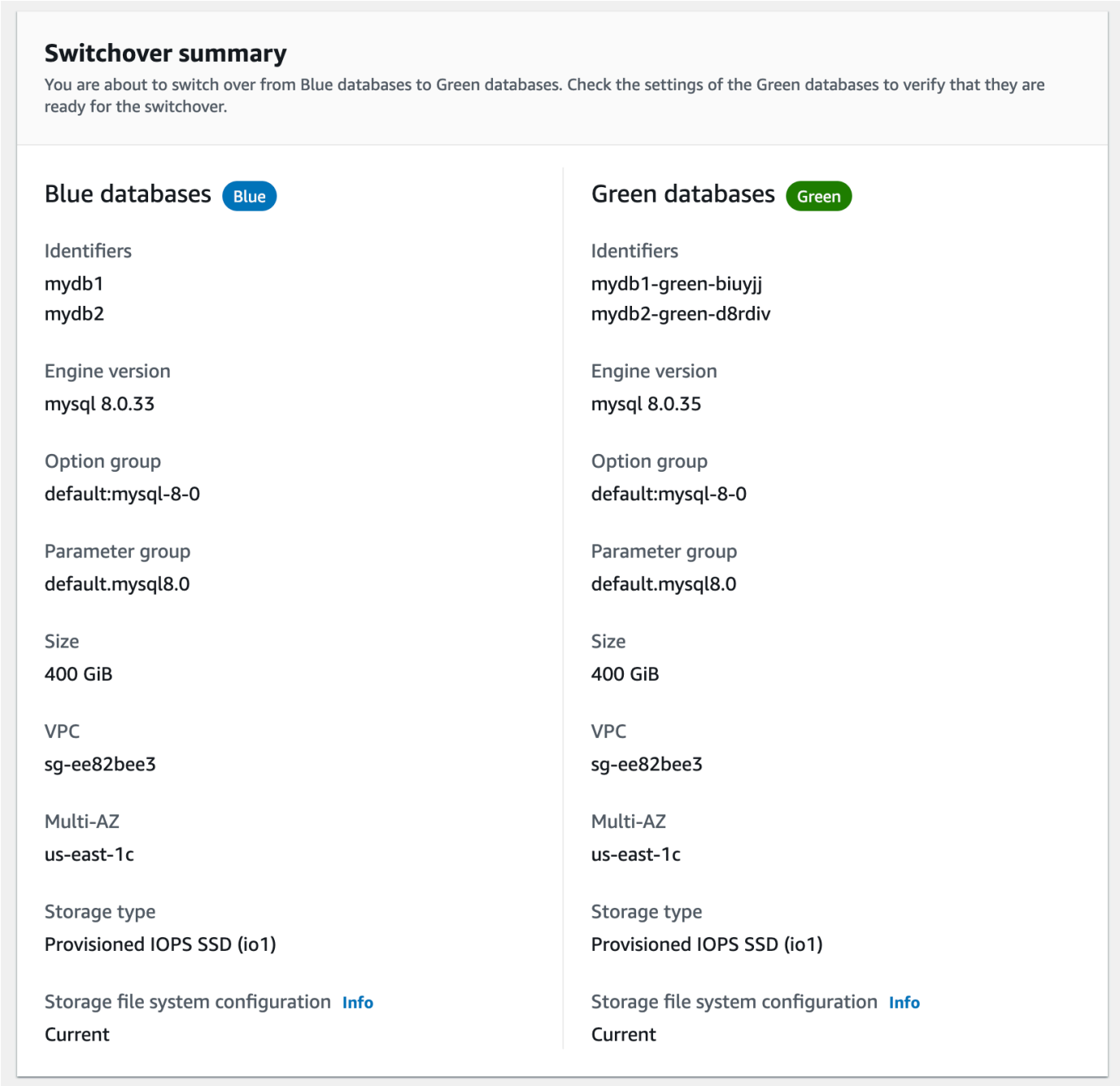
Anda dapat mengalihkan penerapan biru/hijau menggunakan AWS Management Console, API AWS CLI, atau RDS.

Konsol

Untuk melakukan switchover pada deployment blue/green

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data, lalu pilih deployment blue/green yang ingin Anda switchover.
3. Untuk Tindakan, pilih Switchover.

   Halaman Switchover muncul.



4. Di halaman Switchover, tinjau ringkasan switchover. Pastikan sumber daya di kedua lingkungan sesuai dengan yang Anda harapkan. Jika tidak, pilih Batalkan.
5. Untuk Pengaturan waktu habis, masukkan batas waktu untuk switchover.
6. Jika instans menjalankan RDS for PostgreSQL, tinjau dan konfirmasi rekomendasi pra-switchover. Untuk informasi selengkapnya, lihat the section called “Batasan replikasi logis PostgreSQL”.
7. Pilih Switchover.

AWS CLI

Untuk beralih penyebaran biru/hijau dengan menggunakan AWS CLI, gunakan switchover-blue-green-deploymentperintah dengan opsi berikut:

- `--blue-green-deployment-identifier`— Tentukan ID sumber daya dari penerapan biru/hijau.
- `--switchover-timeout` – Tentukan waktu habis untuk switchover, dalam hitungan detik. Angka default-nya adalah 300.

Example Melakukan switchover pada deployment blue/green

Untuk Linux, macOS, atau Unix:

```
aws rds switchover-blue-green-deployment \
    --blue-green-deployment-identifier bgd-1234567890abcdef \
    --switchover-timeout 600
```

Untuk Windows:

```
aws rds switchover-blue-green-deployment ^
    --blue-green-deployment-identifier bgd-1234567890abcdef ^
    --switchover-timeout 600
```

API RDS

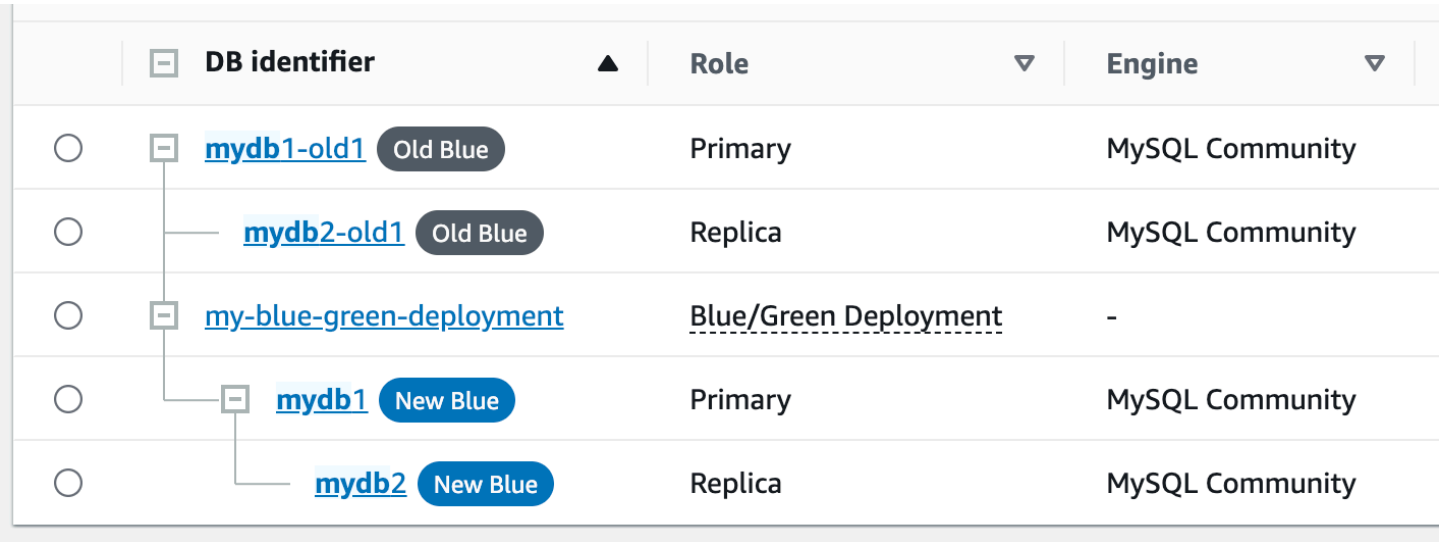
Untuk melakukan switchover pada deployment blue/green menggunakan API Amazon RDS, gunakan operasi `SwitchoverBlueGreenDeployment` dengan parameter berikut:

- `BlueGreenDeploymentIdentifier`— Tentukan ID sumber daya dari penerapan biru/hijau.
- `SwitchoverTimeout` – Tentukan waktu habis untuk switchover, dalam hitungan detik. Angka default-nya adalah 300.

Setelah switchover

Setelah switchover, instans DB di lingkungan biru sebelumnya akan dipertahankan. Biaya standar berlaku untuk sumber daya ini. Replikasi antara lingkungan biru dan hijau berhenti.

RDS mengganti nama instans DB di lingkungan biru dengan menambahkan `-old`*n* ke nama sumber daya saat ini, dengan *n* adalah angka. Instans DB bersifat hanya baca hingga Anda menetapkan parameter `read_only` menjadi `0`.



Memperbarui node induk untuk konsumen

Setelah Anda mengalihkan RDS untuk MariaDB atau RDS untuk MySQL MySQL penyebaran biru/hijau, jika cluster DB biru memiliki replika eksternal atau konsumen log biner sebelum peralihan, Anda harus memperbarui node induk mereka setelah peralihan untuk mempertahankan kontinuitas replikasi.

Setelah peralihan, instance DB yang sebelumnya berada di lingkungan hijau memancarkan peristiwa yang berisi nama file log master dan posisi log master. Sebagai contoh:

```
aws rds describe-events --output json --source-type db-instance --source-identifier db-
instance-identifier

{
    "Events": [
...
        {
            "SourceIdentifier": "db-instance-identifier",
            "SourceType": "db-instance",
            "Message": "Binary log coordinates in green environment after switchover:
             file mysql-bin-changelog.000003 and position 804",
            "EventCategories": [],
            "Date": "2023-11-10T01:33:41.911Z",
            "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:db-instance-identifier"
```

```
        }
    ]
}
```

Pertama, pastikan bahwa konsumen atau replika telah menerapkan semua log biner dari lingkungan biru tua. Kemudian, gunakan koordinat log biner yang disediakan untuk melanjutkan aplikasi pada konsumen. Misalnya, jika Anda menjalankan replika MySQL di EC2, Anda dapat menggunakan perintah: `CHANGE MASTER TO`

```
CHANGE MASTER TO MASTER_HOST='{new-writer-endpoint}', MASTER_LOG_FILE='mysql-bin-
changelog.000003', MASTER_LOG_POS=804;
```

> **Note**
>
> Jika konsumen adalah RDS lain untuk MariaDB atau RDS untuk instance MariaDB DB, Anda dapat menjalankan prosedur tersimpan berikut secara berurutan:,,, dan. the section called “mysql.rds_stop_replication” the section called “mysql.rds_reset_external_master” the section called “mysql.rds_set_external_master” the section called “mysql.rds_start_replication”

Menghapus deployment blue/green

Anda dapat menghapus deployment blue/green sebelum atau setelah switchover.

Saat Anda menghapus deployment blue/green sebelum switchover, Amazon RDS secara opsional menghapus instans DB di lingkungan hijau:

- Jika Anda memilih untuk menghapus instans DB di lingkungan hijau (`--delete-target`), instans harus menonaktifkan perlindungan penghapusan.
- Jika Anda tidak menghapus instans DB di lingkungan hijau (`--no-delete-target`), instans tersebut dipertahankan, tetapi tidak lagi menjadi bagian dari deployment blue/green. Replikasi berlanjut di antara lingkungan.

Opsi untuk menghapus basis data hijau tidak tersedia di konsol setelah switchover. Saat Anda menghapus penerapan biru/hijau menggunakan AWS CLI, Anda tidak dapat menentukan `--delete-target` opsi jika status penerapannya. `SWITCHOVER_COMPLETED`

> **Important**
>
> Menghapus deployment blue/green tidak memengaruhi lingkungan biru.

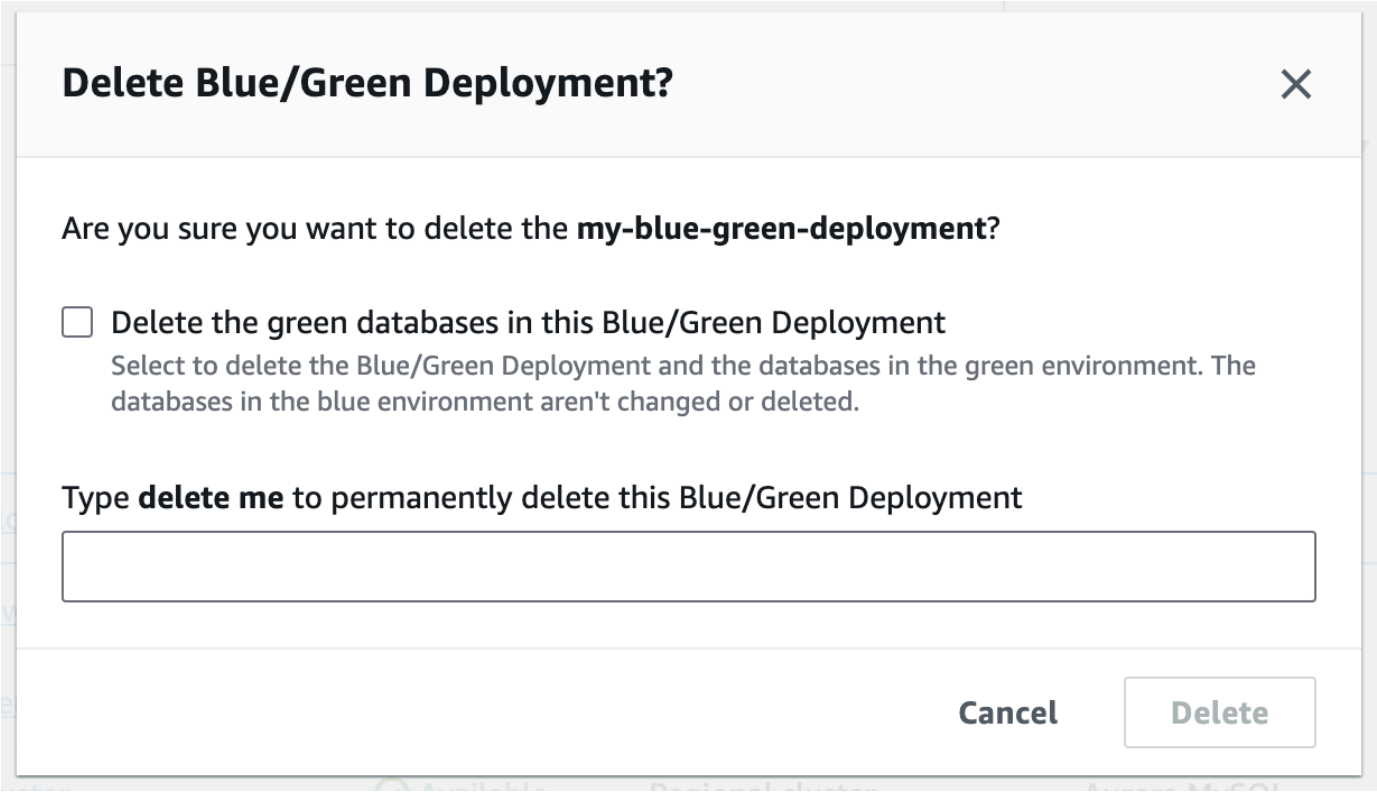
Anda dapat menghapus penerapan biru/hijau menggunakan AWS Management Console, API AWS CLI, atau RDS.

Konsol

Untuk menghapus deployment blue/green

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data, lalu pilih deployment blue/green yang ingin dihapus.
3. Untuk Tindakan, pilih Hapus.

   Jendela Hapus Deployment Blue/Green? muncul.



   Untuk menghapus basis data hijau, pilih Hapus basis data hijau di Deployment Blue/Green ini.
4. Masukkan **`delete me`** di kotak.

5. Pilih Hapus.

AWS CLI

Untuk menghapus penyebaran biru/hijau dengan menggunakan AWS CLI, gunakan delete-blue-green-deploymentperintah dengan opsi berikut:

- `--blue-green-deployment-identifier`— ID sumber daya dari penyebaran biru/hijau yang akan dihapus.
- `--delete-target` – Menentukan bahwa instans DB di lingkungan hijau dihapus. Anda tidak dapat menentukan opsi ini jika deployment blue/green memiliki status `SWITCHOVER_COMPLETED`.
- `--no-delete-target` – Menentukan bahwa instans DB di lingkungan hijau dipertahankan.

Example Menghapus deployment blue/green dan instans DB di lingkungan hijau

Untuk Linux, macOS, atau Unix:

```
aws rds delete-blue-green-deployment \
    --blue-green-deployment-identifier bgd-1234567890abcdef \
    --delete-target
```

Untuk Windows:

```
aws rds delete-blue-green-deployment ^
    --blue-green-deployment-identifier bgd-1234567890abcdef ^
    --delete-target
```

Example Menghapus deployment blue/green tetapi mempertahankan instans DB di lingkungan hijau

Untuk Linux, macOS, atau Unix:

```
aws rds delete-blue-green-deployment \
    --blue-green-deployment-identifier bgd-1234567890abcdef \
    --no-delete-target
```

Untuk Windows:

```
aws rds delete-blue-green-deployment ^
```

```
    --blue-green-deployment-identifier bgd-1234567890abcdef ^
    --no-delete-target
```

API RDS

Untuk menghapus deployment blue/green menggunakan API Amazon RDS, gunakan operasi `DeleteBlueGreenDeployment` dengan parameter berikut:

- `BlueGreenDeploymentIdentifier`— ID sumber daya dari penyebaran biru/hijau yang akan dihapus.
- `DeleteTarget` – Menentukan `TRUE` untuk menghapus instans DB di lingkungan hijau atau `FALSE` untuk mempertahankannya. Tidak bisa `TRUE` jika deployment blue/green memiliki status `SWITCHOVER_COMPLETED`.

Mencadangkan, memulihkan, dan mengekspor data

Bagian ini menunjukkan cara mencadangkan, memulihkan, dan mengekspor data dari instans Amazon RDS DB atau cluster DB multi-AZ.

Topik

Pengantar cadangan

Amazon RDS membuat dan menyimpan cadangan otomatis instans DB atau klaster DB Multi-AZ Anda selama periode pencadangan instans DB Anda. RDS membuat snapshot volume penyimpanan instans DB Anda, sehingga mencadangkan seluruh instans DB data dan bukan hanya masing-masing basis data. RDS menyimpan cadangan otomatis instans DB Anda sesuai dengan periode retensi cadangan yang Anda tentukan. Jika perlu, Anda dapat memulihkan instans DB Anda ke titik waktu mana pun selama periode retensi cadangan.

Pencadangan otomatis mengikuti aturan ini:

- Instans DB Anda harus dalam status `available` agar pencadangan otomatis dapat terjadi. Pencadangan otomatis tidak terjadi saat instans DB Anda berada dalam status selain `available`, misalnya `storage_full`.
- Pencadangan otomatis tidak terjadi saat salinan snapshot DB berjalan di Wilayah AWS yang sama untuk basis data yang sama.

Anda juga dapat mencadangkan instans DB Anda secara manual dengan membuat snapshot DB. Untuk informasi selengkapnya tentang cara membuat snapshot DB secara manual, lihat Membuat snapshot DB untuk instans DB Single-AZ.

Snapshot pertama instans DB berisi data untuk basis data lengkap. Snapshot berikutnya dari basis data yang sama bersifat inkremental, artinya hanya data yang berubah setelah snapshot terbaru Anda yang disimpan.

Anda dapat menyalin snapshot DB otomatis dan manual, dan berbagi snapshot DB manual. Untuk informasi selengkapnya tentang menyalin snapshot DB, lihat Menyalin snapshot DB. Untuk informasi selengkapnya tentang berbagi snapshot DB, lihat Berbagi snapshot DB.

Penyimpanan cadangan

Penyimpanan cadangan Amazon RDS Anda untuk masing-masing Wilayah AWS terdiri dari pencadangan otomatis dan snapshot DB manual untuk Wilayah tersebut. Ruang penyimpanan cadangan total sama dengan jumlah penyimpanan untuk semua cadangan di Wilayah tersebut. Pemindahan snapshot DB ke Wilayah lain akan meningkatkan penyimpanan cadangan di Wilayah tujuan. Cadangan disimpan di Amazon S3.

Untuk informasi selengkapnya tentang biaya penyimpanan cadangan, lihat Harga Amazon RDS.

Jika Anda memilih untuk mempertahankan cadangan otomatis saat Anda menghapus instans DB, cadangan otomatis disimpan selama periode retensi penuh. Jika Anda tidak memilih Pertahankan cadangan otomatis saat Anda menghapus instans DB, semua cadangan otomatis dihapus dengan instans DB. Setelah dihapus, cadangan otomatis tidak dapat dipulihkan. Jika Anda memilih untuk membuat Amazon RDS, buat snapshot DB akhir sebelum menghapus instans DB Anda, Anda dapat menggunakannya untuk memulihkan instans DB Anda. Secara opsional, Anda dapat menggunakan snapshot manual yang dibuat sebelumnya. Snapshot manual tidak dihapus. Anda dapat memiliki hingga 100 snapshot manual per Wilayah.

Mengelola backup otomatis

Bagian ini menunjukkan cara mengelola backup otomatis untuk instans DB dan cluster DB.

Topik

Periode pencadangan

Pencadangan otomatis terjadi setiap hari selama periode pencadangan yang dipilih. Jika pencadangan memerlukan waktu lebih dari yang dialokasikan untuk jendela cadangan, pencadangan akan berlanjut setelah periode berakhir, hingga selesai. Periode pencadangan tidak dapat tumpang-tindih dengan periode pemeliharaan mingguan untuk instans DB atau klaster DB Multi-AZ.

Selama periode pencadangan otomatis, I/O penyimpanan dapat ditangguhkan sesaat sementara proses pencadangan dimulai (biasanya kurang dari beberapa detik). Anda mungkin akan mengalami peningkatan latensi selama beberapa menit saat pencadangan dilakukan untuk deployment Multi-AZ. Untuk MariaDB, MySQL, dan PostgreSQL, aktivitas I/O tidak ditangguhkan pada instans primer Anda selama pencadangan untuk deployment Multi-AZ karena cadangan diambil dari instans siaga. Untuk SQL Server, aktivitas I/O ditangguhkan sesaat selama pencadangan untuk deployment AZ Tunggal dan Multi-AZ karena cadangan diambil dari instans primer. Untuk Db2, aktivitas I/O juga ditangguhkan sesaat selama pencadangan meskipun cadangan diambil dari instans siaga.

Pencadangan otomatis mungkin terkadang dilewati jika instans atau klaster DB memiliki beban kerja yang berat pada saat pencadangan seharusnya dimulai. Jika cadangan dilewati, Anda masih dapat melakukan point-in-time-recovery (PITR), dan cadangan masih dicoba selama jendela cadangan

berikutnya. Untuk informasi selengkapnya tentang PITR, lihat Memulihkan instans DB dengan waktu yang ditentukan.

Jika Anda tidak menentukan periode pencadangan yang diinginkan saat Anda membuat instans DB atau klaster DB Multi-AZ, Amazon RDS menetapkan periode pencadangan 30 menit default. Jendela ini dipilih secara acak dari blok waktu 8 jam untuk masing-masing Wilayah AWS. Tabel berikut mencantumkan blok waktu untuk masing-masing Wilayah AWS dari mana jendela cadangan default ditetapkan.

Nama Wilayah	Wilayah	Blok Waktu
AS Timur (Ohio)	us-east-2	03:00–11:00 UTC
AS Timur (Virginia Utara)	us-east-1	03:00–11:00 UTC
AS Barat (California Utara)	us-west-1	06.00–14.00 UTC
AS Barat (Oregon)	us-west-2	06.00–14.00 UTC
Afrika (Cape Town)	af-south-1	03:00–11:00 UTC
Asia Pasifik (Hong Kong)	ap-east-1	06.00–14.00 UTC
Asia Pasifik (Hyderabad)	ap-south-2	06.30–14.30 UTC
Asia Pasifik (Jakarta)	ap-southeast-3	08.00–16.00 UTC
Asia Pasifik (Melbourne)	ap-southeast-4	11.00–19.00 UTC
Asia Pasifik (Mumbai)	ap-south-1	16.30–00.30 UTC
Asia Pasifik (Osaka)	ap-northeast-3	00.00–08.00 UTC
Asia Pasifik (Seoul)	ap-northeast-2	13.00–21.00 UTC

Nama Wilayah	Wilayah	Blok Waktu
Asia Pasifik (Singapura)	ap-southeast-1	14.00–22.00 UTC
Asia Pasifik (Sydney)	ap-southeast-2	12.00–20.00 UTC
Asia Pasifik (Tokyo)	ap-northeast-1	13.00–21.00 UTC
Kanada (Pusat)	ca-central-1	03:00–11:00 UTC
Kanada Barat (Calgary)	ca-west-1	18:00 — 02:00 UTC
Tiongkok (Beijing)	cn-north-1	06.00–14.00 UTC
Tiongkok (Ningxia)	cn-northwest-1	06.00–14.00 UTC
Eropa (Frankfurt)	eu-central-1	20.00–04.00 UTC
Eropa (Irlandia)	eu-west-1	22.00–06.00 UTC
Eropa (London)	eu-west-2	22.00–06.00 UTC
Eropa (Milan)	eu-south-1	02.00–10.00 UTC
Eropa (Paris)	eu-west-3	07.29–14.29 UTC
Eropa (Spanyol)	eu-south-2	02.00–10.00 UTC
Eropa (Stockholm)	eu-north-1	23.00–07.00 UTC
Eropa (Zurich)	eu-central-2	02.00–10.00 UTC
Israel (Tel Aviv)	il-central-1	03:00–11:00 UTC
Timur Tengah (Bahrain)	me-south-1	06.00–14.00 UTC
Timur Tengah (UEA)	me-central-1	05.00–13.00 UTC

Nama Wilayah	Wilayah	Blok Waktu
Amerika Selatan (Sao Paulo)	sa-east-1	23.00–07.00 UTC
AWS GovCloud (AS-Timur)	us-gov-east-1	17.00–01.00 UTC
AWS GovCloud (AS-Barat)	us-gov-west-1	06.00–14.00 UTC

Periode retensi cadangan

Anda dapat mengatur periode retensi cadangan saat Anda membuat instans DB atau klaster DB Multi-AZ. Jika Anda tidak mengatur periode retensi cadangan, periode retensi cadangan default adalah satu hari jika Anda membuat instans DB menggunakan API Amazon RDS atau AWS CLI. Periode retensi cadangan default adalah tujuh hari jika Anda membuat instans DB menggunakan konsol.

Setelah Anda membuat instans atau klaster DB, Anda dapat memodifikasi periode retensi cadangan. Anda dapat mengatur periode retensi cadangan instans DB antara 0 dan 35 hari. Penetapan periode retensi cadangan menjadi 0 akan menonaktifkan pencadangan otomatis. Anda dapat mengatur periode retensi cadangan klaster DB Multi-AZ antara 1 dan 35 hari. Batas snapshot manual (100 per Wilayah) tidak berlaku untuk cadangan otomatis.

Cadangan otomatis tidak dibuat ketika instans DB atau klaster berhenti. Cadangan dapat dipertahankan lebih lama dari periode retensi cadangan jika instans DB telah dihentikan. RDS tidak menyertakan waktu yang dihabiskan dalam status `stopped` saat periode retensi cadangan dihitung.

> **Important**
>
> Pemadaman terjadi jika Anda mengubah periode retensi cadangan dari 0 ke nilai non-nol atau dari nilai non-nol ke 0. Hal ini berlaku untuk instans DB AZ Tunggal dan Multi-AZ.

Mengaktifkan pencadangan otomatis

Jika instans DB Anda tidak memiliki pencadangan otomatis yang diaktifkan, Anda dapat mengaktifkannya kapan saja. Anda mengaktifkan pencadangan otomatis dengan mengatur periode retensi cadangan ke nilai non-nol positif. Saat pencadangan otomatis diaktifkan, instans DB Anda akan dibuat offline dan cadangan segera dibuat.

> **Note**
>
> Jika Anda mengelola cadangan AWS Backup, Anda tidak dapat mengaktifkan pencadangan otomatis. Untuk informasi selengkapnya, lihat Menggunakan AWS Backup untuk mengelola backup otomatis.

Konsol

Untuk langsung mengaktifkan pencadangan otomatis

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data, lalu pilih instans DB atau klaster DB Multi-AZ yang ingin Anda ubah.
3. Pilih Ubah.
4. Untuk Periode retensi cadangan, pilih nilai positif bukan nol, misalnya 3 hari.
5. Pilih Lanjutkan.
6. Pilih Terapkan langsung.
7. Pilih Ubah instans DB atau Ubah klaster untuk menyimpan perubahan dan mengaktifkan pencadangan otomatis.

AWS CLI

Untuk mengaktifkan backup otomatis, gunakan perintah AWS CLI `modify-db-instance`or `modify-db-cluster`.

Sertakan parameter berikut:

- `--db-instance-identifier` (atau `--db-cluster-identifier` untuk klaster DB Multi-AZ)

- `--backup-retention-period`
- `--apply-immediately` atau `--no-apply-immediately`

Pada contoh berikut, kami mengaktifkan pencadangan otomatis dengan mengatur periode retensi cadangan menjadi tiga hari. Perubahan langsung diterapkan.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance  \
    --backup-retention-period 3 \
    --apply-immediately
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance  ^
    --backup-retention-period 3 ^
    --apply-immediately
```

RDS API

Untuk mengaktifkan pencadangan otomatis, gunakan operasi RDS API `ModifyDBInstance` atau `ModifyDBCluster` dengan parameter yang diperlukan sebagai berikut:

- `DBInstanceIdentifier` atau `DBClusterIdentifier`
- `BackupRetentionPeriod`

Melihat cadangan otomatis

Untuk melihat cadangan otomatis Anda, pilih Pencadangan otomatis di panel navigasi. Untuk melihat snapshot individual yang terkait dengan cadangan otomatis, pilih Snapshot di panel navigasi. Alternatifnya, Anda dapat mendeskripsikan snapshot individual yang terkait dengan cadangan otomatis. Dari sana, Anda dapat memulihkan instans DB langsung dari salah satu snapshot tersebut.

Untuk menjelaskan pencadangan otomatis untuk instans DB Anda yang ada menggunakan AWS CLI, gunakan salah satu perintah berikut:

```
aws rds describe-db-instance-automated-backups --db-instance-
identifier DBInstanceIdentifier
```

atau

```
aws rds describe-db-instance-automated-backups --dbi-resource-id DbiResourceId
```

Untuk mendeskripsikan cadangan otomatis yang dipertahankan untuk instans DB yang ada menggunakan API RDS, panggil tindakan `DescribeDBInstanceAutomatedBackups` dengan salah satu parameter berikut:

- `DBInstanceIdentifier`
- `DbiResourceId`

Mempertahankan cadangan otomatis

> **Note**
>
> Anda hanya dapat mempertahankan cadangan otomatis instans DB, bukan klaster DB Multi-AZ.

Saat menghapus instans DB, Anda dapat memilih untuk mempertahankan cadangan otomatis. Cadangan otomatis dapat dipertahankan selama beberapa hari sama dengan periode retensi cadangan yang dikonfigurasi untuk instans DB pada saat Anda menghapusnya.

Cadangan otomatis yang dipertahankan berisi snapshot sistem dan log transaksi dari instans DB. Hal ini juga termasuk properti instans DB Anda seperti penyimpanan yang dialokasikan dan kelas instans DB, yang diperlukan untuk memulihkannya ke instans aktif.

Cadangan otomatis dan snapshot manual yang dipertahankan dikenai biaya penagihan hingga dihapus. Untuk informasi selengkapnya, lihat Biaya retensi.

Anda dapat mempertahankan cadangan otomatis untuk instans RDS yang menjalankan mesin Db2, MariaDB, MySQL, PostgreSQL, dan Microsoft SQL Server.

Anda dapat memulihkan atau menghapus cadangan otomatis yang dipertahankan menggunakan AWS Management Console, RDS API, dan. AWS CLI

Topik

Periode retensi

Snapshot sistem dan log transaksi dalam cadangan otomatis yang dipertahankan akan kedaluwarsa dengan cara yang sama seperti untuk instans DB sumber. Karena tidak ada snapshot atau log baru yang dibuat untuk instans ini, cadangan otomatis yang dipertahankan pada akhirnya kedaluwarsa sepenuhnya. Pada dasarnya, cadangan otomatis akan aktif selama snapshot sistem terakhirnya aktif, berdasarkan pengaturan untuk periode retensi yang dimiliki sumber instans saat Anda menghapusnya. Cadangan otomatis yang dipertahankan dihapus oleh sistem setelah snapshot sistem terakhirnya kedaluwarsa.

Anda dapat menghapus cadangan otomatis yang dipertahankan dengan cara yang sama seperti Anda dapat menghapus instans DB. Anda dapat menghapus cadangan otomatis yang dipertahankan menggunakan konsol atau operasi API RDS `DeleteDBInstanceAutomatedBackup`.

Snapshot akhir bersifat independen dari cadangan otomatis yang dipertahankan. Kami sangat menyarankan agar Anda mengambil snapshot akhir bahkan jika Anda mempertahankan cadangan otomatis karena cadangan otomatis yang dipertahankan pada akhirnya kedaluwarsa. Snapshot akhir tidak kedaluwarsa.

Melihat cadangan yang dipertahankan

Untuk melihat cadangan otomatis yang dipertahankan, pilih Pencadangan otomatis di panel navigasi, lalu pilih Dipertahankan. Untuk melihat snapshot individual yang terkait dengan cadangan otomatis yang dipertahankan, pilih Snapshot di panel navigasi. Alternatifnya, Anda dapat mendeskripsikan snapshot individual yang terkait dengan cadangan otomatis yang dipertahankan. Dari sana, Anda dapat memulihkan instans DB langsung dari salah satu snapshot tersebut.

Untuk menjelaskan backup otomatis Anda yang dipertahankan menggunakan AWS CLI, gunakan perintah berikut:

```
aws rds describe-db-instance-automated-backups --dbi-resource-id DbiResourceId
```

Untuk mendeskripsikan cadangan otomatis yang dipertahankan menggunakan API RDS, panggil tindakan `DescribeDBInstanceAutomatedBackups` dengan parameter `DbiResourceId`.

Pemulihan

Untuk informasi tentang memulihkan instans DB dari cadangan otomatis, lihat Memulihkan instans DB dengan waktu yang ditentukan.

Biaya retensi

Biaya cadangan otomatis yang dipertahankan adalah biaya penyimpanan total snapshot sistem yang dikaitkan dengannya. Tidak ada biaya tambahan untuk log transaksi atau metadata instans. Semua aturan penetapan harga lainnya untuk cadangan berlaku untuk instans yang dapat dipulihkan.

Misalnya, anggaplah total penyimpanan yang dialokasikan untuk menjalankan proses adalah 100 GB. Misalkan Anda juga memiliki 50 GB snapshot manual ditambah 75 GB snapshot sistem yang terkait dengan cadangan otomatis yang dipertahankan. Dalam hal ini, Anda hanya dikenai biaya untuk penyimpanan cadangan tambahan sebesar 25 GB, seperti ini: (50 GB + 75 GB) – 100 GB = 25 GB.

Batasan

Batasan berikut berlaku untuk cadangan otomatis yang dipertahankan:

- Jumlah maksimum cadangan otomatis yang dipertahankan dalam satu AWS Wilayah adalah 40. Hal ini tidak termasuk dalam kuota instans DB. Anda dapat memiliki 40 instans DB yang berjalan dan 40 cadangan otomatis tambahan yang dipertahankan secara bersamaan.
- Cadangan otomatis yang dipertahankan tidak berisi informasi tentang parameter atau grup opsi.
- Anda dapat memulihkan instans yang dihapus ke titik waktu yang berada dalam periode retensi pada saat penghapusan.
- Anda tidak dapat mengubah cadangan otomatis yang dipertahankan. Hal ini karena cadangan tersebut terdiri dari cadangan sistem, log transaksi, dan properti instans DB yang ada saat Anda menghapus instans sumber.

Menghapus cadangan otomatis yang dipertahankan

Anda dapat menghapus cadangan otomatis yang dipertahankan ketika tidak diperlukan lagi.

Konsol

Untuk menghapus cadangan otomatis yang dipertahankan

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Pencadangan otomatis.
3. Pada tab Dipertahankan, pilih cadangan otomatis yang dipertahankan yang ingin Anda hapus.
4. Untuk Tindakan, pilih Hapus.
5. Di halaman konfirmasi, masukkan **`delete me`** dan pilih Hapus.

AWS CLI

Anda dapat menghapus cadangan otomatis yang dipertahankan dengan menggunakan AWS CLI perintah delete-db-instance-automated-backup dengan opsi berikut:

- `--dbi-resource-id` – Pengidentifikasi sumber daya untuk instans DB sumber.

  Anda dapat menemukan pengenal sumber daya untuk instance DB sumber dari cadangan otomatis yang dipertahankan dengan menjalankan AWS CLI perintah describe-db-instance-automated -backup.

Example

Contoh berikut menghapus cadangan otomatis yang dipertahankan dengan pengidentifikasi sumber daya instans DB sumber `db-123ABCEXAMPLE`.

Untuk Linux, macOS, atau Unix:

```
aws rds delete-db-instance-automated-backup \
    --dbi-resource-id db-123ABCEXAMPLE
```

Untuk Windows:

```
aws rds delete-db-instance-automated-backup ^
```

```
    --dbi-resource-id db-123ABCEXAMPLE
```

API RDS

Anda dapat menghapus cadangan otomatis yang dipertahankan dengan menggunakan operasi Amazon RDS API DeleteDB InstanceAutomatedBackup dengan parameter berikut:

- `DbiResourceId` – Pengidentifikasi sumber daya untuk instans DB sumber.

  Anda dapat menemukan pengenal sumber daya untuk instans DB sumber cadangan otomatis yang dipertahankan menggunakan operasi Amazon RDS API DescribeDB. InstanceAutomatedBackups

Menonaktifkan pencadangan otomatis

Anda mungkin ingin menonaktifkan sementara pencadangan otomatis dalam situasi tertentu, misalnya saat memuat data dalam jumlah besar.

> **Important**
>
> Kami sangat tidak menyarankan untuk menonaktifkan pencadangan otomatis karena menonaktifkan pemulihan. point-in-time Penonaktifan pencadangan otomatis untuk instans DB atau klaster DB Multi-AZ akan menghapus semua cadangan otomatis yang ada. Jika Anda menonaktifkan lalu mengaktifkan kembali pencadangan otomatis, Anda hanya dapat memulihkan mulai dari saat Anda mengaktifkan kembali pencadangan otomatis.

Konsol

Untuk segera menonaktifkan pencadangan otomatis

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data, lalu pilih instans DB atau klaster DB Multi-AZ yang ingin Anda ubah.
3. Pilih Ubah.
4. Untuk Periode retensi cadangan, pilih 0 hari.
5. Pilih Lanjutkan.
6. Pilih Terapkan segera.

7. Pilih Modifikasi instans DB atau Ubah klaster untuk menyimpan perubahan dan menonaktifkan pencadangan otomatis.

AWS CLI

Untuk segera menonaktifkan pencadangan otomatis, gunakan modify-db-clusterperintah modify-db-instanceor dan atur periode retensi cadangan ke 0 dengan. `--apply-immediately`

Example

Contoh berikut segera menonaktifkan pencadangan otomatis pada klaster DB Multi-AZ.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-cluster \
    --db-cluster-identifier mydbcluster \
    --backup-retention-period 0 \
    --apply-immediately
```

Untuk Windows:

```
aws rds modify-db-cluster ^
    --db-cluster-identifier mydbcluster ^
    --backup-retention-period 0 ^
    --apply-immediately
```

Untuk mengetahui kapan modifikasi berlaku, panggil `describe-db-instances` untuk instans DB (atau `describe-db-clusters` untuk klaster DB Multi-AZ) hingga nilai untuk periode retensi cadangan adalah 0 dan status `mydbcluster` tersedia.

```
aws rds describe-db-clusters --db-cluster-identifier mydcluster
```

API RDS

Untuk segera menonaktifkan pencadangan otomatis, lakukan operasi ModifyDBInstance atau ModifyDBClaster dengan parameter berikut:

- `DBInstanceIdentifier = mydbinstance` (atau `DBClusterIdentifier = mydbcluster`)
- `BackupRetentionPeriod = 0`

Example

```
https://rds.amazonaws.com/
    ?Action=ModifyDBInstance
    &DBInstanceIdentifier=mydbinstance
    &BackupRetentionPeriod=0
    &SignatureVersion=2
    &SignatureMethod=HmacSHA256
    &Timestamp=2009-10-14T17%3A48%3A21.746Z
    &AWSAccessKeyId=<&AWS; Access Key ID>
    &Signature=<Signature>
```

Cadangan otomatis dengan mesin penyimpanan MySQL yang tidak didukung

Untuk mesin DB MySQL, cadangan otomatis hanya didukung untuk mesin penyimpanan InnoDB. Penggunaan fitur ini dengan mesin penyimpanan MySQL lainnya, termasuk MyISAM, dapat menyebabkan perilaku yang tidak andal saat Anda memulihkan dari cadangan. Khususnya, karena mesin penyimpanan seperti MyISAM tidak mendukung pemulihan crash yang andal, tabel Anda dapat rusak jika terjadi crash. Karena alasan ini, kami mendorong Anda untuk menggunakan mesin penyimpanan InnoDB.

- Untuk mengonversi tabel MyISAM yang ada ke tabel InnoDB, Anda dapat menggunakan perintah `ALTER TABLE`, misalnya: `ALTER TABLE table_name ENGINE=innodb, ALGORITHM=COPY;`
- Jika Anda memilih untuk menggunakan MyISAM, Anda dapat mencoba memperbaiki secara manual tabel yang rusak setelah terjadi crash dengan menggunakan perintah `REPAIR`. Untuk informasi selengkapnya, lihat REPAIR TABLE statement dalam dokumentasi MySQL. Namun, sebagaimana dijelaskan dalam dokumentasi MySQL, ada kemungkinan besar bahwa Anda tidak dapat memulihkan semua data Anda.
- Jika Anda ingin mengambil snapshot tabel MyISAM Anda sebelum memulihkan, ikuti langkah-langkah berikut:
  1. Hentikan semua aktivitas ke tabel MyISAM Anda (yaitu, tutup semua sesi).

     Anda dapat menutup semua sesi dengan memanggil perintah mysql.rds_kill untuk setiap proses yang ditampilkan dari perintah `SHOW FULL PROCESSLIST`.
  2. Kunci dan lakukan flushing terhadap setiap tabel MyISAM Anda. Misalnya, perintah berikut mengunci dan melakukan flushing terhadap dua tabel yang bernama `myisam_table1` dan `myisam_table2`:

```
mysql> FLUSH TABLES myisam_table, myisam_table2 WITH READ LOCK;
```

3. Buat cuplikan instans DB atau klaster DB Multi-AZ Anda. Saat snapshot selesai, lepaskan kunci dan lanjutkan aktivitas di tabel MyISAM. Anda dapat melepaskan kunci di tabel menggunakan perintah berikut:

```
mysql> UNLOCK TABLES;
```

Langkah-langkah ini memaksa MyISAM untuk melakukan flushing terhadap data yang disimpan dalam memori ke disk, sehingga memastikan awal yang bersih saat Anda memulihkan dari snapshot DB. Untuk informasi selengkapnya tentang membuat snapshot DB, lihat Membuat snapshot DB untuk instans DB Single-AZ.

Cadangan otomatis dengan mesin penyimpanan MariaDB yang tidak didukung

Untuk mesin DB MariaDB, cadangan otomatis hanya didukung untuk mesin penyimpanan InnoDB. Penggunaan fitur ini dengan mesin penyimpanan MariaDB lainnya, termasuk Aria, dapat menyebabkan perilaku yang tidak andal saat Anda memulihkan dari cadangan. Meskipun Aria adalah alternatif yang tahan crash untuk MyISAM, tabel Anda masih dapat rusak jika terjadi crash. Karena alasan ini, kami mendorong Anda untuk menggunakan mesin penyimpanan InnoDB.

- Untuk mengonversi tabel Aria yang ada menjadi tabel InnoDB, Anda dapat menggunakan perintah `ALTER TABLE`. Sebagai contoh: `ALTER TABLE table_name ENGINE=innodb, ALGORITHM=COPY;`
- Jika Anda memilih menggunakan Aria, Anda dapat mencoba memperbaiki secara manual tabel yang rusak setelah terjadi crash dengan menggunakan perintah `REPAIR TABLE`. Untuk informasi selengkapnya, lihat http://mariadb.com/kb/en/mariadb/repair-table/.
- Jika Anda ingin mengambil snapshot dari tabel Aria Anda sebelum memulihkannya, ikuti langkah-langkah berikut ini:
  1. Hentikan semua aktivitas di tabel Aria Anda (yaitu, tutup semua sesi).
  2. Kunci dan lakukan flushing terhadap setiap tabel Aria Anda.
  3. Buat cuplikan instans DB atau klaster DB Multi-AZ Anda. Setelah snapshot selesai, lepaskan kunci dan lanjutkan aktivitas pada tabel Aria. Langkah ini memaksa Aria untuk melakukan

flushing terhadap data yang disimpan di memori ke disk, sehingga memastikan awal yang bersih saat Anda memulihkan dari snapshot DB.

Mereplikasi backup otomatis ke yang lain Wilayah AWS

Untuk menambahkan kemampuan pemulihan bencana, Anda dapat mengonfigurasi instans database Amazon RDS untuk mereplikasi snapshot dan log transaksi ke tujuan Wilayah AWS pilihan Anda. Saat replikasi cadangan dikonfigurasi untuk instans DB, RDS akan memulai penyalinan lintas Wilayah terhadap semua snapshot dan log transaksi begitu snapshot dan log transaksi ini siap di instans DB.

Biaya penyalinan snapshot DB berlaku untuk transfer data. Setelah snapshot DB disalin, biaya standar berlaku untuk penyimpanan di Wilayah tujuan. Untuk detail selengkapnya, lihat Harga RDS.

Untuk contoh menggunakan replikasi cadangan, lihat pembicaraan teknologi AWS online Managed Disaster Recovery dengan Amazon RDS for Oracle Cross-Region Automated Backup.

> **Note**
>
> Replikasi cadangan otomatis tidak didukung untuk cluster DB multi-AZ.

Topik

- Ketersediaan Wilayah dan versi
- Wilayah AWS Dukungan sumber dan tujuan
- Mengaktifkan pencadangan otomatis lintas Wilayah
- Menemukan informasi tentang cadangan yang direplikasi
- Memulihkan ke waktu yang ditentukan dari cadangan yang direplikasi
- Menghentikan replikasi cadangan otomatis
- Menghapus cadangan yang direplikasi

Ketersediaan Wilayah dan versi

Ketersediaan dan dukungan fitur bervariasi di berbagai versi khusus dari setiap mesin basis data, dan di seluruh Wilayah AWS. Untuk informasi selengkapnya tentang ketersediaan versi dan Wilayah dengan cadangan otomatis lintas Wilayah, lihat Pencadangan otomatis lintas Wilayah.

Wilayah AWS Dukungan sumber dan tujuan

Backup replikasi didukung antara berikut Wilayah AWS ini.

Wilayah Sumber	Wilayah Tujuan yang tersedia
Asia Pasifik (Mumbai)	Asia Pasifik (Singapura) AS Timur (Virginia Utara), AS Timur (Ohio), AS Barat (Oregon)
Asia Pasifik (Osaka)	Asia Pasifik (Tokyo)
Asia Pasifik (Seoul)	Asia Pasifik (Singapura), Asia Pasifik (Tokyo) AS Timur (Virginia Utara), AS Timur (Ohio), AS Barat (Oregon)
Asia Pasifik (Singapura)	Asia Pasifik (Mumbai), Asia Pasifik (Seoul), Asia Pasifik (Sydney), Asia Pasifik (Tokyo) AS Timur (Virginia Utara), AS Timur (Ohio), AS Barat (Oregon)
Asia Pasifik (Sydney)	Asia Pasifik (Singapura) AS Timur (Virginia Utara), AS Barat (California Utara), AS Barat (Oregon)
Asia Pasifik (Tokyo)	Asia Pasifik (Osaka), Asia Pasifik (Seoul), Asia Pasifik (Singapura) AS Timur (Virginia Utara), AS Timur (Ohio), AS Barat (Oregon)
Kanada (Pusat)	Eropa (Irlandia) AS Timur (Virginia Utara), AS Timur (Ohio), AS Barat (California Utara), AS Barat (Oregon)
Tiongkok (Beijing)	Tiongkok (Ningxia)
Tiongkok (Ningxia)	Tiongkok (Beijing)
Eropa (Frankfurt)	Eropa (Irlandia), Eropa (London), Eropa (Paris), Eropa (Stockholm) AS Timur (Virginia Utara), AS Timur (Ohio), AS Barat (Oregon)
Eropa (Irlandia)	Kanada (Pusat) Eropa (Frankfurt), Eropa (London), Eropa (Paris), Eropa (Stockholm)

Wilayah Sumber	Wilayah Tujuan yang tersedia
	AS Timur (Virginia Utara), AS Timur (Ohio), AS Barat (California Utara), AS Barat (Oregon)
Eropa (London)	Eropa (Frankfurt), Eropa (Irlandia), Eropa (Paris), Eropa (Stockholm) AS Timur (Virginia Utara)
Eropa (Paris)	Eropa (Frankfurt), Eropa (Irlandia), Eropa (London), Eropa (Stockholm) AS Timur (Virginia Utara)
Eropa (Stockholm)	Eropa (Frankfurt), Eropa (Irlandia), Eropa (London), Eropa (Paris) AS Timur (Virginia Utara)
Amerika Selatan (Sao Paulo)	AS Timur (Virginia Utara), AS Timur (Ohio)
AWS GovCloud (AS-Timur)	AWS GovCloud (AS-Barat)
AWS GovCloud (AS-Barat)	AWS GovCloud (AS-Timur)
AS Timur (Virginia Utara)	Asia Pasifik (Mumbai), Asia Pasifik (Seoul), Asia Pasifik (Singapura), Asia Pasifik (Sydney), Asia Pasifik (Tokyo) Kanada (Pusat) Eropa (Frankfurt), Eropa (Irlandia), Eropa (London), Eropa (Paris), Eropa (Stockholm) Amerika Selatan (Sao Paulo) AS Timur (Ohio), AS Barat (California Utara), AS Barat (Oregon)

Wilayah Sumber	Wilayah Tujuan yang tersedia
AS Timur (Ohio)	Asia Pasifik (Mumbai), Asia Pasifik (Seoul), Asia Pasifik (Singapura), Asia Pasifik (Tokyo) Kanada (Pusat) Eropa (Frankfurt), Eropa (Irlandia) Amerika Selatan (Sao Paulo) AS Timur (Virginia Utara), AS Barat (California Utara), AS Barat (Oregon)
AS Barat (California Utara)	Asia Pasifik (Sydney) Kanada (Pusat) Eropa (Irlandia) AS Timur (Virginia Utara), AS Timur (Ohio), AS Barat (Oregon)
AS Barat (Oregon)	Asia Pasifik (Mumbai), Asia Pasifik (Seoul), Asia Pasifik (Singapura), Asia Pasifik (Sydney), Asia Pasifik (Tokyo) Kanada (Pusat) Eropa (Frankfurt), Eropa (Irlandia) AS Timur (Virginia Utara), AS Timur (Ohio), AS Barat (California Utara)

Anda juga dapat menggunakan `describe-source-regions` AWS CLI perintah untuk mencari tahu mana yang Wilayah AWS dapat mereplikasi satu sama lain. Untuk informasi selengkapnya, lihat Menemukan informasi tentang cadangan yang direplikasi.

Mengaktifkan pencadangan otomatis lintas Wilayah

Anda dapat mengaktifkan replikasi cadangan pada instans DB baru atau yang sudah ada menggunakan konsol Amazon RDS. Anda juga dapat menggunakan `start-db-`

`instance-automated-backups-replication` AWS CLI perintah atau operasi `StartDBInstanceAutomatedBackupsReplication` RDS API. Anda dapat mereplikasi hingga 20 cadangan ke setiap tujuan Wilayah AWS untuk masing-masing. Akun AWS

> **Note**
>
> Untuk dapat mereplikasi cadangan otomatis, pastikan untuk mengaktifkannya. Untuk informasi selengkapnya, lihat Mengaktifkan pencadangan otomatis.

Konsol

Anda dapat mengaktifkan replikasi cadangan untuk instans DB baru atau yang sudah ada:

- Untuk instans DB baru, aktifkan saat Anda meluncurkan instans. Untuk informasi selengkapnya, lihat Pengaturan untuk instans DB.
- Untuk instans DB yang sudah ada, gunakan prosedur berikut.

Untuk mengaktifkan replikasi cadangan untuk instans DB yang sudah ada

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Pencadangan otomatis.
3. Di tab Wilayah Saat Ini, pilih instans DB yang ingin Anda aktifkan replikasi cadangannya.
4. Untuk Tindakan, pilih Kelola replikasi lintas Wilayah.
5. Di bagian Replikasi cadangan, pilih Aktifkan replikasi ke Wilayah AWS lain.
6. Pilih Wilayah Tujuan.
7. Pilih Periode penyimpanan cadangan yang direplikasi.
8. Jika Anda telah mengaktifkan enkripsi pada instans DB sumber, pilih AWS KMS key untuk mengenkripsi cadangan.
9. Pilih Simpan.

Di Wilayah sumber, replikasi cadangan tercantum pada tab Wilayah Saat Ini di halaman Pencadangan otomatis. Di Wilayah tujuan, cadangan yang direplikasi tercantum pada tab Cadangan tereplikasi di halaman Pencadangan otomatis.

AWS CLI

Aktifkan replikasi cadangan dengan menggunakan `start-db-instance-automated-backups-replication` AWS CLI perintah.

Contoh CLI berikut mereplikasi cadangan otomatis dari instans DB di Wilayah AS Barat (Oregon) ke Wilayah AS Timur (Virginia Utara). Ini juga mengenkripsi cadangan yang direplikasi, menggunakan di Wilayah tujuan. AWS KMS key

Untuk mengaktifkan replikasi cadangan

- Gunakan salah satu perintah berikut ini.

  Untuk Linux, macOS, atau Unix:

```
aws rds start-db-instance-automated-backups-replication \
--region us-east-1 \
--source-db-instance-arn "arn:aws:rds:us-west-2:123456789012:db:mydatabase" \
--kms-key-id "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE" \
--backup-retention-period 7
```

  Untuk Windows:

```
aws rds start-db-instance-automated-backups-replication ^
--region us-east-1 ^
--source-db-instance-arn "arn:aws:rds:us-west-2:123456789012:db:mydatabase" ^
--kms-key-id "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE" ^
--backup-retention-period 7
```

  `--source-region`Opsi ini diperlukan saat Anda mengenkripsi cadangan antara Wilayah AWS GovCloud (AS-Timur) dan AWS GovCloud (AS-Barat). Untuk `--source-region`, tentukan Wilayah AWS instans DB sumber.

  Jika `--source-region` tidak ditentukan, pastikan untuk menentukan nilai `--pre-signed-url`. URL yang telah ditandatangani adalah URL yang berisi permintaan bertanda tangan Signature Versi 4 untuk perintah `start-db-instance-automated-backups-replication` yang dipanggil di Wilayah AWS sumber. Untuk mempelajari lebih lanjut tentang *`pre-signed-url`* opsi, lihat start-db-instance-automated-backups-replikasi di Referensi Perintah. AWS CLI

API RDS

Mengaktifkan replikasi cadangan dengan menggunakan operasi `StartDBInstanceAutomatedBackupsReplication` API RDS dengan parameter berikut:

- `Region`
- `SourceDBInstanceArn`
- `BackupRetentionPeriod`
- `KmsKeyId` (opsional)
- `PreSignedUrl` (wajib jika Anda menggunakan `KmsKeyId`)

> **Note**
>
> Jika Anda mengenkripsi cadangan, Anda juga harus menyertakan URL yang telah ditandatangani sebelumnya. Untuk informasi lebih lanjut tentang URL yang telah ditandatangani sebelumnya, lihat Mengautentikasi Permintaan: Menggunakan Parameter Kueri (AWS Signature Versi 4) dalam Referensi API Amazon Simple Storage Service dan Proses penandatanganan Signature Versi 4 dalam Referensi Umum AWS .

Menemukan informasi tentang cadangan yang direplikasi

Anda dapat menggunakan perintah CLI berikut untuk menemukan informasi tentang cadangan yang direplikasi:

- `describe-source-regions`
- `describe-db-instances`
- `describe-db-instance-automated-backups`

`describe-source-regions`Contoh berikut mencantumkan sumber Wilayah AWS dari mana cadangan otomatis dapat direplikasi ke Wilayah tujuan AS Barat (Oregon).

Untuk menampilkan informasi tentang Wilayah sumber

- Jalankan perintah berikut.

```
aws rds describe-source-regions --region us-west-2
```

Output menunjukkan bahwa cadangan dapat direplikasi dari AS Timur (Virginia Utara), tetapi tidak dari AS Timur (Ohio) atau AS Barat (California Utara), ke AS Barat (Oregon).

```
{
    "SourceRegions": [
        ...
        {
            "RegionName": "us-east-1",
            "Endpoint": "https://rds.us-east-1.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": true
        },
        {
            "RegionName": "us-east-2",
            "Endpoint": "https://rds.us-east-2.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": false
        },
            "RegionName": "us-west-1",
            "Endpoint": "https://rds.us-west-1.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": false
        }
    ]
}
```

Contoh `describe-db-instances` berikut menunjukkan cadangan otomatis untuk instans DB.

Untuk menampilkan cadangan yang direplikasi untuk instans DB

- Gunakan salah satu perintah berikut ini.

  Untuk Linux, macOS, atau Unix:

  ```
  aws rds describe-db-instances \
  --db-instance-identifier mydatabase
  ```

  Untuk Windows:

```
aws rds describe-db-instances ^
--db-instance-identifier mydatabase
```

Output ini mencakup cadangan yang direplikasi.

```
{
    "DBInstances": [
        {
            "StorageEncrypted": false,
            "Endpoint": {
                "HostedZoneId": "Z1PVIF0B656C1W",
                "Port": 1521,
            ...

            "BackupRetentionPeriod": 7,
            "DBInstanceAutomatedBackupsReplications":
 [{"DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-east-1:123456789012:auto-backup:ab-
L2IJCEXJP7XQ7HOJ4SIEXAMPLE"}]
        }
    ]
}
```

Contoh `describe-db-instance-automated-backups` berikut menunjukkan cadangan otomatis untuk instans DB.

Untuk menampilkan cadangan otomatis untuk instans DB

- Gunakan salah satu perintah berikut ini.

  Untuk Linux, macOS, atau Unix:

  ```
  aws rds describe-db-instance-automated-backups \
  --db-instance-identifier mydatabase
  ```

  Untuk Windows:

  ```
  aws rds describe-db-instance-automated-backups ^
  --db-instance-identifier mydatabase
  ```

Output menunjukkan instans DB sumber dan cadangan otomatis di AS Barat (Oregon), dengan replikasi cadangan ke AS Timur (Virginia Utara).

```
{
    "DBInstanceAutomatedBackups": [
        {
            "DBInstanceArn": "arn:aws:rds:us-west-2:868710585169:db:mydatabase",
            "DbiResourceId": "db-L2IJCEXJP7XQ7HOJ4SIEXAMPLE",
            "DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-west-2:123456789012:auto-
backup:ab-L2IJCEXJP7XQ7HOJ4SIEXAMPLE",
            "BackupRetentionPeriod": 7,
            "DBInstanceAutomatedBackupsReplications":
 [{"DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-east-1:123456789012:auto-backup:ab-
L2IJCEXJP7XQ7HOJ4SIEXAMPLE"}]
            "Region": "us-west-2",
            "DBInstanceIdentifier": "mydatabase",
            "RestoreWindow": {
                "EarliestTime": "2020-10-26T01:09:07Z",
                "LatestTime": "2020-10-31T19:09:53Z",
            }
            ...
        }
    ]
}
```

Contoh `describe-db-instance-automated-backups` berikut menggunakan opsi `--db-instance-automated-backups-arn` untuk menampilkan cadangan yang direplikasi di Wilayah tujuan.

Untuk menampilkan cadangan yang direplikasi

- Gunakan salah satu perintah berikut ini.

  Untuk Linux, macOS, atau Unix:

  ```
  aws rds describe-db-instance-automated-backups \
  --db-instance-automated-backups-arn "arn:aws:rds:us-east-1:123456789012:auto-
  backup:ab-L2IJCEXJP7XQ7HOJ4SIEXAMPLE"
  ```

  Untuk Windows:

  ```
  aws rds describe-db-instance-automated-backups ^
  ```

```
--db-instance-automated-backups-arn "arn:aws:rds:us-east-1:123456789012:auto-
backup:ab-L2IJCEXJP7XQ7HOJ4SIEXAMPLE"
```

Output menunjukkan instans DB sumber di AS Barat (Oregon), dengan replikasi cadangan di AS Timur (Virginia Utara).

```
{
    "DBInstanceAutomatedBackups": [
        {
            "DBInstanceArn": "arn:aws:rds:us-west-2:868710585169:db:mydatabase",
            "DbiResourceId": "db-L2IJCEXJP7XQ7HOJ4SIEXAMPLE",
            "DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-east-1:123456789012:auto-
backup:ab-L2IJCEXJP7XQ7HOJ4SIEXAMPLE",
            "Region": "us-west-2",
            "DBInstanceIdentifier": "mydatabase",
            "RestoreWindow": {
                "EarliestTime": "2020-10-26T01:09:07Z",
                "LatestTime": "2020-10-31T19:01:23Z"
            },
            "AllocatedStorage": 50,
            "BackupRetentionPeriod": 7,
            "Status": "replicating",
            "Port": 1521,
            ...
        }
    ]
}
```

Memulihkan ke waktu yang ditentukan dari cadangan yang direplikasi

Anda dapat memulihkan instans DB ke titik waktu tertentu dari cadangan yang direplikasi menggunakan konsol Amazon RDS. Anda juga dapat menggunakan `restore-db-instance-to-point-in-time` AWS CLI perintah atau operasi `RestoreDBInstanceToPointInTime` RDS API.

Untuk informasi umum tentang point-in-time pemulihan (PITR), lihatMemulihkan instans DB dengan waktu yang ditentukan.

> **Note**
>
> Pada RDS untuk SQL Server, grup opsi tidak disalin Wilayah AWS saat pencadangan otomatis direplikasi. Jika Anda telah mengaitkan grup opsi kustom dengan instans DB RDS

for SQL Server, Anda dapat membuat ulang grup opsi tersebut di Wilayah tujuan. Kemudian, pulihkan instans DB di Wilayah tujuan dan kaitkan grup opsi kustom dengannya. Untuk informasi selengkapnya, lihat Menggunakan grup opsi.

Konsol

Untuk memulihkan instans DB ke waktu yang ditentukan dari cadangan yang direplikasi

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Pilih Wilayah tujuan (tempat cadangan direplikasi) dari pemilih Wilayah.
3. Di panel navigasi, pilih Pencadangan otomatis.
4. Di tab Cadangan tereplikasi, pilih instans DB yang ingin Anda pulihkan.
5. Untuk Tindakan, pilih Pulihkan ke titik waktu.
6. Pilih Waktu pemulihan terbaru untuk memulihkan ke waktu terbaru yang dimungkinkan atau pilih Kustom untuk memilih waktu.

   Jika Anda memilih Kustom, masukkan tanggal dan waktu yang Anda inginkan untuk memulihkan instans.

   > **Note**
   >
   > Waktu ditampilkan dalam zona waktu lokal Anda, yang ditunjukkan dengan offset dari Waktu Universal Terkoordinasi (UTC). Misalnya, UTC-5 adalah Waktu Standar Timur/Waktu Musim Panas Tengah.

7. Untuk Pengidentifikasi instans DB, masukkan nama instans DB target yang dipulihkan.
8. (Opsional) Pilih opsi lain sesuai kebutuhan, seperti mengaktifkan penskalaan otomatis.
9. Pilih Pulihkan ke titik waktu.

AWS CLI

Gunakan `restore-db-instance-to-point-in-time` AWS CLI perintah untuk membuat instance DB baru.

Untuk memulihkan instans DB ke waktu yang ditentukan dari cadangan yang direplikasi

- Gunakan salah satu perintah berikut ini.

  Untuk Linux, macOS, atau Unix:

```
aws rds restore-db-instance-to-point-in-time \
    --source-db-instance-automated-backups-arn "arn:aws:rds:us-
east-1:123456789012:auto-backup:ab-L2IJCEXJP7XQ7HOJ4SIEXAMPLE" \
    --target-db-instance-identifier mytargetdbinstance \
    --restore-time 2020-10-14T23:45:00.000Z
```

  Untuk Windows:

```
aws rds restore-db-instance-to-point-in-time ^
    --source-db-instance-automated-backups-arn "arn:aws:rds:us-
east-1:123456789012:auto-backup:ab-L2IJCEXJP7XQ7HOJ4SIEXAMPLE" ^
    --target-db-instance-identifier mytargetdbinstance ^
    --restore-time 2020-10-14T23:45:00.000Z
```

API RDS

Untuk memulihkan instans DB ke waktu yang ditentukan, panggil operasi API Amazon RDS `RestoreDBInstanceToPointInTime` dengan parameter berikut ini:

- `SourceDBInstanceAutomatedBackupsArn`
- `TargetDBInstanceIdentifier`
- `RestoreTime`

Menghentikan replikasi cadangan otomatis

Anda dapat mengaktifkan replikasi cadangan untuk instans DB menggunakan konsol Amazon RDS. Anda juga dapat menggunakan `stop-db-instance-automated-backups-replication` AWS CLI perintah atau operasi `StopDBInstanceAutomatedBackupsReplication` RDS API.

Cadangan yang direplikasi dipertahankan, tergantung pada periode retensi cadangan yang ditetapkan saat pembuatannya.

Konsol

Menghentikan replikasi cadangan dari halaman Pencadangan otomatis di Wilayah sumber.

Untuk menghentikan replikasi cadangan ke Wilayah AWS

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Pilih Wilayah sumber dari Pemilih Wilayah.
3. Di panel navigasi, pilih Pencadangan otomatis.
4. Di tab Wilayah saat ini, pilih instans DB yang ingin Anda hentikan replikasi cadangannya.
5. Untuk Tindakan, pilih Kelola replikasi lintas Wilayah.
6. Di bagian Replikasi cadangan, kosongkan kotak centang Aktifkan replikasi ke Wilayah AWS lain.
7. Pilih Simpan.

Cadangan yang direplikasi tercantum pada tab Dipertahankan di halaman Pencadangan otomatis di Wilayah tujuan.

AWS CLI

Hentikan replikasi cadangan dengan menggunakan `stop-db-instance-automated-backups-replication` AWS CLI perintah.

Contoh CLI berikut menghentikan cadangan otomatis instans DB agar tidak direplikasi di Wilayah AS Barat (Oregon).

Untuk menghentikan replikasi cadangan

- Gunakan salah satu perintah berikut ini.

  Untuk Linux, macOS, atau Unix:

```
aws rds stop-db-instance-automated-backups-replication \
--region us-east-1 \
--source-db-instance-arn "arn:aws:rds:us-west-2:123456789012:db:mydatabase"
```

  Untuk Windows:

```
aws rds stop-db-instance-automated-backups-replication ^
```

```
--region us-east-1 ^
--source-db-instance-arn "arn:aws:rds:us-west-2:123456789012:db:mydatabase"
```

API RDS

Hentikan replikasi cadangan dengan menggunakan operasi [StopDBInstanceAutomatedBackupsReplication](#) API RDS dengan parameter berikut:

- `Region`
- `SourceDBInstanceArn`

Menghapus cadangan yang direplikasi

Anda dapat menghapus cadangan yang direplikasi untuk instans DB menggunakan konsol Amazon RDS. Anda juga dapat menggunakan `delete-db-instance-automated-backups` AWS CLI perintah atau operasi `DeleteDBInstanceAutomatedBackup` RDS API.

Konsol

Hapus cadangan yang direplikasi di Wilayah tujuan dari halaman Pencadangan otomatis.

Untuk menghapus cadangan yang direplikasi

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Pilih Wilayah tujuan dari Pemilih Wilayah.
3. Di panel navigasi, pilih Pencadangan otomatis.
4. Di tab Cadangan tereplikasi, pilih instans DB yang ingin Anda hapus cadangan tereplikasinya.
5. Untuk Tindakan, pilih Hapus.
6. Di halaman konfirmasi, masukkan **`delete me`** dan pilih Hapus.

AWS CLI

Hapus cadangan yang direplikasi dengan menggunakan perintah. [delete-db-instance-automated-backup](#) AWS CLI

Anda dapat menggunakan perintah `describe-db-instances` CLI untuk menemukan Amazon Resource Names (ARN) dari cadangan yang direplikasi. Untuk informasi selengkapnya, lihat Menemukan informasi tentang cadangan yang direplikasi.

Untuk menghapus cadangan yang direplikasi

- Gunakan salah satu perintah berikut ini.

  Untuk Linux, macOS, atau Unix:

```
aws rds delete-db-instance-automated-backup \
--db-instance-automated-backups-arn "arn:aws:rds:us-east-1:123456789012:auto-
backup:ab-L2IJCEXJP7XQ7HOJ4SIEXAMPLE"
```

  Untuk Windows:

```
aws rds delete-db-instance-automated-backup ^
--db-instance-automated-backups-arn "arn:aws:rds:us-east-1:123456789012:auto-
backup:ab-L2IJCEXJP7XQ7HOJ4SIEXAMPLE"
```

API RDS

Hapus replikasi cadangan dengan menggunakan operasi `DeleteDBInstanceAutomatedBackup` API RDS dengan parameter `DBInstanceAutomatedBackupsArn`.

Mengelola backup manual

Bagian ini menunjukkan cara mengelola backup otomatis untuk instans DB dan cluster DB.

Topik

Membuat snapshot DB untuk instans DB Single-AZ

Amazon RDS membuat cuplikan volume penyimpanan instans basis data Anda, sehingga mencadangkan seluruh instans basis data dan bukan hanya masing-masing basis data. Membuat snapshot DB ini pada instans DB Single-AZ menghasilkan suspensi I/O singkat yang dapat bertahan beberapa detik hingga beberapa menit, tergantung pada ukuran dan kelas instans DB Anda. Untuk MariaDB, MySQL, Oracle, dan PostgreSQL, aktivitas I/O tidak ditangguhkan pada primer Anda selama pencadangan untuk deployment Multi-AZ, karena pencadangan diambil dari mode siaga. Untuk SQL Server, aktivitas I/O ditangguhkan sebentar selama pencadangan untuk deployment Multi-AZ.

Saat membuat snapshot DB, Anda perlu mengidentifikasi instans DB mana yang akan Anda cadangkan, kemudian beri nama snapshot DB sehingga Anda dapat memulihkannya nanti. Jumlah waktu yang diperlukan untuk membuat snapshot bervariasi sesuai ukuran basis data Anda. Karena snapshot menyertakan seluruh volume penyimpanan, ukuran file, seperti file sementara, juga memengaruhi jumlah waktu yang diperlukan untuk membuat snapshot.

> **Note**
>
> Instans DB Anda harus dalam status `available` untuk mengambil snapshot DB. Untuk instans DB PostgreSQL, data dalam tabel yang tidak masuk log mungkin tidak dipulihkan dari snapshot. Untuk informasi selengkapnya, lihat Praktik terbaik untuk menggunakan PostgreSQL.

Tidak seperti pencadangan otomatis, snapshot manual tidak bergantung pada periode retensi pencadangan. Snapshots tidak kedaluwarsa.

Untuk pencadangan jangka panjang data MariaDB, MySQL, dan PostgreSQL, sebaiknya ekspor data snapshot ke Amazon S3. Jika versi utama mesin DB Anda tidak didukung lagi, Anda tidak dapat memulihkan ke versi tersebut dari snapshot. Untuk informasi selengkapnya, lihat Mengekspor data snapshot DB ke Amazon S3.

Anda dapat membuat snapshot DB menggunakan AWS Management Console, AWS CLI, atau RDS API.

Konsol

Untuk membuat snapshot DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Snapshot.

   Daftar Snapshot manual akan muncul.
3. Pilih Ambil snapshot.

   Jendela Ambil snapshot DB akan muncul.
4. Pilih instance DB yang ingin Anda ambil snapshot.
5. Masukkan nama Snapshot.
6. Pilih Ambil snapshot.

Daftar snapshot Manual muncul, dengan status snapshot DB baru ditampilkan sebagai `Creating` Setelah statusnya adalah `Available`, Anda dapat melihat waktu pembuatannya.

AWS CLI

Saat Anda membuat snapshot DB menggunakan AWS CLI, Anda perlu mengidentifikasi instans DB mana yang akan Anda cadangkan, dan kemudian beri nama snapshot DB Anda sehingga Anda dapat memulihkannya nanti. Anda dapat melakukan ini dengan menggunakan AWS CLI `create-db-snapshot`perintah dengan parameter berikut:

- `--db-instance-identifier`
- `--db-snapshot-identifier`

Dalam contoh ini, Anda membuat snapshot klaster DB yang bernama *`mydbsnapshot`* untuk klaster DB yang disebut *`mydbinstance`*.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-snapshot \
    --db-instance-identifier mydbinstance \
```

```
    --db-snapshot-identifier mydbsnapshot
```

Untuk Windows:

```
aws rds create-db-snapshot ^
    --db-instance-identifier mydbinstance ^
    --db-snapshot-identifier mydbsnapshot
```

RDS API

Saat membuat snapshot DB menggunakan Amazon RDS API, Anda perlu mengidentifikasi instans DB mana yang akan Anda cadangkan, kemudian beri nama snapshot DB sehingga Anda dapat memulihkannya nanti. Anda dapat melakukannya dengan menggunakan perintah Amazon RDS API `CreateDBSnapshot` dengan parameter berikut:

- `DBInstanceIdentifier`
- `DBSnapshotIdentifier`

Membuat snapshot klaster DB Multi-AZ

Saat Anda membuat snapshot klaster DB Multi-AZ, pastikan untuk mengidentifikasi klaster DB Multi-AZ mana yang akan Anda cadangkan, lalu beri nama snapshot klaster DB sehingga Anda dapat memulihkannya nanti. Anda juga dapat membagikan snapshot klaster DB Multi-AZ. Untuk mengetahui petunjuknya, lihat the section called “Berbagi snapshot DB”.

Anda dapat membuat snapshot cluster DB multi-AZ menggunakan AWS Management Console, AWS CLI, atau RDS API.

Konsol

Untuk membuat snapshot klaster DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data.
3. Dalam daftar, pilih klaster DB Multi-AZ yang ingin Anda ambil snapshotnya.
4. Untuk Tindakan, pilih Ambil snapshot.

   Jendela Ambil snapshot DB akan muncul.
5. Untuk Nama snapshot, masukkan nama snapshot.
6. Pilih Ambil snapshot.

Halaman Snapshot muncul, dengan status snapshot klaster DB Multi-AZ ditampilkan sebagai `Creating`. Setelah statusnya adalah `Available`, Anda dapat melihat waktu pembuatannya.

AWS CLI

Anda dapat membuat snapshot cluster DB multi-AZ dengan menggunakan AWS CLI create-db-cluster-snapshotperintah dengan opsi berikut:

- `--db-cluster-identifier`
- `--db-cluster-snapshot-identifier`

Dalam contoh ini, Anda membuat snapshot klaster DB Multi-AZ yang disebut *`mymultiazdbclustersnapshot`* untuk klaster DB yang disebut *`mymultiazdbcluster`*.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-cluster-snapshot \
    --db-cluster-identifier mymultiazdbcluster \
    --db-cluster-snapshot-identifier mymultiazdbclustersnapshot
```

Untuk Windows:

```
aws rds create-db-cluster-snapshot ^
    --db-cluster-identifier mymultiazdbcluster ^
    --db-cluster snapshot-identifier mymultiazdbclustersnapshot
```

API RDS

Anda dapat membuat snapshot cluster DB multi-AZ dengan menggunakan operasi Amazon RDS API CreateDB ClusterSnapshot dengan parameter berikut:

- `DBClusterIdentifier`
- `DBClusterSnapshotIdentifier`

Menghapus snapshot klaster DB Multi-AZ

Anda dapat menghapus snapshot DB Multi-AZ yang dikelola Amazon RDS saat Anda tidak lagi membutuhkannya. Untuk mengetahui petunjuknya, lihat the section called “Menghapus snapshot DB”.

Menghapus snapshot DB

Anda dapat menghapus snapshot DB yang dikelola Amazon RDS saat Anda tidak lagi membutuhkannya.

> **Note**
>
> Untuk menghapus pencadangan yang dikelola oleh AWS Backup, gunakan konsol AWS Backup. Untuk mengetahui informasi selengkapnya tentang AWS Backup, lihat Panduan Pengembang AWS Backup.

Menghapus snapshot DB

Anda dapat menghapus snapshot DB manual, bersama, atau publik menggunakan AWS Management Console, AWS CLI, atau RDS API.

Untuk menghapus snapshot bersama atau publik, Anda harus masuk ke akun AWS yang memiliki snapshot.

Jika Anda memiliki snapshot DB otomatis yang ingin Anda hapus tanpa menghapus instans DB, ubah periode retensi pencadangan untuk instans DB menjadi 0. Snapshot otomatis dihapus ketika perubahan diterapkan. Anda dapat segera menerapkan perubahan jika Anda tidak ingin menunggu hingga periode pemeliharaan berikutnya. Setelah perubahan selesai, Anda dapat mengaktifkan kembali pencadangan otomatis dengan mengatur periode retensi backup ke angka yang lebih besar dari 0. Untuk mengetahui informasi tentang cara mengubah instans DB, lihat Memodifikasi instans DB Amazon RDS.

Pencadangan otomatis yang dipertahankan dan snapshot manual dikenakan biaya penagihan hingga dihapus. Untuk informasi selengkapnya, lihat Biaya retensi.

Jika Anda menghapus instans DB, Anda dapat menghapus snapshot DB otomatis dengan menghapus pencadangan otomatis untuk instans DB tersebut. Untuk mengetahui informasi tentang pencadangan otomatis, lihat Pengantar cadangan.

Konsol

Menghapus snapshot DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.

2. Di panel navigasi, pilih Snapshot.

   Daftar Snapshot manual akan muncul.
3. Pilih snapshot DB yang ingin Anda hapus.
4. Di bawah Tindakan, pilih Hapus snapshot.
5. Pilih Hapus di halaman konfirmasi.

AWS CLI

Anda dapat menghapus snapshot DB dengan menggunakan AWS CLI perintah delete-db-snapshot.

Opsi berikut digunakan untuk menghapus snapshot DB.

- `--db-snapshot-identifier` – Pengidentifikasi untuk snapshot DB.

Example

Kode berikut menghapus snapshot DB `mydbsnapshot`.

Untuk Linux, macOS, atau Unix:

```
aws rds delete-db-snapshot \
    --db-snapshot-identifier mydbsnapshot
```

Untuk Windows:

```
aws rds delete-db-snapshot ^
    --db-snapshot-identifier mydbsnapshot
```

RDS API

Anda dapat menghapus snapshot DB dengan menggunakan operasi Amazon RDS API DeleteDBSnapshot.

Parameter berikut digunakan untuk menghapus snapshot DB.

- `DBSnapshotIdentifier` – Pengidentifikasi untuk snapshot DB.

Memulihkan dari snapshot DB

Bagian ini menunjukkan cara mengembalikan dari snapshot DB.

Topik

Amazon RDS membuat snapshot volume penyimpanan instans DB Anda, sehingga mencadangkan seluruh instans DB data dan bukan hanya masing-masing basis data. Anda dapat membuat instans DB baru dengan memulihkan dari snapshot DB. Anda memberikan nama snapshot DB untuk memulihkannya, kemudian memberikan nama untuk instans DB baru yang dibuat dari pemulihan tersebut. Anda tidak dapat memulihkan dari snapshot DB ke instans DB yang sudah ada; instans DB baru dibuat saat Anda memulihkan.

Anda dapat menggunakan instans DB yang dipulihkan segera setelah statusnya `available`. instans DB terus memuat data di latar belakang. Hal ini dikenal sebagai lazy loading.

Jika Anda mengakses data yang belum dimuat, instans DB segera mengunduh data yang diminta dari Amazon S3, lalu melanjutkan pemuatan sisa data di latar belakang. Untuk informasi selengkapnya, lihat snapshot Amazon EBS.

Untuk membantu mengurangi efek "lazy loading" pada tabel yang harus diakses dengan cepat, Anda dapat melakukan operasi yang mencakup pemindaian tabel lengkap, seperti `SELECT *`. Hal ini memungkinkan Amazon RDS mengunduh semua data tabel yang dicadangkan dari S3.

Anda dapat memulihkan instans DB dan menggunakan jenis penyimpanan yang berbeda dari snapshot DB sumber. Dalam kasus ini, proses pemulihan lebih lambat karena pekerjaan tambahan diperlukan untuk memigrasikan data ke jenis penyimpanan yang baru. Jika Anda memulihkan ke atau dari penyimpanan magnetik, proses migrasinya akan berjalan paling lambat. Hal tersebut karena penyimpanan magnetik tidak memiliki kemampuan IOPS dari penyimpanan IOPS yang Tersedia atau Tujuan Umum (SSD).

Anda dapat menggunakan AWS CloudFormation untuk memulihkan instans DB dari snapshot instans DB. Untuk informasi selengkapnya, lihat AWS::RDS::DBInstance dalam Panduan Pengguna AWS CloudFormation .

> **Note**
>
> Anda tidak dapat memulihkan instans DB dari snapshot DB yang dibagikan dan dienkripsi. Sebagai gantinya, Anda dapat membuat salinan snapshot DB dan memulihkan instans DB dari salinan tersebut. Untuk informasi selengkapnya, lihat Menyalin snapshot DB.

Untuk informasi tentang memulihkan instans DB dengan versi RDS Extended Support, lihat. Memulihkan instans DB atau cluster DB multi-AZ, cluster Support

Pertimbangan grup parameter

Kami menyarankan agar Anda mempertahankan grup parameter DB untuk snapshot DB apa pun yang Anda buat, sehingga Anda dapat mengaitkan instans DB yang dipulihkan dengan grup parameter yang benar.

Grup parameter DB default akan dikaitkan dengan instans yang dipulihkan, kecuali jika Anda memilih yang berbeda. Tidak ada pengaturan parameter kustom yang tersedia di grup parameter default.

Anda dapat menentukan grup parameter saat memulihkan instans DB.

Untuk informasi selengkapnya tentang grup parameter DB, lihat Bekerja dengan grup parameter.

Pertimbangan grup keamanan

Saat Anda memulihkan instans DB, cloud privat virtual (VPC), grup subnet DB, dan grup keamanan VPC default akan dikaitkan dengan instans yang dipulihkan, kecuali jika Anda memilih yang berbeda.

- Jika Anda menggunakan konsol Amazon RDS, Anda dapat menentukan grup keamanan VPC kustom yang akan dikaitkan dengan instans atau membuat grup keamanan VPC baru.
- Jika Anda menggunakan AWS CLI, Anda dapat menentukan grup keamanan VPC kustom untuk dikaitkan dengan instance dengan menyertakan `--vpc-security-group-ids` opsi dalam perintah. `restore-db-instance-from-db-snapshot`
- Jika Anda menggunakan API Amazon RDS, Anda dapat menyertakan `VpcSecurityGroupIds.VpcSecurityGroupId.N` di dalam tindakan `RestoreDBInstanceFromDBSnapshot`.

Segera setelah pemulihan selesai dan instans DB baru Anda tersedia, Anda juga dapat mengubah pengaturan VPC dengan memodifikasi instans DB. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Pertimbangan grup opsi

Saat Anda memulihkan instans DB, grup opsi DB default akan dikaitkan dengan instans DB yang dipulihkan dalam sebagian besar kasus.

Pengecualiannya adalah ketika instans DB sumber dikaitkan dengan grup opsi yang berisi opsi persisten atau permanen. Misalnya, jika instans DB sumber menggunakan Enkripsi Data Transparan (TDE) Oracle, instans DB yang dipulihkan harus menggunakan grup opsi yang memiliki opsi TDE.

Jika Anda memulihkan instans DB ke VPC yang berbeda, Anda harus melakukan salah satu hal berikut untuk menetapkan grup opsi DB:

- Tetapkan grup opsi default untuk grup VPC tersebut ke instans.
- Tetapkan grup opsi lain yang dikaitkan ke VPC tersebut.
- Buat grup opsi baru dan tetapkan ke instans DB. Dengan opsi persisten atau permanen, seperti TDE Oracle, Anda harus membuat grup opsi baru yang mencakup opsi persisten atau permanen.

Untuk informasi selengkapnya tentang grup opsi DB, lihat Menggunakan grup opsi.

Pertimbangan pemberian tag sumber daya

Saat Anda memulihkan instans DB dari snapshot DB, RDS akan memeriksa apakah Anda menentukan tag baru. Jika ya, tag baru akan ditambahkan ke instans DB yang dipulihkan. Jika tidak ada tag baru, RDS akan menambahkan tag dari instans DB sumber pada saat pembuatan snapshot ke instans DB yang dipulihkan.

Untuk informasi selengkapnya, lihat Menyalin tag ke snapshot instans DB.

Pertimbangan Db2

Dengan model BYOL, instans DB RDS for Db2 Anda harus dikaitkan dengan grup parameter kustom yang berisi IBM Site ID dan IBM Customer ID Anda. Jika tidak, upaya memulihkan instans DB dari snapshot akan gagal. Lihat informasi yang lebih lengkap di Bawa Lisensi Sendiri dan rdsadmin.restore_database.

Pertimbangan Microsoft SQL Server

Saat memulihkan snapshot DB RDS for Microsoft SQL Server ke instans baru, Anda selalu dapat memulihkan ke edisi yang sama seperti snapshot Anda. Dalam beberapa kasus, Anda juga dapat mengubah edisi instans DB. Batasan berikut berlaku saat Anda mengubah edisi:

- Snapshot DB harus memiliki cukup penyimpanan yang dialokasikan untuk edisi baru.
- Hanya perubahan edisi berikut yang didukung:
  - Dari Standard Edition ke Enterprise Edition
  - Dari Web Edition ke Standard Edition atau Enterprise Edition
  - Dari Express Edition ke Web Edition, Standard Edition, atau Enterprise Edition

Jika Anda ingin mengubah dari satu edisi ke edisi baru yang tidak didukung dengan memulihkan snapshot, Anda dapat mencoba menggunakan fitur pencadangan dan pemulihan native. SQL Server akan memverifikasi apakah basis data Anda kompatibel dengan edisi baru berdasarkan fitur SQL Server yang telah Anda aktifkan di basis data tersebut. Untuk informasi selengkapnya, lihat Mengimpor dan mengekspor basis data SQL Server menggunakan pencadangan dan pemulihan native.

Pertimbangan Oracle Database

Saat Anda memulihkan basis data Oracle dari snapshot DB, pertimbangkan hal berikut:

- Sebelum Anda memulihkan snapshot DB, Anda dapat meng-upgrade-nya ke rilis basis data Oracle yang lebih baru. Untuk informasi selengkapnya, lihat Meng-upgrade snapshot DB Oracle.
- Jika Anda memulihkan snapshot dari instans CDB yang menggunakan konfigurasi penghuni tunggal, Anda dapat mengubah nama PDB. Anda tidak dapat mengubah nama PDB saat instans CDB Anda menggunakan konfigurasi multi-penghuni. Untuk informasi selengkapnya, lihat Mencadangkan dan memulihkan CDB.
- Anda tidak dapat mengubah nama CDB, yaitu `RDSCDB`. Nama CDB ini sama untuk semua instans CDB.
- Anda tidak dapat langsung berinteraksi dengan basis data penghuni dalam snapshot DB. Jika Anda memulihkan snapshot dari instans CDB yang menggunakan konfigurasi multi-penghuni, Anda akan memulihkan semua basis data penghuni. Anda dapat menggunakan describe-db-snapshot-tenant-databases untuk memeriksa database penyewa dalam snapshot DB sebelum memulihkannya.
- Jika Anda menggunakan Oracle GoldenGate, selalu pertahankan grup parameter dengan `compatible` parameter. Saat memulihkan instans DB dari snapshot DB, Anda harus menentukan grup parameter yang memiliki nilai `compatible` yang sama atau lebih besar.
- Anda dapat memilih untuk mengubah nama basis data Anda ketika Anda memulihkan snapshot DB. Jika ukuran total log redo online lebih besar dari 20GB, RDS mungkin mengatur ulang ukuran log redo online Anda ke pengaturan default 512MB (4 x 128MB). Ukuran yang lebih kecil memungkinkan operasi pemulihan selesai dalam waktu yang wajar. Anda dapat membuat ulang log redo online nanti dan mengubah ukurannya.

Memulihkan dari snapshot

Anda dapat memulihkan instans DB dari snapshot DB menggunakan AWS Management Console, AWS CLI, atau RDS API.

> **Note**
>
> Anda tidak dapat mengurangi jumlah penyimpanan saat Anda memulihkan instans DB. Saat Anda meningkatkan alokasi penyimpanan, peningkatannya setidaknya harus 10 persen. Jika Anda mencoba meningkatkan nilai sebesar kurang dari 10 persen, Anda akan mendapat kesalahan. Anda tidak dapat meningkatkan penyimpanan yang dialokasikan saat memulihkan RDS untuk instans DB SQL Server.

Konsol

Untuk memulihkan instans DB dari snapshot DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Snapshot.
3. Pilih snapshot DB yang ingin Anda pulihkan.
4. Untuk Tindakan, pilih Pulihkan snapshot.
5. Di halaman Pulihkan snapshot, untuk Pengidentifikasi instans DB, masukkan nama untuk instans DB yang dipulihkan.
6. Tentukan pengaturan lain, seperti ukuran penyimpanan yang dialokasikan.

   Untuk informasi tentang setiap pengaturan, lihat Pengaturan untuk instans DB.
7. Pilih Pulihkan instans DB.

AWS CLI

Untuk mengembalikan instans DB dari snapshot DB, gunakan AWS CLI perintah restore-db-instance-from-db-snapshot.

Dalam contoh ini, Anda memulihkan dari snapshot DB yang dibuat sebelumnya yang bernama `mydbsnapshot`. Anda memulihkan ke instans DB baru yang bernama `mynewdbinstance`. Contoh ini juga menetapkan ukuran penyimpanan yang dialokasikan.

Anda dapat menentukan pengaturan lain. Untuk informasi tentang setiap pengaturan, lihat Pengaturan untuk instans DB.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds restore-db-instance-from-db-snapshot \
    --db-instance-identifier mynewdbinstance \
    --db-snapshot-identifier mydbsnapshot \
    --allocated-storage 100
```

Untuk Windows:

```
aws rds restore-db-instance-from-db-snapshot ^
    --db-instance-identifier mynewdbinstance ^
    --db-snapshot-identifier mydbsnapshot ^
    --allocated-storage 100
```

Perintah ini menampilkan output seperti yang berikut ini:

```
DBINSTANCE  mynewdbinstance  db.t3.small  MySQL     50       sa              creating
 3  n  8.0.28  general-public-license
```

API RDS

Untuk memulihkan instans DB dari snapshot DB, panggil fungsi Amazon RDS API InstanceFromRestoreDB DBSnapshot dengan parameter berikut:

- `DBInstanceIdentifier`
- `DBSnapshotIdentifier`

Memulihkan instans DB dengan waktu yang ditentukan

Anda dapat memulihkan instans DB ke titik waktu tertentu, yang membuat instans DB baru.

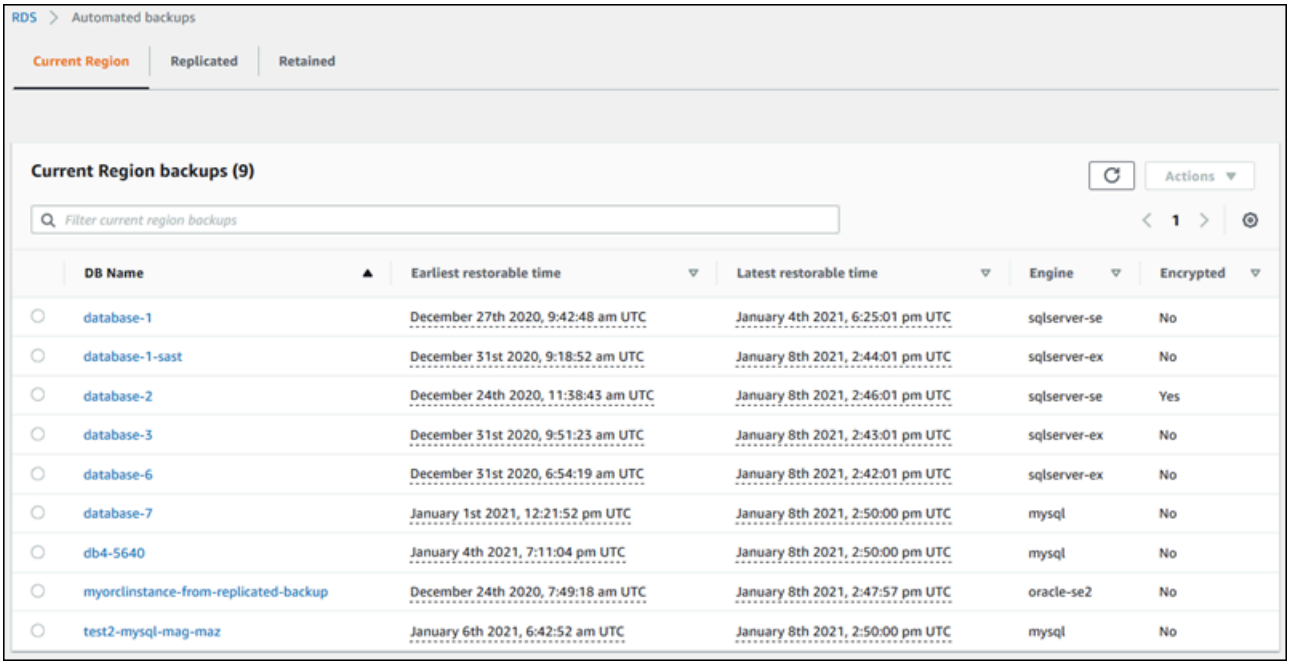
Saat Anda memulihkan instans DB ke titik waktu tertentu, Anda dapat memilih grup keamanan cloud privat virtual (VPC) default. Atau Anda dapat menerapkan grup keamanan VPC kustom ke instans DB Anda.

Instans DB yang dipulihkan secara otomatis dikaitkan dengan parameter DB default dan grup opsi. Namun, Anda dapat menerapkan grup parameter kustom dan grup opsi dengan menentukannya selama pemulihan.

Jika instans DB sumber memiliki tag sumber daya, RDS menambahkan tag terbaru ke instans DB yang dipulihkan.

RDS mengunggah log transaksi untuk instans DB ke Amazon S3 setiap lima menit. Untuk melihat waktu restorable terbaru untuk instans DB, gunakan AWS CLI describe-db-instancesperintah dan lihat nilai yang dikembalikan di `LatestRestorableTime` bidang untuk instans DB. Untuk melihat waktu pemulihan terbaru setiap instans DB di konsol Amazon RDS, pilih Cadangan otomatis.

Anda dapat memulihkan ke titik waktu mana pun dalam periode retensi cadangan Anda. Untuk melihat waktu pemulihan terbaru setiap instans DB, pilih Cadangan otomatis di konsol Amazon RDS.



> **Note**
>
> Sebaiknya Anda memulihkan ke ukuran instans DB yang sama atau serupa—dan IOPS jika menggunakan penyimpanan Provisioned IOPS—sebagai instans DB sumber. Anda mungkin mengalami kesalahan jika, misalnya, Anda memilih ukuran instans DB dengan nilai IOPS yang tidak kompatibel.

Untuk informasi tentang memulihkan instans DB dengan versi RDS Extended Support, lihat. Memulihkan instans DB atau cluster DB multi-AZ, cluster Support

Beberapa mesin basis data yang digunakan oleh Amazon RDS memiliki pertimbangan khusus ketika memulihkan dari satu titik waktu:

- Jika Anda menggunakan autentikasi kata sandi dengan RDS untuk instans Db2 DB, tindakan manajemen pengguna, termasuk `rdsadmin.add_user`, tidak akan ditangkap dalam log. Tindakan ini memerlukan cadangan snapshot lengkap.

  Dengan model BYOL, instans RDS untuk Db2 DB Anda harus dikaitkan dengan grup parameter kustom yang berisi IBM Site ID dan IBM Customer ID Anda. Jika tidak, upaya untuk memulihkan instans DB ke titik waktu tertentu akan gagal. Lihat informasi yang lebih lengkap di Bawa Lisensi Sendiri dan rdsadmin.restore_database.
- Saat Anda memulihkan instans DB Oracle ke suatu titik waktu, Anda dapat menentukan mesin DB Oracle, model lisensi, dan DBName (SID) untuk digunakan oleh instans DB baru.
- Saat Anda memulihkan instans DB Microsoft SQL Server ke suatu titik waktu, setiap basis data dalam instans tersebut disimpan ke suatu titik waktu dalam 1 detik dari basis data satu sama lain dalam instans tersebut. Transaksi yang menjangkau beberapa basis data dalam instans tersebut mungkin dipulihkan secara tidak konsisten.
- Untuk instans DB SQL Server, mode `OFFLINE`, `EMERGENCY`, dan `SINGLE_USER` tidak didukung. Mengatur basis data apa pun ke salah satu mode ini akan menyebabkan waktu pemulihan terakhir untuk berhenti berjalan untuk seluruh instans.
- Beberapa tindakan, seperti mengubah model pemulihan database SQL Server, dapat merusak urutan log yang digunakan untuk point-in-time pemulihan. Dalam beberapa kasus, Amazon RDS dapat mendeteksi masalah ini dan waktu pemulihan terbaru dicegah untuk berjalan. Dalam kasus lain, seperti ketika basis data SQL Server menggunakan model pemulihan `BULK_LOGGED`, jeda dalam urutan log tidak terdeteksi. Ada kemungkinan bahwa pemulihan instans DB SQL Server ke

titik waktu mustahil dilakukan jika terdapat jeda dalam urutan log. Karena alasan ini, Amazon RDS tidak mendukung perubahan model pemulihan basis data SQL Server.

Anda juga dapat menggunakan AWS Backup untuk mengelola cadangan instans Amazon RDS DB. Jika instans DB Anda dikaitkan dengan rencana cadangan di AWS Backup, paket cadangan itu digunakan untuk point-in-time pemulihan. Cadangan yang dibuat dengan AWS Backup memiliki nama yang diakhiri dengan. `awsbackup:`*`AWS-Backup-job-number`* Untuk selengkapnya AWS Backup, lihat Panduan AWS Backup Pengembang.

> **Note**
>
> Informasi dalam topik ini berlaku untuk Amazon RDS. Untuk informasi tentang memulihkan klaster DB Amazon Aurora, lihat Memulihkan klaster DB ke waktu tertentu.

Anda dapat mengembalikan instans DB ke titik waktu menggunakan AWS Management Console, AWS CLI, atau RDS API.

> **Note**
>
> Anda tidak dapat mengurangi jumlah penyimpanan saat Anda memulihkan instans DB. Saat Anda meningkatkan alokasi penyimpanan, peningkatannya setidaknya harus 10 persen. Jika Anda mencoba meningkatkan nilai sebesar kurang dari 10 persen, Anda akan mendapat kesalahan. Anda tidak dapat meningkatkan penyimpanan yang dialokasikan saat memulihkan RDS untuk instans DB SQL Server.

Konsol

Untuk memulihkan instans DB dengan waktu yang ditentukan

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Pencadangan otomatis.

   Pencadangan otomatis ditampilkan di tab Wilayah Saat Ini.
3. Pilih instans DB yang ingin Anda pulihkan.
4. Untuk Tindakan, pilih Pulihkan ke titik waktu.

Jendela Pulihkan ke titik waktu akan muncul.

5. Pilih Waktu pemulihan terbaru untuk memulihkan ke waktu terbaru yang dimungkinkan atau pilih Kustom untuk memilih waktu.

   Jika Anda memilih Kustom, masukkan tanggal dan waktu untuk memulihkan instans.

   > **Note**
   >
   > Waktu ditampilkan dalam zona waktu lokal Anda, yang ditunjukkan dengan offset dari Waktu Universal Terkoordinasi (UTC). Misalnya, UTC-5 adalah Waktu Standar Timur/ Waktu Musim Panas Tengah.

6. Untuk Pengidentifikasi instans DB, masukkan nama target instans DB yang dipulihkan. Nama harus unik.
7. Pilih opsi lain sesuai kebutuhan, seperti kelas instans DB, penyimpanan, dan apakah Anda ingin menggunakan penskalaan otomatis penyimpanan atau tidak.

   Untuk informasi tentang setiap pengaturan, lihat Pengaturan untuk instans DB.
8. Pilih Pulihkan ke titik waktu.

AWS CLI

Untuk mengembalikan instance DB ke waktu tertentu, gunakan AWS CLI perintah restore-db-instance-to- point-in-time untuk membuat instance DB baru. Contoh ini juga menetapkan ukuran penyimpanan yang dialokasikan dan memungkinkan penyimpanan penskalaan otomatis.

Pemberian tag sumber daya didukung untuk operasi ini. Saat Anda menggunakan opsi `--tags`, tag instans DB sumber diabaikan dan tag yang disediakan digunakan. Jika tidak, tag terbaru dari instans sumber digunakan.

Anda dapat menentukan pengaturan lain. Untuk informasi tentang setiap pengaturan, lihat Pengaturan untuk instans DB.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds restore-db-instance-to-point-in-time \
```

```
    --source-db-instance-identifier mysourcedbinstance \
    --target-db-instance-identifier mytargetdbinstance \
    --restore-time 2017-10-14T23:45:00.000Z \
    --allocated-storage 100 \
    --max-allocated-storage 1000
```

Untuk Windows:

```
aws rds restore-db-instance-to-point-in-time ^
    --source-db-instance-identifier mysourcedbinstance ^
    --target-db-instance-identifier mytargetdbinstance ^
    --restore-time 2017-10-14T23:45:00.000Z ^
    --allocated-storage 100 ^
    --max-allocated-storage 1000
```

API RDS

Untuk memulihkan instans DB ke waktu yang ditentukan, panggil operasi API Amazon RDS `RestoreDBInstanceToPointInTime` dengan parameter berikut ini:

- `SourceDBInstanceIdentifier`
- `TargetDBInstanceIdentifier`
- `RestoreTime`

Memulihkan klaster DB Multi-AZ ke waktu tertentu

Anda dapat memulihkan klaster DB Multi-AZ ke titik waktu tertentu, membuat klaster DB Multi-AZ baru.

RDS mengunggah log transaksi untuk klaster DB Multi-AZ ke Amazon S3 terus menerus. Anda dapat memulihkan ke titik waktu mana pun dalam periode retensi cadangan Anda. Untuk melihat waktu restorable paling awal untuk cluster DB multi-AZ, gunakan perintah. AWS CLI describe-db-clusters Lihatlah nilai yang ditampilkan di kolom `EarliestRestorableTime` untuk klaster DB. Untuk melihat waktu pemulihan paling awal untuk klaster DB Multi-AZ, lihat nilai yang ditampilkan dalam kolom `LatestRestorableTime` untuk klaster DB Multi-AZ.

Saat memulihkan cluster DB multi-AZ ke titik waktu tertentu, Anda dapat memilih grup keamanan VPC default untuk cluster DB multi-AZ Anda, atau Anda dapat menerapkan grup keamanan VPC khusus ke cluster DB multi-AZ Anda.

Klaster DB Multi-AZ yang dipulihkan secara otomatis dikaitkan dengan grup parameter klaster DB default. Namun, Anda dapat menerapkan grup parameter cluster DB kustom dengan menentukannya selama pemulihan.

Jika cluster DB sumber memiliki tag sumber daya, RDS menambahkan tag terbaru ke cluster DB yang dipulihkan.

> **Note**
>
> Sebaiknya Anda memulihkan ke ukuran klaster DB Multi-AZ sebagai klaster DB Multi-AZ. Sebaiknya Anda juga memulihkan dengan nilai IOPS yang sama atau serupa jika Anda menggunakan penyimpanan IOPS yang Tersedia. Anda mungkin mengalami kesalahan jika, misalnya, Anda memilih ukuran klaster DB dengan nilai IOPS yang tidak kompatibel.
> Jika cluster DB Multi-AZ sumber menggunakan penyimpanan General Purpose SSD (gp3) dan memiliki kurang dari 400 GiB penyimpanan yang dialokasikan, Anda tidak dapat memodifikasi IOPS yang disediakan untuk cluster DB yang dipulihkan.

Untuk informasi tentang memulihkan cluster DB multi-AZ dengan versi RDS Extended Support, lihat. Memulihkan instans DB atau cluster DB multi-AZ, cluster Support

Anda dapat memulihkan cluster DB multi-AZ ke titik waktu menggunakan AWS Management Console, API AWS CLI, atau RDS.

Konsol

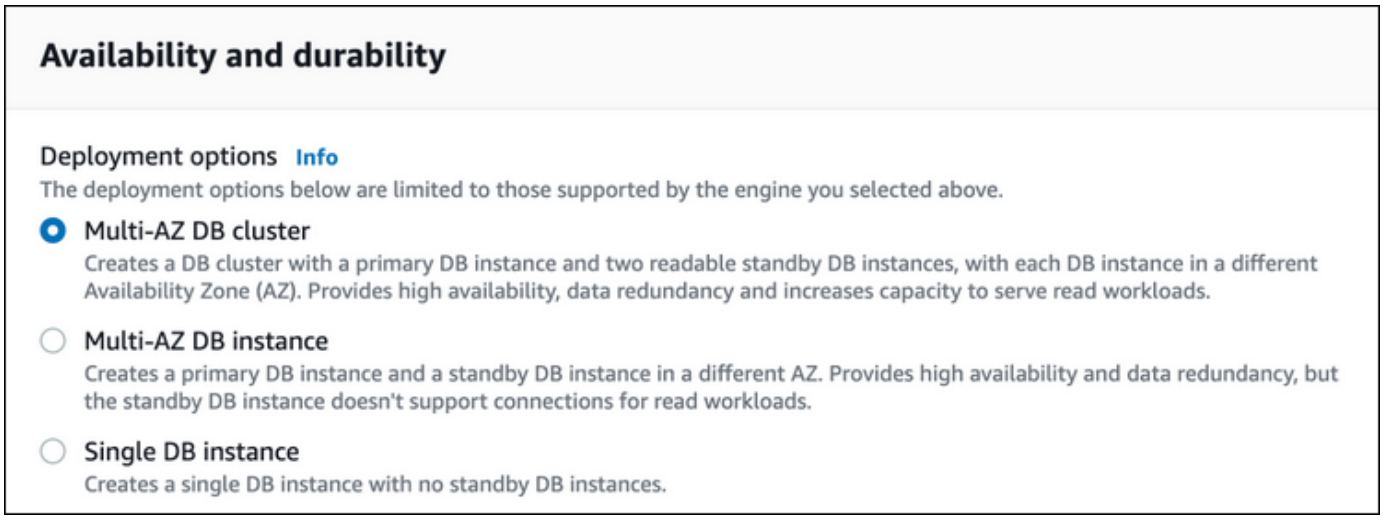
Untuk memulihkan klaster DB Multi-AZ ke waktu tertentu

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data.
3. Pilih klaster basis data Multi-AZ yang ingin Anda pulihkan.
4. Untuk Tindakan, pilih Pulihkan ke titik waktu.

   Jendela Pulihkan ke titik waktu akan muncul.
5. Pilih Waktu pemulihan terbaru untuk memulihkan ke waktu terbaru yang dimungkinkan atau pilih Kustom untuk memilih waktu.

   Jika Anda memilih Kustom, masukkan tanggal dan waktu untuk memulihkan klaster DB Multi-AZ.

   > **Note**
   >
   > Waktu ditampilkan dalam zona waktu lokal Anda, yang ditunjukkan dengan offset dari Waktu Universal Terkoordinasi (UTC). Misalnya, UTC-5 adalah Waktu Standar Timur/Waktu Musim Panas Tengah.
6. Untuk Pengidentifikasi klaster DB, masukkan nama klaster DB Multi-AZ Anda yang dipulihkan.
7. Dalam Ketersediaan dan daya tahan, pilih Klaster DB Multi-AZ.



8. Di Kelas instans DB, pilih kelas instans DB.

Saat ini, klaster basis data Multi-AZ hanya mendukung kelas-kelas instans basis data db.m6gd dan db.r6gd. Lihat informasi selengkapnya tentang kelas instans basis data di Kelas instans DB .

9. Untuk bagian yang lainnya, tentukan pengaturan klaster DB Anda. Untuk informasi tentang setiap pengaturan, lihat Pengaturan untuk membuat klaster DB Multi-AZ.
10. Pilih Pulihkan ke titik waktu.

AWS CLI

Untuk mengembalikan cluster DB multi-AZ ke waktu yang ditentukan, gunakan AWS CLI perintah restore-db-cluster-to- point-in-time untuk membuat cluster DB multi-AZ baru.

Saat ini, klaster basis data Multi-AZ hanya mendukung kelas-kelas instans basis data db.m6gd dan db.r6gd. Lihat informasi selengkapnya tentang kelas instans basis data di Kelas instans DB .

Example

Untuk Linux, macOS, atau Unix:

```
aws rds restore-db-cluster-to-point-in-time \
    --source-db-cluster-identifier mysourcemultiazdbcluster \
    --db-cluster-identifier mytargetmultiazdbcluster \
    --restore-to-time 2021-08-14T23:45:00.000Z \
    --db-cluster-instance-class db.r6gd.xlarge
```

Untuk Windows:

```
aws rds restore-db-cluster-to-point-in-time ^
    --source-db-cluster-identifier mysourcemultiazdbcluster ^
    --db-cluster-identifier mytargetmultiazdbcluster ^
    --restore-to-time 2021-08-14T23:45:00.000Z ^
    --db-cluster-instance-class db.r6gd.xlarge
```

API RDS

Untuk memulihkan cluster DB ke waktu yang ditentukan, panggil ClusterToPointInTime operasi Amazon RDS API RestoreDB dengan parameter berikut:

- `SourceDBClusterIdentifier`
- `DBClusterIdentifier`

- `RestoreToTime`

Memulihkan dari snapshot ke klaster DB Multi-AZ

Anda dapat mengembalikan snapshot ke cluster DB multi-AZ menggunakan AWS Management Console, AWS CLI, atau RDS API. Anda dapat memulihkan masing-masing jenis snapshot ini ke klaster DB Multi-AZ:

- Snapshot deployment AZ Tunggal
- Cuplikan penerapan cluster DB multi-AZ dengan satu instans DB
- Snapshot klaster DB Multi-AZ

Untuk informasi tentang deployment Multi-AZ, lihat Mengonfigurasi dan mengelola deployment Multi-AZ.

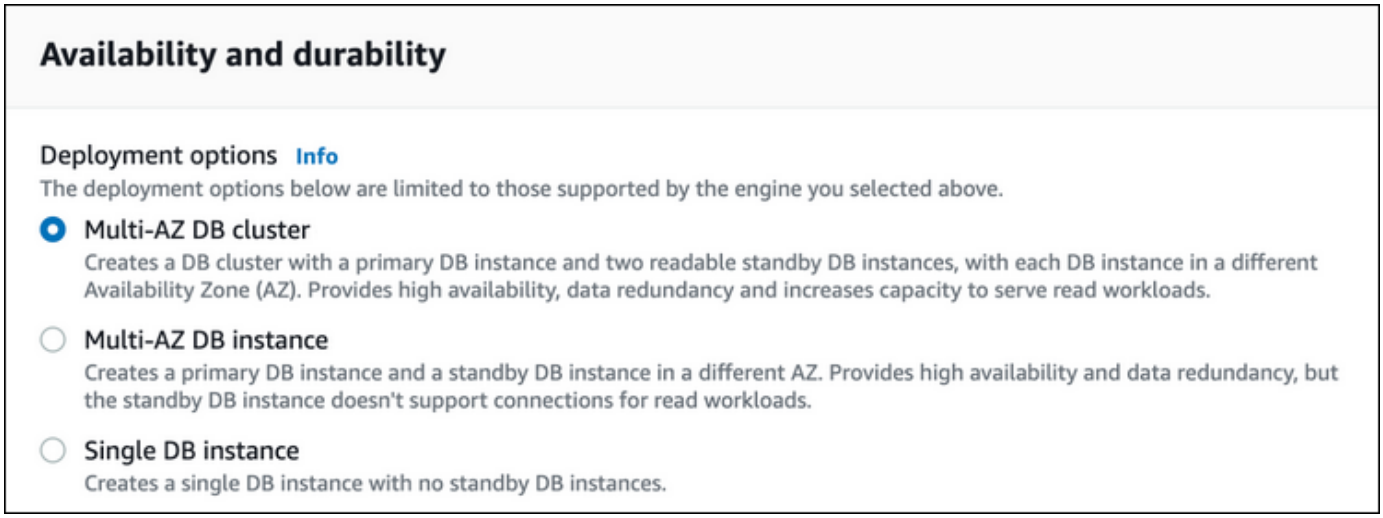
> **Tip**
>
> Anda dapat memigrasikan penerapan AZ tunggal atau penerapan klaster DB multi-AZ ke penerapan klaster DB multi-AZ dengan memulihkan snapshot.

Untuk informasi tentang memulihkan klaster DB multi-AZ dengan versi RDS Extended Support, lihat. Memulihkan instans DB atau cluster DB multi-AZ, cluster Support

Konsol

Untuk memulihkan snapshot ke klaster DB Multi-AZ

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Snapshot.
3. Pilih snapshot yang ingin Anda pulihkan.
4. Untuk Tindakan, pilih Pulihkan snapshot.
5. Pada halaman Pulihkan snapshot, dalam Ketersediaan dan daya tahan, pilih Klaster DB Multi-AZ.



6. Untuk Pengidentifikasi klaster DB, masukkan nama untuk klaster DB Multi-AZ.
7. Untuk bagian yang tersisa, tentukan pengaturan klaster DB Anda. Untuk informasi tentang setiap pengaturan, lihat Pengaturan untuk membuat klaster DB Multi-AZ.
8. Pilih Pulihkan instans DB.

AWS CLI

Untuk mengembalikan snapshot ke cluster DB multi-AZ, gunakan AWS CLI perintah restore-db-cluster-from -snapshot.

Dalam contoh berikut, Anda memulihkan dari snapshot yang dibuat sebelumnya bernama `mysnapshot`. Anda memulihkan ke klaster DB baru yang bernama `mynewmultiazdbcluster`. Anda juga perlu menentukan kelas instans DB yang digunakan oleh instans DB di klaster DB Multi-AZ. Tentukan `mysql` atau `postgres` untuk mesin DB.

Untuk opsi `--snapshot-identifier`, Anda dapat menggunakan nama atau Amazon Resource Name (ARN) untuk menentukan snapshot klaster DB. Namun, Anda dapat menggunakan ARN saja untuk menentukan snapshot DB.

Untuk opsi `--db-cluster-instance-class`, tentukan kelas instans DB untuk klaster DB Multi-AZ baru. Klaster DB Multi-AZ hanya mendukung kelas instans DB tertentu, seperti kelas instans DB db.m6gd dan db.r6gd. Untuk informasi selengkapnya tentang kelas instans DB, lihat Kelas instans DB .

Anda juga dapat menentukan opsi lainnya.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds restore-db-cluster-from-snapshot \
    --db-cluster-identifier mynewmultiazdbcluster \
    --snapshot-identifier mysnapshot \
    --engine mysql|postgres \
    --db-cluster-instance-class db.r6gd.xlarge
```

Untuk Windows:

```
aws rds restore-db-cluster-from-snapshot ^
    --db-cluster-identifier mynewmultiazdbcluster ^
    --snapshot-identifier mysnapshot ^
    --engine mysql|postgres ^
    --db-cluster-instance-class db.r6gd.xlarge
```

Setelah memulihkan klaster DB, Anda dapat menambahkan klaster DB Multi-AZ ke grup keamanan yang terkait dengan klaster DB atau instans DB yang Anda gunakan untuk membuat snapshot, jika berlaku. Setelah menyelesaikan tindakan ini, fungsi yang sama dari klaster DB atau instans DB sebelumnya akan tersedia.

API RDS

Untuk mengembalikan snapshot ke cluster DB multi-AZ, panggil operasi RDS API ClusterFromSnapshotRestoreDB dengan parameter berikut:

- `DBClusterIdentifier`
- `SnapshotIdentifier`
- `Engine`

Anda juga dapat menentukan parameter opsional lainnya.

Setelah memulihkan klaster DB, Anda dapat menambahkan klaster DB Multi-AZ ke grup keamanan yang terkait dengan klaster DB atau instans DB yang Anda gunakan untuk membuat snapshot, jika berlaku. Setelah menyelesaikan tindakan ini, fungsi yang sama dari klaster DB atau instans DB sebelumnya akan tersedia.

Memulihkan dari snapshot cluster DB multi-AZ ke instans DB AZ tunggal

Snapshot klaster DB Multi-AZ adalah snapshot volume penyimpanan klaster DB Anda, sehingga mencadangkan seluruh klaster DB, bukan hanya basis data individual. Anda dapat memulihkan snapshot klaster DB Multi-AZ ke deployment AZ Tunggal atau deployment instans DB Multi-AZ. Untuk informasi tentang deployment Multi-AZ, lihat Mengonfigurasi dan mengelola deployment Multi-AZ.

> **Note**
>
> Anda juga dapat memulihkan snapshot klaster DB Multi-AZ ke klaster DB Multi-AZ baru. Untuk petunjuk, lihat Memulihkan dari snapshot ke klaster DB Multi-AZ.

Untuk informasi tentang memulihkan klaster DB multi-AZ dengan versi RDS Extended Support, lihat. Memulihkan instans DB atau cluster DB multi-AZ, cluster Support

Gunakan, API AWS Management Console AWS CLI, atau RDS API untuk memulihkan snapshot cluster DB multi-AZ ke penerapan AZ tunggal atau penerapan instans DB multi-AZ.

Konsol

Untuk memulihkan snapshot klaster DB Multi-AZ ke deployment AZ Tunggal atau deployment instans DB Multi-AZ

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Snapshot.
3. Pilih snapshot klaster DB Multi-AZ yang ingin Anda pulihkan.
4. Untuk Tindakan, pilih Pulihkan snapshot.
5. Pada halaman Pulihkan snapshot, dalam Ketersediaan dan daya tahan, pilih salah satu dari yang berikut ini:
   - Instans DB tunggal – Memulihkan snapshot ke satu instans DB tanpa instans DB siaga.
   - Instans DB Multi-AZ – Memulihkan snapshot ke deployment instans DB Multi-AZ dengan satu instans DB primer dan satu instans DB siaga.
6. Untuk Pengidentifikasi instans DB, masukkan nama untuk instans DB yang dipulihkan.

7. Untuk bagian yang tersisa, tentukan pengaturan instans DB Anda. Untuk informasi tentang setiap pengaturan, lihat Pengaturan untuk instans DB.
8. Pilih Pulihkan instans DB.

AWS CLI

Untuk mengembalikan snapshot cluster DB multi-AZ ke penerapan instans DB, gunakan perintah -db-snapshot. AWS CLI restore-db-instance-from

Dalam contoh berikut, Anda memulihkan dari snapshot klaster DB Multi-AZ yang dibuat sebelumnya bernama `myclustersnapshot`. Anda memulihkan ke deployment instans DB Multi-AZ baru dengan instans DB primer bernama `mynewdbinstance`. Untuk opsi `--db-cluster-snapshot-identifier`, tentukan nama snapshot klaster Multi-AZ DB.

Untuk opsi `--db-instance-class`, tentukan kelas instans DB untuk deployment instans DB baru. Untuk informasi selengkapnya tentang kelas instans DB, lihat Kelas instans DB .

Anda juga dapat menentukan opsi lainnya.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds restore-db-instance-from-db-snapshot \
    --db-instance-identifier mynewdbinstance \
    --db-cluster-snapshot-identifier myclustersnapshot \
    --engine mysql \
    --multi-az \
    --db-instance-class db.r6g.xlarge
```

Untuk Windows:

```
aws rds restore-db-instance-from-db-snapshot ^
    --db-instance-identifier mynewdbinstance ^
    --db-cluster-snapshot-identifier myclustersnapshot ^
    --engine mysql ^
    --multi-az ^
    --db-instance-class db.r6g.xlarge
```

Setelah memulihkan instans DB, Anda dapat menambahkannya ke grup keamanan yang terkait dengan klaster DB Multi-AZ yang Anda gunakan untuk membuat snapshot, jika berlaku. Setelah menyelesaikan tindakan ini, fungsi yang sama dari klaster DB Multi-AZ sebelumnya akan tersedia.

API RDS

Untuk mengembalikan snapshot cluster DB multi-AZ ke penerapan instans DB, panggil operasi RDS API RestoreDB InstanceFrom DBSnapshot dengan parameter berikut:

- `DBInstanceIdentifier`
- `DBClusterSnapshotIdentifier`
- `Engine`

Anda juga dapat menentukan parameter opsional lainnya.

Setelah memulihkan instans DB, Anda dapat menambahkannya ke grup keamanan yang terkait dengan klaster DB Multi-AZ yang Anda gunakan untuk membuat snapshot, jika berlaku. Setelah menyelesaikan tindakan ini, fungsi yang sama dari klaster DB Multi-AZ sebelumnya akan tersedia.

Tutorial: Memulihkan instans DB Amazon RDS dari snapshot DB

Sering kali, saat menggunakan Amazon RDS, Anda mungkin memiliki instans DB yang hanya digunakan sesekali tetapi tidak memerlukan waktu penuh. Misalnya, Anda memiliki survei pelanggan per kuartal yang menggunakan instans Amazon EC2 untuk meng-host situs web survei pelanggan. Anda juga memiliki instans DB yang digunakan untuk menyimpan hasil survei. Salah satu cara untuk menghemat biaya pada skenario tersebut adalah dengan mengambil snapshot DB dari instans DB setelah survei selesai. Anda kemudian dapat menghapus instans DB dan memulihkannya saat perlu melakukan survei lagi.

Saat memulihkan instans DB, masukkan nama snapshot DB yang akan digunakan untuk memulihkan. Kemudian, masukkan nama untuk instans DB baru yang dibuat dari operasi pemulihan.

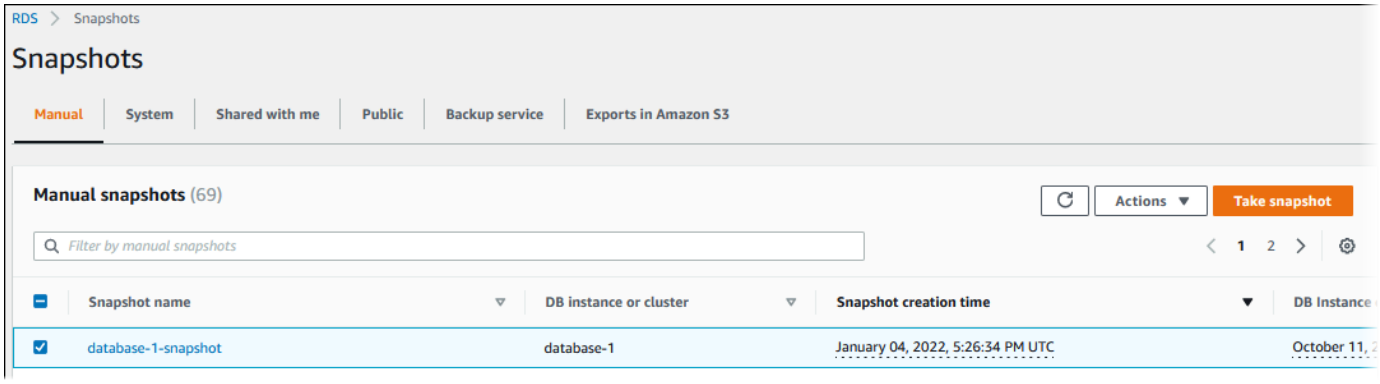
Untuk informasi lebih mendetail tentang memulihkan instans DB dari snapshot, lihat Memulihkan dari snapshot DB.

Memulihkan instans DB dari snapshot DB

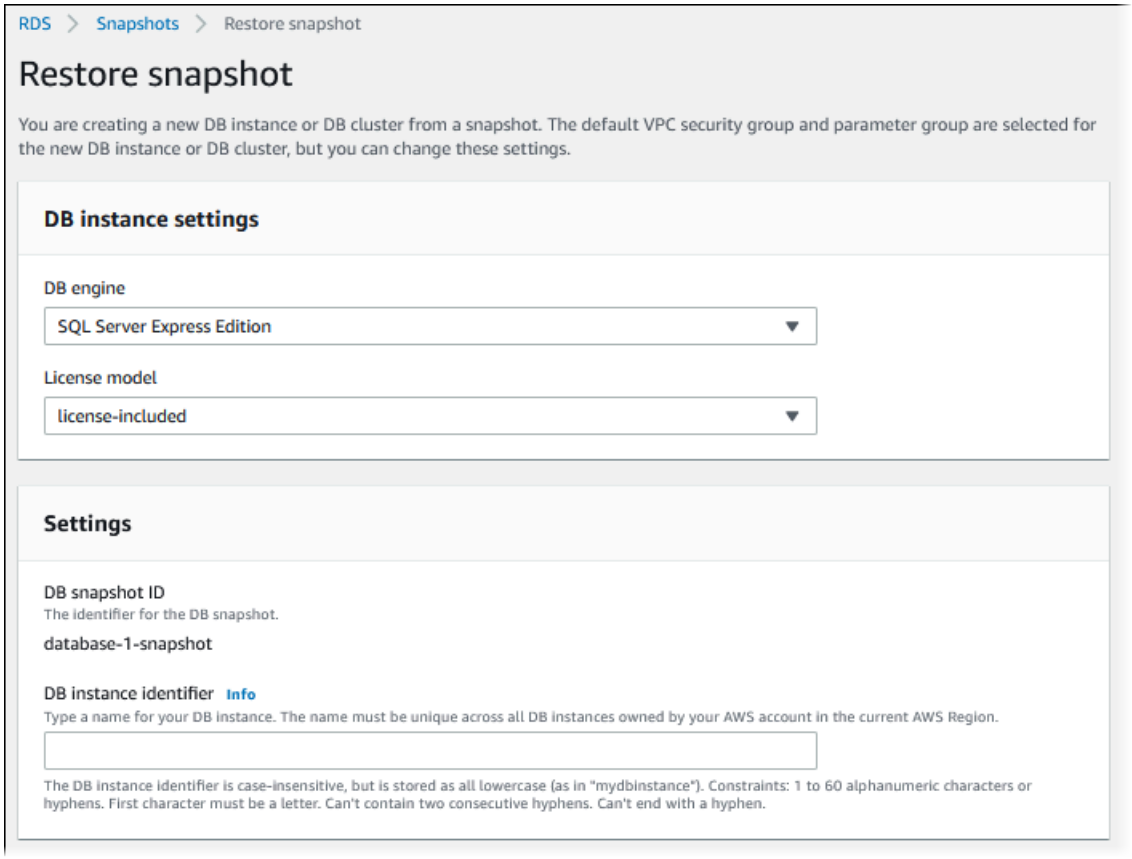
Gunakan prosedur berikut untuk memulihkan dari snapshot di AWS Management Console.

Memulihkan instans DB dari snapshot DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Snapshot.
3. Pilih snapshot DB yang ingin dipulihkan.
4. Untuk Tindakan, pilih Pulihkan snapshot.



Halaman Pulihkan snapshot ditampilkan.



5. Di bagian pengaturan instans DB, gunakan pengaturan default untuk mesin DB dan model Lisensi (untuk Oracle atau Microsoft SQL Server).
6. Di bagian Pengaturan, untuk pengidentifikasi instans DB masukkan nama unik yang ingin Anda gunakan untuk instans DB yang dipulihkan, misalnya **`mynewdbinstance`**.

   Jika memulihkan dari instans DB yang Anda hapus setelah membuat snapshot DB, Anda dapat menggunakan nama instans DB tersebut.
7. Di bagian Ketersediaan & daya tahan, pilih apakah akan membuat instans siaga di Zona Ketersediaan lain.

   Untuk tutorial ini, jangan membuat instans siaga.
8. Di bagian Konektivitas, gunakan pengaturan default untuk berikut ini:

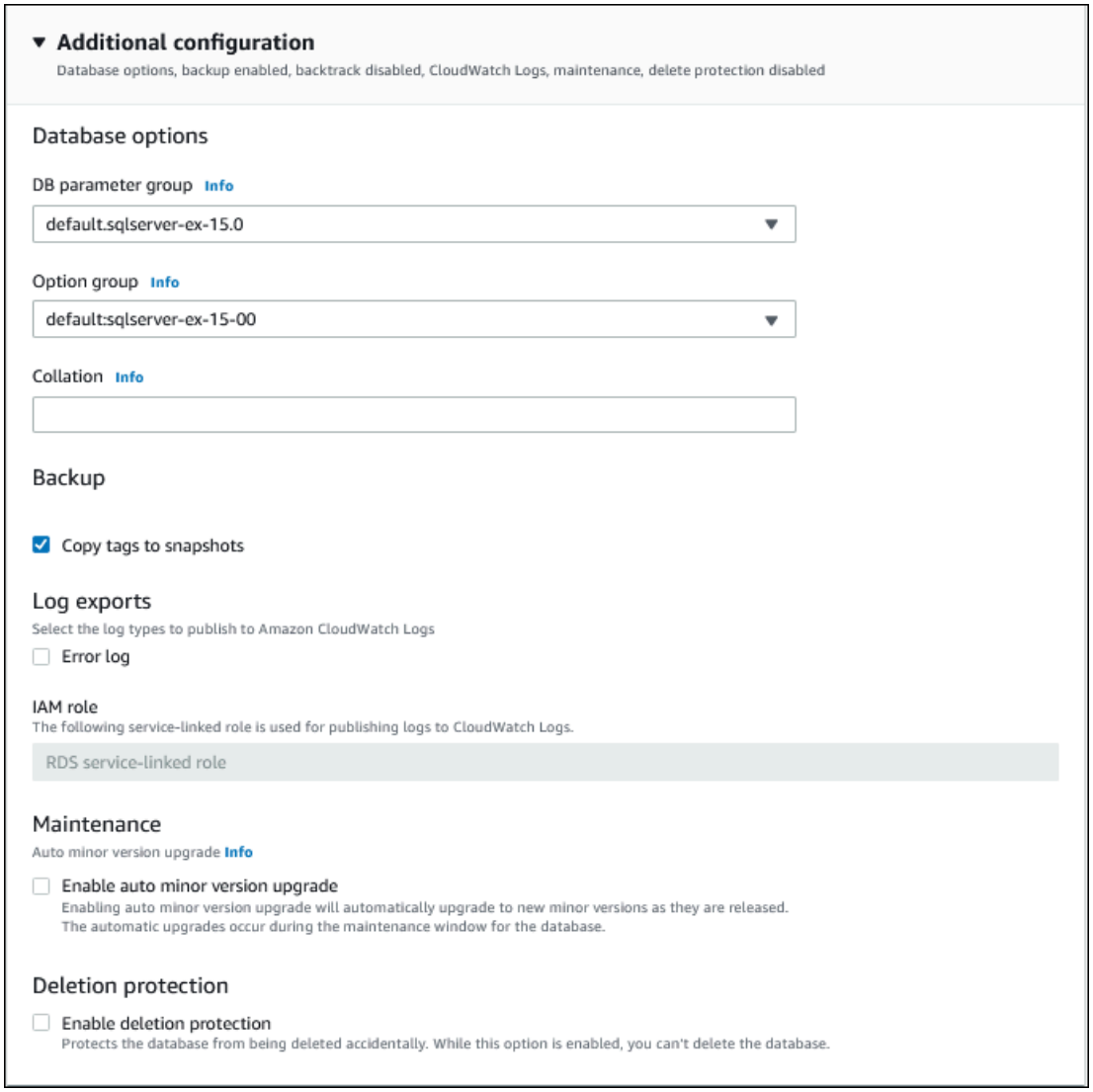
   - Cloud privat virtual (VPC)
   - Grup subnet DB
   - Akses publik
   - Grup keamanan VPC (firewall)
9. Pilih kelas instans DB.

Untuk tutorial ini, pilih Kelas runtutan (termasuk kelas t), lalu pilih db.t3.small.

10. Untuk Enkripsi, gunakan pengaturan default.

    Jika instans DB sumber untuk snapshot dienkripsi, instans DB yang dipulihkan juga dienkripsi. Anda tidak dapat membuatnya tidak terenkripsi.

11. Luaskan Konfigurasi tambahan di bagian bawah halaman.



12. Lakukan hal berikut di bagian Opsi basis data:

    a. Pilih Grup parameter DB.

       Untuk tutorial ini, gunakan grup parameter default.

    b. Pilih Grup opsi.

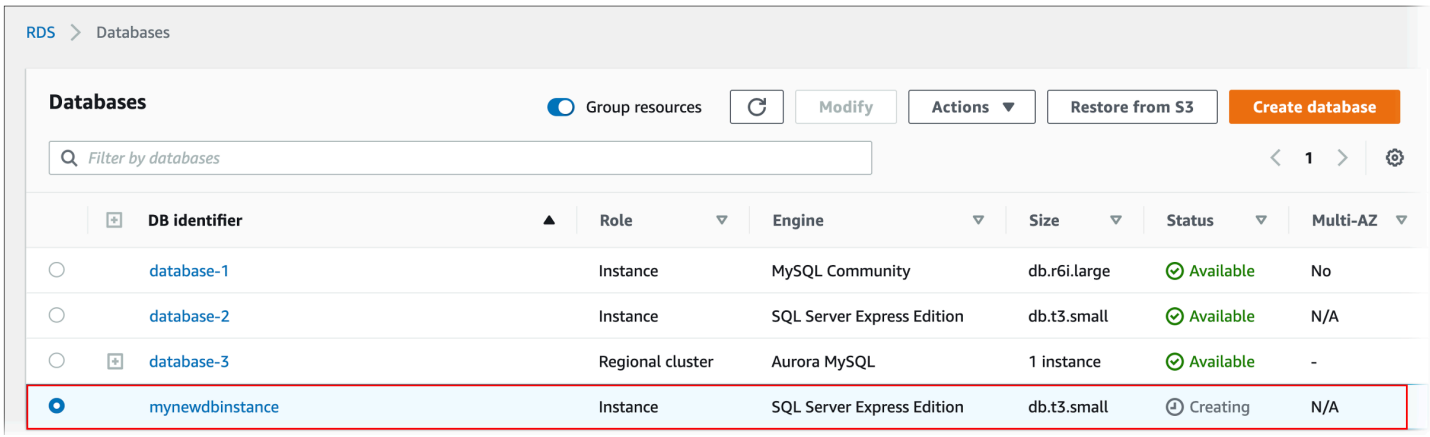
Untuk tutorial ini, gunakan grup opsi default.

> **Important**
>
> Dalam beberapa kasus, Anda dapat memulihkan dari snapshot DB instans DB yang menggunakan opsi persisten atau permanen. Jika demikian, pastikan untuk memilih grup opsi yang menggunakan opsi yang sama.

c. Untuk Perlindungan penghapusan, centang kotak Aktifkan perlindungan penghapusan.

13. Pilih Pulihkan instans DB.

Halaman Basis Data menampilkan instans DB yang dipulihkan, dengan status `Creating`.



Menyalin snapshot DB

Dengan Amazon RDS, Anda dapat menyalin pencadangan otomatis atau snapshot DB manual. Setelah Anda menyalin snapshot, salinannya adalah snapshot manual. Anda dapat membuat beberapa salinan cadangan otomatis atau snapshot manual, tetapi setiap salinan harus memiliki pengidentifikasi unik.

Anda dapat menyalin snapshot dalam hal yang sama Wilayah AWS, Anda dapat menyalin snapshot di seluruh Wilayah AWS, dan Anda dapat menyalin snapshot bersama.

Batasan

Berikut ini adalah beberapa batasan saat Anda menyalin snapshot:

- Anda tidak dapat menyalin snapshot ke atau dari Wilayah Tiongkok (Beijing) atau Wilayah Tiongkok (Ningxia).
- Anda dapat menyalin snapshot antara AWS GovCloud (AS-Timur) dan AWS GovCloud (AS-Barat). Namun, Anda tidak dapat menyalin snapshot antara Wilayah dan Wilayah GovCloud (AS) ini yang bukan Wilayah GovCloud (AS).
- Jika Anda menghapus snapshot sumber sebelum snapshot target tersedia, penyalinan snapshot mungkin gagal. Verifikasi bahwa snapshot target memiliki status `AVAILABLE` sebelum Anda menghapus snapshot sumber.
- Anda dapat memiliki hingga 20 snapshot permintaan salinan yang berlangsung ke satu Wilayah tujuan per akun.
- Saat Anda meminta beberapa salinan snapshot untuk instans DB sumber yang sama, salinan tersebut diantrekan secara internal. Salinan yang diminta nanti tidak akan dimulai hingga salinan snapshot sebelumnya selesai. Untuk informasi selengkapnya, lihat Mengapa pembuatan snapshot EC2 AMI atau EBS saya lambat? di pusat AWS pengetahuan.
- Bergantung pada yang Wilayah AWS terlibat dan jumlah data yang akan disalin, salinan snapshot lintas wilayah dapat memakan waktu berjam-jam untuk diselesaikan. Dalam beberapa kasus, mungkin ada sejumlah besar permintaan penyalinan snapshot lintas Wilayah dari Wilayah sumber tertentu. Dalam kasus seperti itu, Amazon RDS mungkin menempatkan permintaan penyalinan lintas Wilayah baru dari Wilayah sumber tersebut ke dalam antrean hingga beberapa penyalinan yang sedang berlangsung selesai. Tidak ada informasi progres yang ditampilkan tentang permintaan penyalinan saat permintaan tersebut ada di dalam antrean. Informasi kemajuan ditampilkan saat penyalinan dimulai.

- Jika salinan masih tertunda saat Anda memulai salinan lain, salinan kedua dimulai hanya setelah salinan pertama selesai.
- Anda tidak dapat menyalin snapshot dari cluster DB multi-AZ.

Penyimpanan Snapshot

Amazon RDS menghapus cadangan otomatis dalam beberapa situasi:

- Pada akhir periode retensi snapshot.
- Saat Anda menonaktifkan pencadangan otomatis untuk instans DB.
- Saat Anda menghapus instans DB.

Jika Anda ingin mempertahankan pencadangan otomatis untuk jangka waktu lebih lama, salin untuk membuat snapshot manual, yang dipertahankan hingga Anda menghapusnya. Biaya penyimpanan Amazon RDS mungkin berlaku untuk snapshot manual jika melebihi ruang penyimpanan default.

Untuk informasi selengkapnya tentang biaya penyimpanan cadangan, lihat Harga Amazon RDS.

Menyalin snapshot bersama

Anda dapat menyalin snapshot yang dibagikan kepada Anda oleh orang lain Akun AWS. Dalam beberapa kasus, Anda mungkin menyalin snapshot terenkripsi yang telah dibagikan dari yang lain. Akun AWS Dalam kasus ini, Anda harus memiliki akses ke AWS KMS key yang digunakan untuk mengenkripsi snapshot.

Anda dapat menyalin snapshot DB bersama Wilayah AWS jika snapshot tidak dienkripsi. Namun, jika snapshot DB bersama tersebut dienkripsi, Anda hanya dapat menyalinnya di Wilayah yang sama.

> **Note**
>
> Menyalin snapshot inkremental bersama yang sama didukung saat tidak Wilayah AWS dienkripsi, atau dienkripsi menggunakan kunci KMS yang sama dengan snapshot penuh awal. Jika Anda menggunakan kunci KMS yang berbeda untuk mengenkripsi snapshot berikutnya saat menyalinnya, snapshot yang dibagikan tersebut adalah snapshot penuh. Untuk informasi selengkapnya, lihat Penyalinan snapshot inkremental.

Menangani enkripsi

Anda dapat menyalin snapshot yang telah dienkripsi menggunakan kunci KMS. Jika Anda menyalin snapshot terenkripsi, salinan snapshot tersebut juga harus dienkripsi. Jika Anda menyalin snapshot terenkripsi dalam yang sama Wilayah AWS, Anda dapat mengenkripsi salinan dengan kunci KMS yang sama dengan snapshot asli. Atau Anda dapat menentukan kunci KMS yang berbeda.

Jika Anda menyalin snapshot terenkripsi di seluruh Wilayah, Anda harus menentukan kunci KMS yang valid di Wilayah AWS tujuan. Kunci ini dapat berupa kunci KMS khusus Wilayah, atau kunci multi-Wilayah. Untuk informasi selengkapnya tentang kunci KMS multi-Wilayah, lihat Menggunakan kunci multi-Wilayah di AWS KMS.

Snapshot sumber tetap terenkripsi selama proses penyalinan. Untuk informasi selengkapnya, lihat .

Anda juga dapat mengenkripsi salinan snapshot yang tidak terenkripsi. Dengan cara ini, Anda dapat dengan cepat menambahkan enkripsi ke instans DB yang tidak dienkripsi sebelumnya. Untuk melakukan ini, Anda membuat snapshot instans DB saat Anda siap mengenkripsinya. Kemudian, Anda membuat salinan snapshot tersebut dan menentukan kunci KMS untuk mengenkripsi salinan snapshot tersebut. Anda kemudian dapat memulihkan instans DB terenkripsi dari snapshot terenkripsi.

Penyalinan snapshot inkremental

Snapshot inkremental hanya berisi data yang telah berubah setelah snapshot terbaru dari instans DB yang sama. Penyalinan snapshot inkremental lebih cepat dan menghasilkan biaya penyimpanan yang lebih rendah daripada penyalinan snapshot penuh.

> **Note**
>
> Saat Anda menyalin snapshot sumber yang merupakan salinan snapshot itu sendiri, salinan baru tidak bersifat inkremental. Hal ini karena salinan snapshot sumber tidak menyertakan metadata yang diperlukan untuk salinan inkremental.

Apakah salinan snapshot bersifat tambahan ditentukan oleh salinan snapshot yang baru saja diselesaikan. Jika salinan snapshot terbaru dihapus, salinan berikutnya adalah salinan penuh, bukan salinan inkremental.

Saat Anda menyalin snapshot Akun AWS, salinannya adalah salinan tambahan hanya jika semua kondisi berikut terpenuhi:

- Snapshot yang berbeda dari instans DB yang sama sebelumnya disalin ke akun tujuan.
- Salinan snapshot terbaru masih ada di akun tujuan.
- Semua salinan snapshot di akun tujuan tidak dienkripsi atau dienkripsi menggunakan kunci KMS yang sama.
- Jika instans DB sumber adalah instans Multi-AZ, instans tersebut tidak gagal ke AZ lain sejak snapshot terakhir diambil.

Contoh berikut ini menggambarkan perbedaan antara snapshot penuh dan inkremental. Contoh ini berlaku untuk snapshot yang dibagikan dan tidak dibagikan.

Snapshot	Kunci enkripsi	Penuh atau inkremental
S1	K1	Penuh
S2	K1	Inkremental S1
S3	K1	Inkremental S2
S4	K1	Inkremental S3
Salinan S1 (S1C)	K2	Penuh
Salinan S2 (S2C)	K3	Penuh
Salinan S3 (S3C)	K3	Inkremental S2C
Salinan S4 (S4C)	K3	Inkremental S3C
Salinan 2 dari S4 (S4C2)	K4	Penuh

> **Note**
>
> Dalam contoh ini, snapshot S2, S3, dan S4 hanya bersifat inkremental jika snapshot sebelumnya masih ada.
> Hal yang sama berlaku untuk salinan. Salinan snapshot S3C dan S4C hanya inkremental jika salinan sebelumnya masih ada.

Untuk informasi tentang menyalin snapshot inkremental di seluruh Wilayah AWS, lihat. Salinan penuh dan bersifat inkremental

Penyalinan snapshot lintas Wilayah

Anda dapat menyalin snapshot DB di seluruh Wilayah AWS. Namun, ada kendala dan pertimbangan tertentu penyalinan snapshot lintas Wilayah.

Meminta salinan snapshot DB lintas Wilayah

Untuk berkomunikasi dengan Wilayah sumber untuk meminta salinan snapshot DB lintas Wilayah, pemohon (peran IAM atau pengguna IAM) harus memiliki akses ke sumber snapshot DB dan Wilayah sumber.

Kondisi tertentu dalam kebijakan IAM peminta dapat menyebabkan permintaan gagal. Contoh berikut berasumsi bahwa Anda menyalin snapshot DB dari AS Timur (Ohio) ke AS Timur (Virginia Utara). Contoh ini menunjukkan kondisi dalam kebijakan IAM pemohon yang menyebabkan permintaan gagal:

- Kebijakan milik pemohon memiliki kondisi untuk `aws:RequestedRegion`.

```
...
"Effect": "Allow",
"Action": "rds:CopyDBSnapshot",
"Resource": "*",
"Condition": {
    "StringEquals": {
        "aws:RequestedRegion": "us-east-1"
    }
}
```

Permintaan gagal karena kebijakan tidak mengizinkan akses ke Wilayah sumber. Untuk permintaan yang berhasil, tentukan Wilayah sumber dan tujuan.

```
...
"Effect": "Allow",
"Action": "rds:CopyDBSnapshot",
"Resource": "*",
"Condition": {
    "StringEquals": {
        "aws:RequestedRegion": [
```

```
            "us-east-1",
            "us-east-2"
        ]
    }
}
```

- Kebijakan milik pemohon tidak memungkinkan akses ke snapshot DB sumber.

```
...
"Effect": "Allow",
"Action": "rds:CopyDBSnapshot",
"Resource": "arn:aws:rds:us-east-1:123456789012:snapshot:target-snapshot"
...
```

Untuk permintaan yang berhasil, tentukan snapshot sumber dan target.

```
...
"Effect": "Allow",
"Action": "rds:CopyDBSnapshot",
"Resource": [
    "arn:aws:rds:us-east-1:123456789012:snapshot:target-snapshot",
    "arn:aws:rds:us-east-2:123456789012:snapshot:source-snapshot"
]
...
```

- Kebijakan pemohon menolak `aws:ViaAWSService`.

```
...
"Effect": "Allow",
"Action": "rds:CopyDBSnapshot",
"Resource": "*",
"Condition": {
    "Bool": {"aws:ViaAWSService": "false"}
}
```

Komunikasi dengan Wilayah sumber dibuat oleh RDS atas nama pemohon. Untuk permintaan yang berhasil, jangan menolak panggilan yang dilakukan oleh AWS layanan.

- Kebijakan milik pemohon memiliki kondisi untuk `aws:SourceVpc` atau `aws:SourceVpce`.

Permintaan ini mungkin gagal karena ketika RDS membuat panggilan ke Wilayah jarak jauh, panggilan tersebut bukan dari VPC atau titik akhir VPC yang ditentukan.

Jika Anda perlu menggunakan salah satu kondisi sebelumnya yang akan menyebabkan permintaan gagal, Anda dapat menyertakan pernyataan kedua dengan `aws:CalledVia` dalam kebijakan Anda untuk membuat permintaan berhasil. Misalnya, Anda dapat menggunakan `aws:CalledVia` dengan `aws:SourceVpce` seperti yang ditunjukkan di sini:

```
...
"Effect": "Allow",
"Action": "rds:CopyDBSnapshot",
"Resource": "*",
"Condition": {
    "Condition" : {
        "ForAnyValue:StringEquals" : {
          "aws:SourceVpce": "vpce-1a2b3c4d"
        }
     }
},
{
    "Effect": "Allow",
    "Action": [
        "rds:CopyDBSnapshot"
    ],
    "Resource": "*",
    "Condition": {
        "ForAnyValue:StringEquals": {
            "aws:CalledVia": [
                "rds.amazonaws.com"
            ]
        }
    }
}
```

Untuk mengetahui informasi selengkapnya, lihat Kebijakan dan izin di IAM pada Panduan Pengguna IAM.

Mengotorisasi salinan snapshot

Setelah permintaan salinan snapshot DB lintas Wilayah menampilkan `success`, RDS mulai menyalin di latar belakang. Otorisasi untuk RDS untuk mengakses snapshot sumber telah dibuat. Otorisasi ini mengaitkan snapshot DB sumber ke snapshot DB target, dan mengizinkan RDS untuk menyalin hanya ke snapshot target yang ditentukan.

Otorisasi ini diverifikasi oleh RDS menggunakan izin `rds:CrossRegionCommunication` dalam peran IAM yang tertaut dengan layanan. Jika salinan sudah diotorisasi, RDS berkomunikasi dengan Wilayah sumber dan menyelesaikan salinan.

RDS tidak memiliki akses ke snapshot DB yang tidak diotorisasi sebelumnya oleh permintaan `CopyDBSnapshot`. Otorisasi dicabut saat penyalinan selesai.

RDS menggunakan peran yang terkait layanan untuk memverifikasi otorisasi di Wilayah sumber. Jika Anda menghapus peran yang tertaut dengan layanan selama proses penyalinan, salinan akan gagal.

Untuk mengetahui informasi selengkapnya, lihat Menggunakan peran tertaut layanan di Panduan Pengguna IAM.

Menggunakan AWS Security Token Service kredensil

Token sesi dari titik akhir global AWS Security Token Service (AWS STS) hanya valid di Wilayah AWS yang diaktifkan secara default (Wilayah komersial). Jika Anda menggunakan kredensil dari operasi `assumeRole` API di AWS STS, gunakan titik akhir regional jika Wilayah sumber adalah Region keikutsertaan. Jika tidak, permintaan akan gagal. Hal ini terjadi karena kredensyal Anda harus valid di kedua Wilayah, yang berlaku untuk Wilayah keikutsertaan hanya jika titik AWS STS akhir regional digunakan.

Untuk menggunakan titik akhir global, pastikan bahwa titik akhir tersebut diaktifkan untuk kedua Wilayah dalam operasi. Setel titik akhir global ke `Valid in all Wilayah AWS` dalam pengaturan AWS STS akun.

Aturan yang sama berlaku untuk kredensial dalam parameter URL yang telah ditandatangani sebelumnya.

Untuk informasi selengkapnya, lihat Mengelola AWS STS Wilayah AWS dalam Panduan Pengguna IAM.

Latensi dan permintaan beberapa salinan

Bergantung pada yang Wilayah AWS terlibat dan jumlah data yang akan disalin, salinan snapshot lintas wilayah dapat memakan waktu berjam-jam untuk diselesaikan.

Dalam beberapa kasus, mungkin ada sejumlah besar permintaan penyalinan snapshot lintas Wilayah dari Wilayah AWS sumber tertentu. Dalam kasus seperti itu, Amazon RDS mungkin menempatkan permintaan salinan Lintas wilayah baru dari sumber tersebut Wilayah AWS ke dalam antrian hingga beberapa salinan yang sedang berlangsung selesai. Tidak ada informasi progres yang ditampilkan

tentang permintaan penyalinan saat permintaan tersebut ada di dalam antrean. Informasi kemajuan ditampilkan saat penyalinan dimulai.

Salinan penuh dan bersifat inkremental

Saat Anda menyalin snapshot ke snapshot yang berbeda Wilayah AWS dari sumber, salinan pertama adalah salinan snapshot lengkap, bahkan jika Anda menyalin snapshot tambahan. Salinan snapshot lengkap berisi semua data dan metadata yang diperlukan untuk memulihkan instans DB. Setelah salinan snapshot pertama, Anda dapat menyalin snapshot tambahan dari instans DB yang sama ke Wilayah tujuan yang sama dalam hal yang sama. Akun AWS Untuk mengetahui informasi selengkapnya tentang snapshot inkremental, lihat Penyalinan snapshot inkremental.

Penyalinan snapshot tambahan di seluruh didukung untuk snapshot Wilayah AWS yang tidak terenkripsi dan terenkripsi.

Saat Anda menyalin snapshot Wilayah AWS, salinannya adalah salinan tambahan jika kondisi berikut terpenuhi:

- Snapshot sebelumnya disalin ke Wilayah tujuan.
- Salinan snapshot terbaru masih ada di Wilayah tujuan.
- Semua salinan snapshot di Wilayah tujuan tidak dienkripsi atau dienkripsi menggunakan kunci KMS yang sama.

Pertimbangan grup opsi

Grup opsi DB khusus untuk tempat Wilayah AWS mereka dibuat, dan Anda tidak dapat menggunakan grup opsi dari satu Wilayah AWS sama lain Wilayah AWS.

Untuk database Oracle, Anda dapat menggunakan AWS CLI atau RDS API untuk menyalin grup opsi DB kustom dari snapshot yang telah dibagikan dengan Anda. Akun AWS Anda hanya dapat menyalin grup opsi dalam Wilayah AWS yang sama. Grup opsi tidak disalin jika telah disalin ke akun tujuan dan tidak ada perubahan yang dilakukan sejak disalin. Jika grup opsi sumber telah disalin sebelumnya, tetapi telah berubah sejak disalin, RDS menyalin versi baru ke akun tujuan. Grup opsi default tidak disalin.

Saat Anda menyalin snapshot lintas Wilayah, Anda dapat menentukan kelompok opsi baru untuk snapshot. Sebaiknya siapkan grup opsi baru sebelum menyalin snapshot. Di tujuan Wilayah AWS, buat grup opsi dengan pengaturan yang sama dengan instans DB asli. Jika sudah ada di yang baru Wilayah AWS, Anda dapat menggunakan yang itu.

Dalam beberapa kasus, Anda mungkin menyalin snapshot dan tidak menentukan grup opsi baru untuk snapshot. Dalam kasus ini, jika Anda memulihkan snapshot, instans DB mendapatkan grup opsi default. Untuk memberikan instans DB baru opsi yang sama seperti aslinya, Anda harus melakukan hal berikut:

1. Di tujuan Wilayah AWS, buat grup opsi dengan pengaturan yang sama dengan instans DB asli. Jika sudah ada di yang baru Wilayah AWS, Anda dapat menggunakan yang itu.
2. Setelah Anda mengembalikan snapshot di tujuan Wilayah AWS, ubah instans DB baru dan tambahkan grup opsi baru atau yang sudah ada dari langkah sebelumnya.

Pertimbangan grup parameter

Saat Anda menyalin snapshot lintas Wilayah, salinan tersebut tidak menyertakan grup parameter yang digunakan oleh instans DB asli. Saat Anda mengembalikan snapshot untuk membuat instance DB baru, instans DB itu mendapatkan grup parameter default untuk Wilayah AWS itu dibuat. Untuk memberikan instans DB baru parameter yang sama seperti aslinya, Anda harus melakukan hal berikut:

1. Di tujuan Wilayah AWS, buat grup parameter DB dengan pengaturan yang sama dengan instans DB asli. Jika sudah ada di yang baru Wilayah AWS, Anda dapat menggunakan yang itu.
2. Setelah Anda mengembalikan snapshot di tujuan Wilayah AWS, ubah instans DB baru dan tambahkan grup parameter baru atau yang sudah ada dari langkah sebelumnya.

Menyalin snapshot DB

Gunakan prosedur dalam topik ini untuk menyalin snapshot DB. Untuk ringkasan penyalinan snapshot, lihat Menyalin snapshot DB

Untuk masing-masing Akun AWS, Anda dapat menyalin hingga 20 snapshot DB sekaligus dari satu Wilayah AWS ke yang lain. Jika Anda menyalin snapshot DB ke yang lain Wilayah AWS, Anda membuat snapshot DB manual yang dipertahankan di dalamnya. Wilayah AWS Menyalin snapshot DB dari sumber menimbulkan biaya transfer Wilayah AWS data Amazon RDS.

Untuk informasi selengkapnya tentang biaya transfer data, lihat Harga Amazon RDS.

Setelah salinan snapshot DB dibuat di yang baru Wilayah AWS, salinan snapshot DB berperilaku sama seperti semua snapshot DB lainnya di dalamnya. Wilayah AWS

Anda dapat menyalin snapshot DB menggunakan AWS Management Console, AWS CLI, atau RDS API.

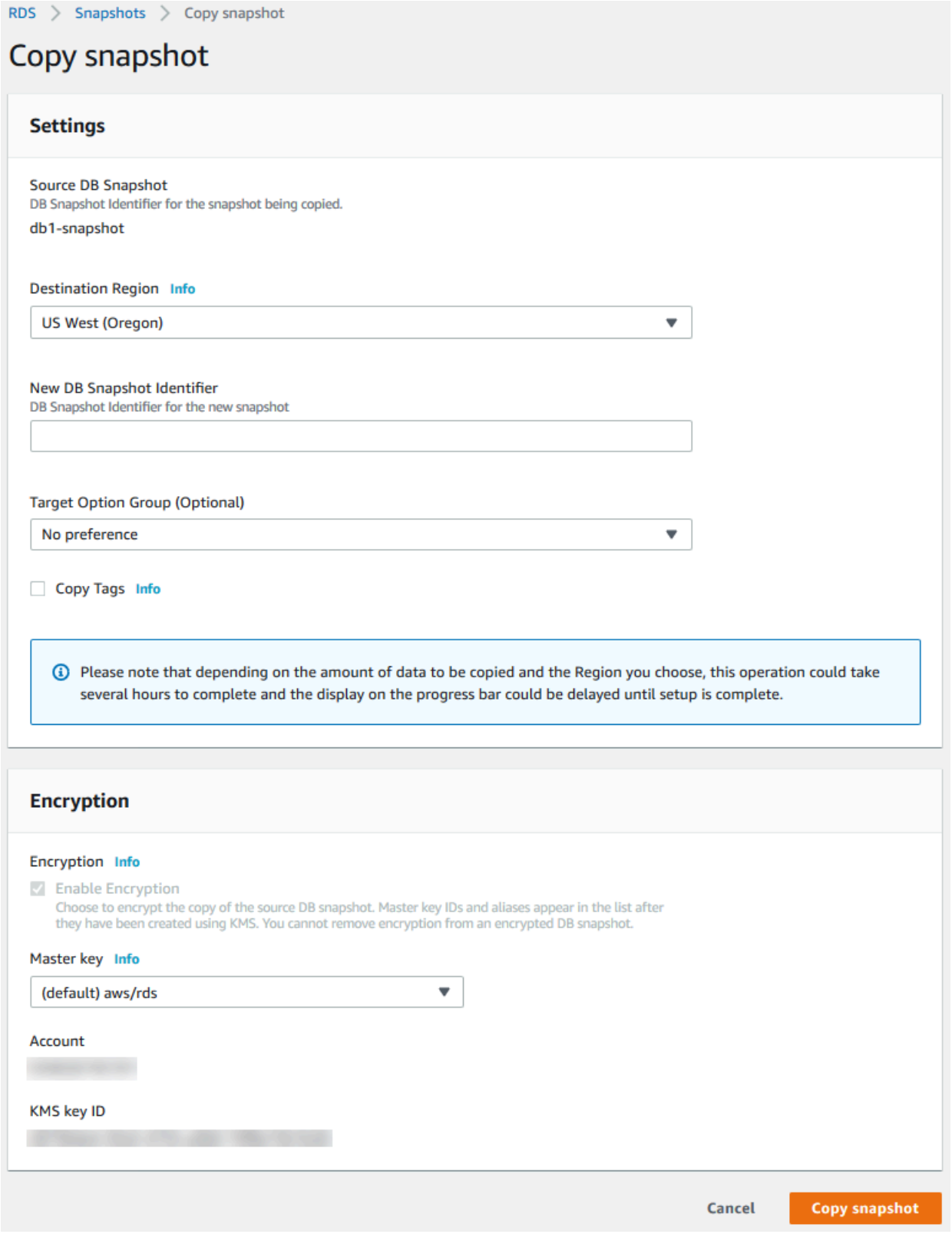
Konsol

Prosedur berikut menyalin snapshot DB terenkripsi atau tidak terenkripsi, di wilayah yang sama Wilayah AWS atau di seluruh Wilayah, dengan menggunakan file. AWS Management Console

Untuk menyalin snapshot DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Snapshot.
3. Pilih snapshot DB yang ingin Anda salin.
4. Untuk Tindakan, pilih Salin snapshot.

   Halaman Salin Snapshot akan muncul.



5. Untuk Grup opsi target (opsional), pilih grup opsi baru jika Anda mau.

Tentukan opsi ini jika Anda menyalin snapshot dari satu Wilayah AWS ke yang lain, dan instans DB Anda menggunakan grup opsi nondefault.

Jika sumber instans DB Anda menggunakan Enkripsi Data Transparan untuk Oracle atau Microsoft SQL Server, Anda harus menentukan opsi ini saat menyalin di seluruh Wilayah. Untuk informasi selengkapnya, lihat Pertimbangan grup opsi.

6. (Opsional) Untuk menyalin snapshot DB ke yang lain Wilayah AWS, untuk Wilayah Tujuan, pilih yang baru Wilayah AWS.

> **Note**
>
> Tujuan Wilayah AWS harus memiliki versi mesin database yang sama yang tersedia sebagai sumbernya Wilayah AWS.

7. Untuk Pengidentifikasi snapshot DB baru, ketikkan nama salinan snapshot DB.

   Anda dapat membuat beberapa salinan pencadangan otomatis atau snapshot manual, tetapi setiap salinan harus memiliki pengidentifikasi unik.

8. (Opsional) Pilih Salin Tag untuk menyalin tag dan nilai dari snapshot ke salinan snapshot.
9. (Opsional) Untuk Enkripsi, lakukan hal berikut:

   a. Pilih Aktifkan Enkripsi jika snapshot DB tidak dienkripsi, tetapi Anda ingin mengenkripsi salinan tersebut.

   > **Note**
   >
   > Jika snapshot DB terenkripsi, Anda harus mengenkripsi salinan, sehingga kotak centang sudah dipilih.

   b. Untuk AWS KMS key, tentukan pengidentifikasi kunci KMS untuk mengenkripsi salinan snapshot DB.

10. Pilih Salin snapshot.

AWS CLI

Anda dapat menyalin snapshot DB dengan menggunakan AWS CLI perintah copy-db-snapshot. Jika Anda menyalin snapshot ke yang baru Wilayah AWS, jalankan perintah di yang baru. Wilayah AWS

Opsi berikut digunakan untuk menghapus snapshot DB. Tidak semua opsi diperlukan untuk semua skenario. Gunakan deskripsi dan contoh berikut untuk menentukan opsi mana yang akan digunakan.

- `--source-db-snapshot-identifier` – Pengidentifikasi untuk snapshot DB sumber.
  - Jika snapshot sumber Wilayah AWS sama dengan salinan, tentukan pengidentifikasi snapshot DB yang valid. Misalnya, `rds:mysql-instance1-snapshot-20130805`.
  - Jika snapshot sumber Wilayah AWS sama dengan salinan, dan telah dibagikan dengan Anda Akun AWS, tentukan ARN snapshot DB yang valid. Misalnya, `arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20130805`.
  - Jika snapshot sumber berbeda Wilayah AWS dari salinan, tentukan ARN snapshot DB yang valid. Sebagai contoh, `arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20130805`.
  - Jika Anda menyalin dari snapshot DB manual bersama, parameter ini harus merupakan Amazon Resource Name (ARN) dari snapshot DB yang telah dibagikan.
  - Jika Anda menyalin snapshot terenkripsi, parameter ini harus dalam format ARN untuk sumbernya Wilayah AWS, dan harus cocok dengan parameter. `SourceDBSnapshotIdentifier` `PreSignedUrl`
- `--target-db-snapshot-identifier` – Pengidentifikasi untuk salinan baru snapshot DB terenkripsi.
- `--copy-option-group` – Salin grup opsi dari snapshot yang telah dibagikan dengan Akun AWS Anda.
- `--copy-tags` – Sertakan opsi salin tag untuk menyalin tag dan nilai dari snapshot ke salinan snapshot.
- `--option-group-name` – Grup opsi untuk dikaitkan dengan salinan snapshot.

  Tentukan opsi ini jika Anda menyalin snapshot dari satu Wilayah AWS ke yang lain, dan instans DB Anda menggunakan grup opsi non-default.

  Jika sumber instans DB Anda menggunakan Enkripsi Data Transparan untuk Oracle atau Microsoft SQL Server, Anda harus menentukan opsi ini saat menyalin di seluruh Wilayah. Untuk informasi selengkapnya, lihat Pertimbangan grup opsi.
- `--kms-key-id` – Pengidentifikasi kunci KMS untuk snapshot DB terenkripsi. Pengidentifikasi kunci KMS adalah Amazon Resource Name (ARN), pengidentifikasi kunci, atau alias kunci untuk kunci KMS.

- Jika Anda menyalin snapshot DB terenkripsi dari Anda Akun AWS, Anda dapat menentukan nilai untuk parameter ini untuk mengenkripsi salinan dengan kunci KMS baru. Jika Anda tidak menentukan nilai untuk parameter ini, salinan snapshot DB dienkripsi dengan kunci KMS yang sama dengan snapshot DB sumber.
- Jika Anda menyalin snapshot DB terenkripsi yang dibagikan dari yang lain Akun AWS, maka Anda harus menentukan nilai untuk parameter ini.
- Jika Anda menetapkan parameter ini saat menyalin snapshot yang tidak dienkripsi, salinannya dienkripsi.
- Jika Anda menyalin snapshot terenkripsi ke yang berbeda Wilayah AWS, maka Anda harus menentukan kunci KMS untuk tujuan. Wilayah AWS Kunci KMS khusus untuk tempat Wilayah AWS mereka dibuat, dan Anda tidak dapat menggunakan kunci enkripsi dari satu Wilayah AWS sama lain Wilayah AWS.

Example dari kunci yang tidak terenkripsi, ke Wilayah yang sama

Kode berikut membuat salinan snapshot, dengan nama baru`mydbsnapshotcopy`, Wilayah AWS sama dengan snapshot sumber. Ketika salinan dibuat, grup opsi DB dan tag pada snapshot asli disalin ke salinan snapshot.

Untuk Linux, macOS, atau Unix:

```
aws rds copy-db-snapshot \
    --source-db-snapshot-identifier arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20130805 \
    --target-db-snapshot-identifier mydbsnapshotcopy \
    --copy-option-group \
    --copy-tags
```

Untuk Windows:

```
aws rds copy-db-snapshot ^
    --source-db-snapshot-identifier arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20130805 ^
    --target-db-snapshot-identifier mydbsnapshotcopy ^
    --copy-option-group ^
    --copy-tags
```

Example dari kunci yang tidak terenkripsi, di seluruh Wilayah

Kode berikut membuat salinan snapshot, dengan nama barumydbsnapshotcopy, Wilayah AWS di mana perintah dijalankan.

Untuk Linux, macOS, atau Unix:

```
aws rds copy-db-snapshot \
    --source-db-snapshot-identifier arn:aws:rds:us-east-1:123456789012:snapshot:mysql-instance1-snapshot-20130805 \
    --target-db-snapshot-identifier mydbsnapshotcopy
```

Untuk Windows:

```
aws rds copy-db-snapshot ^
    --source-db-snapshot-identifier arn:aws:rds:us-east-1:123456789012:snapshot:mysql-instance1-snapshot-20130805 ^
    --target-db-snapshot-identifier mydbsnapshotcopy
```

Example dari kunci yang terenkripsi, di seluruh Wilayah

Contoh kode berikut ini menyalin snapshot DB terenkripsi dari AS Barat (Oregon) dalam Wilayah AS Timur (Virginia Utara). Jalankan perintah di Wilayah tujuan (us-east-1).

Untuk Linux, macOS, atau Unix:

```
aws rds copy-db-snapshot \
    --source-db-snapshot-identifier arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20161115 \
    --target-db-snapshot-identifier mydbsnapshotcopy \
    --kms-key-id my-us-east-1-key \
    --option-group-name custom-option-group-name
```

Untuk Windows:

```
aws rds copy-db-snapshot ^
    --source-db-snapshot-identifier arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20161115 ^
    --target-db-snapshot-identifier mydbsnapshotcopy ^
    --kms-key-id my-us-east-1-key ^
    --option-group-name custom-option-group-name
```

`--source-region`Parameter diperlukan saat Anda menyalin snapshot terenkripsi antara Wilayah AWS GovCloud (AS-Timur) dan AWS GovCloud (AS-Barat). Untuk `--source-region`, tentukan Wilayah AWS instans DB sumber.

Jika `--source-region` tidak ditentukan, tentukan nilai `--pre-signed-url`. URL yang telah ditandatangani adalah URL yang berisi permintaan bertanda tangan Signature Versi 4 untuk perintah `copy-db-snapshot` yang dipanggil di Wilayah AWS sumber. Untuk mempelajari lebih lanjut tentang `pre-signed-url` opsi, lihat copy-db-snapshotdi Referensi AWS CLI Perintah.

API RDS

Anda dapat menghapus snapshot DB menggunakan operasi Amazon RDS API CopyDBSnapshot. Jika Anda menyalin snapshot ke yang baru Wilayah AWS, lakukan tindakan di yang baru. Wilayah AWS

Parameter berikut digunakan untuk menghapus snapshot DB. Tidak semua parameter diperlukan untuk semua skenario. Gunakan deskripsi dan contoh berikut untuk menentukan parameter mana yang akan digunakan.

- `SourceDBSnapshotIdentifier` – Pengidentifikasi untuk snapshot DB sumber.
  - Jika snapshot sumber Wilayah AWS sama dengan salinan, tentukan pengidentifikasi snapshot DB yang valid. Misalnya, `rds:mysql-instance1-snapshot-20130805`.
  - Jika snapshot sumber Wilayah AWS sama dengan salinan, dan telah dibagikan dengan Anda Akun AWS, tentukan ARN snapshot DB yang valid. Misalnya, `arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20130805`.
  - Jika snapshot sumber berbeda Wilayah AWS dari salinan, tentukan ARN snapshot DB yang valid. Sebagai contoh, `arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20130805`.
  - Jika Anda menyalin dari snapshot DB manual bersama, parameter ini harus merupakan Amazon Resource Name (ARN) dari snapshot DB yang telah dibagikan.
  - Jika Anda menyalin snapshot terenkripsi, parameter ini harus dalam format ARN untuk sumbernya Wilayah AWS, dan harus cocok dengan parameter. `SourceDBSnapshotIdentifier` `PreSignedUrl`
- `TargetDBSnapshotIdentifier` – Pengidentifikasi untuk salinan baru snapshot DB terenkripsi.
- `CopyOptionGroup` – Atur parameter ini ke `true` untuk menyalin grup opsi dari snapshot bersama ke salinan snapshot. Default-nya adalah `false`.

- `CopyTags` – Atur parameter ini ke `true` untuk menyalin tag dan nilai dari snapshot ke salinan snapshot. Defaultnya adalah `false`.
- `OptionGroupName` – Grup opsi untuk dikaitkan dengan salinan snapshot.

  Tentukan parameter ini jika Anda menyalin snapshot dari satu Wilayah AWS ke yang lain, dan instans DB Anda menggunakan grup opsi non-default.

  Jika sumber instans DB Anda menggunakan Enkripsi Data Transparan untuk Oracle atau Microsoft SQL Server, Anda harus menentukan parameter ini saat menyalin di seluruh Wilayah. Untuk informasi selengkapnya, lihat Pertimbangan grup opsi.
- `KmsKeyId` – Pengidentifikasi kunci KMS untuk snapshot DB terenkripsi. Pengidentifikasi kunci KMS adalah Amazon Resource Name (ARN), pengidentifikasi kunci, atau alias kunci untuk kunci KMS.
  - Jika Anda menyalin snapshot DB terenkripsi dari Anda Akun AWS, Anda dapat menentukan nilai untuk parameter ini untuk mengenkripsi salinan dengan kunci KMS baru. Jika Anda tidak menentukan nilai untuk parameter ini, salinan snapshot DB dienkripsi dengan kunci KMS yang sama dengan snapshot DB sumber.
  - Jika Anda menyalin snapshot DB terenkripsi yang dibagikan dari yang lain Akun AWS, maka Anda harus menentukan nilai untuk parameter ini.
  - Jika Anda menetapkan parameter ini saat menyalin snapshot yang tidak dienkripsi, salinannya dienkripsi.
  - Jika Anda menyalin snapshot terenkripsi ke yang berbeda Wilayah AWS, maka Anda harus menentukan kunci KMS untuk tujuan. Wilayah AWS Kunci KMS khusus untuk tempat Wilayah AWS mereka dibuat, dan Anda tidak dapat menggunakan kunci enkripsi dari satu Wilayah AWS sama lain Wilayah AWS.
- `PreSignedUrl`— URL yang berisi permintaan ditandatangani Signature Version 4 untuk operasi `CopyDBSnapshot` API di sumber Wilayah AWS yang berisi snapshot DB sumber untuk disalin.

  Tentukan parameter ini saat Anda menyalin snapshot DB terenkripsi dari yang lain Wilayah AWS dengan menggunakan Amazon RDS API. Anda dapat menentukan opsi Region sumber alih-alih parameter ini saat Anda menyalin snapshot DB terenkripsi dari Wilayah AWS lainnya dengan menggunakan AWS CLI.

  URL yang ditandatangani sebelumnya harus merupakan permintaan yang valid untuk operasi API `CopyDBSnapshot` yang dapat dijalankan di Wilayah AWS sumber yang berisi snapshot DB

terenkripsi yang akan disalin. Permintaan URL yang ditandatangani sebelumnya harus berisi nilai parameter berikut:

- `DestinationRegion`— Wilayah AWS Yang snapshot DB terenkripsi akan disalin. Ini Wilayah AWS adalah salah satu yang sama di mana `CopyDBSnapshot` operasi dipanggil yang berisi URL presigned ini.

  Misalnya, Anda menyalin snapshot DB terenkripsi dari Wilayah us-west-2 ke Wilayah us-east-1. Anda kemudian memanggil operasi `CopyDBSnapshot` di Wilayah us-east-1 dan memberikan URL yang telah ditandatangani sebelumnya yang berisi panggilan ke operasi `CopyDBSnapshot` di Wilayah us-west-2. Untuk contoh ini, `DestinationRegion` dalam URL yang telah ditandatangani sebelumnya harus diatur ke Wilayah us-east-1.
- `KmsKeyId` – Pengidentifikasi kunci KMS untuk kunci yang digunakan mengenkripsi salinan snapshot DB di Wilayah AWS tujuan. Ini adalah pengenal yang sama untuk kedua `CopyDBSnapshot` operasi yang dipanggil di tujuan Wilayah AWS, dan operasi yang terkandung dalam URL yang ditetapkan sebelumnya.
- `SourceDBSnapshotIdentifier` – Pengidentifikasi snapshot DB untuk snapshot terenkripsi yang akan disalin. Pengidentifikasi ini harus dalam format Amazon Resource Name (ARN) untuk Wilayah AWS sumber. Misalnya, jika Anda menyalin snapshot DB terenkripsi dari Wilayah us-barat-2, maka tampilan Anda `SourceDBSnapshotIdentifier` seperti contoh berikut:. `arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20161115`

Untuk mengetahui informasi selengkapnya tentang permintaan bertanda tangan Signature Versi 4, lihat berikut ini:

- Mengautentikasi permintaan: Menggunakan parameter kueri (versi AWS tanda tangan 4) di Referensi API Amazon Simple Storage Service
- Proses penandatanganan versi 4 tanda tangan di Referensi Umum AWS

Example dari kunci yang tidak terenkripsi, ke Wilayah yang sama

Kode berikut membuat salinan snapshot, dengan nama barumydbsnapshotcopy, Wilayah AWS sama dengan snapshot sumber. Setelah salinan dibuat, semua tanda di snapshot asli akan disalin ke salinan snapshot.

```
https://rds.us-west-1.amazonaws.com/
 ?Action=CopyDBSnapshot
 &CopyTags=true
```

```
 &SignatureMethod=HmacSHA256
 &SignatureVersion=4
 &SourceDBSnapshotIdentifier=mysql-instance1-snapshot-20130805
 &TargetDBSnapshotIdentifier=mydbsnapshotcopy
 &Version=2013-09-09
 &X-Amz-Algorithm=AWS4-HMAC-SHA256
 &X-Amz-Credential=AKIADQKE4SARGYLE/20140429/us-west-1/rds/aws4_request
 &X-Amz-Date=20140429T175351Z
 &X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
 &X-Amz-Signature=9164337efa99caf850e874a1cb7ef62f3cea29d0b448b9e0e7c53b288ddffed2
```

Example dari kunci yang tidak terenkripsi, di seluruh Wilayah

Kode berikut membuat salinan snapshot, dengan nama baru `mydbsnapshotcopy`, di AS Barat (California Utara).

```
https://rds.us-west-1.amazonaws.com/
 ?Action=CopyDBSnapshot
 &SignatureMethod=HmacSHA256
 &SignatureVersion=4
 &SourceDBSnapshotIdentifier=arn%3Aaws%3Ards%3Aus-east-1%3A123456789012%3Asnapshot
%3Amysql-instance1-snapshot-20130805
 &TargetDBSnapshotIdentifier=mydbsnapshotcopy
 &Version=2013-09-09
 &X-Amz-Algorithm=AWS4-HMAC-SHA256
 &X-Amz-Credential=AKIADQKE4SARGYLE/20140429/us-west-1/rds/aws4_request
 &X-Amz-Date=20140429T175351Z
 &X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
 &X-Amz-Signature=9164337efa99caf850e874a1cb7ef62f3cea29d0b448b9e0e7c53b288ddffed2
```

Example dari kunci yang terenkripsi, di seluruh Wilayah

Kode berikut membuat salinan snapshot, dengan nama baru `mydbsnapshotcopy`, di AS Timur (Virginia Utara).

```
https://rds.us-east-1.amazonaws.com/
    ?Action=CopyDBSnapshot
    &KmsKeyId=my-us-east-1-key
    &OptionGroupName=custom-option-group-name
    &PreSignedUrl=https%253A%252F%252Frds.us-west-2.amazonaws.com%252F
         %253FAction%253DCopyDBSnapshot
         %2526DestinationRegion%253Dus-east-1
```

```
        %2526KmsKeyId%253Dmy-us-east-1-key
        %2526SourceDBSnapshotIdentifier%253Darn%25253Aaws%25253Ards%25253Aus-
west-2%25253A123456789012%25253Asnapshot%25253Amysql-instance1-snapshot-20161115
        %2526SignatureMethod%253DHmacSHA256
        %2526SignatureVersion%253D4
        %2526Version%253D2014-10-31
        %2526X-Amz-Algorithm%253DAWS4-HMAC-SHA256
        %2526X-Amz-Credential%253DAKIADQKE4SARGYLE%252F20161117%252Fus-west-2%252Frds
%252Faws4_request
        %2526X-Amz-Date%253D20161117T215409Z
        %2526X-Amz-Expires%253D3600
        %2526X-Amz-SignedHeaders%253Dcontent-type%253Bhost%253Buser-agent%253Bx-amz-
content-sha256%253Bx-amz-date
        %2526X-Amz-Signature
%253D255a0f17b4e717d3b67fad163c3ec26573b882c03a65523522cf890a67fca613
    &SignatureMethod=HmacSHA256
    &SignatureVersion=4
    &SourceDBSnapshotIdentifier=arn%3Aaws%3Ards%3Aus-west-2%3A123456789012%3Asnapshot
%3Amysql-instance1-snapshot-20161115
    &TargetDBSnapshotIdentifier=mydbsnapshotcopy
    &Version=2014-10-31
    &X-Amz-Algorithm=AWS4-HMAC-SHA256
    &X-Amz-Credential=AKIADQKE4SARGYLE/20161117/us-east-1/rds/aws4_request
    &X-Amz-Date=20161117T221704Z
    &X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
    &X-Amz-Signature=da4f2da66739d2e722c85fcfd225dc27bba7e2b8dbea8d8612434378e52adccf
```

Berbagi snapshot DB

Dengan Amazon RDS, Anda dapat berbagi snapshot DB manual dengan cara berikut:

- Berbagi snapshot DB manual, baik terenkripsi atau tidak terenkripsi, memungkinkan otorisasi untuk menyalin snapshot. Akun AWS
- Berbagi snapshot DB manual yang tidak terenkripsi memungkinkan otorisasi Akun AWS untuk secara langsung memulihkan instance DB dari snapshot alih-alih mengambil salinannya dan memulihkannya. Namun, Anda tidak dapat memulihkan instans DB dari snapshot DB yang dibagikan dan dienkripsi. Sebagai gantinya, Anda dapat membuat salinan snapshot DB dan memulihkan instans DB dari salinan tersebut.

> **Note**
>
> Untuk berbagi snapshot DB otomatis, buat snapshot DB manual dengan menyalin snapshot otomatis, lalu membagikan salinannya. Proses ini juga berlaku untuk sumber daya yang AWS dihasilkan oleh Backup.

Untuk informasi selengkapnya tentang cara menyalin snapshot, lihat Menyalin snapshot DB. Untuk informasi selengkapnya tentang cara memulihkan instans DB dari snapshot DB, lihat Memulihkan dari snapshot DB.

Anda dapat berbagi snapshot manual dengan hingga 20 lainnya Akun AWS.

Batasan berikut berlaku saat berbagi foto manual dengan yang lain Akun AWS:

- Saat memulihkan instans DB dari snapshot bersama menggunakan AWS Command Line Interface (AWS CLI) atau Amazon RDS API, Anda harus menentukan Nama Sumber Daya Amazon (ARN) dari snapshot bersama sebagai pengidentifikasi snapshot.
- Anda tidak dapat membagikan snapshot DB yang menggunakan grup opsi dengan opsi permanen atau persisten, kecuali untuk instans DB Oracle yang memiliki opsi `Timezone` atau `OLS` (atau keduanya).

  Opsi permanen tidak dapat dihapus dari grup opsi. Grup opsi dengan opsi persisten tidak dapat dihapus dari instans DB setelah grup opsi ditugaskan ke instans DB.

  Tabel berikut ini menampilkan daftar opsi permanen dan persisten dan mesin DB terkait.

Nama opsi	Persisten	Permanen	Mesin DB
TDE	Ya	Tidak	Microsoft SQL Server Enterprise Edition
TDE	Ya	Ya	Oracle Enterprise Edition
Zona waktu	Ya	Ya	Oracle Enterprise Edition Oracle Standard Edition Oracle Standard Edition One Oracle Edisi Standar 2

Untuk instans DB Oracle, Anda dapat menyalin snapshot DB yang dibagikan yang memiliki opsi `Timezone` atau `OLS` (atau keduanya). Untuk melakukannya, tentukan grup opsi target yang mencakup opsi ini saat Anda menyalin snapshot DB. Opsi OLS bersifat permanen dan persisten hanya untuk instans DB Oracle yang menjalankan Oracle versi 12.2 atau lebih tinggi. Untuk informasi selengkapnya tentang opsi ini, lihat Zona waktu Oracle dan Keamanan Label Oracle.

- Anda tidak dapat membagikan snapshot dari cluster DB multi-AZ.

Daftar Isi

Berbagi snapshot

Anda dapat membagikan snapshot DB menggunakan AWS Management Console, AWS CLI, atau RDS API.

Konsol

Menggunakan konsol Amazon RDS, Anda dapat membagikan snapshot DB manual hingga 20. Akun AWS Anda juga dapat menggunakan konsol untuk berhenti berbagi snapshot manual dengan satu atau beberapa akun.

Untuk berbagi snapshot DB manual dengan menggunakan konsol Amazon RDS

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Snapshot.
3. Pilih snapshot manual yang ingin Anda bagikan.
4. Untuk Tindakan, pilih Bagikan snapshot.
5. Pilih salah satu opsi berikut untuk Visibilitas snapshot DB.

   - Jika sumbernya tidak terenkripsi, pilih Publik untuk mengizinkan semua AWS akun memulihkan instans DB dari snapshot DB manual Anda, atau pilih Private untuk mengizinkan hanya Akun AWS yang Anda tentukan untuk memulihkan instans DB dari snapshot DB manual Anda.

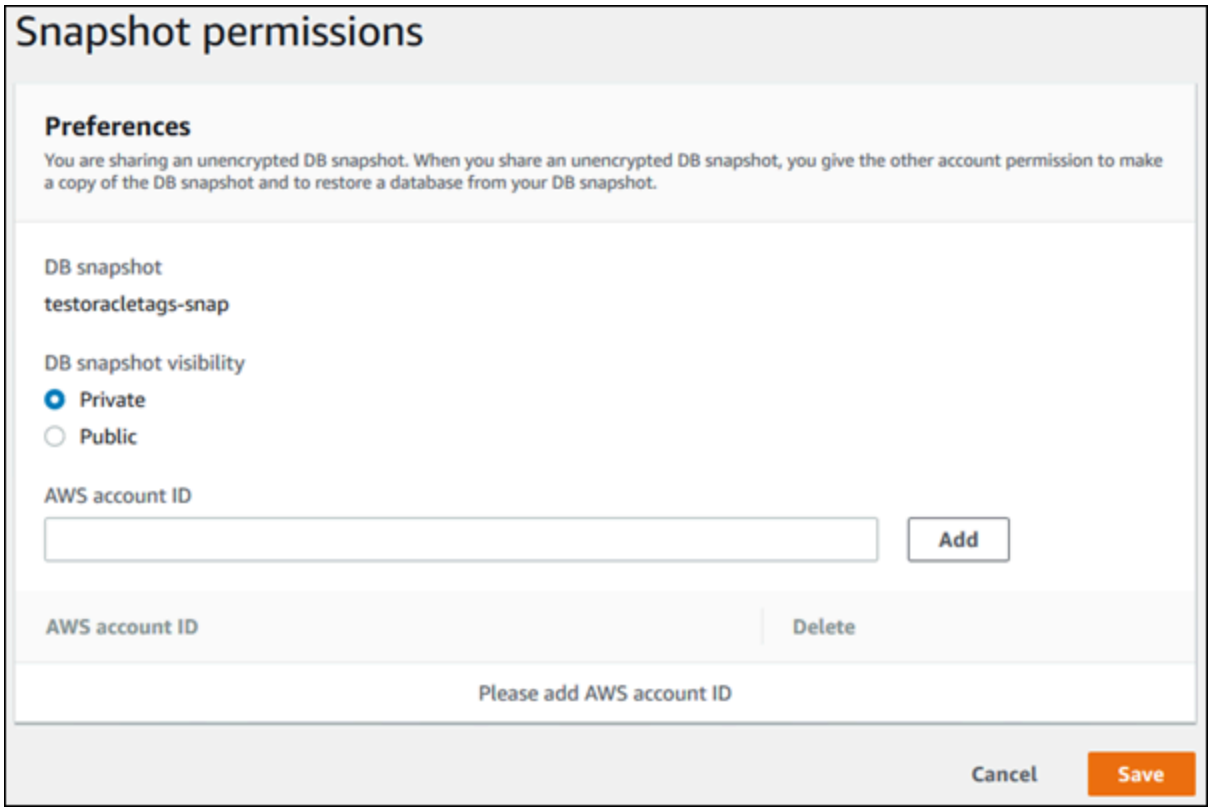
     > **Warning**
     >
     > Jika Anda menyetel visibilitas snapshot DB ke Publik, semua Akun AWS dapat memulihkan instans DB dari snapshot DB manual Anda dan memiliki akses ke data Anda. Jangan berbagi snapshot DB manual apa pun yang berisi informasi privat sebagai Publik.
     > Untuk informasi selengkapnya, lihat Berbagi snapshot publik.

   - Jika sumbernya dienkripsi, Visibilitas snapshot DB ditetapkan sebagai Privat karena snapshot terenkripsi tidak dapat dibagikan sebagai publik.

> **Note**
>
> Snapshot yang telah dienkripsi dengan default tidak AWS KMS key dapat dibagikan. Untuk informasi tentang cara mengatasi masalah ini, lihatBerbagi snapshot terenkripsi.

6. Untuk ID AWS Akun, masukkan Akun AWS pengenal untuk akun yang ingin Anda izinkan untuk memulihkan instans DB dari snapshot manual Anda, lalu pilih Tambah. Ulangi untuk menyertakan Akun AWS pengidentifikasi tambahan, hingga 20 Akun AWS.

   Jika Anda membuat kesalahan saat menambahkan Akun AWS pengenal ke daftar akun yang diizinkan, Anda dapat menghapusnya dari daftar dengan memilih Hapus di sebelah kanan Akun AWS pengidentifikasi yang salah.



7. Setelah Anda menambahkan pengenal untuk semua Akun AWS yang ingin Anda izinkan untuk memulihkan snapshot manual, pilih Simpan untuk menyimpan perubahan Anda.

AWS CLI

Untuk berbagi snapshot DB, gunakan perintah `aws rds modify-db-snapshot-attribute`. Gunakan parameter `--values-to-add` untuk menambahkan daftar ID untuk Akun AWS yang diotorisasi untuk memulihkan snapshot manual.

Example berbagi snapshot dengan satu akun

Contoh berikut memungkinkan Akun AWS identifier `123456789012` untuk mengembalikan snapshot DB bernama. `db7-snapshot`

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-snapshot-attribute \
--db-snapshot-identifier db7-snapshot \
--attribute-name restore \
--values-to-add 123456789012
```

Untuk Windows:

```
aws rds modify-db-snapshot-attribute ^
--db-snapshot-identifier db7-snapshot ^
--attribute-name restore ^
--values-to-add 123456789012
```

Example berbagi snapshot dengan beberapa akun

Contoh berikut memungkinkan dua Akun AWS pengidentifikasi, `111122223333` dan`444455556666`, untuk mengembalikan snapshot DB bernama. `manual-snapshot1`

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-snapshot-attribute \
--db-snapshot-identifier manual-snapshot1 \
--attribute-name restore \
--values-to-add {"111122223333","444455556666"}
```

Untuk Windows:

```
aws rds modify-db-snapshot-attribute ^
--db-snapshot-identifier manual-snapshot1 ^
```

```
--attribute-name restore ^
--values-to-add "[\"111122223333\",\"444455556666\"]"
```

> **Note**
>
> Saat menggunakan command prompt Windows, Anda harus meng-escape tanda kutip ganda (") dalam kode JSON dengan memberikan garis miring terbalik (\) di depannya.

Untuk membuat daftar yang Akun AWS diaktifkan untuk memulihkan snapshot, gunakan `describe-db-snapshot-attributes` AWS CLI perintah.

API RDS

Anda juga dapat berbagi snapshot DB manual dengan yang lain Akun AWS dengan menggunakan Amazon RDS API. Untuk melakukannya, panggil operasi `ModifyDBSnapshotAttribute`. Tentukan `restore` untuk`AttributeName`, dan gunakan `ValuesToAdd` parameter untuk menambahkan daftar ID untuk Akun AWS yang berwenang untuk mengembalikan snapshot manual.

Untuk membuat snapshot manual publik dan dapat dipulihkan oleh semua orang Akun AWS, gunakan nilainya. `all` Namun, berhati-hatilah untuk tidak menambahkan `all` nilai untuk setiap snapshot manual yang berisi informasi pribadi yang Anda tidak ingin tersedia untuk semua Akun AWS. Selain itu, jangan tentukan `all` untuk snapshot terenkripsi karena menjadikan snapshot tersebut berstatus publik tidak didukung.

Untuk mencantumkan semua yang Akun AWS diizinkan untuk memulihkan snapshot, gunakan operasi `DescribeDBSnapshotAttributes`API.

Berbagi snapshot publik

Anda juga dapat membagikan snapshot manual yang tidak terenkripsi sebagai publik, yang membuat snapshot tersedia untuk semua. Akun AWS Pastikan saat berbagi snapshot sebagai publik bahwa tidak ada informasi pribadi yang dimasukkan ke dalam snapshot publik.

Ketika snapshot dibagikan secara publik, ia memberikan semua Akun AWS izin untuk menyalin snapshot dan membuat instance DB darinya.

Anda tidak ditagih untuk penyimpanan cadangan snapshot publik yang dimiliki oleh akun lain. Anda hanya ditagih untuk snapshot yang Anda miliki.

Jika Anda menyalin snapshot publik, Anda memiliki salinannya. Anda ditagih untuk penyimpanan cadangan salinan snapshot Anda. Jika Anda membuat instans DB dari snapshot publik, Anda ditagih untuk instans DB tersebut. Untuk informasi harga Amazon RDS, lihat Halaman produk Amazon RDS.

Anda hanya dapat menghapus snapshot publik yang Anda miliki. Untuk menghapus snapshot bersama atau publik, pastikan untuk masuk ke Akun AWS yang memiliki snapshot.

Melihat snapshot publik yang dimiliki oleh orang lain Akun AWS

Anda dapat melihat snapshot publik yang dimiliki oleh akun lain di AWS Wilayah tertentu pada tab Publik halaman Snapshots di konsol Amazon RDS. Snapshot Anda (yang dimiliki oleh akun Anda) tidak muncul di tab ini.

Untuk melihat snapshot publik

1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Snapshot.
3. Pilih tab Publik.

   Snapshot publik muncul. Anda dapat melihat akun mana yang memiliki snapshot publik di kolom Pemilik.

   > **Note**
   >
   > Anda mungkin harus mengubah preferensi halaman, dengan memilih ikon roda gigi di kanan atas daftar Snapshot publik, untuk melihat kolom ini.

Melihat snapshot publik Anda sendiri

Anda dapat menggunakan AWS CLI perintah berikut (hanya Unix) untuk melihat snapshot publik yang dimiliki oleh Anda Akun AWS di Wilayah tertentu AWS .

```
aws rds describe-db-snapshots --snapshot-type public --include-public |
 grep account_number
```

Output ditampilkan seperti contoh berikut jika Anda memiliki snapshot publik:

```
"DBSnapshotArn": "arn:aws:rds:us-east-1:123456789012:snapshot:mysnapshot1",
```

```
"DBSnapshotArn": "arn:aws:rds:us-east-1:123456789012:snapshot:mysnapshot2",
```

> **Note**
>
> Anda mungkin melihat entri duplikat untuk `DBSnapshotIdentifier` atau `SourceDBSnapshotIdentifier`.

Berbagi snapshot publik dari versi mesin DB yang tidak digunakan lagi

Memulihkan atau menyalin snapshot publik dari versi mesin DB yang tidak digunakan lagi tidak didukung.

Mesin RDS untuk Oracle dan RDS untuk PostgreSQL DB mendukung peningkatan versi mesin snapshot DB secara langsung. Anda dapat memutakhirkan snapshot Anda, lalu membagikannya kembali secara publik. Untuk informasi selengkapnya, lihat hal berikut:

- Meng-upgrade snapshot DB Oracle
- Meng-upgrade versi mesin snapshot DB PostgreSQL

Untuk mesin DB lainnya, lakukan langkah-langkah berikut untuk membuat snapshot publik yang tidak didukung yang ada tersedia untuk dipulihkan atau disalin:

1. Tandai snapshot sebagai pribadi.
2. Pulihkan snapshot yang telah disalin.
3. Tingkatkan instans DB yang dipulihkan ke versi mesin yang didukung.
4. Buat snapshot baru.
5. Bagikan kembali snapshot secara publik.

Berbagi snapshot terenkripsi

Anda dapat berbagi snapshot DB yang telah dienkripsi "saat diam" menggunakan algoritma enkripsi AES-256, sebagaimana dijelaskan dalam Mengenkripsi sumber daya Amazon RDS.

Pembatasan berikut berlaku untuk berbagi snapshot terenkripsi:

- Anda tidak dapat membagikan snapshot terenkripsi sebagai publik.

- Anda tidak dapat membagikan snapshot Oracle atau Microsoft SQL Server yang dienkripsi menggunakan Enkripsi Data Transparan (TDE).
- Anda tidak dapat membagikan snapshot yang telah dienkripsi menggunakan kunci KMS default dari Akun AWS yang membagikan snapshot.

Untuk mengatasi masalah kunci KMS default, lakukan tugas-tugas berikut:

1. Buat kunci yang dikelola pelanggan dan berikan akses ke sana.
2. Salin dan bagikan snapshot dari akun sumber.
3. Salin snapshot bersama di akun target.

Buat kunci yang dikelola pelanggan dan berikan akses ke sana

Pertama, Anda membuat kunci KMS kustom Wilayah AWS sama dengan snapshot DB terenkripsi. Saat membuat kunci yang dikelola pelanggan, Anda memberikan akses ke sana untuk yang lain Akun AWS.

Untuk membuat kunci yang dikelola pelanggan dan memberikan akses ke sana

1. Masuk ke AWS Management Console dari sumbernya Akun AWS.
2. Buka AWS KMS konsol di https://console.aws.amazon.com/kms.
3. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
4. Di panel navigasi, pilih Kunci yang dikelola pelanggan.
5. Pilih Buat kunci.
6. Pada halaman tombol Konfigurasi:
    a. Untuk Key type, pilih Symmetric.
    b. Untuk penggunaan Kunci, pilih Enkripsi dan dekripsi.
    c. Perluas Opsi lanjutan.
    d. Untuk asal bahan utama, pilih KMS.
    e. Untuk Regionalitas, pilih kunci Wilayah Tunggal.
    f. Pilih Berikutnya.
7. Pada halaman Tambahkan label:
    a. Untuk Alias. masukkan nama tampilan untuk kunci KMS Anda, misalnya. **`share-snapshot`**

b. (Opsional) Masukkan deskripsi untuk kunci KMS Anda.

c. (Opsional) Tambahkan tag ke kunci KMS Anda.

d. Pilih Berikutnya.

8. Pada halaman Tentukan izin administratif kunci, pilih Berikutnya.
9. Pada halaman Tentukan izin penggunaan kunci:

a. Untuk Lainnya Akun AWS, pilih Tambahkan yang lain Akun AWS.

b. Masukkan ID yang Akun AWS ingin Anda berikan aksesnya.

Anda dapat memberikan akses ke beberapa Akun AWS.

c. Pilih Berikutnya.

10. Tinjau kunci KMS Anda, lalu pilih Selesai.

Salin dan bagikan snapshot dari akun sumber

Selanjutnya Anda menyalin snapshot DB sumber ke snapshot baru menggunakan kunci yang dikelola pelanggan. Kemudian Anda membagikannya dengan target Akun AWS.

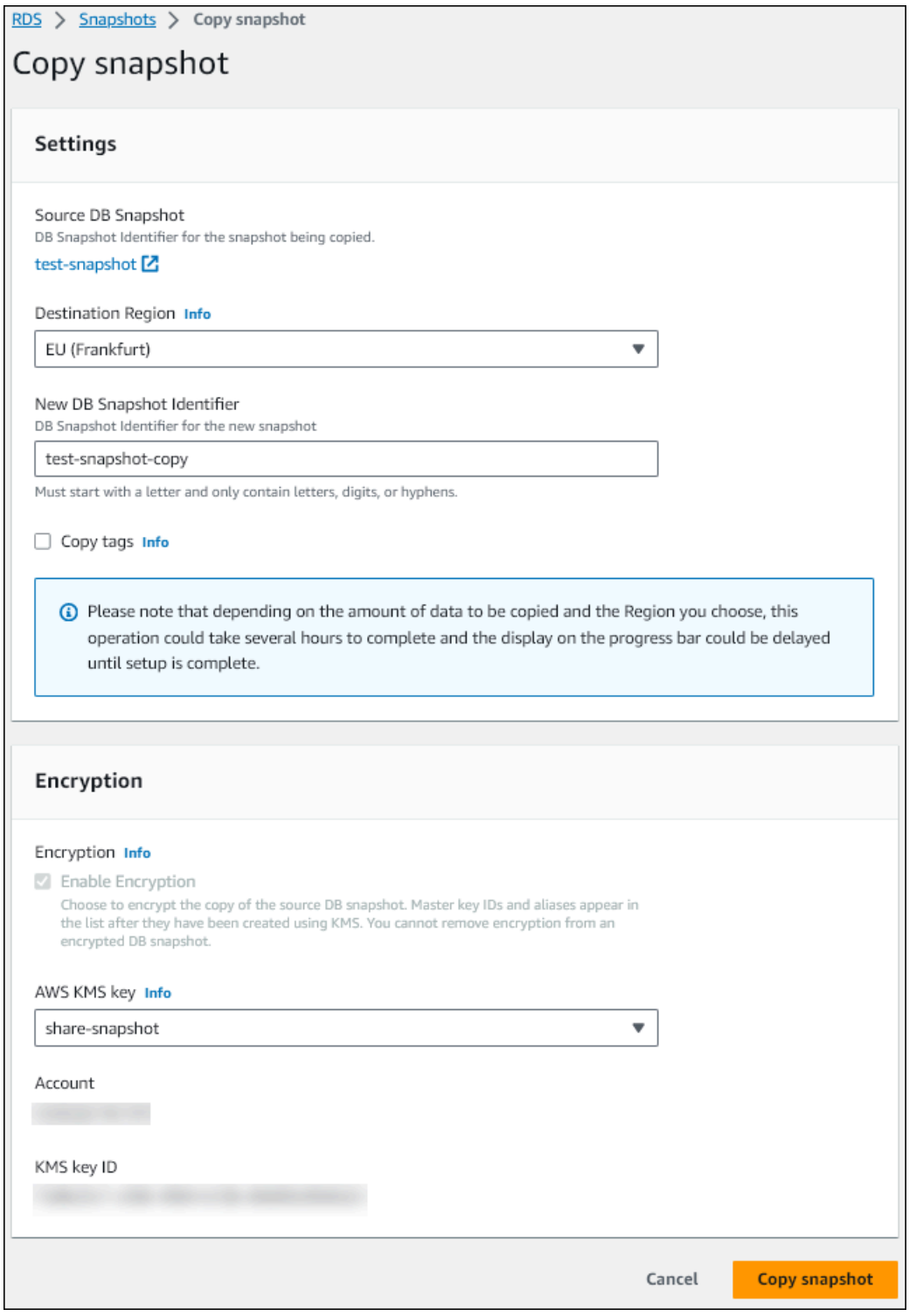
Untuk menyalin dan membagikan snapshot

1. Masuk ke AWS Management Console dari sumbernya Akun AWS.
2. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/
3. Di panel navigasi, pilih Snapshot.
4. Pilih snapshot DB yang ingin Anda salin.
5. Untuk Tindakan, pilih Salin snapshot.
6. Pada halaman Salin snapshot:

a. Untuk Wilayah Tujuan, pilih Wilayah AWS tempat Anda membuat kunci yang dikelola pelanggan di prosedur sebelumnya.

b. Masukkan nama salinan snapshot DB di New DB Snapshot Identifier.

c. Untuk AWS KMS key, pilih kunci yang dikelola pelanggan yang Anda buat.



d. Pilih Salin snapshot.

7. Ketika salinan snapshot tersedia, pilih.
8. Untuk Tindakan, pilih Bagikan snapshot.
9. Pada halaman izin Snapshot:

a. Masukkan Akun AWS ID yang Anda gunakan untuk membagikan salinan snapshot, lalu pilih Tambah.

b. Pilih Simpan.

Snapshot dibagikan.

Salin snapshot bersama di akun target

Sekarang Anda dapat menyalin snapshot bersama di target Akun AWS.

Untuk menyalin snapshot bersama

1. Masuk ke AWS Management Console dari target Akun AWS.
2. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/
3. Di panel navigasi, pilih Snapshot.
4. Pilih tab Berbagi dengan saya.
5. Pilih snapshot bersama.
6. Untuk Tindakan, pilih Salin snapshot.
7. Pilih pengaturan Anda untuk menyalin snapshot seperti pada prosedur sebelumnya, tetapi gunakan AWS KMS key yang termasuk dalam akun target.

   Pilih Salin snapshot.

Menghentikan berbagi snapshot

Untuk berhenti membagikan snapshot DB, Anda menghapus izin dari target Akun AWS.

Konsol

Untuk berhenti berbagi snapshot DB manual dengan Akun AWS

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Snapshot.
3. Pilih snapshot manual yang pembagiannya ingin Anda hentikan.

4. Pilih Tindakan, lalu pilih Bagikan snapshot.
5. Untuk menghapus izin untuk Akun AWS, pilih Hapus untuk pengenal AWS akun untuk akun tersebut dari daftar akun resmi.
6. Pilih Simpan untuk menyimpan perubahan Anda.

CLI

Untuk menghapus Akun AWS pengenal dari daftar, gunakan `--values-to-remove` parameter.

Example menghentikan berbagi snapshot

Contoh berikut mencegah Akun AWS ID 444455556666 memulihkan snapshot.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-snapshot-attribute \
--db-snapshot-identifier manual-snapshot1 \
--attribute-name restore \
--values-to-remove 444455556666
```

Untuk Windows:

```
aws rds modify-db-snapshot-attribute ^
--db-snapshot-identifier manual-snapshot1 ^
--attribute-name restore ^
--values-to-remove 444455556666
```

API RDS

Untuk menghapus izin berbagi untuk sebuah Akun AWS, gunakan `ModifyDBSnapshotAttribute`operasi dengan `AttributeName` set to `restore` dan `ValuesToRemove` parameter. Untuk menandai snapshot manual sebagai privat, hapus nilai `all` dari daftar nilai untuk atribut `restore`.

Mengekspor data snapshot DB ke Amazon S3

Anda dapat mengekspor data snapshot DB ke bucket Amazon S3. Proses ekspor berjalan di latar belakang dan tidak memengaruhi performa instans DB aktif Anda.

Saat Anda mengekspor snapshot DB, Amazon RDS mengekstrak data dari snapshot dan menyimpannya di bucket Amazon S3. Data disimpan dalam format Apache Parquet yang dikompresi dan konsisten.

Anda dapat mengekspor semua jenis snapshot DB—termasuk snapshot manual, snapshot sistem otomatis, dan snapshot yang dibuat oleh layanan. AWS Backup Secara default, semua data dalam snapshot akan diekspor. Namun, Anda dapat memilih untuk mengekspor set basis data, skema, atau tabel tertentu.

Setelah data diekspor, Anda dapat menganalisis data yang diekspor secara langsung melalui alat seperti Amazon Athena atau Amazon Redshift Spectrum. Untuk informasi lebih lanjut tentang menggunakan Athena untuk membaca data Parket, lihat Parket di Panduan Pengguna SerDe Amazon Athena. Untuk informasi lebih lanjut tentang menggunakan Redshift Spectrum untuk membaca data Parquet, lihat COPY dari format data columnar dalam Panduan Developer Basis Data Amazon Redshift.

Topik

Ketersediaan wilayah dan versi

Ketersediaan dan dukungan fitur bervariasi di seluruh versi spesifik dari setiap mesin basis data dan di seluruh Wilayah AWS. Untuk mengetahui informasi selengkapnya tentang versi dan ketersediaan Wilayah pengeksporan snapshot ke S3, lihat Ekspor snapshot ke S3.

Batasan

Mengekspor data snapshot DB ke Amazon S3 memiliki batasan sebagai berikut:

- Anda tidak dapat menjalankan beberapa tugas ekspor untuk snapshot DB yang sama secara bersamaan. Ini berlaku untuk ekspor penuh dan sebagian.
- Mengekspor snapshot dari instans DB yang menggunakan penyimpanan magnetik tidak didukung.
- Ekspor ke S3 tidak mendukung awalan S3 yang berisi titik dua (:).
- Karakter berikut di jalur file S3 akan diubah menjadi garis bawah (_) selama ekspor berlangsung:

```
\ ` " (space)
```

- Jika basis data, skema, atau tabel memiliki karakter dalam namanya selain yang berikut ini, maka ekspor parsial tidak didukung. Namun, Anda dapat mengekspor seluruh snapshot DB.
  - Huruf latin (A-Z)
  - Digit (0–9)
  - Simbol dolar ($)
  - Garis bawah (_)
- Spasi ( ) dan karakter-karakter tertentu tidak didukung dalam nama kolom tabel basis data. Tabel yang nama kolomnya berisi karakter berikut akan dilewati selama ekspor berlangsung:

```
, ; { } ( ) \n \t = (space)
```

- Tabel dengan garis miring (/) di namanya dilewati selama ekspor.
- Tabel sementara dan yang tidak tercatat RDS for PostgreSQL dilewati selama ekspor.
- Jika data berisi objek besar, seperti BLOB atau CLOB, yang mendekati atau lebih dari 500 MB, ekspor tersebut gagal.
- Jika suatu tabel berisi baris besar yang berukuran mendekati atau lebih dari 2 GB, maka tabel tersebut akan dilewati selama ekspor berlangsung.

- Sebaiknya Anda menggunakan nama unik untuk setiap tugas ekspor. Jika tidak menggunakan nama tugas yang unik, Anda mungkin menerima pesan kesalahan berikut:

  ExportTaskAlreadyExistsFault: Terjadi kesalahan (ExportTaskAlreadyExists) saat memanggil StartExportTask operasi: Tugas ekspor dengan ID *xxxxx* sudah ada.
- Anda dapat menghapus snapshot saat mengekspor datanya ke S3, tetapi Anda masih dikenakan biaya penyimpanan untuk snapshot tersebut hingga tugas ekspor selesai.
- Anda tidak dapat memulihkan data snapshot yang diekspor dari S3 ke instans DB baru.

Ringkasan pengeksporan data snapshot

Anda menggunakan proses berikut untuk mengekspor data snapshot DB ke bucket Amazon S3. Untuk detail selengkapnya, lihat bagian berikut.

1. Identifikasi snapshot yang akan diekspor.

   Gunakan snapshot otomatis atau manual yang ada, atau buat snapshot manual instans DB.
2. Siapkan akses ke bucket Amazon S3.

   Bucket adalah kontainer untuk objek atau file Amazon S3. Untuk memberikan informasi agar dapat mengakses bucket, lakukan langkah-langkah berikut:

   a. Identifikasi bucket S3 tempat snapshot akan diekspor. Bucket S3 harus berada di AWS Wilayah yang sama dengan snapshot. Untuk informasi selengkapnya, lihat Mengidentifikasi bucket Amazon S3 untuk ekspor.

   b. Buat peran AWS Identity and Access Management (IAM) yang memberikan akses tugas ekspor snapshot ke bucket S3. Untuk informasi selengkapnya, lihat Memberikan akses ke bucket Amazon S3 menggunakan peran IAM.
3. Buat enkripsi simetris AWS KMS key untuk enkripsi sisi server. Kunci KMS digunakan oleh tugas ekspor snapshot untuk mengatur enkripsi AWS KMS sisi server saat menulis data ekspor ke S3.

   Kebijakan kunci KMS harus menyertakan izin `kms:CreateGrant` dan `kms:DescribeKey`. Untuk informasi selengkapnya tentang menggunakan kunci KMS di Amazon RDS, lihat Manajemen AWS KMS key.

   Jika Anda memiliki pernyataan penolakan dalam kebijakan kunci KMS Anda, pastikan untuk secara eksplisit mengecualikan prinsip layanan. AWS `export.rds.amazonaws.com`

Anda dapat menggunakan kunci KMS dalam AWS akun Anda, atau Anda dapat menggunakan kunci KMS lintas akun. Untuk informasi selengkapnya, lihat Menggunakan akun silang AWS KMS key untuk mengenkripsi ekspor Amazon S3.

4. Ekspor snapshot ke Amazon S3 menggunakan konsol atau perintah CLI `start-export-task`. Untuk informasi selengkapnya, lihat Mengekspor data snapshot DB ke bucket Amazon S3.
5. Untuk mengakses data Anda yang diekspor di bucket Amazon S3 lihat Mengunggah, mengunduh, dan mengelola objek dalam Panduan Pengguna Amazon Simple Storage Service.

Menyiapkan akses ke bucket Amazon S3

Untuk mengekspor data snapshot DB ke file Amazon S3 Anda terlebih dahulu memberikan izin snapshot untuk mengakses bucket Amazon S3. Kemudian, Anda membuat peran IAM untuk memungkinkan layanan Amazon RDS menulis ke bucket Amazon S3.

Topik

- Mengidentifikasi bucket Amazon S3 untuk ekspor
- Memberikan akses ke bucket Amazon S3 menggunakan peran IAM
- Menggunakan bucket Amazon S3 lintas akun
- Menggunakan akun silang AWS KMS key untuk mengenkripsi ekspor Amazon S3

Mengidentifikasi bucket Amazon S3 untuk ekspor

Identifikasi bucket Amazon S3 untuk mengekspor snapshot DB. Gunakan bucket S3 yang ada atau buat bucket S3 baru.

> **Note**
>
> Bucket S3 yang akan diekspor harus berada di AWS Wilayah yang sama dengan snapshot.

Untuk informasi selengkapnya tentang cara bekerja dengan bucket Amazon S3, lihat informasi berikut dalam Panduan Pengguna Amazon Simple Storage Service:

- Bagaimana cara melihat properti untuk bucket S3?
- Bagaimana cara mengaktifkan enkripsi default untuk bucket Amazon S3?

- Bagaimana cara membuat bucket S3?

Memberikan akses ke bucket Amazon S3 menggunakan peran IAM

Sebelum Anda mengekspor data snapshot DB ke Amazon S3, beri tugas ekspor snapshot izin akses tulis ke bucket Amazon S3.

Untuk memberikan izin ini, buat kebijakan IAM yang memberikan akses ke bucket, lalu buat peran IAM dan lampirkan kebijakan ke peran tersebut. Kemudian, tetapkan peran IAM ke tugas ekspor snapshot Anda.

> **Important**
>
> Jika Anda berencana untuk menggunakan AWS Management Console untuk mengekspor snapshot Anda, Anda dapat memilih untuk membuat kebijakan IAM dan peran secara otomatis ketika Anda mengekspor snapshot. Untuk mendapatkan petunjuk, lihat Mengekspor data snapshot DB ke bucket Amazon S3.

Untuk memberi tugas snapshot DB akses ke Amazon S3

1. Buat kebijakan IAM. Kebijakan ini memberikan bucket dan izin objek yang memungkinkan tugas ekspor snapshot Anda mengakses Amazon S3.

   Dalam kebijakan tersebut, sertakan tindakan yang diperlukan berikut untuk mengizinkan transfer file dari Amazon RDS ke bucket S3:

   - `s3:PutObject*`
   - `s3:GetObject*`
   - `s3:ListBucket`
   - `s3:DeleteObject*`
   - `s3:GetBucketLocation`

   Dalam kebijakan tersebut, sertakan sumber daya berikut untuk mengidentifikasi bucket S3 dan objek dalam bucket. Daftar sumber daya berikut menunjukkan format Amazon Resource Name (ARN) untuk mengakses Amazon S3.

   - `arn:aws:s3:::`*`your-s3-bucket`*

- `arn:aws:s3:::your-s3-bucket/*`

Untuk mengetahui informasi selengkapnya tentang cara membuat kebijakan IAM untuk Amazon RDS, lihat Membuat dan menggunakan kebijakan IAM untuk akses basis data IAM. Lihat juga Tutorial: Membuat dan melampirkan kebijakan yang dikelola pelanggan pertama Anda di Panduan Pengguna IAM.

AWS CLI Perintah berikut membuat kebijakan IAM bernama `ExportPolicy` dengan opsi ini. Perintah ini akan memberikan akses ke bucket bernama `your-s3-bucket`.

> **Note**
>
> Setelah Anda membuat kebijakan, catat ARN kebijakan tersebut. Anda memerlukan ARN ini untuk langkah berikutnya ketika Anda melampirkan kebijakan ke peran IAM.

```
aws iam create-policy  --policy-name ExportPolicy --policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ExportPolicy",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject*",
                "s3:ListBucket",
                "s3:GetObject*",
                "s3:DeleteObject*",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::your-s3-bucket",
                "arn:aws:s3:::your-s3-bucket/*"
            ]
        }
    ]
}'
```

2. Buat peran IAM, sehingga Amazon RDS dapat mengambil peran IAM ini atas nama Anda untuk mengakses bucket Amazon S3. Lihat mengetahui informasi yang lebih lengkap di Membuat peran untuk melimpahkan izin ke pengguna IAM dalam Panduan Pengguna IAM.

Contoh berikut menunjukkan menggunakan AWS CLI perintah untuk membuat peran bernama`rds-s3-export-role`.

```
aws iam create-role  --role-name rds-s3-export-role  --assume-role-policy-document
 '{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
           "Service": "export.rds.amazonaws.com"
         },
        "Action": "sts:AssumeRole"
      }
    ]
  }'
```

3. Lampirkan kebijakan IAM yang Anda buat pada peran IAM yang Anda buat.

   AWS CLI Perintah berikut melampirkan kebijakan yang dibuat sebelumnya ke peran bernama`rds-s3-export-role`. Ganti *`your-policy-arn`* dengan ARN kebijakan yang Anda catat di langkah sebelumnya.

```
aws iam attach-role-policy  --policy-arn your-policy-arn  --role-name rds-s3-
export-role
```

Menggunakan bucket Amazon S3 lintas akun

Anda dapat menggunakan bucket Amazon S3 di seluruh akun. AWS Untuk menggunakan bucket lintas akun, tambahkan kebijakan bucket untuk mengizinkan akses ke peran IAM yang Anda gunakan untuk ekspor S3. Untuk informasi selengkapnya, lihat Contoh 2: Pemilik bucket yang memberikan izin bucket lintas akun.

- Lampirkan kebijakan bucket pada bucket Anda, seperti yang ditunjukkan dalam contoh berikut.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
```

```
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:role/Admin"
            },
            "Action": [
                "s3:PutObject*",
                "s3:ListBucket",
                "s3:GetObject*",
                "s3:DeleteObject*",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::mycrossaccountbucket",
                "arn:aws:s3:::mycrossaccountbucket/*"
            ]
        }
    ]
}
```

Menggunakan akun silang AWS KMS key untuk mengenkripsi ekspor Amazon S3

Anda dapat menggunakan akun silang AWS KMS key untuk mengenkripsi ekspor Amazon S3. Pertama-tama, tambahkan kebijakan kunci ke akun lokal, lalu tambahkan kebijakan IAM di akun eksternal. Untuk informasi selengkapnya, lihat Mengizinkan pengguna di akun lain untuk menggunakan kunci KMS.

Untuk menggunakan kunci KMS lintas akun

1. Tambahkan kebijakan kunci ke akun lokal.

   Contoh berikut memberi `ExampleRole` dan `ExampleUser` di akun eksternal 444455556666 izin di akun lokal 123456789012.

```
{
    "Sid": "Allow an external account to use this KMS key",
    "Effect": "Allow",
    "Principal": {
        "AWS": [
            "arn:aws:iam::444455556666:role/ExampleRole",
            "arn:aws:iam::444455556666:user/ExampleUser"
        ]
    },
```

```
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:CreateGrant",
        "kms:DescribeKey",
        "kms:RetireGrant"
    ],
    "Resource": "*"
}
```

2. Tambahkan kebijakan IAM ke akun eksternal tersebut.

   Contoh kebijakan IAM berikut memungkinkan pengguna utama menggunakan kunci KMS di akun 123456789012 untuk operasi kriptografi. Untuk memberikan izin ini ke `ExampleRole` dan `ExampleUser` di akun 444455556666, lampirkan kebijakan di akun tersebut.

```
{
    "Sid": "Allow use of KMS key in account 123456789012",
    "Effect": "Allow",
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:CreateGrant",
        "kms:DescribeKey",
        "kms:RetireGrant"
    ],
    "Resource": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

Mengekspor data snapshot DB ke bucket Amazon S3

Anda dapat memiliki hingga lima tugas ekspor snapshot DB bersamaan yang sedang berlangsung per. Akun AWS

> **Note**
>
> Mengekspor snapshot RDS dapat memerlukan banyak waktu, tergantung pada jenis dan ukuran basis data Anda. Tugas ekspor akan terlebih dahulu memulihkan dan menskalakan seluruh basis data sebelum mengekstrak data ke Amazon S3. Dalam fase ini, progres tugas tersebut akan ditampilkan sebagai Memulai. Saat tugas beralih menjadi mengekspor data ke S3, progres akan ditampilkan sebagai Sedang berlangsung.
> Waktu yang diperlukan untuk menyelesaikan ekspor tergantung pada data yang disimpan di basis data. Misalnya, tabel yang berisi kolom indeks atau kunci primer numerik yang terdistribusi dengan baik akan diekspor paling cepat. Tabel yang tidak berisi kolom yang sesuai untuk partisi dan tabel yang hanya berisi satu indeks pada kolom berbasis string memerlukan waktu lebih lama. Waktu ekspor yang lebih lama ini terjadi karena ekspor menggunakan proses alur tunggal yang lebih lambat.

Anda dapat mengekspor snapshot DB ke Amazon S3 menggunakan, AWS Management Console API, AWS CLI atau RDS.

Jika Anda menggunakan fungsi Lambda untuk mengekspor snapshot, tambahkan tindakan `kms:DescribeKey` ke kebijakan fungsi Lambda. Untuk informasi selengkapnya, lihat izin AWS Lambda.

Konsol

Opsi konsol Ekspor ke Amazon S3 hanya muncul untuk snapshot yang dapat diekspor ke Amazon S3. Snapshot mungkin tidak tersedia untuk diekspor karena alasan berikut:

- Mesin DB tidak didukung untuk ekspor S3.
- Versi instans DB tidak didukung untuk ekspor S3.
- Ekspor S3 tidak didukung di AWS Wilayah tempat snapshot dibuat.

Untuk mengekspor snapshot DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Snapshot.
3. Dari tabnya, pilih jenis snapshot yang ingin Anda ekspor.

4. Dalam daftar snapshot, pilih snapshot yang ingin Anda ekspor.
5. Untuk Tindakan, pilih Ekspor ke Amazon S3.

   Jendela Ekspor ke Amazon S3 akan muncul.
6. Untuk Pengidentifikasi ekspor, masukkan nama untuk mengidentifikasi tugas ekspor. Nilai ini juga akan digunakan untuk nama file yang dibuat di bucket S3.
7. Pilih data yang akan diekspor:

   - Pilih Semua untuk mengekspor semua data dalam snapshot.
   - Pilih Sebagian untuk mengekspor bagian tertentu dari snapshot. Untuk mengidentifikasi bagian snapshot yang akan diekspor, masukkan satu atau beberapa basis data, skema, atau tabel untuk Pengidentifikasi, dipisahkan dengan spasi.

     Gunakan format berikut:

     ```
     database[.schema][.table] database2[.schema2][.table2] ... databasen[.scheman]
     [.tablen]
     ```

     Contohnya:

     ```
     mydatabase mydatabase2.myschema1 mydatabase2.myschema2.mytable1
      mydatabase2.myschema2.mytable2
     ```
8. Untuk Bucket S3, pilih bucket yang akan dijadikan tujuan ekspor.

   Untuk menetapkan data yang diekspor ke jalur folder dalam bucket S3, masukkan jalur opsional untuk Prefiks S3.
9. Untuk Peran IAM, pilih peran yang memberi Anda akses tulis ke bucket S3 yang Anda pilih, atau buat peran baru.

   - Jika Anda membuat peran dengan mengikuti langkah-langkah di Memberikan akses ke bucket Amazon S3 menggunakan peran IAM, pilih peran tersebut.
   - Jika Anda tidak membuat peran yang memberi Anda akses tulis ke bucket S3 yang Anda pilih, pilih Buat peran baru untuk membuat peran secara otomatis. Berikutnya, masukkan nama untuk peran tersebut di Nama peran IAM.
10. Untuk AWS KMS key, masukkan ARN untuk kunci yang akan digunakan untuk mengenkripsi data yang diekspor.

11. Pilih Ekspor ke Amazon S3.

AWS CLI

Untuk mengekspor snapshot DB ke Amazon S3 menggunakan AWS CLI, gunakan perintah dengan start-export-taskopsi yang diperlukan berikut:

- `--export-task-identifier`
- `--source-arn`
- `--s3-bucket-name`
- `--iam-role-arn`
- `--kms-key-id`

Dalam contoh berikut, tugas ekspor snapshot diberi nama *`my-snapshot-export`*, yang mengekspor snapshot ke bucket S3 bernama. *`my-export-bucket`*

Example

Untuk Linux, macOS, atau Unix:

```
aws rds start-export-task \
    --export-task-identifier my-snapshot-export \
    --source-arn arn:aws:rds:AWS_Region:123456789012:snapshot:snapshot-name \
    --s3-bucket-name my-export-bucket \
    --iam-role-arn iam-role \
    --kms-key-id my-key
```

Untuk Windows:

```
aws rds start-export-task ^
    --export-task-identifier my-snapshot-export ^
    --source-arn arn:aws:rds:AWS_Region:123456789012:snapshot:snapshot-name ^
    --s3-bucket-name my-export-bucket ^
    --iam-role-arn iam-role ^
    --kms-key-id my-key
```

Berikut adalah contoh output.

```
{
```

```
    "Status": "STARTING",
    "IamRoleArn": "iam-role",
    "ExportTime": "2019-08-12T01:23:53.109Z",
    "S3Bucket": "my-export-bucket",
    "PercentProgress": 0,
    "KmsKeyId": "my-key",
    "ExportTaskIdentifier": "my-snapshot-export",
    "TotalExtractedDataInGB": 0,
    "TaskStartTime": "2019-11-13T19:46:00.173Z",
    "SourceArn": "arn:aws:rds:AWS_Region:123456789012:snapshot:snapshot-name"
}
```

Untuk menyediakan jalur folder di bucket S3 untuk ekspor snapshot, sertakan `--s3-prefix` opsi dalam perintah. start-export-task

API RDS

Untuk mengekspor snapshot DB ke Amazon S3 menggunakan Amazon RDS API, gunakan operasi dengan parameter StartExportTaskyang diperlukan berikut:

- `ExportTaskIdentifier`
- `SourceArn`
- `S3BucketName`
- `IamRoleArn`
- `KmsKeyId`

Memantau ekspor snapshot

Anda dapat memantau ekspor snapshot DB menggunakan AWS Management Console, AWS CLI, atau RDS API.

Konsol

Untuk memantau ekspor snapshot DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Snapshot.

3. Untuk melihat daftar ekspor snapshot, pilih tab Ekspor di Amazon S3.
4. Untuk melihat informasi tentang ekspor snapshot khusus, pilih tugas ekspor.

AWS CLI

Untuk memantau ekspor snapshot DB menggunakan AWS CLI, gunakan perintah. describe-export-tasks

Contoh berikut menunjukkan cara menampilkan informasi saat ini tentang semua ekspor snapshot Anda.

Example

```
aws rds describe-export-tasks

{
    "ExportTasks": [
        {
            "Status": "CANCELED",
            "TaskEndTime": "2019-11-01T17:36:46.961Z",
            "S3Prefix": "something",
            "ExportTime": "2019-10-24T20:23:48.364Z",
            "S3Bucket": "examplebucket",
            "PercentProgress": 0,
            "KmsKeyId": "arn:aws:kms:AWS_Region:123456789012:key/K7MDENG/
bPxRfiCYEXAMPLEKEY",
            "ExportTaskIdentifier": "anewtest",
            "IamRoleArn": "arn:aws:iam::123456789012:role/export-to-s3",
            "TotalExtractedDataInGB": 0,
            "TaskStartTime": "2019-10-25T19:10:58.885Z",
            "SourceArn": "arn:aws:rds:AWS_Region:123456789012:snapshot:parameter-
groups-test"
        },
{
            "Status": "COMPLETE",
            "TaskEndTime": "2019-10-31T21:37:28.312Z",
            "WarningMessage": "{\"skippedTables\":[],\"skippedObjectives\":[],\"general
\":[{\"reason\":\"FAILED_TO_EXTRACT_TABLES_LIST_FOR_DATABASE\"}]}",
            "S3Prefix": "",
            "ExportTime": "2019-10-31T06:44:53.452Z",
            "S3Bucket": "examplebucket1",
            "PercentProgress": 100,
```

```
            "KmsKeyId": "arn:aws:kms:AWS_Region:123456789012:key/2Zp9Utk/
h3yCo8nvbEXAMPLEKEY",
            "ExportTaskIdentifier": "thursday-events-test",
            "IamRoleArn": "arn:aws:iam::123456789012:role/export-to-s3",
            "TotalExtractedDataInGB": 263,
            "TaskStartTime": "2019-10-31T20:58:06.998Z",
            "SourceArn":
 "arn:aws:rds:AWS_Region:123456789012:snapshot:rds:example-1-2019-10-31-06-44"
        },
        {
            "Status": "FAILED",
            "TaskEndTime": "2019-10-31T02:12:36.409Z",
            "FailureCause": "The S3 bucket edgcuc-export isn't located in the current
 AWS Region. Please, review your S3 bucket name and retry the export.",
            "S3Prefix": "",
            "ExportTime": "2019-10-30T06:45:04.526Z",
            "S3Bucket": "examplebucket2",
            "PercentProgress": 0,
            "KmsKeyId": "arn:aws:kms:AWS_Region:123456789012:key/2Zp9Utk/
h3yCo8nvbEXAMPLEKEY",
            "ExportTaskIdentifier": "wednesday-afternoon-test",
            "IamRoleArn": "arn:aws:iam::123456789012:role/export-to-s3",
            "TotalExtractedDataInGB": 0,
            "TaskStartTime": "2019-10-30T22:43:40.034Z",
            "SourceArn":
 "arn:aws:rds:AWS_Region:123456789012:snapshot:rds:example-1-2019-10-30-06-45"
        }
    ]
}
```

Untuk menampilkan informasi tentang ekspor snapshot tertentu, sertakan opsi `--export-task-identifier` dengan perintah `describe-export-tasks`. Sertakan opsi `--Filters` untuk memfilter output. Untuk opsi lainnya, lihat describe-export-tasksperintah.

API RDS

Untuk menampilkan informasi tentang ekspor snapshot DB menggunakan Amazon RDS API, gunakan operasi. DescribeExportTasks

Untuk melacak penyelesaian alur kerja ekspor atau memulai alur kerja lainnya, Anda dapat berlangganan topik Amazon Simple Notification Service. Untuk informasi selengkapnya tentang Amazon SNS, lihat Menggunakan pemberitahuan peristiwa Amazon RDS.

Membatalkan tugas ekspor snapshot

Anda dapat membatalkan tugas ekspor snapshot DB menggunakan AWS Management Console, AWS CLI, atau RDS API.

> **Note**
>
> Membatalkan tugas ekspor snapshot tidak akan menghapus data apa pun yang telah diekspor ke Amazon S3. Untuk informasi tentang cara menghapus data menggunakan konsol, lihat Bagaimana cara menghapus objek dari bucket S3? Untuk menghapus data menggunakan CLI, gunakan perintah delete-object.

Konsol

Untuk membatalkan tugas ekspor snapshot

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Snapshot.
3. Pilih tab Ekspor di Amazon S3.
4. Pilih tugas ekspor snapshot yang ingin Anda batalkan.
5. Pilih Batalkan.
6. Pilih Batalkan tugas ekspor di halaman konfirmasi.

AWS CLI

Untuk membatalkan tugas ekspor snapshot menggunakan AWS CLI, gunakan cancel-export-taskperintah. Perintah tersebut memerlukan opsi `--export-task-identifier`.

Example

```
aws rds cancel-export-task --export-task-identifier my_export
{
    "Status": "CANCELING",
    "S3Prefix": "",
    "ExportTime": "2019-08-12T01:23:53.109Z",
    "S3Bucket": "examplebucket",
```

```
    "PercentProgress": 0,
    "KmsKeyId": "arn:aws:kms:AWS_Region:123456789012:key/K7MDENG/bPxRfiCYEXAMPLEKEY",
    "ExportTaskIdentifier": "my_export",
    "IamRoleArn": "arn:aws:iam::123456789012:role/export-to-s3",
    "TotalExtractedDataInGB": 0,
    "TaskStartTime": "2019-11-13T19:46:00.173Z",
    "SourceArn": "arn:aws:rds:AWS_Region:123456789012:snapshot:export-example-1"
}
```

API RDS

Untuk membatalkan tugas ekspor snapshot menggunakan Amazon RDS API, gunakan CancelExportTaskoperasi dengan parameter. `ExportTaskIdentifier`

Pesan kegagalan untuk tugas ekspor Amazon S3

Tabel berikut menjelaskan pesan yang akan ditampilkan jika tugas ekspor Amazon S3 gagal.

Pesan kegagalan	Deskripsi
Terjadi kesalahan internal yang tidak diketahui.	Tugas telah gagal karena kesalahan yang tidak diketahui, pengecualian, atau kegagalan.
Terjadi kesalahan internal yang tidak diketahui saat menulis metadata tugas ekspor ke bucket S3 [nama bucket].	Tugas telah gagal karena kesalahan yang tidak diketahui, pengecualian, atau kegagalan.
Ekspor RDS gagal menulis metadata tugas ekspor karena tidak dapat mengambil peran IAM [peran ARN].	Tugas ekspor mengambil peran IAM Anda untuk memvalidasi apakah tugas tersebut diperbolehkan menulis metadata ke bucket S3 Anda. Jika tidak dapat mengambil peran IAM Anda, berarti tugas tersebut gagal.
Ekspor RDS gagal menulis metadata tugas ekspor ke bucket S3 [nama bucket] menggunakan peran IAM [ARN peran] dengan kunci KMS [ID kunci]. Kode kesalahan: [kode kesalahan]	Satu atau beberapa izin tidak ada, sehingga tugas ekspor tidak dapat mengakses bucket S3. Pesan kegagalan ini muncul saat menerima salah satu kode kesalahan berikut: • `AWSSecurityTokenServiceException` dengan kode kesalahan `AccessDenied`

Pesan kegagalan	Deskripsi
	• `AmazonS3Exception` dengan kode kesalahan `NoSuchBucket`, `AccessDenied`, `KMS.KMSInvalidStateException`, `403 Forbidden`, atau `KMS.DisabledException` Kode kesalahan ini menunjukkan pengaturan salah konfigurasi untuk peran IAM, bucket S3, atau kunci KMS.
Peran IAM [role ARN] tidak diizinkan untuk memanggil [tindakan S3] pada bucket S3 [nama bucket]. Tinjau izin Anda dan coba lagi ekspor.	Kebijakan IAM salah dikonfigurasi. Izin untuk tindakan S3 tertentu pada bucket S3 tidak ada, sehingga menyebabkan tugas ekspor gagal.
Pemeriksaan kunci KMS gagal. Periksa kredensial pada kunci KMS Anda, lalu coba lagi.	Pemeriksaan kredensial kunci KMS gagal.
Pemeriksaan kredensial S3 gagal. Periksa izin pada bucket S3 dan kebijakan IAM Anda.	Pemeriksaan kredensial S3 gagal.
Bucket S3 [nama bucket] tidak valid. Baik tidak terletak di Wilayah AWS saat ini maupun tidak ada. Tinjau nama bucket S3 Anda dan coba lagi ekspor.	Bucket S3 tidak valid.
Bucket S3 [nama bucket] tidak terletak di AWS Wilayah saat ini. Tinjau nama bucket S3 Anda, lalu coba lagi ekspor.	Bucket S3 berada di AWS Wilayah yang salah.

Memecahkan masalah kesalahan izin PostgreSQL

Saat mengekspor basis data PostgreSQL ke Amazon S3, Anda mungkin melihat kesalahan `PERMISSIONS_DO_NOT_EXIST` yang menyatakan bahwa tabel tertentu dilewati. Kesalahan ini

biasanya terjadi saat pengguna super, yang Anda tetapkan saat membuat instans DB, tidak memiliki izin untuk mengakses tabel tersebut.

Untuk memperbaiki kesalahan ini, jalankan perintah berikut:

```
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA schema_name TO superuser_name
```

Untuk informasi selengkapnya tentang hak pengguna super, lihat Hak akses akun pengguna master.

Konvensi penamaan file

Data yang diekspor untuk tabel tertentu disimpan dalam format *base_prefix*/*files*, dengan prefiks dasar sebagai berikut:

```
export_identifier/database_name/schema_name.table_name/
```

Contohnya:

```
export-1234567890123-459/rdststdb/rdststdb.DataInsert_7ADB5D19965123A2/
```

Ada dua konvensi cara penamaan file.

- Konvensi saat ini:

  ```
  batch_index/part-partition_index-random_uuid.format-based_extension
  ```

  Indeks batch adalah nomor urut yang mewakili batch data yang dibaca dari tabel. Jika kami tidak dapat mempartisi tabel Anda menjadi bagian-bagian kecil untuk diekspor secara paralel, akan ada beberapa indeks batch. Hal yang sama akan terjadi jika tabel Anda dipartisi menjadi beberapa tabel. Beberapa indeks batch akan tersedia, dengan satu untuk setiap partisi tabel dari tabel utama Anda.

  Jika kami dapat mempartisi tabel Anda menjadi bagian-bagian kecil yang akan dibaca secara paralel, hanya akan ada folder 1 indeks batch.

  Di dalam folder indeks batch, akan ada satu atau beberapa file Parquet yang berisi data tabel Anda. Prefiks file Parket adalah `part-`*partition_index*. Jika tabel Anda dipartisi, akan ada beberapa file yang diawali dengan indeks partisi `00000`.

Mungkin ada kesenjangan dalam urutan indeks partisi. Hal ini terjadi karena setiap partisi diperoleh dari kueri dengan rentang di tabel Anda. Jika tidak ada data dalam rentang partisi tersebut, maka nomor urut itu akan dilewati.

Misalnya, anggap kolom `id` adalah kunci primer tabel, dan nilai minimum dan maksimumnya adalah `100` dan `1000`. Saat kami mencoba mengekspor tabel ini dengan sembilan partisi, kami membacanya dengan kueri paralel seperti berikut:

```
SELECT * FROM table WHERE id <= 100 AND id < 200
SELECT * FROM table WHERE id <= 200 AND id < 300
```

Partisi ini akan menghasilkan sembilan file, dari `part-00000-`*`random_uuid`*`.gz.parquet` hingga `part-00008-`*`random_uuid`*`.gz.parquet`. Namun, jika tidak ada baris dengan ID antara `200` dan `350`, maka salah satu partisi yang telah selesai akan kosong, dan tidak ada file yang dibuat untuk partisi itu. Dalam contoh sebelumnya, `part-00001-`*`random_uuid`*`.gz.parquet` tidak dibuat.

- Konvensi yang lebih lama:

```
part-partition_index-random_uuid.format-based_extension
```

Konvensi ini sama seperti konvensi saat ini, tetapi tanpa prefiks *`batch_index`*, contohnya:

```
part-00000-c5a881bb-58ff-4ee6-1111-b41ecff340a3-c000.gz.parquet
part-00001-d7a881cc-88cc-5ab7-2222-c41ecab340a4-c000.gz.parquet
part-00002-f5a991ab-59aa-7fa6-3333-d41eccd340a7-c000.gz.parquet
```

Konvensi penamaan file dapat berubah sewaktu-waktu. Oleh karena itu, saat membaca tabel target, sebaiknya baca segala sesuatu di dalam prefiks dasar untuk tabel tersebut.

Konversi data saat mengekspor ke bucket Amazon S3

Saat Anda mengekspor snapshot DB ke bucket Amazon S3, Amazon RDS mengonversi data ke, mengekspor data dalam, dan menyimpan data dalam format Parquet. Untuk mengetahui informasi selengkapnya tentang Parquet, lihat situs web Parquet Apache.

Parquet menyimpan semua data sebagai salah satu jenis primitif berikut:

- BOOLEAN
- INT32
- INT64
- INT96
- FLOAT
- DOUBLE
- BYTE_ARRAY – Array byte dengan panjang variabel, juga dikenal sebagai biner
- FIXED_LEN_BYTE_ARRAY – Array byte dengan panjang tetap yang digunakan saat nilai memiliki ukuran konstan

Jenis data Parquet berjumlah sedikit untuk mengurangi kerumitan membaca dan menulis format. Parquet menyediakan jenis logis untuk memperluas jenis primitif. Jenis logis diimplementasikan sebagai anotasi dengan data di kolom metadata `LogicalType`. Anotasi jenis logis menjelaskan cara menginterpretasikan jenis primitif.

Saat jenis logis `STRING` yang menandai jenis `BYTE_ARRAY`, menunjukkan bahwa rangkaian byte harus diinterpretasikan sebagai string karakter yang dienkode UTF-8. Setelah tugas ekspor selesai, Amazon RDS memberi tahu Anda jika terjadi konversi string. Data ekspor yang mendasarinya selalu sama dengan data dari sumber. Namun, karena perbedaan encoding dalam UTF-8, beberapa karakter mungkin terlihat berbeda dari sumber saat dibaca di alat seperti Athena.

Untuk mengetahui informasi selengkapnya, lihat Parquet logical type definitions dalam dokumentasi Parquet.

Topik

- Pemetaan jenis data MySQL dan MariaDB ke Parquet
- Pemetaan jenis data PostgreSQL ke Parquet

Pemetaan jenis data MySQL dan MariaDB ke Parquet

Tabel berikut menunjukkan pemetaan dari jenis data MySQL dan MariaDB ke jenis data Parquet saat data dikonversi dan diekspor ke Amazon S3.

Jenis data sumber	Jenis primitif Parquet	Anotasi jenis logis	Catatan konversi
Jenis data numerik			
BIGINT	INT64		
BIGINT UNSIGNED	FIXED_LEN _BYTE_ARRAY(9)	DECIMAL(20,0)	Parquet hanya mendukung jenis yang ditandatangani, sehingga pemetaann ya memerlukan tambahan byte (8 plus 1) untuk menyimpan jenis BIGINT_UN SIGNED.
BIT	BYTE_ARRAY		
DECIMAL	INT32	DECIMAL(p,s)	Jika nilai sumber kurang dari 2^{31}, maka nilai tersebut akan disimpan sebagai INT32.
	INT64	DECIMAL(p,s)	Jika nilai sumber adalah 2^{31} atau lebih besar, tetapi kurang dari 2^{63}, maka nilai tersebut akan disimpan sebagai INT64.
	FIXED_LEN _BYTE_ARRAY(N)	DECIMAL(p,s)	Jika nilai sumber adalah 2^{63} atau lebih besar, maka nilai tersebut akan disimpan

Jenis data sumber	Jenis primitif Parquet	Anotasi jenis logis	Catatan konversi
			sebagai FIXED_LEN _BYTE_ARRAY(N).
	BYTE_ARRAY	STRING	Parquet tidak mendukung presisi Desimal yang lebih besar dari 38. Nilai Desimal akan dikonversi menjadi string dalam jenis BYTE_ARRAY dan dienkode sebagai UTF8.
DOUBLE	DOUBLE		
FLOAT	DOUBLE		
INT	INT32		
INT UNSIGNED	INT64		
MEDIUMINT	INT32		
MEDIUMINT UNSIGNED	INT64		
NUMERIC	INT32	DECIMAL(p,s)	Jika nilai sumber kurang dari 2^{31}, maka nilai tersebut akan disimpan sebagai INT32.

Jenis data sumber	Jenis primitif Parquet	Anotasi jenis logis	Catatan konversi
	INT64	DECIMAL(p,s)	Jika nilai sumber adalah 2^{31} atau lebih besar, tetapi kurang dari 2^{63}, maka nilai tersebut akan disimpan sebagai INT64.
	FIXED_LEN _ARRAY(N)	DECIMAL(p,s)	Jika nilai sumber adalah 2^{63} atau lebih besar, maka nilai tersebut akan disimpan sebagai FIXED_LEN _BYTE_ARRAY(N).
	BYTE_ARRAY	STRING	Parquet tidak mendukung presisi Numerik yang lebih besar dari 38. Nilai Numerik ini akan dikonversi menjadi string dalam jenis BYTE_ARRAY dan dienkode sebagai UTF8.
SMALLINT	INT32		
SMALLINT UNSIGNED	INT32		
TINYINT	INT32		
TINYINT UNSIGNED	INT32		

Jenis data sumber	Jenis primitif Parquet	Anotasi jenis logis	Catatan konversi
Jenis data string			
BINARY	BYTE_ARRAY		
BLOB	BYTE_ARRAY		
CHAR	BYTE_ARRAY		
ENUM	BYTE_ARRAY	STRING	
LINESTRING	BYTE_ARRAY		
LONGBLOB	BYTE_ARRAY		
LONGTEXT	BYTE_ARRAY	STRING	
MEDIUMBLOB	BYTE_ARRAY		
MEDIUMTEXT	BYTE_ARRAY	STRING	
MULTILINESTRING	BYTE_ARRAY		
SET	BYTE_ARRAY	STRING	
TEXT	BYTE_ARRAY	STRING	
TINYBLOB	BYTE_ARRAY		
TINYTEXT	BYTE_ARRAY	STRING	
VARBINARY	BYTE_ARRAY		
VARCHAR	BYTE_ARRAY	STRING	
Jenis data tanggal dan waktu			

Jenis data sumber	Jenis primitif Parquet	Anotasi jenis logis	Catatan konversi
DATE	BYTE_ARRAY	STRING	Tanggal akan dikonversi menjadi string dalam jenis BYTE_ARRAY dan dienkode sebagai UTF8.
DATETIME	INT64	TIMESTAMP_MICROS	
TIME	BYTE_ARRAY	STRING	Jenis TIME akan dikonversi menjadi string dalam jenis BYTE_ARRAY dan dienkode sebagai UTF8.
TIMESTAMP	INT64	TIMESTAMP_MICROS	
YEAR	INT32		
Jenis data geometris			
GEOMETRY	BYTE_ARRAY		
GEOMETRYCOLLECTION	BYTE_ARRAY		
MULTIPOINT	BYTE_ARRAY		
MULTIPOLYGON	BYTE_ARRAY		
POINT	BYTE_ARRAY		
POLYGON	BYTE_ARRAY		
Jenis data JSON			

Jenis data sumber	Jenis primitif Parquet	Anotasi jenis logis	Catatan konversi
JSON	BYTE_ARRAY	STRING	

Pemetaan jenis data PostgreSQL ke Parquet

Tabel berikut menunjukkan pemetaan dari dan jenis data PostgreSQL ke jenis data Parquet saat data dikonversi dan diekspor ke Amazon S3.

Jenis data PostgreSQL	Jenis primitif Parquet	Anotasi jenis logis	Catatan pemetaan
Jenis data numerik			
BIGINT	INT64		
BIGSERIAL	INT64		
DECIMAL	BYTE_ARRAY	STRING	Jenis DECIMAL akan dikonversi ke string dalam jenis BYTE_ARRAY dan dienkode sebagai UTF8. Konversi ini dimaksudkan untuk menghindari kerumitan akibat presisi data dan nilai data yang bukan berupa angka (NaN).
DOUBLE PRECISION	DOUBLE		
INTEGER	INT32		
MONEY	BYTE_ARRAY	STRING	

Jenis data PostgreSQL	Jenis primitif Parquet	Anotasi jenis logis	Catatan pemetaan
REAL	FLOAT		
SERIAL	INT32		
SMALLINT	INT32	INT_16	
SMALLSERIAL	INT32	INT_16	
Jenis data string dan terkait			
ARRAY	BYTE_ARRAY	STRING	Array akan dikonversi menjadi string dan dienkode sebagai BINARY (UTF8). Konversi ini dimaksudkan untuk menghindari kerumitan akibat presisi data, nilai data yang bukan berupa angka (NaN), dan nilai data waktu.
BIT	BYTE_ARRAY	STRING	
BIT VARYING	BYTE_ARRAY	STRING	
BYTEA	BINARY		
CHAR	BYTE_ARRAY	STRING	
CHAR(N)	BYTE_ARRAY	STRING	
ENUM	BYTE_ARRAY	STRING	

Jenis data PostgreSQL	Jenis primitif Parquet	Anotasi jenis logis	Catatan pemetaan
NAME	BYTE_ARRAY	STRING	
TEXT	BYTE_ARRAY	STRING	
TEXT SEARCH	BYTE_ARRAY	STRING	
VARCHAR(N)	BYTE_ARRAY	STRING	
XML	BYTE_ARRAY	STRING	
Jenis data tanggal dan waktu			
DATE	BYTE_ARRAY	STRING	
INTERVAL	BYTE_ARRAY	STRING	
TIME	BYTE_ARRAY	STRING	
TIME WITH TIME ZONE	BYTE_ARRAY	STRING	
TIMESTAMP	BYTE_ARRAY	STRING	
TIMESTAMP WITH TIME ZONE	BYTE_ARRAY	STRING	
Jenis data geometris			
BOX	BYTE_ARRAY	STRING	
CIRCLE	BYTE_ARRAY	STRING	
LINE	BYTE_ARRAY	STRING	
LINESEGMENT	BYTE_ARRAY	STRING	
PATH	BYTE_ARRAY	STRING	
POINT	BYTE_ARRAY	STRING	

Jenis data PostgreSQL	Jenis primitif Parquet	Anotasi jenis logis	Catatan pemetaan
POLYGON	BYTE_ARRAY	STRING	
Jenis data JSON			
JSON	BYTE_ARRAY	STRING	
JSONB	BYTE_ARRAY	STRING	
Jenis data lainnya			
BOOLEAN	BOOLEAN		
CIDR	BYTE_ARRAY	STRING	Jenis data jaringan
COMPOSITE	BYTE_ARRAY	STRING	
DOMAIN	BYTE_ARRAY	STRING	
INET	BYTE_ARRAY	STRING	Jenis data jaringan
MACADDR	BYTE_ARRAY	STRING	
OBJECT IDENTIFIER	N/A		
PG_LSN	BYTE_ARRAY	STRING	
RANGE	BYTE_ARRAY	STRING	
UUID	BYTE_ARRAY	STRING	

Menggunakan AWS Backup untuk mengelola backup otomatis

AWS Backup adalah layanan pencadangan yang dikelola sepenuhnya yang memudahkan untuk memusatkan dan mengotomatiskan cadangan data di seluruh AWS layanan di cloud dan di tempat. Anda dapat mengelola cadangan basis data Amazon RDS Anda di AWS Backup.

> **Note**
>
> Pencadangan yang dikelola oleh AWS Backup dianggap snapshot DB manual, tetapi tidak dihitung dalam kuota snapshot DB untuk RDS. Cadangan yang dibuat dengan AWS Backup memiliki nama yang diakhiri dengan. `awsbackup:`*`backup-job-number`*

Untuk informasi selengkapnya AWS Backup, lihat Panduan AWS Backup Pengembang.

Untuk melihat cadangan yang dikelola oleh AWS Backup

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Snapshot.
3. Pilih tab Layanan pencadangan.

   AWS Backup Cadangan Anda tercantum di bawah snapshot layanan Backup.

Memantau metrik dalam instans Amazon RDS

Di bagian berikut, Anda dapat menemukan gambaran umum tentang pemantauan Amazon RDS dan penjelasan tentang cara mengakses metrik. Untuk mempelajari cara memantau peristiwa, log, dan aliran aktivitas basis data, lihat Memantau peristiwa, log, dan aliran di instans DB Amazon RDS .

Topik

Ikhtisar metrik pemantauan di Amazon RDS

Pemantauan adalah bagian penting dari upaya memelihara keandalan, ketersediaan, dan kinerja Amazon RDS dan solusi AWS Anda. Agar dapat menelusuri dengan mudah kegagalan multipoin, kami menganjurkan supaya Anda mengumpulkan data pemantauan dari semua bagian solusi AWS Anda.

Topik

Rencana pemantauan

Sebelum Anda memulai pemantauan Amazon RDS, buat rencana pemantauan. Rencana ini sepatutnya menjawab pertanyaan-pertanyaan berikut:

- Apa sajakah sasaran pemantauan Anda?
- Sumber daya manakah yang akan Anda pantau?
- Seberapa seringkah Anda akan memantau sumber daya ini?
- Apa sajakah alat pemantauan yang akan Anda gunakan?
- Siapakah yang akan melakukan tugas pemantauan?
- Siapakah yang harus diberi tahu apabila ada berjalan salah?

Garis dasar kinerja

Untuk mencapai semua sasaran pemantauan, Anda perlu menetapkan garis dasar. Untuk itu, ukur kinerja pada berbagai kondisi beban dan waktu di lingkungan Amazon RDS Anda. Anda dapat memantau metrik-metrik seperti:

- Throughput jaringan
- Koneksi klien
- I/O untuk operasi baca, tulis, atau metadata
- Keseimbangan kredit lonjakan untuk instans basis data Anda

Kami menyarankan agar Anda menyimpan data riwayat kinerja untuk Amazon RDS. Dengan menggunakan data yang disimpan, Anda dapat membandingkan kinerja saat ini dengan tren masa lalu. Anda juga dapat membedakan pola kinerja normal dari anomali, dan merancang teknik untuk mengatasi masalah.

Pedoman kinerja

Secara umum, nilai-nilai yang diperkenankan untuk metrik kinerja bergantung pada kinerja aplikasi Anda secara relatif terhadap garis dasar Anda. Selidiki variansi yang konsisten atau sedang tren dari garis dasar Anda. Metrik-metrik berikut sering menjadi sumber masalah kinerja:

- Konsumsi CPU atau RAM tinggi – Nilai-nilai tinggi untuk konsumsi CPU atau RAM mungkin layak, jika keduanya mengikuti sasaran untuk aplikasi Anda (seperti throughput atau konkurensi) dan diharapkan.
- Konsumsi ruang disk – Selidiki konsumsi ruang disk jika ruang yang digunakan selalu berada pada atau di atas 85 persen dari total ruang disk. Lihat apakah menghapus data dari instans atau mengarsipkan data ke sistem yang lain layak dilakukan guna melegakan ruang.
- Lalu lintas jaringan – Untuk lalu lintas jaringan, bicaralah dengan administrator sistem Anda untuk memahami throughput yang diharapkan bagi jaringan domain dan koneksi internet Anda. Selidiki lalu lintas jaringan jika throughput selalu di bawah yang diharapkan.
- Koneksi basis data – Jika Anda melihat jumlah koneksi pengguna yang tinggi dan juga penurunan kinerja dan waktu respons instans, pertimbangkan untuk membatasi koneksi basis data. Jumlah koneksi pengguna terbaik untuk instans basis data bervariasi berdasarkan kelas instans dan kerumitan operasi yang dilakukan. Untuk menentukan jumlah koneksi basis data, kaitkan instans basis data Anda dengan grup parameter dengan parameter `User Connections` diatur ke nilai selain 0 (tidak terbatas). Anda dapat menggunakan grup parameter yang ada atau membuat grup baru. Untuk informasi selengkapnya, lihat Bekerja dengan grup parameter.
- Metrik IOPS – Nilai yang diharapkan untuk metrik IOPS bergantung pada spesifikasi disk dan konfigurasi server; jadi, gunakan garis dasar Anda untuk mengetahui nilai yang lazim. Selidiki apakah nilai-nilai selalu berbeda dengan garis dasar Anda. Untuk kinerja IOPS terbaik, pastikan bahwa set kerja Anda yang biasa sesuai dengan memori untuk meminimalkan operasi baca dan tulis.

Ketika kinerja berada di luar garis dasar yang telah ditetapkan, Anda mungkin perlu membuat perubahan untuk mengoptimalkan ketersediaan basis data bagi beban kerja Anda. Misalnya, Anda

mungkin perlu mengubah kelas instans dari instans basis data Anda. Atau Anda mungkin perlu mengubah jumlah instans basis data dan replika baca yang tersedia untuk klien.

Alat-alat pemantauan

Pemantauan adalah bagian penting dari upaya memelihara keandalan, ketersediaan, dan kinerja Amazon RDS dan solusi Anda AWS yang lain. AWS menyediakan berbagai alat pemantauan untuk mengawasi Amazon RDS, melaporkan saat ada yang tidak beres, dan mengambil tindakan otomatis jika layak.

Topik

Alat-alat pemantauan otomatis

Kami menyarankan agar Anda mengotomatiskan tugas-tugas pemantauan sebanyak mungkin.

Topik

Status dan rekomendasi instans Amazon RDS

Anda dapat menggunakan alat-alat otomatis berikut untuk memantau Amazon RDS dan melaporkan saat ada yang salah:

- Status instans Amazon RDS — Lihat detail status klaster Anda saat ini dengan menggunakan konsol Amazon RDS, AWS CLI, atau API RDS.
- Rekomendasi Amazon RDS — Tanggapi rekomendasi otomatis untuk sumber daya basis data, seperti instans basis data, , replika baca, dan . Untuk informasi selengkapnya, lihat Melihat dan menanggapi rekomendasi Amazon Aurora RDS.

CloudWatch

Amazon RDS Aurora terintegrasi dengan CloudWatch Amazon untuk kemampuan pemantauan tambahan.

- Amazon CloudWatch — Layanan ini memantau AWS sumber daya Anda dan aplikasi yang Anda jalankan AWS secara real time. Anda dapat menggunakan CloudWatch fitur Amazon berikut dengan Amazon RDS :
  - CloudWatch Metrik Amazon — Amazon RDS Amazon secara otomatis mengirimkan metrik ke setiap menit CloudWatch untuk setiap basis data aktif. Anda tidak mendapatkan biaya tambahan untuk metrik Amazon RDS di. CloudWatch Lihat informasi yang lebih lengkap di Memantau metrik Amazon RDS dengan Amazon CloudWatch.
  - CloudWatch Alarm Amazon - Anda dapat menonton satu metrik Amazon RDS Aurora selama periode waktu tertentu. Anda lalu dapat melakukan satu atau beberapa tindakan berdasarkan nilai metrik itu relatif terhadap ambang batas yang Anda tetapkan. Lihat informasi yang lebih lengkap di Memantau metrik Amazon RDS dengan Amazon CloudWatch.

Wawasan Performa Amazon RDS dan pemantauan sistem operasi

Anda dapat menggunakan alat-alat otomatis berikut untuk memantau kinerja Amazon RDS:

- Wawasan Performa Amazon RDS – Telaah beban pada basis data Anda, dan tentukan kapan dan di mana harus mengambil tindakan. Untuk informasi selengkapnya, lihat Memantau muatan DB dengan Wawasan Performa di Amazon RDS.
- Pemantauan Disempurnakan Amazon RDS – Lihat secara waktu nyata metrik-metrik untuk sistem operasi. Untuk informasi selengkapnya, lihat Memantau metrik OS dengan Pemantauan yang Disempurnakan.

Layanan terintegrasi

Layanan AWS berikut terintegrasi dengan Amazon RDS:

- Amazon EventBridge adalah layanan bus acara tanpa server yang memudahkan untuk menghubungkan aplikasi Anda dengan data dari berbagai sumber. Untuk informasi selengkapnya, lihat Memantau peristiwa Amazon RDS.

- Amazon CloudWatch Logs memungkinkan Anda memantau, menyimpan, dan mengakses file log Anda dari instans Amazon Aurora Amazon RDS Aurora CloudTrail, dan sumber lainnya. Untuk informasi selengkapnya, lihat Memantau file log Amazon RDS.
- AWS CloudTrail menangkap panggilan API dan peristiwa terkait yang dibuat oleh atau atas nama Akun AWS Anda dan mengirimkan berkas log ke bucket Amazon S3 yang Anda tentukan. Untuk informasi selengkapnya, lihat Memantau panggilan API Amazon RDS di AWS CloudTrail.
- Aliran Aktivitas Database Untuk informasi selengkapnya, lihat Memantau Amazon RDS dengan Aliran Aktivitas Basis Data.

Alat-alat pemantauan manual

Anda perlu memonitor secara manual item yang tidak CloudWatch tercakup oleh alarm. Amazon RDS, CloudWatch, AWS Trusted Advisor dan dasbor AWS konsol lainnya memberikan at-a-glance tampilan keadaan lingkungan AndaAWS. Kami menyarankan agar Anda juga memeriksa file log pada instans basis data Anda.

- Dari konsol Amazon RDS, Anda dapat memantau butir-butir berikut untuk sumber daya Anda:
  - Jumlah koneksi dengan instans basis data
  - Jumlah operasi baca dan tulis ke instans basis data
  - Jumlah penyimpanan yang saat ini digunakan oleh instans basis data
  - Jumlah memori dan CPU yang sedang digunakan untuk instans basis data
  - Jumlah lalu lintas jaringan ke dan dari instans basis data
- Dari dasbor Trusted Advisor, Anda dapat meninjau pemeriksan-pemeriksaan optimasi biaya, keamanan, toleransi kesalahan, dan peningkatan kinerja berikut:
  - Amazon RDS Idle DB Instances
  - Amazon RDS Security Group Access Risk
  - Amazon RDS Backups
  - Amazon RDS Multi-AZ

  Lihat informasi yang lebih lengkap tentang pemeriksaan-pemeriksaan ini di Praktik terbaik (pemeriksaan) Trusted Advisor.
- CloudWatch halaman rumah menunjukkan:
  - Alarm dan status saat ini
  - Grafik alarm dan sumber daya

- Status kesehatan layanan

Selain itu, Anda dapat menggunakan CloudWatch untuk melakukan hal berikut:

- Membuat dasbor yang disesuaikan untuk memantau layanan yang Anda pedulikan.
- Data metrik grafik untuk memecahkan masalah dan mengungkap tren.
- Mencari dan menelusuri semua metrik sumber daya AWS Anda.
- Membuat dan mengedit alarm agar diberi tahu tentang masalah.

Melihat status instance

Menggunakan konsol Amazon RDS, Anda dapat dengan cepat mengakses status instans DB Anda.

Topik

- Melihat status instans DB Amazon RDS

Melihat status instans DB Amazon RDS

Status instans DB menunjukkan kondisi instans DB. Anda dapat menggunakan prosedur berikut untuk melihat status instans DB di konsol Amazon RDS, AWS CLI perintah, atau operasi API.

> **Note**
>
> Amazon RDS juga menggunakan status lain yang disebut status pemeliharaan, yang ditunjukkan di kolom Pemeliharaan pada konsol Amazon RDS. Nilai ini menunjukkan status patch pemeliharaan yang perlu diterapkan ke instans DB. Status pemeliharaan bergantung pada status instans DB. Untuk informasi lebih lanjut tentang status pemeliharaan, lihat Menerapkan pembaruan untuk instans DB.

Temukan kemungkinan nilai status untuk instans DB dalam tabel berikut. Tabel ini juga menunjukkan apakah Anda akan ditagih untuk instans DB dan penyimpanan, ditagih hanya untuk penyimpanan, atau Anda tidak dikenai tagihan. Untuk semua status instans DB, Anda selalu ditagih untuk penggunaan cadangan.

Status instans DB	Ditagih	Deskripsi
Available	Ditagih	Instans DB berkondisi baik dan tersedia.
Backing-up	Ditagih	Instans DB saat ini sedang dicadangkan.
C onfiguring-enhanced-monitoring	Ditagih	Pemantauan yang ditingkatkan diaktifkan atau dinonaktifkan untuk instans DB ini.
C onfiguring-iam-database-auth	Ditagih	AWS Identity and Access Management (IAM) otentikasi database sedang diaktifkan atau dinonaktifkan untuk instans DB ini.
C onfiguring-log-exports	Ditagih	Menerbitkan file log ke Amazon CloudWatch Logs sedang diaktifkan atau dinonaktifkan untuk instans DB ini.
C onverting-to-vpc	Ditagih	Instans DB sedang dikonversi dari instans DB yang tidak ada di Amazon Virtual Private Cloud (Amazon VPC) menjadi instans DB yang ada di Amazon VPC.

Status instans DB	Ditagih	Deskripsi
Creating	Tidak ditagih	Instans DB sedang dibuat. Instans DB tidak dapat diakses saat sedang dibuat.
Delete-precheck	Tidak ditagih	Amazon RDS memvalidasi bahwa replika baca berkondisi baik dan aman untuk dihapus.
Deleting	Tidak ditagih	Instans DB sedang dihapus.
Failed	Tidak ditagih	Instans DB gagal dan Amazon RDS tidak dapat dipulihkan. Lakukan point-in-time pemulihan ke waktu restorable terbaru dari instans DB untuk memulihkan data.
Aku naccessible-encryption-credentials	Tidak ditagih	Yang AWS KMS key digunakan untuk mengenkripsi atau mendekripsi instans DB tidak dapat diakses atau dipulihkan.
Aku naccessible-encryption-credentials-recoverable	Ditagih untuk penyimpan	Kunci KMS digunakan untuk mengenkripsi atau mendekripsi instans DB tidak dapat diakses. Namun, jika kunci KMS aktif, Anda dapat memulai ulang instans DB untuk memulihkannya. Untuk informasi selengkapnya, lihat Mengenkripsi instans DB.
Incompatible-network	Tidak ditagih	Amazon RDS mencoba melakukan tindakan pemulihan di instans DB tetapi tidak dapat melakukannya karena VPC berada dalam keadaan yang mencegah penyelesaian tindakan. Status ini dapat terjadi jika, misalnya, semua alamat IP yang tersedia di subnet sedang digunakan dan Amazon RDS tidak bisa mendapatkan alamat IP untuk instans DB.
Aku ncompatible-option-group	Ditagih	Amazon RDS mencoba menerapkan perubahan grup opsi tetapi tidak dapat melakukannya, dan Amazon RDS tidak dapat melakukan roll back ke status grup opsi sebelumnya. Untuk informasi selengkapnya, lihat daftar Peristiwa Terbaru untuk instans DB. Status ini dapat terjadi jika, misalnya, grup opsi berisi opsi seperti TDE dan instans DB tidak berisi informasi terenkripsi.

Status instans DB	Ditagih	Deskripsi
Incompatible-parameters	Ditagih	Amazon RDS tidak dapat memulai instans DB karena parameter yang ditentukan dalam grup parameter DB instans DB tidak kompatibel dengan instans DB. Kembalikan perubahan parameter atau buat parameter kompatibel dengan instans DB untuk mendapatkan akses ke instans DB Anda. Untuk informasi selengkapnya tentang parameter yang tidak kompatibel, periksa daftar Peristiwa Terbaru untuk instans DB.
Incompatible-restore	Tidak ditagih	Amazon RDS tidak dapat melakukan point-in-time pemulihan. Penyebab umum untuk status ini termasuk penggunaan tabel sementara , penggunaan tabel MyISAM dengan MySQL, atau penggunaan tabel Aria dengan MariaDB.
Insufficient-capacity	Tidak ditagih	Amazon RDS tidak dapat membuat instans karena kapasitas yang tersedia saat ini tidak cukup. Untuk membuat instans DB di AZ yang sama dengan jenis instans yang sama, hapus instans DB Anda, tunggu beberapa jam, lalu coba untuk membuat lagi. Atau, buat instans baru menggunakan kelas instans atau AZ yang berbeda.
Maintenance	Ditagih	Amazon RDS menerapkan pembaruan pemeliharaan pada instans DB. Status ini digunakan untuk pemeliharaan tingkat instans yang dijadwalkan RDS sejak jauh hari sebelumnya.
Modifying	Ditagih	Instans DB sedang dimodifikasi karena ada permintaan pelanggan untuk memodifikasi instans DB.
M oving-to-vpc	Ditagih	Instans DB sedang dipindahkan ke Amazon Virtual Private Cloud (Amazon VPC) baru.
Rebooting	Ditagih	Instans DB sedang di-boot ulang karena permintaan pelanggan atau proses Amazon RDS yang memerlukan boot ulang instans DB.
R esetting-master-credentials	Ditagih	Kredensial master untuk instans DB sedang direset karena ada permintaan pelanggan untuk meresetnya.

Status instans DB	Ditagih	Deskripsi
Renaming	Ditagih	Nama instans DB sedang diganti karena ada permintaan pelanggan untuk mengganti namanya.
Restore-error	Ditagih	Instans DB mengalami kesalahan saat mencoba mengembalikan ke point-in-time atau dari snapshot.
Starting	Ditagih untuk penyimpan an	Instans DB dimulai.
Stopped	Ditagih untuk penyimpan an	Instans DB dihentikan.
Stopping	Ditagih untuk penyimpan an	Instans DB sedang dihentikan.
S torage-config-upgrade	Ditagih	Konfigurasi sistem file penyimpanan instans DB sedang ditingkatkan. Status ini hanya berlaku untuk basis data green dalam deployment blue/green, atau untuk replika baca instans DB.
Storage-full	Ditagih	Instans DB telah mencapai alokasi kapasitas penyimpanannya. Ini merupakan status kritis. Sebaiknya segera perbaiki masalah ini. Untuk melakukannya, perbesar penyimpanan Anda dengan memodifikasi instans DB. Untuk menghindari situasi ini, setel CloudWatch alarm Amazon untuk memperingatkan Anda saat ruang penyimpanan semakin rendah.

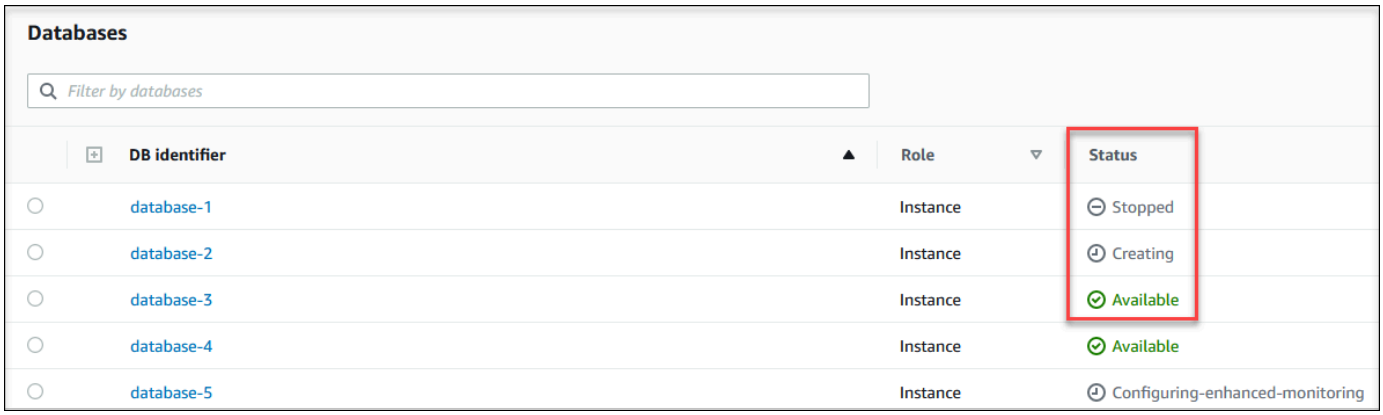
Status instans DB	Ditagih	Deskripsi
Storage-optimization	Ditagih	Amazon RDS mengoptimalkan penyimpanan instans DB Anda. Instans DB berfungsi sepenuhnya. Proses pengoptim alan penyimpanan biasanya singkat, tetapi terkadang dapat memakan waktu hingga lebih dari 24 jam.
Meningkatkan	Ditagih	Versi mesin basis data sedang ditingkatkan.

Konsol

Untuk melihat status instans DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data.

   Halaman Basis Data muncul dengan daftar instans DB. Nilai status setiap instans DB akan ditampilkan.



CLI

Untuk melihat instans DB dan informasi statusnya dengan menggunakan AWS CLI, gunakan describe-db-instancesperintah. Misalnya, AWS CLI perintah berikut mencantumkan semua informasi instance DB.

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

Untuk melihat instans DB tertentu dan statusnya, panggil describe-db-instancesperintah dengan opsi berikut:

- `DBInstanceIdentifier` – Nama instans DB.

```
aws rds describe-db-instances --db-instance-identifier mydbinstance
```

Untuk melihat status semua instans DB, gunakan kueri berikut di AWS CLI.

```
aws rds describe-db-instances --query 'DBInstances[*].
[DBInstanceIdentifier,DBInstanceStatus]' --output table
```

API

Untuk melihat status instans DB menggunakan API Amazon RDS, panggil operasi DescribeDBInstances.

Melihat dan menanggapi rekomendasi Amazon Aurora RDS

Amazon RDS Aurora memberikan rekomendasi otomatis untuk sumber daya database, seperti instans DB, replika baca, dan grup parameter DB. Rekomendasi ini memberikan panduan praktik terbaik dengan menganalisis konfigurasi instans DB, penggunaan, dan data performa.

Amazon RDS Performance Insights memantau metrik tertentu dan secara otomatis membuat ambang batas dengan menganalisis level apa yang dianggap berpotensi bermasalah untuk sumber daya tertentu. Ketika nilai metrik baru melewati ambang batas yang telah ditentukan selama periode waktu tertentu, Performance Insights menghasilkan rekomendasi proaktif. Rekomendasi ini membantu mencegah dampak kinerja database future. Misalnya, rekomendasi “Idle In Transaction” dihasilkan untuk RDS untuk PostgreSQL Aurora PostgreSQL tidak melakukan pekerjaan aktif, tetapi dapat membuat sumber daya database diblokir. Untuk menerima rekomendasi proaktif, Anda harus mengaktifkan Performance Insights dengan periode retensi tingkat berbayar. Untuk informasi tentang mengaktifkan Performance Insights, lihat. Mengaktifkan dan menonaktifkan Wawasan Performa Untuk informasi tentang harga dan retensi data untuk Performance Insights, lihat. Harga dan retensi data untuk Wawasan Performa

DevOpsGuru untuk RDS memantau metrik tertentu untuk mendeteksi kapan perilaku metrik menjadi sangat tidak biasa atau anomali. Anomali ini dilaporkan sebagai wawasan reaktif dengan rekomendasi. Misalnya, DevOps Guru untuk RDS mungkin menyarankan Anda untuk mempertimbangkan peningkatan kapasitas CPU atau menyelidiki peristiwa tunggu yang berkontribusi pada pemuatan DB. DevOpsGuru untuk RDS juga memberikan rekomendasi proaktif berbasis ambang batas. Untuk rekomendasi ini, Anda harus mengaktifkan DevOps Guru untuk RDS. Untuk informasi tentang mengaktifkan DevOps Guru untuk RDS, lihatMengaktifkan DevOps Guru dan menentukan cakupan sumber daya.

Rekomendasi akan berada dalam salah satu status berikut: aktif, diberhentikan, tertunda, atau diselesaikan. Rekomendasi yang diselesaikan tersedia selama 365 hari.

Anda dapat melihat atau mengabaikan rekomendasi. Anda dapat segera menerapkan rekomendasi aktif berbasis konfigurasi, menjadwalkannya di jendela pemeliharaan berikutnya, atau mengabaikannya. Untuk rekomendasi reaktif berbasis proaktif dan pembelajaran mesin berbasis ambang batas, Anda perlu meninjau penyebab masalah yang disarankan dan kemudian melakukan tindakan yang disarankan untuk memperbaiki masalah.

Topik

- Melihat rekomendasi Amazon RDS

- Menanggapi rekomendasi Amazon RDS

Melihat rekomendasi Amazon RDS

Amazon RDS memberikan rekomendasi untuk sumber daya ketika sumber daya tersebut dibuat atau dimodifikasi.

Rekomendasi berbasis konfigurasi didukung di wilayah berikut:

- AS Timur (Ohio)
- AS Timur (Virginia Utara)
- AS Barat (California Utara)
- AS Barat (Oregon)
- Asia Pasifik (Mumbai)
- Asia Pasifik (Seoul)
- Asia Pasifik (Singapura)
- Asia Pasifik (Sydney)
- Asia Pasifik (Tokyo)
- Kanada (Pusat)
- Eropa (Frankfurt)
- Eropa (Irlandia)
- Eropa (London)
- Eropa (Paris)
- Amerika Selatan (São Paulo)

Anda dapat menemukan contoh rekomendasi berbasis konfigurasi dalam tabel berikut.

Tipe	Deskripsi	Rekomendasi	Diperlu n downtir	Informasi tambahan
Volume magnetik sedang digunakan	Instans DB Anda menggunakan penyimpanan magnetik. Penyimpan an magnetik tidak	Pilih jenis penyimpan an yang berbeda: General Purpose (SSD) atau Provision ed IOPS.	Ya	Volume generasi sebelumnya dalam dokumentasi Amazon EC2.

Tipe	Deskripsi	Rekomendasi	Diperlu n downtir	Informasi tambahan
	disarankan untuk sebagian besar instans DB. Pilih jenis penyimpanan yang berbeda: General Purpose (SSD) atau Provisioned IOPS.			
Sumber Daya Pencadang an otomatis dimatikan	Pencadangan otomatis tidak diaktifkan untuk instans DB Anda. Pencadangan otomatis direkomen dasikan karena memungkinkan point-in-time pemulihan instans DB Anda.	Aktifkan pencadang an otomatis dengan periode retensi hingga 14 hari.	Ya	Mengaktifkan pencadangan otomatis Mengungkap biaya penyimpan an cadangan Amazon RDS di Blog Database AWS
Diperlukan upgrade versi minor engine	Sumber daya database Anda tidak menjalank an versi mesin DB minor terbaru. Versi minor terbaru berisi perbaikan keamanan terbaru dan peningkat an lainnya.	Tingkatkan ke versi mesin terbaru.	Ya	Meng-upgrade versi mesin instans DB

Tipe	Deskripsi	Rekomendasi	Diperlu n downtir	Informasi tambahan
Peningkatan Monitoring dimatikan	Sumber daya database Anda tidak mengaktif kan Enhanced Monitoring. Peningkat an Pemantauan menyediakan metrik sistem operasi waktu nyata untuk pemantauan dan pemecahan masalah.	Aktifkan Pemantauan yang Ditingkatkan.	Tidak	Memantau metrik OS dengan Pemantauan yang Disempurnakan

Tipe	Deskripsi	Rekomendasi	Diperlu n downtir	Informasi tambahan
Enkripsi penyimpanan dimatikan	Amazon RDS mendukung enkripsi saat istirahat untuk semua mesin database dengan menggunakan kunci yang Anda kelola di AWS Key Management Service (AWSKMS). Pada instans DB aktif dengan enkripsi Amazon RDS, data yang disimpan saat istirahat di penyimpan an dienkripsi, mirip dengan pencadang an otomatis, replika baca, dan snapshot. Jika enkripsi tidak diaktifkan saat membuat instans DB, Anda harus membuat dan mengembalikan salinan terenkripsi dari snapshot yang didekripsi dari instans DB sebelum Anda mengaktifkan enkripsi.	Aktifkan enkripsi data saat istirahat untuk instans DB Anda.	Ya	Keamanan dalam Amazon RDS Menyalin snapshot DB

Tipe	Deskripsi	Rekomendasi	Diperlu n downtir	Informasi tambahan
Performan ce Insights dimatikan	Performance Insights memantau pemuatan instans DB untuk membantu Anda menganalisis dan menyelesaikan masalah kinerja database. Sebaiknya aktifkan Performance Insights.	Mengaktifkan Wawasan Performa.	Tidak	Memantau muatan DB dengan Wawasan Performa di Amazon RDS
Instans DB memiliki penyimpanan autoscaling dimatikan	Penskalaan otomatis penyimpanan tidak diaktifkan untuk instans DB Anda. Ketika beban kerja database meningkat, penskalaan otomatis penyimpanan RDS secara otomatis menskalakan kapasitas penyimpan an dengan nol waktu henti.	Aktifkan penskalaan otomatis penyimpan an Amazon RDS dengan ambang penyimpanan maksimum yang ditentukan	Tidak	Mengelola kapasitas secara otomatis dengan penskalaan otomatis penyimpan an Amazon RDS

Tipe	Deskripsi	Rekomendasi	Diperlu n downtir	Informasi tambahan
Sumber daya RDS pembaruan versi utama diperlukan	Database dengan versi utama saat ini untuk mesin DB tidak akan didukung. Kami menyarankan Anda meningkatkan ke versi utama terbaru yang mencakup fungsiona litas dan peningkatan baru.	Tingkatkan ke versi utama terbaru untuk mesin DB.	Ya	Meng-upgrade versi mesin instans DB Menggunakan Deployment Blue/ Green Amazon RDS untuk pembaruan basis data
Pembaruan kelas instance sumber daya RDS diperlukan	Instans DB Anda menjalankan kelas instans DB generasi sebelumnya. Kami telah mengganti kelas instans DB dari generasi sebelumny a dengan kelas instans DB dengan biaya, kinerja, atau keduanya yang lebih baik. Kami menyarank an Anda menjalank an instans DB Anda dengan kelas instans DB dari generasi yang lebih baru.	Tingkatkan kelas instans DB.	Ya	Mesin DB yang didukung untuk kelas instans DB

Tipe	Deskripsi	Rekomendasi	Diperlu n downtir	Informasi tambahan
Sumber daya RDS menggunak an akhir edisi mesin pendukung di bawah lisensi yang disertakan	Kami menyarank an Anda meningkat kan versi utama ke versi mesin terbaru yang didukung oleh Amazon RDS untuk melanjutkan dukungan lisensi saat ini. Versi mesin database Anda tidak akan didukung dengan lisensi saat ini.	Kami menyarankan Anda meningkatkan database Anda ke versi terbaru yang didukung di Amazon RDS untuk terus menggunakan model berlisensi.	Ya	Upgrade versi mayor Oracle

Tipe	Deskripsi	Rekomendasi	Diperlu n downtir	Informasi tambahan
Instans DB tidak menggunak an penerapan Multi-AZ	Sebaiknya gunakan deployment Multi-AZ. Deployment Multi-AZ meningkatkan ketersediaan dan ketahanan instans DB.	Siapkan Multi-AZ untuk instans DB yang terkena dampak	Tidak Tidak terjadi waktu henti selama peruba ini. Namun ada kemun an dampa pada perform Lihat informa yang lebih lengka di Mengu instans DB menjad deploy t instans DB Multi-AZ	Harga untuk Amazon RDS Multi-AZ

Tipe	Deskripsi	Rekomendasi	Diperlu n downtir	Informasi tambahan
Parameter memori DB menyimpang dari default	Parameter memori instans DB berbeda secara signifikan dari nilai default. Pengaturan ini dapat memengaruhi kinerja dan menyebabkan kesalahan. Kami menyarankan Anda mengatur ulang parameter memori khusus untuk instans DB ke nilai defaultnya di grup parameter DB.	Setel ulang parameter memori ke nilai defaultnya.	Tidak	Praktik terbaik untuk mengonfigurasi parameter kinerja untuk Amazon RDS for MySQL di Blog Database AWS

Tipe	Deskripsi	Rekomendasi	Diperlu n downtir	Informasi tambahan
`InnoDB_Ch ange_Buff ering` parame menggunak an kurang dari nilai optimal	Perubahan buffering memungkinkan instance MySQL DB untuk menunda beberapa penulisan, yang diperlukan untuk mempertahankan indeks sekunder. Fitur ini berguna di lingkungan dengan disk lambat. Konfigura si buffering perubahan sedikit meningkat kan kinerja DB tetapi menyebabkan penundaan pemulihan kerusakan dan waktu shutdown yang lama selama peningkatan.	Tetapkan nilai `InnoDB_Ch ange_Buffering` parameter ke `NONE` dalam grup parameter DB Anda.	Tidak	Praktik terbaik untuk mengonfigurasi parameter kinerja untuk Amazon RDS for MySQL di Blog Database AWS

Tipe	Deskripsi	Rekomendasi	Diperlu n downtir	Informasi tambahan
Parameter cache kueri diaktifkan	Ketika perubahan mengharuskan cache kueri Anda dibersihk an, instans DB Anda akan tampak macet. Cache kueri tidak bermanfaat untuk sebagian besar beban kerja. Cache kueri dihapus dari MySQL versi 8.0. Kami menyarank an Anda mengatur parameter query_cac he_type ke 0.	Tetapkan nilai `query_cac he_type` parameter ke `0` dalam grup parameter DB Anda.	Ya	Praktik terbaik untuk mengonfigurasi parameter kinerja untuk Amazon RDS for MySQL di Blog Database AWS
`log_outpu t` parameter diatur ke tabel	Ketika `log_output` diatur ke`TABLE`, lebih banyak penyimpanan digunakan daripada ketika `log_outpu t` diatur ke`FILE`. Kami menyarank an Anda mengatur parameter ke`FILE`, untuk menghinda ri mencapai batas ukuran penyimpanan.	Tetapkan nilai `log_output` parameter ke `FILE` dalam grup parameter DB Anda.	Tidak	File log basis data MySQL

Tipe	Deskripsi	Rekomendasi	Diperlu n downtir	Informasi tambahan
Grup parameter tidak menggunak an halaman besar	Halaman besar dapat meningkat kan skalabilitas database, tetapi instans DB Anda tidak menggunak an halaman besar. Kami menyarankan Anda mengatur nilai `use_large_pages` parameter ke `ONLY` dalam grup parameter DB untuk instans DB Anda.	Tetapkan nilai `use_large_pages` parameter ke `ONLY` dalam grup parameter DB Anda.	Ya	Mengaktifkan HugePages untuk instans RDS for Oracle
`autovacuu m` parameter dimatikan	Parameter autovacuu m dimatikan untuk instans DB Anda. Mematikan autovacuu m meningkatkan tabel dan indeks kembung dan berdampak pada kinerja. Kami menyarankan Anda mengaktifkan autovacuum di grup parameter DB Anda.	Aktifkan parameter autovacuum di grup parameter DB Anda.	Tidak	Memahami autovacuu m di Amazon RDS untuk lingkungan PostgreSQL di Blog Database AWS

Tipe	Deskripsi	Rekomendasi	Diperlu n downtir	Informasi tambahan
`synchrono us_commit` parameter dimatikan	Ketika `synchrono us_commit` parameter dimatikan , data dapat hilang dalam kerusakan database. Daya tahan database berisiko. Sebaiknya aktifkan parameter `synchrono us_commit` .	Aktifkan `synchrono us_commit` parameter di grup parameter DB Anda.	Ya	Parameter Amazon Aurora PostgreSQL: Replikasi, keamanan, dan logging di Blog Database AWS
`track_cou nts` parameter dimatikan	Ketika `track_cou nts` parameter dimatikan, database tidak mengumpul kan statistik aktivitas database. Autovacuu m membutuhkan statistik ini untuk berfungsi dengan benar. Sebaiknya tetapkan parameter `track_counts` ke 1.	Setel `track_cou nts` parameter ke1.	Tidak	Statistik Run-time untuk PostgreSQL

Tipe	Deskripsi	Rekomendasi	Diperlu n downtir	Informasi tambahan
`enable_in dexonlysc an` parameter dimatikan	Perencana kueri atau pengoptimal tidak dapat menggunakan jenis paket pemindaia n khusus indeks saat dimatikan. Kami menyarank an Anda mengatur nilai `enable_in dexonlyscan` parameter ke`1`.	Tetapkan nilai `enable_in dexonlyscan` parameter ke`1`.	Tidak	Konfigurasi Metode Perencana untuk PostgreSQL
`enable_in dexscan` para dimatikan	Perencana kueri atau pengoptimal tidak dapat menggunak an jenis rencana pemindaian indeks saat dimatikan. Kami menyarankan Anda menetapka n `enable_in dexscan` nilainya`1`.	Tetapkan nilai `enable_in dexscan` parameter ke`1`.	Tidak	Konfigurasi Metode Perencana untuk PostgreSQL

Tipe	Deskripsi	Rekomendasi	Diperlu n downtir	Informasi tambahan
`innodb_fl ush_log_a t_trx` parame dimatikan	Nilai `innodb_fl ush_log_at_trx` parameter instans DB Anda bukanlah nilai aman. Parameter ini mengontrol persisten si operasi commit ke disk. Sebaiknya tetapkan parameter `innodb_fl ush_log_at_trx` ke 1.	Tetapkan nilai `innodb_fl ush_log_at_trx` parameter ke1.	Tidak	Praktik terbaik untuk mengonfigurasi parameter kinerja untuk Amazon RDS for MySQL di Blog Database AWS
`sync_binl og` parameter dimatikan	Sinkronisasi log biner ke disk tidak diberlaku kan sebelum komit transaksi diakui dalam instans DB Anda. Kami menyarank an Anda mengatur nilai `sync_binlog` parameter ke1.	Tetapkan nilai `sync_binlog` parameter ke1.	Tidak	Praktik terbaik untuk mengonfigurasi parameter replikasi untuk Amazon RDS for MySQL di Blog Database AWS

Tipe	Deskripsi	Rekomendasi	Diperlu n downtir	Informasi tambahan
`innodb_st ats_persi stent parame` dimatikan	Instans DB Anda tidak dikonfigurasi untuk mempertahankan statistik InnoDB ke disk. Ketika statistik tidak disimpan, mereka dihitung ulang setiap kali instance restart dan tabel diakses. Hal ini menyebabkan variasi dalam rencana eksekusi query. Anda dapat memodifikasi nilai parameter global ini di tingkat tabel. Kami menyarank an Anda mengatur nilai `innodb_st ats_persistent` parameter ke`0N`.	Tetapkan nilai `innodb_st ats_persistent` parameter ke`0N`.	Tidak	Praktik terbaik untuk mengonfigurasi parameter kinerja untuk Amazon RDS for MySQL di Blog Database AWS

Tipe	Deskripsi	Rekomendasi	Diperlu n downtir	Informasi tambahan
`innodb_op en_files` Par rendah	`innodb_op en_files` Parameter mengontrol jumlah file InnoDB dapat membuka pada satu waktu. InnoDB membuka semua log dan file tablespace sistem saat mysqld berjalan. Instans DB Anda memiliki nilai rendah untuk jumlah maksimum file yang dapat dibuka InnoDB pada satu waktu. Sebaiknya tetapkan parameter `innodb_op en_files` ke nilai minimum 65.	Atur `innodb_op en_files` parameter ke nilai minimum65.	Ya	InnoDB membuka file untuk MySQL

Tipe	Deskripsi	Rekomendasi	Diperlu n downtir	Informasi tambahan
`max_user_ connectio ns` Parameter rendah	Instans DB Anda memiliki nilai rendah untuk jumlah maksimum koneksi simultan untuk setiap akun basis data. Kami merekomen dasikan pengatura n `max_user_ connections` parameter ke angka yang lebih besar dari5.	Tingkatkan nilai `max_user_ connections` parameter ke angka yang lebih besar dari5.	Ya	Menetapkan Batas Sumber Daya Akun untuk MySQL
Baca Replika terbuka dalam mode yang dapat ditulis	Instans DB Anda memiliki replika baca dalam mode yang dapat ditulis, yang memungkinkan pembaruan dari klien. Kami menyarank an Anda mengatur `read_only` parameter `TrueIfReplica` agar replika baca tidak dalam mode yang dapat ditulis.	Tetapkan nilai `read_only` parameter ke`TrueIfReplica` .	Tidak	Praktik terbaik untuk mengonfigurasi parameter replikasi untuk Amazon RDS for MySQL di Blog Database AWS

Tipe	Deskripsi	Rekomendasi	Diperlu n downtir	Informasi tambahan
`innodb_de fault_row _format` peng n parameter tidak aman	Instans DB Anda mengalami masalah yang diketahui: Tabel yang dibuat dalam versi MySQL yang lebih rendah dari 8.0.26 dengan `row_format` set `COMPACT` ke `REDUNDANT` atau akan tidak dapat diakses dan tidak dapat dipulihkan ketika indeks melebihi 767 byte. Kami menyarank an Anda mengatur nilai `innodb_de fault_row _format` parameter ke`DYNAMIC`.	Tetapkan nilai `innodb_de fault_row _format` parameter ke`DYNAMIC`.	Tidak	Perubahan MySQL 8.0.26

Tipe	Deskripsi	Rekomendasi	Diperlu n downtir	Informasi tambahan
`general_l ogging` param dihidupkan	Pencatatan umum diaktifkan untuk instans DB Anda. Pengaturan ini berguna saat memecahkan masalah database. Namun, menyalaka n logging umum meningkatkan jumlah operasi I/O dan ruang penyimpanan yang dialokasikan, yang dapat mengakibatkan pertengkaran dan penurunan kinerja. Periksa persyarat an Anda untuk penggunaan logging umum. Kami menyarankan Anda mengatur nilai `general_logging` parameter ke`0`.	Periksa persyarat an Anda untuk penggunaan logging umum. Jika tidak wajib, kami sarankan Anda untuk mengatur nilai `general_l ogging` parameter ke`0`.	Tidak	Ikhtisar log basis data RDS for MySQL

Tipe	Deskripsi	Rekomendasi	Diperlu n downtir	Informasi tambahan
Instans RDS kurang disediaka n untuk kapasitas sistem	Kami menyarank an Anda menyetel kueri Anda untuk menggunakan memori yang lebih rendah atau menggunak an jenis instans DB dengan memori yang dialokasikan lebih tinggi. Ketika instance kehabisan memori, maka kinerja database terpengar uh.	Upsize kelas instance	Ya	Menskalakan Instans Amazon RDS Anda Secara Vertikal dan Horizontal di Blog Database AWS Jenis instans Amazon RDS Penetapan Harga

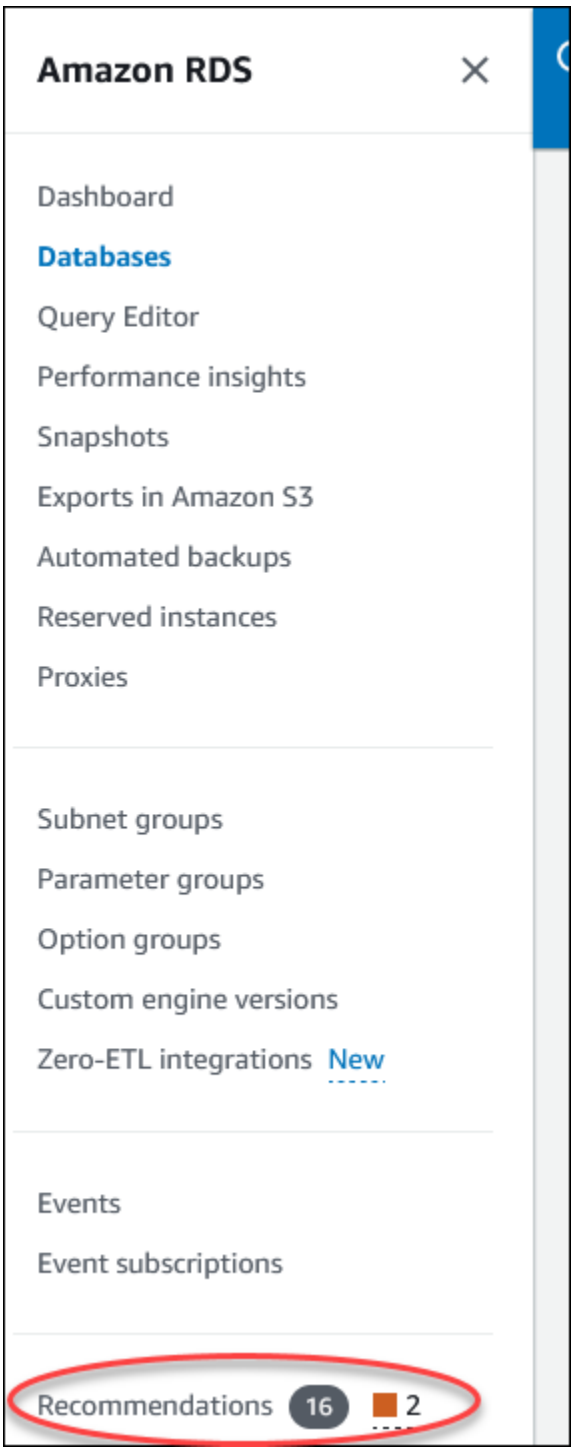
Menggunakan konsol Amazon RDS, Anda dapat melihat rekomendasi Amazon RDS Amazon untuk sumber daya database Anda.

Konsol

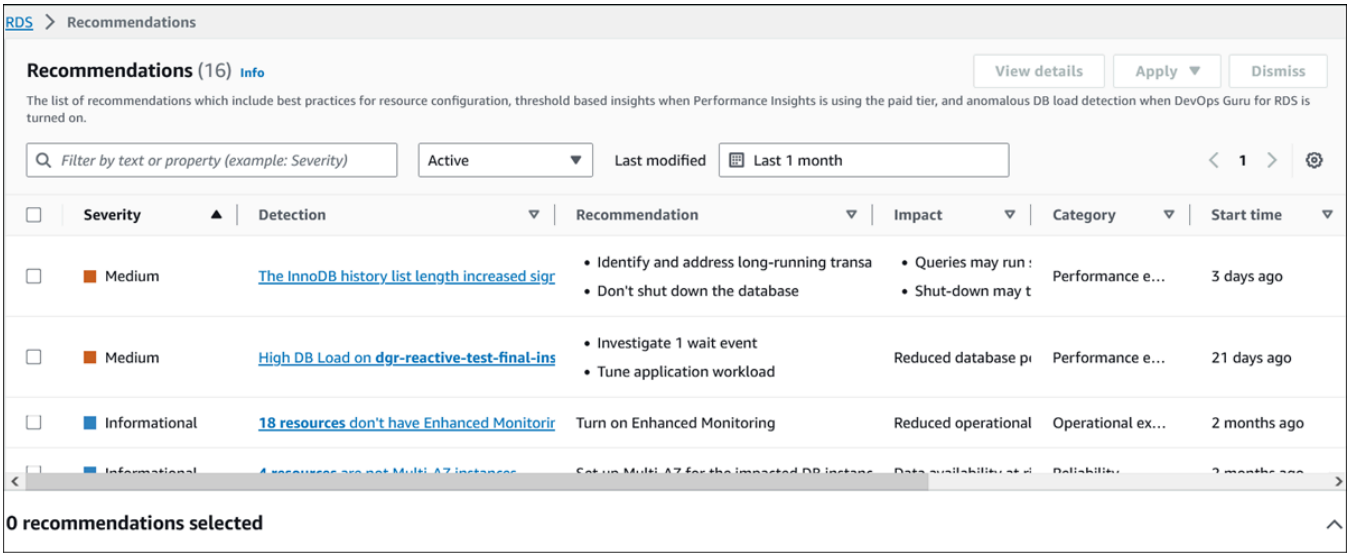
Untuk melihat rekomendasi Amazon RDS Aurora

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, lakukan salah satu hal berikut:

   - Pilih Rekomendasi. Jumlah rekomendasi aktif untuk sumber daya Anda dan jumlah rekomendasi dengan tingkat keparahan tertinggi tersedia di sebelah Rekomendasi. Untuk menemukan jumlah rekomendasi aktif untuk setiap tingkat keparahan, pilih angka yang menunjukkan tingkat keparahan paling tinggi.

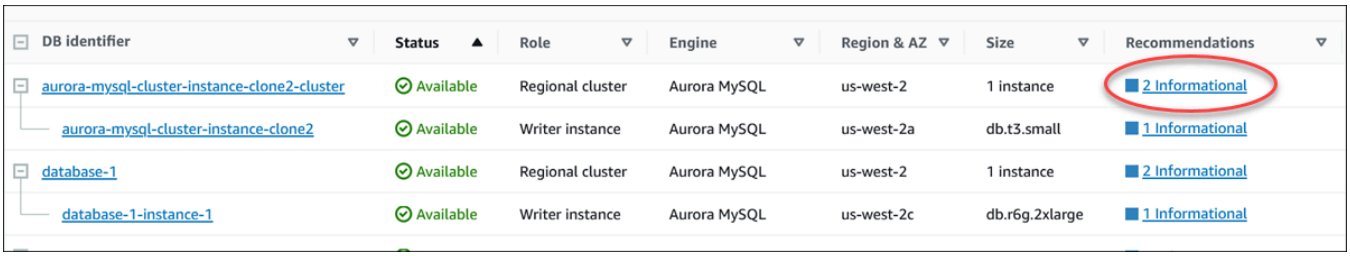


Halaman Rekomendasi menampilkan daftar rekomendasi yang diurutkan berdasarkan tingkat keparahan semua sumber daya di akun Anda.

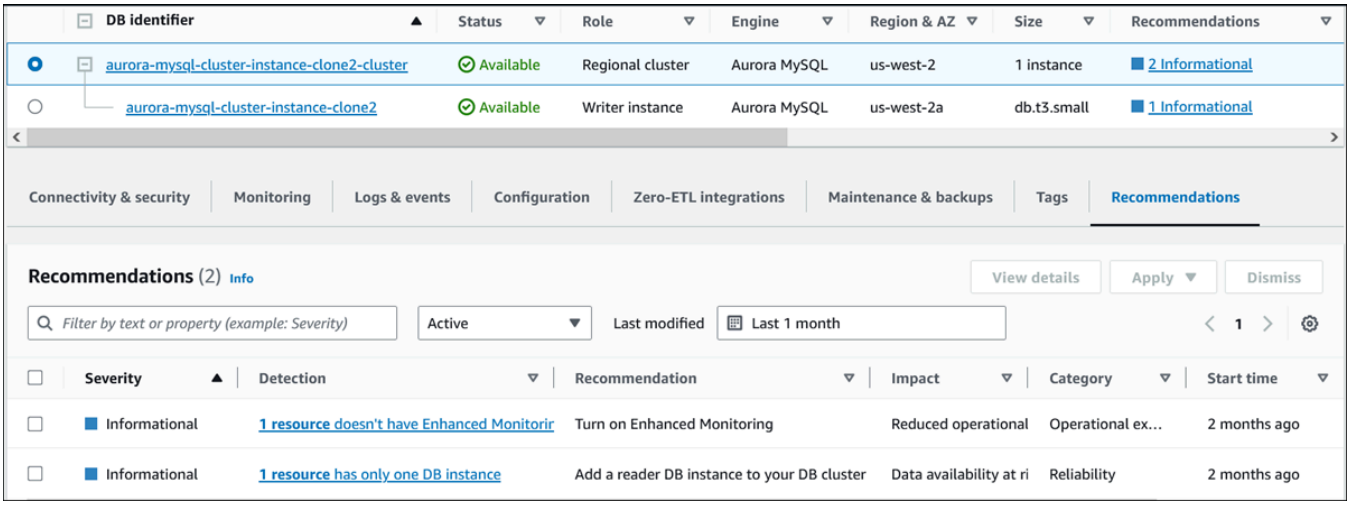


Anda dapat memilih rekomendasi untuk melihat bagian di bagian bawah halaman yang berisi sumber daya yang terpengaruh dan detail tentang bagaimana rekomendasi akan diterapkan.

- Di halaman Database, pilih Rekomendasi untuk sumber daya.



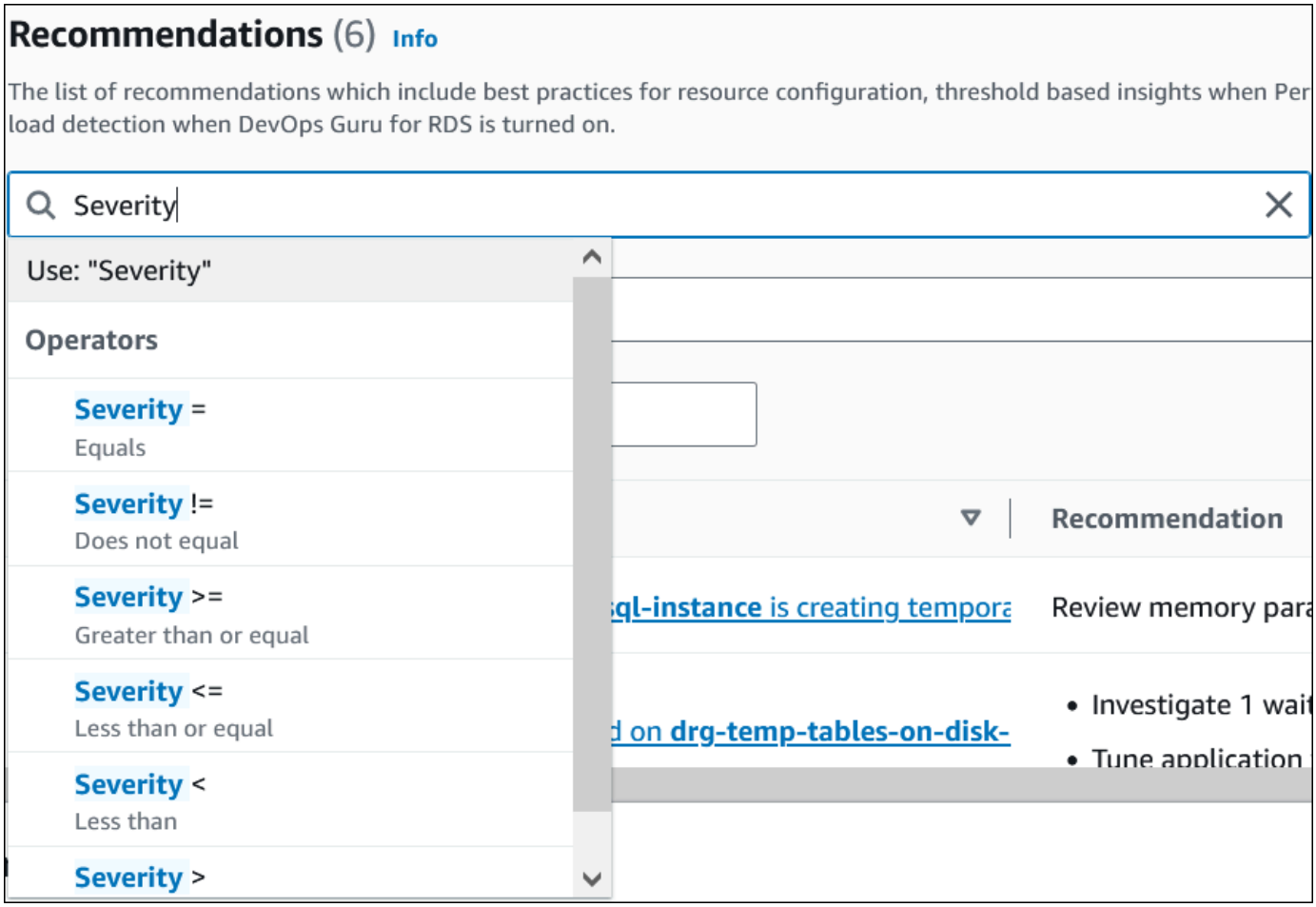
Tab Rekomendasi menampilkan rekomendasi dan detailnya untuk sumber daya yang dipilih.



Rincian berikut tersedia untuk rekomendasi:

- Keparahan — Tingkat implikasi dari masalah ini. Tingkat keparahannya adalah Tinggi, Sedang, Rendah, dan Informasi.
- Deteksi — Jumlah sumber daya yang terpengaruh dan deskripsi singkat tentang masalah ini. Pilih tautan ini untuk melihat rekomendasi dan detail analisis.
- Rekomendasi — Deskripsi singkat tentang tindakan yang disarankan untuk diterapkan.
- Dampak — Deskripsi singkat tentang kemungkinan dampak ketika rekomendasi tidak diterapkan.
- Kategori — Jenis rekomendasi. Kategori tersebut adalah Efisiensi kinerja, Keamanan, Keandalan, Optimalisasi biaya, keunggulan operasional, dan Keberlanjutan.
- Status — Status rekomendasi saat ini. Status yang mungkin adalah Semua, Aktif, Diberhentikan, Diselesaikan, dan Tertunda.
- Waktu mulai — Waktu ketika masalah dimulai. Misalnya, 18 jam yang lalu.
- Terakhir diubah - Waktu ketika rekomendasi terakhir diperbarui oleh sistem karena perubahan Tingkat Keparahan, atau waktu Anda menanggapi rekomendasi. Misalnya, 10 jam yang lalu.
- Waktu akhir - Waktu ketika masalah berakhir. Waktu tidak akan ditampilkan untuk masalah yang berkelanjutan.
- Pengenal sumber daya — Nama satu atau lebih sumber daya.

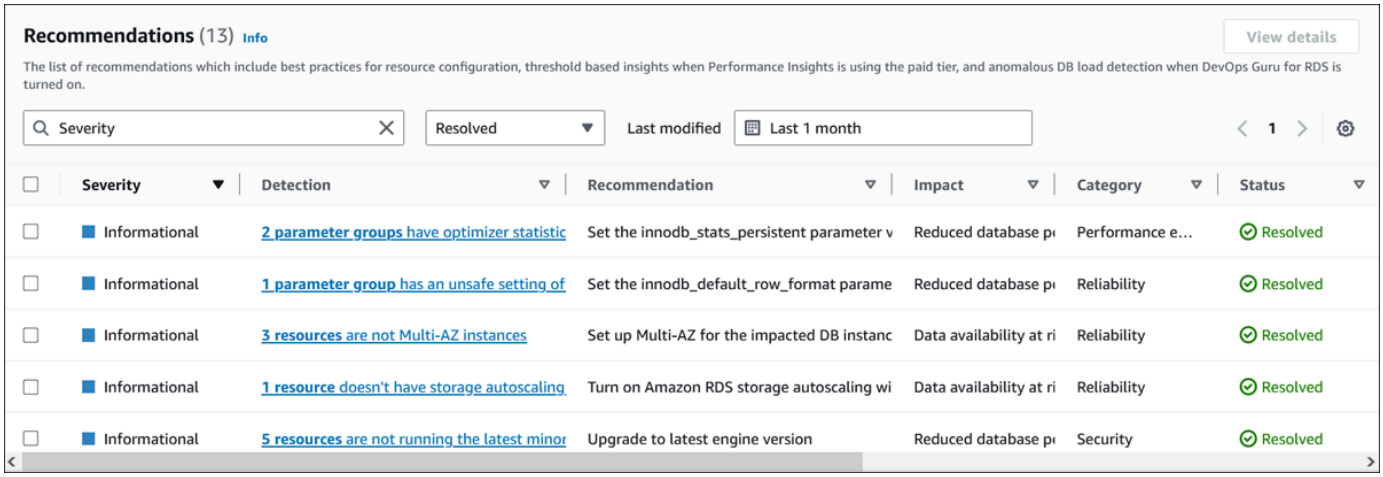
3. (Opsional) Pilih operator Keparahan atau Kategori di bidang untuk memfilter daftar rekomendasi.



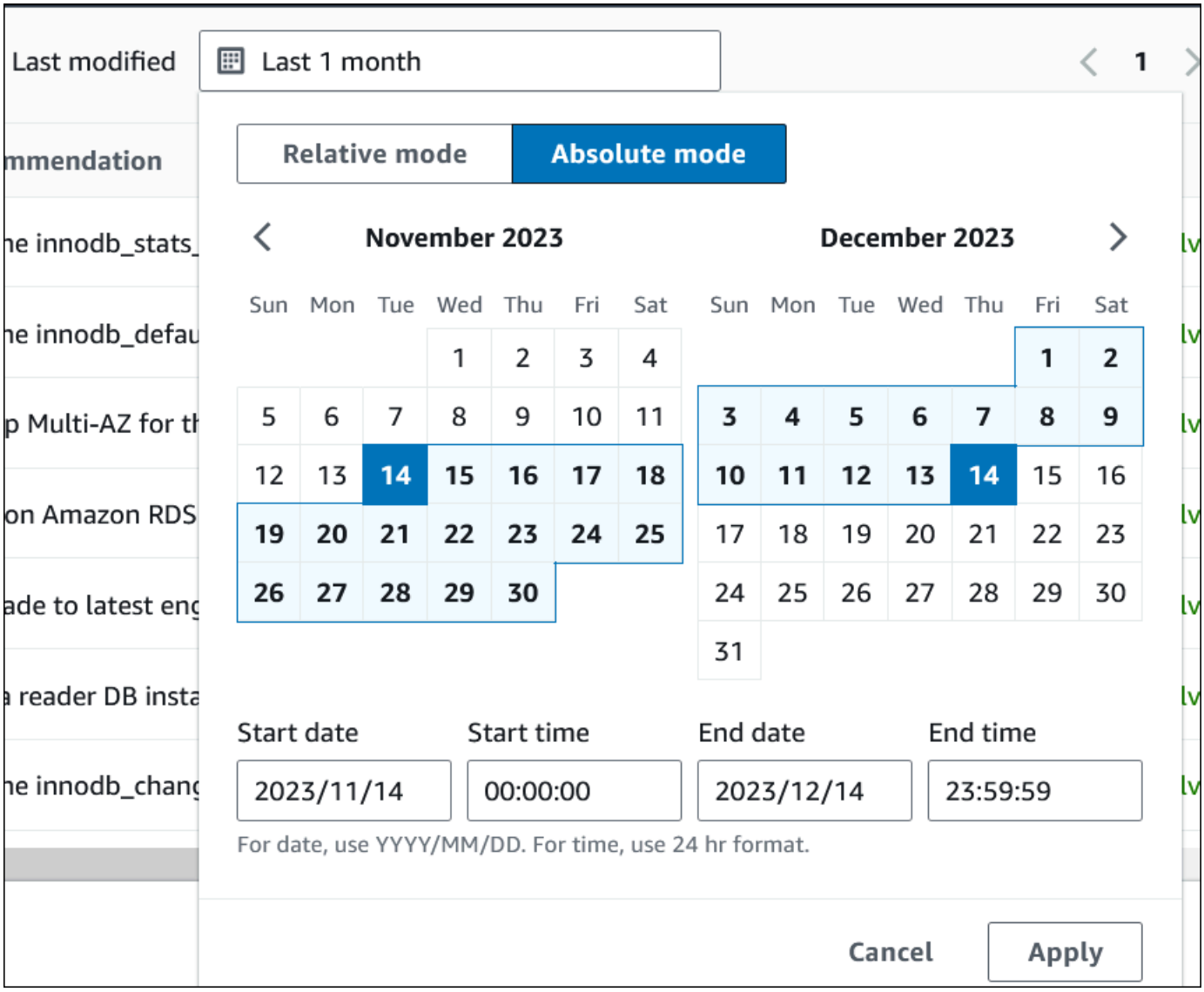
Rekomendasi untuk operasi yang dipilih muncul.

4. (Opsional) Pilih salah satu status rekomendasi berikut:

   - Aktif (default) - Menampilkan rekomendasi saat ini yang dapat Anda terapkan, menjadwalkannya untuk jendela pemeliharaan berikutnya, atau memberhentikan.
   - Semua - Menampilkan semua rekomendasi dengan status saat ini.
   - Diberhentikan — Menunjukkan rekomendasi yang diberhentikan.
   - Terselesaikan - Menunjukkan rekomendasi yang diselesaikan.
   - Tertunda — Menunjukkan rekomendasi yang tindakan rekomendasinya sedang berlangsung atau dijadwalkan untuk jendela pemeliharaan berikutnya.

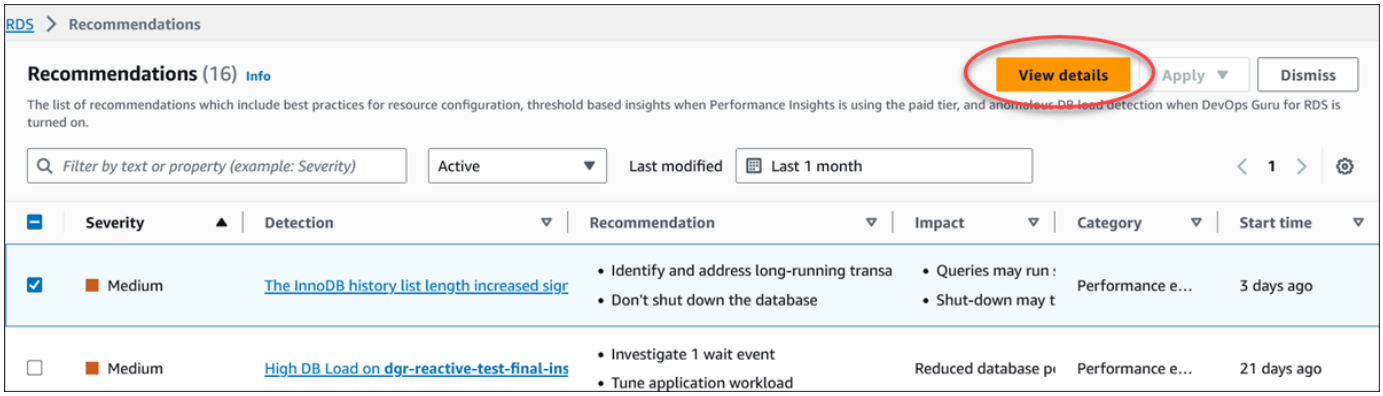


5. (Opsional) Pilih mode Relatif atau Mode absolut di Terakhir diubah untuk mengubah periode waktu untuk menampilkan rekomendasi. Dalam mode Absolute, Anda dapat memilih periode waktu, atau memasukkan waktu di bidang Tanggal mulai dan Tanggal akhir.



Rekomendasi untuk tampilan periode waktu yang ditetapkan.

6. (Opsional) Pilih Preferensi di sebelah kanan untuk menyesuaikan detail yang akan ditampilkan. Anda dapat memilih ukuran halaman, membungkus baris teks, dan mengizinkan atau menyembunyikan kolom.
7. (Opsional) Pilih rekomendasi dan kemudian pilih Lihat detail.



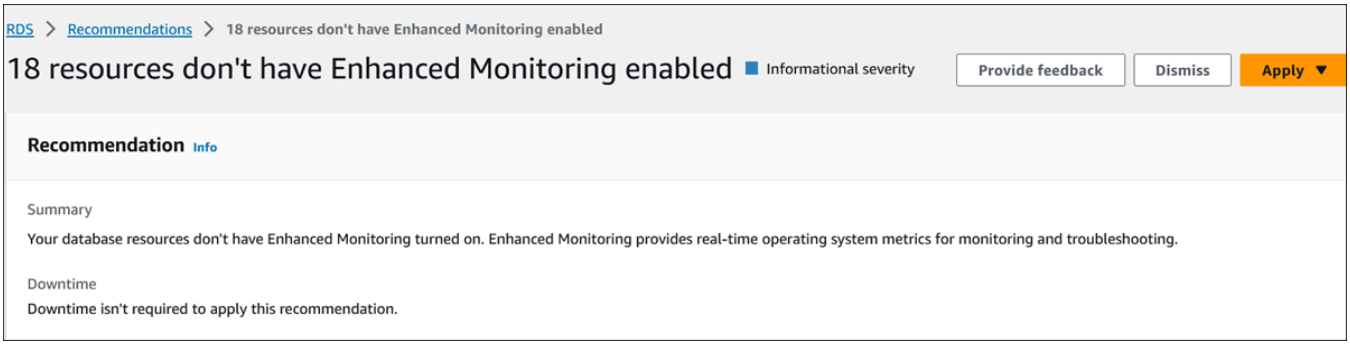
Halaman detail rekomendasi muncul. Judul memberikan jumlah total sumber daya dengan masalah yang terdeteksi dan tingkat keparahannya.

Untuk informasi tentang komponen di halaman detail untuk rekomendasi reaktif berbasis anomali, lihat Melihat anomali reaktif di Panduan Pengguna Amazon Guru. DevOps

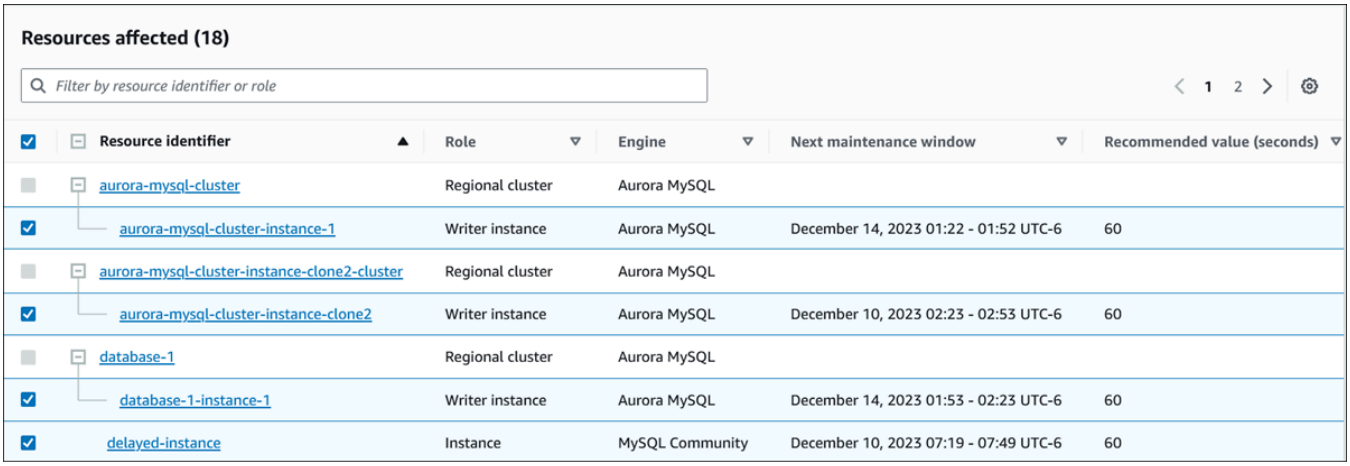
Untuk informasi tentang komponen pada halaman detail untuk rekomendasi proaktif berbasis ambang batas, lihatMelihat rekomendasi proaktif Performance Insights.

Rekomendasi otomatis lainnya menampilkan komponen berikut di halaman detail rekomendasi:

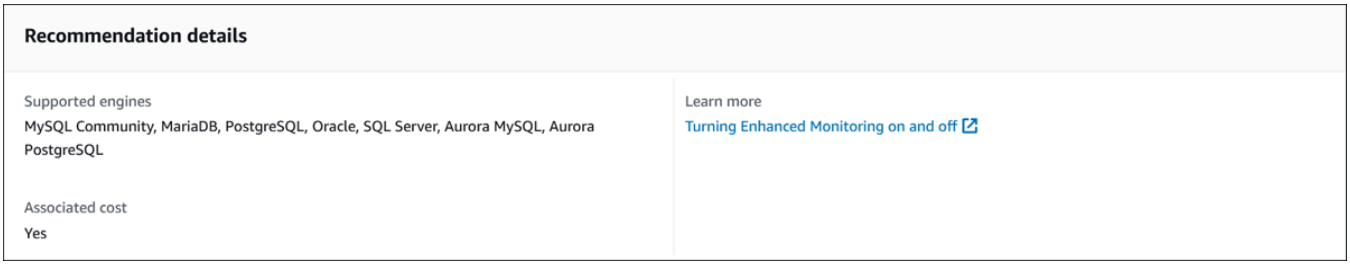
- Rekomendasi — Ringkasan rekomendasi dan apakah downtime diperlukan untuk menerapkan rekomendasi.



- Sumber daya yang terpengaruh — Detail sumber daya yang terpengaruh.



- Detail rekomendasi — Informasi mesin yang didukung, biaya terkait yang diperlukan untuk menerapkan rekomendasi, dan tautan dokumentasi untuk mempelajari lebih lanjut.



CLI

Untuk melihat rekomendasi Amazon RDS dari instans DB , gunakan perintah berikut di. AWS CLI

```
aws rds describe-db-recommendations
```

API RDS

Untuk melihat rekomendasi Amazon RDS menggunakan Amazon RDS API, gunakan operasi DescribedBrecommendations.

Menanggapi rekomendasi Amazon RDS

Dari daftar rekomendasi RDS , Anda dapat:

- Terapkan rekomendasi berbasis konfigurasi segera atau tunda hingga jendela pemeliharaan berikutnya.
- Singkirkan satu atau lebih rekomendasi.

- Pindahkan satu atau lebih rekomendasi yang diberhentikan ke rekomendasi aktif.

Menerapkan rekomendasi Amazon RDS Aurora

Menggunakan konsol Amazon RDS, pilih rekomendasi berbasis konfigurasi atau sumber daya yang terpengaruh di halaman detail, dan segera terapkan rekomendasi atau jadwalkan untuk jendela pemeliharaan berikutnya. Sumber daya mungkin perlu dimulai ulang agar perubahan diterapkan. Untuk beberapa rekomendasi grup parameter DB, Anda mungkin perlu memulai ulang sumber daya.

Rekomendasi reaktif berbasis proaktif atau anomali berbasis ambang batas tidak akan memiliki opsi penerapan dan mungkin memerlukan tinjauan tambahan.

Konsol

Untuk menerapkan rekomendasi berbasis konfigurasi

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, lakukan salah satu hal berikut:

   - Pilih Rekomendasi.

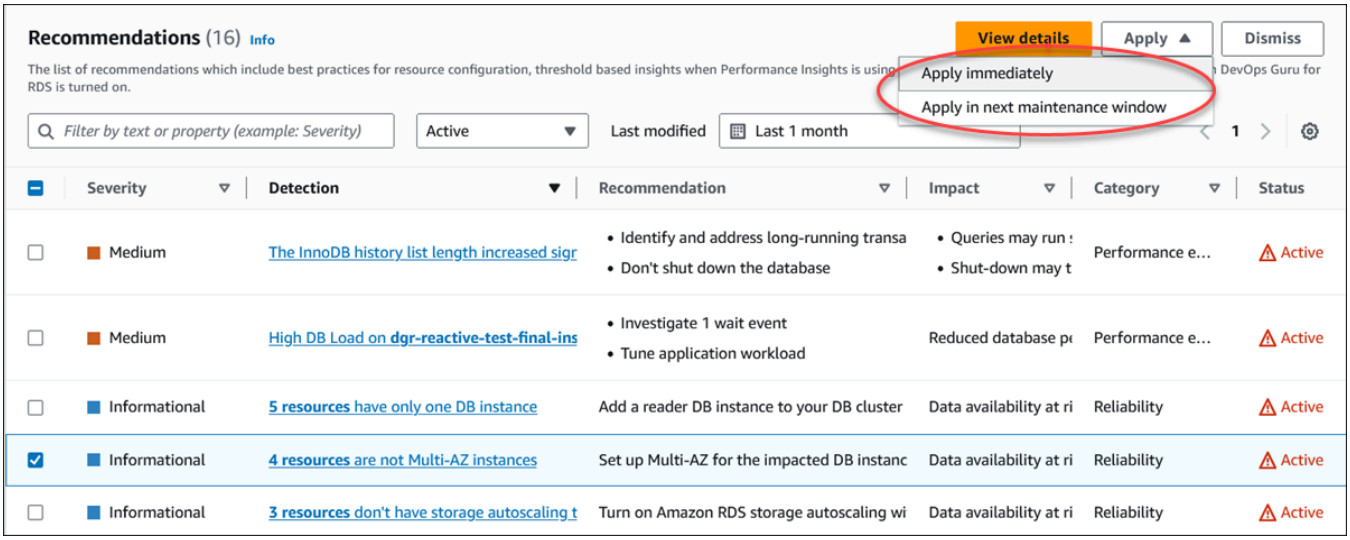
     Halaman Rekomendasi muncul dengan daftar semua rekomendasi.
   - Pilih Database dan kemudian pilih Rekomendasi untuk sumber daya di halaman database.

     Detailnya muncul di tab Rekomendasi untuk rekomendasi yang dipilih.
   - Pilih Deteksi untuk rekomendasi aktif di halaman Rekomendasi atau tab Rekomendasi di halaman Database.

     Halaman detail rekomendasi muncul.
3. Pilih rekomendasi, atau satu atau beberapa sumber daya yang terpengaruh di halaman detail rekomendasi, dan lakukan salah satu hal berikut:

   - Pilih Terapkan dan kemudian pilih Terapkan segera untuk segera menerapkan rekomendasi.
   - Pilih Terapkan dan kemudian pilih Terapkan di jendela pemeliharaan berikutnya untuk menjadwalkan di jendela pemeliharaan berikutnya.

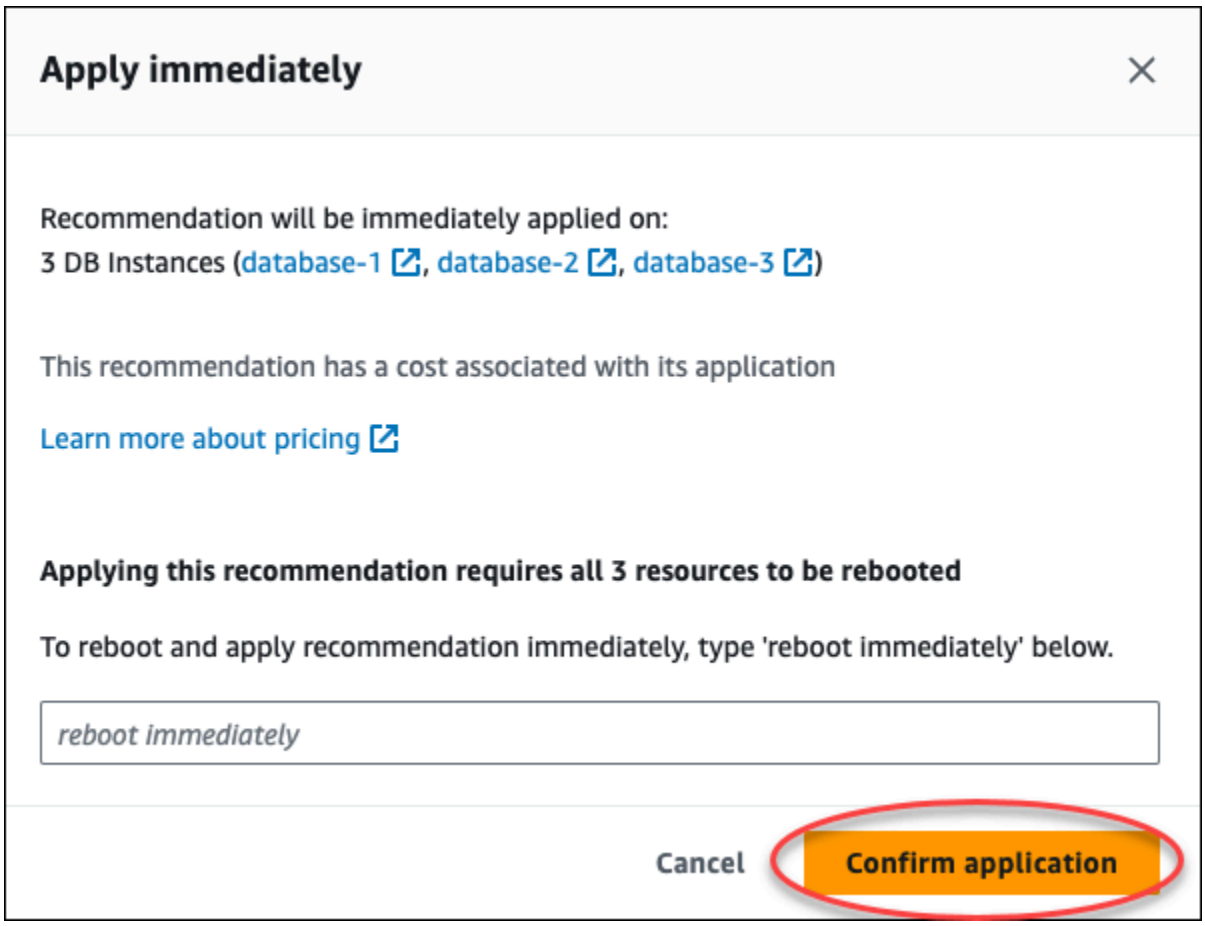
     Status rekomendasi yang dipilih diperbarui ke pending hingga jendela pemeliharaan berikutnya.



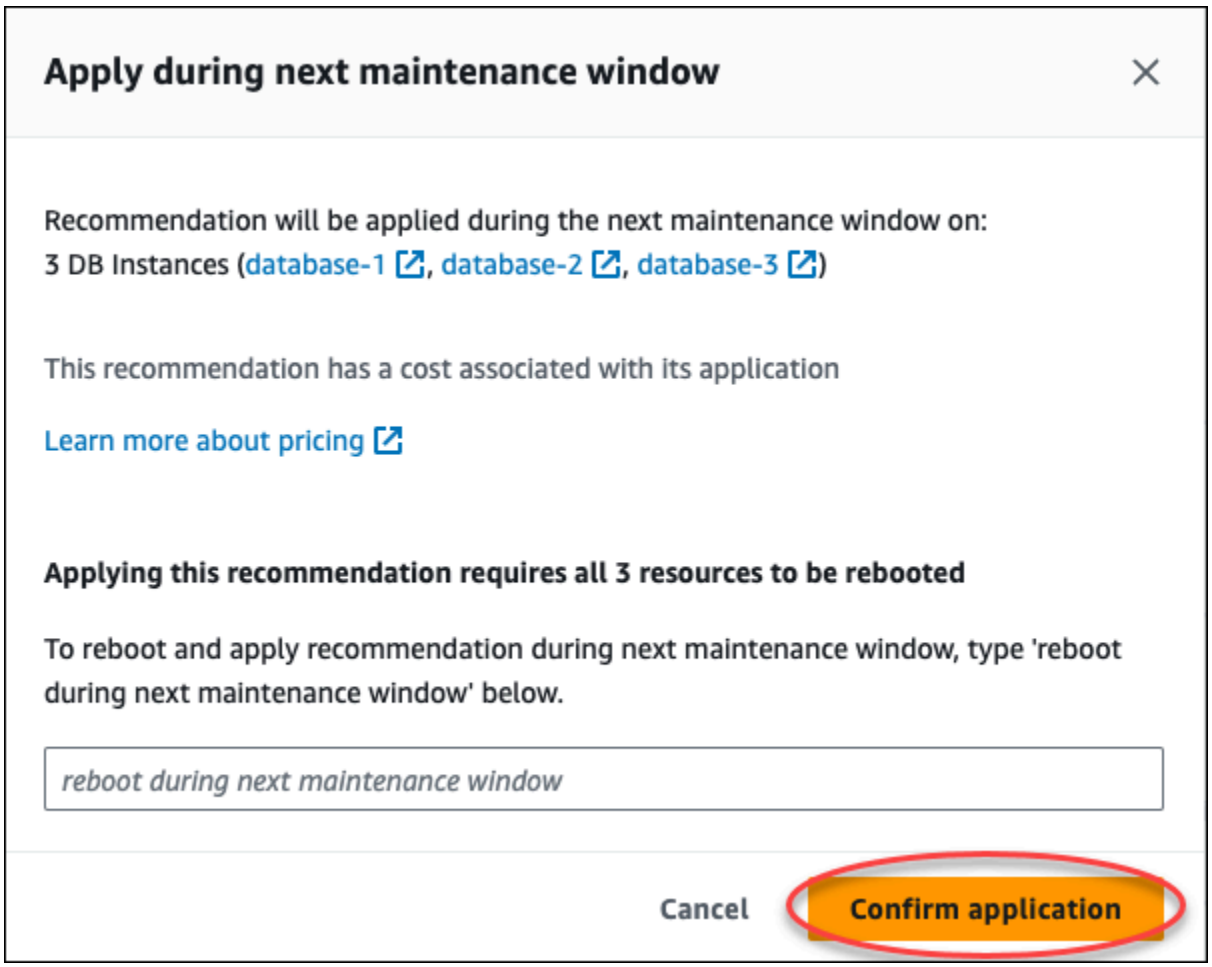
Jendela konfirmasi muncul.

4. Pilih Konfirmasi aplikasi untuk menerapkan rekomendasi. Jendela ini mengonfirmasi apakah sumber daya memerlukan restart otomatis atau manual agar perubahan diterapkan.

   Contoh berikut menunjukkan jendela konfirmasi untuk segera menerapkan rekomendasi.

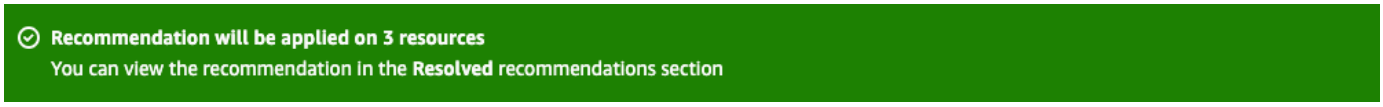


Contoh berikut menunjukkan jendela konfirmasi untuk menjadwalkan penerapan rekomendasi di jendela pemeliharaan berikutnya.



Spanduk menampilkan pesan ketika rekomendasi yang diterapkan berhasil atau gagal.

Contoh berikut menunjukkan spanduk dengan pesan yang berhasil.



Contoh berikut menunjukkan spanduk dengan pesan kegagalan.

API RDS

Untuk menerapkan rekomendasi RDS berbasis konfigurasi menggunakan Amazon RDS API

1. Gunakan operasi DescribedBrecommendations. `RecommendedActions`Dalam output dapat memiliki satu atau lebih tindakan yang direkomendasikan.
2. Gunakan RecommendedActionobjek untuk setiap tindakan yang direkomendasikan dari langkah 1. Outputnya berisi `Operation` dan`Parameters`.

   Contoh berikut menunjukkan output dengan satu tindakan yang direkomendasikan.

```
   "RecommendedActions": [
       {
           "ActionId": "0b19ed15-840f-463c-a200-b10af1b552e3",
           "Title": "Turn on auto backup", // localized
           "Description": "Turn on auto backup for my-mysql-instance-1", //
localized
           "Operation": "ModifyDbInstance",
           "Parameters": [
               {
                   "Key": "DbInstanceIdentifier",
                   "Value": "my-mysql-instance-1"
               },
               {
                   "Key": "BackupRetentionPeriod",
                   "Value": "7"
               }
           ],
           "ApplyModes": ["immediately", "next-maintenance-window"],
           "Status": "applied"
       },
       ... // several others
   ],
```

3. Gunakan `operation` untuk setiap tindakan yang direkomendasikan dari output pada langkah 2 dan masukkan `Parameters` nilainya.
4. Setelah operasi di langkah 2 berhasil, gunakan operasi ModifydBreCommendation untuk memodifikasi status rekomendasi.

Mengabaikan rekomendasi Amazon RDS Amazon

Anda dapat mengabaikan satu atau lebih rekomendasi.

Konsol

Untuk mengabaikan satu atau lebih rekomendasi

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, lakukan salah satu hal berikut:

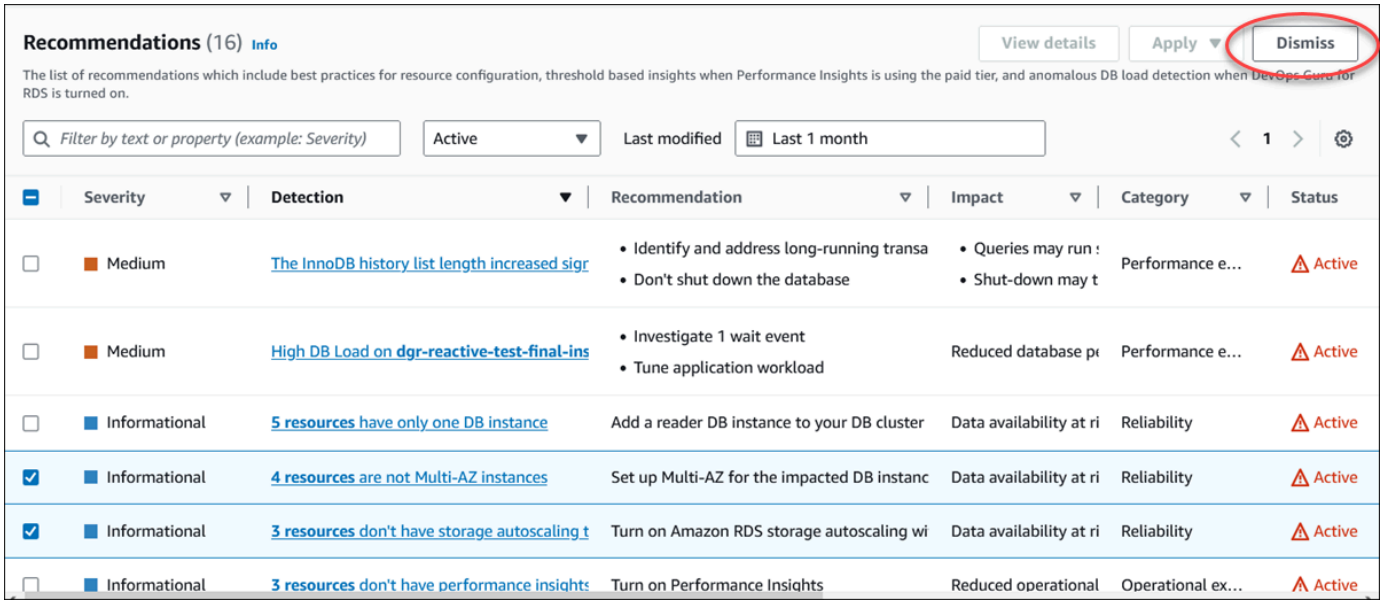
   - Pilih Rekomendasi.

     Halaman Rekomendasi muncul dengan daftar semua rekomendasi.
   - Pilih Database dan kemudian pilih Rekomendasi untuk sumber daya di halaman database.

     Detailnya muncul di tab Rekomendasi untuk rekomendasi yang dipilih.
   - Pilih Deteksi untuk rekomendasi aktif di halaman Rekomendasi atau tab Rekomendasi di halaman Database.

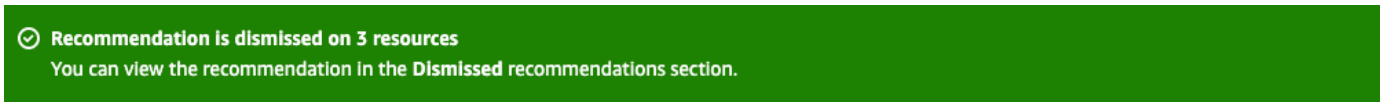
     Halaman detail rekomendasi menampilkan daftar sumber daya yang terpengaruh.
3. Pilih satu atau beberapa rekomendasi, atau satu atau beberapa sumber daya yang terpengaruh di halaman detail rekomendasi, lalu pilih Singkirkan.

   Contoh berikut menunjukkan halaman Rekomendasi dengan beberapa rekomendasi aktif yang dipilih untuk diberhentikan.



Spanduk menampilkan pesan ketika satu atau beberapa rekomendasi yang dipilih diberhentikan.

Contoh berikut menunjukkan spanduk dengan pesan yang berhasil.



Contoh berikut menunjukkan spanduk dengan pesan kegagalan.

CLI

Untuk mengabaikan rekomendasi RDS menggunakanAWS CLI

1. Jalankan perintah `aws rds describe-db-recommendations --filters "Name=status,Values=active"`.

   Output menyediakan daftar rekomendasi dalam `active` status.
2. Temukan `recommendationId` rekomendasi yang ingin Anda abaikan dari langkah 1.
3. Jalankan perintah `>aws rds modify-db-recommendation --status dismissed --recommendationId <ID>` dengan `recommendationId` dari langkah 2 untuk mengabaikan rekomendasi.

API RDS

Untuk mengabaikan rekomendasi RDS dan menggunakan Amazon RDS API, gunakan operasi ModifydBrecommendation.

Memodifikasi rekomendasi Amazon RDS Amazon Aurora yang diberhentikan menjadi

Anda dapat memindahkan satu atau lebih rekomendasi yang diberhentikan ke rekomendasi aktif.

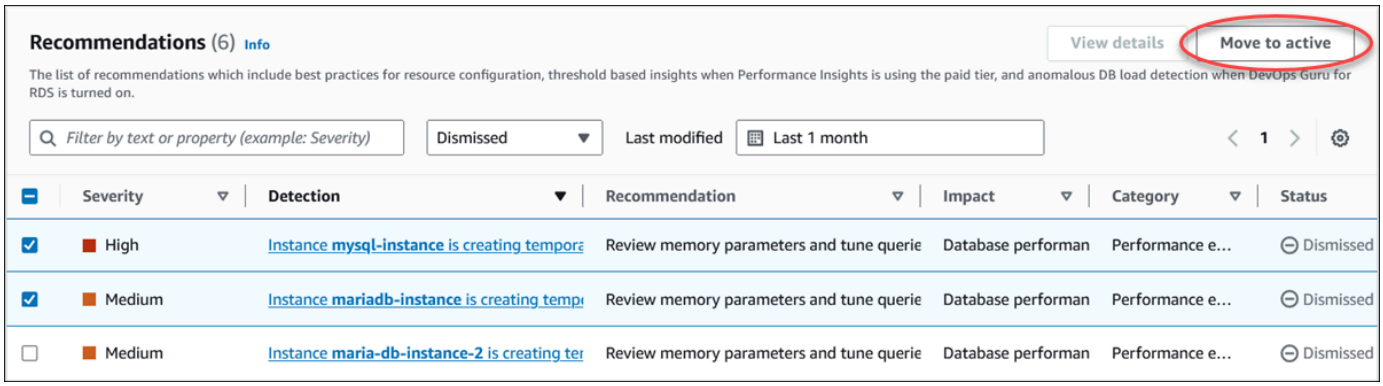
Konsol

Untuk memindahkan satu atau lebih rekomendasi yang diberhentikan ke rekomendasi aktif

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, lakukan salah satu hal berikut:
   - Pilih Rekomendasi.

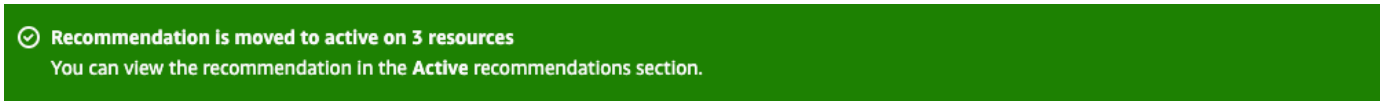
     Halaman Rekomendasi menampilkan daftar rekomendasi yang diurutkan berdasarkan tingkat keparahan semua sumber daya di akun Anda.
   - Pilih Database dan kemudian pilih Rekomendasi untuk sumber daya di halaman database.

     Tab Rekomendasi menampilkan rekomendasi dan detailnya untuk sumber daya yang dipilih.
3. Pilih satu atau beberapa rekomendasi yang diberhentikan dari daftar dan kemudian pilih Pindah ke aktif.



Spanduk menampilkan pesan yang berhasil atau gagal saat memindahkan rekomendasi yang dipilih dari diberhentikan ke status aktif.

Contoh berikut menunjukkan spanduk dengan pesan yang berhasil.



Contoh berikut menunjukkan spanduk dengan pesan kegagalan.

CLI

Untuk mengubah rekomendasi RDS yang diberhentikan menjadi rekomendasi aktif menggunakan AWS CLI

1. Jalankan perintah `aws rds describe-db-recommendations --filters "Name=status,Values=dismissed"`.

   Output menyediakan daftar rekomendasi dalam `dismissed` status.
2. Temukan `recommendationId` rekomendasi yang ingin Anda ubah statusnya dari langkah 1.
3. Jalankan perintah `>aws rds modify-db-recommendation --status active --recommendationId <ID>` dengan `recommendationId` dari langkah 2 untuk mengubah ke rekomendasi aktif.

API RDS

Untuk mengubah rekomendasi RDS yang diberhentikan menjadi rekomendasi aktif menggunakan Amazon RDS API, gunakan operasi ModifydBrecommendation.

Melihat metrik di konsol Amazon RDS

Amazon RDS terintegrasi dengan Amazon CloudWatch untuk menampilkan berbagai metrik instans DB RDS di konsol RDS. Untuk deskripsi metrik ini, lihat Referensi metrik untuk Amazon RDS.

Untuk instans DB, kategori metrik berikut dipantau:

- CloudWatch – Menampilkan metrik Amazon CloudWatch untuk RDS yang dapat diakses di konsol RDS. Anda dapat metrik-metrik ini di konsol CloudWatch. Setiap metrik berisi grafik yang menunjukkan metrik yang dipantau selama rentang waktu tertentu. Untuk daftar metrik CloudWatch, lihat CloudWatch Metrik Amazon untuk Amazon RDS.
- Pemantauan yang disempurnakan – Menampilkan ringkasan metrik sistem operasi saat instans DB Aurora telah mengaktifkan Pemantauan yang Disempurnakan. RDS mengirimkan metrik dari Pemantauan yang Disempurnakan ke akun Log Amazon CloudWatch Anda. Setiap metrik OS berisi grafik yang menunjukkan metrik yang dipantau selama rentang waktu tertentu. Untuk ringkasan, lihat Memantau metrik OS dengan Pemantauan yang Disempurnakan. Untuk daftar metrik Pemantauan yang Disempurnakan, lihat Metrik OS dalam Pemantauan yang Disempurnakan.
- Daftar Proses OS – Menampilkan detail setiap proses yang berjalan dalam instans DB Anda.
- Wawasan Performa – Membuka dasbor Wawasan Performa Amazon RDS untuk instans DB. Untuk ringkasan Wawasan Performa, lihat Memantau muatan DB dengan Wawasan Performa di Amazon RDS. Untuk daftar metrik Wawasan Performa, lihat CloudWatch Metrik Amazon untuk Performance Insights.

Amazon RDS kini menyediakan tampilan gabungan metrik Wawasan Performa dan CloudWatch di dasbor Wawasan Performa. Untuk menggunakan tampilan ini, Wawasan Performa harus diaktifkan instans DB Anda. Anda dapat memilih tampilan pemantauan baru di tab Pemantauan atau Wawasan Performa di panel navigasi. Untuk melihat petunjuk cara memilih tampilan ini, lihat Menampilkan metrik gabungan di konsol Amazon RDS.

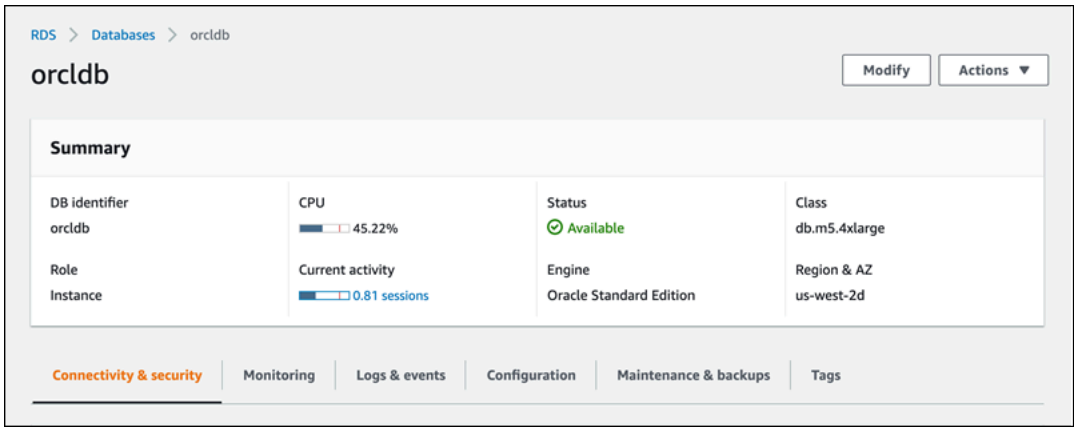
Jika Anda ingin melanjutkan tampilan pemantauan lama, lanjutkan dengan prosedur ini.

> **Note**
>
> Tampilan pemantauan lama akan dihentikan pada tanggal 15 Desember 2023.

Untuk melihat metrik instans DB Anda di tampilan pemantauan lama:

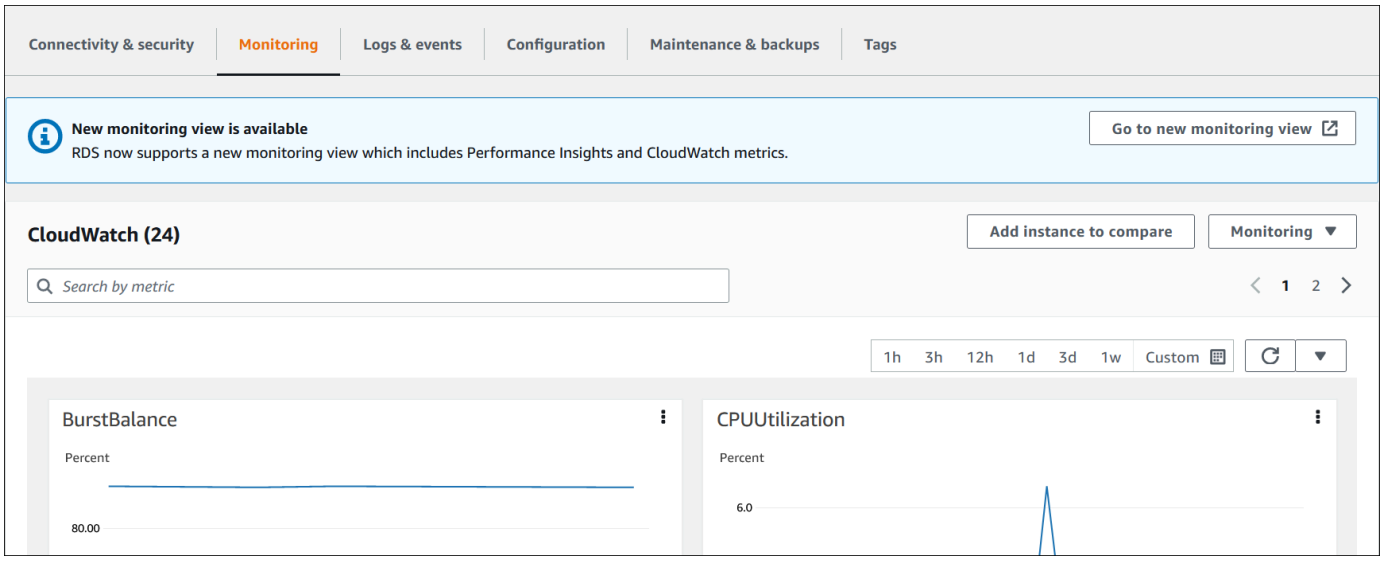
1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data.
3. Pilih nama instans DB yang ingin dipantau.

   Halaman basis data akan muncul. Contoh berikut menunjukkan basis data Oracle bernama `orclb`.

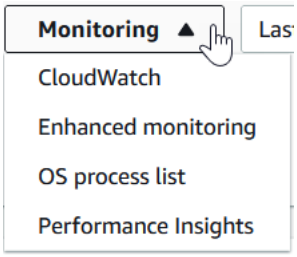


4. Gulir ke bawah dan pilih Pemantauan.

   Bagian pemantauan muncul. Secara default, metrik CloudWatch ditampilkan. Untuk deskripsi metrik ini, lihat CloudWatch Metrik Amazon untuk Amazon RDS.



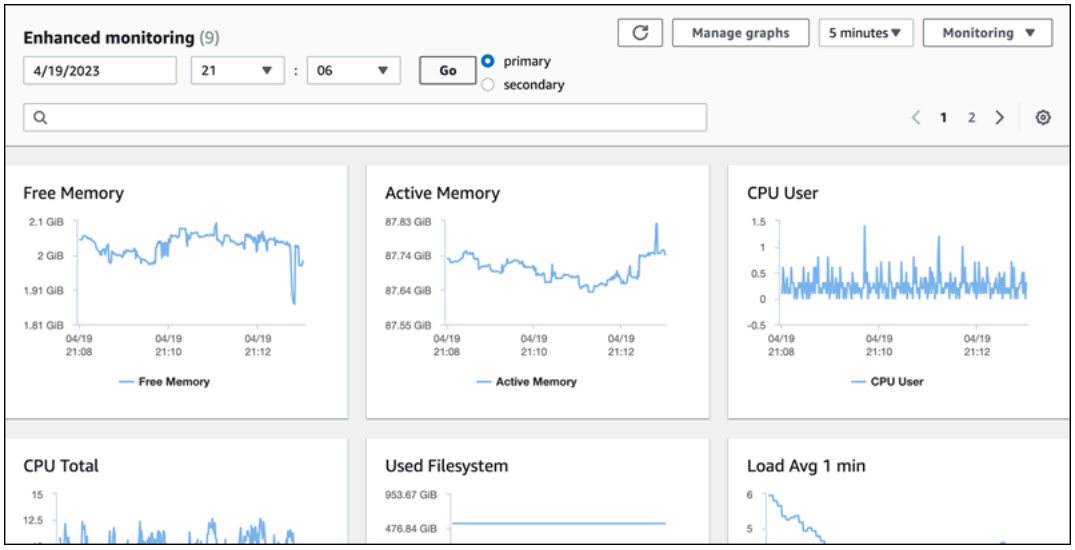
5. Pilih Pemantauan untuk melihat kategori metrik.



6. Pilih kategori metrik yang ingin dilihat.

   Contoh berikut menunjukkan metrik Pemantauan yang Disempurnakan. Untuk deskripsi metrik ini, lihat Metrik OS dalam Pemantauan yang Disempurnakan.

> **Note**
>
> Saat ini, melihat metrik OS untuk replika siaga Multi-AZ tidak didukung untuk instans DB MariaDB.



> **Tip**
>
> Untuk memilih rentang waktu metrik yang ditampilkan oleh grafik, Anda dapat menggunakan daftar rentang waktu.
> Untuk memunculkan tampilan yang lebih mendetail, Anda dapat memilih grafik mana pun. Anda juga dapat menerapkan filter spesifik metrik pada data.

Menampilkan metrik gabungan di konsol Amazon RDS

Amazon RDS kini menyediakan tampilan metrik Wawasan Performa dan CloudWatch yang terpadu untuk instans DB Anda di dasbor Wawasan Performa. Anda dapat menggunakan dasbor yang telah dikonfigurasi sebelumnya atau membuat dasbor kustom. Dasbor yang telah dikonfigurasi sebelumnya menyediakan metrik yang paling umum digunakan untuk membantu mendiagnosis masalah performa untuk mesin basis data. Alternatifnya, Anda dapat membuat dasbor kustom dengan metrik untuk mesin basis data yang memenuhi persyaratan analisis Anda. Kemudian, gunakan dasbor ini untuk semua instans DB dari jenis mesin basis data tersebut di akun AWS Anda.

Anda dapat memilih tampilan pemantauan baru di tab Pemantauan atau Wawasan Performa di panel navigasi. Saat menavigasi ke halaman Wawasan Performa, Anda akan melihat opsi untuk memilih antara tampilan pemantauan baru dan tampilan lama. Opsi yang Anda pilih disimpan sebagai tampilan default.

Wawasan Performa harus diaktifkan untuk instans DB Anda untuk melihat metrik gabungan di dasbor Wawasan Performa. Untuk informasi selengkapnya tentang mengaktifkan Wawasan Performa, lihat Mengaktifkan dan menonaktifkan Wawasan Performa.

> **Note**
>
> Sebaiknya Anda memilih tampilan pemantauan baru. Anda dapat terus menggunakan tampilan pemantauan lama hingga dihentikan pada 15 Desember 2023.

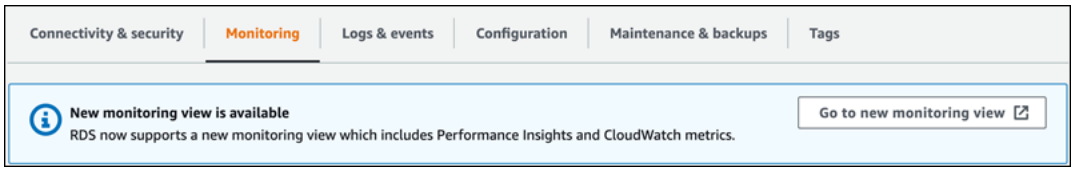
Memilih tampilan pemantauan baru di tab Pemantauan

Untuk memilih tampilan pemantauan baru di tab Pemantauan:

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Pada panel navigasi kiri, pilih Basis data.
3. Pilih instans DB yang ingin Anda pantau.

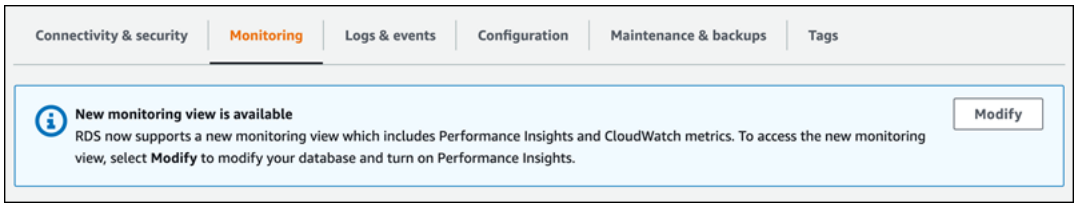
   Halaman basis data akan muncul.
4. Gulir ke bawah dan pilih tab Pemantauan.

Banner akan muncul dengan opsi untuk memilih tampilan pemantauan baru. Contoh berikut menampilkan banner untuk memilih tampilan pemantauan baru.



5. Pilih Buka tampilan pemantauan baru untuk membuka dasbor Wawasan Performa dengan metrik Wawasan Performa dan CloudWatch untuk instans DB Anda.
6. (Opsional) Jika Wawasan Performa dinonaktifkan untuk instans DB Anda, banner akan muncul dengan opsi untuk memodifikasi klaster DB dan mengaktifkan Wawasan Performa.

   Contoh berikut menampilkan banner untuk memodifikasi klaster DB di tab Pemantauan.



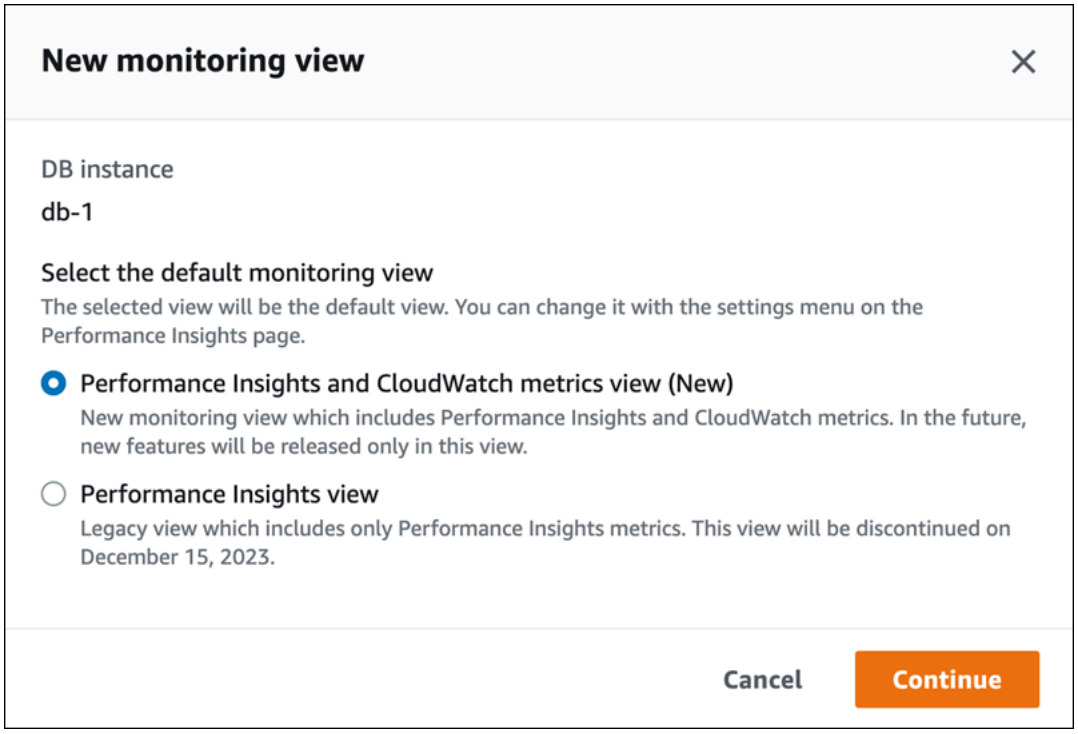
   Pilih Modifikasi untuk memodifikasi klaster DB dan mengaktifkan Wawasan Performa. Untuk informasi selengkapnya tentang mengaktifkan Wawasan Performa, lihat Mengaktifkan dan menonaktifkan Wawasan Performa.

Memilih tampilan pemantauan baru dengan Wawasan Performa di panel navigasi

Untuk memilih tampilan pemantauan baru dengan Wawasan Performa di panel navigasi:

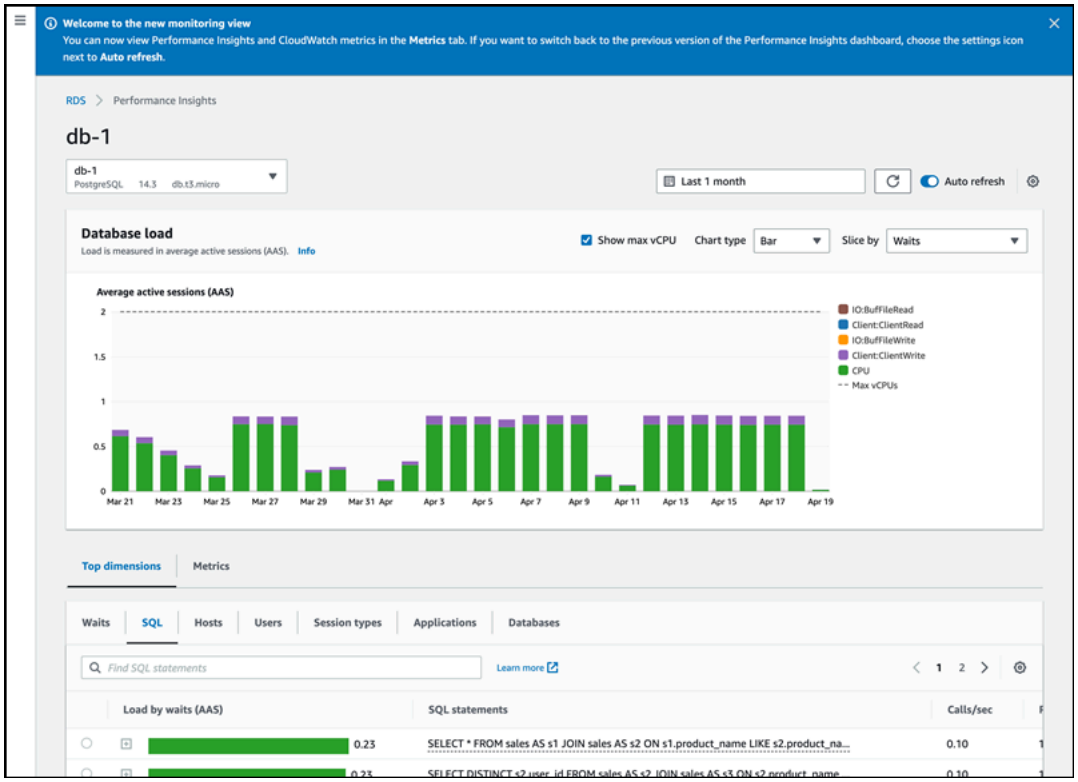
1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi kiri, pilih Wawasan Performa.
3. Pilih instans DB untuk membuka jendela yang memiliki opsi tampilan pemantauan.

   Contoh berikut menunjukkan jendela dengan opsi tampilan pemantauan.



4. Pilih opsi Tampilan metrik Wawasan Performa dan CloudWatch (Baru), lalu pilih Lanjutkan.

   Sekarang Anda dapat melihat dasbor Wawasan Performa yang menampilkan metrik Wawasan Performa dan CloudWatch untuk instans DB Anda. Contoh berikut menunjukkan metrik Wawasan Performa dan CloudWatch di dasbor.



Memilih tampilan lama dengan Wawasan Performa di panel navigasi

Anda dapat memilih tampilan pemantauan lama untuk hanya melihat metrik Wawasan Performa untuk instans DB Anda.

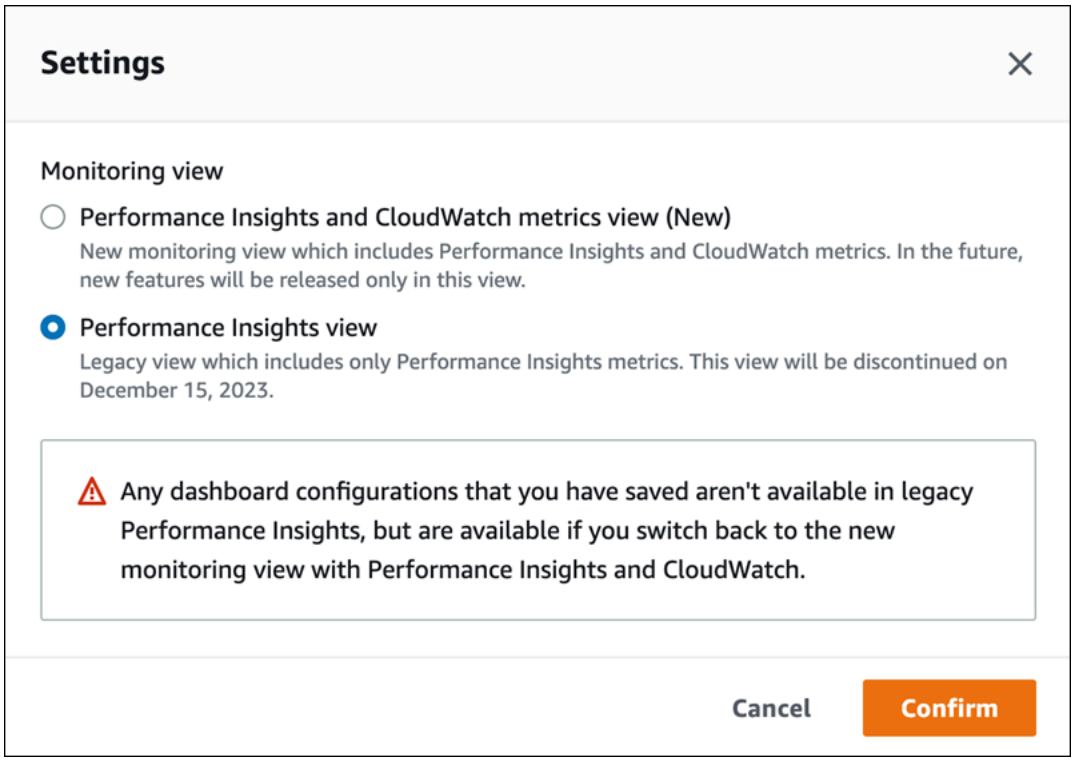
> **Note**
>
> Tampilan ini akan dihentikan pada 15 Desember 2023.

Untuk memilih tampilan pemantauan lama dengan Wawasan Performa di panel navigasi:

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi kiri, pilih Wawasan Performa.
3. Pilih instans DB.
4. Pilih ikon pengaturan di dasbor Wawasan Performa.

Sekarang Anda dapat melihat jendela Pengaturan yang menampilkan opsi untuk memilih tampilan Wawasan Performa lama.

Contoh berikut menunjukkan jendela dengan opsi untuk tampilan pemantauan lama.



5. Pilih opsi Tampilan Wawasan Performa dan pilih Lanjutkan.

   Pesan peringatan akan muncul. Konfigurasi dasbor apa pun yang Anda simpan tidak akan tersedia dalam tampilan ini.
6. Pilih Konfirmasi untuk melanjutkan ke tampilan Wawasan Performa lama.

   Sekarang Anda dapat melihat dasbor Wawasan Performa yang hanya menampilkan metrik Wawasan Performa untuk instans DB.

Membuat dasbor kustom dengan Wawasan Performa di panel navigasi

Di tampilan pemantauan baru, Anda dapat membuat dasbor kustom dengan metrik yang Anda butuhkan untuk memenuhi persyaratan analisis Anda.

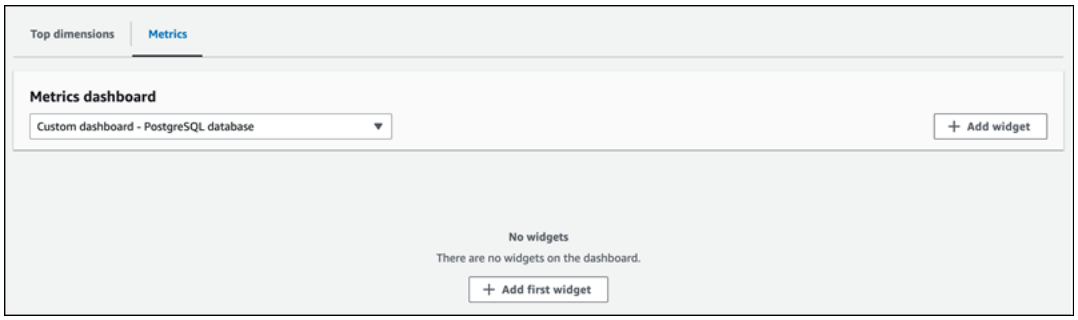
Anda dapat membuat dasbor kustom dengan memilih metrik Wawasan Performa dan CloudWatch untuk instans DB Anda. Anda dapat menggunakan dasbor kustom ini untuk instans DB lain dari jenis mesin basis data yang sama di akun AWS Anda.

> **Note**
>
> Dasbor yang disesuaikan mendukung hingga 50 metrik.

Gunakan menu pengaturan widget untuk mengedit atau menghapus dasbor, dan memindahkan atau mengubah ukuran jendela widget.

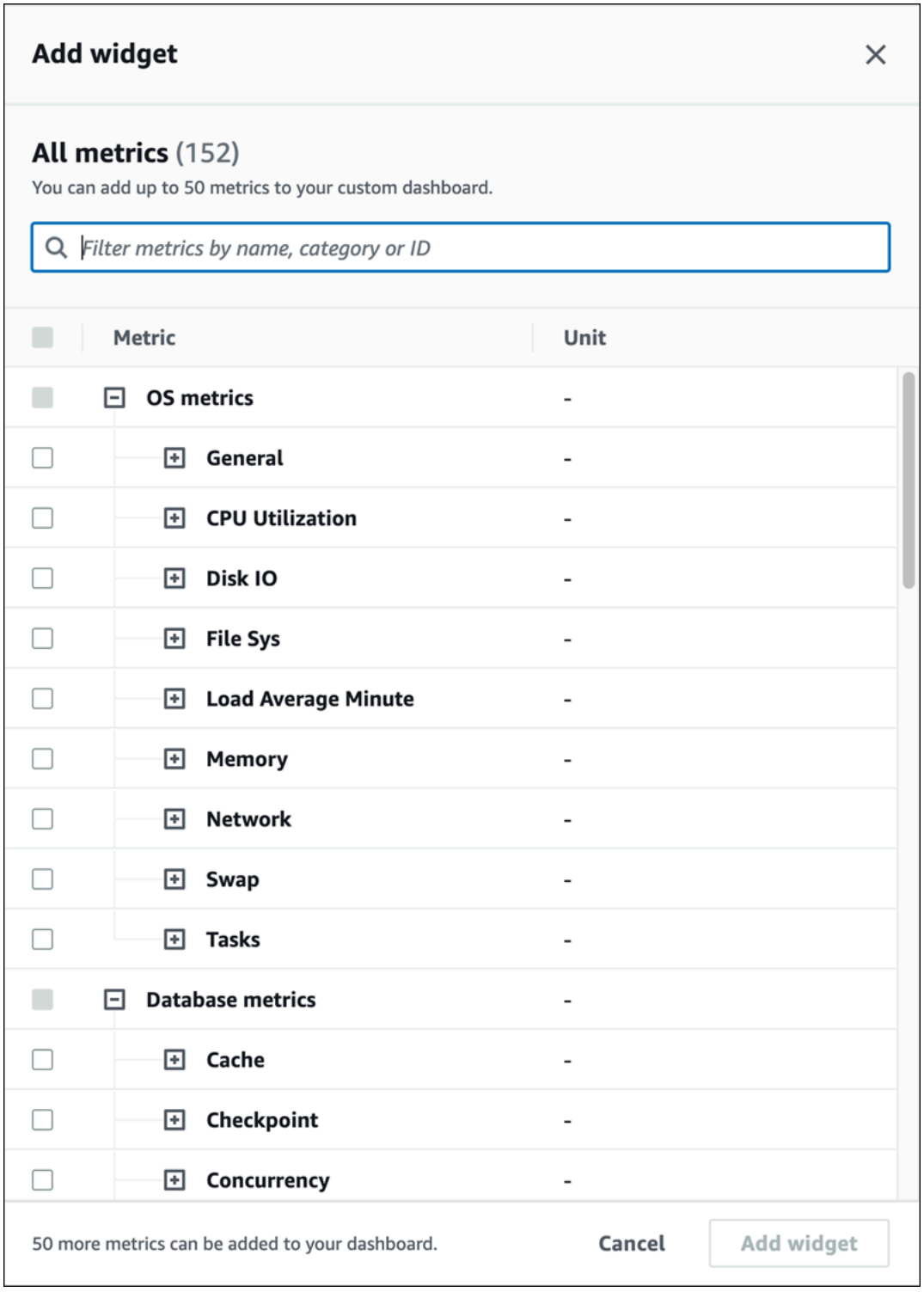
Untuk membuat dasbor kustom dengan Wawasan Performa di panel navigasi:

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi kiri, pilih Wawasan Performa.
3. Pilih instans DB.
4. Gulir ke bawah ke tab Metrik di jendela.
5. Pilih dasbor kustom dari daftar drop-down. Contoh berikut menunjukkan pembuatan dasbor kustom.



6. Pilih Tambahkan widget untuk membuka jendela Tambahkan widget. Anda dapat membuka dan melihat metrik sistem operasi (OS), metrik basis data, dan metrik CloudWatch yang tersedia di jendela.

   Contoh berikut menampilkan jendela Tambahkan widget dengan metrik.



7. Pilih metrik yang ingin Anda lihat di dasbor dan pilih Tambahkan widget. Anda dapat menggunakan bidang pencarian untuk menemukan metrik tertentu.

   Metrik yang dipilih akan muncul di dasbor Anda.

8. (Opsional) Jika Anda ingin memodifikasi atau menghapus dasbor Anda, pilih ikon pengaturan di kanan atas widget, lalu pilih salah satu tindakan berikut di menu.
   - Edit - Memodifikasi daftar metrik di jendela. Pilih Perbarui widget setelah Anda memilih metrik untuk dasbor Anda.
   - Hapus – Menghapus widget. Di jendela konfirmasi, pilih Hapus.

Memilih dasbor yang telah dikonfigurasi sebelumnya dengan Wawasan Performa di panel navigasi

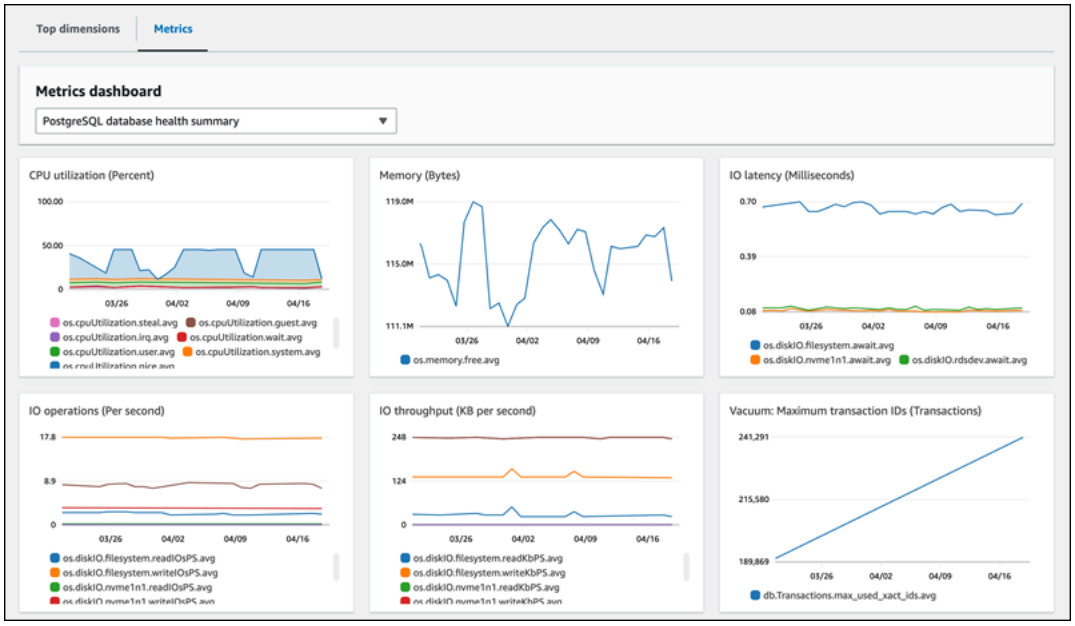
Anda dapat melihat metrik yang paling umum digunakan dengan dasbor yang telah dikonfigurasi sebelumnya. Dasbor ini membantu mendiagnosis masalah performa mesin basis data dan mengurangi waktu pemulihan rata-rata dari hitungan jam menjadi menit.

> **Note**
>
> Dasbor ini tidak dapat diedit.

Untuk memilih dasbor yang telah dikonfigurasi sebelumnya dengan Wawasan Performa di panel navigasi:

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi kiri, pilih Wawasan Performa.
3. Pilih instans DB.
4. Gulir ke bawah ke tab Metrik di jendela
5. Pilih dasbor yang telah dikonfigurasi sebelumnya dari daftar drop-down.

   Anda dapat melihat metrik untuk instans DB di dasbor. Contoh berikut menampilkan dasbor metrik yang telah dikonfigurasi sebelumnya.


Top dimensions
Metrics
Metrics dashboard
PostgreSQL database health summary
CPU utilization (Percent)
100.00
50.00
0
03/26
04/02
04/09
04/16
os.cpuUtilization.steal.avg
os.cpuUtilization.guest.avg
os.cpuUtilization.irq.avg
os.cpuUtilization.wait.avg
os.cpuUtilization.user.avg
os.cpuUtilization.system.avg
Memory (Bytes)
119.0M
115.0M
111.1M
os.memory.free.avg
IO latency (Milliseconds)
0.70
0.39
0.08
os.diskIO.filesystem.await.avg
os.diskIO.nvme1n1.await.avg
os.diskIO.rdsdev.await.avg
IO operations (Per second)
17.8
8.9
os.diskIO.filesystem.readIOsPS.avg
os.diskIO.filesystem.writeIOsPS.avg
os.diskIO.nvme1n1.readIOsPS.avg
IO throughput (KB per second)
248
124
os.diskIO.filesystem.readKbPS.avg
os.diskIO.filesystem.writeKbPS.avg
os.diskIO.nvme1n1.readKbPS.avg
Vacuum: Maximum transaction IDs (Transactions)
241,291
215,580
189,869
db.Transactions.max_used_xact_ids.avg

Memantau metrik Amazon RDS dengan Amazon CloudWatch

Amazon CloudWatch adalah sebuah repositori metrik. Repositori ini mengumpulkan dan mengolah data mentah dari Amazon RDS menjadi metrik-metrik waktu nyaris nyata yang dapat dibaca. Lihat daftar lengkap metrik-metrik Amazon RDS yang dikirim ke CloudWatch di Referensi metrik untuk Amazon RDS

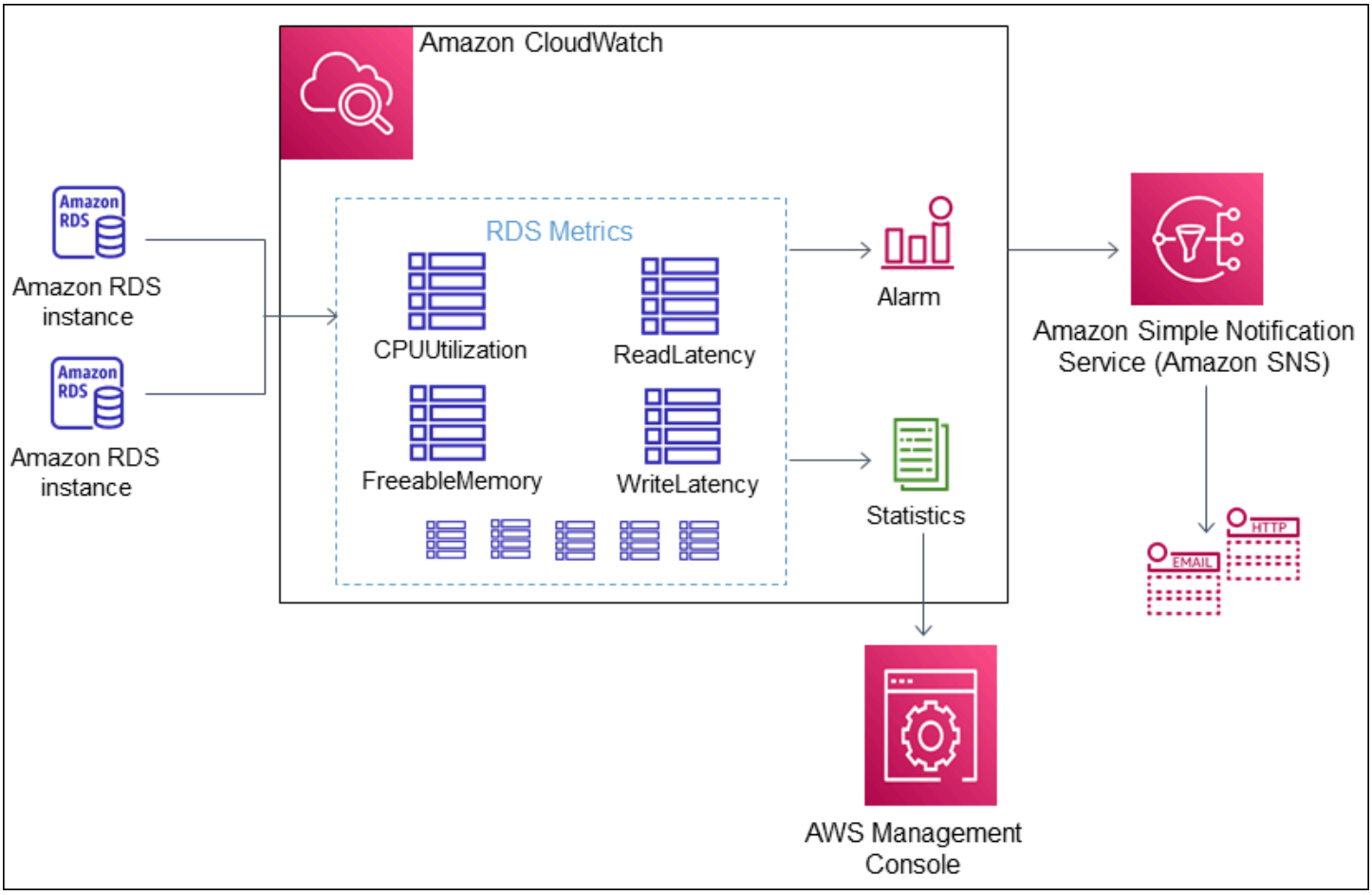
Topik

Ikhtisar Amazon RDS dan Amazon CloudWatch

Secara bawaan, Amazon RDS mengirim secara otomatis data metrik ke CloudWatch dalam periode-periode 1 menit. Misalnya, metrik `CPUUtilization` mencatat persentase pemanfaatan CPU untuk instans basis data dari waktu ke waktu. Titik data dengan periode 60 detik (1 menit) tersedia selama 15 hari. Ini berarti bahwa Anda dapat mengakses informasi historis dan melihat bagaimana aplikasi web atau layanan Anda berkinerja.

Anda kini mengekspor dasbor metrik Wawasan Performa dari Amazon RDS ke Amazon CloudWatch. Anda dapat mengekspor dasbor metrik yang telah dikonfigurasikan atau yang disesuaikan sebagai dasbor baru atau menambahkannya ke dasbor CloudWatch yang ada. Dasbor yang diekspor tersedia untuk dilihat di konsol CloudWatch. Lihat informasi yang lebih lengkap tentang cara mengekspor dasbor metrik Wawasan Performa ke CloudWatch di Mengekspor metrik Performance Insights ke CloudWatch.

Seperti ditunjukkan pada diagram berikut, Anda dapat menyiapkan alarm untuk metrik CloudWatch. Misalnya, Anda dapat membuat alarm yang memberi sinyal ketika penggunaan CPU untuk sebuah instans melebihi 70%. Anda dapat mengonfigurasikan Amazon Simple Notification Service agar mengirimi Anda email saat ambang batas itu dilewati.



Amazon RDS menerbitkan jenis-jenis metrik berikut ke Amazon CloudWatch:

- Metrik-metrik untuk instans basis data RDS

  Lihat tabel metrik-metrik ini di CloudWatch Metrik Amazon untuk Amazon RDS.

- Metrik-metrik Wawasan Performa

  Lihat tabel metrik-metrik ini di CloudWatch Metrik Amazon untuk Performance Insights dan Metrik penghitung Wawasan Performa.

- Metrik-metrik Pemantauan Disempurnakan (dipublikasikan ke Log Amazon CloudWatch)

  Lihat tabel metrik-metrik ini di Metrik OS dalam Pemantauan yang Disempurnakan.

- Metrik-metrik penggunaan untuk kuota layanan Amazon RDS di Akun AWS Anda

  Lihat tabel metrik-metrik ini di . Lihat informasi yang lebih lengkap tentang kuota Amazon RDS di Kuota dan batasan untuk Amazon RDS.

Lihat informasi yang lebih lengkap tentang CloudWatch di Apakah Amazon CloudWatch? dalam Panduan Pengguna Amazon CloudWatch. Lihat informasi yang lebih lengkap tentang retensi metrik CloudWatch di Retensi metrik.

Melihat metrik instans DB di CloudWatch konsol dan AWS CLI

Setelah itu, Anda dapat menemukan detail tentang cara melihat metrik untuk instans DB Anda menggunakan CloudWatch. Untuk informasi tentang metrik pemantauan untuk sistem operasi instans DB Anda secara real time menggunakan CloudWatch Log, lihatMemantau metrik OS dengan Pemantauan yang Disempurnakan.

Saat Anda menggunakan sumber daya Amazon RDS Aurora, Amazon RDS Amazon metrik dan dimensi ke Amazon setiap menit. CloudWatch

Sekarang Anda dapat mengekspor dasbor metrik Performance Insights dari Amazon RDS ke CloudWatch Amazon dan melihat metrik ini di konsol. CloudWatch Untuk informasi selengkapnya tentang cara mengekspor dasbor metrik Performance Insights ke, lihat. CloudWatch Mengekspor metrik Performance Insights ke CloudWatch

Gunakan prosedur berikut untuk melihat metrik Amazon RDS Amazon di CloudWatch konsol dan CLI.

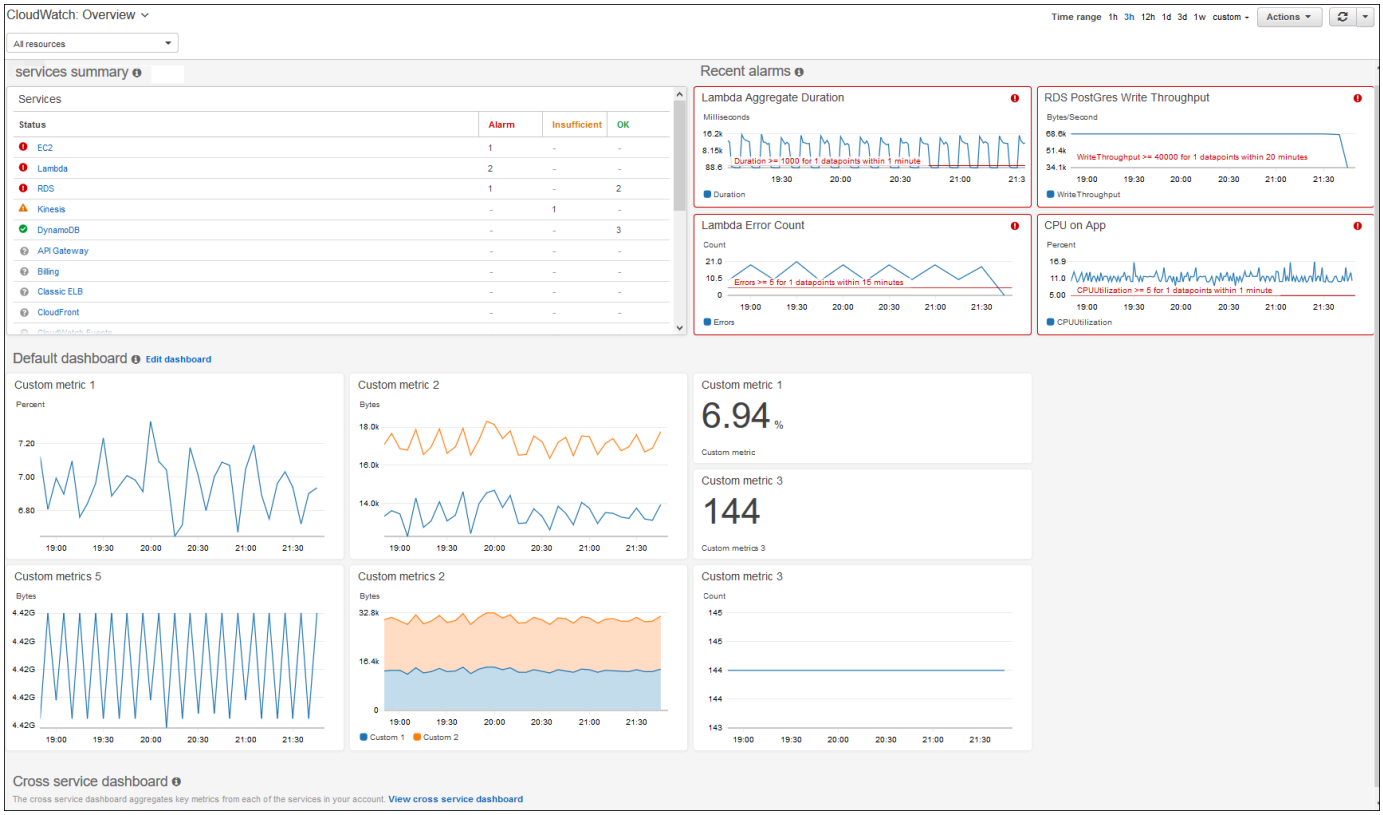
Konsol

Untuk melihat metrik menggunakan konsol Amazon CloudWatch

Metrik dikelompokkan berdasarkan ruang nama layanan dahulu, lalu berdasarkan berbagai kombinasi dimensi dalam setiap ruang nama.

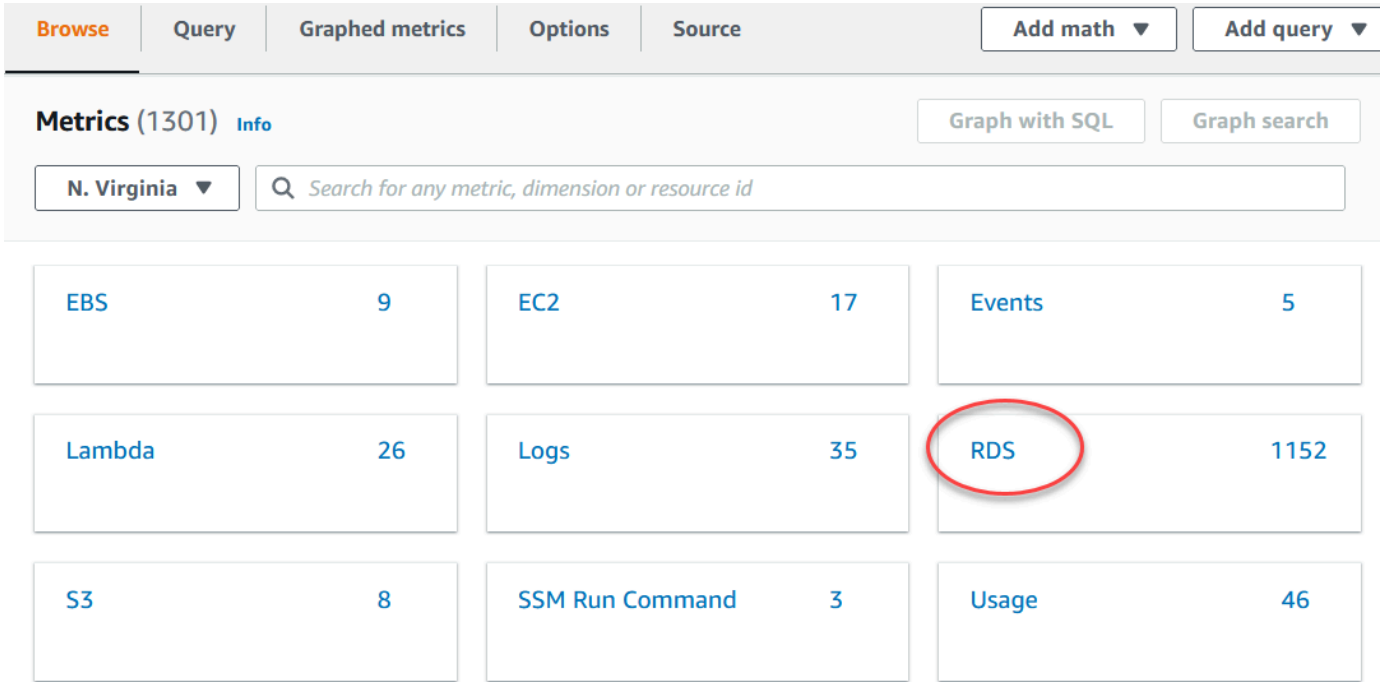
1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.

   Halaman beranda CloudWatch ikhtisar muncul.



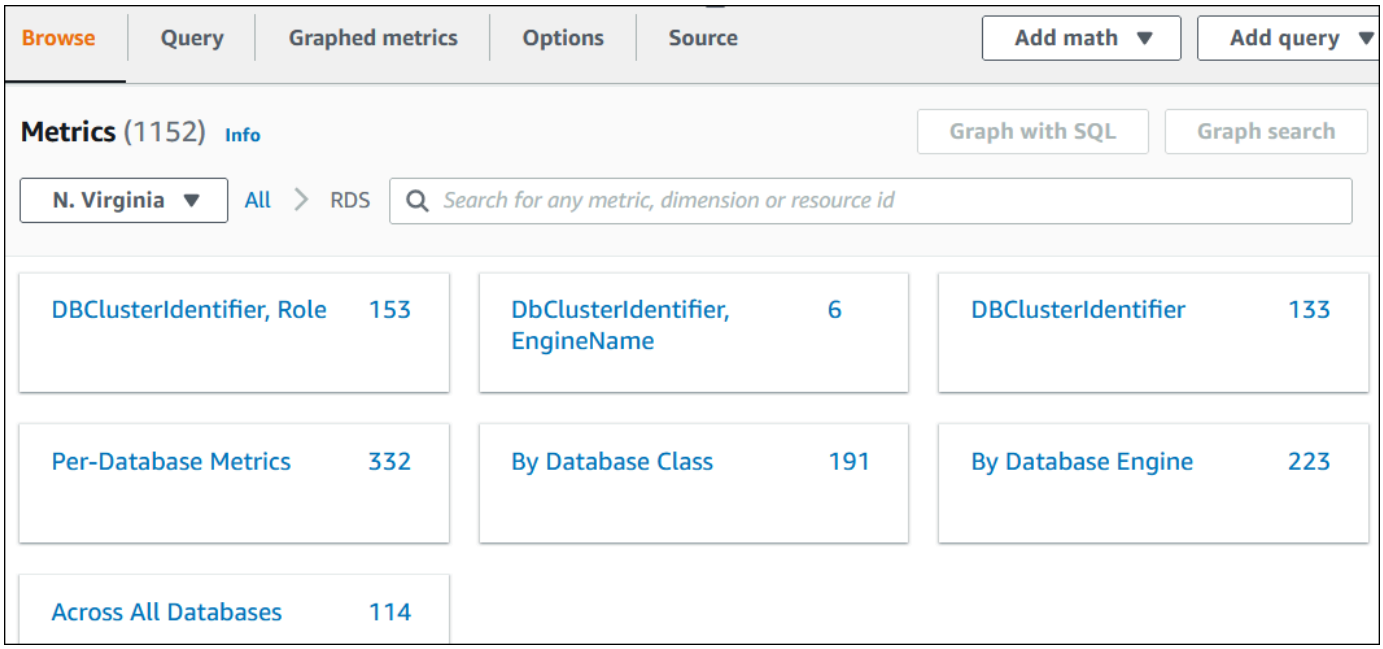
2. Jika perlu, ubah Wilayah AWS. Dari bilah navigasi, pilih Wilayah AWS tempat sumber daya AWS Anda berada. Lihat informasi yang lebih lengkap di Kawasan dan titik akhir.

3. Di panel navigasi, pilih Metrik, dan lalu Semua metrik.

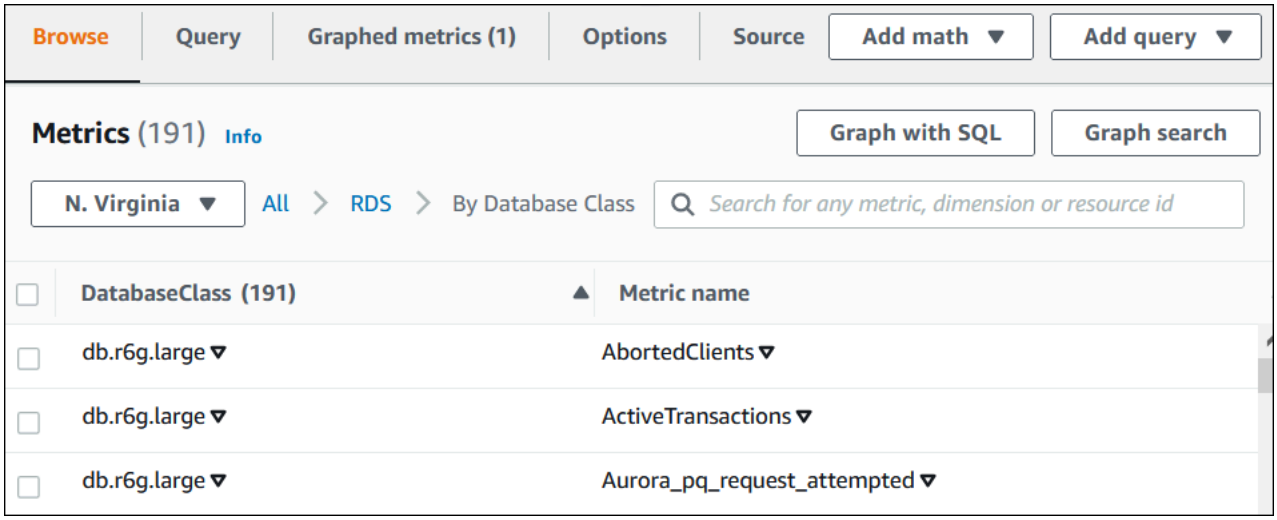


4. Gulir turun dan pilih ruang nama metrik RDS.

Halaman ini menampilkan dimensi-dimensi Amazon RDS Aurora. Untuk deskripsi semua dimensi ini, lihat Dimensi-dimensi Amazon CloudWatch untuk Amazon RDS.



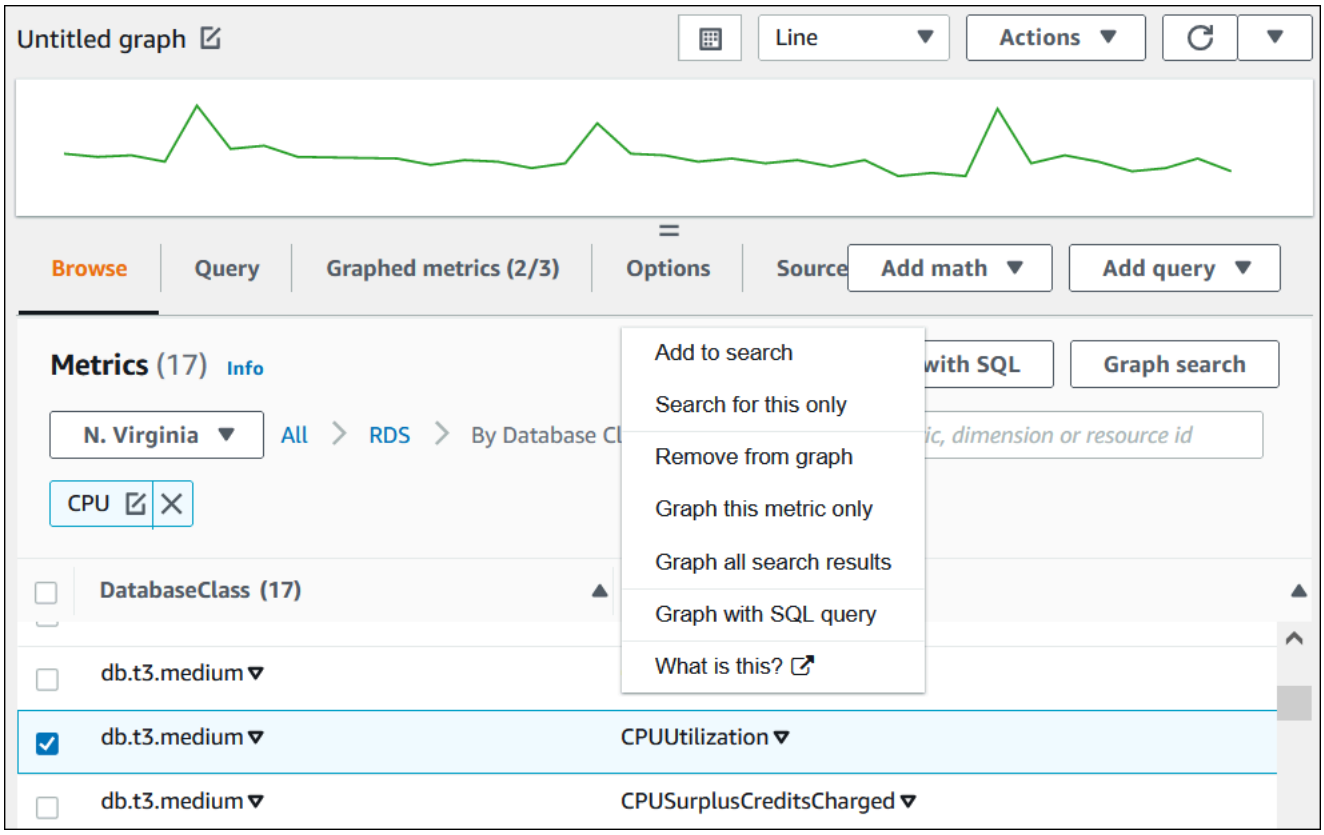
5. Pilih dimensi metrik, misalnya Berdasarkan Kelas Basis Data.



6. Lakukan salah satu tindakan berikut:

- Untuk mengurutkan metrik, gunakan judul kolom.
- Untuk membuat grafik metrik, pilih kotak centang di samping metrik.
- Untuk memfilter berdasarkan sumber daya, pilih ID sumber daya, lalu pilih Tambahkan ke pencarian.
- Untuk memfilter berdasarkan metrik, pilih nama metrik, lalu pilih Tambahkan ke pencarian.

Contoh berikut memfilter kelas db.t3.medium dan membuat grafik metrik CPUUtilization.



AWS CLI

Untuk mendapatkan informasi metrik dengan menggunakanAWS CLI, gunakan CloudWatch perintah `list-metrics`. Dalam contoh berikut, Anda memerinci semua metrik di ruang nama `AWS/RDS`.

```
aws cloudwatch list-metrics --namespace AWS/RDS
```

Untuk mendapatkan data metrik, gunakan perintah `get-metric-data`.

Contoh berikut mendapatkan `CPUUtilization` statistik misalnya `my-instance` selama periode 24 jam tertentu, dengan perincian 5 menit.

Buat file JSON `CPU_metric.json` dengan konten berikut.

```
{
   "StartTime" : "2023-12-25T00:00:00Z",
   "EndTime" : "2023-12-26T00:00:00Z",
   "MetricDataQueries" : [{
```

```
     "Id" : "cpu",
     "MetricStat" : {
    "Metric" : {
        "Namespace" : "AWS/RDS",
        "MetricName" : "CPUUtilization",
        "Dimensions" : [{ "Name" : "DBInstanceIdentifier" , "Value" : my-instance}]
    },
       "Period" : 360,
       "Stat" : "Minimum"
     }
   }]
}
```

Example

Untuk Linux, macOS, atau Unix:

```
aws cloudwatch get-metric-data \
    --cli-input-json file://CPU_metric.json
```

Untuk Windows:

```
aws cloudwatch get-metric-data ^
     --cli-input-json file://CPU_metric.json
```

Contoh output tampil sebagai berikut:

```
{
    "MetricDataResults": [
        {
            "Id": "cpu",
            "Label": "CPUUtilization",
            "Timestamps": [
                "2023-12-15T23:48:00+00:00",
                "2023-12-15T23:42:00+00:00",
                "2023-12-15T23:30:00+00:00",
                "2023-12-15T23:24:00+00:00",
                ...
            ],
            "Values": [
                13.299778337027714,
                13.677507543049558,
                14.24976250395827,
```

```
                13.02521708695145,
                ...
            ],
            "StatusCode": "Complete"
        }
    ],
    "Messages": []
}
```

Untuk informasi selengkapnya, lihat Mendapatkan statistik untuk metrik di Panduan CloudWatch Pengguna Amazon.

Mengekspor metrik Performance Insights ke CloudWatch

Performance Insights memungkinkan Anda mengekspor dasbor metrik yang telah dikonfigurasi sebelumnya atau kustom untuk instans DB Anda ke Amazon. CloudWatch Anda dapat mengekspor dasbor metrik sebagai dasbor baru atau menambahkannya ke CloudWatch dasbor yang ada. Saat Anda memilih untuk menambahkan dasbor ke CloudWatch dasbor yang ada, Anda dapat membuat label header sehingga metrik muncul di bagian terpisah di CloudWatch dasbor.

Anda dapat melihat dasbor metrik yang diekspor di konsol. CloudWatch Jika menambahkan metrik baru ke dasbor metrik Performance Insights setelah mengekspornya, Anda harus mengekspor dasbor ini lagi untuk melihat metrik baru di konsol. CloudWatch

Anda juga dapat memilih widget metrik di dasbor Performance Insights dan melihat data metrik di konsol. CloudWatch

Untuk informasi selengkapnya tentang melihat metrik di CloudWatch konsol, lihatMelihat metrik instans DB di CloudWatch konsol dan AWS CLI.

Mengekspor metrik Performance Insights sebagai dasbor baru CloudWatch

Pilih dasbor metrik yang telah dikonfigurasi sebelumnya atau kustom dari dasbor Performance Insights dan ekspor sebagai dasbor baru. CloudWatch Anda dapat melihat dasbor yang diekspor di CloudWatch konsol.

Untuk mengekspor dasbor metrik Performance Insights sebagai dasbor baru ke CloudWatch

1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi kiri, pilih Wawasan Performa.
3. Pilih instans DB.

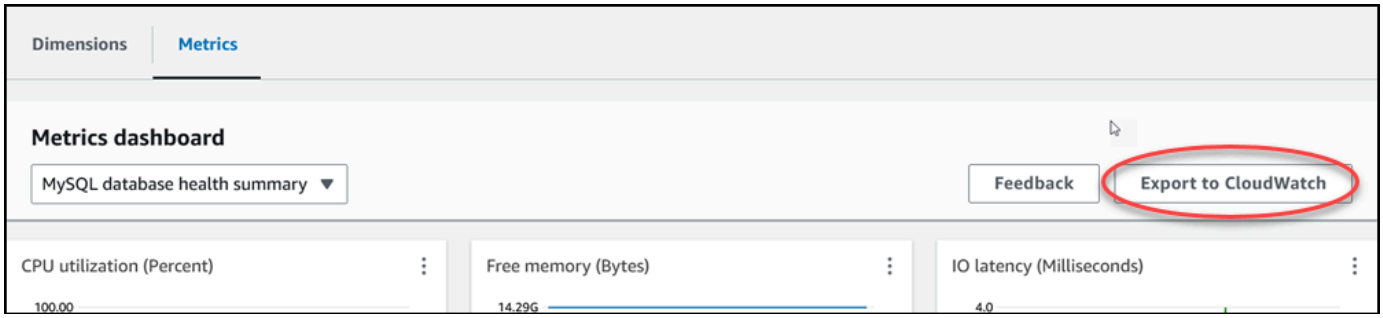
Dasbor Wawasan Performa ditampilkan untuk instans DB.

4. Gulir ke bawah dan pilih Metrik.

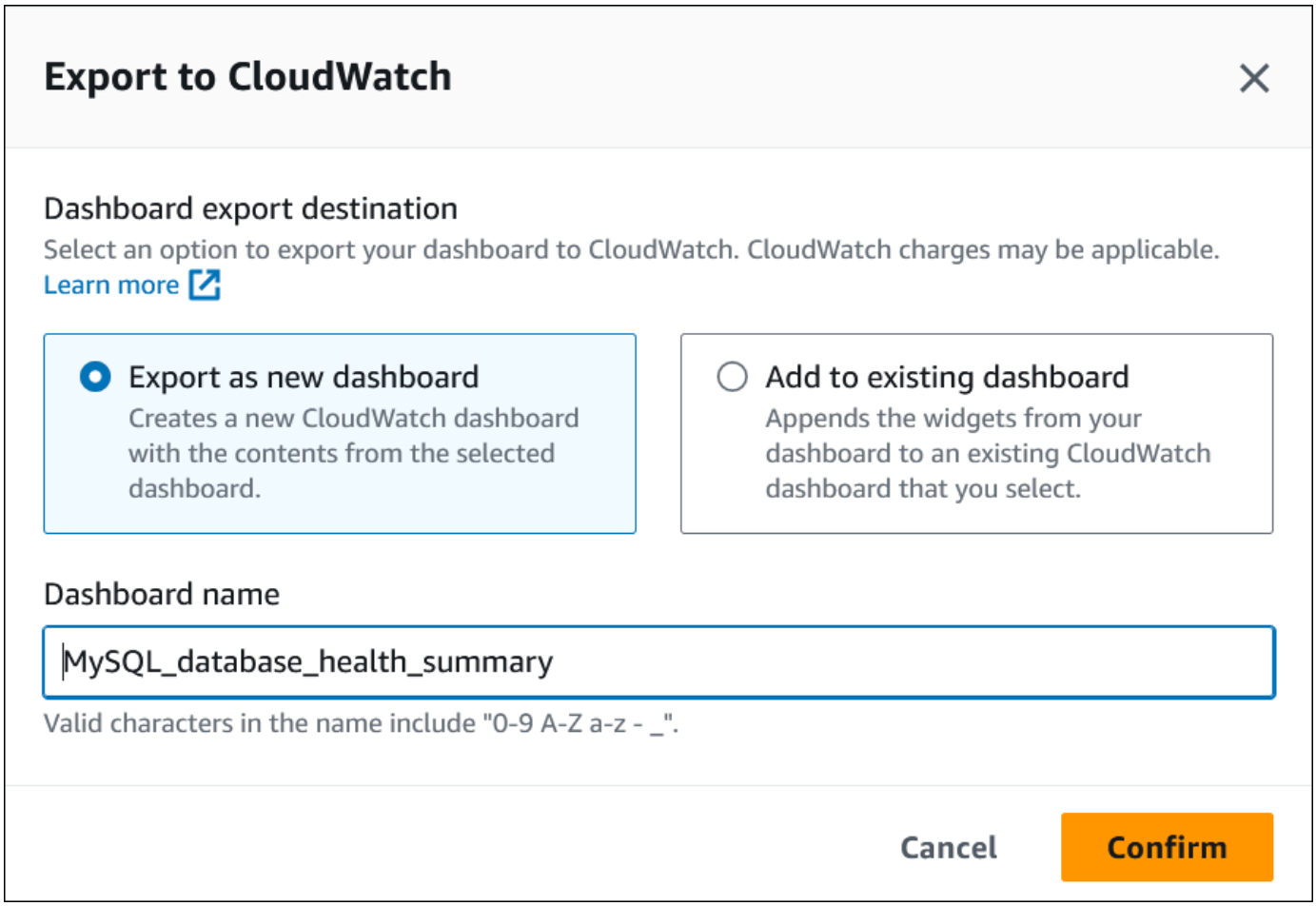
   Secara default, dasbor yang telah dikonfigurasi sebelumnya dengan metrik Wawasan Performa akan muncul.

5. Pilih dasbor yang telah dikonfigurasi atau kustom, lalu pilih Ekspor ke CloudWatch.

   CloudWatchJendela Ekspor ke muncul.

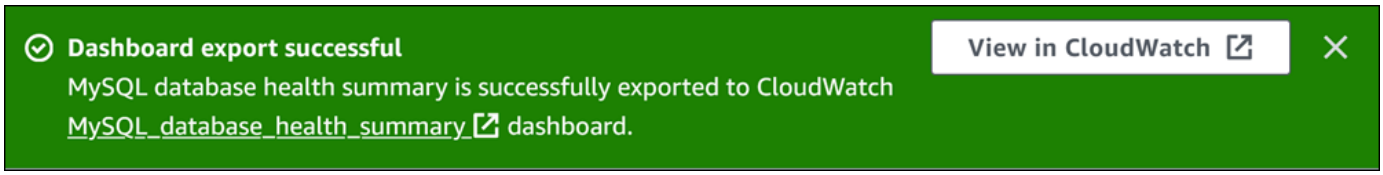


6. Pilih Ekspor sebagai dasbor baru.



7. Masukkan nama dasbor baru di kolom Nama dasbor dan pilih Konfirmasi.

   Banner menampilkan pesan setelah ekspor dasbor berhasil.



8. Pilih tautan atau Lihat CloudWatch di spanduk untuk melihat dasbor metrik di CloudWatch konsol.

Menambahkan metrik Performance Insights ke dasbor yang ada CloudWatch

Tambahkan dasbor metrik yang telah dikonfigurasi sebelumnya atau kustom ke dasbor yang ada CloudWatch . Anda dapat menambahkan label ke dasbor metrik untuk muncul di bagian terpisah di CloudWatch dasbor.

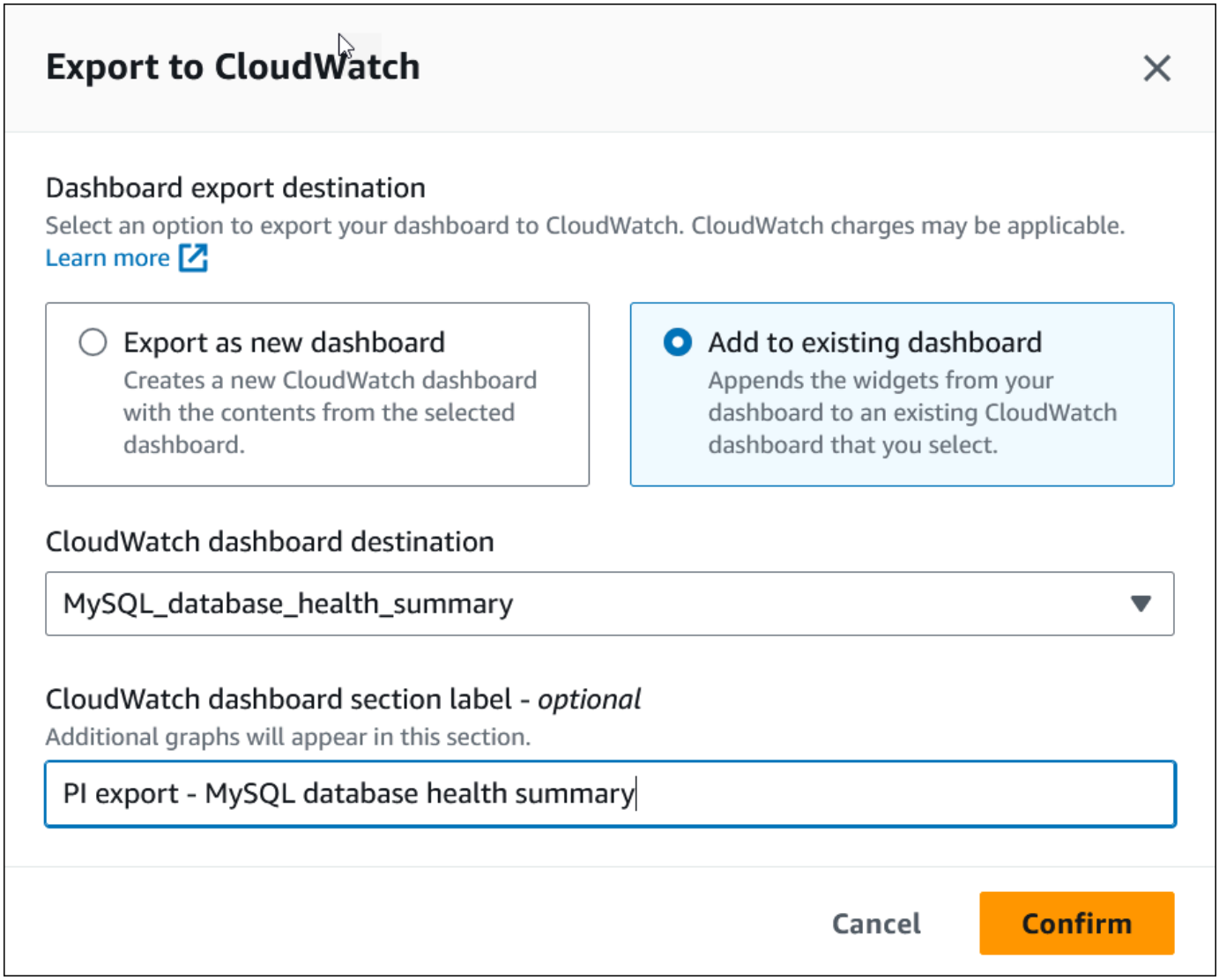
Untuk mengekspor metrik ke dasbor yang ada CloudWatch

1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi kiri, pilih Wawasan Performa.
3. Pilih instans DB.

   Dasbor Wawasan Performa ditampilkan untuk instans DB.
4. Gulir ke bawah dan pilih Metrik.

   Secara default, dasbor yang telah dikonfigurasi sebelumnya dengan metrik Wawasan Performa akan muncul.
5. Pilih dasbor yang telah dikonfigurasi atau kustom, lalu pilih Ekspor ke CloudWatch.

   CloudWatchJendela Ekspor ke muncul.
6. Pilih Tambahkan ke dasbor yang ada.



7. Tentukan tujuan dan label dasbor, lalu pilih Konfirmasi.

   - CloudWatch tujuan dasbor - Pilih CloudWatch dasbor yang ada.
   - CloudWatch label bagian dasbor - opsional - Masukkan nama untuk metrik Performance Insights untuk muncul di bagian ini di dasbor. CloudWatch

   Banner menampilkan pesan setelah ekspor dasbor berhasil.

8. Pilih tautan atau Lihat CloudWatch di spanduk untuk melihat dasbor metrik di CloudWatch konsol.

Melihat widget metrik Performance Insights di CloudWatch

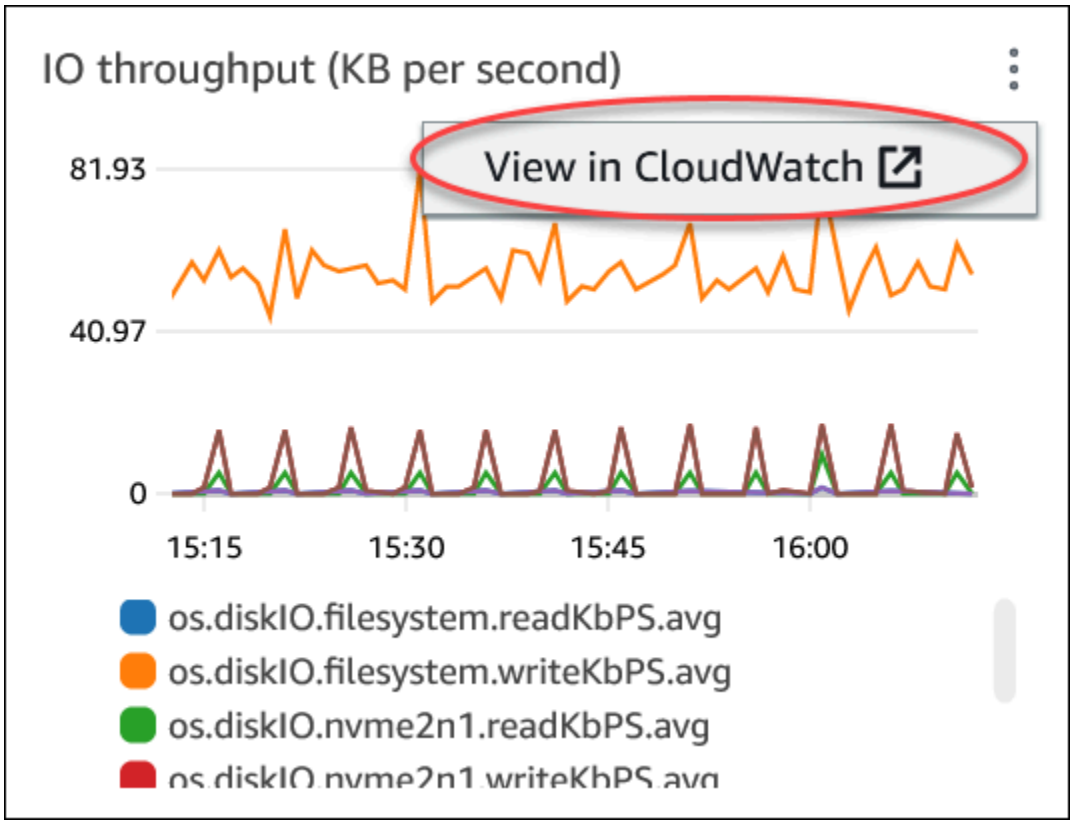
Pilih widget metrik Performance Insights di dasbor Amazon RDS Performance Insights dan lihat data metrik di konsol. CloudWatch

Untuk mengekspor widget metrik dan melihat data metrik di konsol CloudWatch

1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi kiri, pilih Wawasan Performa.
3. Pilih instans DB.

   Dasbor Wawasan Performa ditampilkan untuk instans DB.
4. Gulir ke bawah ke Metrik.

   Secara default, dasbor yang telah dikonfigurasi sebelumnya dengan metrik Wawasan Performa akan muncul.
5. Pilih widget metrik dan kemudian pilih Lihat CloudWatch di dalam menu.



   Data metrik muncul di CloudWatch konsol.

Membuat alarm CloudWatch untuk memantau Amazon RDS

Anda dapat membuat alarm CloudWatch yang mengirimkan pesan Amazon SNS ketika status alarm berubah. Alarm mengawasi metrik tunggal selama periode waktu yang Anda tentukan. Alarm tersebut juga dapat melakukan satu atau beberapa tindakan berdasarkan nilai metrik yang relatif terhadap ambang batas tertentu selama beberapa periode waktu. Tindakan tersebut adalah pemberitahuan yang dikirim ke topik Amazon SNS atau kebijakan Amazon EC2 Auto Scaling.

Alarm hanya menginvokasi tindakan untuk status dengan perubahan berkelanjutan. Alarm CloudWatch tidak menginvokasi tindakan karena alarm tersebut berada dalam status tertentu. Status harus diubah dan dipertahankan selama jangka waktu tertentu.

Anda dapat menggunakan fungsi matematika metrik DB_PERF_INSIGHTS di konsol CloudWatch guna melakukan kueri Amazon RDS untuk metrik penghitung Wawasan Kinerja. Fungsi DB_PERF_INSIGHTS juga menyertakan metrik DBLoad pada interval sub-menit. Anda dapat mengatur alarm CloudWatch berdasarkan metrik ini.

Untuk detail selengkapnya tentang cara membuat alarm, lihat Membuat alarm di metrik penghitung Wawasan Kinerja dari basis data AWS.

Cara mengatur alarm menggunakan AWS CLI

- Panggil `put-metric-alarm`. Untuk informasi selengkapnya, lihat Referensi Perintah AWS CLI.

Cara mengatur alarm menggunakan API CloudWatch

- Panggil `PutMetricAlarm`. Untuk informasi selengkapnya, lihat Referensi API Amazon CloudWatch

Untuk informasi selengkapnya tentang mengatur topik Amazon SNS dan membuat alarm, lihat Menggunakan alarm Amazon CloudWatch.

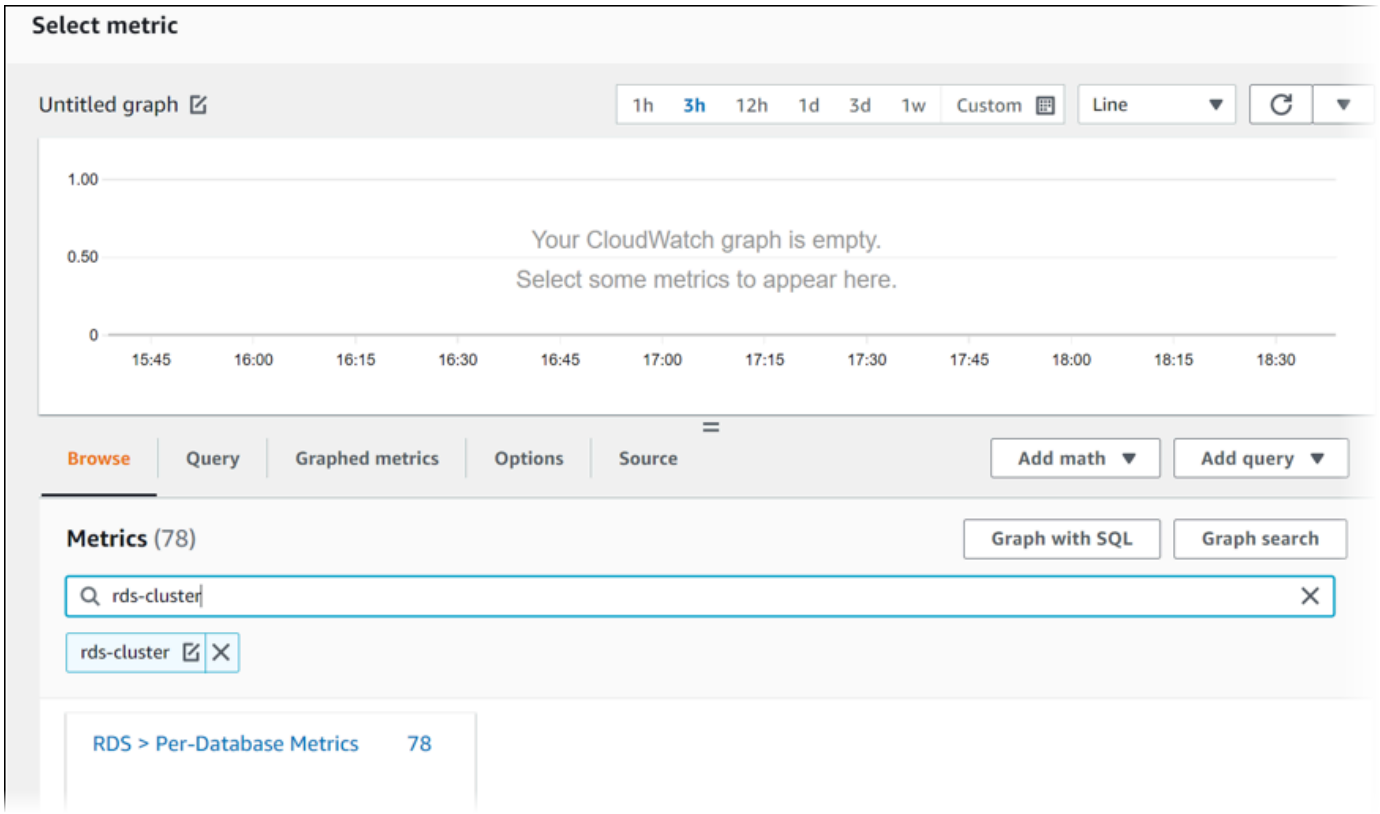
Tutorial: Membuat alarm Amazon CloudWatch untuk kelambatan replika klaster basis data Multi-AZ

Anda dapat membuat alarm Amazon CloudWatch yang mengirimkan pesan ke Amazon SNS ketika kelambatan replika untuk klaster basis data Multi-AZ melampaui ambang batas. Alarm mengawasi metrik `ReplicaLag` selama suatu periode waktu yang Anda tentukan. Tindakannya adalah notifikasi yang dikirim ke topik Amazon SNS atau kebijakan Amazon EC2 Auto Scaling.

Untuk mengatur alarm CloudWatch bagi kelambatan replika klaster basis data Multi-AZ

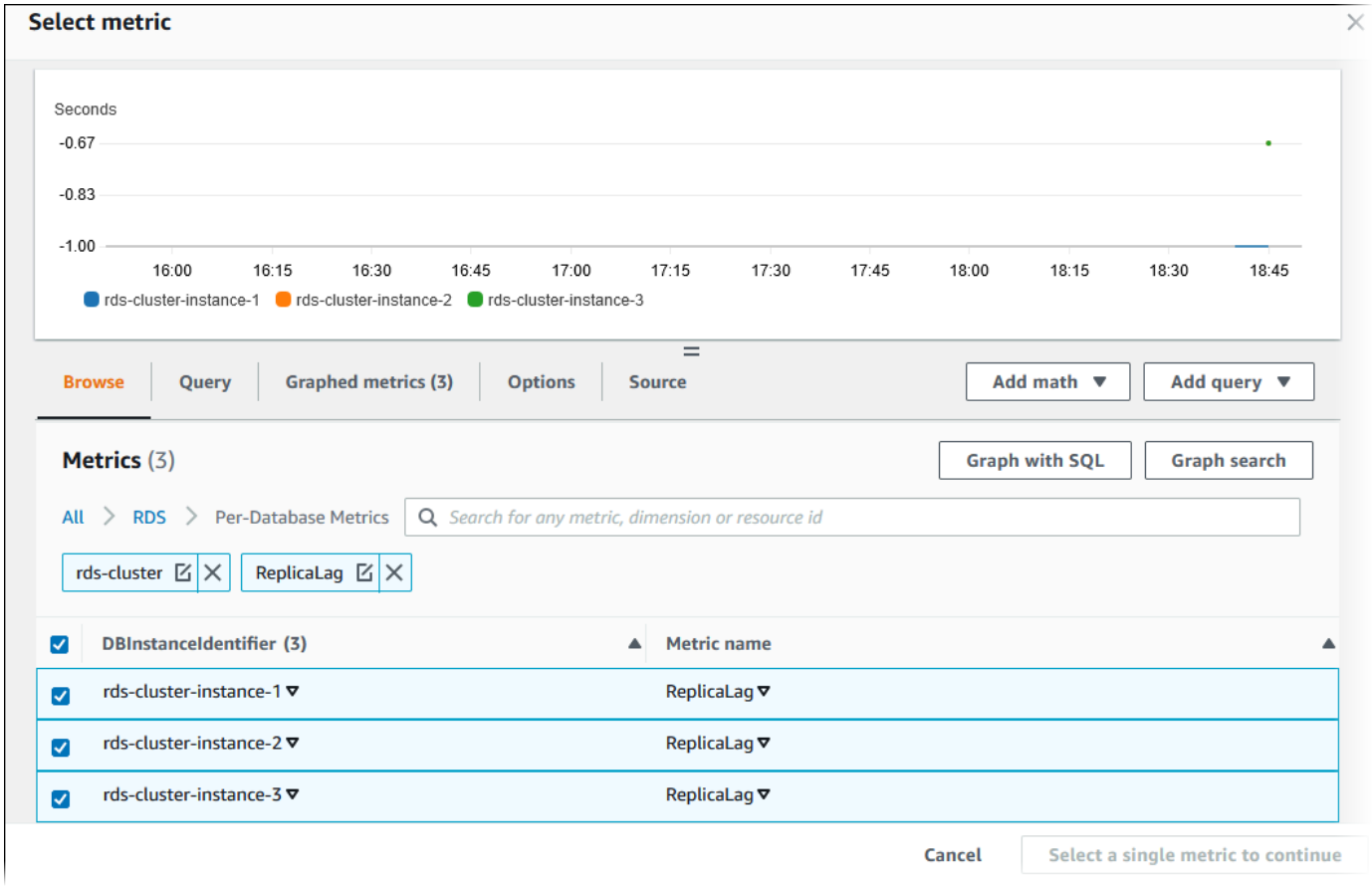
1. Masuk ke AWS Management Console dan buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Alarm, Semua alarm.
3. Pilih Buat alarm.
4. Di halaman Tentukan metrik dan kondisi, pilih Pilih metrik.
5. Di kotak pencarian, masukkan nama klaster Multi-AZ basis data Anda dan tekan Enter.

   Gambar berikut menunjukkan halaman Pilih metrik dengan klaster basis data Multi-AZ bernama `rds-cluster` dimasukkan.



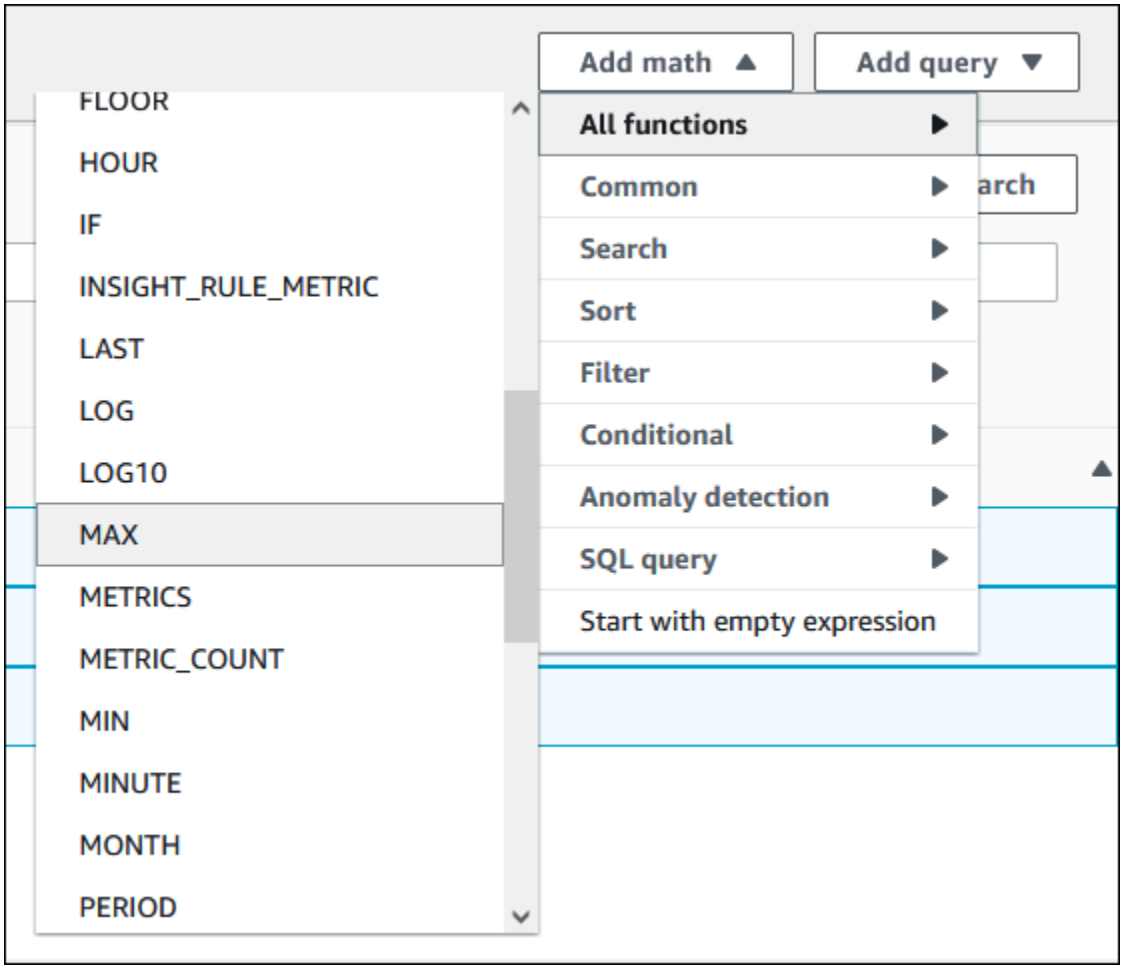
6. Pilih RDS, Metrik Per Basis Data.
7. Di kotak pencarian, masukkan **`ReplicaLag`** dan tekan Enter, lalu pilih setiap instans basis data di klaster basis data.

   Gambar berikut menunjukkan halaman Pilih metrik dengan instans-instans basis data yang dipilih untuk metrik ReplicaLag.



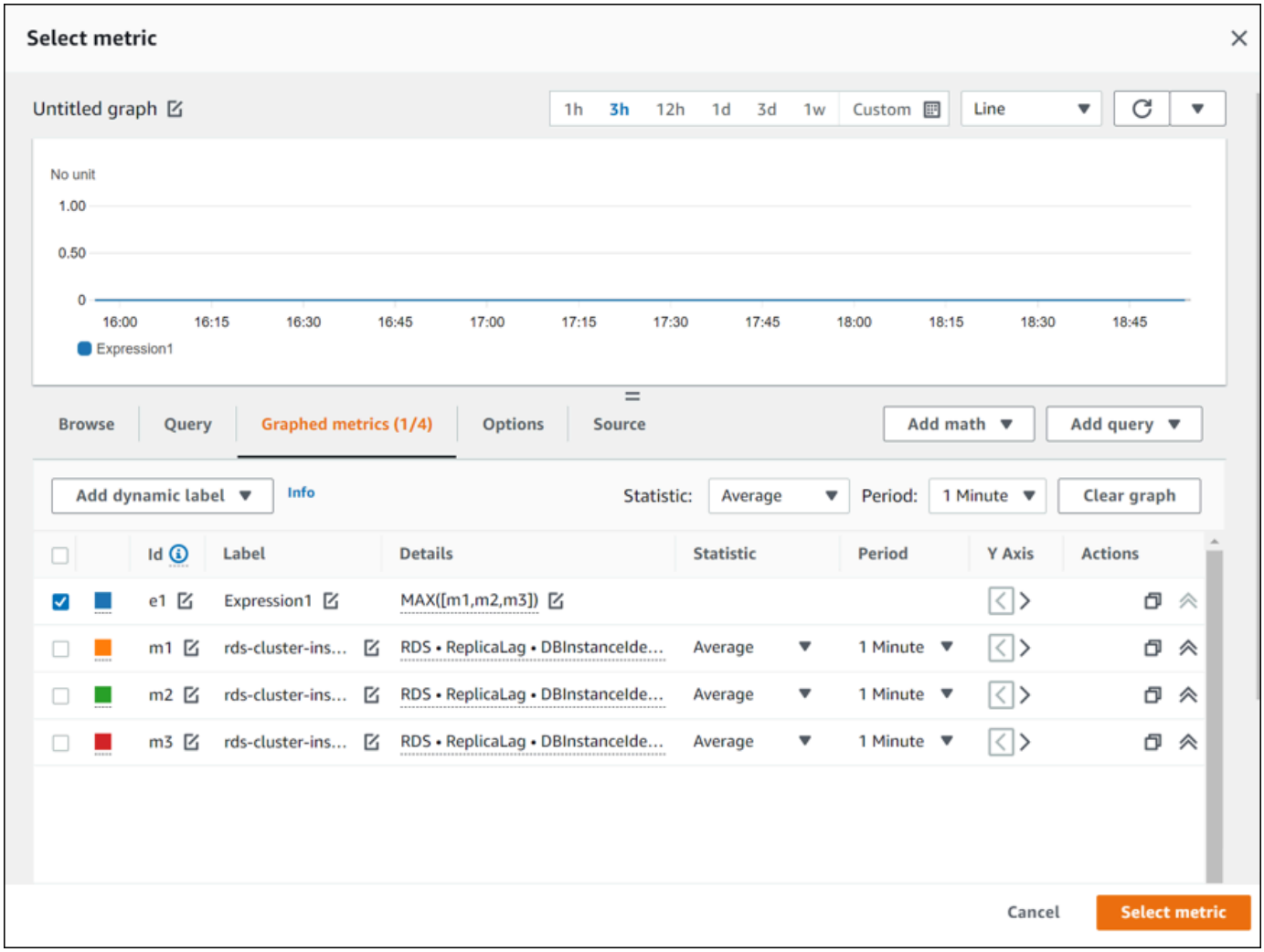
Alarm ini mengawasi kelambatan replika untuk ketiga instans basis data di klaster basis data Multi-AZ. Alarm bertindak ketika ada instans basis data yang melampaui ambang batas. Alarm menggunakan ekspresi matematika yang menghasilkan nilai maksimum dari ketiga metrik. Mulai dengan mengurutkan berdasarkan nama metrik, lalu pilih ketiga metrik ReplicaLag.

8. Dari Tambahkan matematika, pilih Semua fungsi, MAKS.



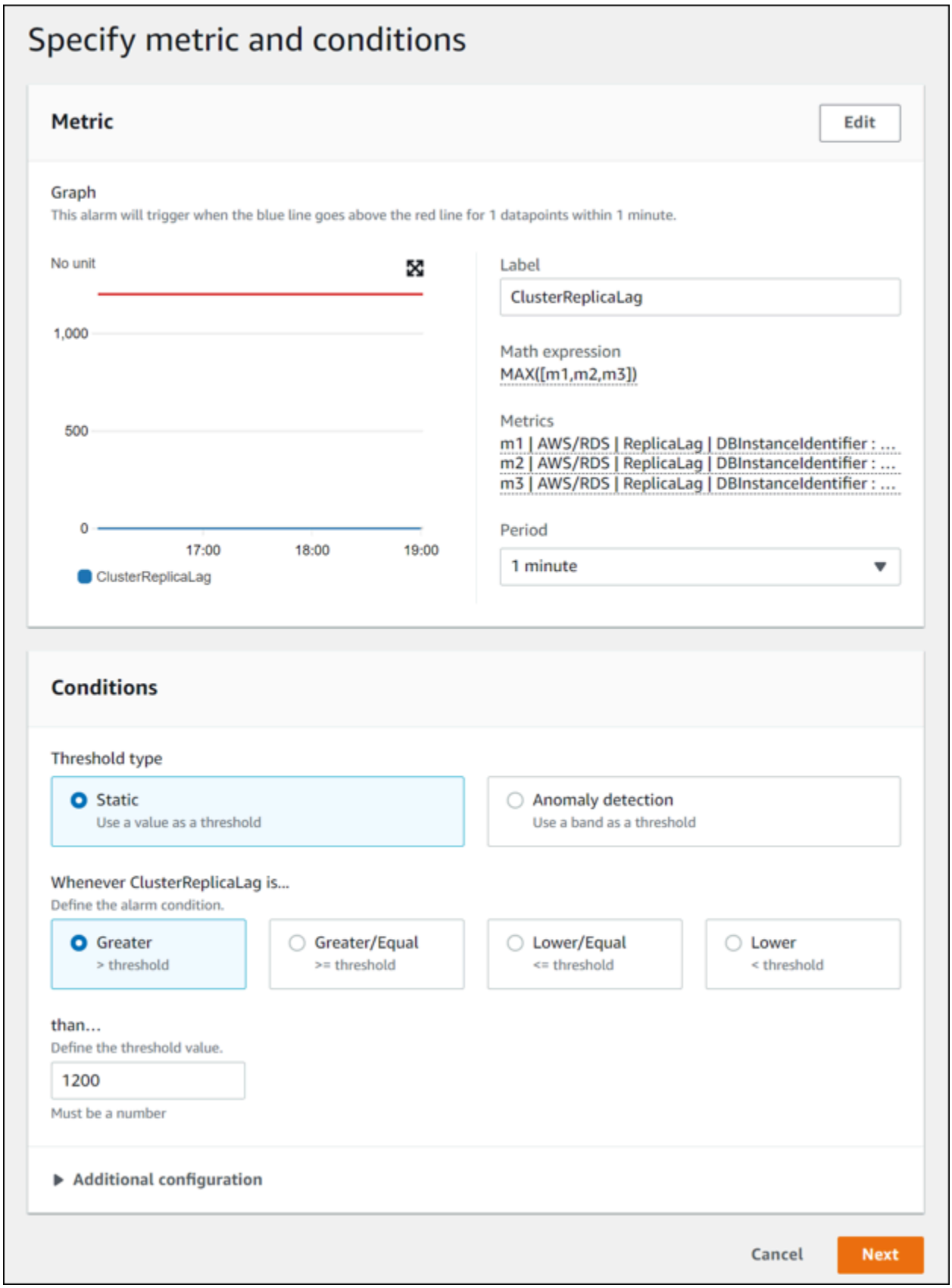
9. Pilih tab Metrik bergrafik, dan edit detail untuk Expression1 menjadi **`MAX([m1,m2,m3])`**.
10. Untuk ketiga metrik ReplicaLag, ubah Periode menjadi 1 menit.
11. Hapus pilihan dari semua metrik kecuali untuk Expression1.

    Halaman Pilih metrik semestinya tampil seperti gambar berikut.



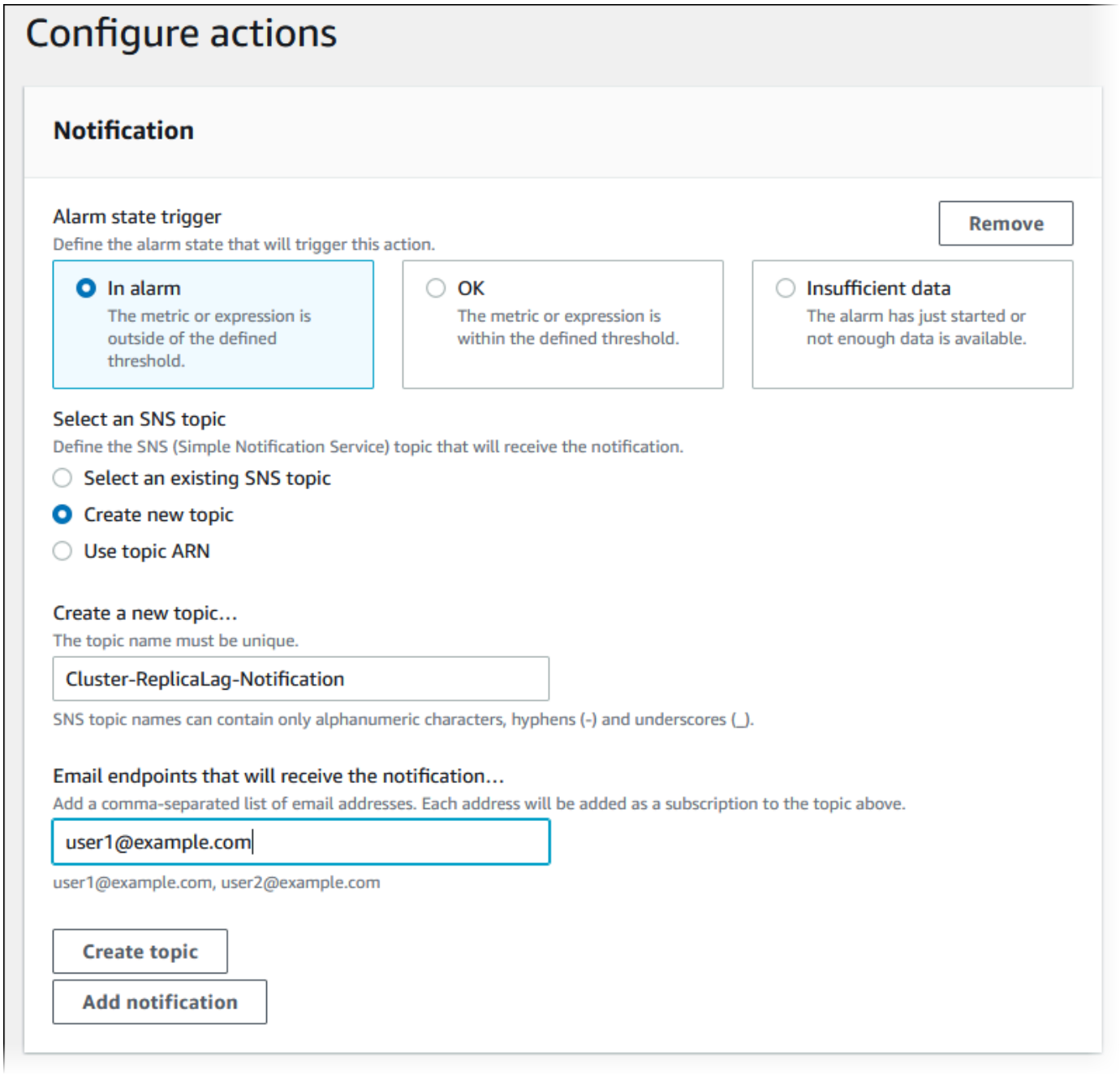
12. Pilih Pilih metrik.
13. Pada halaman Tentukan metrik dan kondisi, ubah label menjadi nama yang bermakna, seperti **`ClusterReplicaLag`**, dan masukkan jumlah detik di Tentukan nilai ambang batas. Untuk tutorial ini, masukkan **`1200`** detik (20 menit). Anda dapat menyesuaikan nilai ini untuk kebutuhan beban kerja Anda.

    Halaman Tentukan metrik dan kondisi semestinya tampil seperti gambar berikut.

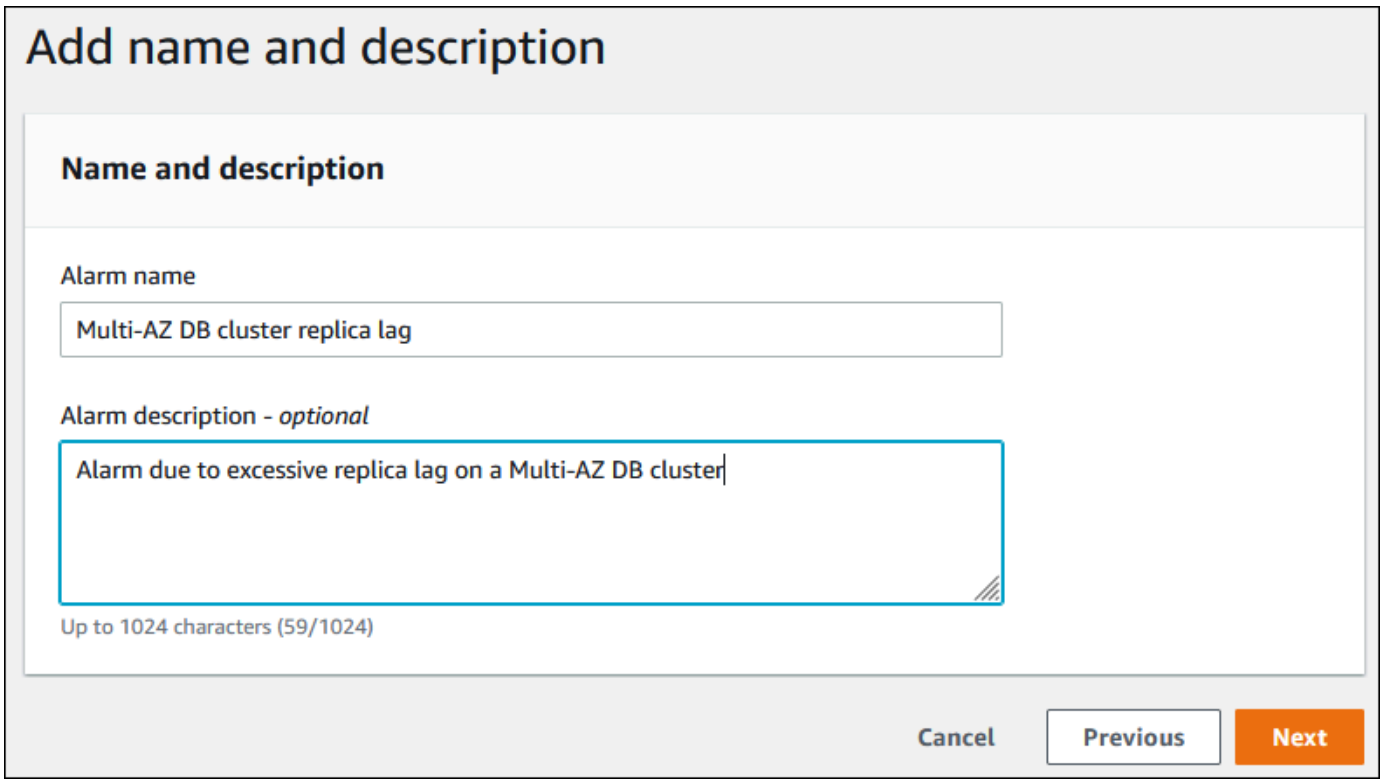


14. Pilih Berikutnya, dan halaman Konfigurasikan tindakan muncul.

15. Tetap jadikan Dalam alarm dipilih, pilih Buat topik baru, lalu masukkan nama topik dan alamat email yang valid.



16. Pilih Buat topik, lalu pilih Berikutnya.
17. Pada halaman Tambahkan nama dan deskripsi, masukkan Nama alarm dan Deskripsi alarm, lalu pilih Berikutnya.



18. Periksa alarm yang akan Anda buat di halaman Pratinjau dan buat, lalu pilih Buat alarm.

Memantau muatan DB dengan Wawasan Performa di Amazon RDS

Wawasan Performa memperluas fitur pemantauan Amazon RDS yang sudah ada untuk mengilustrasikan dan membantu Anda menganalisis performa basis data. Dengan dasbor Wawasan Performa, Anda dapat memvisualisasikan muatan basis data pada muatan instans DB Amazon RDS dan memfilter muatan menurut peristiwa tunggu, pernyataan SQL, host, atau pengguna. Untuk informasi tentang cara menggunakan Wawasan Performa dengan Amazon DocumentDB, lihat Panduan Developer Amazon DocumentDB.

Topik

Ringkasan Wawasan Performa tentang Amazon RDS

Secara default, Wawasan Performa diaktifkan di wizard pembuatan konsol untuk semua mesin Amazon RDS. Jika Anda memiliki lebih dari satu basis data di instans DB, Wawasan Performa akan menggabungkan data performa.

Anda dapat menemukan ringkasan Wawasan Performa untuk Amazon RDS dalam video berikut.

Menggunakan Wawasan Performa untuk Menganalisis Performa Amazon Aurora PostgreSQL

> **Important**
>
> Topik berikut menjelaskan penggunaan Wawasan Performa Amazon RDS dengan mesin DB non-Aurora. Untuk informasi tentang penggunaan Wawasan Performa Amazon RDS dengan

Amazon Aurora, lihat Menggunakan Wawasan Performa Amazon RDS dalam Panduan Pengguna Amazon Aurora.

Topik

- Muatan basis data
- CPU Maksimum
- Dukungan kelas instans, Wilayah, dan mesin DB Amazon RDS untuk Wawasan Performa
- Harga dan retensi data untuk Wawasan Performa

Muatan basis data

Muatan basis data (muatan DB) mengukur tingkat aktivitas sesi dalam basis data Anda. Metrik utama dalam Wawasan Performa adalah `DBLoad`, yang dikumpulkan setiap detik.

Topik

- Sesi aktif
- Sesi aktif rata-rata
- Eksekusi aktif rata-rata
- Dimensi

Sesi aktif

Sesi basis data mewakili dialog aplikasi dengan basis data relasional. Sesi aktif adalah koneksi yang mengirimkan tugas ke mesin DB dan sedang menunggu tanggapan.

Sesi dianggap aktif jika berjalan di CPU atau menunggu sumber daya tersedia sehingga dapat dilanjutkan. Misalnya, sesi aktif mungkin menunggu halaman (atau blok) dibaca ke dalam memori, dan kemudian menggunakan CPU saat membaca data dari halaman.

Sesi aktif rata-rata

Sesi aktif rata-rata (AAS) adalah unit untuk metrik `DBLoad` dalam Wawasan Performa. Ini mengukur berapa banyak sesi yang aktif secara bersamaan di basis data.

Setiap detik, Wawasan Performa mengambil sampel jumlah sesi yang secara bersamaan menjalankan kueri. Untuk setiap sesi aktif, Wawasan Performa mengumpulkan data berikut:

- Pernyataan SQL
- Status sesi (berjalan pada CPU atau menunggu)
- Host
- Pengguna yang menjalankan SQL

Wawasan Performa menghitung AAS dengan membagi jumlah total sesi dengan jumlah sampel selama periode waktu tertentu. Misalnya, tabel berikut menunjukkan 5 sampel berturut-turut dari kueri yang berjalan yang diambil dengan interval 1 detik.

Sampel	Jumlah sesi yang menjalankan kueri	AAS	Penghitungan
1	2	2	Total 2 sesi/1 sampel
2	0	1	Total 2 sesi/2 sampel
3	4	2	Total 6 sesi/3 sampel
4	0	1,5	Total 6 sesi/4 sampel
5	4	2	Total 10 sesi/5 sampel

Pada contoh sebelumnya, muatan DB untuk interval waktu tersebut adalah 2 AAS. Pengukuran ini berarti bahwa rata-rata ada 2 sesi aktif pada waktu tertentu selama interval tersebut ketika 5 sampel diambil.

Analogi untuk muatan DB adalah aktivitas pekerja di gudang. Misalkan gudang mempekerjakan 100 pekerja. Jika 1 pesanan masuk, 1 pekerja memenuhi pesanan sedangkan 99 pekerja menganggur. Jika 100 pesanan masuk, 100 pekerja semuanya memenuhi pesanan secara bersamaan. Jika setiap 15 menit seorang manajer menuliskan berapa banyak pekerja yang aktif secara bersamaan, menambahkan angka-angka ini di penghujung hari, dan kemudian membagi totalnya dengan jumlah sampel, manajer menghitung jumlah rata-rata pekerja yang aktif pada waktu tertentu. Jika rata-rata 50 pekerja kemarin dan 75 pekerja hari ini, maka tingkat aktivitas rata-rata di gudang meningkat. Demikian pula, muatan DB meningkat seiring dengan meningkatnya aktivitas sesi basis data.

Eksekusi aktif rata-rata

Eksekusi aktif rata-rata (AAE) per detik berkaitan dengan AAS. Untuk menghitung AAE, Wawasan Performa membagi total waktu eksekusi kueri dengan interval waktu. Tabel berikut menunjukkan penghitungan AAE untuk kueri yang sama dalam tabel sebelumnya.

Waktu yang telah berlalu (detik)	Total waktu eksekusi (detik)	AAE	Penghitungan
60	120	2	120 detik eksekusi/60 detik berlalu
120	120	1	120 detik eksekusi/120 detik berlalu
180	380	2,11	380 detik eksekusi/180 detik berlalu
240	380	1,58	380 detik eksekusi/240 detik berlalu
300	600	2	600 detik eksekusi/300 detik berlalu

Dalam kebanyakan kasus, AAS dan AAE untuk sebuah kueri kira-kira sama. Namun, karena input ke penghitungan berupa sumber data yang berbeda, penghitungannya sering sedikit berbeda.

Dimensi

Metrik `db.load` berbeda dengan metrik seri waktu lainnya karena Anda dapat membaginya menjadi beberapa sub-komponen yang disebut dimensi. Anda dapat menganggap dimensi sebagai kategori "potong menurut" untuk berbagai karakteristik metrik `DBLoad`.

Saat Anda mendiagnosis masalah performa, dimensi berikut sering kali paling berguna:

Topik

- Peristiwa tunggu
- SQL Teratas

- Rencana

Untuk daftar lengkap dimensi untuk mesin Amazon RDS, lihat Muatan DB diiris berdasarkan dimensi.

Peristiwa tunggu

Peristiwa tunggu menyebabkan pernyataan SQL menunggu peristiwa tertentu terjadi sebelum dapat terus berjalan. Peristiwa tunggu adalah dimensi penting, atau kategori, untuk muatan DB karena menunjukkan di mana pekerjaan terhambat.

Setiap sesi aktif berjalan di CPU atau menunggu. Misalnya, sesi menggunakan CPU ketika mencari memori untuk buffer, melakukan penghitungan, atau menjalankan kode prosedural. Ketika tidak menggunakan CPU, sesi mungkin menunggu buffer memori menjadi kosong, file data dibaca, atau log untuk ditulis. Semakin banyak waktu untuk sesi menunggu sumber daya, semakin sedikit waktu untuk sesi dijalankan di CPU.

Ketika Anda menyetel basis data, Anda sering mencoba mencari tahu sumber daya yang sedang menunggu sesi. Misalnya, dua atau tiga peristiwa tunggu mungkin menyumbang 90 persen dari muatan DB. Ukuran ini berarti bahwa, rata-rata, sesi aktif menghabiskan sebagian besar waktunya menunggu sejumlah kecil sumber daya. Jika Anda dapat mengetahui penyebab peristiwa tunggu ini, Anda dapat mencoba solusinya.

Pertimbangkan analogi pekerja gudang. Pesanan masuk. Pekerja mungkin terlambat memenuhi pesanan. Misalnya, pekerja lain mungkin sedang mengisi ulang rak, troli mungkin tidak tersedia. Atau sistem yang digunakan untuk memasukkan status pesanan lambat. Semakin lama pekerja menunggu, semakin lama waktu yang dibutuhkan untuk memenuhi pesanan. Menunggu adalah bagian alami dari alur kerja gudang, tetapi jika waktu tunggunya berlebihan, produktivitasnya menurun. Sama halnya, menunggu sesi berulang atau panjang dapat menurunkan performa basis data. Untuk informasi selengkapnya, lihat Menyetel peristiwa tunggu untuk Aurora PostgreSQL dan Menyetel peristiwa tunggu untuk Aurora MySQL di Panduan Pengguna Amazon Aurora.

Peristiwa tunggu bervariasi berdasarkan mesin DB:

- Untuk informasi tentang semua kejadian tunggu MariaDB dan MySQL, lihat Tabel Ringkasan Peristiwa Tunggu dalam dokumentasi MySQL.
- Untuk informasi tentang semua kejadian tunggu PostgreSQL, lihat Tabel Pengumpul Statistik > Peristiwa Tunggu dalam dokumentasi PostgreSQL.
- Untuk informasi tentang semua peristiwa tunggu Oracle, lihat Deskripsi Peristiwa Tunggu dalam dokumentasi Oracle.

- Untuk informasi tentang semua peristiwa tunggu SQL Server, lihat Jenis Peristiwa Tunggu dalam dokumentasi SQL Server.

> **Note**
>
> Untuk Oracle, proses latar belakang terkadang berfungsi tanpa pernyataan SQL terkait. Dalam kasus ini, Wawasan Performa melaporkan jenis proses latar belakang yang digabungkan dengan titik dua dan kelas tunggu yang terkait dengan proses latar belakang tersebut. Jenis proses latar belakang meliputi `LGWR`, `ARC0`, `PMON`, dan sebagainya. Sebagai contoh, ketika pengarsip melakukan I/O, laporan Wawasan Performa untuk I/O mirip dengan `ARC1:System I/O`. Kadang-kadang, jenis proses latar belakang juga hilang, dan Wawasan Performa hanya melaporkan kelas tunggu, misalnya `:System I/O`.

SQL Teratas

Saat kejadian tunggu menunjukkan kemacetan, SQL teratas menunjukkan kueri mana yang paling berkontribusi pada pemuatan DB. Misalnya, saat ini mungkin ada banyak kueri yang berjalan di basis data, tetapi kueri tunggal mungkin menggunakan 99 persen dari muatan DB. Dalam hal ini, muatan tinggi mungkin menunjukkan masalah dalam kueri.

Secara default, konsol Wawasan Performa menampilkan kueri SQL teratas yang berkontribusi pada muatan basis data. Konsol juga menunjukkan statistik yang relevan untuk setiap pernyataan. Untuk mendiagnosis masalah performa untuk pernyataan tertentu, Anda dapat memeriksa rencana pelaksanaannya.

Rencana

Rencana eksekusi, juga cukup disebut rencana, adalah urutan langkah-langkah yang mengakses data. Misalnya, rencana untuk menggabungkan tabel `t1` dan `t2` mungkin mengulang semua baris di `t1` dan membandingkan setiap baris dengan baris di `t2`. Dalam basis data relasional, pengoptimal adalah kode bawaan yang menentukan rencana paling efisien untuk kueri SQL.

Untuk instans DB Oracle, Wawasan Performa mengumpulkan rencana eksekusi secara otomatis. Untuk mendiagnosis masalah performa SQL, periksa rencana yang diambil untuk kueri Oracle SQL sumber daya tinggi. Rencana menunjukkan bagaimana Oracle Database telah mengurai dan menjalankan kueri.

Untuk mempelajari cara menganalisis muatan DB menggunakan rencana, lihat Menganalisis rencana eksekusi Oracle menggunakan dasbor Wawasan Performa.

Penangkapan rencana

Setiap lima menit, Wawasan Performa mengidentifikasi kueri Oracle yang paling intensif sumber daya dan menangkap rencananya. Dengan demikian, Anda tidak perlu mengumpulkan dan mengelola rencana dalam jumlah besar secara manual. Sebagai alternatif, Anda dapat menggunakan tab SQL Teratas untuk berfokus pada rencana untuk kueri yang paling bermasalah.

> **Note**
>
> Wawasan Performa tidak menangkap rencana untuk kueri yang teksnya melebihi batas teks kueri maksimum yang dapat dikumpulkan. Untuk informasi selengkapnya, lihat Mengakses lebih banyak teks SQL di dasbor Wawasan Performa.

Periode retensi untuk rencana eksekusi sama dengan data Wawasan Performa Anda. Pengaturan retensi di tingkat gratis adalah Default (7 hari). Untuk mempertahankan data kinerja Anda lebih lama, tetapkan 1–24 bulan. Untuk informasi selengkapnya tentang periode retensi, lihat Harga dan retensi data untuk Wawasan Performa.

Kueri digest

Tab SQL Teratas menunjukkan kueri digest secara default. Kueri digest sendiri tidak memiliki rencana, tetapi semua kueri yang menggunakan nilai literal memiliki rencana. Misalnya, kueri digest mungkin menyertakan teks `WHERE \`email\`=?`. Digest mungkin berisi dua kueri, satu dengan teks `WHERE email=user1@example.com` dan satu lagi dengan `WHERE email=user2@example.com`. Masing-masing kueri literal ini mungkin mencakup beberapa rencana.

Jika Anda memilih kueri digest, konsol akan menampilkan semua rencana untuk pernyataan turunan dari digest yang dipilih. Dengan demikian, Anda tidak perlu melihat semua pernyataan turunan untuk menemukan rencana. Anda mungkin melihat rencana yang tidak ada dalam daftar 10 pernyataan turunan teratas yang ditampilkan. Konsol menampilkan rencana untuk semua kueri turunan yang rencananya telah dikumpulkan, terlepas dari apakah kueri tercantum dalam daftar 10 teratas.

CPU Maksimum

Di dasbor, bagan Basis data muatan mengumpulkan, menggabungkan, dan menampilkan informasi sesi. Untuk mengetahui apakah sesi aktif melebihi CPU maksimum, lihat hubungannya dengan baris vCPU Maks. Nilai vCPU Maks ditentukan oleh jumlah inti vCPU (CPU virtual) untuk instans DB Anda.

Satu proses dapat berjalan pada vCPU pada satu waktu. Jika jumlah proses melebihi jumlah vCPU, proses ini akan mulai mengantre. Jika antrean meningkat, performanya akan terpengaruh. Jika muatan DB sering melampaui baris vCPU Maks dan status tunggu utamanya adalah CPU, CPU akan kelebihan muatan. Dalam kasus ini, sebaiknya Anda membatasi koneksi ke instans, menyesuaikan kueri SQL apa pun dengan muatan CPU yang tinggi, atau mempertimbangkan kelas instans yang lebih besar. Instans yang tinggi dan konsisten dari setiap status tunggu menunjukkan bahwa mungkin terjadi kemacetan atau masalah ketidakcocokan sumber daya yang perlu diselesaikan. Hal ini bisa terjadi meski muatan DB tidak melampaui baris vCPU Maks.

Dukungan kelas instans, Wilayah, dan mesin DB Amazon RDS untuk Wawasan Performa

Tabel berikut berisi mesin DB Amazon RDS yang mendukung Wawasan Performa.

> **Note**
>
> Untuk Amazon Aurora, lihat Dukungan mesin DB Amazon Aurora untuk Wawasan Performa di Panduan Pengguna Amazon Aurora.

Mesin DB Amazon RDS	Versi mesin dan Wilayah yang didukung	Pembatasan kelas instans
Amazon RDS for MariaDB	Untuk informasi selengkapnya tentang versi dan ketersediaan Wilayah Wawasan Performa dengan RDS for MariaDB, lihat Wawasan Performa.	Wawasan Performa tidak didukung untuk kelas instans berikut: • db.t2.micro • db.t2.small • db.t3.micro •

Mesin DB Amazon RDS	Versi mesin dan Wilayah yang didukung	Pembatasan kelas instans
		db.t3.small • db.t4g.micro • db.t4g.small
RDS for MySQL	Untuk informasi selengkapnya tentang versi dan ketersediaan Wilayah Wawasan Performa dengan RDS for MySQL, lihat Wawasan Performa.	Wawasan Performa tidak didukung untuk kelas instans berikut: • db.t2.micro • db.t2.small • db.t3.micro • db.t3.small • db.t4g.micro • db.t4g.small
Amazon RDS for Microsoft SQL Server	Untuk informasi selengkapnya tentang versi dan ketersediaan Wilayah Wawasan Performa dengan RDS for SQL Server, lihat Wawasan Performa.	N/A
Amazon RDS for PostgreSQL	Untuk informasi selengkapnya tentang versi dan ketersediaan Wilayah Wawasan Performa dengan RDS for PostgreSQL, lihat Wawasan Performa.	N/A

Mesin DB Amazon RDS	Versi mesin dan Wilayah yang didukung	Pembatasan kelas instans
Amazon RDS for Oracle	Untuk informasi selengkapnya tentang versi dan ketersediaan Wilayah Wawasan Performa dengan RDS for Oracle, lihat Wawasan Performa.	N/A

Dukungan kelas instans, Wilayah, dan mesin DB Amazon RDS untuk fitur Wawasan Performa

Tabel berikut berisi mesin DB Amazon RDS yang mendukung fitur Wawasan Performa.

Fitur	Tingkat harga	Wilayah yang didukung	Mesin DB yang didukung	Kelas instans yang didukung
Statistik SQL untuk Wawasan Performa	Semua	Semua	Semua	Semua
Menganalisis rencana eksekusi Oracle menggunakan dasbor Wawasan Performa	Semua	Semua	RDS for Oracle	Semua
Menganalisis performa basis data selama periode waktu tertentu	Khusus tingkat berbayar	• AS Timur (Ohio) • AS Timur (Virginia Utara) • AS Barat (California Utara)	RDS for PostgreSQL	Semua

Fitur	Tingkat harga	Wilayah yang didukung	Mesin DB yang didukung	Kelas instans yang didukung
		• AS Barat (Oregon) • Asia Pasifik (Mumbai) • Asia Pasifik (Seoul) • Asia Pasifik (Singapura) • Asia Pasifik (Sydney) • Asia Pasifik (Tokyo) • Kanada (Pusat) • Eropa (Frankfurt) • Eropa (Irlandia) • Eropa (London) • Eropa (Paris) • Eropa (Stockholm)		

Fitur	Tingkat harga	Wilayah yang didukung	Mesin DB yang didukung	Kelas instans yang didukung
Melihat rekomendasi proaktif Performance Insights	Khusus tingkat berbayar	• AS Timur (Ohio) • AS Timur (Virginia Utara) • AS Barat (California Utara) • AS Barat (Oregon) • Asia Pasifik (Mumbai) • Asia Pasifik (Seoul) • Asia Pasifik (Singapura) • Asia Pasifik (Sydney) • Asia Pasifik (Tokyo) • Kanada (Pusat) • Eropa (Frankfurt) • Eropa (Irlandia) • Eropa (London) • Eropa (Paris) • Eropa (Stockholm)	Semua	Semua

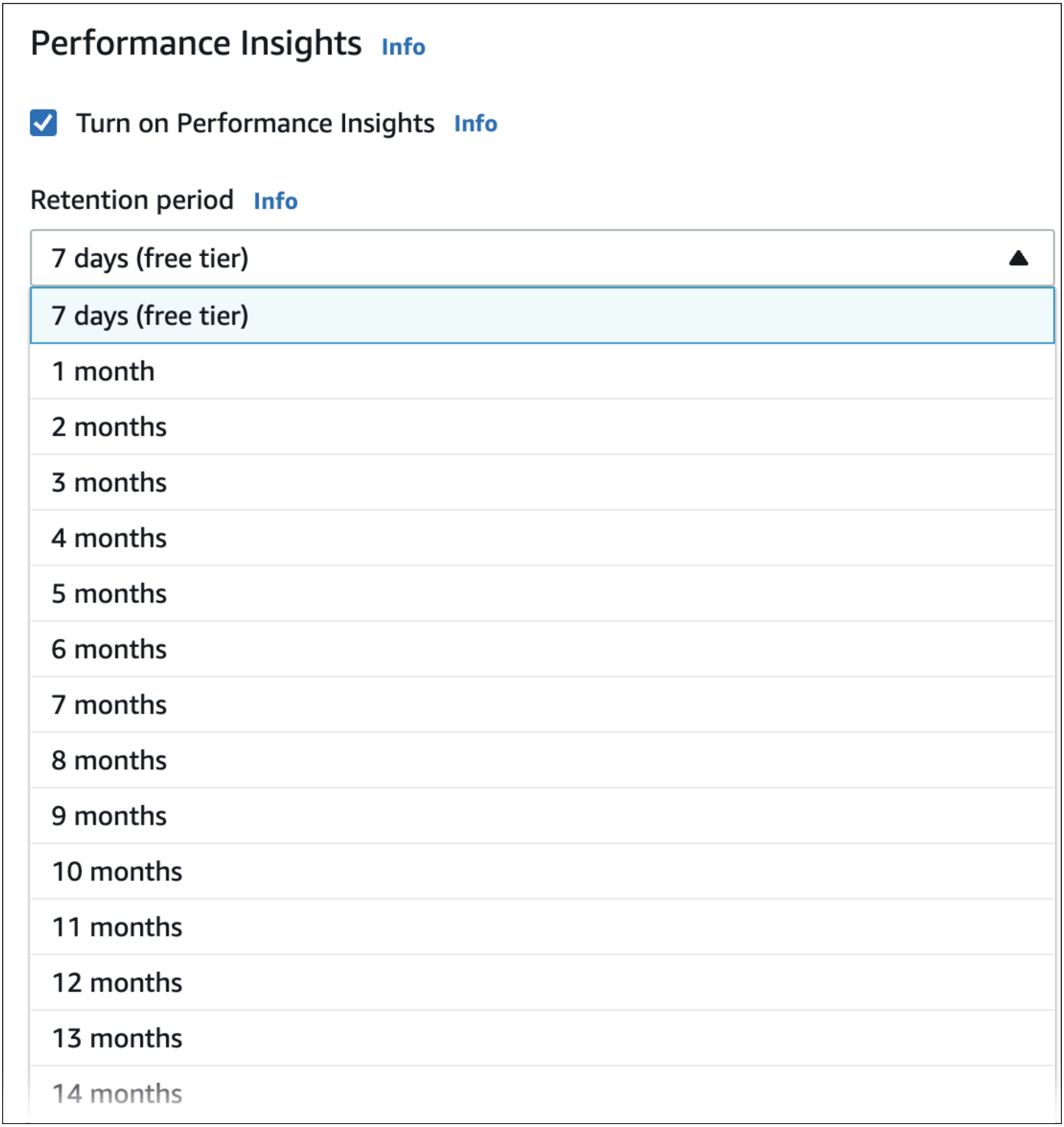
Fitur	Tingkat harga	Wilayah yang didukung	Mesin DB yang didukung	Kelas instans yang didukung
		• Amerika Selatan (Sao Paulo)		

Harga dan retensi data untuk Wawasan Performa

Secara default, Wawasan Performa menawarkan tingkat gratis yang mencakup riwayat data performa selama 7 hari dan 1 juta permintaan API per bulan. Anda juga dapat membeli periode retensi yang lebih lama. Untuk informasi harga selengkapnya, lihat Harga Wawasan Performa.

Di konsol RDS, Anda dapat memilih salah satu periode retensi berikut untuk data Wawasan Performa:

- Default (7 hari)
- ***n*** bulan, di mana ***n*** adalah angka dari 1–24



Untuk mempelajari cara menetapkan periode retensi menggunakanAWS CLI, lihat AWS CLI.

Mengaktifkan dan menonaktifkan Wawasan Performa

Anda dapat mengaktifkan Wawasan Performa untuk instans DB atau klaster DB multi-AZ saat Anda membuatnya. Jika diperlukan, Anda dapat menonaktifkannya nanti. Mengaktifkan dan menonaktifkan Wawasan Performa tidak menyebabkan waktu henti, reboot, atau failover.

> **Note**
>
> Skema Performa adalah alat performa opsional yang digunakan oleh Amazon RDS for MariaDB atau MySQL. Jika Anda mengaktifkan atau menonaktifkan Skema Performa, Anda perlu me-reboot. Namun, jika Anda mengaktifkan atau menonaktifkan Wawasan Performa, Anda tidak perlu me-reboot. Untuk informasi selengkapnya, lihat Mengaktifkan Skema Performa untuk Wawasan Performa di Amazon RDS for MariaDB atau MySQL.

Agen Wawasan Performa menggunakan CPU dan memori terbatas di host DB. Ketika muatan DB tinggi, agen membatasi dampak performa dengan mengurangi frekuensi pengumpulan data.

Konsol

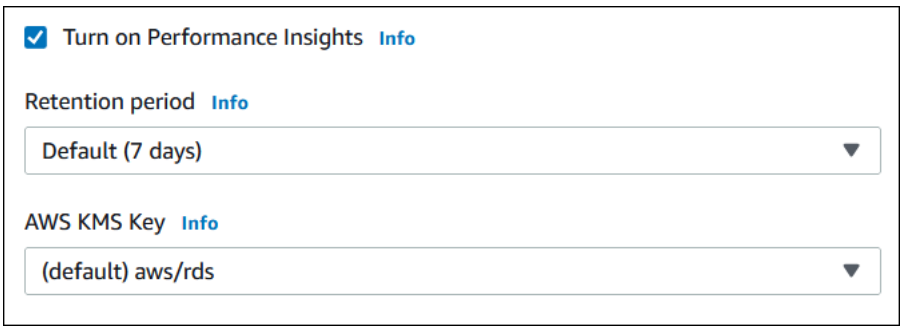
Di konsol, Anda dapat mengaktifkan atau menonaktifkan Wawasan Performa saat membuat atau mengubah instans DB atau klaster DB Multi-AZ.

Mengaktifkan atau menonaktifkan Wawasan Performa saat membuat instans DB atau klaster DB Multi-AZ

Saat Anda membuat instans DB atau klaster DB Multi-AZ baru, aktifkan Wawasan Performa dengan memilih Aktifkan Wawasan Performa di bagian Wawasan Performa. Atau pilih Nonaktifkan Wawasan Performa. Untuk informasi selengkapnya, lihat topik berikut:

- Untuk membuat instans DB, ikuti petunjuk untuk mesin DB Anda di Membuat instans DB Amazon RDS.
- Untuk membuat klaster DB Multi-AZ, ikuti petunjuk untuk mesin DB Anda di Membuat klaster DB Multi-AZ.

Tangkapan layar berikut menunjukkan bagian Wawasan Performa.



Jika memilih Aktifkan Wawasan Performa, Anda akan memiliki opsi berikut:

- Retensi – Jumlah waktu untuk mempertahankan data Wawasan Performa. Pengaturan retensi di tingkat gratis adalah Default (7 hari). Untuk mempertahankan data kinerja Anda lebih lama, tetapkan 1–24 bulan. Untuk informasi selengkapnya tentang periode retensi, lihat Harga dan retensi data untuk Wawasan Performa.
- AWS KMS key – Tentukan AWS KMS key Anda. Wawasan Performa mengenkripsi semua data yang berpotensi sensitif menggunakan kunci KMS Anda. Data dienkripsi saat dipindahkan dan saat tidak aktif. Untuk informasi selengkapnya, lihat Mengonfigurasi kebijakan AWS KMS untuk Wawasan Performa.

Mengaktifkan atau menonaktifkan Wawasan Performa saat memodifikasi instans DB atau klaster DB multi-AZ

Di konsol, Anda dapat memodifikasi instans DB atau klaster DB Multi-AZ untuk mengaktifkan atau menonaktifkan Wawasan Performa.

Untuk mengaktifkan atau menonaktifkan Wawasan Performa untuk instans DB atau klaster DB Multi-AZ menggunakan konsol

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Pilih Basis data.
3. Pilih instans DB atau klaster DB multi-AZ, dan pilih Modifikasi.
4. Di bagian Wawasan Performa, pilih Aktifkan Wawasan Performa atau Nonaktifkan Wawasan Performa.

   Jika memilih Aktifkan Wawasan Performa, Anda akan memiliki opsi berikut:

- Retensi – Jumlah waktu untuk mempertahankan data Wawasan Performa. Pengaturan retensi di tingkat gratis adalah Default (7 hari). Untuk mempertahankan data kinerja Anda lebih lama, tetapkan 1–24 bulan. Untuk informasi selengkapnya tentang periode retensi, lihat Harga dan retensi data untuk Wawasan Performa.
- AWS KMS key – Tentukan kunci KMS Anda. Wawasan Performa mengenkripsi semua data yang berpotensi sensitif menggunakan kunci KMS Anda. Data dienkripsi saat dipindahkan dan saat tidak aktif. Untuk informasi selengkapnya, lihat Mengenkripsi sumber daya Amazon RDS.

5. Pilih Lanjutkan.
6. Untuk Penjadwalan Modifikasi, pilih Terapkan langsung. Jika Anda memilih Terapkan selama periode pemeliharaan terjadwal berikutnya, instans Anda akan mengabaikan pengaturan ini dan segera mengaktifkan Wawasan Performa.
7. Pilih Modifikasi instans.

AWS CLI

Saat Anda menggunakan create-db-instanceAWS CLIperintah, aktifkan Performance Insights dengan menentukan. `--enable-performance-insights` Atau nonaktifkan Wawasan Performa dengan menentukan `--no-enable-performance-insights`.

Anda juga dapat menentukan nilai ini menggunakan perintah AWS CLI berikut:

- create-db-instance-read-replika
- modify-db-instance
- restore-db-instance-from-s3
- create-db-cluster(Kluster DB multi-AZ)
- modify-db-cluster(Kluster DB multi-AZ)

Prosedur berikut menjelaskan cara mengaktifkan atau menonaktifkan Wawasan Performa untuk instans DB yang ada menggunakan AWS CLI.

Untuk mengaktifkan atau menonaktifkan Wawasan Performa untuk instans DB menggunakan AWS CLI

- Panggil modify-db-instanceAWS CLIperintah dan berikan nilai-nilai berikut:
  - `--db-instance-identifier` – Nama instans DB.

- `--enable-performance-insights` untuk mengaktifkan atau `--no-enable-performance-insights` untuk menonaktifkan

Contoh berikut mengaktifkan Wawasan Performa untuk `sample-db-instance`.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier sample-db-instance \
    --enable-performance-insights
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier sample-db-instance ^
    --enable-performance-insights
```

Saat mengaktifkan Wawasan Performa di CLI, Anda dapat secara opsional menentukan jumlah hari untuk mempertahankan data Wawasan Performa dengan opsi `--performance-insights-retention-period`. Anda dapat menentukan 7, *`month`* * 31 (di mana *`month`* adalah jumlah dari 1-23), atau `731`. Misalnya, jika Anda ingin mempertahankan data performa selama 3 bulan, tentukan 93, yakni 3 * 31. Nilai default-nya adalah 7 hari. Untuk informasi selengkapnya tentang periode retensi, lihat Harga dan retensi data untuk Wawasan Performa.

Contoh berikut mengaktifkan Wawasan Performa untuk `sample-db-instance` dan menentukan bahwa data Wawasan Performa dipertahankan selama 93 hari (3 bulan).

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier sample-db-instance \
    --enable-performance-insights \
    --performance-insights-retention-period 93
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier sample-db-instance ^
    --enable-performance-insights ^
```

```
    --performance-insights-retention-period 93
```

Jika Anda menentukan periode retensi seperti 94 hari, yang bukan merupakan nilai yang valid, RDS akan mengeluarkan kesalahan.

```
An error occurred (InvalidParameterValue) when calling the CreateDBInstance operation:
Invalid Performance Insights retention period. Valid values are: [7, 31, 62, 93, 124,
 155, 186, 217,
248, 279, 310, 341, 372, 403, 434, 465, 496, 527, 558, 589, 620, 651, 682, 713, 731]
```

API RDS

Saat Anda membuat instans DB baru menggunakan operasi Amazon RDS API operasi CreateDBInstance, aktifkan Wawasan Performa dengan mengatur ke `EnablePerformanceInsights` ke `True`. Untuk menonaktifkan Wawasan Performa, atur `EnablePerformanceInsights` ke `False`.

Anda juga dapat menentukan nilai `EnablePerformanceInsights` menggunakan operasi API berikut:

- ModifyDBInstance
- dibuatB InstanceReadReplica
- DirestoredB S3 InstanceFrom
- CreateDBCluster (klaster DB Multi-AZ)
- ModifyDBCluster (klaster DB Multi-AZ)

Saat mengaktifkan Wawasan Performa, Anda dapat secara opsional menentukan jumlah waktu, dalam hari, untuk mempertahankan data Wawasan Performa dengan parameter `PerformanceInsightsRetentionPeriod`. Anda dapat menentukan 7, *`month`* * 31 (di mana *`month`* adalah jumlah dari 1-23), atau `731`. Misalnya, jika Anda ingin mempertahankan data performa selama 3 bulan, tentukan `93`, yakni 3 * 31. Nilai default-nya adalah 7 hari. Untuk informasi selengkapnya tentang periode retensi, lihat Harga dan retensi data untuk Wawasan Performa.

Mengaktifkan Skema Performa untuk Wawasan Performa di Amazon RDS for MariaDB atau MySQL

Skema Performa adalah fitur opsional untuk memantau performa runtime Amazon RDS for MariaDB atau MySQL dengan tingkat detail rendah. Skema Performa dirancang untuk memiliki dampak

minimal terhadap performa basis data. Wawasan Performa adalah fitur terpisah yang dapat digunakan dengan atau tanpa Skema Performa.

Topik

Ringkasan Skema Performa

Skema Performa memantau peristiwa dalam basis data MariaDB dan MySQL. Peristiwa adalah tindakan server basis data yang memakan waktu dan telah diinstrumentasi sehingga informasi waktu dapat dikumpulkan. Contoh peristiwa antara lain:

- Panggilan fungsi
- Peristiwa tunggu untuk sistem operasi
- Tahapan eksekusi SQL
- Grup pernyataan SQL

Mesin penyimpanan `PERFORMANCE_SCHEMA` adalah mekanisme untuk mengimplementasikan fitur Skema Performa. Mesin ini mengumpulkan data peristiwa menggunakan instrumentasi dalam kode sumber basis data. Mesin menyimpan peristiwa dalam tabel hanya memori di basis data `performance_schema`. Anda dapat mengkueri `performance_schema` sama seperti Anda mengkueri tabel lainnya. Untuk informasi selengkapnya, lihat Skema Performa MySQL di Panduan Referensi MySQL.

Wawasan Performa dan Skema Performa

Wawasan Performa dan Skema Performa adalah fitur terpisah, tetapi terhubung. Perilaku Wawasan Performa untuk Amazon RDS for MariaDB or MySQL bergantung pada apakah Skema Performa diaktifkan, dan jika demikian, apakah Wawasan Performa mengelola Skema Performa secara otomatis. Tabel berikut menjelaskan perilaku tersebut.

Skema Performa diaktifkan	Mode manajemen Wawasan Performa	Perilaku Wawasan Performa
Ya	Otomatis	• Mengumpulkan informasi pemantauan tingkat rendah yang mendetail • Mengumpulkan metrik sesi aktif setiap detik • Menampilkan muatan DB yang dikategorikan berdasarkan peristiwa tunggu mendetail, yang dapat digunakan untuk mengidentifikasi kemacetan
Ya	Manual	• Mengumpulkan peristiwa tunggu dan metrik per SQL • Mengumpulkan metrik sesi aktif setiap lima detik, bukan setiap detik • Melaporkan status pengguna seperti memasukkan dan mengirim, yang tidak membantu Anda mengidentifikasi kemacetan
Tidak	N/A	• Tidak mengumpulkan peristiwa tunggu, metrik per SQL, atau informasi pemantauan tingkat rendah mendetail lainnya • Mengumpulkan metrik sesi aktif setiap lima detik, bukan setiap detik • Melaporkan status pengguna seperti memasukkan dan mengirim, yang tidak membantu Anda mengidentifikasi kemacetan

Manajemen Skema Performa otomatis berdasarkan Wawasan Performa

Ketika Anda membuat instans DB Amazon RDS for MariaDB atau MySQL dengan Wawasan Performa diaktifkan, Skema Performa juga diaktifkan. Dalam kasus ini, Wawasan Performa secara otomatis mengelola parameter Skema Performa Anda. Ini adalah konfigurasi yang disarankan.

> **Note**
>
> Manajemen Skema Performa otomatis tidak didukung untuk kelas instans t4g.medium.

Untuk manajemen Skema Performa otomatis, kondisi berikut harus terjadi:

- Parameter `performance_schema` diatur menjadi ke `0`.
- Sumber diatur ke `system`, yakni opsi default.

Jika Anda mengubah nilai parameter `performance_schema` secara manual, dan kemudian ingin mengubah ke manajemen otomatis di lain waktu, lihat Mengkonfigurasi Skema Performa untuk manajemen otomatis.

> **Important**
>
> Saat Wawasan Performa mengaktifkan Skema Performa, nilai grup parameter tidak akan diubah. Namun, nilainya diubah pada instans DB yang sedang berjalan. Satu-satunya cara untuk melihat nilai yang diubah adalah dengan menjalankan perintah `SHOW GLOBAL VARIABLES`.

Pengaruh reboot pada Skema Performa

Wawasan Performa dan Skema Performa berbeda dalam persyaratannya untuk reboot instans DB:

Skema Performa

Untuk mengaktifkan atau menonaktifkan fitur ini, Anda harus me-reboot instans DB.

Wawasan Performa

Untuk mengaktifkan atau menonaktifkan fitur ini, Anda tidak harus me-reboot instans DB.

Jika Skema Performa saat ini tidak diaktifkan, dan Anda mengaktifkan Wawasan Performa tanpa me-reboot instans DB, Skema Performa tidak akan diaktifkan.

Menentukan apakah Wawasan Performa mengelola Skema Performa

Untuk mengetahui apakah Wawasan Performa saat ini mengelola Skema Performa untuk mesin utama versi 5.6, 5.7, dan 8.0, tinjau tabel berikut ini.

Pengaturan parameter performance_schema	Pengaturan kolom Source	Wawasan Performa mengelola Skema Performa?
`0`	`system`	Ya
`0` atau `1`	`user`	Tidak

Untuk menentukan apakah Wawasan Performa mengelola Skema Performa secara otomatis

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Pilih Grup parameter.
3. Pilih nama grup parameter untuk instans DB Anda.
4. Masukkan **`performance_schema`** ke bilah pencarian.
5. Periksa apakah Sumber adalah default sistem dan Nilai adalah 0. Jika demikian, Wawasan Performa mengelola Skema Performa secara otomatis. Jika tidak, Wawasan Performa tidak mengelola Skema Performa secara otomatis.

Mengkonfigurasi Skema Performa untuk manajemen otomatis

Asumsikan bahwa Wawasan Performa diaktifkan untuk instans DB atau klaster DB multi-AZ, tetapi saat ini tidak mengelola Wawasan Performa. Jika Anda ingin mengizinkan Wawasan Performa mengelola Skema Performa secara otomatis, selesaikan langkah-langkah berikut.

Untuk mengonfigurasi Skema Performa untuk manajemen otomatis

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Pilih Grup parameter.
3. Pilih nama grup parameter untuk instans DB atau klaster DB Multi-AZ.
4. Masukkan **`performance_schema`** ke bilah pencarian.
5. Pilih parameter `performance_schema`.
6. Pilih Edit parameter.
7. Pilih parameter `performance_schema`.
8. Di Nilai, pilih 0.
9. Pilih Atur ulang, Atur ulang parameter.
10. Reboot instans DB atau klaster DB multi-AZ.

> **Important**
>
> Setiap kali Anda mengaktifkan atau menonaktifkan Skema Performa, pastikan untuk me-reboot instans DB atau klaster DB multi-AZ.

Untuk informasi tentang cara mengubah parameter instans, lihat Memodifikasi parameter dalam grup parameter DB. Untuk informasi selengkapnya tentang dasbor, lihat Menganalisis metrik dengan dasbor Wawasan Performa. Untuk informasi selengkapnya tentang skema performa MySQL, lihat Panduan Referensi MySQL 8.0.

Mengonfigurasi kebijakan akses untuk Wawasan Performa

Untuk mengakses Wawasan Performa, pengguna utama harus memiliki izin yang sesuai dari AWS Identity and Access Management (IAM). Anda dapat memberikan akses dengan cara berikut:

- Melampirkan kebijakan `AmazonRDSPerformanceInsightsReadOnly` terkelola ke kumpulan izin atau peran untuk mengakses semua operasi hanya-baca dari API Wawasan Performa.
- Melampirkan kebijakan `AmazonRDSPerformanceInsightsFullAccess` terkelola ke kumpulan izin atau peran untuk mengakses semua operasi API Wawasan Performa.
- Membuat kebijakan IAM khusus dan melampirkannya ke kumpulan izin atau peran.

Jika Anda menentukan kunci yang dikelola pelanggan saat mengaktifkan Wawasan Performa, pastikan pengguna di akun Anda memiliki izin `kms:Decrypt` dan `kms:GenerateDataKey` izin pada kunci KMS.

Melampirkan kebijakan AmazonRDSPerformanceInsightsReadOnly ke pengguna utama IAM

`AmazonRDSPerformanceInsightsReadOnly` adalah kebijakan yang dikelola AWS yang memberikan akses ke semua operasi hanya-baca API Wawasan Performa Amazon RDS.

Jika Anda melampirkan `AmazonRDSPerformanceInsightsReadOnly` ke kumpulan izin atau peran, penerima dapat menggunakan Wawasan Performa beserta fitur konsol lainnya.

Untuk informasi selengkapnya, lihat AWS kebijakan terkelola: AmazonRDS PerformanceInsightsReadOnly.

Melampirkan kebijakan AmazonRDSPerformanceInsightsFullAccess ke pengguna utama IAM

`AmazonRDSPerformanceInsightsFullAccess` adalah kebijakan yang dikelola AWS yang memberikan akses ke semua operasi API Wawasan Performa Amazon RDS.

Jika Anda melampirkan `AmazonRDSPerformanceInsightsFullAccess` ke kumpulan izin atau peran, penerima dapat menggunakan Wawasan Performa beserta fitur konsol lainnya.

Untuk informasi selengkapnya, lihat AWS kebijakan terkelola: AmazonRDS PerformanceInsightsFullAccess.

Membuat kebijakan IAM khusus untuk Wawasan Performa

Bagi pengguna yang tidak memiliki kebijakan `AmazonRDSPerformanceInsightsReadOnly` atau `AmazonRDSPerformanceInsightsFullAccess`, Anda dapat memberikan akses ke Wawasan Performa dengan membuat atau memodifikasi kebijakan IAM yang dikelola pengguna. Jika Anda melampirkan kebijakan ini ke kumpulan izin atau peran IAM, penerima dapat menggunakan Wawasan Performa.

Untuk membuat kebijakan khusus

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.

2. Di panel navigasi, pilih Kebijakan.
3. Pilih Buat kebijakan.
4. Di halaman Buat Kebijakan, pilih tab JSON.
5. Salin dan tempel teks yang disediakan di bagian dokumen kebijakan JSON di Panduan Referensi Kebijakan yang Dikelola AWS untuk kebijakan AmazonRDSPerformanceInsightsReadOnly atau AmazonRDSPerformanceInsightsFullAccess.
6. Pilih Tinjau kebijakan.
7. Berikan nama untuk kebijakan tersebut dan secara opsional deskripsi, lalu pilih Buat kebijakan.

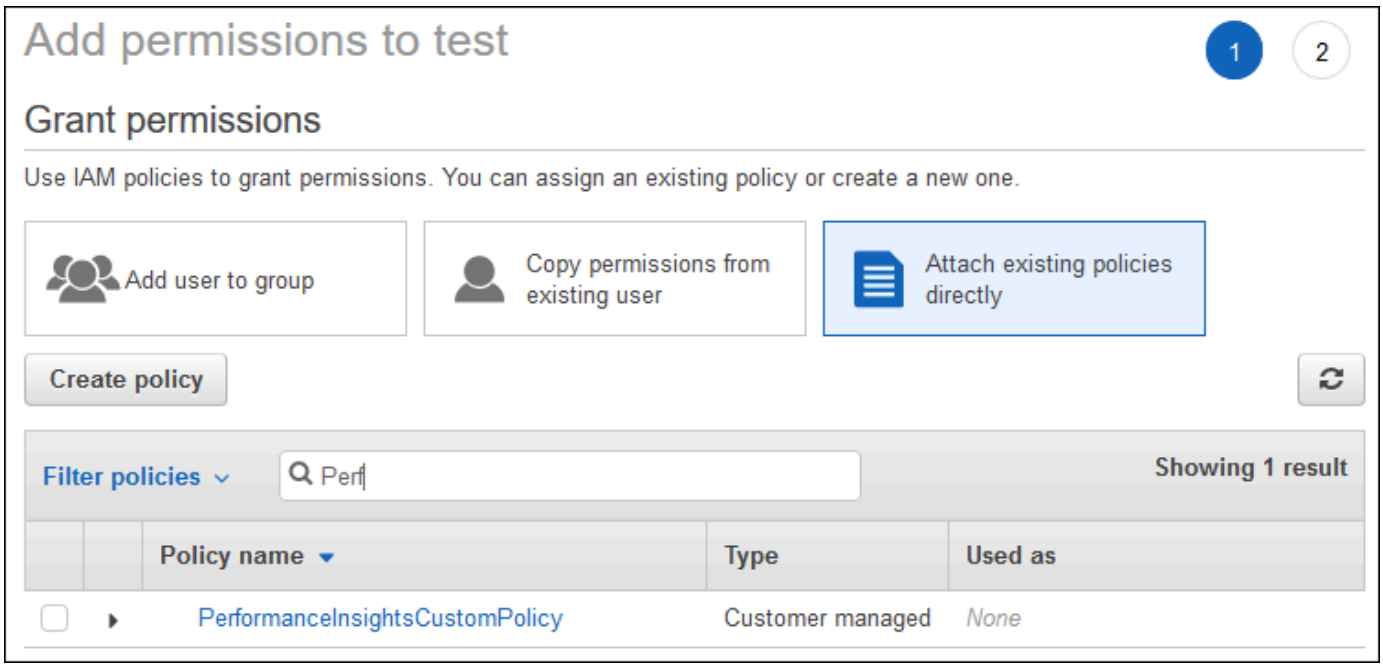
Sekarang, Anda dapat menyisipkan kebijakan ke kumpulan izin atau peran. Prosedur berikut mengasumsikan bahwa Anda sudah memiliki pengguna yang tersedia untuk tujuan ini.

Untuk melampirkan kebijakan ini ke pengguna

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi, pilih Users (Pengguna).
3. Pilih pengguna yang ada dari daftar.

> **Important**
>
> Untuk menggunakan Wawasan Performa, pastikan Anda memiliki akses ke Amazon RDS selain ke kebijakan khusus. Misalnya, kebijakan `AmazonRDSPerformanceInsightsReadOnly` yang ditentukan sebelumnya memberikan akses hanca-baca ke Amazon RDS. Untuk informasi selengkapnya, lihat Mengelola akses menggunakan kebijakan.

4. Di halaman Ringkasan, pilih Tambahkan izin.
5. Pilih Lampirkan kebijakan yang sudah ada secara langsung. Untuk Pencarian, ketik beberapa karakter pertama dari nama kebijakan Anda, seperti yang ditampilkan di bawah ini.



6. Pilih kebijakan Anda, lalu pilih Berikutnya: Tinjauan.
7. Pilih Tambahkan izin.

Mengonfigurasi kebijakan AWS KMS untuk Wawasan Performa

Wawasan Performa menggunakan AWS KMS key untuk mengenkripsi data sensitif. Saat mengaktifkan Wawasan Performa melalui API atau konsol, Anda dapat melakukan salah satu tindakan berikut:

- Memilih Kunci yang dikelola AWS default.

  Amazon RDS Kunci yang dikelola AWS untuk instans DB baru Anda. Amazon RDS membuat Kunci yang dikelola AWS untuk Akun AWS Anda. Akun AWS Anda memiliki Kunci yang dikelola AWS yang berbeda untuk Amazon RDS untuk masing-masing Wilayah AWS.

- Memilih kunci yang dikelola pelanggan.

  Jika Anda menentukan kunci yang dikelola pelanggan, pengguna di akun Anda yang memanggil API Wawasan Performa memerlukan izin `kms:Decrypt` dan `kms:GenerateDataKey` pada kunci KMS. Anda dapat mengonfigurasi izin ini melalui kebijakan IAM. Namun, sebaiknya Anda mengelola izin ini melalui kebijakan kunci KMS Anda. Untuk informasi selengkapnya, lihat Menggunakan kebijakan kunci dalam KMS AWS.

Example

Contoh berikut menunjukkan cara menambahkan pernyataan ke kebijakan kunci KMS Anda. Pernyataan ini mengizinkan akses ke Wawasan Performa. Bergantung pada bagaimana Anda menggunakan kunci KMS, sebaiknya Anda mengubah beberapa pembatasan. Sebelum menambahkan pernyataan ke kebijakan, hapus semua komentar.

```
{
 "Version" : "2012-10-17",
 "Id" : "your-policy",
 "Statement" : [ {
    //This represents a statement that currently exists in your policy.
 }
 ....,
 //Starting here, add new statement to your policy for Performance Insights.
 //We recommend that you add one new statement for every RDS instance
{
    "Sid" : "Allow viewing RDS Performance Insights",
    "Effect": "Allow",
    "Principal": {
        "AWS": [
            //One or more principals allowed to access Performance Insights
            "arn:aws:iam::444455556666:role/Role1"
        ]
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
    ],
    "Resource": "*",
    "Condition" : {
        "StringEquals" : {
            //Restrict access to only RDS APIs (including Performance Insights).
            //Replace region with your AWS Region.
            //For example, specify us-west-2.
            "kms:ViaService" : "rds.region.amazonaws.com"
        },
        "ForAnyValue:StringEquals": {
            //Restrict access to only data encrypted by Performance Insights.
            "kms:EncryptionContext:aws:pi:service": "rds",
            "kms:EncryptionContext:service": "pi",

            //Restrict access to a specific RDS instance.
```

```
            //The value is a DbiResourceId.
           "kms:EncryptionContext:aws:rds:db-id": "db-AAAAABBBBBCCCCDDDDDEEEEE"
        }
    }
}
```

Bagaimana Wawasan Performa menggunakan kunci yang dikelola pelanggan AWS KMS

Wawasan Performa menggunakan kunci yang dikelola pelanggan untuk mengenkripsi data sensitif. Jika mengaktifkan Wawasan Performa, Anda dapat memberikan kunci AWS KMS melalui API. Wawasan Performa membuat izin KMS pada kunci ini. Ini menggunakan kunci dan melakukan operasi yang diperlukan untuk memproses data sensitif. Data sensitif mencakup kolom-kolom seperti pengguna, basis data, aplikasi, dan teks kueri SQL. Wawasan Performa memastikan bahwa data tetap terenkripsi baik saat tidak aktif maupun saat transit.

Cara kerja IAM Wawasan Performa dengan AWS KMS

IAM memberikan izin ke API tertentu. Wawasan Performa memiliki API publik berikut, yang dapat Anda batasi menggunakan kebijakan IAM:

- `DescribeDimensionKeys`
- `GetDimensionKeyDetails`
- `GetResourceMetadata`
- `GetResourceMetrics`
- `ListAvailableResourceDimensions`
- `ListAvailableResourceMetrics`

Anda dapat menggunakan permintaan API berikut untuk mendapatkan data sensitif.

- `DescribeDimensionKeys`
- `GetDimensionKeyDetails`
- `GetResourceMetrics`

Saat Anda menggunakan API untuk mendapatkan data sensitif, Wawasan Performa memanfaatkan kredensial pemanggil. Pemeriksaan ini memastikan bahwa akses ke data sensitif dibatasi pada mereka yang memiliki akses ke kunci KMS.

Saat memanggil API ini, Anda memerlukan izin untuk memanggil API melalui kebijakan IAM dan izin untuk menginvokasi tindakan `kms:decrypt` melalui kebijakan kunci AWS KMS.

API `GetResourceMetrics` dapat menampilkan data sensitif dan non-sensitif. Parameter permintaan menentukan apakah respons harus menyertakan data sensitif. API menampilkan data sensitif ketika permintaan menyertakan dimensi sensitif baik dalam parameter filter atau kelompokkan-menurut.

Untuk informasi selengkapnya tentang dimensi yang dapat digunakan dengan API `GetResourceMetrics`, lihat DimensionGroup.

Example Contoh

Contoh berikut meminta data sensitif untuk grup `db.user`:

```
POST / HTTP/1.1
Host: <Hostname>
Accept-Encoding: identity
X-Amz-Target: PerformanceInsightsv20180227.GetResourceMetrics
Content-Type: application/x-amz-json-1.1
User-Agent: <UserAgentString>
X-Amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
 Signature=<Signature>
Content-Length: <PayloadSizeBytes>
{
  "ServiceType": "RDS",
  "Identifier": "db-ABC1DEFGHIJKL2MNOPQRSTUV3W",
  "MetricQueries": [
    {
      "Metric": "db.load.avg",
      "GroupBy": {
        "Group": "db.user",
        "Limit": 2
      }
    }
  ],
  "StartTime": 1693872000,
  "EndTime": 1694044800,
  "PeriodInSeconds": 86400
}
```

Example

Contoh berikut meminta data non-sensitif untuk metrik `db.load.avg`:

```
POST / HTTP/1.1
Host: <Hostname>
Accept-Encoding: identity
X-Amz-Target: PerformanceInsightsv20180227.GetResourceMetrics
Content-Type: application/x-amz-json-1.1
User-Agent: <UserAgentString>
X-Amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
 Signature=<Signature>
Content-Length: <PayloadSizeBytes>
{
    "ServiceType": "RDS",
    "Identifier": "db-ABC1DEFGHIJKL2MNOPQRSTUV3W",
    "MetricQueries": [
        {
            "Metric": "db.load.avg"
        }
    ],
    "StartTime": 1693872000,
    "EndTime": 1694044800,
    "PeriodInSeconds": 86400
}
```

Menganalisis metrik dengan dasbor Wawasan Performa

Dasbor Wawasan Performa berisi informasi performa basis data untuk membantu Anda menganalisis dan memecahkan masalah performa. Di halaman dasbor utama, Anda dapat melihat informasi tentang muatan basis data. Anda dapat "memotong" muatan DB berdasarkan dimensi seperti peristiwa tunggu atau SQL.

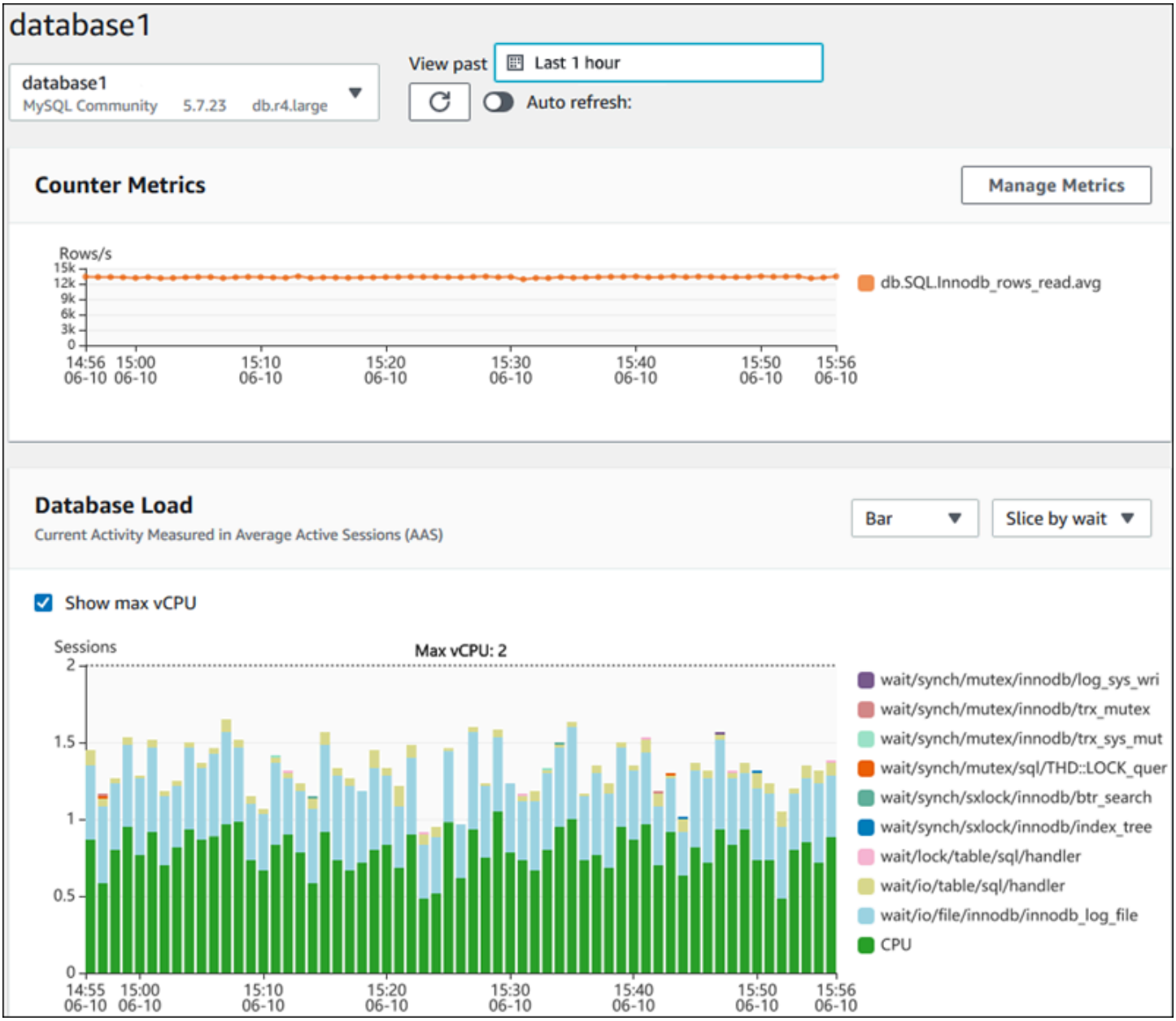
Dasbor Wawasan Performa

- Ringkasan dasbor Wawasan Performa
- Mengakses dasbor Wawasan Performa
- Menganalisis muatan DB menurut peristiwa tunggu
- Menganalisis performa basis data selama periode waktu tertentu

- Menganalisis kueri di dasbor Wawasan Performa
- Menganalisis rencana eksekusi Oracle menggunakan dasbor Wawasan Performa

Ringkasan dasbor Wawasan Performa

Dasbor adalah cara termudah untuk berinteraksi dengan Wawasan Performa. Contoh berikut menunjukkan dasbor untuk instans DB MySQL.



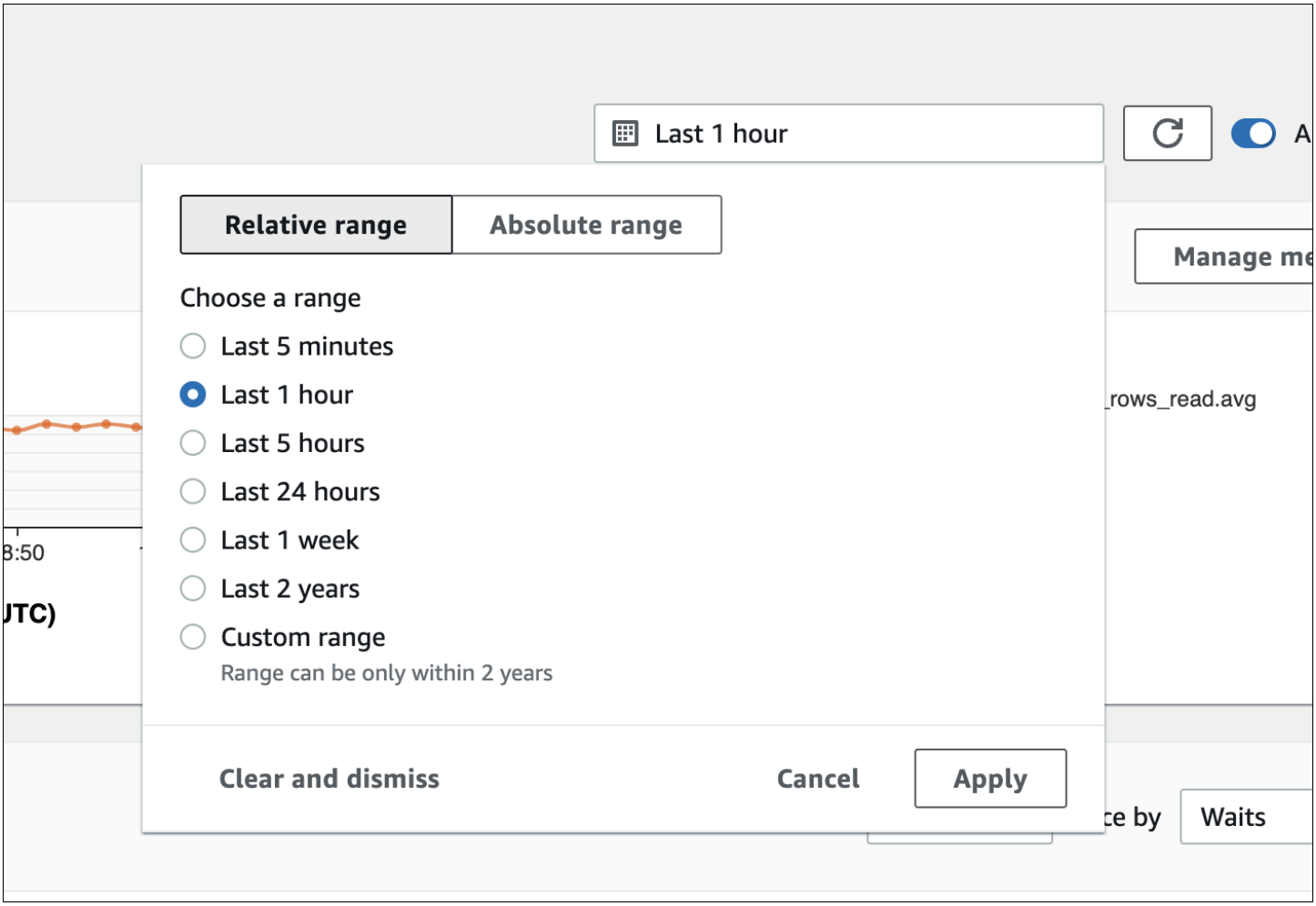
Topik

- Filter rentang waktu

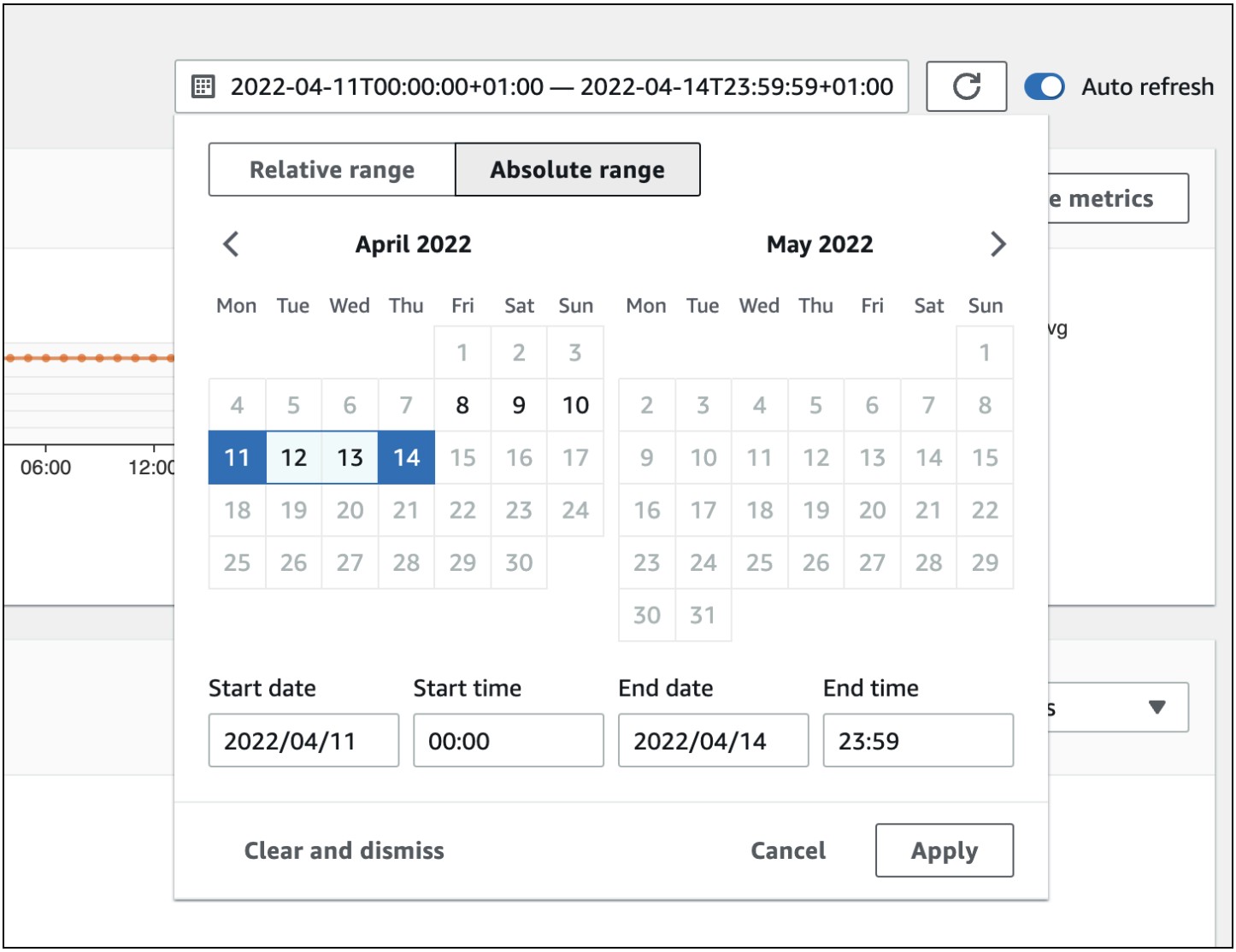
- Bagan metrik penghitung
- Bagan muatan basis data
- Tabel Dimensi teratas

Filter rentang waktu

Secara default, dasbor Wawasan Performa menampilkan muatan DB selama satu jam terakhir. Anda dapat menyesuaikan rentang ini menjadi sesingkat 5 menit atau selama 2 tahun. Anda juga dapat memilih rentang relatif kustom.



Anda dapat memilih rentang absolut dengan tanggal dan waktu awal dan akhir. Contoh berikut menunjukkan rentang waktu yang dimulai tengah malam pada 11/4/22 dan berakhir pukul 23.59 pada 14/4/22.



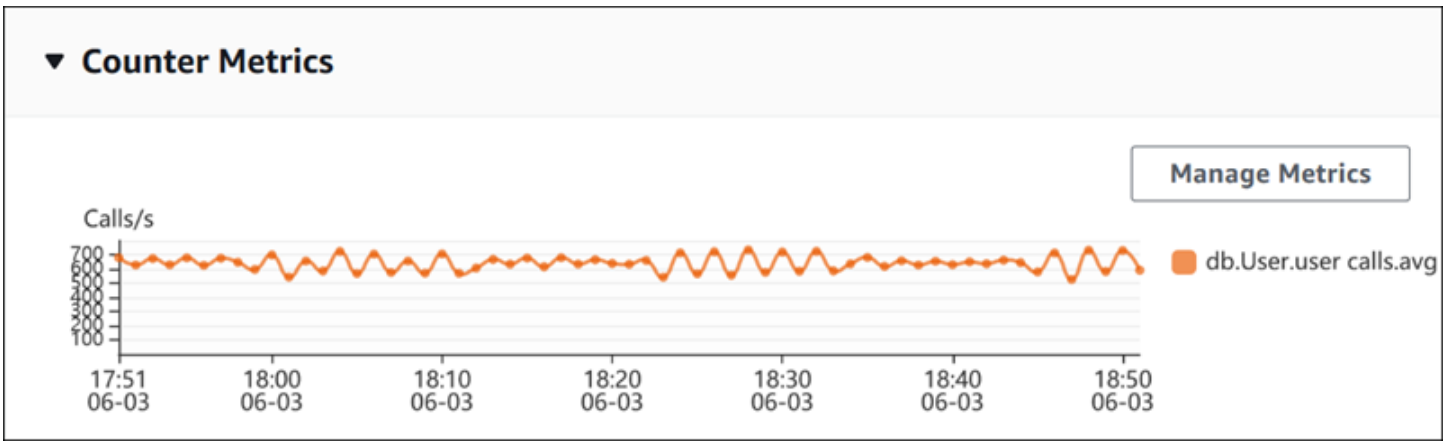
Bagan metrik penghitung

Dengan metrik penghitung, Anda dapat menyesuaikan dasbor Wawasan Performa untuk menyertakan hingga 10 grafik tambahan. Grafik ini menunjukkan pilihan dari sejumlah sistem operasi dan metrik performa basis data. Anda dapat menghubungkan informasi ini dengan muatan DB untuk membantu mengidentifikasi dan menganalisis masalah performa.

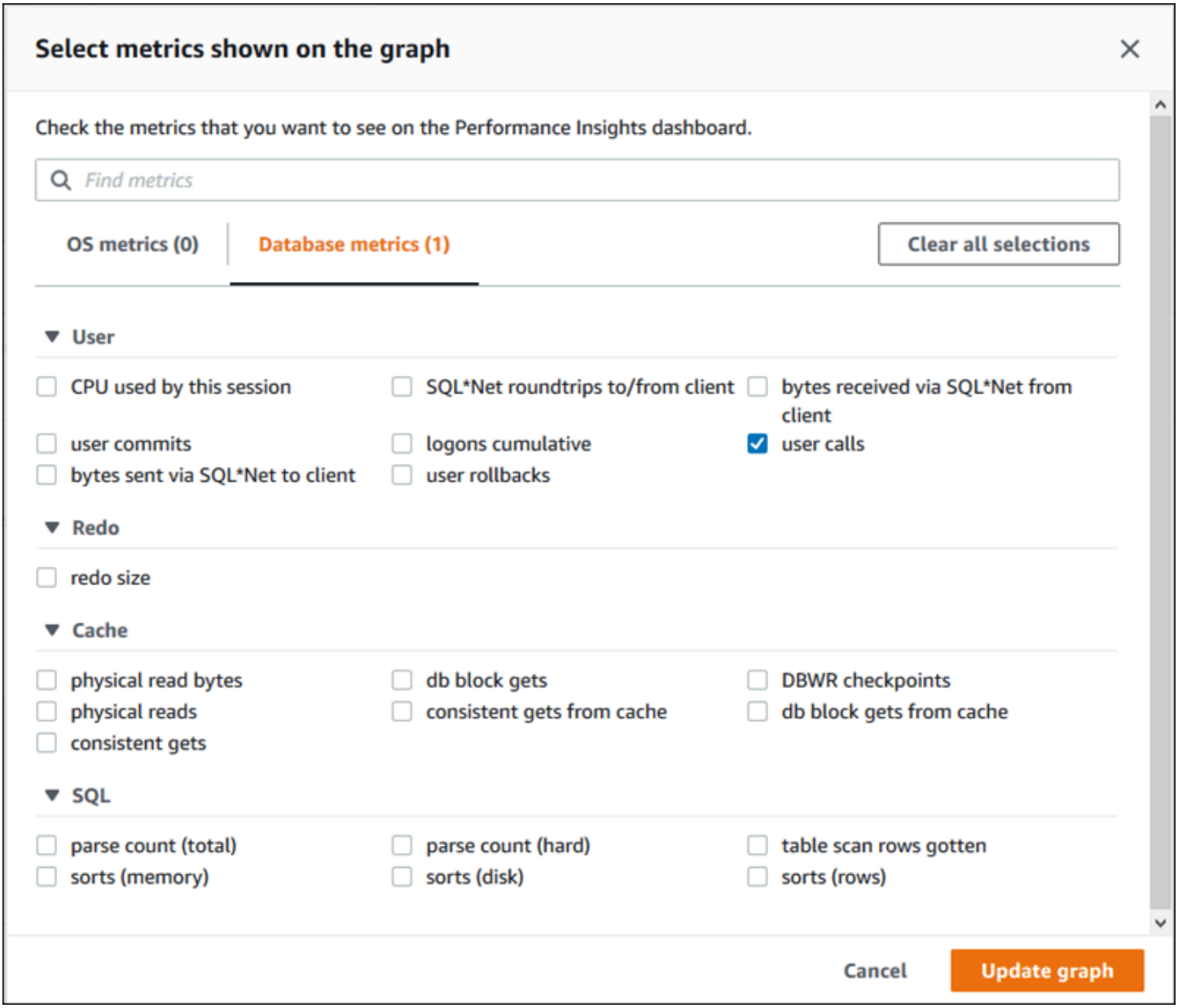
Bagan Metrik Penghitung menampilkan data untuk penghitung performa. Metrik default bergantung pada mesin DB:

- MySQL dan MariaDB – `db.SQL.Innodb_rows_read.avg`
- Oracle – `db.User.user calls.avg`
- Microsoft SQL Server – `db.Databases.Active Transactions(_Total).avg`

- PostgreSQL – `db.Transactions.xact_commit.avg`



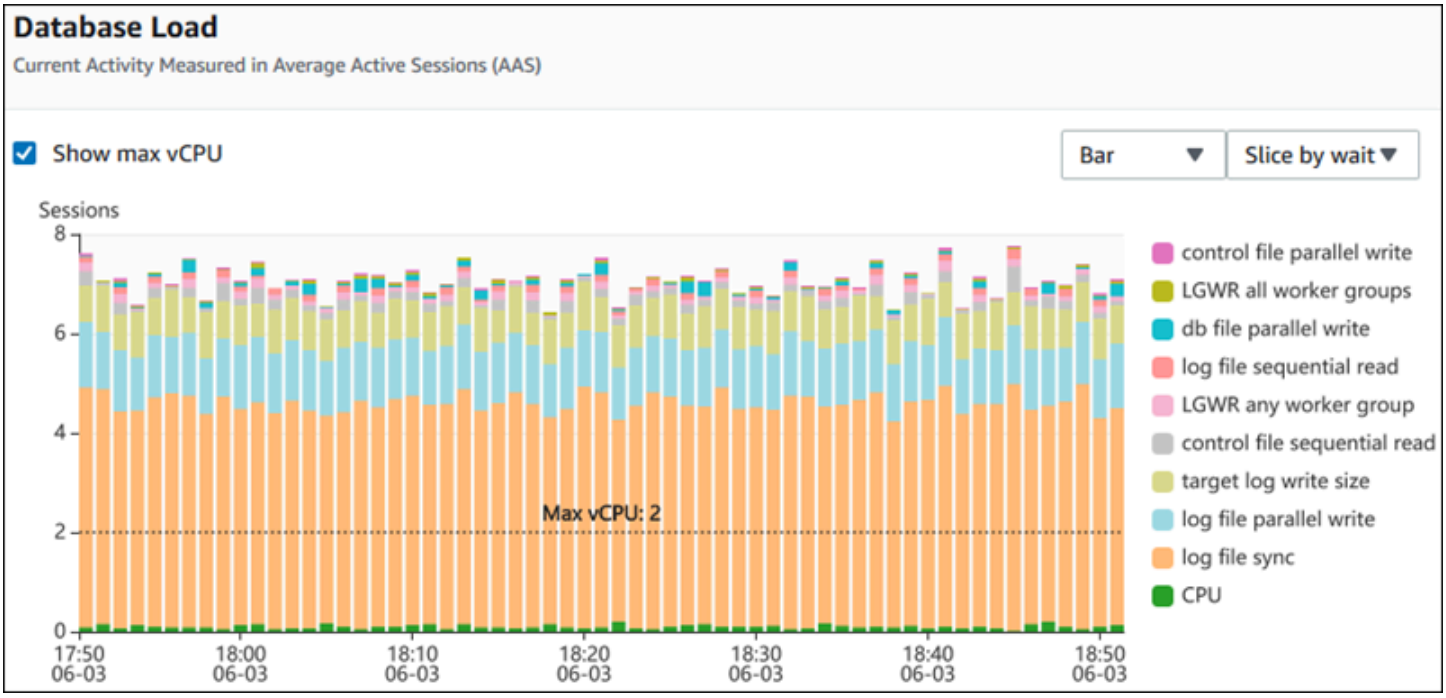
Untuk mengubah penghitung performa, pilih Kelola Metrik. Anda dapat memilih beberapa Metrik OS atau Metrik basis data, seperti yang ditunjukkan di tangkapan layar berikut. Untuk melihat detail setiap metrik, arahkan kursor ke nama metrik.



Untuk deskripsi metrik penghitung yang dapat ditambahkan untuk setiap mesin DB, lihat Metrik penghitung Wawasan Performa.

Bagan muatan basis data

Bagan Muatan basis data menunjukkan perbandingan aktivitas basis data dengan kapasitas instans DB seperti yang ditunjukkan oleh baris vCPU Maks. Secara default, bagan garis bertumpuk mewakili muatan DB sebagai sesi aktif rata-rata per unit waktu. Muatan DB diiris (dikelompokkan) berdasarkan status tunggu.

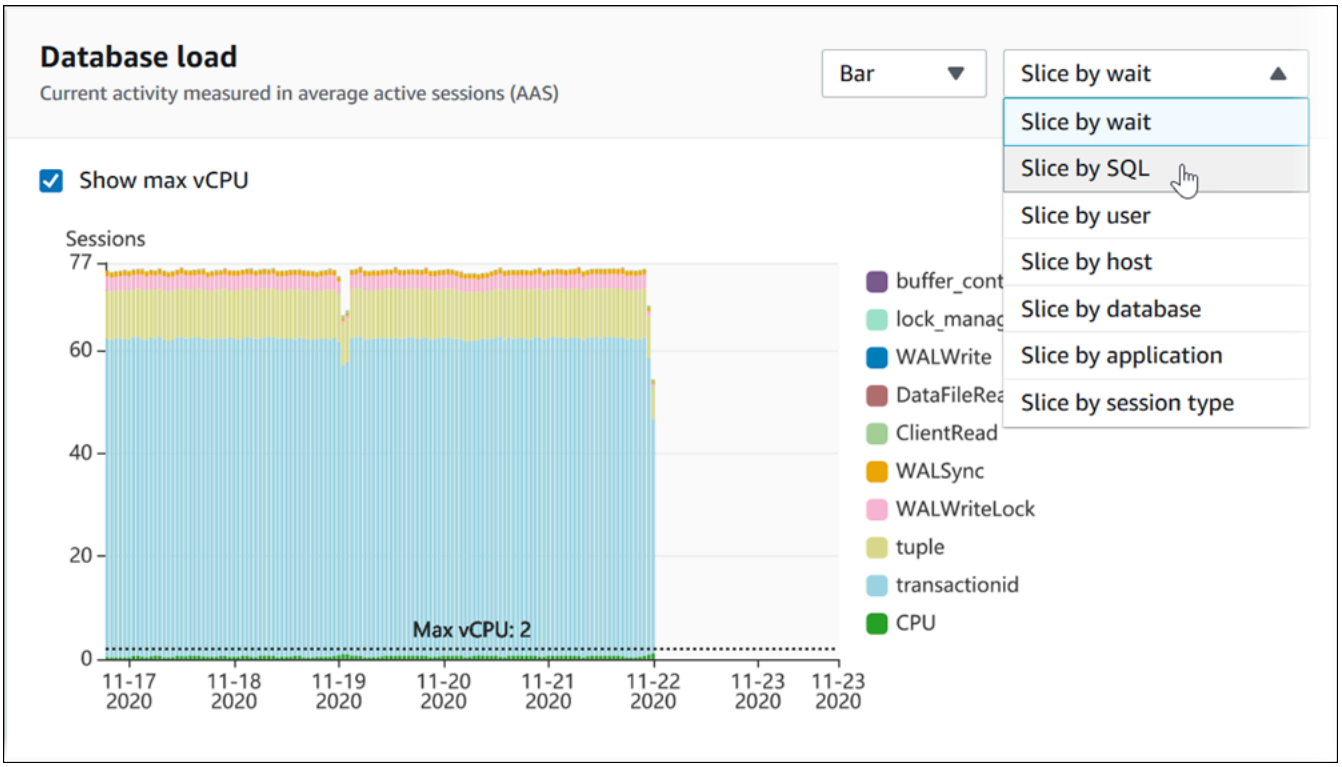


Muatan DB diiris berdasarkan dimensi

Anda dapat memilih untuk menampilkan muatan sebagai sesi aktif yang dikelompokkan berdasarkan dimensi yang didukung. Tabel berikut menunjukkan dimensi yang didukung untuk mesin yang berbeda.

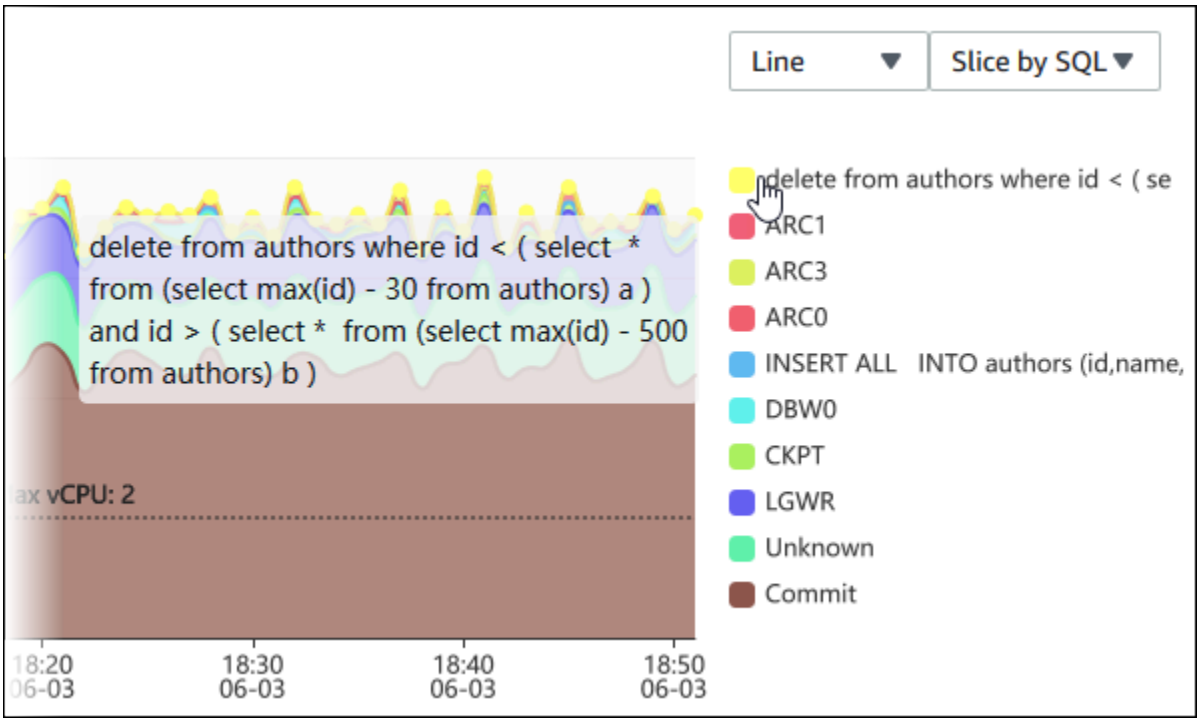
Dimensi	Oracle	SQL Server	PostgreSQL	MySQL
Host	Ya	Ya	Ya	Ya
SQL	Ya	Ya	Ya	Ya
Pengguna	Ya	Ya	Ya	Ya
Tunggu	Ya	Ya	Ya	Ya
Rencana	Ya	Tidak	Tidak	Tidak
Aplikasi	Tidak	Tidak	Ya	Tidak
Basis data	Tidak	Tidak	Ya	Ya
Jenis sesi	Tidak	Tidak	Ya	Tidak

Gambar berikut menunjukkan dimensi untuk instans DB PostgreSQL.

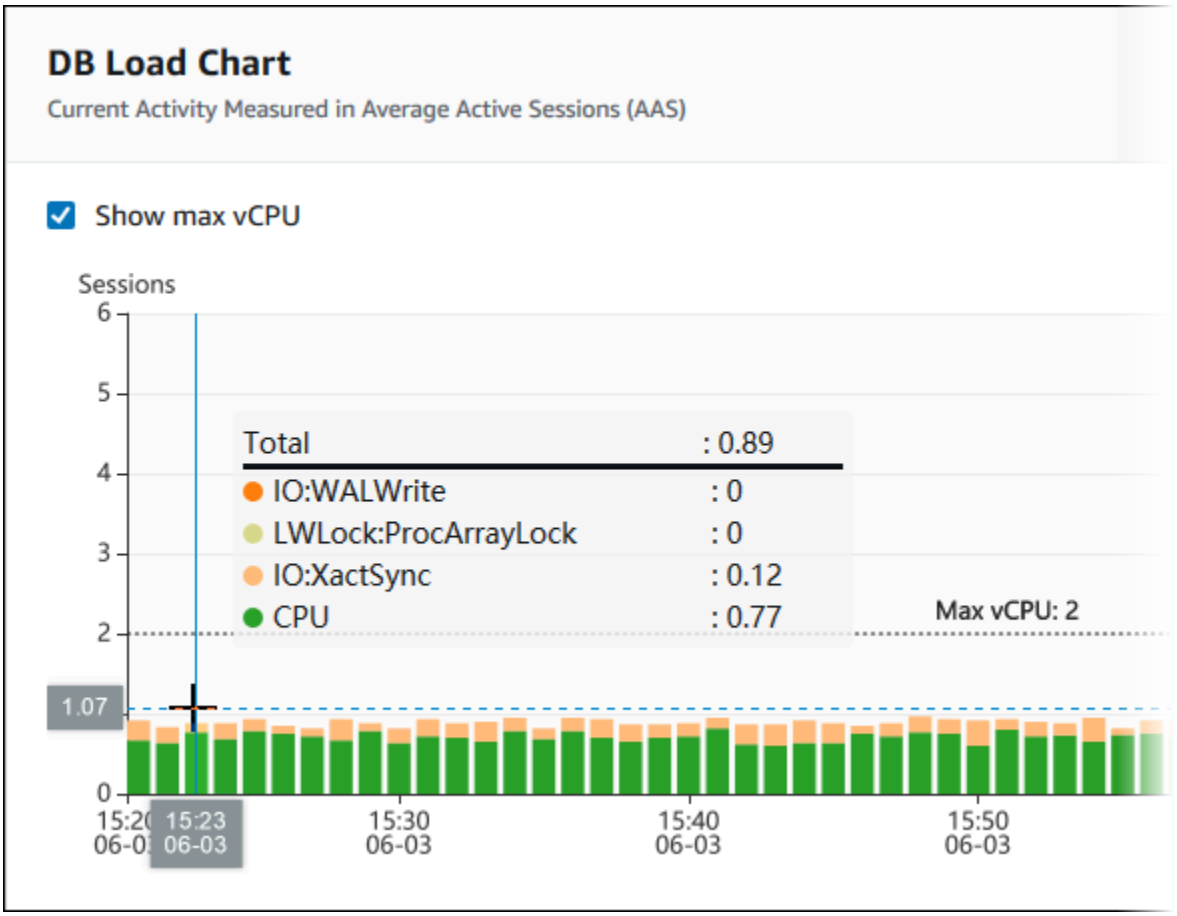


Detail muatan DB untuk item dimensi

Untuk melihat detail tentang item muatan DB dalam dimensi, arahkan kursor ke nama item. Gambar berikut menunjukkan rincian untuk pernyataan SQL.

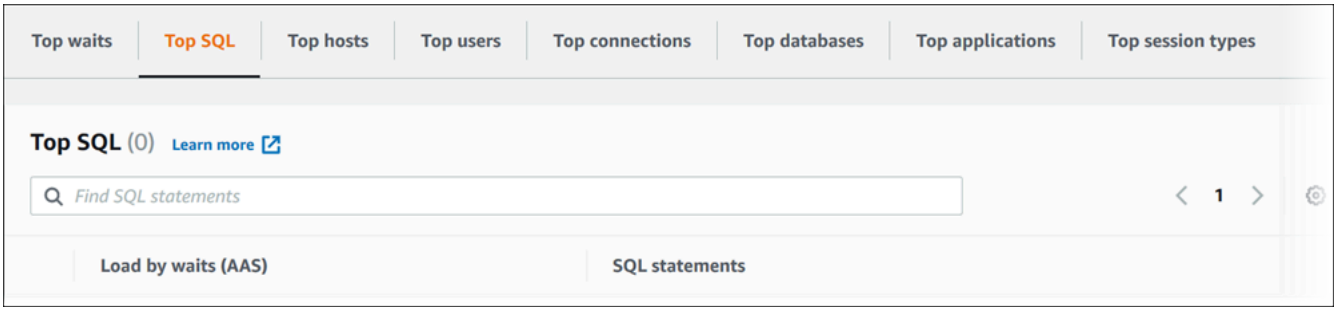


Untuk melihat detail setiap item selama periode waktu yang dipilih dalam legenda, arahkan kursor ke item tersebut.



Tabel Dimensi teratas

Tabel Dimensi teratas mengiris muatan DB dengan dimensi yang berbeda. Dimensi adalah kategori atau "potong menurut" untuk karakteristik muatan DB yang berbeda. Jika dimensinya adalah SQL, SQL Teratas menunjukkan pernyataan SQL yang berkontribusi paling besar terhadap muatan DB.



Pilih salah satu tab dimensi berikut.

Tab	Deskripsi	Mesin yang didukung
SQL Teratas	Pernyataan SQL yang saat ini sedang berjalan	Semua
Tunggu teratas	Peristiwa di mana backend basis data sedang menunggu	Semua
Host teratas	Nama host klien yang terhubung	Semua
Pengguna teratas	Pengguna yang masuk ke basis data	Semua
Basis data teratas	Nama basis data yang terhubung ke klien	Khusus PostgreSQL, MySQL, dan MariaDB
Aplikasi teratas	Nama aplikasi yang terhubung ke basis data	Khusus PostgreSQL
Jenis sesi teratas	Jenis sesi saat ini	Khusus PostgreSQL

Untuk mempelajari cara menganalisis kueri dengan menggunakan tab SQL Teratas, lihat Ringkasan tab SQL Teratas.

Mengakses dasbor Wawasan Performa

Amazon RDS menyediakan tampilan Wawasan Performa dan metrik CloudWatch terpadu di dasbor Wawasan Performa.

Untuk mengakses dasbor Wawasan Performa, gunakan prosedur berikut.

Untuk melihat dasbor Wawasan Performa di Konsol Manajemen AWS

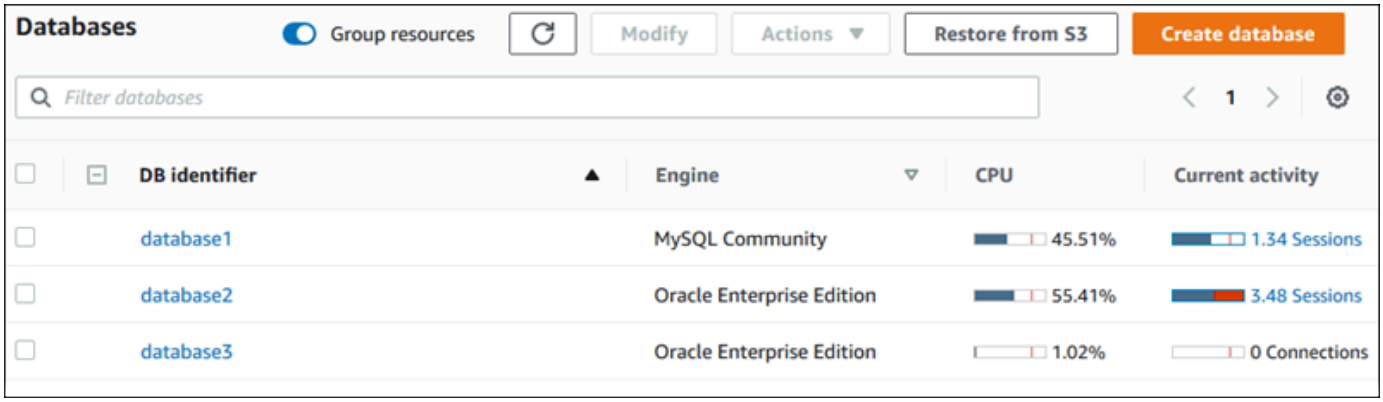
1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi kiri, pilih Wawasan Performa.
3. Pilih instans DB.
4. Pilih tampilan pemantauan default di jendela yang ditampilkan.

- Pilih opsi Tampilan Wawasan Performa dan metrik CloudWatch (Baru), lalu pilih Lanjutkan untuk melihat Wawasan Performa dan metrik CloudWatch.
- Pilih opsi Tampilan Wawasan Performa, lalu pilih Lanjutkan untuk tampilan pemantauan lama. Kemudian, lanjutkan dengan prosedur ini.

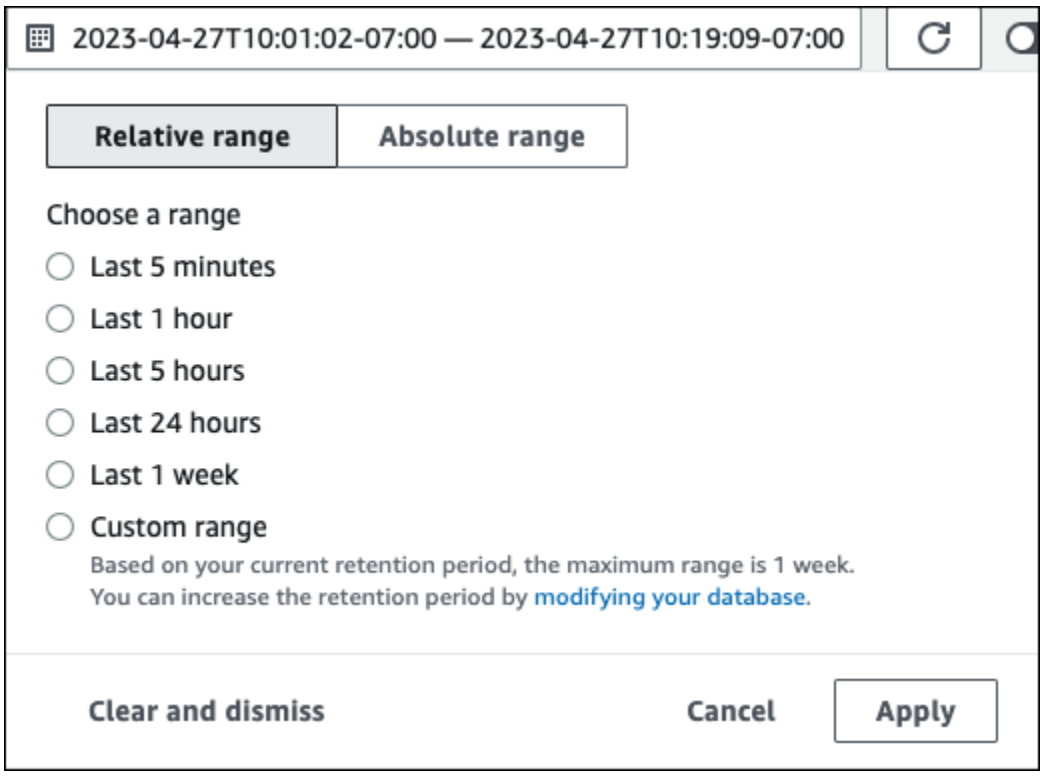
> **Note**
>
> Tampilan ini akan dihentikan pada 15 Desember 2023.

Dasbor Wawasan Performa muncul untuk instans DB.

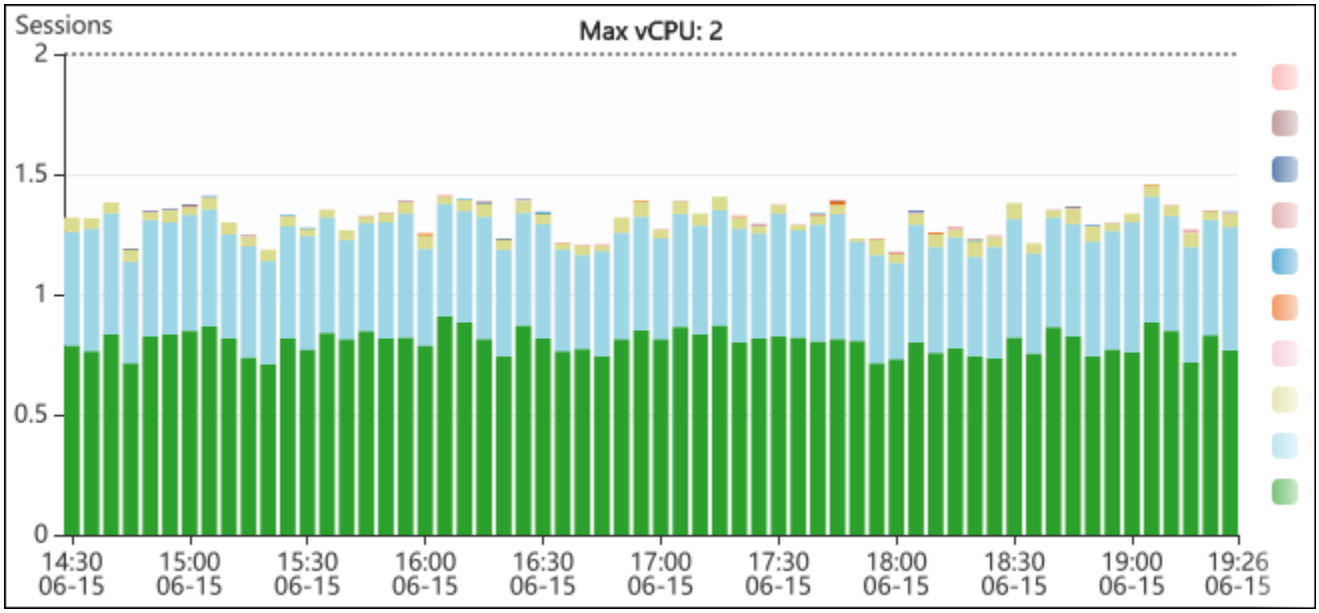
Untuk instans DB dengan Wawasan Performa yang diaktifkan, Anda juga dapat mengakses dasbor dengan memilih item Sesi di daftar instans DB. Di bagian Aktivitas saat ini, item Sesi menunjukkan muatan basis data dalam sesi aktif rata-rata selama lima menit terakhir. Bilah ini secara grafis menunjukkan muatan. Jika bilah kosong, berarti instans DB sedang diam. Saat muatan meningkat, bilah akan terisi dengan warna biru. Saat muatan melewati jumlah CPU virtual (vCPU) pada kelas instans DB, bilah akan berubah menjadi merah, yang menunjukkan adanya potensi kemacetan.



5. (Opsional) Pilih rentang tanggal atau waktu di kanan atas dan tentukan interval waktu relatif atau absolut yang berbeda. Anda kini dapat menentukan periode waktu, dan menghasilkan laporan analisis performa basis data. Laporan ini berisi rekomendasi dan wawasan yang diidentifikasi. Untuk informasi selengkapnya, lihat Membuat laporan analisis performa.

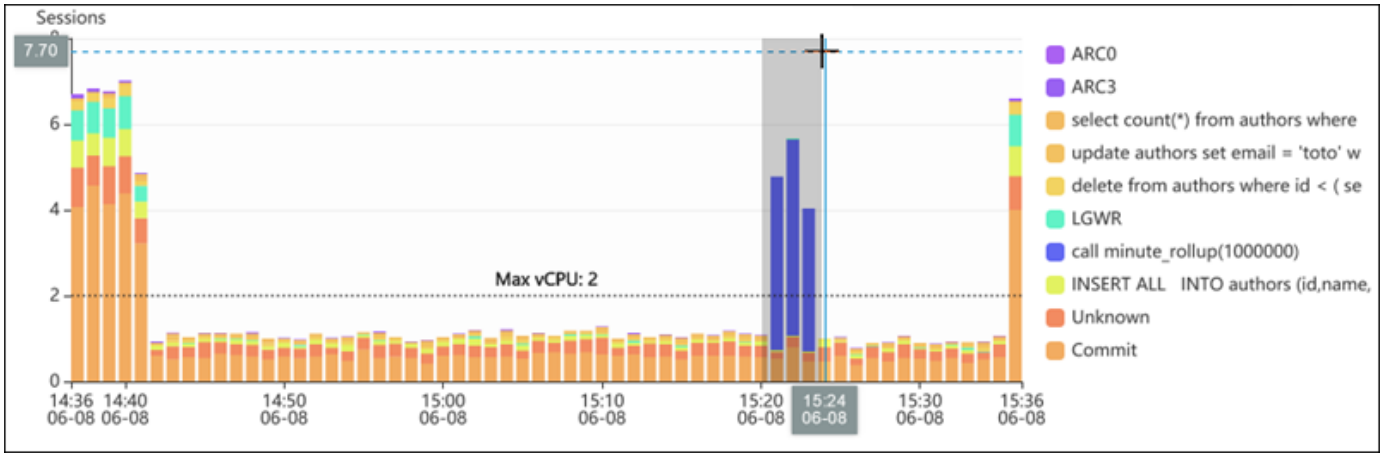


Di tangkapan layar berikut, interval muatan DB adalah 5 jam.

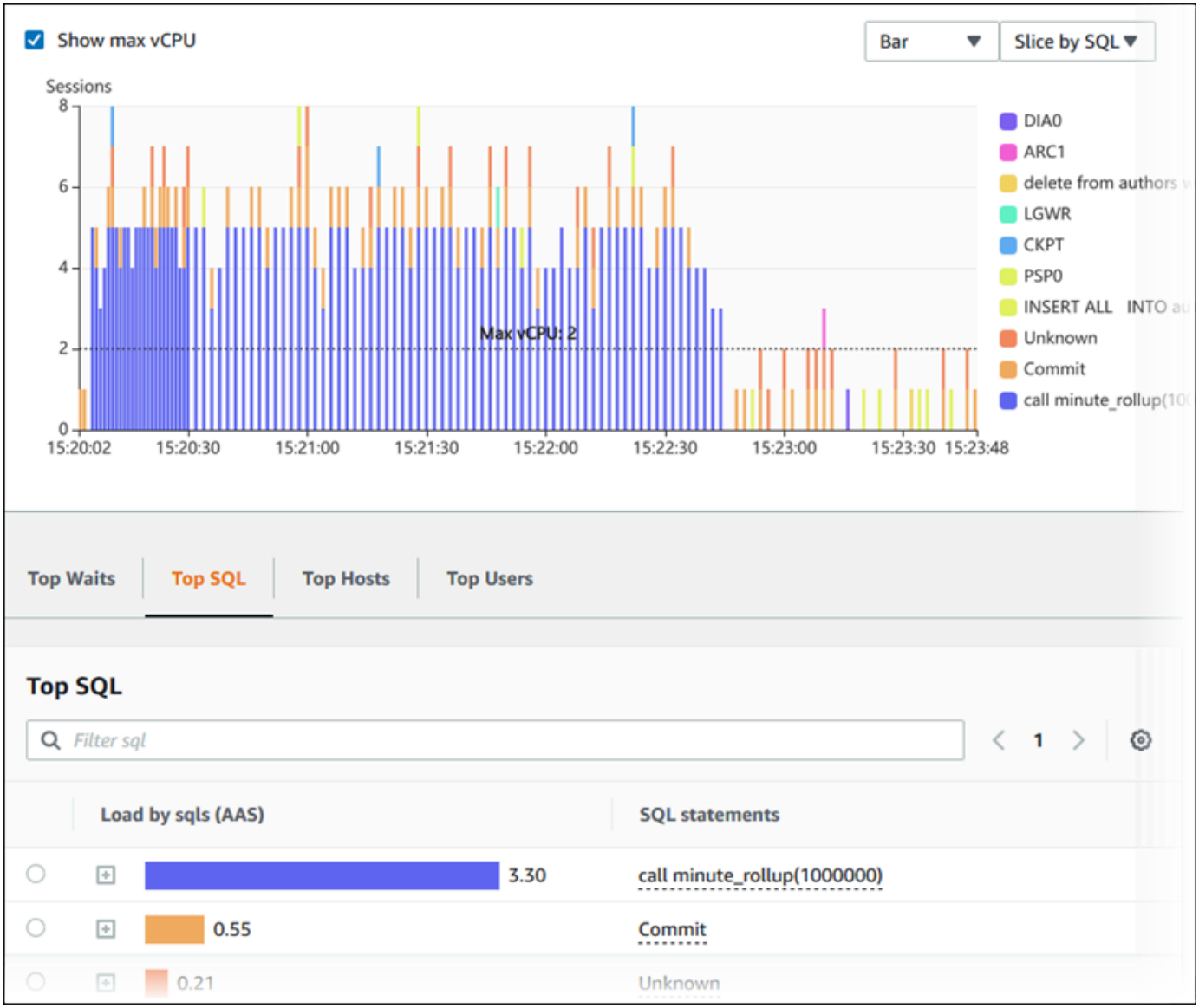


6. (Opsional) Untuk memperbesar sebagian bagan muatan DB, pilih waktu mulai dan seret ke akhir periode waktu yang diinginkan.

   Area yang dipilih disorot dalam bagan muatan DB.



Saat melepaskan mouse, bagan muatan DB akan diperbesar di Wilayah AWS yang dipilih, dan tabel Dimensi teratas dihitung ulang.



7. (Opsional) Untuk menyegarkan data Anda secara otomatis, pilih Segarkan otomatis.

Dasbor Wawasan Performa secara otomatis disegarkan dengan data baru. Laju penyegaran bergantung pada jumlah data yang ditampilkan:

- Penyegaran 5 menit setiap 10 detik.
- Penyegaran 1 jam setiap 5 menit.
- Penyegaran 5 jam setiap 5 menit.
- Penyegaran 24 jam setiap 30 menit.
- Penyegaran 1 minggu setiap hari.

- Penyegaran 1 bulan setiap hari.

Menganalisis muatan DB menurut peristiwa tunggu

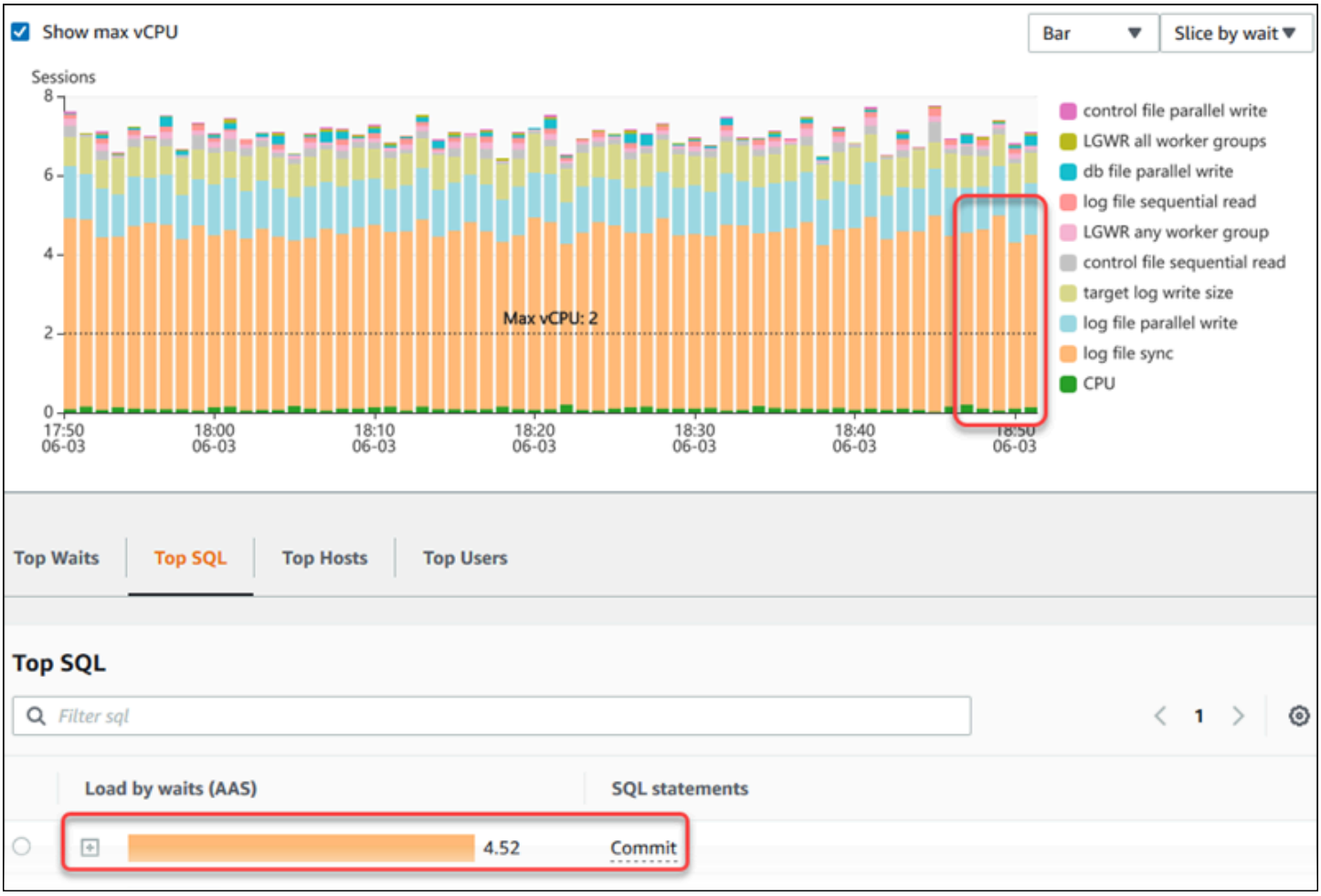
Jika bagan Muatan basis data menunjukkan kemacetan, Anda dapat mengetahui dari mana muatan tersebut berasal. Untuk melakukannya, lihat tabel item muatan teratas di bawah bagan Muatan basis data. Pilih item tertentu, seperti kueri SQL atau pengguna, untuk menelusuri item tersebut dan melihat detailnya.

Muatan DB yang dikelompokkan berdasarkan peristiwa tunggu dan kueri SQL teratas adalah tampilan default dasbor Wawasan Performa. Kombinasi ini biasanya memberikan wawasan paling banyak tentang masalah performa. Muatan DB yang dikelompokkan berdasarkan peristiwa tunggu menunjukkan apakah ada sumber daya atau kemacetan konkurensi dalam basis data. Dalam kasus ini, tab SQL dari tabel item muatan teratas menunjukkan kueri mana yang mendorong muatan tersebut.

Alur kerja tipikal Anda untuk mendiagnosis masalah performa adalah sebagai berikut:

1. Tinjau bagan Muatan basis data dan lihat apakah ada insiden muatan basis data yang melebihi baris CPU Maks.
2. Jika ada, lihat bagan Muatan basis data dan identifikasi satu atau beberapa status tunggu yang paling bertanggung jawab.
3. Identifikasi kueri digest yang menyebabkan muatan dengan melihat kueri mana dari tab SQL di tabel item muatan teratas yang berkontribusi paling besar pada status tunggu tersebut. Anda dapat mengidentifikasinya berdasarkan kolom Muatan DB berdasarkan peristiwa Tunggu.
4. Pilih salah satu kueri digest di tab SQL untuk meluaskannya dan melihat kueri turunan yang menyusunnya.

Misalnya, di dasbor berikut, peristiwa tunggu sinkronisasi file log menyumbang sebagian besar muatan DB. Peristiwa tunggu LGWR semua grup pekerja juga tinggi. Bagan SQL Teratas menunjukkan penyebab peristiwa tunggu sinkronisasi file log: pernyataan `COMMIT` yang sering. Dalam kasus ini, eksekusi yang lebih jarang akan mengurangi muatan DB.



Menganalisis performa basis data selama periode waktu tertentu

Anda dapat membuat laporan analisis performa selama periode waktu tertentu dan mengetahui masalah performa apa pun seperti kemacetan sumber daya atau perubahan dalam kueri di instans DB Anda. Dasbor Wawasan Performa memungkinkan Anda memilih periode waktu dan membuat laporan analisis performa. Anda juga dapat menambahkan satu tag atau lebih ke laporan.

Untuk menggunakan fitur ini, Anda harus menggunakan periode retensi tingkat berbayar. Untuk informasi selengkapnya, lihat Harga dan retensi data untuk Wawasan Performa

Laporan ini dapat dipilih dan dilihat di tab Laporan analisis performa - baru. Laporan ini berisi wawasan, metrik terkait, dan rekomendasi untuk menyelesaikan masalah performa. Laporan ini dapat dilihat selama periode retensi Wawasan Performa.

Laporan dihapus jika waktu mulai periode analisis laporan berada di luar periode retensi. Anda juga dapat menghapus laporan sebelum periode retensi berakhir.

Untuk mendeteksi masalah performa dan menghasilkan laporan analisis untuk instans DB, Anda harus mengaktifkan Wawasan Performa. Untuk informasi selengkapnya tentang mengaktifkan Wawasan Performa, lihat Mengaktifkan dan menonaktifkan Wawasan Performa.

Untuk informasi dukungan wilayah, mesin DB, dan kelas instans untuk fitur ini, lihat Dukungan kelas instans, Wilayah, dan mesin DB Amazon RDS untuk fitur Wawasan Performa

Membuat laporan analisis performa

Anda dapat membuat laporan analisis performa selama periode tertentu di dasbor Wawasan Performa. Anda dapat memilih periode waktu dan menambahkan satu tag atau lebih ke laporan analisis.

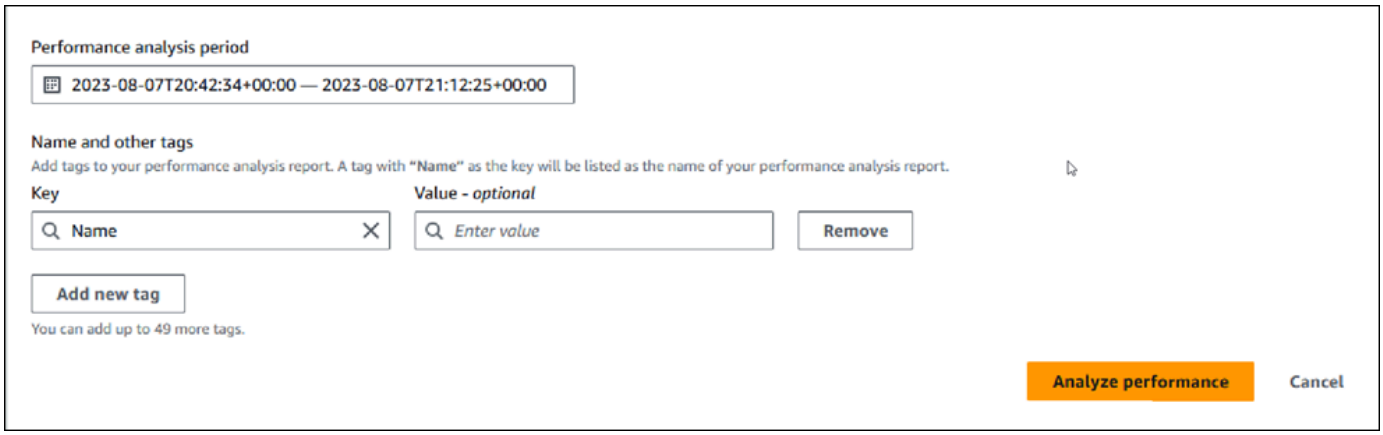
Periode analisis bisa berkisar dari 5 menit hingga 6 hari. Harus ada data performa setidaknya 24 jam sebelum waktu mulai analisis.

Untuk membuat laporan analisis performa selama periode waktu tertentu

1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi kiri, pilih Wawasan Performa.
3. Pilih instans DB.

   Dasbor Wawasan Performa muncul untuk instans DB.
4. Pilih Analisis performa di bagian Muatan basis data di dasbor.

   Kolom untuk mengatur periode waktu dan menambahkan satu atau beberapa tag ke laporan analisis performa ditampilkan.



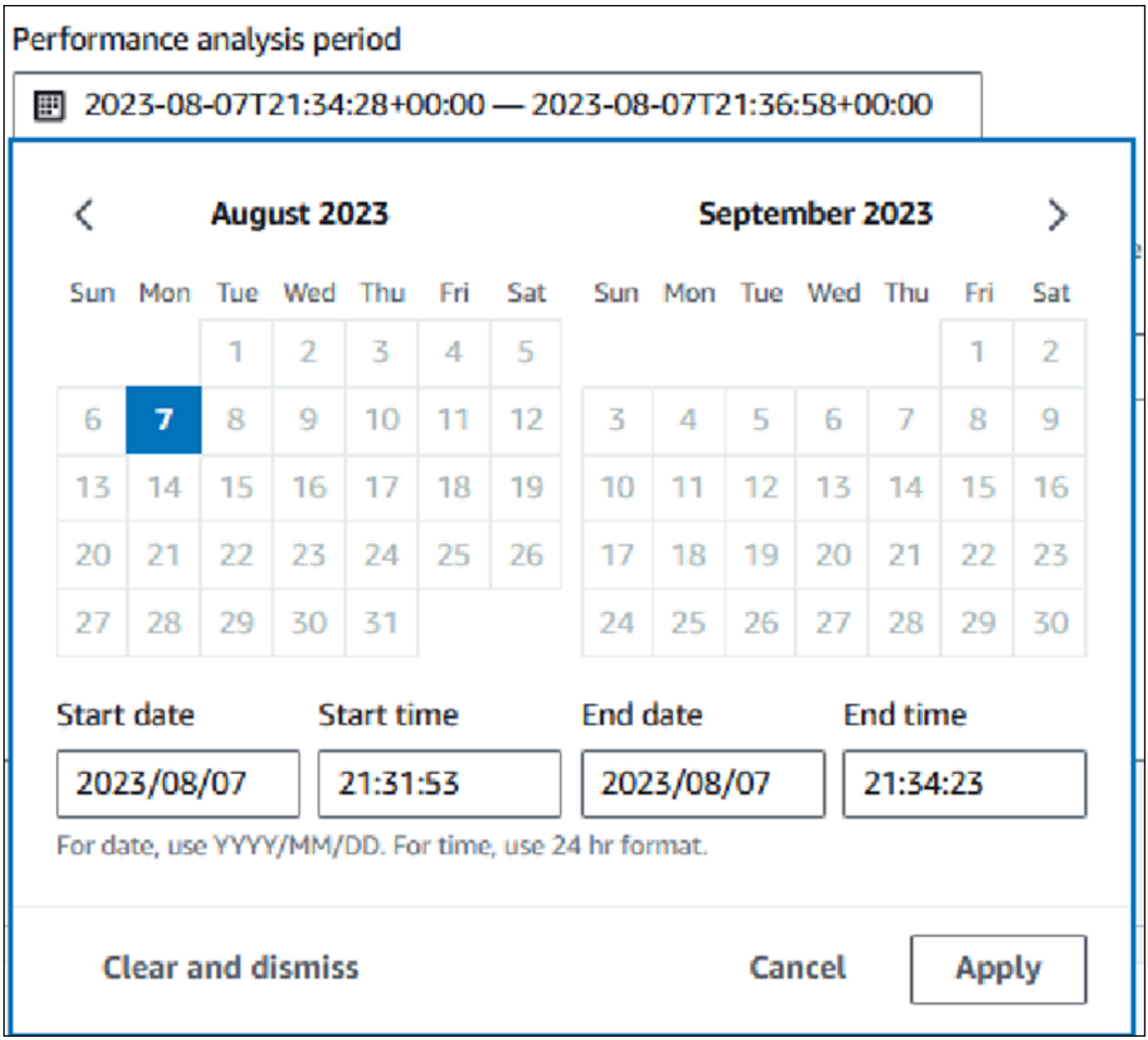
5. Pilih periode waktu. Jika menetapkan periode waktu dalam Rentang relatif atau Rentang absolut di kanan atas, Anda hanya dapat memasukkan atau memilih tanggal dan waktu laporan analisis

dalam periode waktu ini. Jika Anda memilih periode analisis di luar periode waktu ini, pesan kesalahan akan muncul.

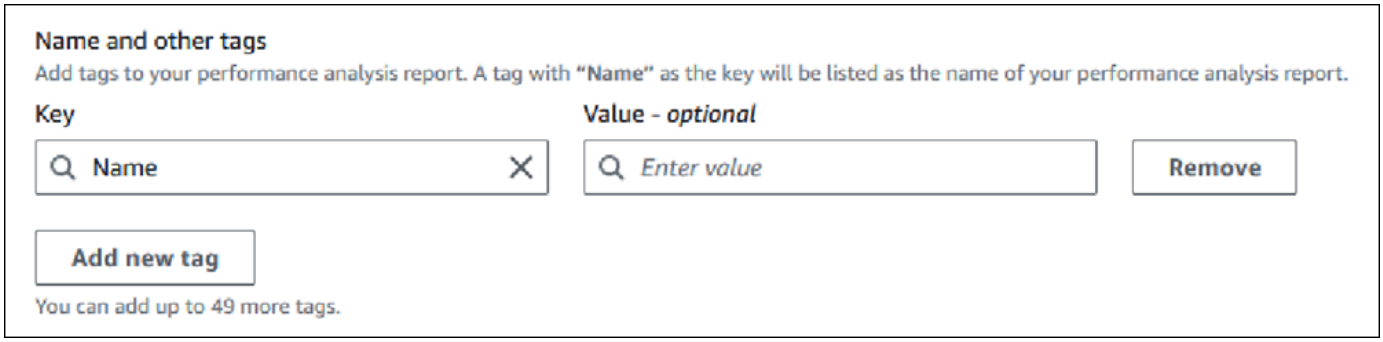
Untuk mengatur periode waktu, Anda dapat mengikuti langkah-langkah berikut:

- Tekan dan seret salah satu slider pada bagan muatan DB.

  Kotak Periode analisis performa menampilkan periode waktu yang dipilih dan bagan muatan DB menyoroti periode waktu yang dipilih.
- Pilih Tanggal mulai, Waktu mulai, Tanggal akhir, dan Waktu akhir di kotak Periode analisis performa.



6. (Opsional) Masukkan Kunci dan Nilai-opsional untuk menambahkan tag untuk laporan.



7. Pilih Analisis performa.

   Spanduk menampilkan pesan apakah pembuatan laporan berhasil atau gagal. Pesan juga menyediakan tautan untuk melihat laporan.

   Contoh berikut menunjukkan banner dengan pesan pembuatan laporan berhasil.

   Laporan ini dapat dilihat di tab Laporan analisis performa - baru.

Anda dapat membuat laporan analisis performa menggunakan AWS CLI. Untuk contoh tentang cara membuat laporan menggunakan AWS CLI, lihat Membuat laporan analisis performa selama periode waktu tertentu.

Melihat laporan analisis performa

Tab Laporan analisis performa - baru mencantumkan semua laporan yang dibuat untuk instans DB. Berikut ini ditampilkan untuk setiap laporan:

- ID: Pengidentifikasi unik laporan.
- Nama: Kunci tag yang ditambahkan ke laporan.
- Waktu pembuatan laporan: Waktu Anda membuat laporan.
- Waktu mulai analisis: Waktu mulai analisis dalam laporan.
- Waktu akhir analisis: Waktu akhir analisis dalam laporan.

Untuk melihat laporan analisis performa

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi kiri, pilih Wawasan Performa.
3. Pilih instans DB yang laporan analisisnya ingin dilihat.

   Dasbor Wawasan Performa muncul untuk instans DB.
4. Gulir ke bawah dan pilih tab Laporan analisis performa - baru.

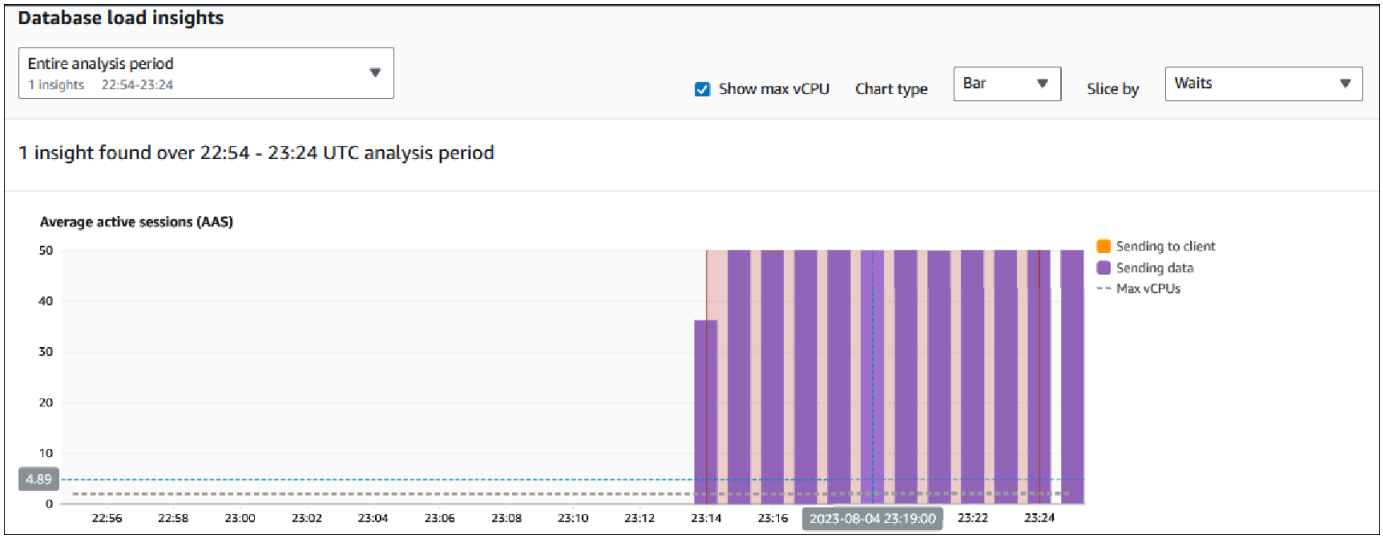
   Semua laporan analisis untuk periode waktu yang berbeda ditampilkan.

5. Pilih ID laporan yang ingin dilihat.

   Bagan muatan DB menampilkan seluruh periode analisis secara default jika ada lebih dari satu wawasan yang diidentifikasi. Jika laporan telah mengidentifikasi satu wawasan, bagan muatan DB akan menampilkan wawasan secara default.

   Dasbor juga mencantumkan tag untuk laporan di bagian Tag.

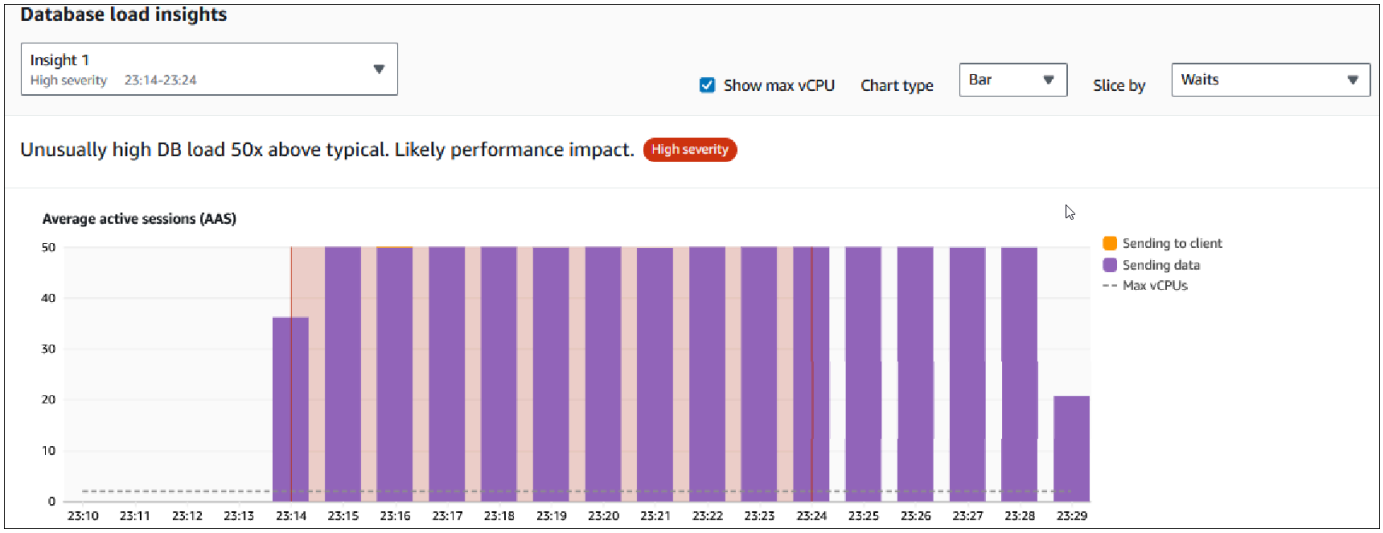
   Contoh berikut menunjukkan seluruh periode analisis untuk laporan.

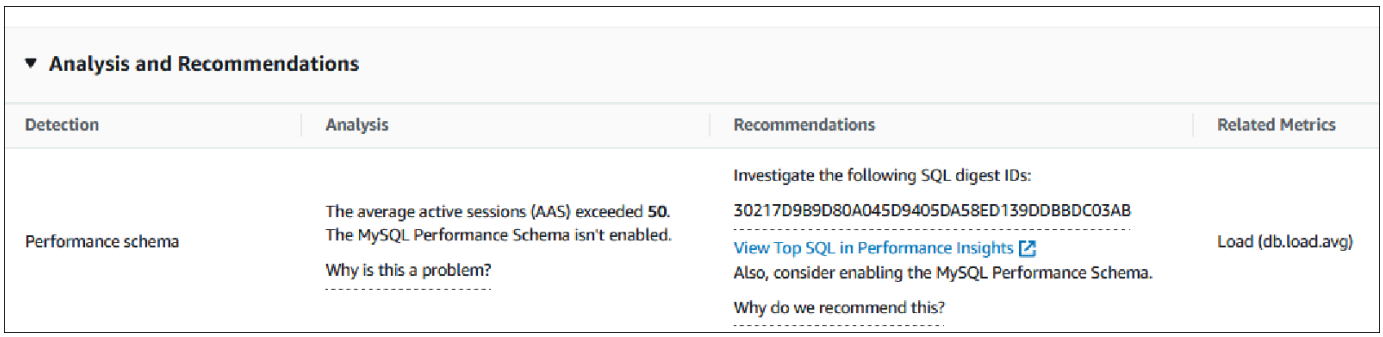


6. Pilih wawasan dalam daftar Wawasan muatan basis data yang ingin dilihat jika ada lebih dari satu wawasan yang diidentifikasi dalam laporan.

   Dasbor menampilkan pesan wawasan, bagan muatan DB yang menyoroti periode waktu wawasan, analisis dan rekomendasi, serta daftar tag laporan.

   Contoh berikut menunjukkan wawasan muatan DB dalam laporan.





Menambahkan tanda ke laporan analisis performa

Anda dapat menambahkan tag saat membuat atau melihat laporan. Anda dapat menambahkan hingga 50 tag untuk sebuah laporan.

Anda memerlukan izin untuk menambahkan tag. Untuk informasi selengkapnya tentang kebijakan akses untuk Wawasan Performa, lihat Mengonfigurasi kebijakan akses untuk Wawasan Performa

Untuk menambahkan satu atau beberapa tag saat membuat laporan, lihat langkah 6 dalam prosedur Membuat laporan analisis performa.

Untuk menambahkan satu tag atau lebih saat melihat laporan

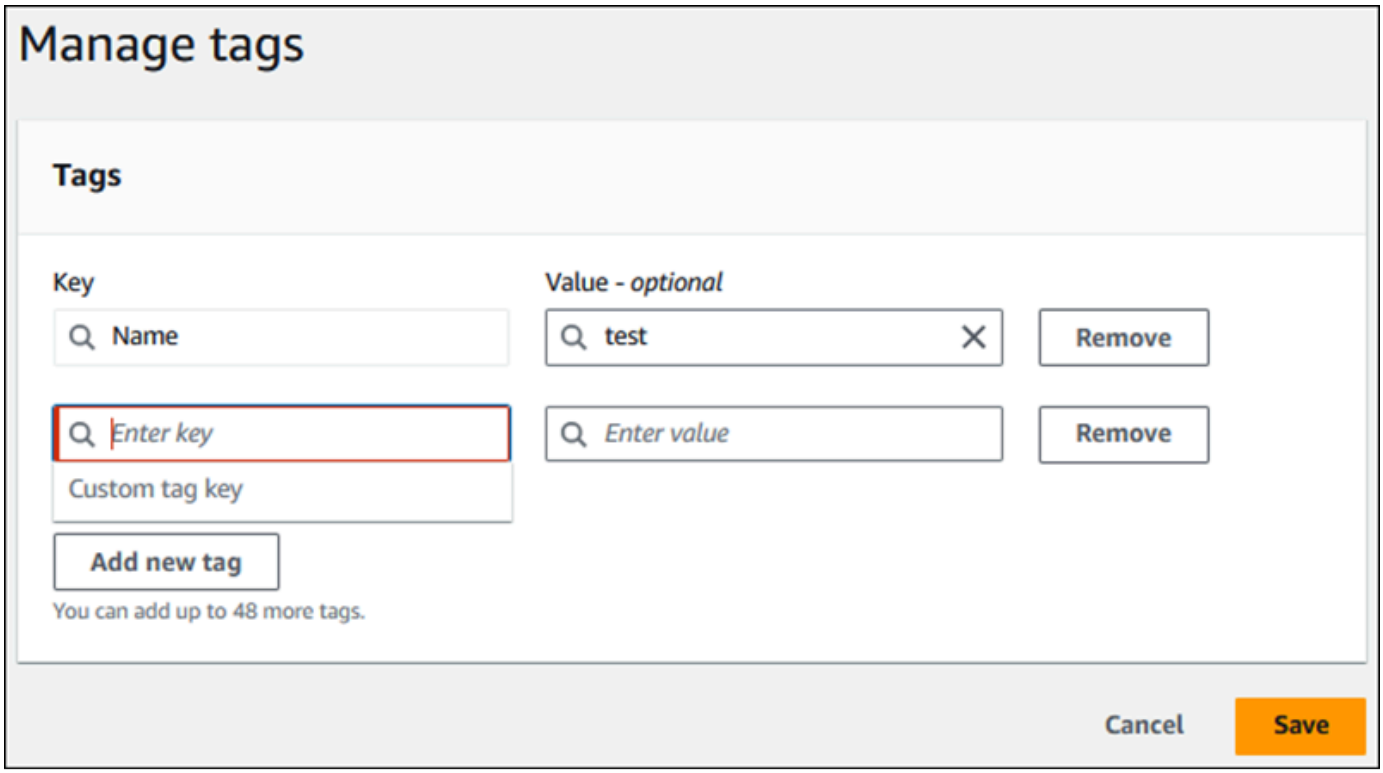
1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi kiri, pilih Wawasan Performa.
3. Pilih instans DB.

   Dasbor Wawasan Performa muncul untuk instans DB.

4. Gulir ke bawah dan pilih tab Laporan analisis performa - baru.
5. Pilih laporan yang ingin diberi tag.

   Dasbor menampilkan laporan.
6. Gulir ke bawah ke Tag dan pilih Kelola tag.
7. Pilih Tambahkan tag baru.
8. Masukkan Kunci dan Nilai - opsional, dan pilih Tambahkan tag baru.

   Contoh berikut memberikan opsi untuk menambahkan tag baru untuk laporan yang dipilih.



   Tag baru dibuat untuk laporan.

   Daftar tag untuk laporan ditampilkan di bagian Tag pada dasbor. Jika Anda ingin menghapus tag dari laporan, pilih Hapus di samping tag.

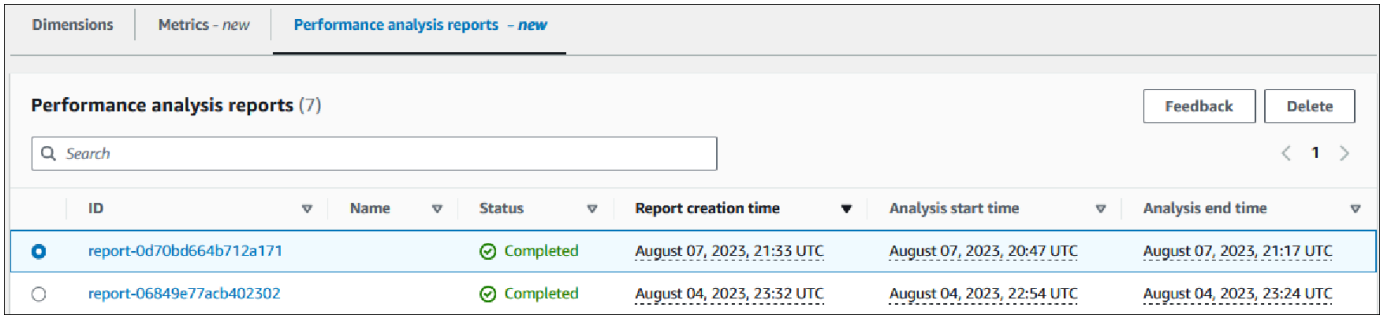
Menghapus laporan analisis performa

Anda dapat menghapus laporan dari daftar laporan yang ditampilkan di tab Laporan analisis performa atau saat melihat laporan.

Untuk menghapus laporan

1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi kiri, pilih Wawasan Performa.
3. Pilih instans DB.

   Dasbor Wawasan Performa muncul untuk instans DB.
4. Gulir ke bawah dan pilih tab Laporan analisis performa - baru.
5. Pilih laporan yang ingin dihapus dan pilih Hapus di kanan atas.



   Jendela konfirmasi ditampilkan. Laporan akan dihapus setelah Anda memilih konfirmasi.
6. (Opsional) Pilih ID laporan yang ingin dihapus.

   Di halaman laporan, pilih Hapus di kanan atas.

   Jendela konfirmasi ditampilkan. Laporan akan dihapus setelah Anda memilih konfirmasi.

Menganalisis kueri di dasbor Wawasan Performa

Di dasbor Wawasan Performa Amazon RDS, Anda dapat menemukan informasi tentang menjalankan kueri terbaru di tab SQL Teratas di tabel Dimensi teratas. Anda dapat menggunakan informasi ini untuk menyesuaikan kueri Anda.

Topik

- Ringkasan tab SQL Teratas
- Mengakses lebih banyak teks SQL di dasbor Wawasan Performa
- Melihat statistik SQL di dasbor Wawasan Performa

Ringkasan tab SQL Teratas

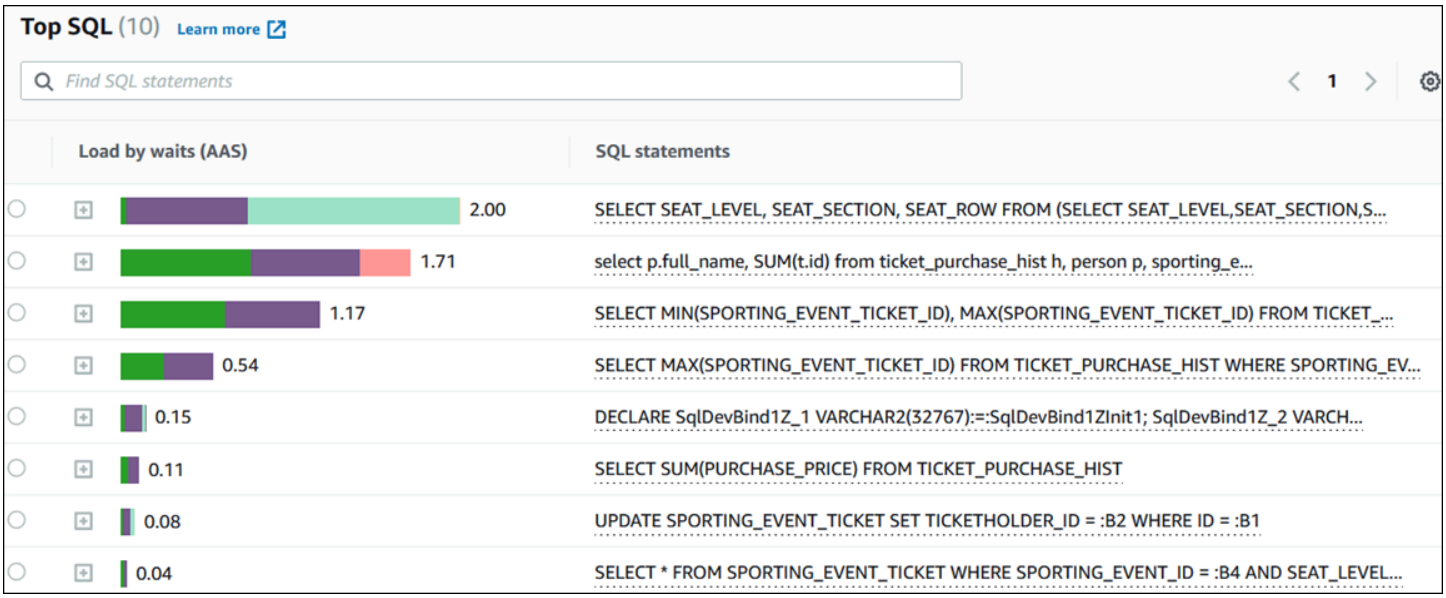
Secara default, tab SQL Teratas menunjukkan 25 kueri yang paling berkontribusi pada muatan DB. Untuk membantu menyetel kueri, Anda dapat menganalisis informasi seperti teks kueri dan statistik SQL. Anda juga dapat memilih statistik yang ingin ditampilkan di tab SQL Teratas.

Topik

- Teks SQL
- Statistik SQL
- Muatan berdasarkan status tunggu (AAS)
- Informasi SQL
- Preferensi

Teks SQL

Secara default, setiap baris dalam tabel SQL Teratas menunjukkan 500 byte teks untuk setiap pernyataan.

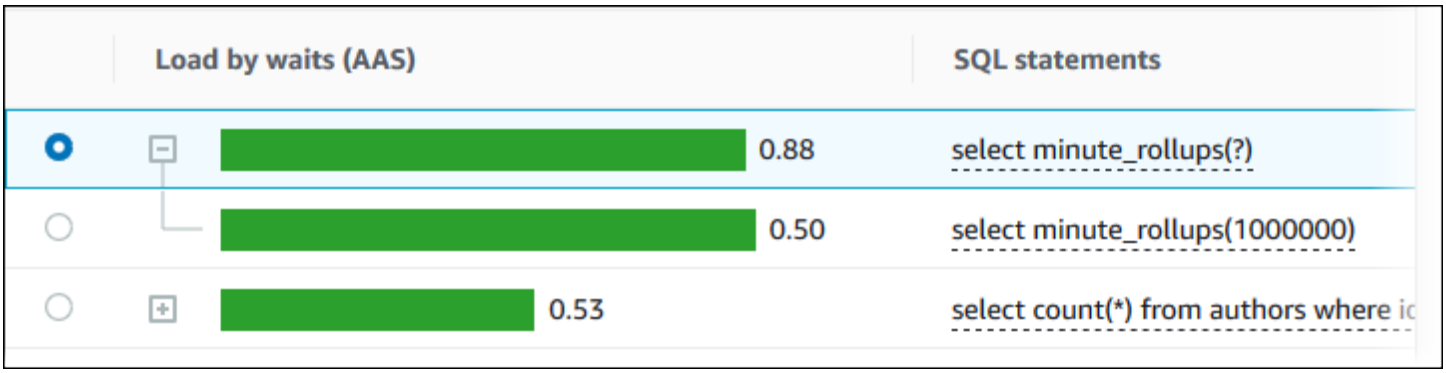


Untuk mempelajari cara melihat lebih dari 500 byte default teks SQL, lihat Mengakses lebih banyak teks SQL di dasbor Wawasan Performa.

Digest SQL adalah gabungan dari beberapa kueri aktual yang secara struktural serupa, tetapi mungkin memiliki nilai literal yang berbeda. Digest menggantikan nilai berkode keras dengan tanda tanya. Misalnya, digest mungkin berupa `SELECT * FROM emp WHERE lname= ?`. Digest ini dapat mencakup kueri turunan berikut:

```
SELECT * FROM emp WHERE lname = 'Sanchez'
SELECT * FROM emp WHERE lname = 'Olagappan'
SELECT * FROM emp WHERE lname = 'Wu'
```

Untuk menampilkan pernyataan SQL literal dalam sebuah digest, pilih kueri, lalu pilih simbol plus (+). Dalam contoh berikut, kueri yang dipilih adalah digest.



> **Note**
>
> Digest SQL mengelompokkan pernyataan SQL yang serupa, tetapi tidak menyunting informasi sensitif.

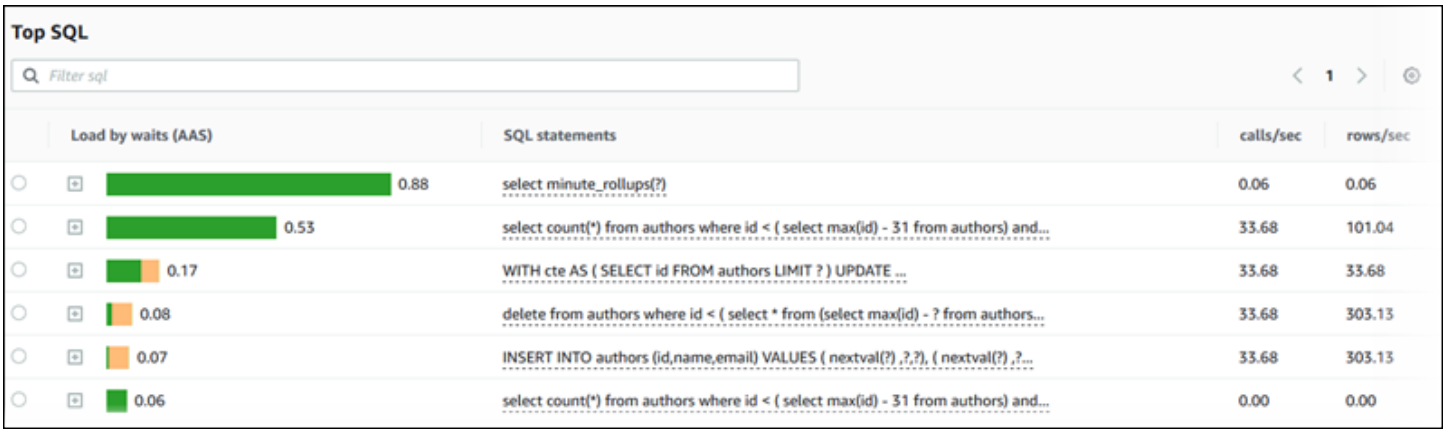
Wawasan Performa dapat menampilkan teks Oracle SQL sebagai Tidak Diketahui. Teks memiliki status ini dalam situasi berikut:

- Pengguna basis data Oracle selain SYS memang aktif, tetapi saat ini tidak menjalankan SQL. Misalnya, ketika kueri paralel selesai, koordinator kueri menunggu proses pembantu untuk mengirim statistik sesinya. Selama menunggu, teks kueri menunjukkan Tidak Diketahui.
- Untuk contoh RDS untuk instans Oracle pada Standard Edition 2, Oracle Resource Manager membatasi jumlah untai paralel. Proses latar belakang yang melakukan pekerjaan ini menyebabkan teks kueri ditampilkan sebagai Tidak Diketahui.

Statistik SQL

Statistik SQL adalah metrik terkait performa tentang kueri SQL. Misalnya, Wawasan Performa mungkin menampilkan eksekusi per detik atau baris yang diproses per detik. Wawasan Performa mengumpulkan statistik hanya untuk kueri yang paling umum. Biasanya, ini cocok dengan kueri teratas berdasarkan muatan yang ditampilkan di dasbor Wawasan Performa.

Setiap baris dalam tabel SQL Teratas menunjukkan statistik yang relevan untuk pernyataan SQL atau digest, seperti yang ditunjukkan dalam contoh berikut.



Wawasan Performa dapat melaporkan `0.00` dan `-` (tidak diketahui) untuk statistik SQL. Situasi ini terjadi dalam kondisi berikut:

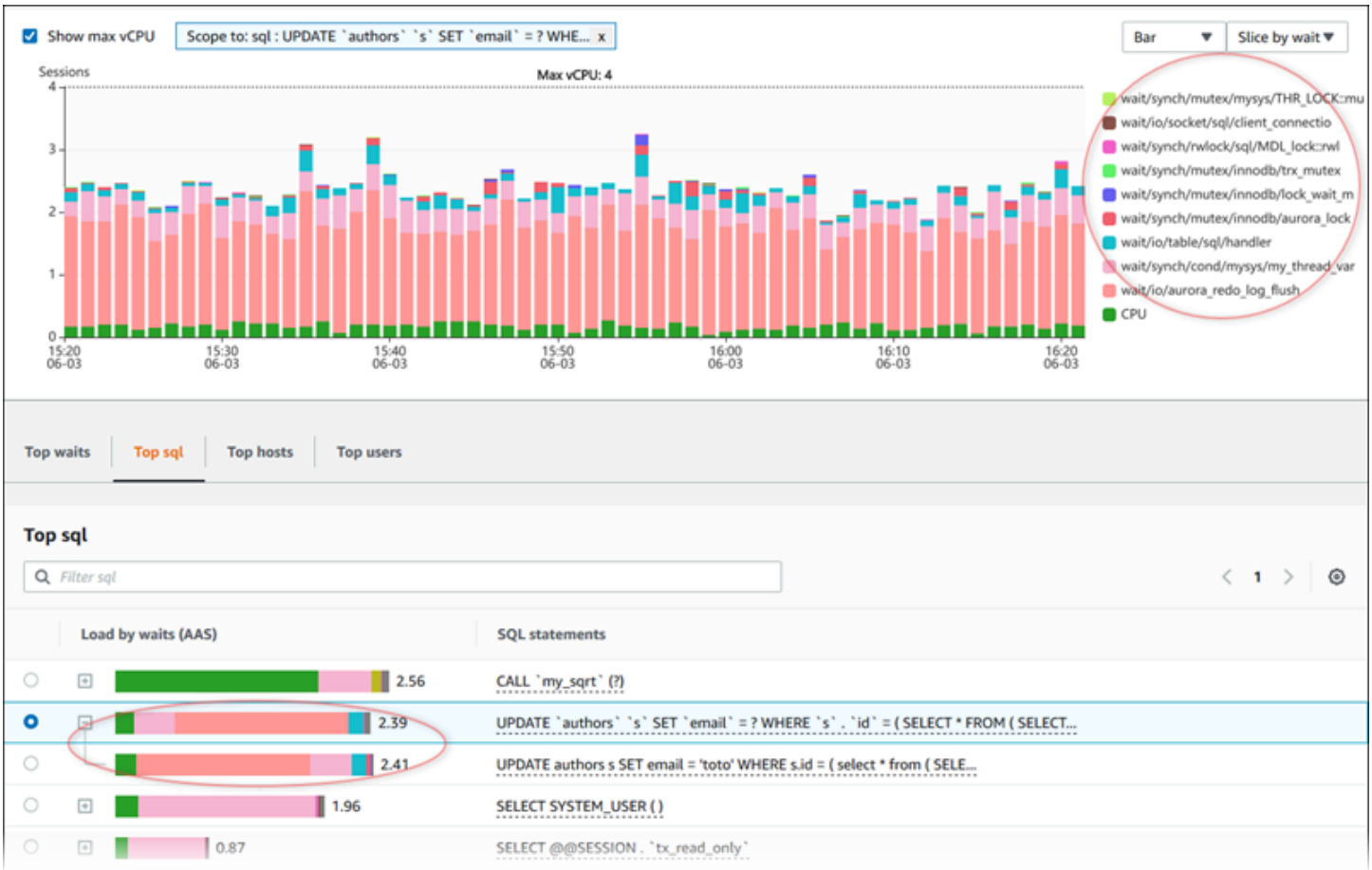
- Hanya ada satu sampel. Misalnya, Wawasan Performa menghitung tingkat perubahan untuk kueri RDS PostgreSQL berdasarkan beberapa sampel dari tampilan `pg_stats_statements`. Ketika beban kerja berjalan untuk waktu yang singkat, Wawasan Performa mungkin hanya mengumpulkan satu sampel, yang berarti tidak dapat menghitung tingkat perubahan. Nilai yang tidak diketahui ditunjukkan dengan tanda hubung (`-`).
- Dua sampel memiliki nilai yang sama. Wawasan Performa tidak dapat menghitung tingkat perubahan karena tidak ada perubahan yang terjadi, sehingga melaporkan tingkatnya sebagai `0.00`.
- Pernyataan RDS PostgreSQL tidak memiliki pengidentifikasi yang valid. PostgreSQL membuat pengidentifikasi untuk pernyataan hanya setelah diurai dan dianalisis. Dengan demikian, pernyataan bisa hadir dalam struktur internal dalam memori PostgreSQL tanpa pengidentifikasi. Karena Wawasan Performa mengambil sampel struktur internal dalam memori sekali per detik, kueri latensi rendah mungkin muncul hanya untuk satu sampel. Jika pengidentifikasi kueri tidak tersedia untuk sampel ini, Wawasan Performa tidak dapat mengaitkan pernyataan ini dengan statistiknya. Nilai yang tidak diketahui ditunjukkan dengan tanda hubung (`-`).

Untuk deskripsi statistik SQL untuk mesin Amazon RDS, lihat Statistik SQL untuk Wawasan Performa.

Muatan berdasarkan status tunggu (AAS)

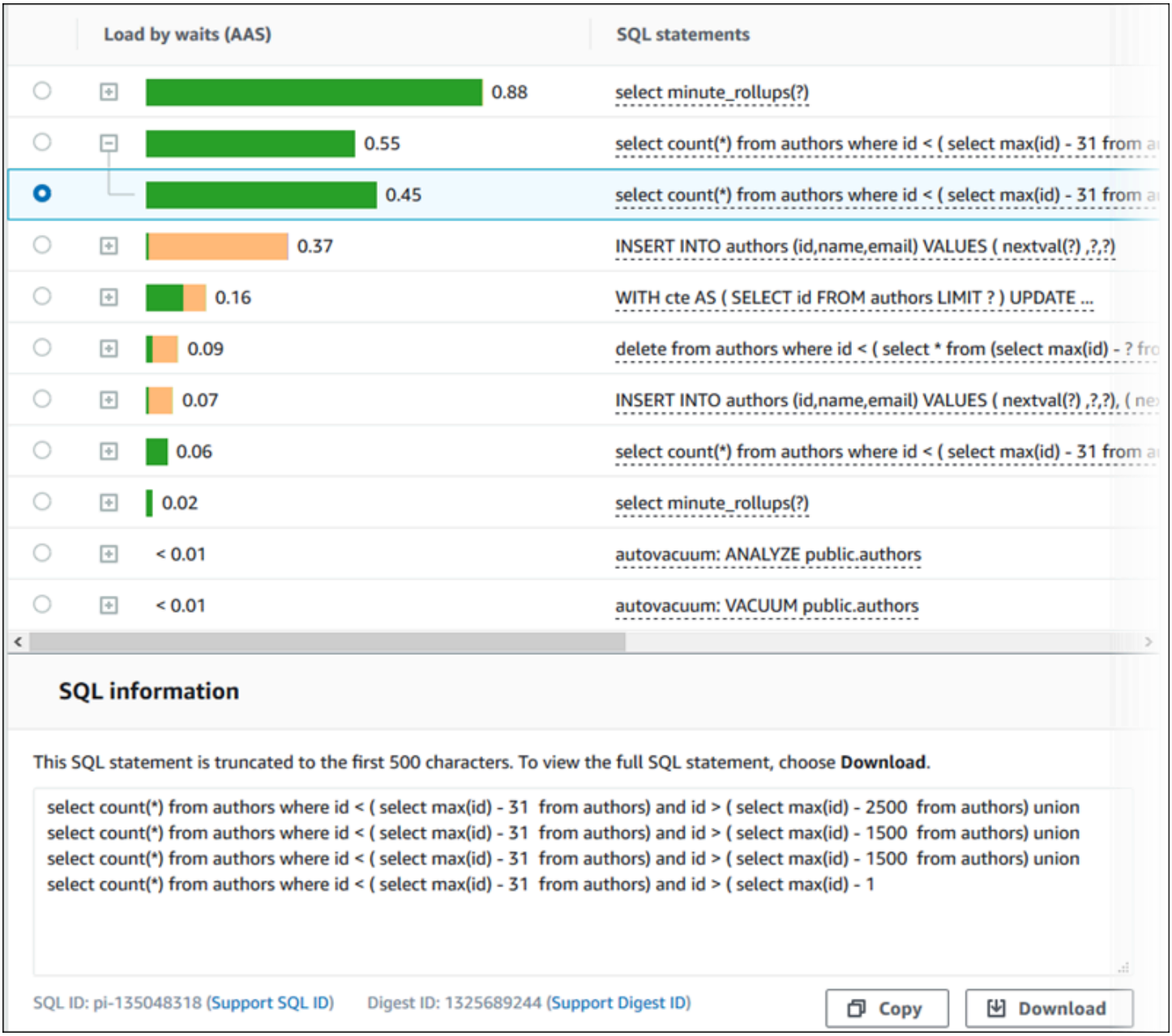
Di SQL Teratas, kolom Muatan berdasarkan status tunggu (AAS) menggambarkan persentase muatan basis data yang terkait dengan setiap item muatan teratas. Kolom ini menunjukkan muatan untuk item tersebut berdasarkan pengelompokan apa pun yang saat ini dipilih di Bagan Muatan DB. Untuk informasi selengkapnya tentang Sesi aktif rata-rata (AAS), lihat Sesi aktif rata-rata.

Misalnya, Anda mungkin mengelompokkan bagan Muatan DB berdasarkan status tunggu. Anda memeriksa kueri SQL di tabel item muatan teratas. Dalam kasus ini, bilah Muatan DB berdasarkan Status Tunggu diberi ukuran, disegmentasi, dan diberi kode warna untuk menunjukkan seberapa banyak status tunggu tertentu yang dikontribusikan oleh kueri. Bilah ini juga menunjukkan status tunggu yang memengaruhi kueri yang dipilih.



Informasi SQL

Di tabel SQL Teratas, Anda dapat membuka pernyataan untuk menampilkan informasinya. Informasi ini muncul di panel bawah.

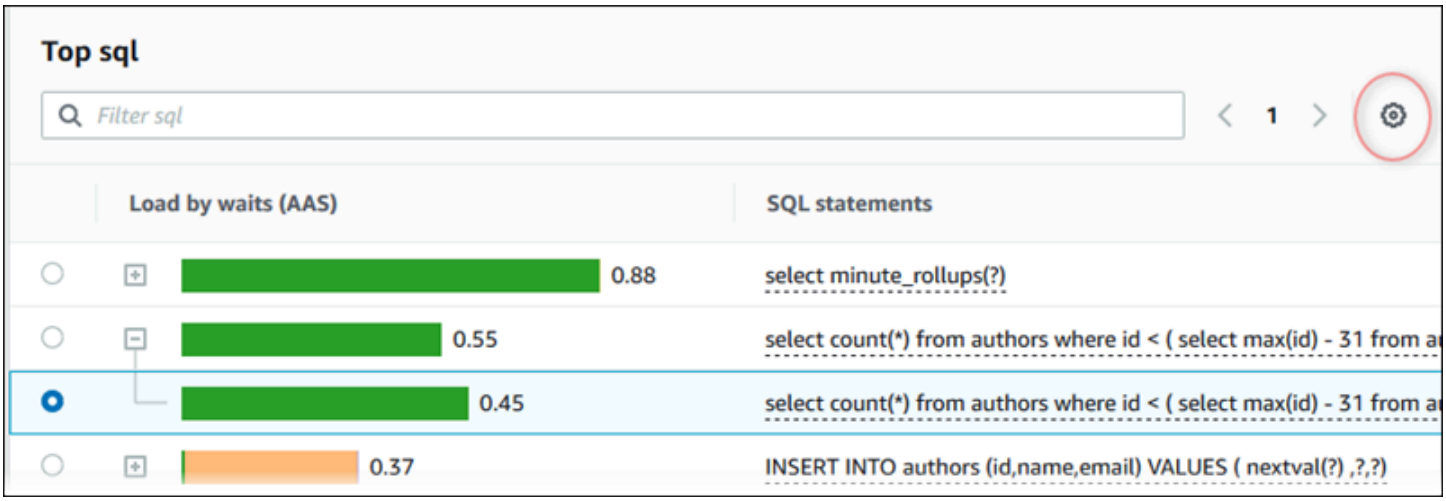


Berikut ini adalah jenis pengidentifikasi (ID) yang terkait dengan pernyataan SQL:

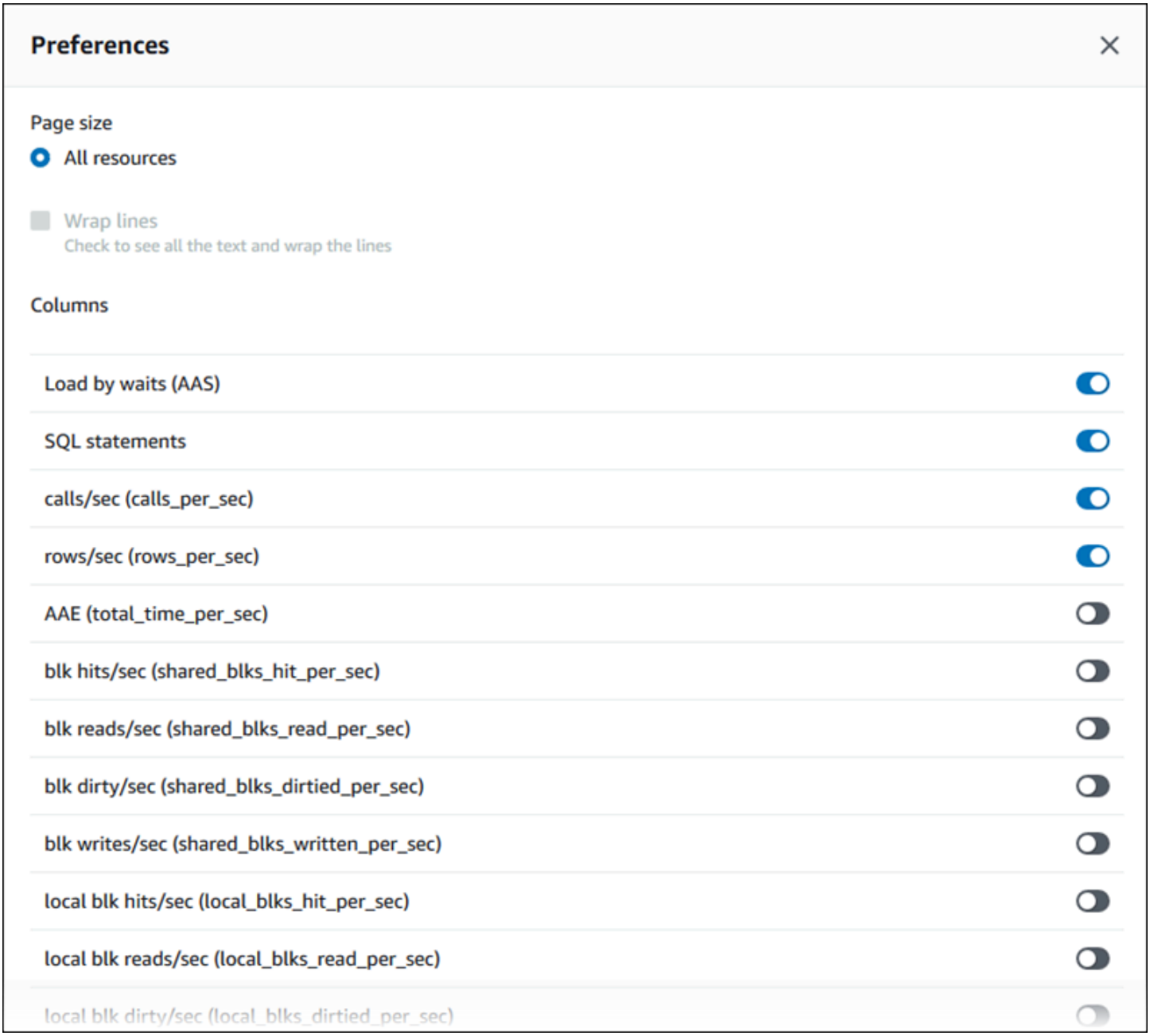
- Support SQL ID – Nilai hash dari ID SQL. Nilai ini hanya untuk mereferensikan ID SQL saat Anda bekerja dengan Dukungan AWS. AWS Dukungan tidak memiliki akses ke ID SQL dan teks SQL Anda yang sebenarnya.
- Support Digest ID – Nilai hash dari ID digest. Nilai ini hanya untuk mereferensikan ID digest saat Anda bekerja dengan Dukungan AWS. AWS Dukungan tidak memiliki akses ke ID digest dan teks SQL Anda yang sebenarnya.

Preferensi

Anda dapat mengontrol statistik yang ditampilkan di tab SQL Teratas dengan memilih ikon Preferensi.



Saat memilih ikon Preferensi, jendela Preferensi akan terbuka. Tangkapan layar berikut adalah contoh jendela Preferensi.



Untuk mengaktifkan statistik yang ingin ditampilkan di tab SQL Teratas, gunakan mouse untuk menggulir ke bagian bawah jendela, lalu pilih Lanjutkan.

Untuk informasi selengkapnya tentang statistik per detik atau per panggilan untuk mesin Amazon RDS, lihat bagian statistik SQL khusus mesin di Statistik SQL untuk Wawasan Performa

Mengakses lebih banyak teks SQL di dasbor Wawasan Performa

Secara default, setiap baris dalam tabel SQL Teratas menunjukkan 500 byte teks SQL untuk setiap pernyataan SQL.

Saat pernyataan SQL melebihi 500 byte, Anda dapat melihat lebih banyak teks di bagian Teks SQL di bawah tabel SQL Teratas. Dalam hal ini, panjang maksimum teks yang ditampilkan dalam Teks SQL adalah 4 KB. Batas ini diperkenalkan oleh konsol dan tunduk pada batas yang ditetapkan oleh mesin basis data. Untuk menyimpan teks yang ditampilkan dalam Teks SQL, pilih Unduh.

Topik

- Batas ukuran teks untuk mesin Amazon RDS
- Mengatur batas teks SQL untuk instans DB Amazon RDS for PostgreSQL
- Melihat dan mengunduh teks SQL di dasbor Wawasan Performa

Batas ukuran teks untuk mesin Amazon RDS

Saat Anda mengunduh teks SQL, mesin basis data menentukan panjang maksimumnya. Anda dapat mengunduh teks SQL hingga batas per mesin berikut.

Mesin DB	Panjang maksimum teks yang diunduh
Amazon RDS for MySQL dan MariaDB	1.024 byte
Amazon RDS for Microsoft SQL Server	4.096 karakter
Amazon RDS for Oracle	100.000 byte

Bagian Teks SQL dari konsol Wawasan Performa menampilkan hingga maksimum yang ditampilkan mesin. Misalnya, jika MySQL menampilkan paling banyak 1 KB ke Wawasan Performa, Aurora MySQL hanya dapat mengumpulkan dan menampilkan 1 KB, meskipun kueri asalnya lebih besar. Jadi, jika Anda melihat kueri dalam Teks SQL atau mengunduhnya, Wawasan Performa akan menampilkan jumlah byte yang sama.

Jika Anda menggunakan AWS CLI atau API, Wawasan Performa tidak memiliki batas 4 KB yang diberlakukan oleh konsol. `DescribeDimensionKeys` dan `GetResourceMetrics` menampilkan paling banyak 500 byte. `GetDimensionKeyDetails` menampilkan kueri lengkap, tetapi ukurannya tunduk pada batas mesin.

Mengatur batas teks SQL untuk instans DB Amazon RDS for PostgreSQL

Amazon RDS for PostgreSQL menangani teks secara berbeda. Anda dapat mengatur batas ukuran teks dengan parameter instans DB `track_activity_query_size`. Parameter ini memiliki karakteristik sebagai berikut:

Ukuran teks default

Di Amazon RDS for PostgreSQL versi 9.6, pengaturan default untuk parameter `track_activity_query_size` adalah 1.024 byte. Di Amazon RDS for PostgreSQL versi 10 atau yang lebih baru, pengaturan default-nya adalah 4.096 byte.

Ukuran teks maksimum

Batas untuk `track_activity_query_size` adalah 102.400 byte untuk Amazon RDS for PostgreSQL versi 12 dan versi yang lebih rendah. Ukuran maksimumnya adalah 1 MB untuk versi 13 dan yang lebih baru.

Jika mesin menampilkan 1 MB ke Wawasan Performa, konsol hanya akan menampilkan 4 KB pertama. Jika mengunduh kueri, Anda akan mendapatkan 1 MB penuh. Dalam hal ini, melihat dan mengunduh menampilkan jumlah byte yang berbeda. Untuk informasi selengkapnya tentang parameter instans DB `track_activity_query_size`, lihat Statistik Runtime dalam dokumentasi PostgreSQL.

Untuk meningkatkan ukuran teks SQL, tingkatkan batas `track_activity_query_size`. Untuk memodifikasi parameter, ubah pengaturan parameter di grup parameter yang terkait dengan instans DB Amazon RDS for PostgreSQL.

Untuk mengubah pengaturan saat instans menggunakan grup parameter default

1. Buat grup parameter instans DB baru untuk mesin DB dan versi mesin DB yang sesuai.
2. Tetapkan parameter di grup parameter baru.
3. Hubungkan grup parameter baru dengan instans DB.

Untuk informasi tentang cara mengatur parameter instans DB, lihat Memodifikasi parameter dalam grup parameter DB.

Melihat dan mengunduh teks SQL di dasbor Wawasan Performa

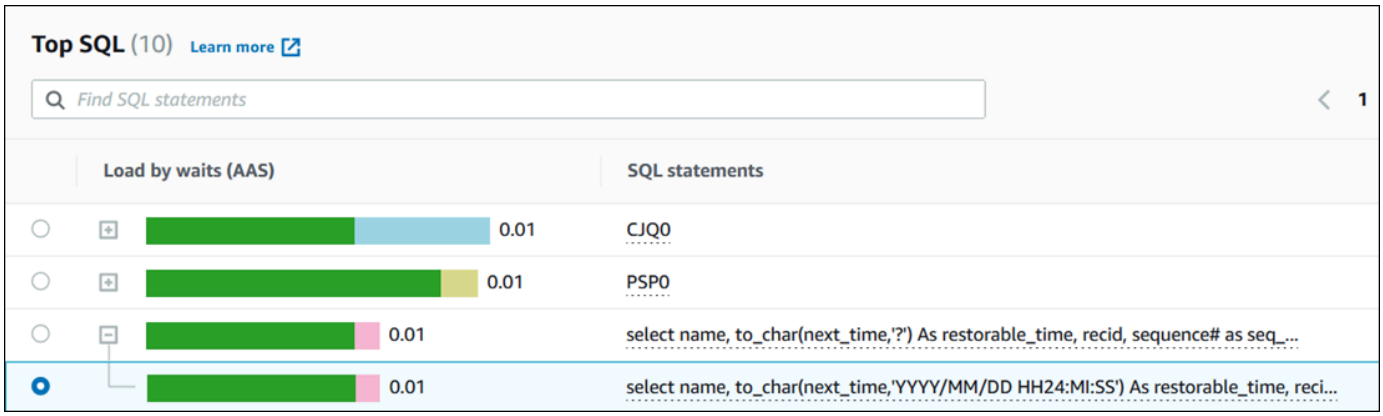
Di dasbor Wawasan Performa, Anda dapat melihat atau mengunduh teks SQL.

Untuk melihat lebih banyak teks SQL di dasbor Wawasan Performa

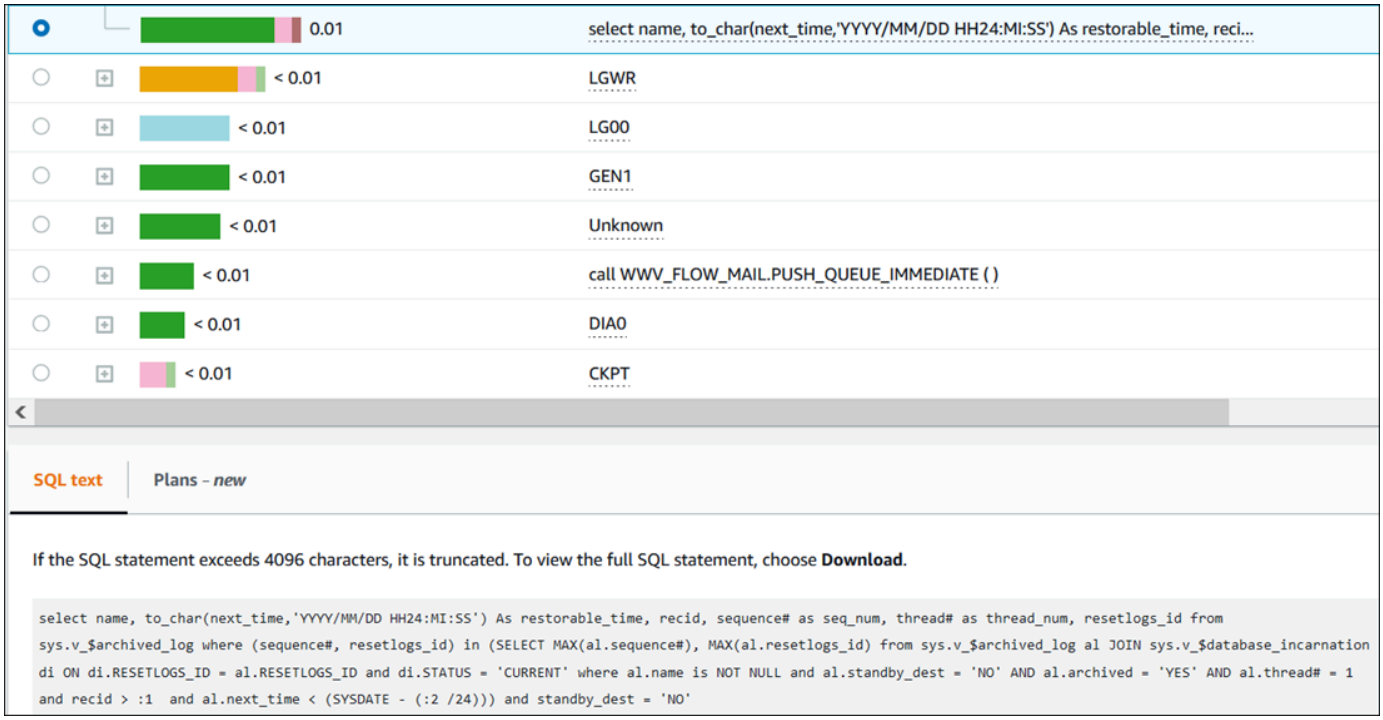
1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Wawasan Performa.
3. Pilih instans DB.

   Dasbor Wawasan Performa ditampilkan untuk instans DB Anda.
4. Gulir ke bawah ke tab SQL Teratas.
5. Pilih pernyataan SQL.

   Pernyataan SQL dengan teks yang lebih besar dari 500 byte terlihat hampir seperti gambar berikut.



6. Gulir ke bawah ke tab Teks SQL.



Dasbor Wawasan Performa dapat menampilkan hingga 4.096 byte untuk setiap pernyataan SQL.

7. (Opsional) Pilih Salin untuk menyalin pernyataan SQL yang ditampilkan, atau pilih Unduh untuk mengunduh pernyataan SQL guna menampilkan teks SQL hingga batas mesin DB.

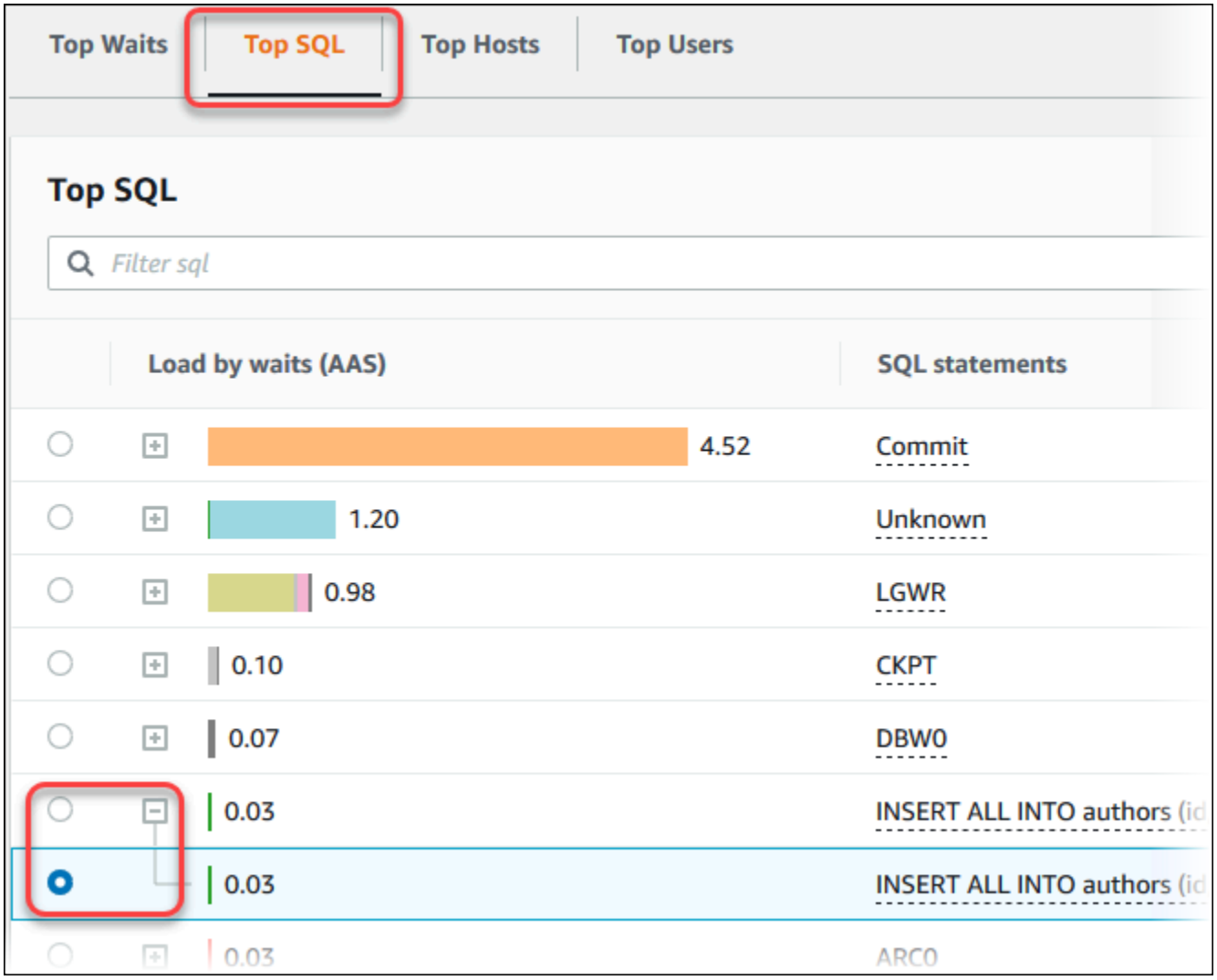
> **Note**
>
> Untuk menyalin atau mengunduh pernyataan SQL, nonaktifkan pemblokir pop-up.

Melihat statistik SQL di dasbor Wawasan Performa

Di dasbor Wawasan Performa, statistik SQL tersedia di tab SQL Teratas pada bagan Muatan basis data.

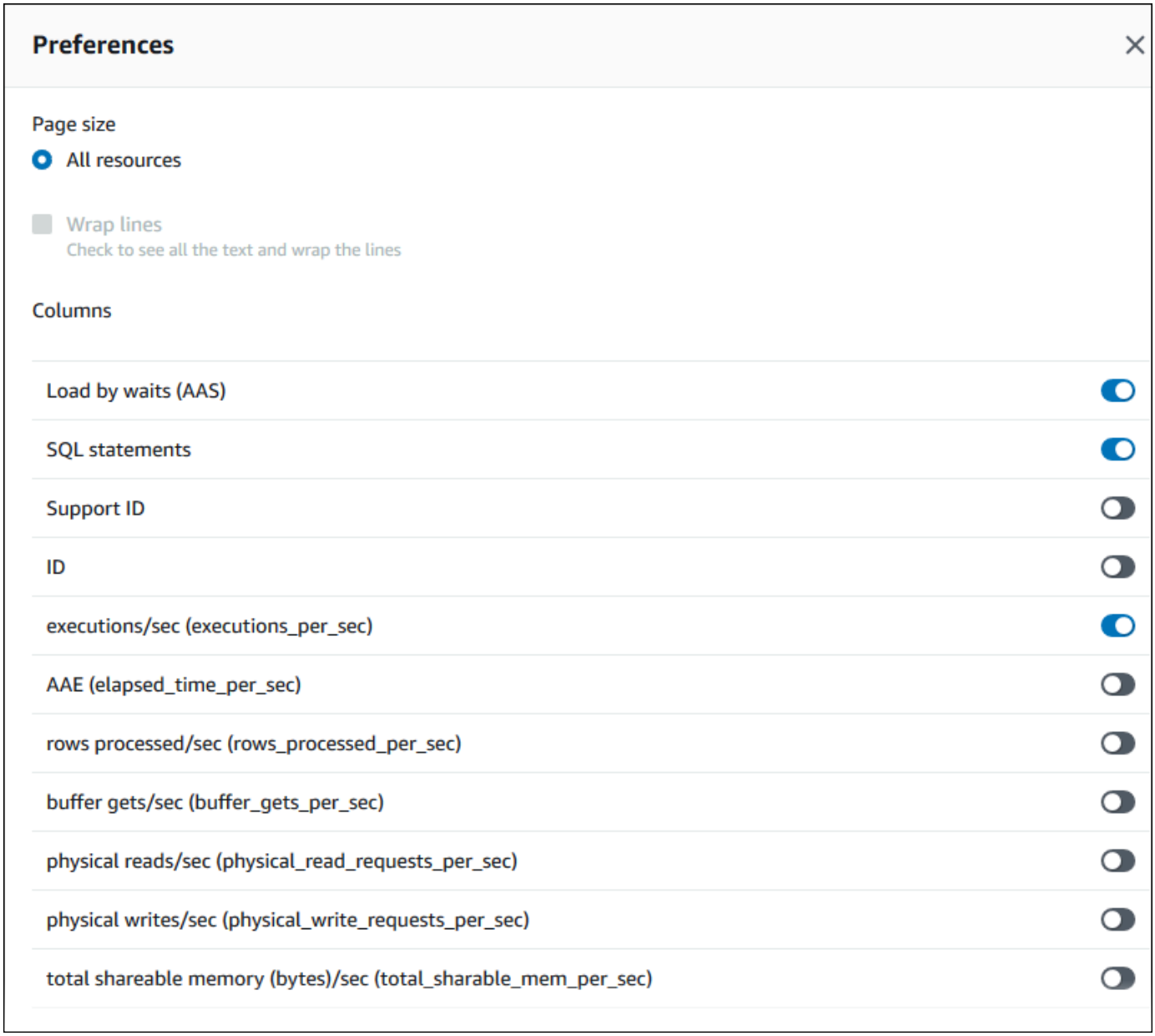
Untuk melihat statistik SQL

1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi kiri, pilih Wawasan Performa.
3. Di bagian atas halaman, pilih basis data yang statistik SQL-nya ingin dilihat.
4. Gulir ke bagian bawah halaman dan pilih tab SQL Teratas.
5. Pilih setiap pernyataan atau kueri digest.

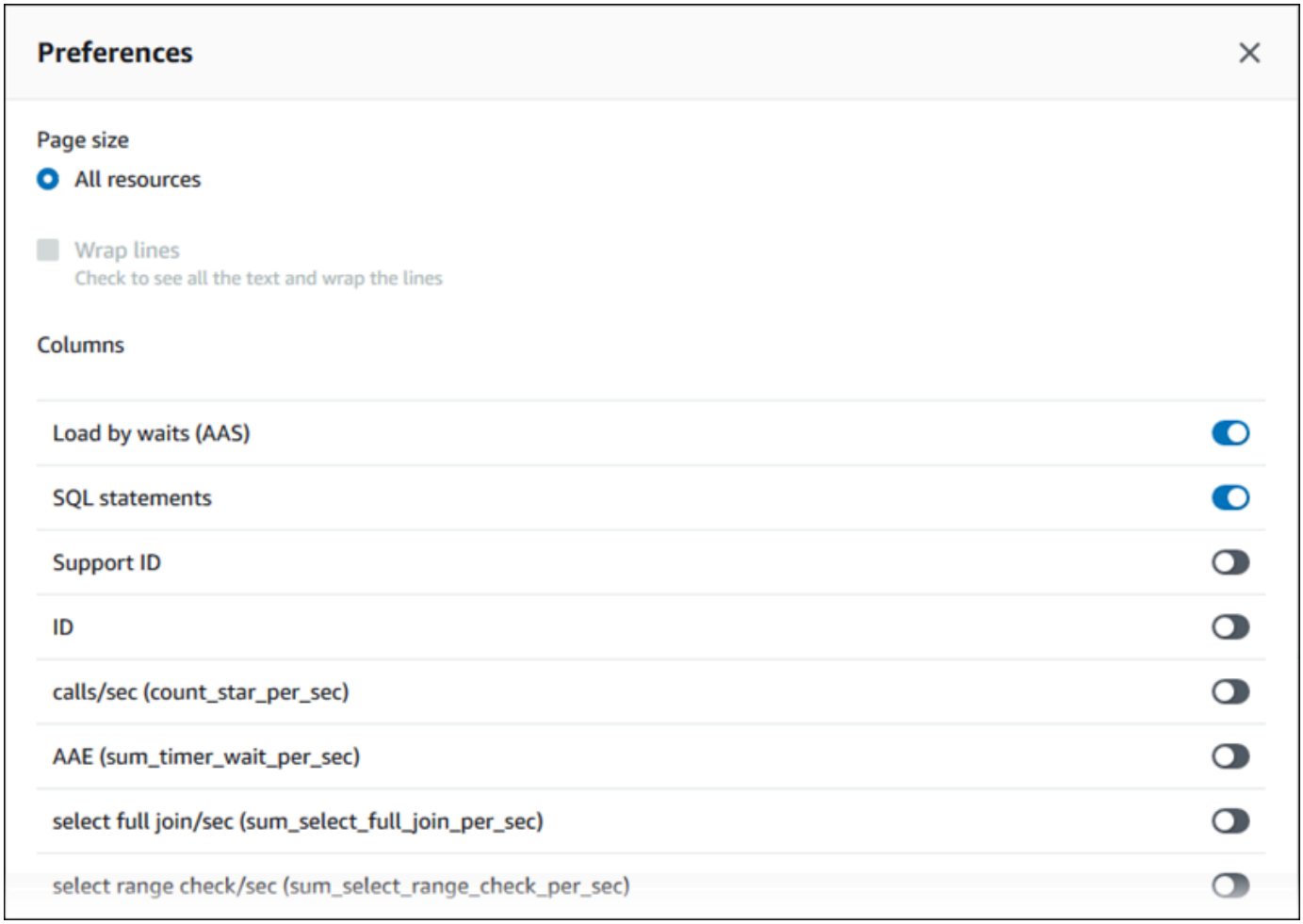


6. Pilih statistik yang akan ditampilkan dengan memilih ikon roda gigi di sudut kanan atas bagan. Untuk deskripsi statistik SQL untuk mesin Amazon RDS, lihat Statistik SQL untuk Wawasan Performa.

   Contoh berikut menunjukkan preferensi statistik untuk instans DB Oracle.



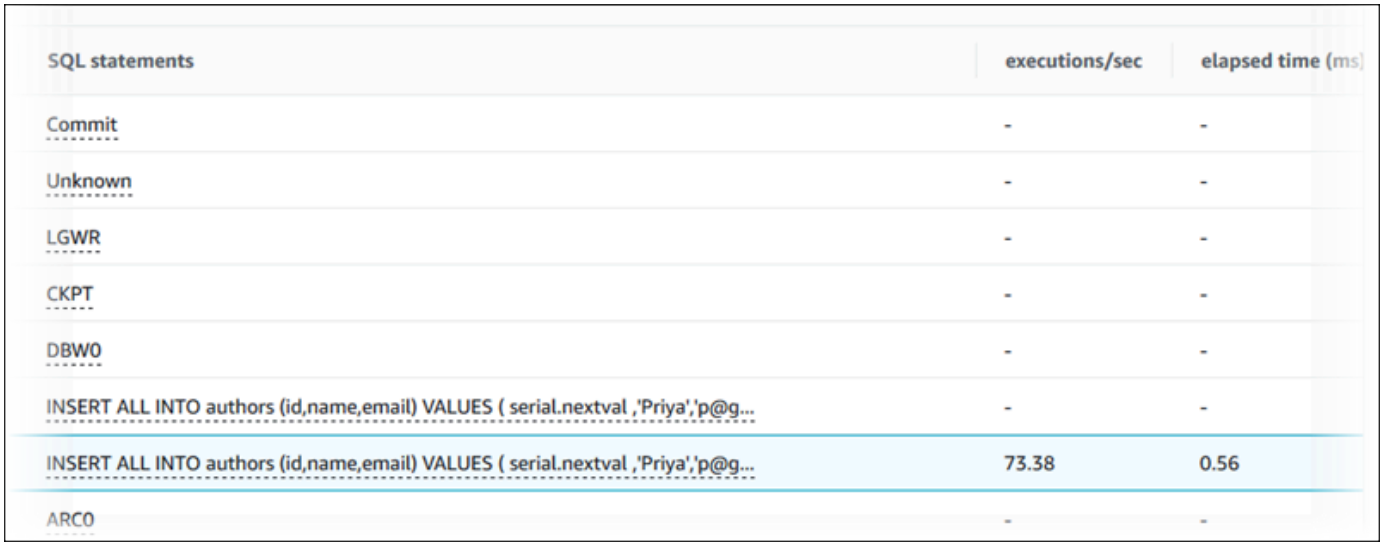
Contoh berikut menunjukkan preferensi untuk instans DB MariaDB dan MySQL.



7. Pilih Simpan untuk menyimpan preferensi Anda.

   Tabel SQL Top dimuat ulang.

   Contoh berikut menunjukkan statistik untuk query Oracle SQL.



SQL statements	executions/sec	elapsed time (ms)
Commit	-	-
Unknown	-	-
LGWR	-	-
CKPT	-	-
DBW0	-	-
INSERT ALL INTO authors (id,name,email) VALUES (serial.nextval ,'Priya','p@g...	-	-
INSERT ALL INTO authors (id,name,email) VALUES (serial.nextval ,'Priya','p@g...	73.38	0.56
ARC0	-	-

Menganalisis rencana eksekusi Oracle menggunakan dasbor Wawasan Performa

Saat menganalisis muatan DB di Basis Data Oracle, sebaiknya Anda mengetahui paket yang paling berkontribusi terhadap muatan DB. Misalnya, pernyataan SQL teratas pada waktu tertentu mungkin menggunakan rencana yang ditunjukkan pada tabel berikut.

SQL Teratas	Rencana
PILIH SUM(amount_sold) DARI penjualan DI MANA prod_id = 10	Rencana A
PILIH SUM(amount_sold) DARI penjualan DI MANA prod_id = 521	Rencana B
PILIH SUM(s_total) DARI penjualan DI MANA region = 10	Rencana A
PILIH * DARI emp DI MANA emp_id = 1000	Rencana C
PILIH SUM(amount_sold) DARI penjualan DI MANA prod_id = 72	Rencana A

Dengan fitur rencana Wawasan Performa, Anda dapat melakukan tindakan berikut:

- Cari tahu paket yang digunakan oleh kueri SQL teratas.

Misalnya, Anda mungkin mengetahui bahwa sebagian besar muatan DB dihasilkan oleh kueri yang menggunakan rencana A dan rencana B, dengan hanya sebagian kecil yang menggunakan paket C.

- Bandingkan rencana yang berbeda untuk kueri yang sama.

  Dalam contoh sebelumnya, tiga kueri identik kecuali untuk ID produk. Dua kueri menggunakan rencana A, tetapi satu kueri menggunakan rencana B. Untuk melihat perbedaan dalam dua rencana tersebut, Anda dapat menggunakan Wawasan Performa.

- Cari tahu kapan kueri beralih ke rencana baru.

  Anda mungkin melihat bahwa kueri menggunakan rencana A, lalu beralih ke rencana B pada waktu tertentu. Apakah ada perubahan dalam basis data pada saat ini? Misalnya, jika tabel kosong, pengoptimal mungkin memilih pemindaian tabel lengkap. Jika tabel dimuat dengan satu juta baris, pengoptimal mungkin beralih ke pemindaian rentang indeks.

- Telusuri langkah-langkah rencana spesifik dengan biaya tertinggi.

  Misalnya, kueri yang berjalan lama mungkin menunjukkan kondisi gabungan yang hilang dalam equijoin. Kondisi yang hilang ini memaksa penggabungan Cartesian, yang menggabungkan semua baris dari dua tabel.

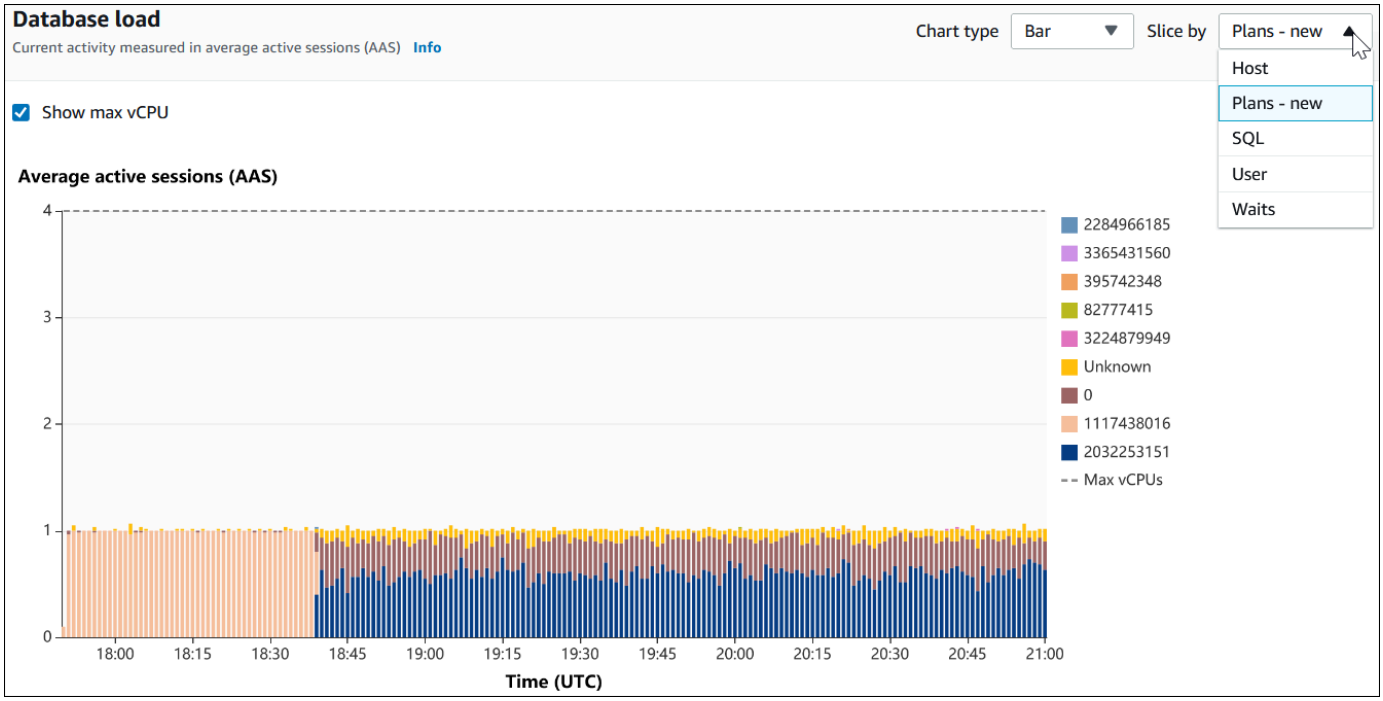
Anda dapat melakukan tugas sebelumnya dengan menggunakan fitur pengambilan rencana dari Wawasan Performa. Seperti halnya mengiris kueri Oracle berdasarkan periwtiwa tunggu dan SQL teratas, Anda dapat mengirisnya berdasarkan dimensi rencana.

Untuk informasi dukungan wilayah, mesin DB, dan kelas instans untuk fitur ini, lihat Dukungan kelas instans, Wilayah, dan mesin DB Amazon RDS untuk fitur Wawasan Performa

Untuk menganalisis rencana eksekusi Oracle menggunakan konsol

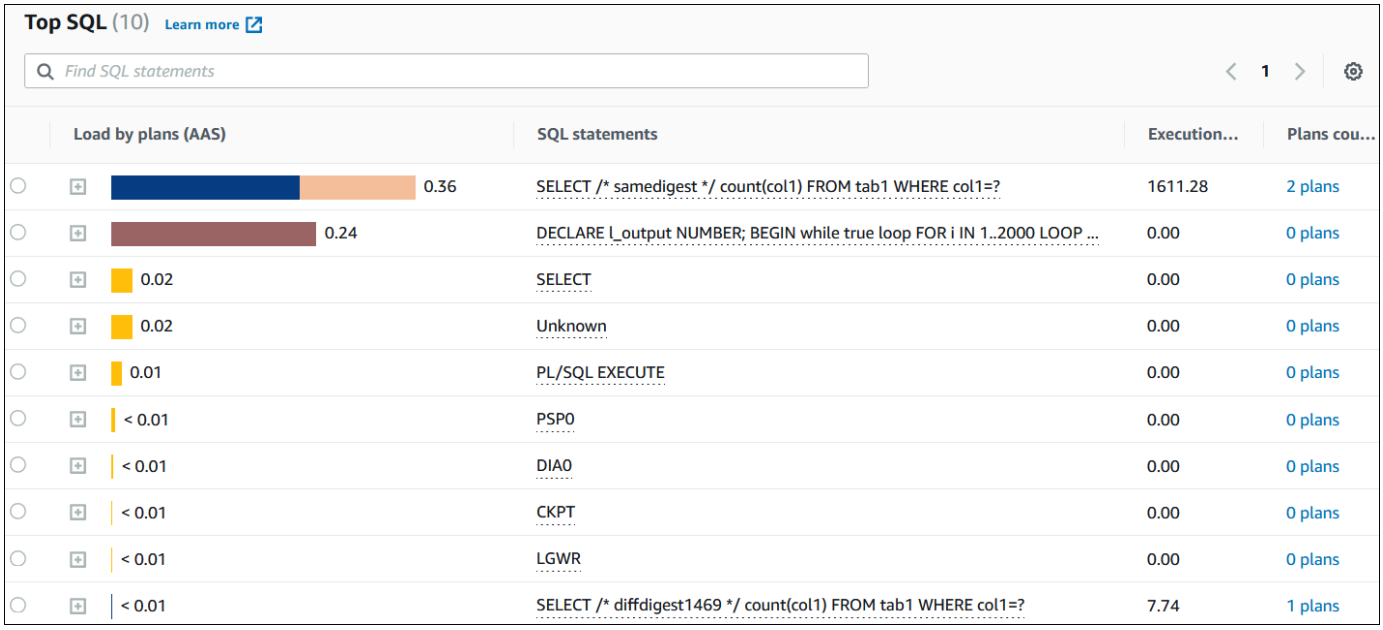
1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Wawasan Performa.
3. Pilih instans DB Oracle. Dasbor Wawasan Performa ditampilkan untuk instans DB tersebut.
4. Di bagian Muatan basis data (muatan DB), pilih Rencana di sebelah Potong menurut.

   Bagan Sesi aktif rata-rata menunjukkan rencana yang digunakan oleh pernyataan SQL teratas. Nilai hash rencana muncul di sebelah kanan kotak kode warna. Setiap nilai hash secara unik mengidentifikasi rencana.



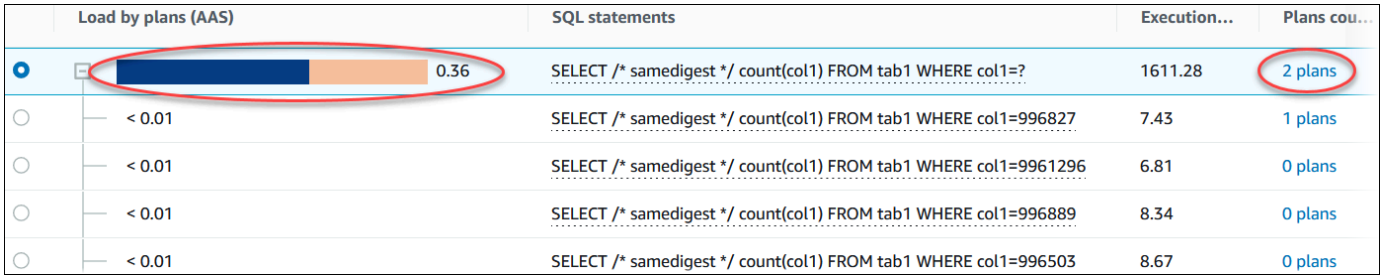
5. Gulir ke bawah ke tab SQL Teratas.

   Dalam contoh berikut, digest SQL teratas memiliki dua rencana. Anda dapat beranggapan bahwa ini adalah digest dengan tanda tanya dalam pernyataan.



6. Pilih digest untuk meluaskannya dalam pernyataan komponennya.

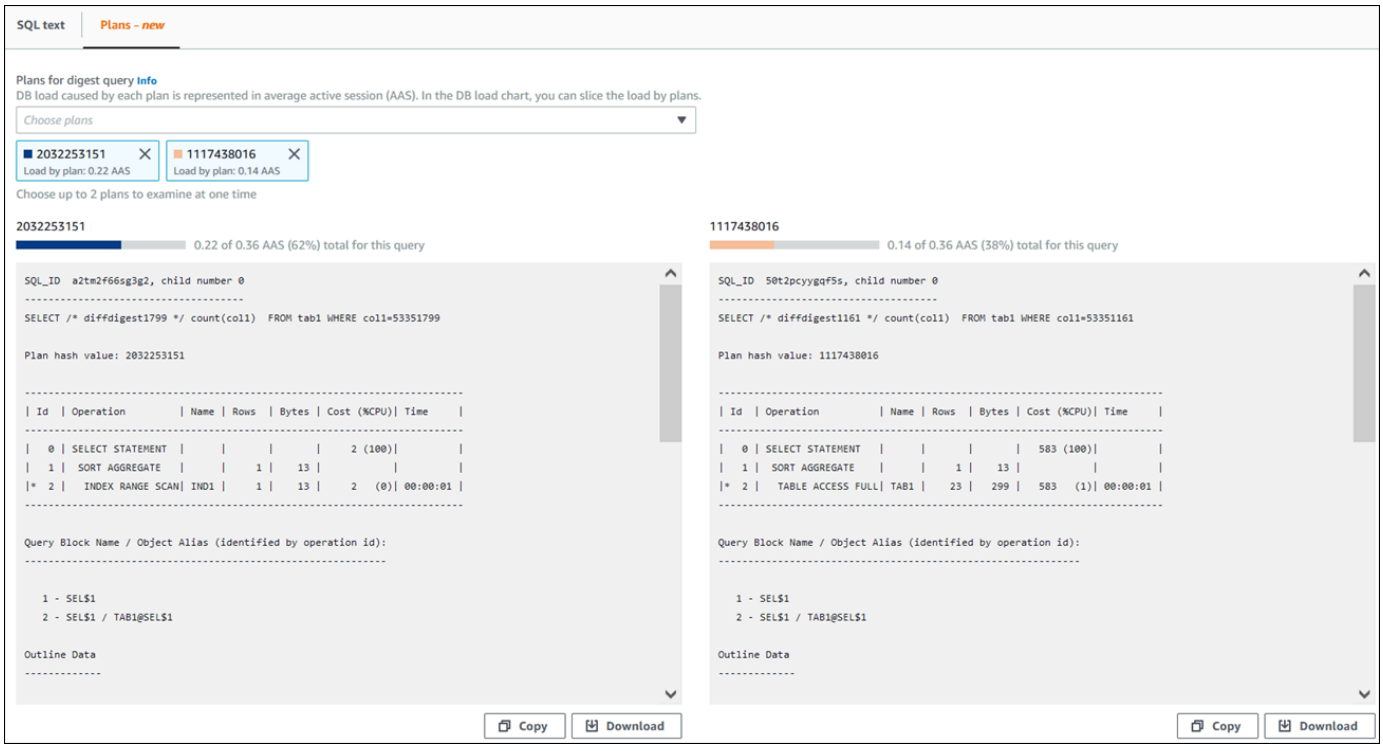
Dalam contoh berikut, pernyataan SELECT merupakan kueri digest. Kueri komponen dalam digest menggunakan dua rencana yang berbeda. Warna rencana sesuai dengan bagan muatan basis data. Jumlah total rencana dalam digest ditampilkan di kolom kedua.



Load by plans (AAS)	SQL statements	Execution...	Plans cou...
0.36	SELECT /* samedigest */ count(col1) FROM tab1 WHERE col1=?	1611.28	2 plans
< 0.01	SELECT /* samedigest */ count(col1) FROM tab1 WHERE col1=996827	7.43	1 plans
< 0.01	SELECT /* samedigest */ count(col1) FROM tab1 WHERE col1=9961296	6.81	0 plans
< 0.01	SELECT /* samedigest */ count(col1) FROM tab1 WHERE col1=996889	8.34	0 plans
< 0.01	SELECT /* samedigest */ count(col1) FROM tab1 WHERE col1=996503	8.67	0 plans

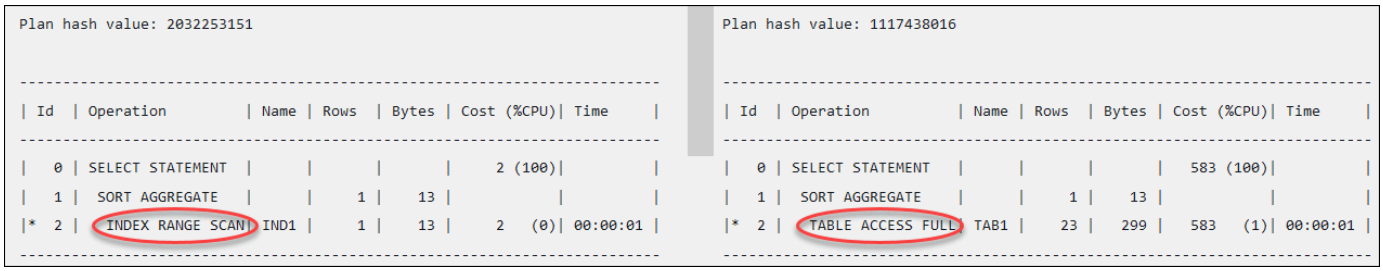
7. Gulir ke bawah dan pilih dua Rencana untuk dibandingkan dari daftar Rencana untuk kueri digest.

Anda dapat melihat salah satu atau dua rencana untuk kueri sekaligus. Tangkapan layar berikut membandingkan dua rencana dalam digest, dengan hash 2032253151 dan hash 1117438016. Dalam contoh berikut, 62% dari sesi aktif rata-rata yang menjalankan kueri digest ini menggunakan rencana di sebelah kiri, sedangkan 38% menggunakan rencana di sebelah kanan.



Dalam contoh ini, rencana ini sangat berbeda. Langkah 2 dalam rencana 2032253151 menggunakan pemindaian indeks, sedangkan rencana 1117438016 menggunakan pemindaian

tabel lengkap. Untuk tabel dengan banyak baris, kueri satu baris hampir selalu lebih cepat dengan pemindaian indeks.



```
Plan hash value: 2032253151

-------------------------------------------------------------------------------
| Id  | Operation         | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT  |      |       |       |     2 (100)|          |
|   1 |  SORT AGGREGATE   |      |     1 |    13 |            |          |
|*  2 |   INDEX RANGE SCAN| IND1 |     1 |    13 |     2   (0)| 00:00:01 |
-------------------------------------------------------------------------------
```

```
Plan hash value: 1117438016

--------------------------------------------------------------------------------
| Id  | Operation          | Name | Rows  | Bytes | Cost (%CPU)| Time     |
--------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |      |       |       |   583 (100)|          |
|   1 |  SORT AGGREGATE    |      |     1 |    13 |            |          |
|*  2 |   TABLE ACCESS FULL| TAB1 |    23 |   299 |   583   (1)| 00:00:01 |
--------------------------------------------------------------------------------
```

8. (Opsional) Pilih Salin untuk menyalin rencana ke papan klip, atau Unduh untuk menyimpan rencana ke hard drive Anda.

Melihat rekomendasi proaktif Performance Insights

Amazon RDS Performance Insights memantau metrik tertentu dan secara otomatis membuat ambang batas dengan menganalisis level apa yang mungkin berpotensi bermasalah untuk sumber daya tertentu. Ketika nilai metrik baru melewati ambang batas yang telah ditentukan selama periode waktu tertentu, Performance Insights menghasilkan rekomendasi proaktif. Rekomendasi ini membantu mencegah dampak kinerja database future. Untuk menerima rekomendasi proaktif ini, Anda harus mengaktifkan Performance Insights dengan periode retensi tingkat berbayar.

Untuk informasi selengkapnya tentang mengaktifkan Wawasan Performa, lihat Mengaktifkan dan menonaktifkan Wawasan Performa. Untuk informasi tentang harga dan retensi data untuk Performance Insights, lihat. Harga dan retensi data untuk Wawasan Performa

Untuk mengetahui wilayah, mesin DB, dan kelas instance yang didukung untuk rekomendasi proaktif, lihatDukungan kelas instans, Wilayah, dan mesin DB Amazon RDS untuk fitur Wawasan Performa.

Anda dapat melihat analisis terperinci dan investigasi rekomendasi proaktif yang direkomendasikan di halaman detail rekomendasi.

Untuk informasi lebih lanjut tentang rekomendasi, lihatMelihat dan menanggapi rekomendasi Amazon Aurora RDS.

Untuk melihat analisis rinci dari rekomendasi proaktif

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, lakukan salah satu hal berikut:

- Pilih Rekomendasi.

  Halaman Rekomendasi menampilkan daftar rekomendasi yang diurutkan berdasarkan tingkat keparahan semua sumber daya di akun Anda.

- Pilih Database dan kemudian pilih Rekomendasi untuk sumber daya di halaman database.

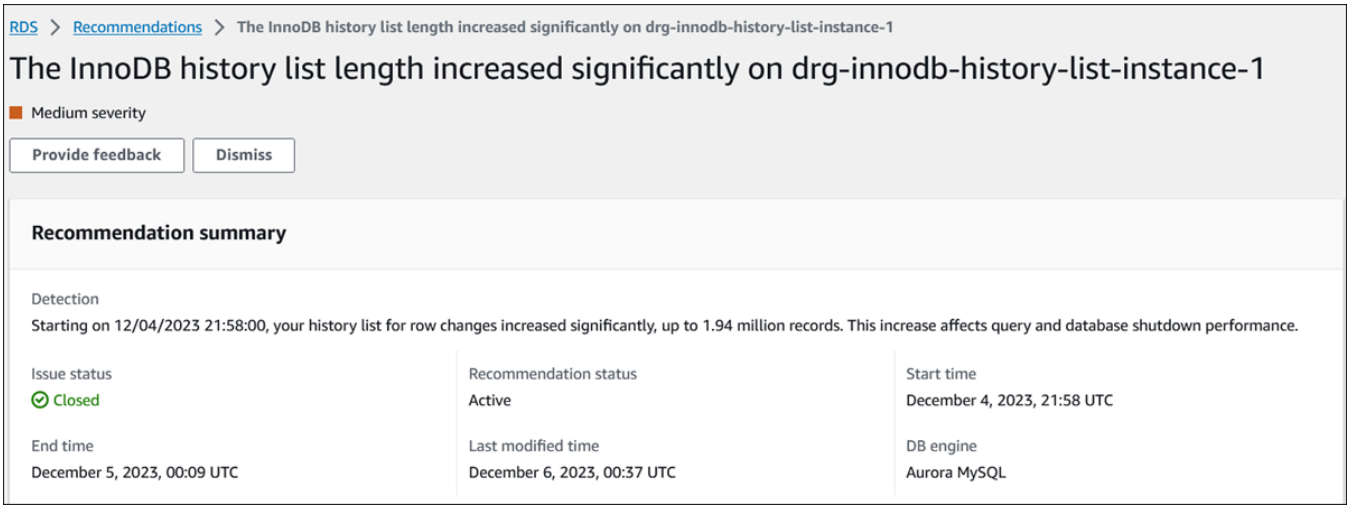
  Tab Rekomendasi menampilkan rekomendasi dan detailnya untuk sumber daya yang dipilih.

3. Temukan rekomendasi proaktif dan pilih Lihat detail.

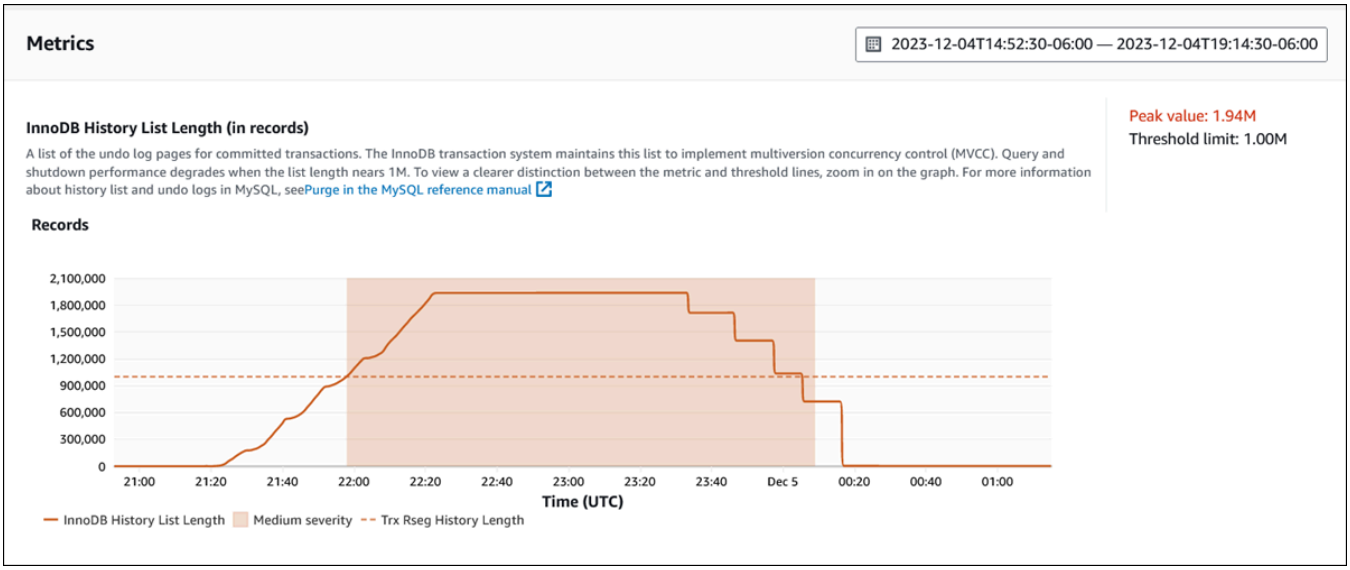
   Halaman detail rekomendasi muncul. Judul memberikan nama sumber daya yang terpengaruh dengan masalah yang terdeteksi dan tingkat keparahannya.

   Berikut ini adalah komponen pada halaman detail rekomendasi:

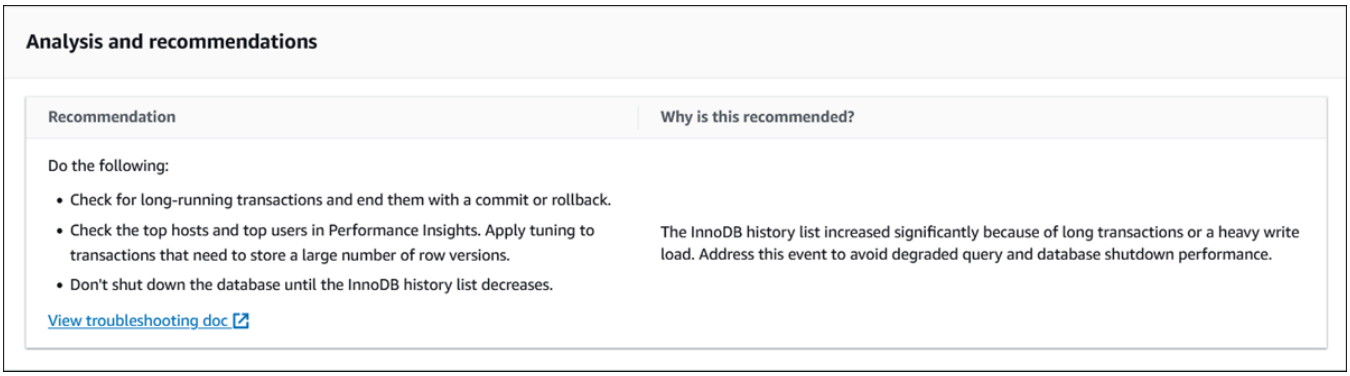
   - Ringkasan rekomendasi — Masalah yang terdeteksi, status rekomendasi dan masalah, waktu mulai dan berakhir masalah, waktu modifikasi rekomendasi, dan jenis mesin.



   - Metrik — Grafik masalah yang terdeteksi. Setiap grafik menampilkan ambang batas yang ditentukan oleh perilaku dasar sumber daya dan data metrik yang dilaporkan dari waktu mulai masalah.



- Analisis dan rekomendasi — Rekomendasi dan alasan rekomendasi yang disarankan.



Anda dapat meninjau penyebab masalah dan kemudian melakukan tindakan yang disarankan untuk memperbaiki masalah, atau memilih Singkirkan di kanan atas untuk mengabaikan rekomendasi.

Mengambil metrik dengan API Wawasan Performa

Saat Wawasan Performa diaktifkan, API menyediakan visibilitas tentang performa instans. Amazon CloudWatch Logs menyediakan sumber otoritatif untuk metrik pemantauan terjual untuk layanan. AWS

Wawasan Performa menawarkan tampilan spesifik domain dari muatan basis data yang diukur sebagai sesi aktif rata-rata (AAS). Metrik ini muncul sebagai set data deret waktu dua dimensi bagi konsumen API. Dimensi waktu data menyediakan data muatan DB untuk setiap titik waktu dalam rentang waktu yang dikueri. Setiap titik waktu menguraikan keseluruhan muatan dalam hubungannya

dengan dimensi yang diminta, seperti `SQL`, `Wait-event`, `User`, atau `Host`, yang diukur pada titik waktu tersebut.

Wawasan Performa Amazon RDS memantau instans DB Amazon RDS Anda sehingga Anda dapat menganalisis dan memecahkan masalah performa basis data. Salah satu cara untuk menampilkan data Wawasan Performa dapat ditemukan di AWS Management Console. Wawasan Performa juga menyediakan API publik sehingga Anda dapat mengkueri data Anda sendiri. Anda dapat menggunakan API untuk melakukan tindakan berikut:

- Membongkar data ke dalam basis data
- Menambahkan data Wawasan Performa ke dasbor pemantauan yang ada
- Merancang alat pemantauan

Untuk menggunakan API Wawasan Performa, aktifkan Wawasan Performa di salah satu instans DB Amazon RDS Anda. Untuk informasi tentang cara mengaktifkan Wawasan Performa, lihat Mengaktifkan dan menonaktifkan Wawasan Performa. Untuk informasi selengkapnya tentang API Wawasan Performa, lihat Referensi API Wawasan Performa Amazon RDS.

API Wawasan Performa menyediakan operasi berikut.

Tindakan Wawasan Performa	Perintah AWS CLI	Deskripsi
`CreatePerformanceAnalysisReport`	`aws pi create-performance-analysis-report`	Membuat laporan analisis performa selama periode waktu tertentu untuk instans DB. Hasilnya adalah `AnalysisReportId` , yakni pengidentifikasi unik laporan.
`DeletePerformanceAnalysisReport`	`aws pi delete-performance-analysis-report`	Menghapus laporan analisis performa.
`DescribeDimensionKeys`	`aws pi describe-dimension-keys`	Mengambil kunci dimensi N teratas untuk metrik selama periode waktu tertentu.

Tindakan Wawasan Performa	Perintah AWS CLI	Deskripsi
GetDimensionKeyDetails	aws pi get-dimension-key-details	Mengambil atribut dari grup dimensi tertentu untuk instans DB atau sumber data. Misalnya, jika Anda menentukan ID SQL, dan jika detail dimensi tersedia, `GetDimensionKeyDetails` akan mengambil teks lengkap dimensi `db.sql.statement` yang terkait dengan ID ini. Operasi ini berguna karena `GetResourceMetrics` dan `DescribeDimensionKeys` tidak mendukung pengambilan teks pernyataan SQL besar.
GetPerformanceAnalysisReport	aws pi get-performance-analysis-report	Mengambil laporan yang mencakup wawasan untuk laporan. Hasilnya mencakup status laporan, ID laporan, detail waktu laporan, wawasan, dan rekomendasi.
GetResourceMetadata	aws pi get-resource-metadata	Mengambil metadata untuk fitur yang berbeda. Misalnya, metadata mungkin menunjukkan bahwa fitur diaktifkan atau dinonaktifkan pada instans DB tertentu.

Tindakan Wawasan Performa	Perintah AWS CLI	Deskripsi
GetResourceMetrics	aws pi get-resource-metrics	Mengambil metrik Wawasan Performa untuk sekumpulan sumber data, selama periode waktu tertentu. Anda dapat menyediakan grup dimensi dan dimensi tertentu, serta memberikan kriteria penggabungan dan pemfilteran untuk setiap grup.
ListAvailableResourceDimensions	aws pi list-available-resource-dimensions	Mengambil dimensi yang dapat dikueri untuk setiap jenis metrik yang ditentukan pada instans tertentu.
ListAvailableResourceMetrics	aws pi list-available-resource-metrics	Mengambil semua metrik jenis metrik tertentu yang tersedia yang dapat dikueri untuk instans DB tertentu.
ListPerformanceAnalysisReports	aws pi list-performance-analysis-reports	Mengambil semua laporan analisis yang tersedia untuk instans DB. Laporan dicantumkan berdasarkan waktu mulai setiap laporan.
ListTagsForResource	aws pi list-tags-for-resource	Daftar semua tag metadata yang ditambahkan ke sumber daya. Daftar ini mencakup nama dan nilai tag.
TagResource	aws pi tag-resource	Menambahkan tag metadata ke sumber daya Amazon RDS. Tag ini termasuk nama dan nilai.

Tindakan Wawasan Performa	Perintah AWS CLI	Deskripsi
UntagResource	aws pi untag-resource	Menghapus tag metadata dari sumber daya.

Topik

- AWS CLI untuk Wawasan Performa
- Mengambil metrik deret waktu
- Contoh AWS CLI untuk Wawasan Performa

AWS CLI untuk Wawasan Performa

Anda dapat melihat data Wawasan Performa menggunakan AWS CLI. Anda dapat menampilkan bantuan untuk perintah AWS CLI untuk Wawasan Performa dengan memasukkan berikut ini di baris perintah.

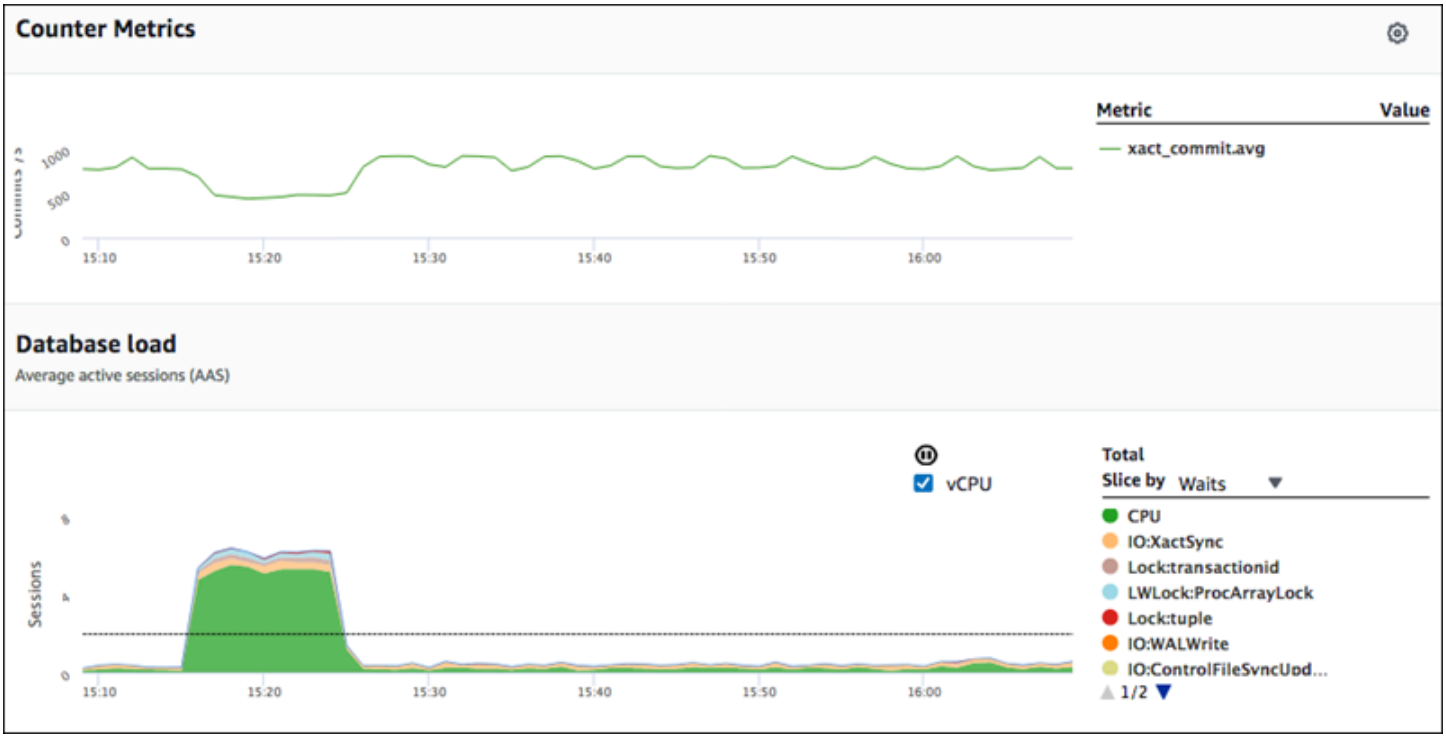
```
aws pi help
```

Jika Anda belum menginstal AWS CLI, lihat Memasang Antarmuka Baris Perintah AWS di Panduan Pengguna AWS CLI untuk informasi tentang cara menginstalnya.

Mengambil metrik deret waktu

Operasi `GetResourceMetrics` mengambil satu atau beberapa metrik deret waktu dari data Wawasan Performa. `GetResourceMetrics` memerlukan metrik dan periode waktu, dan menampilkan respons dengan daftar poin data.

Misalnya, AWS Management Console menggunakan `GetResourceMetrics` untuk mengisi bagan Metrik Penghitung dan bagan Muatan Basis Data seperti yang diperlihatkan pada gambar berikut.



Semua metrik yang ditampilkan oleh `GetResourceMetrics` adalah metrik deret waktu standar, dengan pengecualian `db.load`. Metrik ini ditampilkan dalam diagram Muatan Basis Data. Metrik `db.load` berbeda dengan metrik seri waktu lainnya karena Anda dapat membaginya menjadi beberapa sub-komponen yang disebut dimensi. Di gambar sebelumnya, `db.load` dibagi dan dikelompokkan berdasarkan status tunggu yang membentuk `db.load`.

> **Note**
>
> `GetResourceMetrics` juga dapat menampilkan metrik `db.sampleload`, tetapi metrik `db.load` sesuai di sebagian besar kasus.

Untuk informasi tentang metrik penghitung yang ditampilkan oleh `GetResourceMetrics`, lihat Metrik penghitung Wawasan Performa.

Penghitungan berikut didukung untuk metrik:

- Rata-rata – Nilai rata-rata untuk metrik selama periode waktu tertentu. Tambahkan `.avg` ke nama metrik.
- Minimum – Nilai minimum metrik selama periode waktu tertentu. Tambahkan `.min` ke nama metrik.

- Maksimum – Nilai maksimum metrik selama periode waktu tertentu. Tambahkan `.max` ke nama metrik.
- Jumlah – Jumlah nilai metrik selama periode waktu tertentu. Tambahkan `.sum` ke nama metrik.
- Jumlah sampel – Frekuensi pengumpulan metrik selama periode waktu tertentu. Tambahkan `.sample_count` ke nama metrik.

Sebagai contoh, misalkan sebuah metrik dikumpulkan selama 300 detik (5 menit), dan metrik tersebut dikumpulkan satu kali setiap menit. Nilai untuk setiap menit adalah 1, 2, 3, 4, dan 5. Dalam kasus ini, penghitungan berikut ditampilkan:

- Rata-rata – 3
- Minimum – 1
- Maksimum – 5
- Jumlah – 15
- Jumlah sampel – 5

Untuk informasi tentang cara menggunakan perintah `get-resource-metrics` AWS CLI, lihat `get-resource-metrics`.

Untuk opsi `--metric-queries`, tentukan satu atau beberapa kueri yang diinginkan untuk mendapatkan hasil. Setiap kueri terdiri dari `Metric` wajib dan `GroupBy` opsional serta parameter `Filter`. Berikut ini adalah contoh spesifikasi opsi `--metric-queries`.

```
{
   "Metric": "string",
   "GroupBy": {
     "Group": "string",
     "Dimensions": ["string", ...],
     "Limit": integer
   },
   "Filter": {"string": "string"
     ...}
```

Contoh AWS CLI untuk Wawasan Performa

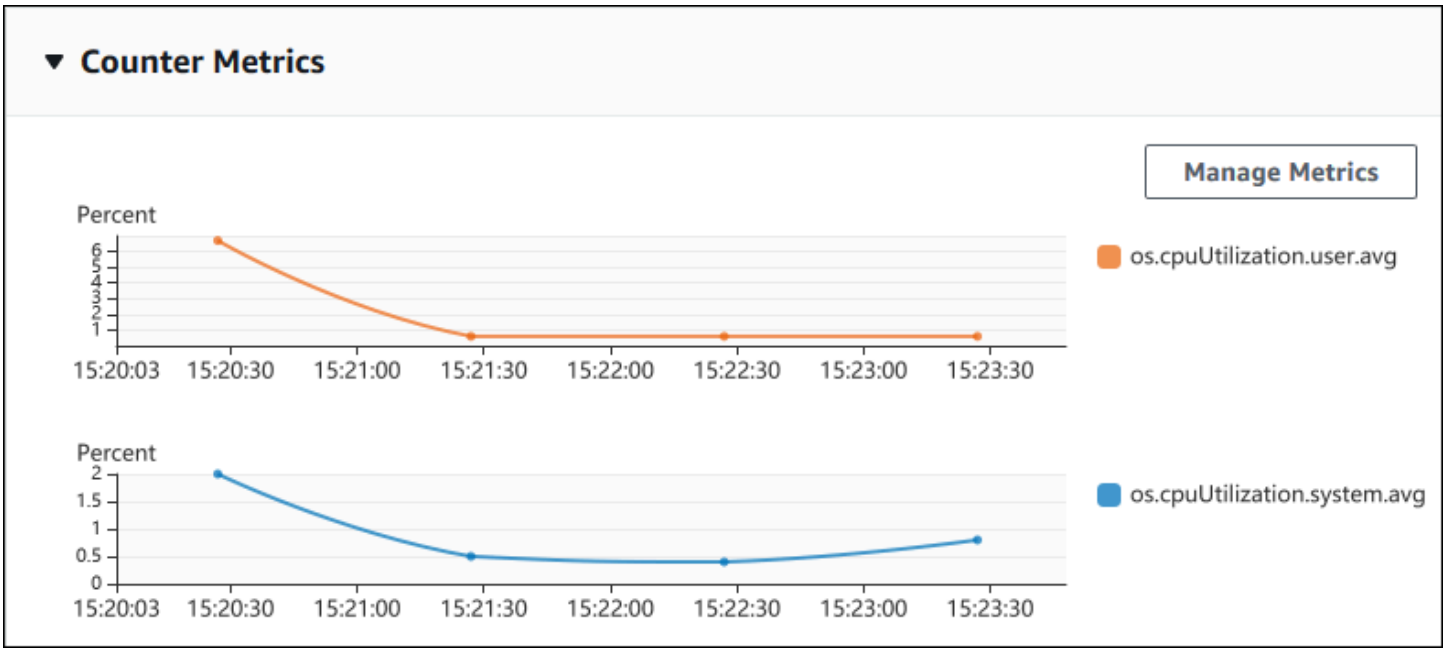
Contoh berikut menunjukkan cara menggunakan AWS CLI untuk Wawasan Performa.

Topik

- Mengambil metrik penghitung
- Mengambil rata-rata muatan DB untuk peristiwa tunggu teratas
- Mengambil rata-rata muatan DB untuk SQL teratas
- Mengambil Rata-Rata Muatan DB yang difilter berdasarkan SQL
- Mengambil teks lengkap pernyataan SQL
- Membuat laporan analisis performa selama periode waktu tertentu
- Mengambil laporan analisis performa
- Daftar semua laporan analisis performa untuk instans DB
- Menghapus laporan analisis performa
- Menambahkan tag ke laporan analisis performa
- Mencantumkan semua tag untuk laporan analisis performa
- Menghapus tag dari laporan analisis performa

Mengambil metrik penghitung

Tangkapan layar berikut menunjukkan dua bagan metrik penghitung dalam AWS Management Console.



Contoh berikut menunjukkan cara mengumpulkan data yang sama yang digunakan oleh AWS Management Console untuk menghasilkan dua bagan metrik penghitung.

Untuk Linux, macOS, atau Unix:

```
aws pi get-resource-metrics \
   --service-type RDS \
   --identifier db-ID \
   --start-time 2018-10-30T00:00:00Z \
   --end-time   2018-10-30T01:00:00Z \
   --period-in-seconds 60 \
   --metric-queries '[{"Metric": "os.cpuUtilization.user.avg"  },
                      {"Metric": "os.cpuUtilization.idle.avg"}]'
```

Untuk Windows:

```
aws pi get-resource-metrics ^
   --service-type RDS ^
   --identifier db-ID ^
   --start-time 2018-10-30T00:00:00Z ^
   --end-time   2018-10-30T01:00:00Z ^
   --period-in-seconds 60 ^
   --metric-queries '[{"Metric": "os.cpuUtilization.user.avg"  },
                      {"Metric": "os.cpuUtilization.idle.avg"}]'
```

Anda juga dapat membaca perintah dengan lebih mudah dengan menentukan file untuk opsi `--metrics-query`. Contoh berikut menggunakan file yang disebut query.json untuk opsi tersebut. File memiliki konten berikut.

```
[
    {
        "Metric": "os.cpuUtilization.user.avg"
    },
    {
        "Metric": "os.cpuUtilization.idle.avg"
    }
]
```

Jalankan perintah berikut untuk menggunakan file.

Untuk Linux, macOS, atau Unix:

```
aws pi get-resource-metrics \
   --service-type RDS \
   --identifier db-ID \
```

```
    --start-time 2018-10-30T00:00:00Z \
    --end-time   2018-10-30T01:00:00Z \
    --period-in-seconds 60 \
    --metric-queries file://query.json
```

Untuk Windows:

```
aws pi get-resource-metrics ^
    --service-type RDS ^
    --identifier db-ID ^
    --start-time 2018-10-30T00:00:00Z ^
    --end-time   2018-10-30T01:00:00Z ^
    --period-in-seconds 60 ^
    --metric-queries file://query.json
```

Contoh sebelumnya menentukan nilai-nilai berikut untuk opsi:

- `--service-type` – RDS untuk Amazon RDS
- `--identifier` – ID sumber daya untuk instans DB
- `--start-time` dan `--end-time` – Nilai `DateTime` ISO 8601 untuk periode kueri, dengan berbagai format yang didukung

Ini akan dikueri selama rentang waktu satu jam:

- `--period-in-seconds` – `60` untuk kueri per menit
- `--metric-queries` – Rangkaian dua kueri, masing-masing hanya untuk satu metrik.

  Nama metrik menggunakan titik untuk mengklasifikasikan metrik dalam kategori yang berguna, dengan elemen terakhir sebagai fungsi. Dalam contoh, fungsinya adalah `avg` untuk setiap kueri. Seperti halnya Amazon CloudWatch, fungsi yang didukung adalah `min`,`max`,`total`, dan`avg`.

Responsnya terlihat seperti berikut.

```
{
    "Identifier": "db-XXX",
    "AlignedStartTime": 1540857600.0,
    "AlignedEndTime": 1540861200.0,
    "MetricList": [
        { //A list of key/datapoints
```

```
            "Key": {
                "Metric": "os.cpuUtilization.user.avg" //Metric1
            },
            "DataPoints": [
                //Each list of datapoints has the same timestamps and same number of
 items
                {
                    "Timestamp": 1540857660.0, //Minute1
                    "Value": 4.0
                },
                {
                    "Timestamp": 1540857720.0, //Minute2
                    "Value": 4.0
                },
                {
                    "Timestamp": 1540857780.0, //Minute 3
                    "Value": 10.0
                }
                //... 60 datapoints for the os.cpuUtilization.user.avg metric
            ]
        },
        {
            "Key": {
                "Metric": "os.cpuUtilization.idle.avg" //Metric2
            },
            "DataPoints": [
                {
                    "Timestamp": 1540857660.0, //Minute1
                    "Value": 12.0
                },
                {
                    "Timestamp": 1540857720.0, //Minute2
                    "Value": 13.5
                },
                //... 60 datapoints for the os.cpuUtilization.idle.avg metric
            ]
        }
    ] //end of MetricList
} //end of response
```

Respons ini memiliki `Identifier`, `AlignedStartTime`, dan `AlignedEndTime`. B nilai `--period-in-seconds` adalah `60`, waktu mulai dan akhir telah disesuaikan dengan menit. Jika `--period-in-seconds` adalah `3600`, waktu mulai dan akhir akan disesuaikan dengan jam.

`MetricList` dalam respons memiliki sejumlah entri, masing-masing dengan entri `Key` dan `DataPoints`. Masing-masing `DataPoint` memiliki `Timestamp` dan `Value`. Masing-masing daftar `Datapoints` memiliki 60 poin data karena kueri tersebut adalah untuk data per menit selama satu jam, dengan `Timestamp1/Minute1`, `Timestamp2/Minute2`, dan seterusnya, hingga `Timestamp60/Minute60`.

Karena kueri tersebut adalah untuk dua metrik penghitung yang berbeda, ada dua elemen dalam respons `MetricList`.

Mengambil rata-rata muatan DB untuk peristiwa tunggu teratas

Contoh berikut adalah kueri yang sama yang digunakan oleh AWS Management Console untuk menghasilkan grafik baris area bertumpuk. Contoh ini mengambil `db.load.avg` selama satu jam terakhir dengan muatan yang dibagi berdasarkan tujuh peristiwa tunggu teratas. Perintah ini sama dengan perintah dalam Mengambil metrik penghitung. Namun, file query.json berisi konten berikut.

```
[
    {
        "Metric": "db.load.avg",
        "GroupBy": { "Group": "db.wait_event", "Limit": 7 }
    }
]
```

Jalankan perintah berikut.

Untuk Linux, macOS, atau Unix:

```
aws pi get-resource-metrics \
   --service-type RDS \
   --identifier db-ID \
   --start-time 2018-10-30T00:00:00Z \
   --end-time   2018-10-30T01:00:00Z \
   --period-in-seconds 60 \
   --metric-queries file://query.json
```

Untuk Windows:

```
aws pi get-resource-metrics ^
   --service-type RDS ^
   --identifier db-ID ^
   --start-time 2018-10-30T00:00:00Z ^
   --end-time   2018-10-30T01:00:00Z ^
```

```
    --period-in-seconds 60 ^
    --metric-queries file://query.json
```

Contoh ini menentukan metrik `db.load.avg` dan `GroupBy` dari tujuh peristiwa tunggu teratas. Untuk detail tentang nilai yang valid untuk contoh ini, lihat DimensionGroup dalam Referensi API Wawasan Kinerja.

Responsnya terlihat seperti berikut.

```
{
    "Identifier": "db-XXX",
    "AlignedStartTime": 1540857600.0,
    "AlignedEndTime": 1540861200.0,
    "MetricList": [
        { //A list of key/datapoints
            "Key": {
                //A Metric with no dimensions. This is the total db.load.avg
                "Metric": "db.load.avg"
            },
            "DataPoints": [
                //Each list of datapoints has the same timestamps and same number of
 items
                {
                    "Timestamp": 1540857660.0, //Minute1
                    "Value": 0.5166666666666667
                },
                {
                    "Timestamp": 1540857720.0, //Minute2
                    "Value": 0.38333333333333336
                },
                {
                    "Timestamp": 1540857780.0, //Minute 3
                    "Value": 0.26666666666666666
                }
                //... 60 datapoints for the total db.load.avg key
            ]
        },
        {
            "Key": {
                //Another key. This is db.load.avg broken down by CPU
                "Metric": "db.load.avg",
                "Dimensions": {
                    "db.wait_event.name": "CPU",
```

```
                    "db.wait_event.type": "CPU"
                }
            },
            "DataPoints": [
                {
                    "Timestamp": 1540857660.0, //Minute1
                    "Value": 0.35
                },
                {
                    "Timestamp": 1540857720.0, //Minute2
                    "Value": 0.15
                },
                //... 60 datapoints for the CPU key
            ]
        },
        //... In total we have 8 key/datapoints entries, 1) total, 2-8) Top Wait Events
    ] //end of MetricList
} //end of response
```

Dalam respon ini, ada delapan entri dalam `MetricList`. Ada satu entri untuk total `db.load.avg`, dan tujuh entri masing-masing untuk `db.load.avg` yang dibagi berdasarkan salah satu dari tujuh peristiwa tunggu teratas. Tidak seperti di contoh pertama, karena ada dimensi pengelompokan, pasti ada satu kunci untuk setiap pengelompokan metrik. Tidak boleh hanya ada satu kunci untuk setiap metrik, seperti dalam kasus penggunaan metrik penghitung dasar.

Mengambil rata-rata muatan DB untuk SQL teratas

Contoh berikut mengelompokkan `db.wait_events` berdasarkan 10 pernyataan SQL teratas. Ada dua grup berbeda untuk pernyataan SQL:

- `db.sql` – Pernyataan SQL lengkap, seperti `select * from customers where customer_id = 123`
- `db.sql_tokenized` – Pernyataan SQL token, seperti `select * from customers where customer_id = ?`

Saat menganalisis performa basis data, sebaiknya pertimbangkan pernyataan SQL yang hanya berbeda dari segi parameternya sebagai satu item logika. Jadi, Anda dapat menggunakan `db.sql_tokenized` saat melakukan kueri. Namun, terutama ketika Anda tertarik untuk menjelaskan rencana, terkadang lebih berguna untuk memeriksa pernyataan SQL lengkap beserta

parameter, dan pengelompokan kueri berdasarkan `db.sql`. Ada hubungan induk-turunan antara SQL token dan lengkap, dengan beberapa SQL lengkap (turunan) yang dikelompokkan dalam SQL token (induk) yang sama.

Perintah dalam contoh ini terlihat seperti perintah dalam Mengambil rata-rata muatan DB untuk peristiwa tunggu teratas. Namun, file query.json berisi konten berikut.

```
[
    {
        "Metric": "db.load.avg",
        "GroupBy": { "Group": "db.sql_tokenized", "Limit": 10 }
    }
]
```

Contoh berikut menggunakan `db.sql_tokenized`.

Untuk Linux, macOS, atau Unix:

```
aws pi get-resource-metrics \
   --service-type RDS \
   --identifier db-ID \
   --start-time 2018-10-29T00:00:00Z \
   --end-time   2018-10-30T00:00:00Z \
   --period-in-seconds 3600 \
   --metric-queries file://query.json
```

Untuk Windows:

```
aws pi get-resource-metrics ^
   --service-type RDS ^
   --identifier db-ID ^
   --start-time 2018-10-29T00:00:00Z ^
   --end-time   2018-10-30T00:00:00Z  ^
   --period-in-seconds 3600 ^
   --metric-queries file://query.json
```

Contoh ini menanyakan lebih dari 24 jam, dengan satu jam period-in-seconds.

Contoh ini menentukan metrik `db.load.avg` dan `GroupBy` dari tujuh peristiwa tunggu teratas. Untuk detail tentang nilai yang valid untuk contoh ini, lihat DimensionGroup dalam Referensi API Wawasan Kinerja.

Responsnya terlihat seperti berikut.

```
{
    "AlignedStartTime": 1540771200.0,
    "AlignedEndTime": 1540857600.0,
    "Identifier": "db-XXX",

    "MetricList": [ //11 entries in the MetricList
        {
            "Key": { //First key is total
                "Metric": "db.load.avg"
            }
            "DataPoints": [ //Each DataPoints list has 24 per-hour Timestamps and a
 value
                {
                    "Value": 1.6964980544747081,
                    "Timestamp": 1540774800.0
                },
                //... 24 datapoints
            ]
        },
        {
            "Key": { //Next key is the top tokenized SQL
                "Dimensions": {
                    "db.sql_tokenized.statement": "INSERT INTO authors (id,name,email)
 VALUES\n( nextval(?)  ,?,?)",
                    "db.sql_tokenized.db_id": "pi-2372568224",
                    "db.sql_tokenized.id": "AKIAIOSFODNN7EXAMPLE"
                },
                "Metric": "db.load.avg"
            },
            "DataPoints": [ //... 24 datapoints
            ]
        },
        // In total 11 entries, 10 Keys of top tokenized SQL, 1 total key
    ] //End of MetricList
} //End of response
```

Respons ini memiliki 11 entri di `MetricList` (1 total, 10 SQL token teratas), dengan setiap entri memiliki 24 `DataPoints` per jam.

Untuk SQL token, ada tiga entri di setiap daftar dimensi:

- `db.sql_tokenized.statement` – Pernyataan SQL token.
- `db.sql_tokenized.db_id` – Baik ID basis data native yang digunakan untuk merujuk ke SQL, maupun ID sintetis yang dihasilkan oleh Wawasan Performa untuk Anda jika ID basis data native tidak tersedia. Contoh ini menampilkan ID sintetis `pi-2372568224`.
- `db.sql_tokenized.id` – ID kueri di dalam Wawasan Performa.

  Di AWS Management Console, ID ini disebut sebagai ID Dukungan. Disebut demikian karena ID ini adalah data yang dapat diperiksa oleh Dukungan AWS untuk membantu memecahkan masalah dalam basis data Anda. AWS menangani keamanan dan privasi data Anda dengan sangat serius, dan hampir semua data disimpan terenkripsi dengan kunci utama pelanggan (CMK) AWS KMS. Oleh karena itu, tidak ada orang di dalam AWS yang dapat melihat data ini. Di contoh sebelumnya, baik `tokenized.statement` maupun `tokenized.db_id` disimpan dengan enkripsi. Jika Anda mengalami masalah terkait basis data, Dukungan AWS dapat membantu Anda dengan merujuk ID Dukungan.

Ketika melakukan kueri, mungkin lebih mudah untuk menentukan `Group` dalam `GroupBy`. Namun, untuk kontrol lebih mendetail atas data yang ditampilkan, tentukan daftar dimensi. Misalnya, jika yang dibutuhkan hanya `db.sql_tokenized.statement`, atribut `Dimensions` dapat ditambahkan ke file query.json.

```
[
    {
        "Metric": "db.load.avg",
        "GroupBy": {
            "Group": "db.sql_tokenized",
            "Dimensions":["db.sql_tokenized.statement"],
            "Limit": 10
        }
    }
]
```

Mengambil Rata-Rata Muatan DB yang difilter berdasarkan SQL

Gambar sebelumnya menunjukkan bahwa kueri tertentu dipilih, dan grafik baris area bertumpuk sesi aktif rata-rata teratas dicakup ke kueri tersebut. Meskipun kueri masih diperuntukkan bagi tujuh peristiwa tunggu teratas secara keseluruhan, nilai responsnya akan difilter. Filter menyebabkannya hanya memperhitungkan sesi yang cocok untuk filter tertentu.

Kueri API terkait dalam contoh ini sama seperti perintah di Mengambil rata-rata muatan DB untuk SQL teratas. Namun, file query.json berisi konten berikut.

```
[
 {
        "Metric": "db.load.avg",
        "GroupBy": { "Group": "db.wait_event", "Limit": 5  },
        "Filter": { "db.sql_tokenized.id": "AKIAIOSFODNN7EXAMPLE" }
    }
]
```

Untuk Linux, macOS, atau Unix:

```
aws pi get-resource-metrics \
   --service-type RDS \
```

```
    --identifier db-ID \
    --start-time 2018-10-30T00:00:00Z \
    --end-time   2018-10-30T01:00:00Z \
    --period-in-seconds 60 \
    --metric-queries file://query.json
```

Untuk Windows:

```
aws pi get-resource-metrics ^
    --service-type RDS ^
    --identifier db-ID ^
    --start-time 2018-10-30T00:00:00Z ^
    --end-time   2018-10-30T01:00:00Z ^
    --period-in-seconds 60 ^
    --metric-queries file://query.json
```

Responsnya terlihat seperti berikut.

```
{
    "Identifier": "db-XXX",
    "AlignedStartTime": 1556215200.0,
    "MetricList": [
        {
            "Key": {
                "Metric": "db.load.avg"
            },
            "DataPoints": [
                {
                    "Timestamp": 1556218800.0,
                    "Value": 1.4878117913832196
                },
                {
                    "Timestamp": 1556222400.0,
                    "Value": 1.192823803967328
                }
            ]
        },
        {
            "Key": {
                "Metric": "db.load.avg",
                "Dimensions": {
                    "db.wait_event.type": "io",
                    "db.wait_event.name": "wait/io/aurora_redo_log_flush"
```

```
                }
            },
            "DataPoints": [
                {
                    "Timestamp": 1556218800.0,
                    "Value": 1.1360544217687074
                },
                {
                    "Timestamp": 1556222400.0,
                    "Value": 1.058051341890315
                }
            ]
        },
        {
            "Key": {
                "Metric": "db.load.avg",
                "Dimensions": {
                    "db.wait_event.type": "io",
                    "db.wait_event.name": "wait/io/table/sql/handler"
                }
            },
            "DataPoints": [
                {
                    "Timestamp": 1556218800.0,
                    "Value": 0.16241496598639457
                },
                {
                    "Timestamp": 1556222400.0,
                    "Value": 0.05163360560093349
                }
            ]
        },
        {
            "Key": {
                "Metric": "db.load.avg",
                "Dimensions": {
                    "db.wait_event.type": "synch",
                    "db.wait_event.name": "wait/synch/mutex/innodb/
aurora_lock_thread_slot_futex"
                }
            },
            "DataPoints": [
                {
                    "Timestamp": 1556218800.0,
```

```
                    "Value": 0.11479591836734694
                },
                {
                    "Timestamp": 1556222400.0,
                    "Value": 0.013127187864644107
                }
            ]
        },
        {
            "Key": {
                "Metric": "db.load.avg",
                "Dimensions": {
                    "db.wait_event.type": "CPU",
                    "db.wait_event.name": "CPU"
                }
            },
            "DataPoints": [
                {
                    "Timestamp": 1556218800.0,
                    "Value": 0.05215419501133787
                },
                {
                    "Timestamp": 1556222400.0,
                    "Value": 0.05805134189031505
                }
            ]
        },
        {
            "Key": {
                "Metric": "db.load.avg",
                "Dimensions": {
                    "db.wait_event.type": "synch",
                    "db.wait_event.name": "wait/synch/mutex/innodb/lock_wait_mutex"
                }
            },
            "DataPoints": [
                {
                    "Timestamp": 1556218800.0,
                    "Value": 0.017573696145124718
                },
                {
                    "Timestamp": 1556222400.0,
                    "Value": 0.002333722287047841
                }
```

```
            ]
        }
    ],
    "AlignedEndTime": 1556222400.0
} //end of response
```

Dalam respons ini, semua nilai difilter sesuai dengan kontribusi AKIAIOSFODNN7EXAMPLE SQL token yang ditentukan dalam file query.json. Kunci mungkin juga mengikuti urutan yang berbeda dari kueri tanpa filter, karena lima peristiwa tunggu teratas tersebutlah yang memengaruhi SQL yang difilter.

Mengambil teks lengkap pernyataan SQL

Contoh berikut mengambil teks lengkap pernyataan SQL untuk instans DB `db-10BCD2EFGHIJ3KL4M5NO6PQRS5`. `--group` adalah `db.sql`, dan `--group-identifier` adalah `db.sql.id`. Dalam contoh ini, *`my-sql-id`*merupakan ID SQL diambil dengan memanggil `pi get-resource-metrics` atau. `pi describe-dimension-keys`

Jalankan perintah berikut.

Untuk Linux, macOS, atau Unix:

```
aws pi get-dimension-key-details \
   --service-type RDS \
   --identifier db-10BCD2EFGHIJ3KL4M5NO6PQRS5 \
   --group db.sql \
   --group-identifier my-sql-id \
   --requested-dimensions statement
```

Untuk Windows:

```
aws pi get-dimension-key-details ^
   --service-type RDS ^
   --identifier db-10BCD2EFGHIJ3KL4M5NO6PQRS5 ^
   --group db.sql ^
   --group-identifier my-sql-id ^
   --requested-dimensions statement
```

Dalam contoh ini, detail dimensinya tersedia. Dengan demikian, Wawasan Performa mengambil teks lengkap pernyataan SQL, tanpa memotongnya.

```
{
    "Dimensions":[
    {
        "Value": "SELECT e.last_name, d.department_name FROM employees e, departments d
 WHERE e.department_id=d.department_id",
        "Dimension": "db.sql.statement",
        "Status": "AVAILABLE"
    },
    ...
    ]
}
```

Membuat laporan analisis performa selama periode waktu tertentu

Contoh berikut membuat laporan analisis performa dengan waktu mulai `1682969503` dan waktu akhir `1682979503` untuk basis data `db-loadtest-0`.

```
        aws pi-test create-performance-analysis-report \
        --service-type RDS \
        --identifier db-loadtest-0 \
        --start-time 1682969503 \
        --end-time 1682979503 \
        --endpoint-url https://api.titan.pi.a2z.com \
        --region us-west-2
```

Responsnya adalah pengidentifikasi unik `report-0234d3ed98e28fb17` untuk laporan tersebut.

```
      {
         "AnalysisReportId": "report-0234d3ed98e28fb17"
      }
```

Mengambil laporan analisis performa

Contoh berikut mengambil detail laporan analisis untuk laporan `report-0d99cc91c4422ee61`.

```
        aws pi-test get-performance-analysis-report \
```

```
--service-type RDS \
--identifier db-loadtest-0 \
--analysis-report-id report-0d99cc91c4422ee61 \
--endpoint-url https://api.titan.pi.a2z.com \
--region us-west-2
```

Respons-nya menampilkan status laporan, ID, detail waktu, dan wawasan.

```
        {
    "AnalysisReport": {
        "Status": "Succeeded",
        "ServiceType": "RDS",
        "Identifier": "db-loadtest-0",
        "StartTime": 1680583486.584,
        "AnalysisReportId": "report-0d99cc91c4422ee61",
        "EndTime": 1680587086.584,
        "CreateTime": 1680587087.139,
        "Insights": [
           ... (Condensed for space)
        ]
    }
}
```

Daftar semua laporan analisis performa untuk instans DB

Contoh berikut mencantumkan semua laporan analisis performa yang tersedia untuk basis data `db-loadtest-0`.

```
           aws pi-test list-performance-analysis-reports \
--service-type RDS \
--identifier db-loadtest-0 \
--endpoint-url https://api.titan.pi.a2z.com \
--region us-west-2
```

Respons ini mencantumkan semua laporan dengan ID laporan, status, dan detail periode waktu.

```
            {
    "AnalysisReports": [
        {
            "Status": "Succeeded",
            "EndTime": 1680587086.584,
            "CreationTime": 1680587087.139,
            "StartTime": 1680583486.584,
            "AnalysisReportId": "report-0d99cc91c4422ee61"
        },
        {
            "Status": "Succeeded",
            "EndTime": 1681491137.914,
            "CreationTime": 1681491145.973,
            "StartTime": 1681487537.914,
            "AnalysisReportId": "report-002633115cc002233"
        },
        {
            "Status": "Succeeded",
            "EndTime": 1681493499.849,
            "CreationTime": 1681493507.762,
            "StartTime": 1681489899.849,
            "AnalysisReportId": "report-043b1e006b47246f9"
        },
        {
            "Status": "InProgress",
            "EndTime": 1682979503.0,
            "CreationTime": 1682979618.994,
            "StartTime": 1682969503.0,
            "AnalysisReportId": "report-01ad15f9b88bcbd56"
        }
    ]
}
```

Menghapus laporan analisis performa

Contoh berikut menghapus laporan analisis untuk basis data `db-loadtest-0`.

```
        aws pi-test delete-performance-analysis-report \
--service-type RDS \
--identifier db-loadtest-0 \
--analysis-report-id report-0d99cc91c4422ee61 \
--endpoint-url https://api.titan.pi.a2z.com \
```

```
--region us-west-2
```

Menambahkan tag ke laporan analisis performa

Contoh berikut menambahkan tag dengan kunci `name` dan nilai `test-tag` ke laporan `report-01ad15f9b88bcbd56`.

```
           aws pi-test tag-resource \
--service-type RDS \
--resource-arn arn:aws:pi:us-west-2:356798100956:perf-reports/RDS/db-loadtest-0/
report-01ad15f9b88bcbd56 \
--tags Key=name,Value=test-tag \
--endpoint-url https://api.titan.pi.a2z.com \
--region us-west-2
```

Mencantumkan semua tag untuk laporan analisis performa

Contoh berikut mencantumkan semua tag untuk laporan `report-01ad15f9b88bcbd56`.

```
       aws pi-test list-tags-for-resource \
--service-type RDS \
--resource-arn arn:aws:pi:us-west-2:356798100956:perf-reports/RDS/db-loadtest-0/
report-01ad15f9b88bcbd56 \
--endpoint-url https://api.titan.pi.a2z.com \
--region us-west-2
```

Respons ini mencantumkan nilai dan kunci untuk semua tag yang ditambahkan ke laporan:

```
        {
    "Tags": [
        {
            "Value": "test-tag",
            "Key": "name"
        }
    ]
}
```

Menghapus tag dari laporan analisis performa

Contoh berikut menghapus tag `name` dari laporan `report-01ad15f9b88bcbd56`.

```
        aws pi-test untag-resource \
--service-type RDS \
--resource-arn arn:aws:pi:us-west-2:356798100956:perf-reports/RDS/db-loadtest-0/
report-01ad15f9b88bcbd56 \
--tag-keys name \
--endpoint-url https://api.titan.pi.a2z.com \
--region us-west-2
```

Setelah tag dihapus, pemanggilan API `list-tags-for-resource` tidak akan mencantumkan tag ini.

Mencatat panggilan Wawasan Performa menggunakan AWS CloudTrail

Wawasan Performa berjalan dengan AWS CloudTrail, layanan yang memberikan data tindakan yang dilakukan oleh pengguna, peran, atau layanan AWS di Wawasan Performa. CloudTrail mengambil semua panggilan API untuk Wawasan Performa sebagai peristiwa. Pengambilan ini mencakup panggilan dari konsol Amazon RDS dan dari panggilan kode ke operasi API Wawasan Performa.

Jika membuat jejak, Anda dapat mengaktifkan pengiriman peristiwa CloudTrail berkelanjutan ke bucket Amazon S3, termasuk peristiwa untuk Wawasan Performa. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru dalam konsol CloudTrail di Riwayat peristiwa. Dengan data yang dikumpulkan oleh CloudTrail, Anda dapat menentukan informasi tertentu. Informasi ini mencakup permintaan yang dibuat untuk Wawasan Performa, alamat IP asal permintaan, siapa yang membuat permintaan, dan kapan permintaan tersebut dibuat. Informasi ini juga mencakup detail tambahan.

Untuk mempelajari selengkapnya tentang CloudTrail, lihat Panduan Pengguna AWS CloudTrail.

Menggunakan informasi Wawasan Performa di CloudTrail

CloudTrail diaktifkan pada akun AWS saat Anda membuat akun tersebut. Saat aktivitas terjadi di Wawasan Performa, aktivitas tersebut dicatat dalam peristiwa CloudTrail beserta peristiwa layanan AWS lain di konsol CloudTrail dalam Riwayat peristiwa. Anda dapat melihat, mencari, dan

mengunduh peristiwa terbaru di akun AWS. Untuk informasi selengkapnya, lihat Melihat Peristiwa dengan Riwayat Peristiwa CloudTrail di Panduan Pengguna AWS CloudTrail.

Untuk data peristiwa yang sedang berlangsung di akun AWS Anda, termasuk peristiwa untuk Wawasan Performa, buatlah jejak. Jejak memungkinkan CloudTrail mengirimkan file log ke bucket Amazon S3. Secara default, ketika Anda membuat jejak di konsol tersebut, jejak diterapkan ke semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah AWS di sebagian AWS dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi layanan AWS lainnya untuk menganalisis lebih lanjut dan bertindak berdasarkan data peristiwa yang dikumpulkan di log CloudTrail. Untuk informasi selengkapnya, lihat topik berikut di AWS CloudTrailPanduan Pengguna :

- Gambaran Umum untuk Membuat Jejak
- Layanan dan Integrasi yang Didukung CloudTrail
- Mengonfigurasi Notifikasi Amazon SNS untuk CloudTrail
- Menerima File Log CloudTrail dari Banyak Wilayah dan Menerima File Log CloudTrail dari Banyak Akun

Semua operasi Wawasan Performa dicatat oleh CloudTrail dan didokumentasikan dalam Referensi API Wawasan Performa. Misalnya, panggilan ke operasi `DescribeDimensionKeys` dan `GetResourceMetrics` menghasilkan entri dalam file log CloudTrail.

Setiap entri peristiwa atau log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan berikut ini:

- Apakah permintaan tersebut dibuat dengan kredensial root atau pengguna IAM.
- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna gabungan.
- Apakah permintaan tersebut dibuat oleh layanan AWS lainnya.

Untuk informasi selengkapnya, lihat CloudTrail userIdentity Element.

Entri file log Wawasan Performa

Jejak adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang telah Anda tentukan. File log CloudTrail berisi satu atau beberapa entri log. Peristiwa menunjukkan satu permintaan dari sumber mana pun. Setiap peristiwa mencakup informasi

tentang operasi yang diminta, tanggal dan waktu operasi, parameter permintaan, dan sebagainya. File log CloudTrail bukanlah jejak tumpukan yang berurutan dari panggilan API publik, sehingga file tersebut tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan entri log CloudTrail yang menunjukkan operasi `GetResourceMetrics`.


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
         "principalId": "AKIAIOSFODNN7EXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/johndoe",
        "accountId": "123456789012",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "userName": "johndoe"
    },
    "eventTime": "2019-12-18T19:28:46Z",
    "eventSource": "pi.amazonaws.com",
    "eventName": "GetResourceMetrics",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "72.21.198.67",
    "userAgent": "aws-cli/1.16.240 Python/3.7.4 Darwin/18.7.0 botocore/1.12.230",
    "requestParameters": {
        "identifier": "db-YTDU5J5V66X7CXSCVDFD2V3SZM",
        "metricQueries": [
            {
                "metric": "os.cpuUtilization.user.avg"
            },
            {
                "metric": "os.cpuUtilization.idle.avg"
            }
        ],
        "startTime": "Dec 18, 2019 5:28:46 PM",
        "periodInSeconds": 60,
        "endTime": "Dec 18, 2019 7:28:46 PM",
        "serviceType": "RDS"
    },
    "responseElements": null,
    "requestID": "9ffbe15c-96b5-4fe6-bed9-9fccff1a0525",
    "eventID": "08908de0-2431-4e2e-ba7b-f5424f908433",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}
```

Menganalisis anomali kinerja dengan Amazon DevOps Guru untuk Amazon RDS

Amazon DevOps Guru adalah layanan operasi yang dikelola sepenuhnya yang membantu pengembang dan operator meningkatkan kinerja dan ketersediaan aplikasi mereka. DevOpsGuru menurunkan tugas yang terkait dengan mengidentifikasi masalah operasional sehingga Anda dapat dengan cepat menerapkan rekomendasi untuk meningkatkan aplikasi Anda. Untuk informasi selengkapnya, lihat Apa itu Amazon DevOps Guru? di Panduan Pengguna Amazon DevOps Guru.

DevOpsGuru mendeteksi, menganalisis, dan membuat rekomendasi untuk masalah operasional yang ada untuk semua mesin Amazon RDS DB. DevOpsGuru untuk RDS memperluas kemampuan ini dengan menerapkan pembelajaran mesin ke metrik Performance Insights untuk . Fitur pemantauan ini memungkinkan DevOps Guru for RDS mendeteksi dan mendiagnosis kemacetan kinerja dan merekomendasikan tindakan korektif tertentu. DevOpsGuru untuk RDS juga dapat mendeteksi kondisi bermasalah di database Anda RDS untuk database PostgreSQL sebelum terjadi.

Anda sekarang dapat melihat rekomendasi ini di konsol RDS. Untuk informasi selengkapnya, lihat Melihat dan menanggapi rekomendasi Amazon Aurora RDS.

Video berikut adalah ikhtisar DevOps Guru untuk RDS.

Untuk menyelam lebih dalam tentang hal ini, lihat Amazon DevOps Guru untuk RDS di bawah tenda.

Topik

- Manfaat DevOps Guru untuk RDS
- Bagaimana DevOps Guru untuk RDS bekerja
- Menyiapkan DevOps Guru untuk RDS

Manfaat DevOps Guru untuk RDS

Jika Anda bertanggung jawab atas sebuah basis data RDS for PostgreSQL, Anda mungkin tidak tahu bahwa sedang terjadi suatu peristiwa atau regresi yang memengaruhi basis data itu. Ketika mengetahui masalah ini, Anda mungkin tidak tahu alasannya terjadi atau apa yang harus dilakukan terhadapnya. Daripada beralih ke administrator database (DBA) untuk bantuan atau mengandalkan alat pihak ketiga, Anda dapat mengikuti rekomendasi dari DevOps Guru untuk RDS.

Anda mendapatkan keuntungan berikut dari analisis rinci DevOps Guru untuk RDS:

Diagnosis cepat

DevOpsGuru untuk RDS terus memantau dan menganalisis telemetri database. DevOpsGuru untuk RDS menggunakan teknik statistik dan pembelajaran mesin untuk menambang data ini dan mendeteksi anomali. Untuk mempelajari lebih lanjut data telemetri, lihat Memantau beban basis data dengan Wawasan Performa di Amazon RDS dan Memantau metrik-metrik OS dengan Pemantauan Disempurnakan dalam Panduan Pengguna Amazon RDS.

Resolusi cepat

Setiap anomali mengidentifikasi masalah kinerja dan menyarankan alur investigasi atau tindakan korektif. Misalnya, DevOps Guru untuk RDS mungkin menyarankan Anda menyelidiki peristiwa menunggu tertentu. Atau mungkin menyarankan agar Anda menyetel pengaturan kumpulan aplikasi Anda untuk membatasi jumlah koneksi basis data. Berdasarkan rekomendasi ini, Anda dapat menyelesaikan masalah kinerja lebih cepat daripada dengan memecahkan masalah secara manual.

Wawasan proaktif

DevOpsGuru untuk RDS menggunakan metrik dari sumber daya Anda untuk mendeteksi perilaku yang berpotensi bermasalah sebelum menjadi masalah yang lebih besar. Misalnya, fitur ini dapat mendeteksi ketika basis data Anda menggunakan makin banyak tabel sementara pada disk, yang dapat mulai mempengaruhi kinerja. DevOpsGuru kemudian memberikan rekomendasi untuk membantu Anda mengatasi masalah sebelum menjadi masalah yang lebih besar.

Pengetahuan mendalam insinyur Amazon dan pembelajaran mesin

Untuk mendeteksi masalah kinerja dan membantu Anda mengatasi kemacetan, DevOps Guru for RDS mengandalkan pembelajaran mesin (ML) dan rumus matematika tingkat lanjut. Insinyur basis data Amazon berkontribusi pada pengembangan temuan DevOps Guru untuk RDS, yang merangkum bertahun-tahun mengelola ratusan ribu database. Dengan memanfaatkan pengetahuan kolektif ini, DevOps Guru untuk RDS dapat mengajari Anda praktik terbaik.

Bagaimana DevOps Guru untuk RDS bekerja

DevOpsGuru for RDS mengumpulkan data tentang database Performance Insights. Metrik yang paling penting adalah`DBLoad`. DevOpsGuru for RDS menggunakan metrik Performance Insights, menganalisisnya dengan pembelajaran mesin, dan menerbitkan wawasan ke dasbor.

Wawasan adalah kumpulan anomali terkait yang terdeteksi oleh DevOps Guru.

Dalam DevOps Guru untuk RDS, anomali adalah pola yang menyimpang dari apa yang dianggap kinerja normal untuk Amazon RDS Anda untuk database PostgreSQL.

Wawasan proaktif

Wawasan proaktif memberi tahu Anda tentang perilaku bermasalah sebelum menimbulkan masalah. Wawasan berisi anomali dengan rekomendasi dan metrik terkait untuk membantu Anda mengatasi masalah di basis data RDS for PostgreSQL sebelum menjadi masalah yang lebih besar. Wawasan ini dipublikasikan di dasbor DevOps Guru.

Misalnya, DevOps Guru mungkin mendeteksi bahwa RDS Anda untuk database PostgreSQL membuat banyak tabel sementara on-disk. Jika tidak ditangani, tren ini dapat menyebabkan masalah kinerja. Setiap wawasan proaktif mencakup rekomendasi untuk perilaku korektif dan penaut ke topik yang relevan di Menyeetel RDS for PostgreSQL dengan wawasan proaktif Amazon DevOps Guru. Untuk informasi selengkapnya, lihat Bekerja dengan wawasan di DevOps Guru di Panduan Pengguna Amazon DevOps Guru.

Wawasan reaktif

Wawasan reaktif mengidentifikasi perilaku anomali saat terjadi. Jika DevOps Guru for RDS menemukan masalah kinerja di Aurora RDS Anda untuk instans PostgreSQL DB, Guru akan menerbitkan wawasan reaktif di dasbor Guru. DevOps Untuk informasi selengkapnya, lihat Bekerja dengan wawasan di DevOps Guru di Panduan Pengguna Amazon DevOps Guru.

Anomali kausal

Anomali kausal adalah anomali tingkat puncak dalam wawasan reaktif. Beban basis data (beban DB) adalah anomali kausal untuk DevOps Guru untuk RDS.

Anomali mengukur dampak kinerja dengan menetapkan tingkat keparahan Tinggi, Sedang, atau Rendah. Untuk mempelajari lebih lanjut, lihat Konsep kunci untuk DevOps Guru for RDS di Panduan Pengguna Amazon DevOps Guru.

Jika DevOps Guru mendeteksi anomali saat ini pada instans DB Anda, Anda akan diberi tahu di halaman Database konsol RDS. Konsol juga memperingatkan Anda tentang anomali yang terjadi dalam 24 jam terakhir. Untuk menuju halaman anomali dari konsol RDS, pilih penaut dalam pesan peringatan. Konsol RDS juga memperingatkan Anda di halaman itu untuk instans basis data RDS for PostgreSQL.

Anomali kontekstual

Anomali kontekstual adalah temuan dalam Beban basis data (Beban DB) yang terkait dengan wawasan reaktif. Setiap anomali kontekstual menjelaskan masalah kinerja RDS for PostgreSQL tertentu yang memerlukan penyelidikan. Misalnya, DevOps Guru untuk RDS mungkin menyarankan Anda mempertimbangkan untuk meningkatkan kapasitas CPU atau menyelidiki peristiwa tunggu yang berkontribusi pada pemuatan DB.

> **Important**
>
> Sebaiknya uji setiap perubahan pada instans uji sebelum diterapkan pada instans produksi. Dengan cara ini, Anda memahami dampak perubahan.

Untuk mempelajari lebih lanjut, lihat Menganalisis anomali di Amazon RDS di Panduan Pengguna Amazon DevOps Guru.

Menyiapkan DevOps Guru untuk RDS

Untuk mengizinkan DevOps Guru for Amazon RDS mempublikasikan wawasan untuk Amazon PostgreSQL, selesaikan tugas-tugas berikut.

Topik

- Mengkonfigurasi kebijakan akses IAM untuk DevOps Guru untuk RDS
- Mengaktifkan Wawasan Performa untuk instans basis data RDS for PostgreSQL Anda
- Mengaktifkan DevOps Guru dan menentukan cakupan sumber daya

Mengkonfigurasi kebijakan akses IAM untuk DevOps Guru untuk RDS

Untuk melihat peringatan dari DevOps Guru di konsol RDS, pengguna atau peran AWS Identity and Access Management (IAM) Anda harus memiliki salah satu dari kebijakan berikut:

- Kebijakan terkelola AWS `AmazonDevOpsGuruConsoleFullAccess`
- Kebijakan terkelola AWS `AmazonDevOpsGuruConsoleReadOnlyAccess` dan salah satu kebijakan berikut:
  - Kebijakan terkelola AWS `AmazonRDSFullAccess`
  - Kebijakan terkelola pelanggan yang mencakup `pi:GetResourceMetrics` dan `pi:DescribeDimensionKeys`

Untuk informasi selengkapnya, lihat Mengonfigurasi kebijakan akses untuk Wawasan Performa.

Mengaktifkan Wawasan Performa untuk instans basis data RDS for PostgreSQL Anda

DevOpsGuru untuk RDS mengandalkan Performance Insights untuk datanya. Tanpa Performance Insights, DevOps Guru menerbitkan anomali, tetapi tidak menyertakan analisis dan rekomendasi terperinci.

Saat membuat atau mengubah instans basis data RDS for PostgreSQL, Anda dapat mengaktifkan Wawasan Performa. Untuk informasi selengkapnya, lihat Mengaktifkan dan menonaktifkan Wawasan Performa.

Mengaktifkan DevOps Guru dan menentukan cakupan sumber daya

Anda dapat mengaktifkan DevOps Guru agar memonitor Anda untuk database PostgreSQL dengan salah satu cara berikut.

Topik

- Menghidupkan DevOps Guru di konsol RDS
- Menambahkan untuk sumber daya PostgreSQL di konsol Guru DevOps
- Menambahkan sumber daya RDS for PostgreSQL dengan menggunakan AWS CloudFormation

Menghidupkan DevOps Guru di konsol RDS

Anda dapat mengambil beberapa jalur di konsol Amazon RDS untuk mengaktifkan DevOps Guru.

Topik

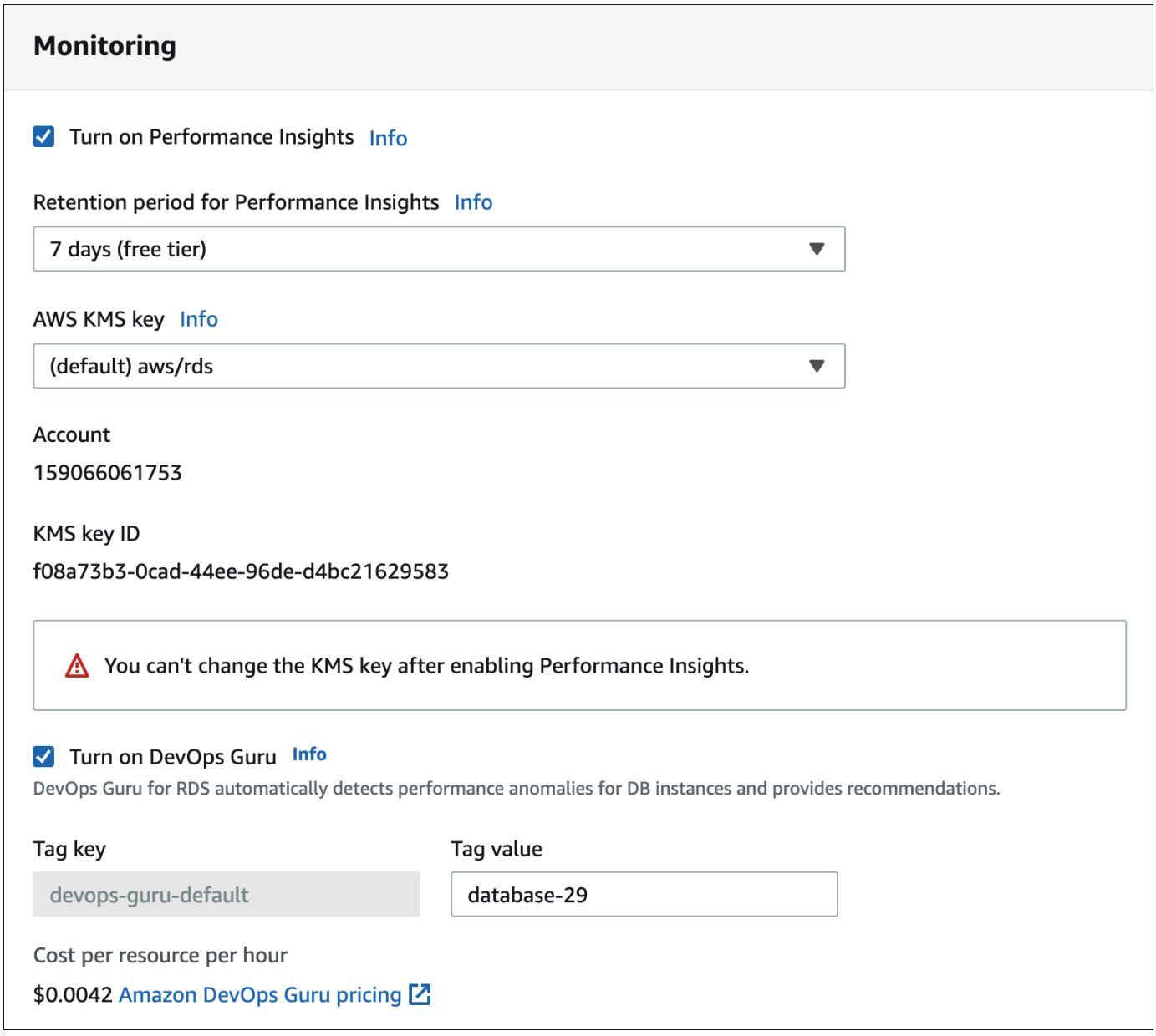
- Mengaktifkan DevOps Guru saat Anda membuat RDS untuk database PostgreSQL
- Menghidupkan DevOps Guru dari spanduk notifikasi
- Menanggapi kesalahan izin saat Anda mengaktifkan Guru DevOps

Mengaktifkan DevOps Guru saat Anda membuat RDS untuk database PostgreSQL

Alur kerja pembuatan mencakup pengaturan yang mengaktifkan cakupan DevOps Guru untuk database Anda. Pengaturan ini diaktifkan secara bawaan saat Anda memilih templat Produksi.

Untuk mengaktifkan DevOps Guru saat Anda membuat RDS untuk database PostgreSQL

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Ikuti langkah-langkah di Membuat instans DB, sampai tetapi tidak meliputi langkah ketika Anda memilih setelan pemantauan.
3. Di Pemantauan, pilih Aktifkan Wawasan Performa. Agar DevOps Guru for RDS dapat memberikan analisis terperinci tentang anomali kinerja, Performance Insights harus diaktifkan.
4. Pilih Aktifkan DevOps Guru.



5. Buat tag untuk database Anda sehingga DevOps Guru dapat memantaunya. Lakukan hal-hal berikut:

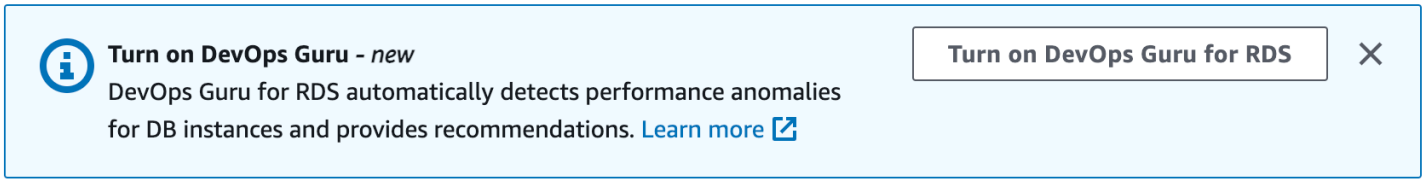
   - Di bidang teks untuk Kunci tag, masukkan nama yang dimulai dengan **`Devops-Guru-`**.
   - Di bidang teks untuk Nilai tag, masukkan nilai apa pun. Misalnya, jika Anda memasukkan **`rds-database-1`** untuk nama basis data RDS for PostgreSQL, Anda juga dapat memasukkan **`rds-database-1`** sebagai nilai tag.

   Untuk informasi selengkapnya tentang tag, lihat "Menggunakan tag untuk mengidentifikasi sumber daya dalam aplikasi DevOps Guru Anda" di Panduan Pengguna Amazon DevOps Guru.
6. Selesaikan langkah-langkah selebihnya di Membuat instans DB.

Menghidupkan DevOps Guru dari spanduk notifikasi

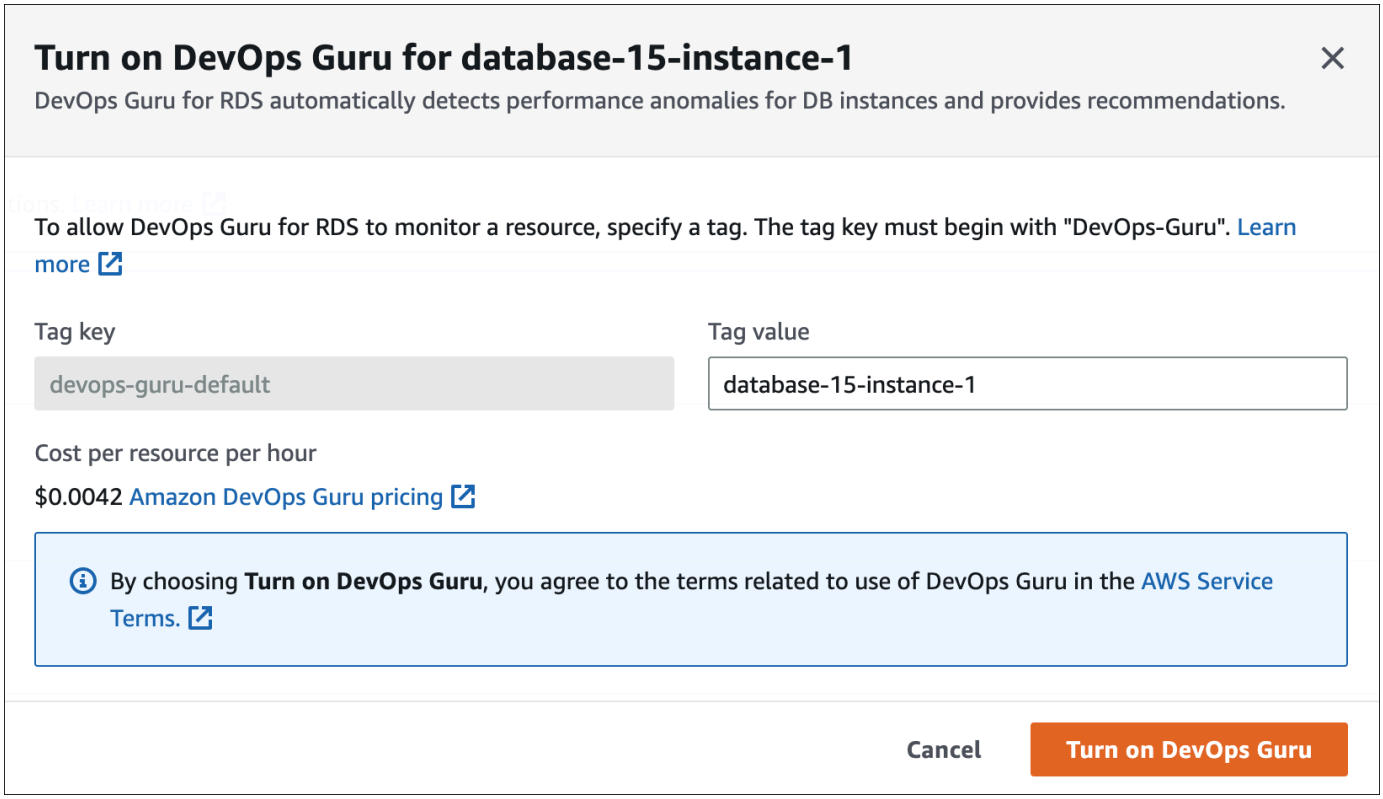
Jika sumber daya Anda tidak dicakup oleh DevOps Guru, Amazon RDS akan memberi tahu Anda dengan spanduk di lokasi berikut:

- Tab Pemantauan instans klaster basis data
- Dasbor Wawasan Performa



Untuk mengaktifkan DevOps Guru untuk RDS Anda untuk database PostgreSQL

1. Di spanduk, pilih Aktifkan DevOps Guru untuk RDS.
2. Masukkan nama kunci dan nilai tag. Untuk informasi selengkapnya tentang tag, lihat "Menggunakan tag untuk mengidentifikasi sumber daya dalam aplikasi DevOps Guru Anda" di Panduan Pengguna Amazon DevOps Guru.



3. Pilih Aktifkan DevOps Guru.

Menanggapi kesalahan izin saat Anda mengaktifkan Guru DevOps

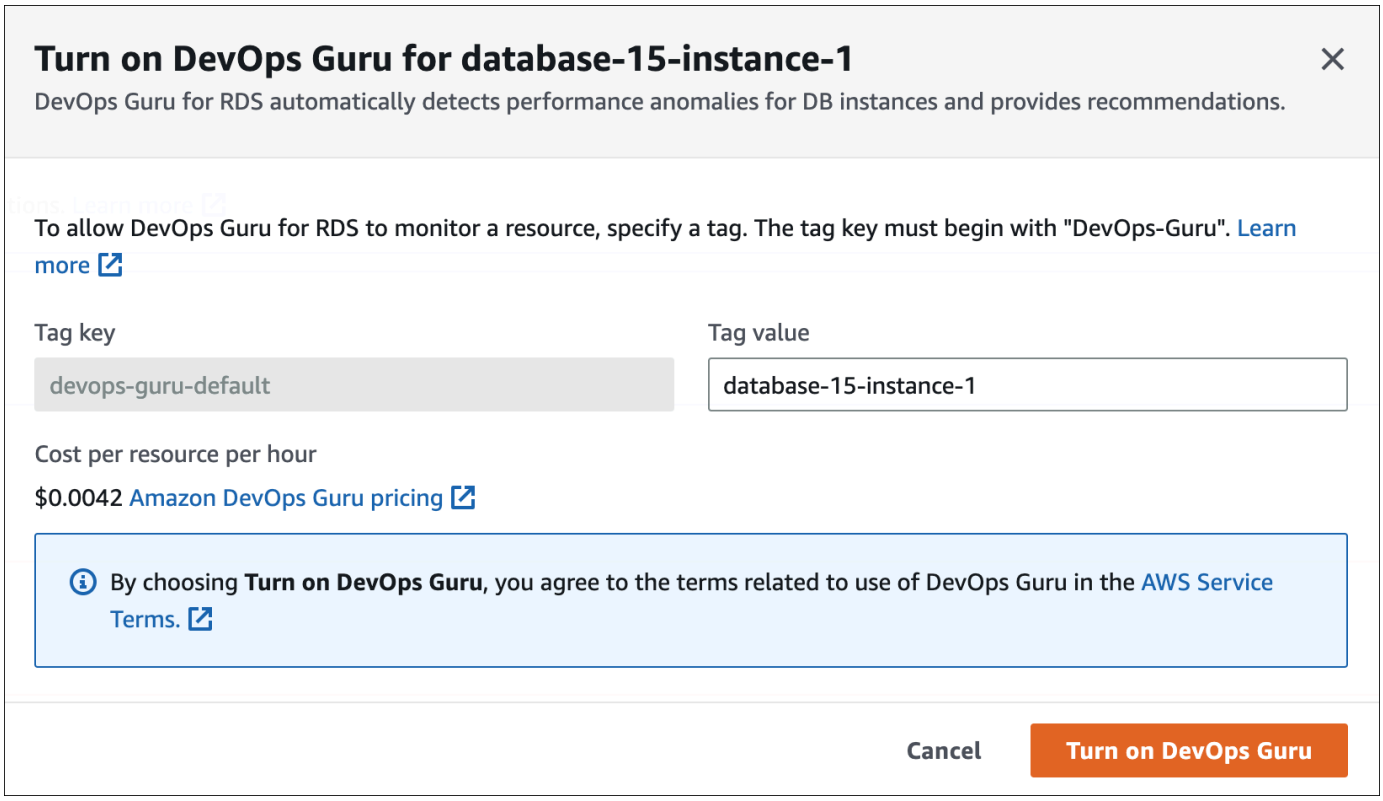
Jika Anda mengaktifkan DevOps Guru dari konsol RDS saat membuat database, RDS mungkin menampilkan spanduk berikut tentang izin yang hilang.

Untuk menanggapi kesalahan izin

1. Beri peran atau pengguna IAM Anda peran terkelola pengguna `AmazonDevOpsGuruConsoleFullAccess`. Untuk informasi selengkapnya, lihat Mengkonfigurasi kebijakan akses IAM untuk DevOps Guru untuk RDS.
2. Buka konsol .
3. Di panel navigasi, pilih Wawasan Performa.
4. Pilih instans basis data di klaster yang baru saja Anda buat.
5. Nyalakan DevOps Guru untuk RDS.

DevOps Guru for RDS

6. Pilih nilai tag. Untuk informasi selengkapnya, lihat "Menggunakan tag untuk mengidentifikasi sumber daya di aplikasi DevOps Guru Anda" di Panduan Pengguna Amazon DevOps Guru.



7. Pilih Aktifkan DevOps Guru.

Menambahkan untuk sumber daya PostgreSQL di konsol Guru DevOps

Anda dapat menentukan cakupan sumber daya DevOps Guru Anda di konsol DevOps Guru. Ikuti langkah yang dijelaskan dalam Tentukan cakupan sumber daya DevOps Guru Anda di Panduan Pengguna Amazon DevOps Guru. Saat Anda mengedit sumber daya yang dianalisis, pilih salah satu opsi berikut:

- Pilih Semua sumber daya akun untuk menganalisis semua sumber daya yang didukung, yang meliputi basis data RDS for PostgreSQL, di Akun AWS dan Kawasan Anda.
- Pilih CloudFormation tumpukan untuk menganalisis RDS untuk database PostgreSQL yang ada di tumpukan yang Anda pilih. Untuk informasi selengkapnya, lihat Menggunakan AWS CloudFormation tumpukan untuk mengidentifikasi sumber daya dalam aplikasi DevOps Guru Anda di Panduan Pengguna Amazon DevOps Guru.

- Pilih Tag untuk menganalisis basis data RDS for PostgreSQL yang telah Anda beri tag. Untuk informasi selengkapnya, lihat Menggunakan tag untuk mengidentifikasi sumber daya dalam aplikasi DevOps Guru Anda di Panduan Pengguna Amazon DevOps Guru.

Untuk informasi selengkapnya, lihat Aktifkan DevOps DevOps Guru di Panduan Pengguna Amazon Guru.

Menambahkan sumber daya RDS for PostgreSQL dengan menggunakan AWS CloudFormation

Anda dapat menggunakan tag untuk menambahkan cakupan untuk RDS Anda untuk sumber daya PostgreSQL ke template Anda. CloudFormation Prosedur berikut mengasumsikan bahwa Anda memiliki CloudFormation template baik untuk RDS Anda untuk instance PostgreSQL DB dan tumpukan Guru. DevOps

Untuk menentukan RDS untuk PostgreSQL DB instance menggunakan tag CloudFormation

1. Dalam CloudFormation template untuk instans DB Anda, tentukan tag menggunakan pasangan kunci/nilai.

   Contoh berikut menetapkan nilai `my-db-instance1` untuk `Devops-guru-cfn-default` bagi instans basis data RDS for PostgreSQL.

   ```
   MyDBInstance1:
     Type: "AWS::RDS::DBInstance"
     Properties:
       DBInstanceIdentifier: my-db-instance1
       Tags:
         - Key: Devops-guru-cfn-default
           Value: devopsguru-my-db-instance1
   ```

2. Dalam CloudFormation template untuk tumpukan DevOps Guru Anda, tentukan tag yang sama di filter pengumpulan sumber daya Anda.

   Contoh berikut mengkonfigurasi DevOps Guru untuk menyediakan cakupan sumber daya dengan nilai `my-db-instance1` tag.

   ```
   DevOpsGuruResourceCollection:
     Type: AWS::DevOpsGuru::ResourceCollection
     Properties:
       ResourceCollectionFilter:
         Tags:
   ```

```
        - AppBoundaryKey: "Devops-guru-cfn-default"
          TagValues:
          - "devopsguru-my-db-instance1"
```

Contoh berikut menyediakan cakupan untuk semua sumber daya dalam batas aplikasi `Devops-guru-cfn-default`.

```
DevOpsGuruResourceCollection:
  Type: AWS::DevOpsGuru::ResourceCollection
  Properties:
    ResourceCollectionFilter:
      Tags:
        - AppBoundaryKey: "Devops-guru-cfn-default"
          TagValues:
          - "*"
```

Untuk informasi selengkapnya, lihat AWS::DevOpsGuru::ResourceCollectionAWS: :RDS: :DBInstance di Panduan Pengguna. AWS CloudFormation

Memantau metrik OS dengan Pemantauan yang Disempurnakan

Dengan Pemantauan yang Disempurnakan, Anda dapat memantau sistem operasi instans DB Anda secara real-time. Metrik Pemantauan yang Disempurnakan berguna saat Anda ingin melihat bagaimana proses atau thread yang berbeda menggunakan CPU.

Topik

Ikhtisar Pemantauan yang Disempurnakan

Amazon RDS menyediakan metrik secara real-time untuk sistem operasi (OS) tempat instans DB Anda dijalankan. Anda dapat melihat semua metrik sistem dan memproses informasi instans DB RDS Anda di konsol. Anda dapat mengelola metrik yang ingin dipantau untuk setiap instans dan menyesuaikan dasbor sesuai kebutuhan Anda. Untuk deskripsi metrik Pemantauan yang Disempurnakan, lihat Metrik OS dalam Pemantauan yang Disempurnakan.

RDS mengirimkan metrik dari Enhanced Monitoring ke akun Amazon CloudWatch Logs Anda. Anda dapat membuat filter metrik CloudWatch dari CloudWatch Log dan menampilkan grafik di dasbor. CloudWatch Anda dapat menggunakan output Enhanced Monitoring JSON dari CloudWatch Log dalam sistem pemantauan pilihan Anda. Untuk informasi selengkapnya, lihat Pemantauan yang Disempurnakan di Tanya Jawab Umum tentang Amazon RDS.

Topik

Ketersediaan Pemantauan yang Disempurnakan

Pemantauan yang Disempurnakan tersedia untuk mesin basis data berikut:

- Db2
- MariaDB
- Microsoft SQL Server
- MySQL
- Oracle
- PostgreSQL

Pemantauan yang Disempurnakan tersedia untuk semua kelas instans DB kecuali untuk kelas instans db.m1.small.

Perbedaan antara CloudWatch dan metrik Pemantauan yang Ditingkatkan

Hipervisor menciptakan dan menjalankan mesin virtual (VM). Menggunakan hypervisor, sebuah instance dapat mendukung beberapa VM tamu dengan berbagi memori dan CPU secara virtual. CloudWatch mengumpulkan metrik tentang pemanfaatan CPU dari hypervisor untuk instance DB. Sebaliknya, Pemantauan yang Disempurnakan mengumpulkan metrik dari agen di instans DB.

Anda mungkin menemukan perbedaan antara pengukuran CloudWatch dan Enhanced Monitoring, karena lapisan hypervisor melakukan sedikit pekerjaan. Perbedaan ini bisa lebih menonjol jika instans DB Anda menggunakan kelas instans yang lebih kecil. Dalam skenario ini, ada lebih banyak mesin virtual (VM) yang mungkin dikelola oleh lapisan hipervisor pada satu instans fisik.

Untuk deskripsi metrik Pemantauan yang Disempurnakan, lihat Metrik OS dalam Pemantauan yang Disempurnakan. Untuk informasi selengkapnya tentang CloudWatch metrik, lihat Panduan CloudWatch Pengguna Amazon.

Retensi metrik Pemantauan yang Disempurnakan

Secara default, metrik Pemantauan yang Ditingkatkan disimpan selama 30 hari di CloudWatch Log. Periode retensi ini berbeda dari CloudWatch metrik biasa.

Untuk mengubah jumlah waktu metrik disimpan di CloudWatch Log, ubah retensi untuk grup `RDSOSMetrics` log di CloudWatch konsol. Untuk informasi selengkapnya, lihat Mengubah penyimpanan data CloudWatch log di log di Panduan Pengguna CloudWatch Log Amazon.

Biaya Pemantauan yang Disempurnakan

Metrik Pemantauan yang Ditingkatkan disimpan di CloudWatch Log, bukan dalam CloudWatch metrik. Biaya Pemantauan yang Disempurnakan ditentukan oleh faktor-faktor berikut:

- Anda dikenakan biaya untuk Enhanced Monitoring hanya jika Anda melebihi tingkat gratis yang disediakan oleh Amazon CloudWatch Logs. Biaya didasarkan pada transfer data CloudWatch Log dan tingkat penyimpanan.
- Jumlah informasi yang ditransfer untuk instans RDS berbanding lurus dengan perincian yang ditentukan untuk fitur Pemantauan yang Disempurnakan. Interval pemantauan yang lebih kecil menghasilkan pelaporan metrik OS yang lebih sering dan meningkatkan biaya pemantauan. Untuk mengelola biaya, atur perincian yang berbeda untuk instans yang berbeda di akun Anda.
- Biaya penggunaan Pemantauan yang Disempurnakan diterapkan untuk setiap instans DB yang fitur Pemantauan yang Disempurnakannya diaktifkan. Pemantauan instans DB dalam jumlah akan lebih mahal dibandingkan dengan pemantauan dalam jumlah sedikit.
- Instans DB yang mendukung beban kerja komputasi yang lebih berat memiliki lebih banyak aktivitas proses OS yang perlu dan biaya untuk Pemantauan yang Disempurnakan lebih tinggi.

Untuk informasi selengkapnya tentang harga, lihat CloudWatch harga Amazon.

Menyiapkan dan mengaktifkan Pemantauan yang Ditingkatkan

Untuk menggunakan Pemantauan yang Ditingkatkan, Anda harus membuat peran IAM, lalu mengaktifkan Pemantauan yang Ditingkatkan.

Topik

- Membuat peran IAM untuk Pemantauan yang Ditingkatkan
- Mengaktifkan dan menonaktifkan Pemantauan yang Ditingkatkan
- Melindungi dari masalah confused deputy

Membuat peran IAM untuk Pemantauan yang Ditingkatkan

Pemantauan yang Ditingkatkan memerlukan izin untuk bertindak atas nama Anda untuk mengirim informasi metrik OS ke CloudWatch Log. Anda memberikan izin Pemantauan yang Ditingkatkan menggunakan peran AWS Identity and Access Management (IAM). Anda dapat membuat peran ini saat mengaktifkan Pemantauan yang Ditingkatkan atau membuatnya terlebih dahulu.

Topik

- Membuat peran IAM saat Anda mengaktifkan Pemantauan yang Ditingkatkan
- Membuat peran IAM saat Anda mengaktifkan Pemantauan yang Ditingkatkan

Membuat peran IAM saat Anda mengaktifkan Pemantauan yang Ditingkatkan

Jika Anda mengaktifkan Pemantauan yang Ditingkatkan di konsol RDS, Amazon RDS dapat membuat peran IAM yang diperlukan untuk Anda. Peran ini bernama `rds-monitoring-role`. RDS menggunakan peran ini untuk instans DB, replika baca, atau klaster DB Multi-AZ tertentu.

Untuk membuat peran IAM saat Anda mengaktifkan Pemantauan yang Ditingkatkan

1. Ikuti langkah-langkah di Mengaktifkan dan menonaktifkan Pemantauan yang Ditingkatkan.
2. Atur Peran Pemantauan ke Default pada langkah tempat Anda memilih peran.

Membuat peran IAM saat Anda mengaktifkan Pemantauan yang Ditingkatkan

Anda dapat membuat peran yang diperlukan sebelum mengaktifkan Pemantauan yang Ditingkatkan. Jika Anda mengaktifkan Pemantauan yang Ditingkatkan, tentukan nama peran baru Anda. Anda harus membuat peran yang diperlukan ini jika Anda mengaktifkan Pemantauan yang Ditingkatkan menggunakan AWS CLI atau API RDS.

Pengguna yang mengaktifkan Pemantauan yang Ditingkatkan harus diberi izin `PassRole`. Untuk informasi selengkapnya, lihat Contoh 2 dalam Memberikan izin pengguna untuk meneruskan peran ke AWS layanan di Panduan Pengguna IAM.

Untuk membuat peran IAM untuk pemantauan yang ditingkatkan Amazon RDS

1. Buka Konsol IAM di https://console.aws.amazon.com.
2. Di panel navigasi, pilih Peran.
3. Pilih Buat peran.
4. Pilih tab Layanan AWS , lalu pilih RDS dari daftar layanan.
5. Pilih RDS - Pemantauan yang Ditingkatkan, lalu pilih Berikutnya.
6. Pastikan kebijakan Izin menampilkan AmazonRDS EnhancedMonitoringRole, lalu pilih Berikutnya.
7. Untuk Nama peran, masukkan nama peran Anda. Misalnya, masukkan **`emaccess`**.

   Entitas tepercaya untuk peran Anda adalah AWS layanan monitoring.rds.amazonaws.com.
8. Pilih Buat peran.

Mengaktifkan dan menonaktifkan Pemantauan yang Ditingkatkan

Anda dapat mengaktifkan dan menonaktifkan Enhanced Monitoring menggunakan AWS Management Console, AWS CLI, atau RDS API. Anda memilih instans DB RDS yang Pemantauan yang Ditingkatkannya ingin diaktifkan. Anda dapat mengatur granularitas yang berbeda untuk pengumpulan metrik pada setiap instans DB.

Konsol

Anda dapat mengaktifkan Pemantauan yang Ditingkatkan saat membuat instans DB, klaster DB Multi-AZ, atau replika baca, atau saat Anda memodifikasi instans DB atau klaster DB Multi-AZ. Jika Anda memodifikasi instans DB untuk mengaktifkan Pemantauan yang Ditingkatkan, Anda tidak perlu mem-boot ulang instans DB Anda agar perubahan diterapkan.

Anda dapat mengaktifkan Pemantauan yang Ditingkatkan di konsol RDS saat Anda melakukan salah satu tindakan berikut di halaman Basis data:

- Buat instans DB atau klaster DB Multi-AZ – Pilih Buat basis data.
- Buat replika baca – Pilih Tindakan, lalu Buat replika baca.
- Modifikasi instans DB atau klaster DB Multi-AZ – Pilih Ubah.

Untuk mengaktifkan atau menonaktifkan Pemantauan yang Ditingkatkan di konsol RDS

1. Gulir ke bagian Konfigurasi tambahan.
2. Di Pemantauan, pilih Aktifkan Pemantauan yang Ditingkatkan untuk instans DB atau replika baca. Untuk menonaktifkan Pemantauan yang Ditingkatkan, pilih Nonaktifkan Pemantauan yang Ditingkatkan.
3. Setel properti Peran Pemantauan ke peran IAM yang Anda buat untuk mengizinkan Amazon RDS berkomunikasi dengan CloudWatch Log Amazon untuk Anda, atau pilih Default agar RDS membuat peran untuk Anda namai. `rds-monitoring-role`
4. Atur properti Granularitas ke interval tersebut, dalam detik, antara titik-titik saat metrik dikumpulkan untuk instans DB atau replika baca. Properti Granularitas dapat diatur ke salah satu nilai berikut: `1`, `5`, `10`, `15`, `30`, atau `60`.

   Waktu yang paling cepat di mana konsol RDS disegarkan adalah setiap 5 detik. Jika Anda mengatur granularitas ke 1 detik di konsol RDS, Anda masih dapat melihat metrik yang diperbarui hanya setiap 5 detik. Anda dapat mengambil pembaruan metrik 1 detik dengan menggunakan CloudWatch Log.

AWS CLI

Untuk mengaktifkan Enhanced Monitoring menggunakan AWS CLI, dalam perintah berikut, atur `--monitoring-interval` opsi ke nilai selain `0` dan atur `--monitoring-role-arn` opsi ke peran yang Anda buatMembuat peran IAM untuk Pemantauan yang Ditingkatkan.

- create-db-instance
- create-db-instance-read-replika
- modify-db-instance
- create-db-cluster(Kluster DB multi-AZ)
- modify-db-cluster(Kluster DB multi-AZ)

Opsi `--monitoring-interval` menentukan interval, dalam detik, antara titik-titik saat metrik Pemantauan yang Ditingkatkan dikumpulkan. Nilai yang valid untuk opsi ini adalah `0`, `1`, `5`, `10`, `15`, `30`, dan `60`.

Untuk mematikan Enhanced Monitoring menggunakan AWS CLI, atur `--monitoring-interval` opsi ke `0` dalam perintah ini.

Example

Contoh berikut mengaktifkan Pemantauan yang Ditingkatkan untuk instans DB:

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --monitoring-interval 30 \
    --monitoring-role-arn arn:aws:iam::123456789012:role/emaccess
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --monitoring-interval 30 ^
    --monitoring-role-arn arn:aws:iam::123456789012:role/emaccess
```

Example

Contoh berikut mengaktifkan Pemantauan yang Ditingkatkan untuk klaster DB Multi-AZ:

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-cluster \
    --db-cluster-identifier mydbcluster \
    --monitoring-interval 30 \
    --monitoring-role-arn arn:aws:iam::123456789012:role/emaccess
```

Untuk Windows:

```
aws rds modify-db-cluster ^
    --db-cluster-identifier mydbcluster ^
    --monitoring-interval 30 ^
    --monitoring-role-arn arn:aws:iam::123456789012:role/emaccess
```

API RDS

Untuk mengaktifkan Pemantauan yang Ditingkatkan menggunakan RDS API, atur parameter `MonitoringInterval` ke nilai selain `0` dan atur parameter `MonitoringRoleArn` ke peran yang Anda buat di Membuat peran IAM untuk Pemantauan yang Ditingkatkan. Tetapkan parameter ini dalam tindakan berikut:

- CreateDBInstance
- dibuatB InstanceReadReplica
- ModifyDBInstance
- CreateDBCluster (klaster DB Multi-AZ)
- ModifyDBCluster (klaster DB Multi-AZ)

Parameter `MonitoringInterval` menentukan interval, dalam detik, antara titik-titik saat metrik Pemantauan yang Ditingkatkan dikumpulkan. Nilai yang valid adalah `0`, `1`, `5`, `10`, `15`, `30`, dan `60`.

Untuk menonaktifkan Pemantauan yang Ditingkatkan menggunakan API RDS, atur `MonitoringInterval` ke `0`.

Melindungi dari masalah confused deputy

Masalah deputi yang bingung adalah masalah keamanan di mana entitas yang tidak memiliki izin untuk melakukan tindakan dapat memaksa entitas yang lebih istimewa 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 pengguna utama layanan yang telah diberi akses ke sumber daya di akun Anda. Untuk informasi selengkapnya, lihat Masalah confused deputy.

Untuk membatasi izin ke sumber daya yang dapat diberikan Amazon RDS kepada layanan lain, sebaiknya gunakan kunci konteks kondisi global `aws:SourceArn` dan `aws:SourceAccount` dalam kebijakan kepercayaan untuk peran Pemantauan yang Ditingkatkan. Jika Anda menggunakan kedua kunci konteks kondisi global, keduanya harus menggunakan ID akun yang sama.

Cara paling efektif untuk melindungi dari masalah confused deputy adalah dengan menggunakan kunci konteks kondisi global `aws:SourceArn` dengan ARN lengkap sumber daya. Untuk Amazon RDS, atur `aws:SourceArn` ke `arn:aws:rds:`*`Region`*`:`*`my-account-id`*`:db:`*`dbname`*.

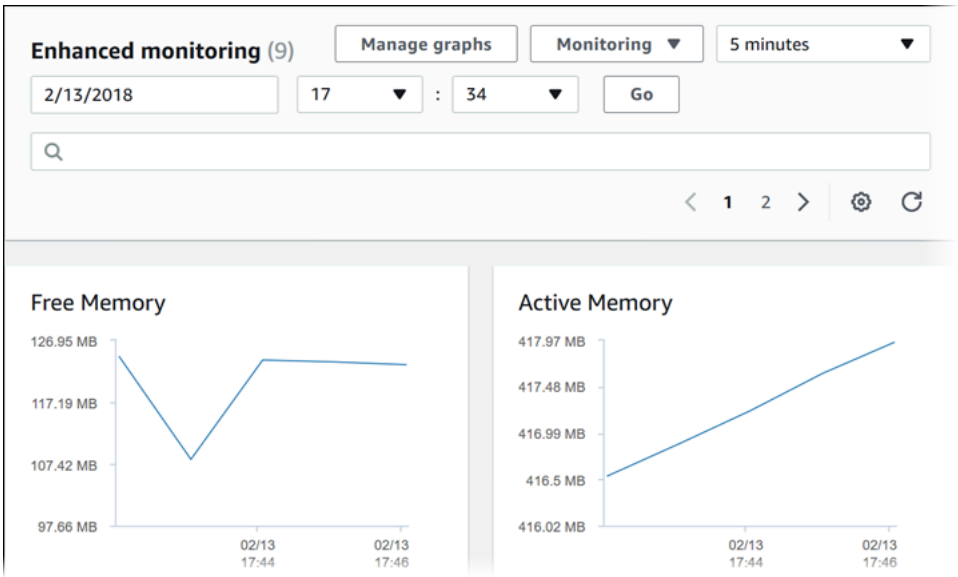
Contoh berikut menggunakan kunci konteks kondisi global `aws:SourceArn` dan `aws:SourceAccount` dalam kebijakan kepercayaan untuk mencegah masalah deputi yang bingung.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "monitoring.rds.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringLike": {
          "aws:SourceArn": "arn:aws:rds:Region:my-account-id:db:dbname"
        },
        "StringEquals": {
          "aws:SourceAccount": "my-account-id"
        }
      }
    }
  ]
}
```

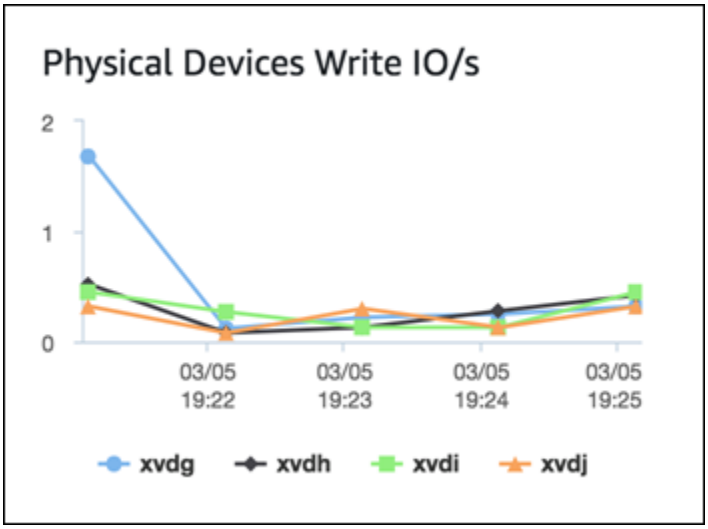
Melihat metrik OS di konsol RDS

Anda dapat melihat metrik OS yang dilaporkan oleh Pemantauan yang Disempurnakan di konsol RDS dengan memilih Pemantauan yang disempurnakan untuk Pemantauan.

Contoh berikut menunjukkan halaman Pemantauan yang Disempurnakan. Untuk deskripsi metrik Pemantauan yang Disempurnakan, lihat Metrik OS dalam Pemantauan yang Disempurnakan.

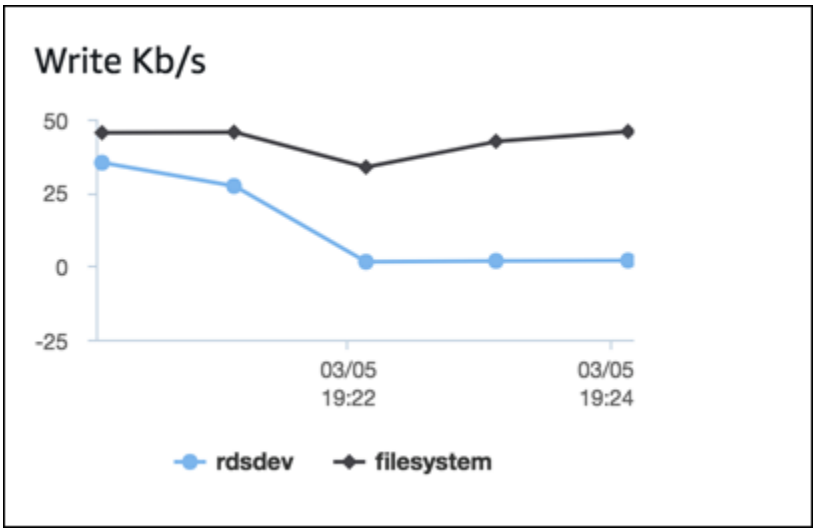


Beberapa instans DB menggunakan lebih dari satu disk untuk volume penyimpanan data instans DB. Pada instans DB tersebut, grafik Perangkat Fisik menunjukkan metrik untuk setiap disk. Misalnya, grafik berikut menunjukkan metrik untuk empat disk.

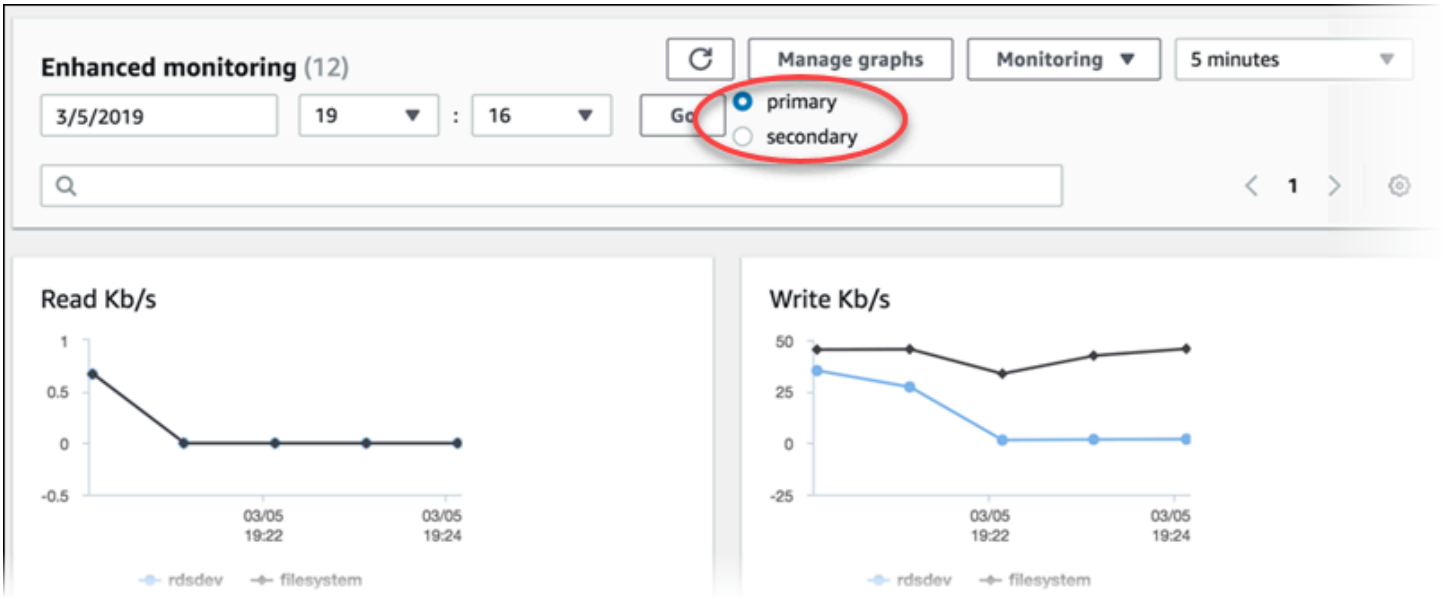


> **Note**
>
> Saat ini, grafik Perangkat Fisik tidak tersedia untuk instans DB Microsoft SQL Server.

Saat Anda melihat grafik I/O Disk dan Sistem file gabungan, perangkat rdsdev berhubungan dengan sistem file `/rdsdbdata`, tempat semua file dan log basis data disimpan. Perangkat filesystem berhubungan dengan sistem file `/` (juga dikenal sebagai root), tempat file yang terkait dengan sistem operasi disimpan.



Jika instans DB berupa deployment Multi-AZ, Anda dapat melihat metrik OS untuk instans DB primer dan replika siaga Multi-AZ. Dalam tampilan Pemantauan yang disempurnakan, pilih primer untuk melihat metrik OS untuk instans DB primer, atau pilih sekunder untuk melihat metrik OS untuk replika siaga.

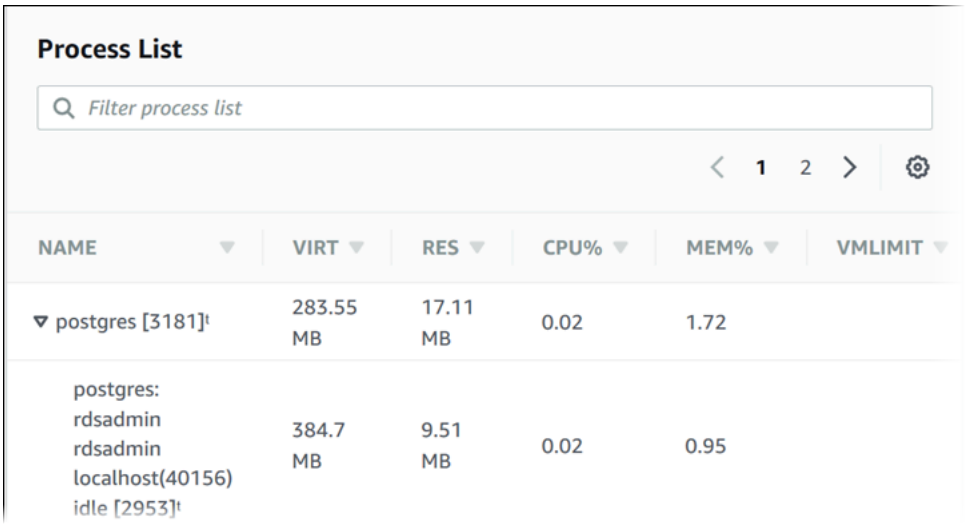


Untuk informasi selengkapnya tentang deployment Multi-AZ, lihat Mengonfigurasi dan mengelola deployment Multi-AZ.

> **Note**
>
> Saat ini, melihat metrik OS untuk replika siaga Multi-AZ tidak didukung untuk instans DB MariaDB.

Jika Anda ingin melihat detail proses yang berjalan pada instans DB Anda, pilih Daftar proses OS untuk Pemantauan.

Tampilan Daftar Proses ditunjukkan sebagai berikut.



Metrik Pemantauan yang Disempurnakan yang ditunjukkan dalam tampilan Daftar proses disusun sebagai berikut:

- Proses turunan RDS – Menampilkan ringkasan proses RDS yang mendukung instans DB, misalnya `mysqld` untuk instans DB MySQL. Rangkaian proses muncul bersarang di bawah proses induk. Rangkaian proses hanya menampilkan penggunaan CPU karena metrik lain sama untuk semua rangkaian proses. Konsol menampilkan maksimal 100 proses dan rangkaian. Hasilnya adalah gabungan dari proses dan rangkaian yang menggunakan CPU dan memori. Jika ada lebih dari 50 proses dan lebih dari 50 rangkaian, konsol akan menampilkan 50 pengonsumsi teratas di setiap kategori. Tampilan ini membantu Anda mengidentifikasi proses mana yang memiliki dampak paling besar pada performa.
- Proses RDS – Menampilkan ringkasan sumber daya yang digunakan oleh agen manajemen RDS, proses pemantauan diagnostik, dan proses AWS lain yang diperlukan untuk mendukung instans DB RDS.
- Proses OS – Menampilkan ringkasan proses sistem dan kernel, yang umumnya berdampak minimal pada performa.

Item yang tercantum untuk setiap proses meliputi:

- VIRT – Menampilkan ukuran virtual proses.
- RES – Menampilkan memori fisik aktual yang sedang digunakan oleh proses.
- CPU% – Menampilkan persentase total bandwidth CPU yang sedang digunakan oleh proses.
- CPU% – Menampilkan persentase total memori yang sedang digunakan oleh proses.

Data pemantauan yang ditampilkan di konsol RDS diambil dari Log Amazon CloudWatch. Anda juga dapat mengambil metrik untuk instans DB sebagai log stream dari Log CloudWatch. Untuk informasi selengkapnya, lihat Melihat metrik OS menggunakan Log CloudWatch.

Metrik Pemantauan yang Disempurnakan tidak ditampilkan selama:

- Failover instans DB.
- Mengubah kelas instans dari instans DB (komputasi skala).

Metrik Pemantauan yang Disempurnakan ditampilkan selama proses reboot instans DB karena hanya mesin basis data yang di-reboot. Metrik untuk sistem operasi tetap dilaporkan.

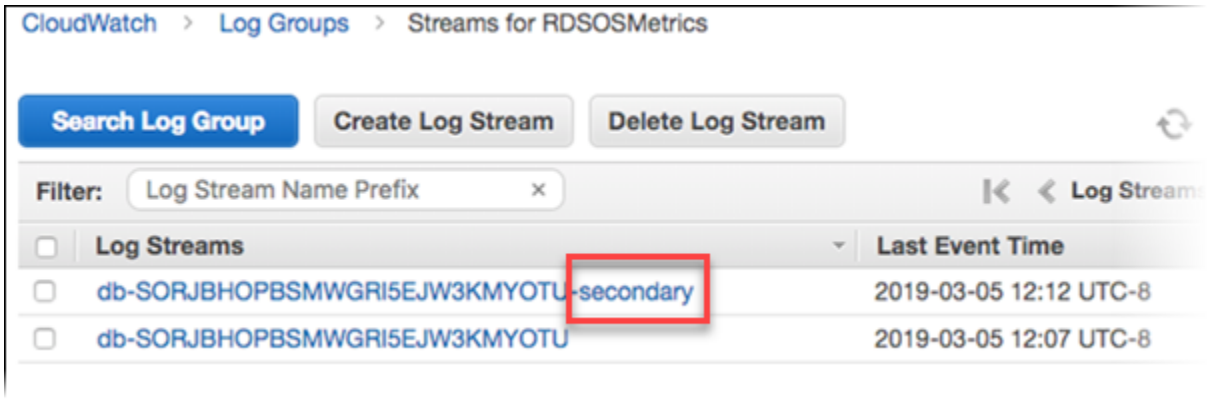
Melihat metrik OS menggunakan Log CloudWatch

Setelah mengaktifkan Pemantauan yang Disempurnakan untuk instans DB atau klaster DB Multi-AZ, Anda dapat melihat metrik menggunakan Log CloudWatch, dengan setiap log stream yang mewakili satu instans DB atau klaster DB yang dipantau. Pengidentifikasi log stream adalah pengidentifikasi sumber daya (`DbiResourceId`) untuk instans DB atau klaster DB.

Untuk melihat data log Pemantauan yang Disempurnakan

1. Buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.
2. Jika perlu, pilih Wilayah AWS tempat instans DB atau klaster DB Multi-AZ Anda berada. Untuk informasi selengkapnya, lihat Wilayah dan titik akhir dalam Referensi Umum Amazon Web Services.
3. Pilih Log di panel navigasi.
4. Pilih RDSOSMetrics dari daftar grup log.

   Dalam deployment instans DB Multi-AZ, file log dengan penambahan `-secondary` ke nama tersebut adalah untuk replika siaga Multi-AZ.



5. Pilih log stream yang ingin dilihat dari daftar log stream.

Referensi metrik untuk Amazon RDS

Dalam referensi ini, Anda dapat menemukan deskripsi metrik-metrik Amazon RDS untuk Amazon CloudWatch, Wawasan Performa, dan Pemantauan Disempurnakan.

Topik

CloudWatch Metrik Amazon untuk Amazon RDS

Amazon RDS menerbitkan metrik ke Amazon CloudWatch di ruang nama dan ruang nama. `AWS/RDS` `AWS/Usage`

Topik

Metrik CloudWatch tingkat instans Amazon untuk Amazon RDS

`AWS/RDS`Namespace di Amazon CloudWatch menyertakan metrik tingkat instans berikut.

> **Note**
>
> Konsol Amazon RDS mungkin menampilkan metrik dalam unit yang berbeda dari unit yang dikirim ke Amazon. CloudWatch Misalnya, konsol Amazon RDS mungkin menampilkan metrik dalam megabyte (MB), sedangkan metrik dikirim ke Amazon CloudWatch dalam byte.

Metrik	Deskripsi	Berlaku untuk	Unit
`BinLogDiskUsage`	Jumlah ruang disk yang ditempati oleh log biner. Jika pencadangan otomatis untuk instans MySQL dan MariaDB diaktifkan, termasuk replika baca, log biner akan dibuat.	MariaDB MySQL	Byte
`BurstBalance`	Persentase kredit I/O lonjakan bucket SSD Tujuan Umum (gp2) yang tersedia.	Semua	Persen
`CheckpointLag`	Jumlah waktu sejak titik pemeriksaan terakhir.		Detik
`ConnectionAttempts`	Jumlah percobaan untuk terhubung ke sebuah instans, baik berhasil maupun tidak.		Jumlah
`CPUUtilization`	Persentase penggunaan CPU.	Semua	Persentase
`CPUCreditUsage`	Jumlah kredit CPU yang digunakan oleh instans untuk pemanfaatan CPU. Satu kredit CPU sama dengan satu vCPU yang berjalan pada penggunaan 100 persen selama satu menit atau kombinasi vCPU, penggunaan, dan waktu yang setara. Sebagai contoh, Anda mungkin memiliki satu vCPU yang berjalan dengan penggunaan 50 persen selama dua menit atau dua vCPU yang berjalan dengan penggunaan 25 persen selama dua menit. Metrik ini hanya berlaku untuk`db.t2`,`db.t3`, dan `db.t4g` instance.		Kredit (vCPU-menit)

Metrik	Deskripsi	Berlaku untuk	Unit
	Note Kami merekomendasikan penggunaan kelas instans DB T hanya untuk server pengembangan dan pengujian, atau server nonproduksi lainnya. Untuk detail lebih lanjut tentang kelas instance T, lihat Jenis kelas instans DB Metrik kredit CPU hanya tersedia dalam frekuensi lima menit. Jika Anda menentukan periode lebih dari lima menit, gunakan statistik `Sum`, bukan `Average`.		

Metrik	Deskripsi	Berlaku untuk	Unit
`CPUCredit Balance`	Jumlah kredit CPU yang diperoleh yang diakumulasi oleh instans sejak diluncurkan atau dimulai. Untuk T2 Standar, `CPUCreditBalance` juga mencakup jumlah kredit peluncuran yang telah diakumulasi. Kredit diakumulasi ke saldo kredit setelah diperoleh, dan dihapus dari saldo kredit saat digunakan. Saldo kredit memiliki batas maksimum, yang ditentukan oleh ukuran instans. Setelah batas ini tercapai, semua kredit baru yang diperoleh akan dibuang. Untuk T2 Standar, kredit peluncuran tidak termasuk dalam penghitungan batas. Kredit dalam `CPUCreditBalance` dapat digunakan instans hingga melebihi penggunaan CPU dasar. Saat instans berjalan, kredit dalam `CPUCreditBalance` tidak akan kedaluwarsa. Saat instans berhenti, `CPUCreditBalance` tidak bertahan, dan semua akumulasi kredit akan hilang. Metrik kredit CPU hanya tersedia dalam frekuensi lima menit. Metrik ini hanya berlaku untuk`db.t2`,`db.t3`, dan `db.t4g` instance.		Kredit (Menit vCPU)

Metrik	Deskripsi	Berlaku untuk	Unit
	Note Kami merekomendasikan penggunaan kelas instans DB T hanya untuk server pengembangan dan pengujian, atau server nonproduksi lainnya. Untuk detail lebih lanjut tentang kelas instance T, lihat Jenis kelas instans DB Kredit peluncuran di Amazon RDS bekerja dengan cara yang sama seperti di Amazon EC2. Untuk informasi selengkapnya, lihat Kredit peluncuran dalam Panduan Pengguna Amazon Elastic Compute Cloud untuk Instans Linux.		

Metrik	Deskripsi	Berlaku untuk	Unit
`CPUSurplusCreditBalance`	Jumlah kredit surplus yang telah digunakan oleh instans tak terbatas saat nilai `CPUCreditBalance` miliknya adalah nol. Nilai `CPUSurplusCreditBalance` dibayarkan oleh kredit CPU yang diperoleh. Jika jumlah kredit surplus melebihi jumlah kredit maksimum yang dapat diperoleh instans dalam jangka waktu 24 jam, kredit surplus yang digunakan melebihi jumlah maksimum akan dikenakan biaya tambahan. Metrik kredit CPU hanya tersedia dengan frekuensi 5 menit.	Semua	Kredit (Menit vCPU)

Metrik	Deskripsi	Berlaku untuk	Unit
`CPUSurplusCreditsCharged`	Jumlah kredit surplus yang digunakan yang tidak dibayarkan oleh kredit CPU yang diperoleh, dan dengan demikian menimbulkan biaya tambahan. Kredit surplus yang digunakan dikenakan tagihan jika salah satu dari hal berikut terjadi: • Kredit surplus yang digunakan melebihi jumlah kredit maksimum yang dapat diperoleh instans dalam periode 24 jam. Kredit surplus yang digunakan di atas jumlah maksimum akan ditagihkan pada akhir jam. • Instans dihentikan atau diakhiri. • Instans dialihkan dari `unlimited` ke `standard`. Metrik kredit CPU hanya tersedia pada frekuensi 5 menit.	Semua	Kredit (Menit vCPU)

Metrik	Deskripsi	Berlaku untuk	Unit
`DatabaseConnections`	Jumlah koneksi jaringan klien ke instans basis data. Jumlah sesi basis data bisa lebih tinggi dari nilai metrik karena nilai metrik tidak termasuk yang berikut: • Sesi yang tidak lagi memiliki koneksi jaringan tetapi basis data belum dikosongkan • Sesi yang dibuat oleh mesin basis data untuk tujuannya sendiri • Sesi yang dibuat oleh kemampuan eksekusi paralel mesin basis data • Sesi yang dibuat oleh penjadwal pekerjaan mesin basis data • Koneksi Amazon RDS	Semua	Jumlah
`DiskQueueDepth`	Jumlah I/O (permintaan baca/tulis) tertunda yang menunggu untuk mengakses disk.	Semua	Jumlah
`DiskQueueDepthLogVolume`	Jumlah I/O (permintaan baca/tulis) tertunda yang menunggu untuk mengakses disk volume log.	Semua	Jumlah

Metrik	Deskripsi	Berlaku untuk	Unit
`EBSByteBalance%`	Persentase sisa kredit throughput dalam bucket lonjakan basis data RDS Anda. Metrik ini hanya tersedia untuk pemantauan dasar. Nilai metrik didasarkan pada throughput semua volume, termasuk volume root, dan bukan hanya pada volume yang berisi file basis data. Untuk menemukan instans yang mendukung metrik ini, lihat ukuran instans dengan tanda bintang (*) di tabel EBS dioptimalkan secara default dalam Panduan Pengguna Amazon EC2 untuk Instans Linux. Statistik `Sum` tidak berlaku untuk metrik ini.	Semua	Persentase

Metrik	Deskripsi	Berlaku untuk	Unit
`EBSIOBalance%`	Persentase sisa kredit I/O dalam bucket lonjakan basis data RDS Anda. Metrik ini hanya tersedia untuk pemantauan dasar. Nilai metrik didasarkan pada IOPS semua volume, termasuk volume root, dan bukan hanya pada volume yang berisi file basis data. Untuk menemukan instans yang mendukung metrik ini, lihat ukuran instans dengan tanda bintang (*) di tabel EBS dioptimalkan secara default dalam Panduan Pengguna Amazon EC2 untuk Instans Linux. Statistik `Sum` tidak berlaku untuk metrik ini. Metrik ini berbeda dari `BurstBalance` . Untuk mempelajari cara menggunakan metrik ini, lihat Meningkatkan performa aplikasi dan mengurangi biaya dengan kapabilitas lonjakan Instans yang Dioptimalkan Amazon EBS.	Semua	Persentase
`FailedSQLServerAgentJobsCount`	Jumlah pekerjaan Microsoft SQL Server Agent yang gagal pada menit terakhir.	Microsoft SQL Server	Jumlah per menit

Metrik	Deskripsi	Berlaku untuk	Unit
`FreeableMemory`	Jumlah memori akses acak yang tersedia. Untuk instans DB MariaDB, MySQL, Oracle, dan PostgreSQL, metrik ini melaporkan nilai bidang `MemAvailable` dari `/proc/meminfo` .	Semua	Byte
`FreeStorageSpace`	Jumlah ruang penyimpanan yang tersedia.	Semua	Byte
`FreeStorageSpaceLogVolume`	Jumlah ruang penyimpanan yang tersedia pada volume log.	Semua	Byte
`MaximumUsedTransactionIDs`	ID transaksi maksimum yang telah digunakan.	PostgreSQL	Jumlah
`NetworkReceiveThroughput`	Lalu lintas jaringan masuk (penerimaan) pada instans DB, termasuk lalu lintas basis data pelanggan dan lalu lintas Amazon RDS yang digunakan untuk pemantauan dan replikasi.	Semua	Byte per detik
`NetworkTransmitThroughput`	Lalu lintas jaringan keluar (transmit) pada instans DB, termasuk lalu lintas basis data pelanggan dan lalu lintas Amazon RDS yang digunakan untuk pemantauan dan replikasi.	Semua	Byte per detik
`OldestReplicationSlotLag`	Ukuran keterlambatan replika yang paling lambat dalam hal data write-ahead log (WAL) yang diterima.	PostgreSQL	Byte

Metrik	Deskripsi	Berlaku untuk	Unit
`ReadIOPS`	Jumlah rata-rata operasi I/O baca disk per detik.	Semua	Jumlah per detik
`ReadIOPSLogVolume`	Jumlah rata-rata operasi I/O baca disk per detik untuk volume log.	Semua	Jumlah per detik
`ReadLatency`	Jumlah rata-rata waktu yang diperlukan per operasi I/O disk.	Semua	Detik
`ReadLatencyLogVolume`	Jumlah rata-rata waktu yang diperlukan per operasi I/O disk untuk volume log operasi.	Semua	Detik
`ReadThroughput`	Jumlah rata-rata byte yang dibaca dari disk per detik.	Semua	Byte per detik
`ReadThroughputLogVolume`	Jumlah rata-rata byte yang dibaca dari disk per detik untuk volume log.	Semua	Byte per detik
`ReplicaLag`	Untuk konfigurasi replika baca, jumlah waktu keterlambatan instans DB replika baca di belakang instans DB sumber. Berlaku untuk replika baca MariaDB, Microsoft SQL Server, MySQL, Oracle, dan PostgreSQL. Untuk klaster DB Multi-AZ, perbedaan waktu antara transaksi terakhir pada instans DB penulis dan transaksi yang terakhir diterapkan pada instans DB pembaca.		Detik

Metrik	Deskripsi	Berlaku untuk	Unit
`ReplicationChannelLag`	Untuk konfigurasi replika multi-sumber, jumlah waktu saluran tertentu pada replika multi-sumber tertinggal di belakang instans DB sumber. Untuk informasi selengkapnya, lihat the section called “Memantau saluran replikasi multi-sumber”.	MySQL	Detik
`ReplicationSlotDiskUsage`	Ruang disk yang digunakan oleh file slot replikasi.	PostgreSQL	Byte
`SwapUsage`	Jumlah ruang swap yang digunakan pada instans DB.	MariaDB MySQL Oracle PostgreSQL	Byte
`TransactionLogsDiskUsage`	Ruang disk yang digunakan oleh log transaksi.	PostgreSQL	Byte
`TransactionLogsGeneration`	Ukuran log transaksi yang dihasilkan per detik.	PostgreSQL	Byte per detik
`WriteIOPS`	Jumlah rata-rata operasi I/O tulis disk per detik.	Semua	Jumlah per detik
`WriteIOPSLogVolume`	Jumlah rata-rata operasi I/O tulis disk per detik untuk volume log.	Semua	Jumlah per detik
`WriteLatency`	Jumlah rata-rata waktu yang diperlukan per operasi I/O disk.	Semua	Detik

Metrik	Deskripsi	Berlaku untuk	Unit
`WriteLatencyLogVolume`	Jumlah rata-rata waktu yang diperlukan per operasi I/O disk untuk volume log operasi.	Semua	Detik
`WriteThroughput`	Jumlah rata-rata byte yang ditulis ke disk per detik.	Semua	Byte per detik
`WriteThroughputLogVolume`	Jumlah rata-rata byte yang ditulis ke disk per detik untuk volume log.	Semua	Byte per detik

`AWS/Usage`Namespace di Amazon CloudWatch menyertakan metrik penggunaan tingkat akun untuk kuota layanan Amazon RDS Anda. CloudWatch mengumpulkan metrik penggunaan secara otomatis untuk semua. Wilayah AWS

Untuk informasi selengkapnya, lihat metrik CloudWatch penggunaan di Panduan CloudWatch Pengguna Amazon. Untuk informasi selengkapnya tentang kuota, lihat Kuota dan batasan untuk Amazon RDS dan Meminta peningkata kuota di Panduan Pengguna Kuota Layanan.

Metrik	Deskripsi	Unit:
`AllocatedStorage`	Total penyimpanan untuk semua instans DB. Jumlah tersebut tidak termasuk instans migrasi sementara.	Gigabyte
`DBClusterParameterGroups`	Jumlah grup parameter klaster DB di Akun AWS Anda. Hitungan tidak termasuk grup parameter default.	Hitungan
`DBClusters`	Jumlah klaster DB Amazon Aurora di Akun AWS Anda.	Hitungan
`DBInstances`	Jumlah instans DB di Akun AWS Anda.	Hitungan
`DBParameterGroups`	Jumlah grup parameter DB diAkun AWS Anda. Hitungan tidak termasuk grup parameter DB default.	Hitungan

Metrik	Deskripsi	Unit:
DBSecurityGroups	Jumlah grup keamanan di Akun AWS Anda. Hitungan tidak termasuk grup keamanan default dan grup keamanan VPC default.	Hitungan
DBSubnetGroups	Jumlah grup subnet DB di Akun AWS Anda. Hitungan tidak termasuk grup subnet default.	Hitungan
ManualClusterSnapshots	Jumlah snapshot klaster DB yang dibuat secara manual di Akun AWS Anda. Hitungan tidak termasuk snapshot yang tidak valid.	Hitungan
ManualSnapshots	Jumlah snapshot DB yang dibuat secara manual di Akun AWS Anda. Hitungan tidak termasuk snapshot yang tidak valid.	Hitungan
OptionGroups	Jumlah grup opsi di Akun AWS Anda. Hitungan tidak termasuk grup opsi default.	Hitungan
ReservedDBInstances	Jumlah instans DB yang dicadangkan di Akun AWS Anda. Hitungan tidak termasuk instans yang dihentikan atau ditolak.	Hitungan

* Amazon RDS tidak mempublikasikan unit untuk metrik penggunaan. CloudWatch Unit hanya muncul di dokumentasi.

Dimensi-dimensi Amazon CloudWatch untuk Amazon RDS

Anda dapat memfilter data metrik Amazon RDS dengan menggunakan dimensi apa pun dalam tabel berikut.

Dimensi	Memfilter data yang diminta terhadap . . .
DBInstanceIdentifier	Instans basis data tertentu.

Dimensi	Memfilter data yang diminta terhadap . . .
`DatabaseClass`	Semua instans dalam kelas basis data. Misalnya, Anda dapat mengagregasikan metrik-metrik untuk semua instans kelas basis data `db.r5.large` .
`EngineName`	Hanya nama mesin diidentifikasi. Misalnya, Anda dapat mengagregasikan metrik-metrik untuk semua instans yang memiliki nama mesin `postgres`.
`SourceRegion`	Hanya Kawasan yang ditentukan. Misalnya, Anda dapat mengagregasikan metrik-metrik untuk semua instans basis data dalam Kawasan `us-east-1` .

CloudWatch Metrik Amazon untuk Performance Insights

Performance Insights secara otomatis menerbitkan beberapa metrik ke Amazon. CloudWatch Data yang sama dapat ditanyakan dari Performance Insights, tetapi memiliki metrik CloudWatch memudahkan untuk menambahkan alarm. CloudWatch Ini juga memudahkan untuk menambahkan metrik ke CloudWatch Dasbor yang ada.

Metrik	Deskripsi
DBLoad	Jumlah sesi aktif untuk mesin DB. Biasanya, Anda menginginkan data untuk jumlah rata-rata sesi aktif. Dalam Wawasan Performa, data ini dikueri sebagai `db.load.avg` .
DBLoadCPU	Jumlah sesi aktif dengan jenis peristiwa tunggu berupa CPU. Dalam Wawasan Performa, data ini dikueri sebagai `db.load.avg` , difilter berdasarkan jenis peristiwa tunggu `CPU`.
LoadNonCPU DB	Jumlah sesi aktif dengan jenis peristiwa tunggu bukan CPU.

> **Note**
>
> Metrik ini dipublikasikan CloudWatch hanya jika ada beban pada instans DB.

Anda dapat memeriksa metrik ini menggunakan CloudWatch konsol, the AWS CLI, atau CloudWatch API. Anda juga dapat memeriksa metrik penghitung Performance Insights lainnya menggunakan fungsi matematika metrik khusus. Untuk informasi selengkapnya, lihat Menanyakan metrik penghitung Performance Insights lainnya di CloudWatch.

Misalnya, Anda bisa mendapatkan statistik untuk `DBLoad` metrik dengan menjalankan get-metric-statisticsperintah.

```
aws cloudwatch get-metric-statistics \
    --region us-west-2 \
    --namespace AWS/RDS \
    --metric-name DBLoad  \
    --period 60 \
    --statistics Average \
    --start-time 1532035185 \
    --end-time 1532036185 \
    --dimensions Name=DBInstanceIdentifier,Value=db-loadtest-0
```

Contoh ini menghasilkan output yang terlihat seperti berikut.

```
{
  "Datapoints": [
  {
  "Timestamp": "2021-07-19T21:30:00Z",
  "Unit": "None",
  "Average": 2.1
  },
  {
  "Timestamp": "2021-07-19T21:34:00Z",
  "Unit": "None",
  "Average": 1.7
  },
  {
  "Timestamp": "2021-07-19T21:35:00Z",
  "Unit": "None",
  "Average": 2.8
```

```
},
{
"Timestamp": "2021-07-19T21:31:00Z",
"Unit": "None",
"Average": 1.5
},
{
"Timestamp": "2021-07-19T21:32:00Z",
"Unit": "None",
"Average": 1.8
},
{
"Timestamp": "2021-07-19T21:29:00Z",
"Unit": "None",
"Average": 3.0
},
{
"Timestamp": "2021-07-19T21:33:00Z",
"Unit": "None",
"Average": 2.4
}
],
"Label": "DBLoad"
}
```


Untuk informasi selengkapnya CloudWatch, lihat Apa itu Amazon CloudWatch? di Panduan CloudWatch Pengguna Amazon.

Menanyakan metrik penghitung Performance Insights lainnya di CloudWatch

Anda dapat melakukan kueri, alarm, dan grafik pada metrik Performance Insights RDS. CloudWatch Anda dapat mengakses informasi tentang instans DB Anda dengan menggunakan fungsi matematika `DB_PERF_INSIGHTS` metrik untuk CloudWatch. Fungsi ini memungkinkan Anda menggunakan metrik Performance Insights yang tidak dilaporkan secara langsung CloudWatch untuk membuat deret waktu baru.

Anda dapat menggunakan fungsi Matematika Metrik baru dengan mengklik menu tarik-turun Tambahkan Matematika di layar Select metric di CloudWatch konsol. Anda dapat menggunakannya untuk membuat alarm dan grafik pada metrik Performance Insights atau pada kombinasi dan metrik CloudWatch Performance Insights, termasuk alarm resolusi tinggi untuk metrik sub-menit. Anda juga dapat menggunakan fungsi secara terprogram dengan menyertakan ekspresi Matematika Metrik

dalam permintaan. `get-metric-data` Untuk informasi selengkapnya, lihat Sintaks dan fungsi matematika metrik dan Membuat alarm pada metrik penghitung Performance Insights dari database. AWS

Metrik penghitung Wawasan Performa

Metrik penghitung adalah metrik performa sistem operasi dan basis data di dasbor Wawasan Performa. Untuk membantu mengidentifikasi dan menganalisis masalah performa, Anda dapat mengaitkan metrik penghitung dengan muatan DB. Anda dapat menambahkan fungsi statistik ke metrik untuk mendapatkan nilai metrik. Misalnya, fungsi yang didukung untuk metrik `os.memory.active` adalah `.avg`, `.min`, `.max`, `.sum`, dan `.sample_count`.

Metrik penghitung dikumpulkan sekali setiap menit. Kumpulan metrik OS bergantung pada apakah Pemantauan yang Ditingkatkan diaktifkan atau dinonaktifkan. Jika Pemantauan yang Ditingkatkan dinonaktifkan, metrik OS dikumpulkan sekali setiap menit. Jika Pemantauan yang Ditingkatkan diaktifkan, metrik OS dikumpulkan untuk periode waktu yang dipilih. Untuk informasi selengkapnya tentang cara mengaktifkan atau menonaktifkan Pemantauan yang Ditingkatkan, lihat Mengaktifkan dan menonaktifkan Pemantauan yang Ditingkatkan.

Topik

- Penghitung sistem operasi Wawasan Performa
- Penghitung Wawasan Performa untuk Amazon RDS for MariaDB dan MySQL
- Penghitung Wawasan Performa untuk Amazon RDS for Microsoft SQL Server
- Penghitung Wawasan Performa untuk Amazon RDS for Oracle
- Penghitung Wawasan Performa untuk Amazon RDS for PostgreSQL

Penghitung sistem operasi Wawasan Performa

Penghitung sistem operasi berikut, yang diawali dengan `os`, tersedia di Wawasan Performa untuk semua mesin RDS kecuali RDS for SQL Server .

Anda dapat menggunakan API `ListAvailableResourceMetrics` untuk mengetahui daftar metrik penghitung yang tersedia untuk instans DB Anda. Untuk informasi selengkapnya, lihat ListAvailableResourceMetricsdi panduan Referensi API Amazon RDS Performance Insights.

Penghitung	Jenis	Metrik	Deskripsi
Aktif	Memori	os.memory.active	Jumlah memori yang ditetapkan, dalam kilobyte.
Buffer	Memori	os.memory.buffers	Jumlah memori yang digunakan untuk buffering permintaa n I/O sebelum menulis ke perangkat penyimpanan, dalam kilobyte.
Di-cache	Memori	os.memory.cached	Jumlah memori yang digunakan untuk meng-cache I/O berbasis sistem file, dalam kilobyte.
Cache DB	Memori	os.memory.db.cache	Jumlah memori yang digunakan untuk cache halaman oleh proses basis data termasuk tmpfs (shmem), dalam byte.
Ukuran Set Residen DB	Memori	os.memory.db. residentSetSize	Jumlah memori yang digunakan untuk cache anonim dan swap oleh proses basis data tidak termasuk tmpfs (shmem), dalam byte.
Swap DB	Memori	os.memory.db.swap	Jumlah memori yang digunakan untuk swap berdasarkan proses

Penghitung	Jenis	Metrik	Deskripsi
			basis data, dalam byte.
Kotor	Memori	os.memory.dirty	Jumlah halaman memori dalam RAM yang telah diubah, tetapi tidak ditulis ke blok data terkaitnya dalam penyimpanan, dalam kilobyte.
Kosong	Memori	os.memory.free	Jumlah memori yang tidak ditetapkan, dalam kilobyte.
Halaman Besar Kosong	Memori	os.memori. hugePagesFree	Jumlah halaman besar yang kosong. Halaman besar adalah fitur dari kernel Linux.
Rsvd Halaman Besar	Memori	os.memori. hugePagesRsvd	Jumlah halaman besar yang khusus.
Ukuran Halaman Besar	Memori	os.memori. hugePagesSize	Ukuran untuk setiap unit halaman besar, dalam kilobyte.
Surplus Halaman Besar	Memori	os.memori. hugePagesSurp	Jumlah halaman besar surplus yang tersedia terhadap total.
Total Halaman Besar	Memori	os.memori. hugePagesTotal	Jumlah total halaman besar.

Penghitung	Jenis	Metrik	Deskripsi
Nonaktif	Memori	os.memory.inactive	Jumlah halaman memori yang jarang digunakan, dalam kilobyte.
Dipetakan	Memori	os.memory.mapped	Jumlah total konten sistem file yang memorinya dipetakan di dalam ruang alamat proses, dalam kilobyte.
Jumlah Kill Kehabisan Memori	Memori	os.memori. outOfMemoryKillCount	Jumlah kill OOM yang terjadi selama interval pengumpulan terakhir.
Tabel Halaman	Memori	os.memory.pageTables	Jumlah memori yang digunakan berdasarkan tabel halaman, dalam kilobyte.
Lempengan	Memori	os.memory.slab	Jumlah struktur data kernel yang dapat digunakan kembali, dalam kilobyte.
Total	Memori	os.memory.total	Jumlah total memori, dalam kilobyte.
Writeback	Memori	os.memory.writeback	Jumlah halaman kotor dalam RAM yang masih ditulis ke penyimpanan cadangan, dalam kilobyte.

Penghitung	Jenis	Metrik	Deskripsi
Tamu	Penggunaan CPU	os.cpuUtilization.guest	Persentase CPU yang digunakan oleh program tamu.
Idle	Penggunaan CPU	os.cpuUtilization.idle	Persentase CPU saat idle.
Irq	Penggunaan CPU	os.cpuUtilization.irq	Persentase CPU yang digunakan oleh gangguan perangkat lunak.
Nice	Penggunaan CPU	os.cpuUtilization.nice	Persentase CPU yang digunakan oleh program yang berjalan pada prioritas terendah.
Steal	Penggunaan CPU	os.cpuUtilization.steal	Persentase CPU yang digunakan oleh mesin virtual lainnya.
Sistem	Penggunaan CPU	os.cpuUtilization.system	Persentase CPU yang digunakan oleh kernel.
Total	Penggunaan CPU	os.cpuUtilization.total	Total persentase CPU yang digunakan. Nilai ini mencakup nilai nice.
Pengguna	Penggunaan CPU	os.cpuUtilization.user	Persentase CPU yang digunakan oleh program pengguna.

Penghitung	Jenis	Metrik	Deskripsi
Tunggu	Penggunaan CPU	os.cpuUtilization.wait	Persentase CPU yang tidak digunakan saat menunggu akses I/O.
PS IOs Baca	IO Disk	os.diskIO.<devicename>.readIOsPS	Jumlah operasi baca per detik.
PS IO Tulis	IO Disk	os.diskIO.<devicename>.writeIOsPS	Jumlah operasi tulis per detik.
Len Antrean Rata-rata	IO Disk	os.Diskio. <devicename>. avgQueueLen	Jumlah permintaan yang menunggu dalam antrean perangkat I/O.
Sz Req Rata-rata	IO Disk	os.Diskio. <devicename>. avgReqSz	Jumlah permintaan yang menunggu dalam antrean perangkat I/O.
Menunggu	IO Disk	os.diskIO.<devicename>.await	Jumlah milidetik yang diperlukan untuk merespons permintaan, termasuk waktu antrean dan waktu layanan.
PS IOs Baca	IO Disk	os.diskIO.<devicename>.readIOsPS	Jumlah operasi baca per detik.
KB Baca	IO Disk	os.diskIO.<devicename>.readKb	Jumlah total kilobyte yang dibaca.

Penghitung	Jenis	Metrik	Deskripsi
PS KB Baca	IO Disk	os.diskIO.<devicename>.readKbPS	Jumlah kilobyte yang dibaca per detik.
PS Rrqm	IO Disk	os.diskIO.<devicename>.rrqmPS	Jumlah permintaan baca gabungan yang diantrekan per detik.
TPS	IO Disk	os.diskIO.<devicename>.tps	Jumlah transaksi I/O per detik.
Utilitas	IO Disk	os.diskIO.<devicename>.util	Persentase waktu CPU saat permintaan dikeluarkan.
KB Tulis	IO Disk	os.diskIO.<devicename>.writeKb	Jumlah total kilobyte yang ditulis.
PS KB Tulis	IO Disk	os.diskIO.<devicename>.writeKbPS	Jumlah kilobyte yang ditulis per detik.
PS Wrqm	IO Disk	os.diskIO.<devicename>.wrqmPS	Jumlah permintaan tulis gabungan yang diantrekan per detik.
Diblokir	Tugas	os.tasks.blocked	Jumlah tugas yang diblokir.
Berjalan	Tugas	os.tasks.running	Jumlah tugas yang berjalan.
Tidur	Tugas	os.tasks.sleeping	Jumlah tugas yang tidur.

Penghitung	Jenis	Metrik	Deskripsi
Dihentikan	Tugas	os.tasks.stopped	Jumlah tugas yang dihentikan.
Total	Tugas	os.tasks.total	Jumlah total tugas.
Zombie	Tugas	os.tasks.zombie	Jumlah tugas turunan yang tidak aktif dengan tugas induk aktif.
Satu	Menit Rata-Rata Muatan	os. loadAvera geMinute.satu	Jumlah proses yang meminta waktu CPU selama satu menit terakhir.
Lima belas	Menit Rata-Rata Muatan	os. loadAvera geMinute.lima belas	Jumlah proses yang meminta waktu CPU selama 15 menit terakhir.
Lima	Menit Rata-Rata Muatan	os. loadAvera geMinute.lima	Jumlah proses yang meminta waktu CPU selama 5 menit terakhir.
Di-cache	Swap	os.swap.cached	Jumlah memori swap, dalam kilobyte, yang digunakan sebagai memori cache.
Kosong	Swap	os.swap.free	Jumlah memori swap yang kosong, dalam kilobyte.

Penghitung	Jenis	Metrik	Deskripsi
Masuk	Swap	os.swap.in	Jumlah memori, dalam kilobyte, yang ditukar ke dalam dari disk.
Keluar	Swap	os.swap.out	Jumlah memori, dalam kilobyte, yang ditukar ke luar dari disk.
Total	Swap	os.swap.total	Jumlah total memori swap yang tersedia dalam kilobyte.
File Maks	Sistem File	os.fileSys.maxFiles	Jumlah maksimum file yang dapat dibuat untuk sistem file.
File yang Digunakan	Sistem File	os.fileSys.usedFiles	Jumlah file dalam sistem file.
Persen File Digunakan	Sistem File	os.Filesys. usedFilePercent	Persentase file tersedia yang digunakan.
Persen Digunakan	Sistem File	os.fileSys.usedPercent	Persentase ruang disk sistem file yang digunakan.
Digunakan	Sistem File	os.fileSys.used	Jumlah ruang disk yang digunakan oleh file dalam sistem file, dalam kilobyte.

Penghitung	Jenis	Metrik	Deskripsi
Total	Sistem File	os.fileSys.total	Jumlah total ruang disk yang tersedia untuk sistem file, dalam kilobyte.
Rx	Jaringan	os.network.rx	Jumlah byte yang diterima per detik.
Tx	Jaringan	os.network.tx	Jumlah byte yang diunggah per detik.
Penggunaan Acu	Umum	os.general.acuUtilization	Persentase kapasitas saat ini dari kapasitas maksimum yang dikonfigurasi.
Acu Maks yang Dikonfigurasi	Umum	os.umum. maxConfiguredAcu	Kapasitas maksimum yang dikonfigurasi oleh pengguna, di ACU.
Acu Min yang Dikonfigurasi	Umum	os.umum. minConfiguredAcu	Kapasitas minimum yang dikonfigurasi oleh pengguna, di ACU.
Jumlah vCPU	Umum	os.general.numVCPUs	Jumlah CPU virtual untuk instans DB.
Kapasitas Basis Data Nirserver	Umum	os.umum. serverlessDatabaseCapacity	Kapasitas instans saat ini, dalam ACU.

Penghitung Wawasan Performa untuk Amazon RDS for MariaDB dan MySQL

Penghitung basis data berikut tersedia di Wawasan Performa untuk Amazon RDS for MariaDB dan MySQL.

Topik

- Penghitung native untuk RDS for MariaDB dan RDS for MySQL
- Penghitung non-native untuk Amazon RDS for MariaDB dan MySQL

Penghitung native untuk RDS for MariaDB dan RDS for MySQL

Metrik native ditentukan oleh mesin basis data, bukan Amazon RDS. Untuk definisi metrik native ini, lihat Server status variables dalam dokumentasi MySQL.

Penghitung	Jenis	Unit	Metrik
Com_analyze	SQL	Kueri per detik	db.SQL.Com_analyze
Com_optimize	SQL	Kueri per detik	db.SQL.Com_optimize
Com_select	SQL	Kueri per detik	db.SQL.Com_select
Koneksi	SQL	Jumlah upaya koneksi per menit (berhasil atau tidak) ke server MySQL	db.Users.Connections
Innodb_rows_deleted	SQL	Baris per detik	db.SQL.Innodb_rows_deleted
Innodb_rows_inserted	SQL	Baris per detik	db.SQL.Innodb_rows_inserted
Innodb_rows_read	SQL	Baris per detik	db.SQL.Innodb_rows_read

Penghitung	Jenis	Unit	Metrik
Innodb_rows_updated	SQL	Baris per detik	db.SQL.Innodb_rows_updated
Select_full_join	SQL	Kueri per detik	db.SQL.Select_full_join
Select_full_range_join	SQL	Kueri per detik	db.SQL.Select_full_range_join
Select_range	SQL	Kueri per detik	db.SQL.Select_range
Select_range_check	SQL	Kueri per detik	db.SQL.Select_range_check
Select_scan	SQL	Kueri per detik	db.SQL.Select_scan
Slow_queries	SQL	Kueri per detik	db.SQL.Slow_queries
Sort_merge_passes	SQL	Kueri per detik	db.SQL.Sort_merge_passes
Sort_range	SQL	Kueri per detik	db.SQL.Sort_range
Sort_rows	SQL	Kueri per detik	db.SQL.Sort_rows
Sort_scan	SQL	Kueri per detik	db.SQL.Sort_scan
Pertanyaan	SQL	Kueri per detik	db.SQL.Questions
Innodb_row_lock_time	Kunci	Milidetik (rata-rata)	db.Locks.Innodb_row_lock_time

Penghitung	Jenis	Unit	Metrik
Table_locks_immediate	Kunci	Permintaan per detik	db.Locks.Table_locks_immediate
Table_locks_waited	Kunci	Permintaan per detik	db.Locks.Table_locks_waited
Aborted_clients	Pengguna	Koneksi	db.Users.Aborted_clients
Aborted_connects	Pengguna	Koneksi	db.Users.Aborted_connects
max_connections	Pengguna	Koneksi	db.User.max_connections
Threads_created	Pengguna	Koneksi	db.Users.Threads_created
Threads_running	Pengguna	Koneksi	db.Users.Threads_running
Innodb_data_writes	I/O	Operasi per detik	db.IO.Innodb_data_writes
Innodb_dblwr_writes	I/O	Operasi per detik	db.IO.Innodb_dblwr_writes
Innodb_log_write_requests	I/O	Operasi per detik	db.IO.Innodb_log_write_requests
Innodb_log_writes	I/O	Operasi per detik	db.IO.Innodb_log_writes
Innodb_pages_written	I/O	Halaman per detik	db.IO.Innodb_pages_written
Created_tmp_disk_tables	Temp	Tabel per detik	db.Temp.Created_tmp_disk_tables
Created_tmp_tables	Temp	Tabel per detik	db.Temp.Created_tmp_tables

Penghitung	Jenis	Unit	Metrik
Innodb_buffer_pool_pages_data	Cache	Halaman	db.Cache.Innodb_buffer_pool_pages_data
Innodb_buffer_pool_pages_total	Cache	Halaman	db.Cache.Innodb_buffer_pool_pages_total
Innodb_buffer_pool_read_requests	Cache	Halaman per detik	db.Cache.Innodb_buffer_pool_read_requests
Innodb_buffer_pool_reads	Cache	Halaman per detik	db.Cache.Innodb_buffer_pool_reads
Opened_tables	Cache	Tabel	db.Cache.Opened_tables
Opened_table_definitions	Cache	Tabel	db.Cache.Opened_table_definitions
Qcache_hits	Cache	Kueri	db.Cache.Qcache_hits

Penghitung non-native untuk Amazon RDS for MariaDB dan MySQL

Metrik penghitung non-native adalah penghitung yang ditentukan oleh Amazon RDS. Metrik non-asli bisa berupa metrik yang Anda dapatkan dengan kueri tertentu. Metrik non-native juga bisa berupa metrik turunan, dengan dua penghitung native atau lebih digunakan dalam penghitungan untuk mengetahui rasio, tingkat hit, atau latensi.

Penghitung	Jenis	Metrik	Deskripsi	Definisi
innodb_buffer_pool_hits	Cache	db.Cache.innoDB_buffer_pool_hits	Jumlah bacaan yang dapat dipenuhi oleh InnoDB dari kumpulan buffer.	`innodb_buffer_pool_read_requests - innodb_buffer_pool_reads`

Penghitung	Jenis	Metrik	Deskripsi	Definisi
innodb_buffer_pool_hit_rate	Cache	db.Cache.innoDB_buffer_pool_hit_rate	Jumlah baca yang dapat dipenuhi oleh InnoDB dari kumpulan buffer.	`100 * innodb_buffer_pool_read_requests / (innodb_buffer_pool_read_requests + innodb_buffer_pool_reads)`

Penghitung	Jenis	Metrik	Deskripsi	Definisi
innodb_buffer_pool_usage	Cache	db.Cache.innoDB_buffer_pool_usage	Persentase kumpulan buffer InnoDB yang berisi data (halaman). **Note** Saat menggunakan tabel terkompresi, nilai ini bisa bervariasi. Untuk informasi selengkapnya, lihat informasi tentang `Innodb_buffer_pool_pages_data` dan `Innodb`	`Innodb_buffer_pool_pages_data / Innodb_buffer_pool_pages_total * 100.0`

Penghitung	Jenis	Metrik	Deskripsi	Definisi
			`ffer_p_pages` `tal` di Server status variables dalam dokumesi MySQL.	
query_cache_hit_rate	Cache	db.Cache.query_cache_hit_rate	Rasio hit cache set hasil MySQL (cache kueri).	`Qcache_hits / (QCache_hits + Com_select) * 100`
innodb_datafile_writes_to_disk	I/O	db.IO.innoDB_datafile_writes_to_disk	Jumlah penulisan file data InnoDB ke disk, tidak termasuk operasi tulis ganda dan tulis pencatatan log redo.	`Innodb_data_writes - Innodb_log_writes - Innodb_dblwr_writes`

Penghitung	Jenis	Metrik	Deskripsi	Definisi
innodb_rows_changed	SQL	db.SQL.innodb_rows_changed	Total operasi baris InnoDB.	`db.SQL.Innodb_rows_inserted + db.SQL.Innodb_rows_deleted + db.SQL.Innodb_rows_updated`
active_transactions	Transaksi	db.Transactions.active_transactions	Total transaksi aktif.	`SELECT COUNT(1) AS active_transactions FROM INFORMATION_SCHEMA.INNODB_TRX`

Penghitung	Jenis	Metrik	Deskripsi	Definisi
trx_rseg_history_len	Transaksi	db.Transactions.trx_rseg_history_len	Daftar halaman log undo untuk transaksi yang telah dijalankan yang dipertahankan oleh sistem transaksi InnoDB untuk menerapkan kontrol konkurensi multi-versi. Untuk informasi selengkapnya tentang detail data log undo, lihat https://dev.mysql.com/doc/refman/8.0/en/innodb-multi-versioning.html dalam dokumentasi MySQL.	`SELECT COUNT AS trx_rseg_history_len FROM INFORMATION_SCHEMA.INNODB_METRICS WHERE NAME='trx_rseg_history_len'`

Penghitung	Jenis	Metrik	Deskripsi	Definisi
innodb_deadlocks	Kunci	db.Locks.innodb_deadlocks	Jumlah total penguncian.	`SELECT COUNT AS innodb_deadlocks FROM INFORMATION_SCHEMA.INNODB_METRICS WHERE NAME='lock_deadlocks'`
innodb_lock_timeouts	Kunci	db.Locks.innodb_lock_timeouts	Jumlah total penguncian yang kehabisan waktu.	`SELECT COUNT AS innodb_lock_timeouts FROM INFORMATION_SCHEMA.INNODB_METRICS WHERE NAME='lock_timeouts'`
innodb_row_lock_waits	Kunci	db.Locks.innodb_row_lock_waits	Jumlah total kunci baris yang menghasilkan peristiwa tunggu.	`SELECT COUNT AS innodb_row_lock_waits FROM INFORMATION_SCHEMA.INNODB_METRICS WHERE NAME='lock_row_lock_waits'`

Penghitung Wawasan Performa untuk Amazon RDS for Microsoft SQL Server

Penghitung basis data berikut tersedia di Wawasan Performa untuk RDS for Microsoft SQL Server.

Penghitung native untuk RDS for Microsoft SQL Server

Metrik native ditentukan oleh mesin basis data, bukan Amazon RDS. Anda dapat menemukan definisi untuk metrik native ini dalam Use SQL Server Objects dalam dokumentasi Microsoft SQL Server.

Penghitung	Jenis	Unit	Metrik
Data yang Diteruskan	Metode Akses	Data per detik	Data db.Access Methods.Forwarded
Pembagian Halaman	Metode Akses	Pembagian per detik	Pembagian db.Access Methods.Page
Rasio hit cache buffer	Pengelola Buffer	Rasio	Rasio hit cache db.Buffer Manager.Buffer
Ekspektansi masa aktif halaman	Pengelola Buffer	Ekspektansi dalam hitungan detik	Ekspektansi masa aktif db.Buffer Manager.Page
Pencarian halaman	Pengelola Buffer	Pencarian per detik	Pencarian db.Buffer Manager.Page
Pembacaan halaman	Pengelola Buffer	Bacaan per detik	Pembacaan db.Buffer Manager.Page
Penulisan halaman	Pengelola Buffer	Penulisan per detik	Penulisan db.Buffer Manager.Page
Transaksi Aktif	Basis Data	Transaksi	Transaksi db.Databases.Active (_Total)
Byte Log Terbuang	Basis Data	Byte terbuang per detik	Byte db.Databases.Log Terbuang (_Total)

Penghitung	Jenis	Unit	Metrik
Tunggu Pembuangan Log	Basis Data	Tunggu per detik	Tunggu Pembuangan db.Databases.Log (_Total)
Pembuangan Log	Basis Data	Pembuangan per detik	Pembuangan db.Databases.Log (_Total)
Transaksi Tulis	Basis Data	Transaksi per detik	Transaksi db.Databases.Write (_Total)
Proses diblokir	Statistik Umum	Proses diblokir	db.General Statistics.Processes diblokir
Koneksi Pengguna	Statistik Umum	Koneksi	db.General Statistics.User Connections
Tunggu Latch	Latch	Tunggu per detik	Tunggu db.Latches.Latch
Jumlah Penguncian	Kunci	Penguncian per detik	db.Locks.Number Penguncian (_Total)
Pemberian Memori Tertunda	Manajer Memori	Pemberian memori	Pemberian db.Memory Manager.Memory Tertunda
Permintaan Batch	Statistik SQL	Permintaan per detik	Permintaan db.SQL Statistics.Batch
Kompilasi SQL	Statistik SQL	Kompilasi per detik	Kompilasi db.SQL Statistics.SQL
Kompilasi Ulang SQL	Statistik SQL	Kompilasi ulang per detik	Kompilasi ulang db.SQL Statistics.SQL

Penghitung Wawasan Performa untuk Amazon RDS for Oracle

Penghitung basis data berikut tersedia di Wawasan Performa untuk RDS for Oracle.

Penghitung native untuk RDS for Oracle

Metrik native ditentukan oleh mesin basis data, bukan Amazon RDS. Anda dapat menemukan definisi metrik native ini di Statistics Descriptions dalam dokumentasi Oracle.

> **Note**
>
> Untuk metrik penghitung `CPU used by this session`, unit telah diubah dari sentidetik native menjadi sesi aktif agar nilainya lebih mudah digunakan. Misalnya, pengiriman CPU dalam bagan Muatan DB menunjukkan permintaan CPU. Metrik penghitung `CPU used by this session` menunjukkan jumlah CPU yang digunakan oleh sesi Oracle. Anda dapat membandingkan pengiriman CPU ke metrik penghitung `CPU used by this session`. Ketika permintaan untuk CPU lebih tinggi dari CPU yang digunakan, sesi akan menunggu waktu CPU.

Penghitung	Jenis	Unit	Metrik
CPU yang digunakan oleh sesi ini	Pengguna	Sesi aktif	db.User.CPU yang digunakan oleh sesi ini
Roundtrip SQL*Net ke/dari klien	Pengguna	Roundtrip per detik	Roundtrip db.User.SQL*Net ke/dari klien
Byte yang diterima melalui SQL*Net dari klien	Pengguna	Byte per detik	db.User.bytes yang diterima melalui SQL*Net dari klien
Commit pengguna	Pengguna	Commit per detik	Commit db.User.user
Kumulatif logon	Pengguna	Logon per detik	Kumulatif db.User.logons

Penghitung	Jenis	Unit	Metrik
Panggilan pengguna	Pengguna	Panggilan per detik	Panggilan db.User.user
Byte yang dikirim melalui SQL*Net ke klien	Pengguna	Byte per detik	db.User.bytes yang dikirim melalui SQL*Net ke klien
Rollback pengguna	Pengguna	Pemulihan per detik	Rollback db.User.user
Ukuran redo	Redo	Byte per detik	Ukuran db.Redo.redo
Jumlah penguraian (total)	SQL	Penguraian per detik	Jumlah db.SQL.parse (total)
Jumlah penguraian (keras)	SQL	Penguraian per detik	Jumlah db.SQL.parse (keras)
Baris pindaian tabel yang diperoleh	SQL	Baris per detik	Baris pemindaian db.SQL.table yang diperoleh
Urutan (memori)	SQL	Urutan per detik	db.SQL.sorts (memori)
Urutan (disk)	SQL	Urutan per detik	db.SQL.sorts (disk)
Urutan (baris)	SQL	Urutan per detik	db.SQL.sorts (baris)
Byte pembacaan fisik	Cache	Byte per detik	Byte pembacaan db.Cache.physical
Perolehan blok DB	Cache	Blok per detik	Perolehan blok db.Cache.db
Titik pemeriksaan DBWR	Cache	Titik pemeriksaan per menit	Titik pemeriksaan db.Cache.DBWR

Penghitung	Jenis	Unit	Metrik
Pembacaan fisik	Cache	Bacaan per detik	Pembacaan db.Cache.physical
Perolehan konsisten dari cache	Cache	Perolehan per detik	Perolehan db.Cache. consistent dari cache
Perolehan blok DB dari cache	Cache	Perolehan per detik	Perolehan blok db.Cache.db dari cache
Perolehan konsisten	Cache	Perolehan per detik	Perolehan db.Cache. consistent

Penghitung Wawasan Performa untuk Amazon RDS for PostgreSQL

Penghitung basis data berikut tersedia di Wawasan Performa untuk Amazon RDS for PostgreSQL.

Topik

- Penghitung native untuk Amazon RDS for PostgreSQL
- Penghitung non-native untuk Amazon RDS for PostgreSQL

Penghitung native untuk Amazon RDS for PostgreSQL

Metrik native ditentukan oleh mesin basis data, bukan Amazon RDS. Anda dapat menemukan definisi untuk metrik native ini dalam Viewing Statistics dalam dokumentasi PostgreSQL.

Penghitung	Jenis	Unit	Metrik
blks_hit	Cache	Blok per detik	db.Cache.blks_hit
buffers_alloc	Cache	Blok per detik	db.Cache.buffers_alloc
buffers_checkpoint	Titik pemeriksaan	Blok per detik	db.Checkpoint.buffers_checkpoint

Penghitung	Jenis	Unit	Metrik
checkpoint_sync_time	Titik pemeriksaan	Milidetik per titik pemeriksaan	db.Checkpoint.checkpoint_sync_time
checkpoint_write_time	Titik pemeriksaan	Milidetik per titik pemeriksaan	db.Checkpoint.checkpoint_write_time
checkpoints_req	Titik pemeriksaan	Titik pemeriksaan per menit	db.Checkpoint.checkpoints_req
checkpoints_timed	Titik pemeriksaan	Titik pemeriksaan per menit	db.Checkpoint.checkpoints_timed
maxwritten_clean	Titik pemeriksaan	Penghentian pembersihan Bgwriter per menit	db.Checkpoint.maxwritten_clean
penguncian	Konkurensi	Penguncian per menit	db.Concurrency.deadlocks
blk_read_time	I/O	Milidetik	db.IO.blk_read_time
blks_read	I/O	Blok per detik	db.IO.blks_read
buffers_backend	I/O	Blok per detik	db.IO.buffers_backend
buffers_backend_fsync	I/O	Blok per detik	db.IO.buffers_backend_fsync
buffers_clean	I/O	Blok per detik	db.IO.buffers_clean
tup_deleted	SQL	Tuple per detik	db.SQL.tup_deleted
tup_fetched	SQL	Tuple per detik	db.SQL.tup_fetched
tup_inserted	SQL	Tuple per detik	db.SQL.tup_inserted

Penghitung	Jenis	Unit	Metrik
tup_returned	SQL	Tuple per detik	db.SQL.tup_returned
tup_updated	SQL	Tuple per detik	db.SQL.tup_updated
idle_in_transactio n_aborted_count	Status	Sesi	db.state.idle_in_transactio n_aborted_count
idle_in_transaction_count	Status	Sesi	db.state.idle_in_transaction_count
idle_in_transactio n_max_time	Status	Detik	db.state.idle_in_transactio n_max_time
temp_bytes	Temp	Byte per detik	db.Temp.temp_bytes
temp_files	Temp	File per menit	db.Temp.temp_files
active_transactions	Transaksi	Transaksi	db.Transactions.active_tran sactions
blocked_transactions	Transaksi	Transaksi	db.Transactions.blocked_tra nsactions
max_used_xact_ids	Transaksi	Transaksi	db.Transactions.max_used_xa ct_ids
xact_commit	Transaksi	Commit per detik	db.Transactions.xact_commit
xact_rollback	Transaksi	Pemulihan per detik	db.Transactions.xact_rollback
max_connections	Pengguna	Koneksi	db.User.max_connections
numbackends	Pengguna	Koneksi	db.User.numbackends
archived_count	Log write-ahead log (WAL)	File per menit	db.WAL.archived_count
archive_failed_count	WAL	File per menit	db.WAL.archive_failed_count

Penghitung non-native untuk Amazon RDS for PostgreSQL

Metrik penghitung non-native adalah penghitung yang ditentukan oleh Amazon RDS. Metrik non-asli bisa berupa metrik yang Anda dapatkan dengan kueri tertentu. Metrik non-native juga bisa berupa metrik turunan, dengan dua penghitung native atau lebih digunakan dalam penghitungan untuk mengetahui rasio, tingkat hit, atau latensi.

Penghitung	Jenis	Metrik	Deskripsi	Definisi
checkpoint_sync_latency	Titik pemeriksaan	db.Checkpoint.checkpoint_sync_latency	Jumlah total waktu yang telah dihabiskan di bagian pemrosesan titik pemeriksaan tempat file disinkronkan ke disk.	`checkpoint_sync_time / (checkpoints_timed + checkpoints_req)`
checkpoint_write_latency	Titik pemeriksaan	db.Checkpoint.checkpoint_write_latency	Jumlah total waktu yang telah dihabiskan di bagian pemrosesan titik pemeriksaan tempat file ditulis ke disk.	`checkpoint_write_time / (checkpoints_timed + checkpoints_req)`
read_latency	I/O	db.IO.read_latency	Waktu yang dihabiskan untuk membaca blok file data oleh backend dalam instans ini.	`blk_read_time / blks_read`

Statistik SQL untuk Wawasan Performa

Statistik SQL adalah metrik terkait performa tentang kueri SQL yang dikumpulkan oleh Wawasan Performa. Wawasan Performa mengumpulkan statistik untuk setiap detik yang digunakan untuk menjalankan kueri dan untuk setiap panggilan SQL. Statistik SQL adalah rata-rata untuk rentang waktu yang dipilih.

Digest SQL adalah gabungan semua kueri yang memiliki pola tertentu, tetapi tidak harus memiliki nilai literal yang sama. Digest menggantikan nilai literal dengan tanda tanya. Misalnya, `SELECT * FROM emp WHERE lname= ?`. Digest ini mungkin terdiri dari kueri turunan berikut:

```
SELECT * FROM emp WHERE lname = 'Sanchez'
SELECT * FROM emp WHERE lname = 'Olagappan'
SELECT * FROM emp WHERE lname = 'Wu'
```

Semua mesin mendukung statistik SQL untuk kueri digest.

Untuk informasi dukungan wilayah, mesin DB, dan kelas instans untuk fitur ini, lihat Dukungan kelas instans, Wilayah, dan mesin DB Amazon RDS untuk fitur Wawasan Performa

Topik

- Statistik SQL untuk MariaDB dan MySQL
- Statistik SQL untuk Oracle
- Statistik SQL untuk SQL Server
- Statistik SQL untuk RDS PostgreSQL

Statistik SQL untuk MariaDB dan MySQL

MariaDB dan MySQL mengumpulkan statistik SQL hanya pada tingkat digest. Tidak ada statistik yang ditampilkan di tingkat pernyataan.

Topik

- Statistik digest untuk MariaDB dan MySQL
- Statistik per detik untuk MariaDB dan MySQL
- Statistik per panggilan untuk MariaDB dan MySQL

Statistik digest untuk MariaDB dan MySQL

Wawasan Performa mengumpulkan statistik digest SQL dari tabel `events_statements_summary_by_digest`. Tabel `events_statements_summary_by_digest` dikelola oleh basis data Anda.

Tabel digest tidak memiliki kebijakan pengosongan. Jika tabel penuh, AWS Management Console menunjukkan pesan berikut:

```
Performance Insights is unable to collect SQL Digest statistics on new queries because
 the table events_statements_summary_by_digest is full.
Please truncate events_statements_summary_by_digest table to clear the issue. Check the
 User Guide for more details.
```

Dalam situasi ini, MariaDB dan MySQL tidak melacak kueri SQL. Untuk mengatasi masalah ini, Wawasan Performa secara otomatis memotong tabel digest jika kedua kondisi berikut terpenuhi:

- Tabel penuh.
- Wawasan Performa mengelola Skema Performa secara otomatis.

  Untuk manajemen otomatis, parameter `performance_schema` harus diatur ke `0` dan Sumber tidak boleh diatur ke `user`. Jika Wawasan Performa tidak mengelola Skema Performa secara otomatis, lihat Mengaktifkan Skema Performa untuk Wawasan Performa di Amazon RDS for MariaDB atau MySQL.

Di AWS CLI, periksa sumber nilai parameter dengan menjalankan perintah describe-db-parameters.

Statistik per detik untuk MariaDB dan MySQL

Statistik SQL berikut ini tersedia untuk klaster DB MariaDB dan MySQL.

Metrik	Unit
db.sql_tokenized.stats.count_star_per_sec	Panggilan per detik
db.sql_tokenized.stats.sum_timer_wait_per_sec	Eksekusi aktif rata-rata per detik (AAE)
db.sql_tokenized.stats.sum_select_full_join_per_sec	Memilih penggabungan penuh per detik
db.sql_tokenized.stats.sum_select_range_check_per_sec	Memilih pemeriksaan rentang per detik
db.sql_tokenized.stats.sum_select_scan_per_sec	Memilih pemindaian per detik

Metrik	Unit
db.sql_tokenized.stats.sum_sort_merge_passes_per_sec	Mengurutkan pass penggabungan per detik
db.sql_tokenized.stats.sum_sort_scan_per_sec	Mengurutkan pemindaian per detik
db.sql_tokenized.stats.sum_sort_range_per_sec	Mengurutkan rentang per detik
db.sql_tokenized.stats.sum_sort_rows_per_sec	Mengurutkan baris per detik
db.sql_tokenized.stats.sum_rows_affected_per_sec	Baris yang terpengaruh per detik
db.sql_tokenized.stats.sum_rows_examined_per_sec	Baris yang diperiksa per detik
db.sql_tokenized.stats.sum_rows_sent_per_sec	Baris yang dikirim per detik
db.sql_tokenized.stats.sum_created_tmp_disk_tables_per_sec	Tabel disk sementara yang dibuat per detik
db.sql_tokenized.stats.sum_created_tmp_tables_per_sec	Tabel disk sementara yang dibuat per detik
db.sql_tokenized.stats.sum_lock_time_per_sec	Waktu pencungian per detik (dalam md)

Statistik per panggilan untuk MariaDB dan MySQL

Metrik berikut menyediakan statistik per panggilan untuk pernyataan SQL.

Metrik	Unit
db.sql_tokenized.stats.sum_timer_wait_per_call	Latensi rata-rata per panggilan (dalam md)
db.sql_tokenized.stats.sum_select_full_join_per_call	Memilih penggabungan penuh per panggilan

Metrik	Unit
db.sql_tokenized.stats.sum_select_range_check_per_call	Memilih pemeriksaan rentang per panggilan
db.sql_tokenized.stats.sum_select_scan_per_call	Memilih pemindaian per panggilan
db.sql_tokenized.stats.sum_sort_merge_passes_per_call	Mengurutkan pass penggabungan per panggilan
db.sql_tokenized.stats.sum_sort_scan_per_call	Mengurutkan pemindaian per panggilan
db.sql_tokenized.stats.sum_sort_range_per_call	Mengurutkan rentang per panggilan
db.sql_tokenized.stats.sum_sort_rows_per_call	Mengurutkan baris per panggilan
db.sql_tokenized.stats.sum_rows_affected_per_call	Baris yang terpengaruh per panggilan
db.sql_tokenized.stats.sum_rows_examined_per_call	Baris yang diperiksa per panggilan
db.sql_tokenized.stats.sum_rows_sent_per_call	Baris yang terkirim per panggilan
db.sql_tokenized.stats.sum_created_tmp_disk_tables_per_call	Tabel disk sementara yang dibuat per panggilan
db.sql_tokenized.stats.sum_created_tmp_tables_per_call	Tabel sementara yang dibuat per panggilan
db.sql_tokenized.stats.sum_lock_time_per_call	Waktu penguncian per panggilan (dalam md)

Statistik SQL untuk Oracle

Amazon RDS for Oracle mengumpulkan statistik SQL baik pada tingkat pernyataan maupun digest. Pada tingkat pernyataan, kolom ID mewakili nilai `V$SQL.SQL_ID`. Pada tingkat digest, kolom ID menunjukkan nilai `V$SQL.FORCE_MATCHING_SIGNATURE`.

Jika ID-nya adalah `0` pada tingkat digest, Basis Data Oracle telah menentukan bahwa pernyataan ini tidak cocok untuk digunakan kembali. Dalam hal ini, pernyataan SQL turunan bisa berada pada tingkat digest yang berbeda. Namun, pernyataan tersebut dikelompokkan bersama di bagian `digest_text` untuk pernyataan SQL pertama yang dikumpulkan.

Topik

Statistik per detik untuk Oracle

Metrik berikut menyediakan statistik per detik untuk Oracle SQL.

Metrik	Unit
db.sql.stats.executions_per_sec	Jumlah eksekusi per detik
db.sql.stats.elapsed_time_per_sec	Eksekusi aktif rata-rata (AAE)
db.sql.stats.rows_processed_per_sec	Baris yang diproses per detik
db.sql.stats.buffer_gets_per_sec	Perolehan buffer per detik
db.sql.stats.physical_read_requests_per_sec	Pembacaan fisik per detik
db.sql.stats.physical_write_requests_per_sec	Penulisan fisik per detik
db.sql.stats.total_sharable_mem_per_sec	Total memori yang dapat dibagikan per detik (dalam byte)
db.sql.stats.cpu_time_per_sec	Waktu CPU per detik (dalam md)

Metrik berikut menyediakan statistik per panggilan untuk kueri digest Oracle SQL.

Metrik	Unit
db.sql_tokenized.stats.executions_per_sec	Jumlah eksekusi per detik

Metrik	Unit
db.sql_tokenized.stats.elapsed_time_per_sec	Eksekusi aktif rata-rata (AAE)
db.sql_tokenized.stats.rows_processed_per_sec	Baris yang diproses per detik
db.sql_tokenized.stats.buffer_gets_per_sec	Perolehan buffer per detik
db.sql_tokenized.stats.physical_read_requests_per_sec	Pembacaan fisik per detik
db.sql_tokenized.stats.physical_write_requests_per_sec	Penulisan fisik per detik
db.sql_tokenized.stats.total_sharable_mem_per_sec	Total memori yang dapat dibagikan per detik (dalam byte)
db.sql_tokenized.stats.cpu_time_per_sec	Waktu CPU per detik (dalam md)

Statistik per panggilan untuk Oracle

Metrik berikut menyediakan statistik per panggilan untuk pernyataan Oracle SQL.

Metrik	Unit
db.sql.stats.elapsed_time_per_exec	Waktu berlalu per eksekusi (dalam md)
db.sql.stats.rows_processed_per_exec	Baris yang diproses per eksekusi
db.sql.stats.buffer_gets_per_exec	Perolehan buffer per eksekusi
db.sql.stats.physical_read_requests_per_exec	Pembacaan fisik per eksekusi
db.sql.stats.physical_write_requests_per_exec	Penulisan fisik per eksekusi
db.sql.stats.total_sharable_mem_per_exec	Total memori yang dapat dibagikan per eksekusi (dalam byte)
db.sql.stats.cpu_time_per_exec	Waktu CPU per eksekusi (dalam md)

Metrik berikut menyediakan statistik per panggilan untuk kueri digest Oracle SQL.

Metrik	Unit
db.sql_tokenized.stats.elapsed_time_per_exec	Waktu berlalu per eksekusi (dalam md)
db.sql_tokenized.stats.rows_processed_per_exec	Baris yang diproses per eksekusi
db.sql_tokenized.stats.buffer_gets_per_exec	Perolehan buffer per eksekusi
db.sql_tokenized.stats.physical_read_requests_per_exec	Pembacaan fisik per eksekusi
db.sql_tokenized.stats.physical_write_requests_per_exec	Penulisan fisik per eksekusi
db.sql_tokenized.stats.total_sharable_mem_per_exec	Total memori yang dapat dibagikan per eksekusi (dalam byte)
db.sql_tokenized.stats.cpu_time_per_exec	Waktu CPU per eksekusi (dalam md)

Statistik SQL untuk SQL Server

Amazon RDS for SQL Server mengumpulkan statistik SQL baik pada tingkat pernyataan maupun digest. Pada tingkat pernyataan, kolom ID mewakili nilai `sql_handle`. Pada tingkat digest, kolom ID menunjukkan nilai `query_hash`.

SQL Server menampilkan nilai NULL untuk `query_hash` beberapa pernyataan. Misalnya, ALTER INDEX, CHECKPOINT, UPDATE STATISTICS, COMMIT TRANSACTION, FETCH NEXT FROM Cursor, dan beberapa pernyataan INSERT, SELECT @<variable>, pernyataan bersyarat, dan prosedur tersimpan yang dapat dieksekusi. Dalam hal ini, nilai `sql_handle` ditampilkan sebagai ID pada tingkat digest untuk pernyataan tersebut.

Topik

- Statistik per detik untuk SQL Server
- Statistik per panggilan untuk SQL Server

Statistik per detik untuk SQL Server

Metrik berikut menyediakan statistik per detik untuk kueri SQL SQL Server.

Metrik	Unit
db.sql.stats.execution_count_per_sec	Jumlah eksekusi per detik
db.sql.stats.total_elapsed_time_per_sec	Total waktu berlalu per detik
db.sql.stats.total_rows_per_sec	Total baris yang diproses per detik
db.sql.stats.total_logical_reads_per_sec	Total pembacaan logis per detik
db.sql.stats.total_logical_writes_per_sec	Total penulisan logis per detik
db.sql.stats.total_physical_reads_per_sec	Total pembacaan fisik per detik
db.sql.stats.total_worker_time_per_sec	Total waktu CPU (dalam md)

Metrik berikut menyediakan statistik per detik untuk kueri digest SQL SQL Server.

Metrik	Unit
db.sql_tokenized.stats.execution_count_per_sec	Jumlah eksekusi per detik
db.sql_tokenized.stats.total_elapsed_time_per_sec	Total waktu berlalu per detik
db.sql_tokenized.stats.total_rows_per_sec	Total baris yang diproses per detik
db.sql_tokenized.stats.total_logical_reads_per_sec	Total pembacaan logis per detik
db.sql_tokenized.stats.total_logical_writes_per_sec	Total penulisan logis per detik
db.sql_tokenized.stats.total_physical_reads_per_sec	Total pembacaan fisik per detik

Metrik	Unit
db.sql_tokenized.stats.total_worker_time_per_sec	Total waktu CPU (dalam md)

Statistik per panggilan untuk SQL Server

Metrik berikut menyediakan statistik per panggilan untuk pernyataan SQL SQL Server.

Metrik	Unit
db.sql.stats.total_elapsed_time_per_call	Total waktu yang berlalu per eksekusi
db.sql.stats.total_rows_per_call	Total baris yang diproses per eksekusi
db.sql.stats.total_logical_reads_per_call	Total pembacaan logis per eksekusi
db.sql.stats.total_logical_writes_per_call	Total penulisan logis per eksekusi
db.sql.stats.total_physical_reads_per_call	Total pembacaan fisik per eksekusi
db.sql.stats.total_worker_time_per_call	Total waktu CPU per eksekusi (dalam md)

Metrik berikut menyediakan statistik per panggilan untuk kueri digest SQL SQL Server.

Metrik	Unit
db.sql_tokenized.stats.total_elapsed_time_per_call	Total waktu yang berlalu per eksekusi
db.sql_tokenized.stats.total_rows_per_call	Total baris yang diproses per eksekusi
db.sql_tokenized.stats.total_logical_reads_per_call	Total pembacaan logis per eksekusi
db.sql_tokenized.stats.total_logical_writes_per_call	Total penulisan logis per eksekusi

Metrik	Unit
db.sql_tokenized.stats.total_physical_reads_per_call	Total pembacaan fisik per eksekusi
db.sql_tokenized.stats.total_worker_time_per_call	Total waktu CPU per eksekusi (dalam md)

Statistik SQL untuk RDS PostgreSQL

Untuk setiap panggilan SQL dan untuk setiap detik eksekusi kueri, Wawasan Performa mengumpulkan statistik SQL. RDS for PostgreSQL mengumpulkan statistik SQL hanya di tingkat digest. Tidak ada statistik yang ditampilkan di tingkat pernyataan.

Berikut ini, Anda dapat menemukan informasi tentang statistik tingkat digest untuk RDS for PostgreSQL.

Topik

- Statistik digest untuk RDS PostgreSQL
- Statistik digest per detik untuk RDS PostgreSQL
- Statistik digest per panggilan untuk RDS PostgreSQL

Statistik digest untuk RDS PostgreSQL

Untuk melihat statistik digest SQL, RDS PostgreSQL harus memuat pustaka `pg_stat_statements`. Untuk instans DB PostgreSQL yang kompatibel dengan PostgreSQL 11 atau versi yang lebih baru, basis data memuat pustaka ini secara default. Untuk instans DB PostgreSQL yang kompatibel dengan PostgreSQL 10 atau versi yang lebih rendah, aktifkan pustaka ini secara manual. Untuk mengaktifkannya secara manual, tambahkan `pg_stat_statements` ke `shared_preload_libraries` di grup parameter DB yang terkait dengan instans DB. Lalu reboot instans DB Anda. Untuk informasi selengkapnya, lihat Bekerja dengan grup parameter.

> **Note**
>
> Wawasan Performa hanya dapat mengumpulkan statistik untuk kueri dalam `pg_stat_activity` yang tidak terpotong. Secara default, basis data PostgreSQL memotong kueri yang lebih panjang dari 1.024 byte. Untuk menambah ukuran kueri, ubah

parameter `track_activity_query_size` dalam grup parameter DB yang terkait dengan instans DB Anda. Jika Anda mengubah parameter ini, instans DB harus di-reboot.

Statistik digest per detik untuk RDS PostgreSQL

Statistik digest SQL berikut tersedia untuk instans DB PostgreSQL.

Metrik	Unit
db.sql_tokenized.stats.calls_per_sec	Panggilan per detik
db.sql_tokenized.stats.rows_per_sec	Baris per detik
db.sql_tokenized.stats.total_time_per_sec	Eksekusi aktif rata-rata per detik (AAE)
db.sql_tokenized.stats.shared_blks_hit_per_sec	Hit blokir per detik
db.sql_tokenized.stats.shared_blks_read_per_sec	Pembacaan blokir per detik
db.sql_tokenized.stats.shared_blks_dirtied_per_sec	Blokir kotor per detik
db.sql_tokenized.stats.shared_blks_written_per_sec	Penulisan blokir per detik
db.sql_tokenized.stats.local_blks_hit_per_sec	Hit blokir lokal per detik
db.sql_tokenized.stats.local_blks_read_per_sec	Pembacaan blokir lokal per detik
db.sql_tokenized.stats.local_blks_dirtied_per_sec	Blokir kotor lokal per detik
db.sql_tokenized.stats.local_blks_written_per_sec	Penulisan blokir lokal per detik
db.sql_tokenized.stats.temp_blks_written_per_sec	Penulisan sementara per detik

Metrik	Unit
db.sql_tokenized.stats.temp_blks_read_per_sec	Pembacaan sementara per detik
db.sql_tokenized.stats.blk_read_time_per_sec	Pembacaan serentak rata-rata per detik
db.sql_tokenized.stats.blk_write_time_per_sec	Penulisan serentak rata-rata per detik

Statistik digest per panggilan untuk RDS PostgreSQL

Metrik berikut menyediakan statistik per panggilan untuk pernyataan SQL.

Metrik	Unit
db.sql_tokenized.stats.rows_per_call	Baris per panggilan
db.sql_tokenized.stats.avg_latency_per_call	Latensi rata-rata per panggilan (dalam md)
db.sql_tokenized.stats.shared_blks_hit_per_call	Hit blokir per panggilan
db.sql_tokenized.stats.shared_blks_read_per_call	Pembacaan blokir per panggilan
db.sql_tokenized.stats.shared_blks_written_per_call	Penulisan blokir per panggilan
db.sql_tokenized.stats.shared_blks_dirtied_per_call	Blokir kotor per panggilan
db.sql_tokenized.stats.local_blks_hit_per_call	Hit blokir lokal per panggilan
db.sql_tokenized.stats.local_blks_read_per_call	Pembcaan blokir lokal per panggilan
db.sql_tokenized.stats.local_blks_dirtied_per_call	Blokir kotor lokal per panggilan
db.sql_tokenized.stats.local_blks_written_per_call	Penulisan blokir lokal per panggilan

Metrik	Unit
db.sql_tokenized.stats.temp_blks_written_per_call	Penulisan blokir sementara per panggilan
db.sql_tokenized.stats.temp_blks_read_per_call	Pembacaan blokir sementara per panggilan
db.sql_tokenized.stats.blk_read_time_per_call	Waktu baca per panggilan (dalam md)
db.sql_tokenized.stats.blk_write_time_per_call	Waktu tulis per panggilan (dalam md)

Untuk informasi selengkapnya tentang metrik ini, lihat pg_stat_statements dalam dokumentasi PostgreSQL.

Metrik OS dalam Pemantauan yang Disempurnakan

Amazon RDS menyediakan metrik secara real-time untuk sistem operasi (OS) tempat instans DB Anda berjalan. RDS memberikan metrik dari Enhanced Monitoring ke akun Amazon Logs Anda. CloudWatch Tabel berikut mencantumkan metrik OS yang tersedia menggunakan Amazon CloudWatch Logs.

Topik

- Metrik OS untuk Db2, MariaDB, MySQL, Oracle, dan PostgreSQL
- Metrik OS for Microsoft SQL Server

Metrik OS untuk Db2, MariaDB, MySQL, Oracle, dan PostgreSQL

Grup	Metrik	Nama konsol	Deskripsi
`General`	`engine`	Tidak berlaku	Mesin basis data untuk instans DB.
	`instanceID`	Tidak berlaku	Pengidentifikasi instans DB.

Grup	Metrik	Nama konsol	Deskripsi
	instanceResourceID	Tidak berlaku	Pengidentifikasi tetap untuk instans DB yang unik untuk Wilayah AWS, juga digunakan sebagai pengidentifikasi log stream.
	numVCPUs	Tidak berlaku	Jumlah CPU virtual untuk instans DB.
	timestamp	Tidak berlaku	Waktu pengambilan metrik.
	uptime	Tidak berlaku	Jumlah waktu saat instans DB telah aktif.
	version	Tidak berlaku	Versi format JSON aliran metrik OS.
cpuUtilization	guest	CPU Tamu	Persentase CPU yang digunakan oleh program tamu.
	idle	CPU Idle	Persentase CPU saat idle.
	irq	CPU IRQ	Persentase CPU yang digunakan oleh gangguan perangkat lunak.
	nice	CPU Nice	Persentase CPU yang digunakan oleh program yang berjalan pada prioritas terendah.
	steal	CPU Steal	Persentase CPU yang digunakan oleh mesin virtual lainnya.
	system	CPU Sistem	Persentase CPU yang digunakan oleh kernel.
	total	CPU Total	Total persentase CPU yang digunakan. Nilai ini mencakup nilai nice.
	user	Pengguna CPU	Persentase CPU yang digunakan oleh program pengguna.

Grup	Metrik	Nama konsol	Deskripsi
	`wait`	CPU Tunggu	Persentase CPU yang tidak digunakan saat menunggu akses I/O.
`diskIO`	`avgQueueLen`	Ukuran Antrean Rata-rata	Jumlah permintaan yang menunggu dalam antrean perangkat I/O.
	`avgReqSz`	Ukuran Permintaan Rata-rata	Ukuran permintaan rata-rata, dalam kilobyte.
	`await`	I/O Disk Tunggu	Jumlah milidetik yang diperlukan untuk merespons permintaan, termasuk waktu antrean dan waktu layanan.
	`device`	Tidak berlaku	Pengidentifikasi perangkat disk yang digunakan.
	`readIOsPS`	IO/d Baca	Jumlah operasi baca per detik.
	`readKb`	Total Baca	Jumlah total kilobyte yang dibaca.
	`readKbPS`	Kb/d Baca	Jumlah kilobyte yang dibaca per detik.
	`readLatency`	Latensi Baca	Waktu yang berlalu antara pengiriman permintaan I/O baca dan penyelesaiannya, dalam milidetik. Metrik ini hanya tersedia untuk Amazon Aurora.
	`readThroughput`	Throughput Baca	Jumlah throughput jaringan yang digunakan oleh permintaan ke klaster DB, dalam byte per detik. Metrik ini hanya tersedia untuk Amazon Aurora.
	`rrqmPS`	Rrqms	Jumlah permintaan baca gabungan yang diantrekan per detik.

Grup	Metrik	Nama konsol	Deskripsi
	`tps`	TPS	Jumlah transaksi I/O per detik.
	`util`	Penggunaan I/O Disk	Persentase waktu CPU saat permintaan dikeluarkan.
	`writeIOsPS`	IO/d Tulis	Jumlah operasi tulis per detik.
	`writeKb`	Total Tulis	Jumlah total kilobyte yang ditulis.
	`writeKbPS`	Kb/d Tulis	Jumlah kilobyte yang ditulis per detik.
	`writeLatency`	Latensi Tulis	Waktu yang berlalu antara pengiriman permintaan I/O tulis dan penyelesaiannya, dalam milidetik. Metrik ini hanya tersedia untuk Amazon Aurora.
	`writeThroughput`	Throughput Tulis	Jumlah throughput jaringan yang digunakan oleh respons dari klaster DB, dalam byte per detik. Metrik ini hanya tersedia untuk Amazon Aurora.
	`wrqmPS`	Wrqms	Jumlah permintaan tulis gabungan yang diantrekan per detik.
`physicalDeviceIO`	`avgQueueLen`	Ukuran Antrean Rata-Rata Perangkat Fisik	Jumlah permintaan yang menunggu dalam antrean perangkat I/O.

Grup	Metrik	Nama konsol	Deskripsi
	`avgReqSz`	Ukuran Permintaan Rata-Rata Perangkat Fisik	Ukuran permintaan rata-rata, dalam kilobyte.
	`await`	I/O Disk Perangkat Fisik Tunggu	Jumlah milidetik yang diperlukan untuk merespons permintaan, termasuk waktu antrean dan waktu layanan.
	`device`	Tidak berlaku	Pengidentifikasi perangkat disk yang digunakan.
	`readIOsPS`	IO/d Baca Perangkat Fisik	Jumlah operasi baca per detik.
	`readKb`	Baca Total Perangkat Fisik	Jumlah total kilobyte yang dibaca.
	`readKbPS`	Kb/d Baca Perangkat Fisik	Jumlah kilobyte yang dibaca per detik.
	`rrqmPS`	Rrqms Perangkat Fisik	Jumlah permintaan baca gabungan yang diantrekan per detik.
	`tps`	TPS Perangkat Fisik	Jumlah transaksi I/O per detik.

Grup	Metrik	Nama konsol	Deskripsi
	`util`	Penggunaan I/O Disk Perangkat Fisik	Persentase waktu CPU saat permintaan dikeluarkan.
	`writeIOsPS`	IO/d Tulis Perangkat Fisik	Jumlah operasi tulis per detik.
	`writeKb`	Total Tulis Perangkat Fisik	Jumlah total kilobyte yang ditulis.
	`writeKbPS`	Kb/d Tulis Perangkat Fisik	Jumlah kilobyte yang ditulis per detik.
	`wrqmPS`	Wrqms Perangkat Fisik	Jumlah permintaan tulis gabungan yang diantrekan per detik.
`fileSys`	`maxFiles`	Inode Maks	Jumlah maksimum file yang dapat dibuat untuk sistem file.
	`mountPoint`	Tidak berlaku	Jalur ke sistem file.
	`name`	Tidak berlaku	Nama sistem file.
	`total`	Total Sistem File	Jumlah total ruang disk yang tersedia untuk sistem file, dalam kilobyte.
	`used`	Sistem file yang digunakan	Jumlah ruang disk yang digunakan oleh file dalam sistem file, dalam kilobyte.

Grup	Metrik	Nama konsol	Deskripsi
	`usedFilePercent`	Inode yang Digunakan	Persentase file tersedia yang digunakan.
	`usedFiles`	% yang Digunakan	Jumlah file dalam sistem file.
	`usedPercent`	Sistem file yang digunakan	Persentase ruang disk sistem file yang digunakan.
`loadAverageMinute`	`fifteen`	Rata-rata Muatan 15 menit	Jumlah proses yang meminta waktu CPU selama 15 menit terakhir.
	`five`	Rata-rata Muatan 5 menit	Jumlah proses yang meminta waktu CPU selama 5 menit terakhir.
	`one`	Rata-rata Muatan 1 menit	Jumlah proses yang meminta waktu CPU selama satu menit terakhir.
`memory`	`active`	Memori Aktif	Jumlah memori yang ditetapkan, dalam kilobyte.
	`buffers`	Memori yang Di-buffer	Jumlah memori yang digunakan untuk buffering permintaan I/O sebelum menulis ke perangkat penyimpanan, dalam kilobyte.
	`cached`	Memori yang Di-cache	Jumlah memori yang digunakan untuk meng-cache file I/O berbasis sistem file.
	`dirty`	Memori Kotor	Jumlah halaman memori dalam RAM yang telah diubah, tetapi tidak ditulis ke blok data terkaitnya dalam penyimpanan, dalam kilobyte.

Grup	Metrik	Nama konsol	Deskripsi
	`free`	Memori Kosong	Jumlah memori yang tidak ditetapkan, dalam kilobyte.
	`hugePagesFree`	Halaman Besar Kosong	Jumlah halaman besar yang kosong. Halaman besar adalah fitur dari kernel Linux.
	`hugePagesRsvd`	Rsvd Halaman Besar	Jumlah halaman besar yang khusus.
	`hugePagesSize`	Ukuran Halaman Besar	Ukuran untuk setiap unit halaman besar, dalam kilobyte.
	`hugePagesSurp`	Surplus Halaman Besar	Jumlah halaman besar surplus yang tersedia terhadap total.
	`hugePagesTotal`	Total Halaman Besar	Jumlah total halaman besar.
	`inactive`	Memori Tidak Aktif	Jumlah halaman memori yang jarang digunakan, dalam kilobyte.
	`mapped`	Memori yang Dipetakan	Jumlah total konten sistem file yang memorinya dipetakan di dalam ruang alamat proses, dalam kilobyte.
	`pageTables`	Tabel Halaman	Jumlah memori yang digunakan berdasarkan tabel halaman, dalam kilobyte.
	`slab`	Memori Slab	Jumlah struktur data kernel yang dapat digunakan kembali, dalam kilobyte.

Grup	Metrik	Nama konsol	Deskripsi
	`total`	Memori Total	Jumlah total memori, dalam kilobyte.
	`writeback`	Memori Writeback	Jumlah halaman kotor dalam RAM yang masih ditulis ke penyimpanan cadangan, dalam kilobyte.
`network`	`interface`	Tidak berlaku	Pengidentifikasi untuk antarmuka jaringan yang digunakan untuk instans DB.
	`rx`	RX	Jumlah byte yang diterima per detik.
	`tx`	TX	Jumlah byte yang diunggah per detik.
`processList`	`cpuUsedPc`	CPU %	Persentase CPU yang digunakan oleh proses.
	`id`	Tidak berlaku	Pengidentifikasi proses.
	`memoryUsedPc`	MEM%	Persentase memori yang digunakan oleh proses.
	`name`	Tidak berlaku	Nama proses.
	`parentID`	Tidak berlaku	ID proses untuk proses induk dari proses.
	`rss`	RES	Jumlah RAM yang dialokasikan untuk proses, dalam kilobyte.
	`tgid`	Tidak berlaku	ID grup utas, yaitu angka yang mewakili ID proses di tempat utas berada. ID ini digunakan untuk mengelompokkan utas dari proses yang sama.
	`vss`	VIRT	Jumlah memori virtual yang dialokasikan untuk proses, dalam kilobyte.

Grup	Metrik	Nama konsol	Deskripsi
swap	swap	Swap	Jumlah memori swap yang tersedia, dalam kilobyte.
	swap in	Swap dalam	Jumlah memori, dalam kilobyte, yang ditukar ke dalam dari disk.
	swap out	Swap luar	Jumlah memori, dalam kilobyte, yang ditukar ke luar dari disk.
	free	Swap Kosong	Jumlah memori swap yang kosong, dalam kilobyte.
	committed	Swap Komit	Jumlah memori swap, dalam kilobyte, yang digunakan sebagai memori cache.
tasks	blocked	Tugas Diblokir	Jumlah tugas yang diblokir.
	running	Tugas Berjalan	Jumlah tugas yang berjalan.
	sleeping	Tugas Tidur	Jumlah tugas yang tidur.
	stopped	Tugas Dihentikan	Jumlah tugas yang dihentikan.
	total	Total Tugas	Jumlah total tugas.
	zombie	Tugas Zombie	Jumlah tugas turunan yang tidak aktif dengan tugas induk aktif.

Metrik OS for Microsoft SQL Server

Grup	Metrik	Nama konsol	Deskripsi
General	engine	Tidak berlaku	Mesin basis data untuk instans DB.
	instanceID	Tidak berlaku	Pengidentifikasi instans DB.
	instanceResourceID	Tidak berlaku	Pengidentifikasi tetap untuk instans DB yang unik untuk Wilayah AWS, juga digunakan sebagai pengidentifikasi log stream.
	numVCPUs	Tidak berlaku	Jumlah CPU virtual untuk instans DB.
	timestamp	Tidak berlaku	Waktu pengambilan metrik.
	uptime	Tidak berlaku	Jumlah waktu saat instans DB telah aktif.
	version	Tidak berlaku	Versi format JSON aliran metrik OS.
cpuUtilization	idle	CPU Idle	Persentase CPU saat idle.
	kern	CPU Kernel	Persentase CPU yang digunakan oleh kernel.
	user	Pengguna CPU	Persentase CPU yang digunakan oleh program pengguna.
disks	name	Tidak berlaku	ID untuk disk.
	totalKb	Total Ruang Disk	Total ruang disk, dalam kilobyte.
	usedKb	Ruang Disk Digunakan	Jumlah ruang disk yang digunakan pada disk, dalam kilobyte.
	usedPc	% Ruang Disk Digunakan	Persentase ruang yang digunakan pada disk.

Grup	Metrik	Nama konsol	Deskripsi
	`availKb`	Ruang Disk Tersedia	Ruang yang tersedia pada disk, dalam kilobyte.
	`availPc`	% Ruang Disk Tersedia	Persentase ruang yang tersedia pada disk.
	`rdCountPS`	Baca/d	Jumlah operasi baca per detik.
	`rdBytesPS`	Kb/d Baca	Jumlah byte baca per detik.
	`wrCountPS`	IO/d Tulis	Jumlah operasi tulis per detik.
	`wrBytesPS`	Kb/d Tulis	Jumlah byte tulis per detik.
`memory`	`commitTotKb`	Total Komitmen	Jumlah ruang alamat virtual yang didukung file halaman yang digunakan, yaitu, biaya komitmen saat ini. Nilai ini terdiri dari memori utama (RAM) dan disk (file halaman).
	`commitLimitKb`	Komitmen Maksimum	Kemungkinan nilai maksimum untuk metrik `commitTotKb` . Nilai ini adalah jumlah ukuran file halaman saat ini ditambah memori fisik yang tersedia untuk konten yang dapat dibuat halamannya, tidak termasuk RAM yang ditetapkan ke area yang tidak dapat dibuat halamannya.
	`commitPeakKb`	Komitmen Puncak	Nilai terbesar dari metrik `commitTotKb` sejak sistem operasi terakhir kali dimulai.
	`kernTotKb`	Total Memori Kernel	Jumlah memori di dalam kumpulan kernel dengan halaman dan tanpa halaman, dalam kilobyte.

Grup	Metrik	Nama konsol	Deskripsi
	kernPagedKb	Memori Kernel dengan Halaman	Jumlah memori di dalam kumpulan kernel berhalaman, dalam kilobyte.
	kernNonpagedKb	Memori Kernel Tanpa Halaman	Jumlah memori di dalam kumpulan kernel tanpa halaman, dalam kilobyte.
	pageSize	Ukuran Halaman	Ukuran halaman, dalam byte.
	physTotKb	Memori Total	Jumlah memori fisik, dalam kilobyte.
	physAvailKb	Memori Tersedia	Jumlah memori fisik tersedia, dalam kilobyte.
	sqlServerTotKb	Total Memori SQL Server	Jumlah memori yang dikhususkan untuk SQL Server, dalam kilobyte.
	sysCacheKb	Cache Sistem	Jumlah memori cache sistem, dalam kilobyte.
network	interface	Tidak berlaku	Pengidentifikasi untuk antarmuka jaringan yang digunakan untuk instans DB.
	rdBytesPS	Kb/d Baca Jaringan	Jumlah byte yang diterima per detik.
	wrBytesPS	Kb/dt Tulis Jaringan	Jumlah byte yang dikirim per detik.
processList	cpuUsedPc	% yang digunakan	Persentase CPU yang digunakan oleh proses.
	memUsedPc	MEM%	Persentase total memori yang digunakan oleh proses.

Grup	Metrik	Nama konsol	Deskripsi
	`name`	Tidak berlaku	Nama proses.
	`pid`	Tidak berlaku	Pengidentifikasi proses. Nilai ini tidak ada untuk proses milik oleh Amazon RDS.
	`ppid`	Tidak berlaku	ID proses untuk induk proses ini. Nilai ini hanya muncul untuk proses turunan.
	`tid`	Tidak berlaku	ID utas. Nilai ini hanya ada untuk utas. Proses kepemilikan dapat diidentifikasi dengan menggunakan nilai `pid`.
	`workingSetKb`	Tidak berlaku	Jumlah memori dalam set kerja pribadi ditambah jumlah memori yang digunakan oleh proses dan dapat dibagikan dengan proses lain, dalam kilobyte.
	`workingSetPrivKb`	Tidak berlaku	Jumlah memori yang digunakan oleh proses, tetapi tidak dapat dibagikan dengan proses lain, dalam kilobyte.
	`workingSetShareableKb`	Tidak berlaku	Jumlah memori yang digunakan oleh proses, dan dapat dibagikan dengan proses lain, dalam kilobyte.
	`virtKb`	Tidak berlaku	Jumlah ruang alamat virtual yang digunakan proses, dalam kilobyte. Penggunaan ruang alamat virtual tidak selalu menyiratkan penggunaan disk atau halaman memori utama yang sesuai.
`system`	`handles`	Handle	Jumlah handle yang digunakan sistem.
	`processes`	Proses	Jumlah proses yang berjalan di sistem.
	`threads`	Utas	Jumlah utas yang berjalan di sistem.

Memantau peristiwa, log, dan aliran di instans DB Amazon RDS

Saat Anda memantau basis data Amazon Aurora RDS dan solusi Anda yang AWS lain, tujuan Anda adalah mempertahankan hal-hal berikut:

- Keandalan
- Ketersediaan
- Performa
- Keamanan

Memantau metrik dalam instans Amazon RDS menjelaskan cara memantau instans menggunakan metrik. Solusi lengkap juga harus memantau peristiwa database, file log, dan aliran aktivitas. AWS menyediakan Anda dengan alat pemantauan berikut:

- Amazon EventBridge adalah layanan bus acara tanpa server yang memudahkan untuk menghubungkan aplikasi Anda dengan data dari berbagai sumber. EventBridge memberikan aliran data real-time dari aplikasi Anda sendiri, aplikasi S oftware-as-a -Service (SaaS), dan layanan. AWS EventBridge merutekan data tersebut ke target seperti AWS Lambda. Dengan demikian, Anda dapat memantau peristiwa yang terjadi di layanan dan membangun arsitektur berbasis peristiwa. Untuk informasi selengkapnya, lihat Panduan EventBridge Pengguna Amazon.
- Amazon CloudWatch Logs menyediakan cara untuk memantau, menyimpan, dan mengakses file log Anda dari instans Amazon Aurora RDS AWS CloudTrail,, dan sumber lainnya. Amazon CloudWatch Logs dapat memantau informasi dalam file log dan memberi tahu Anda ketika ambang batas tertentu terpenuhi. Anda juga dapat mengarsipkan data log dalam penyimpanan yang sangat tahan lama. Untuk informasi selengkapnya, lihat Panduan Pengguna Amazon CloudWatch Logs.
- AWS CloudTrailmenangkap panggilan API dan peristiwa terkait yang dibuat oleh atau atas nama Anda Akun AWS. CloudTrail 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 informasi selengkapnya, silakan lihat Panduan Pengguna AWS CloudTrail.
- Aliran Aktivitas Basis Data adalah fitur Amazon RDS yang menyediakan aliran aktivitas mendekati waktu nyata di instans DB Anda. Amazon RDS mendorong aktivitas ke aliran data Amazon Kinesis.

Aliran Kinesis dibuat secara otomatis. Dari Kinesis, Anda dapat mengonfigurasi AWS layanan seperti Amazon Data Firehose dan AWS Lambda menggunakan aliran dan menyimpan data.

Topik

Melihat log, peristiwa, dan aliran di konsol Amazon RDS

Amazon RDS terintegrasi dengan Layanan AWS untuk menampilkan informasi tentang log, peristiwa, dan aliran aktivitas basis data di konsol RDS.

Tab Log dan peristiwa untuk instans basis data RDS Anda menampilkan informasi berikut:

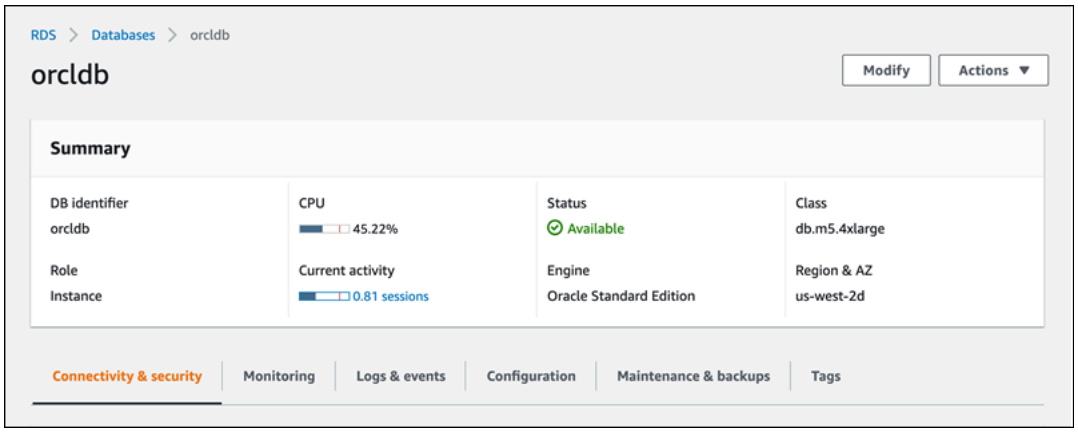
- Alarm Amazon CloudWatch — Menampilkan semua alarm metrik yang telah Anda konfigurasikan untuk instans basis data di Anda. Jika Anda belum mengonfigurasikan alarm, Anda dapat membuatnya di konsol RDS. Lihat informasi yang lebih lengkap di Memantau metrik Amazon RDS dengan Amazon CloudWatch.
- Peristiwa terbaru – Menampilkan ringkasan peristiwa (perubahan lingkungan) untuk instans atau basis data RDS Anda. Lihat informasi yang lebih lengkap di Melihat peristiwa Amazon RDS.
- Log – Menampilkan file log basis data yang dihasilkan oleh instans basis data . Lihat informasi yang lebih lengkap di Memantau file log Amazon RDS.

Tab Konfigurasi akan menayangkan informasi tentang aliran aktivitas basis data.

Untuk melihat log, peristiwa, dan aliran bagi instans basis data Anda di konsol RDS

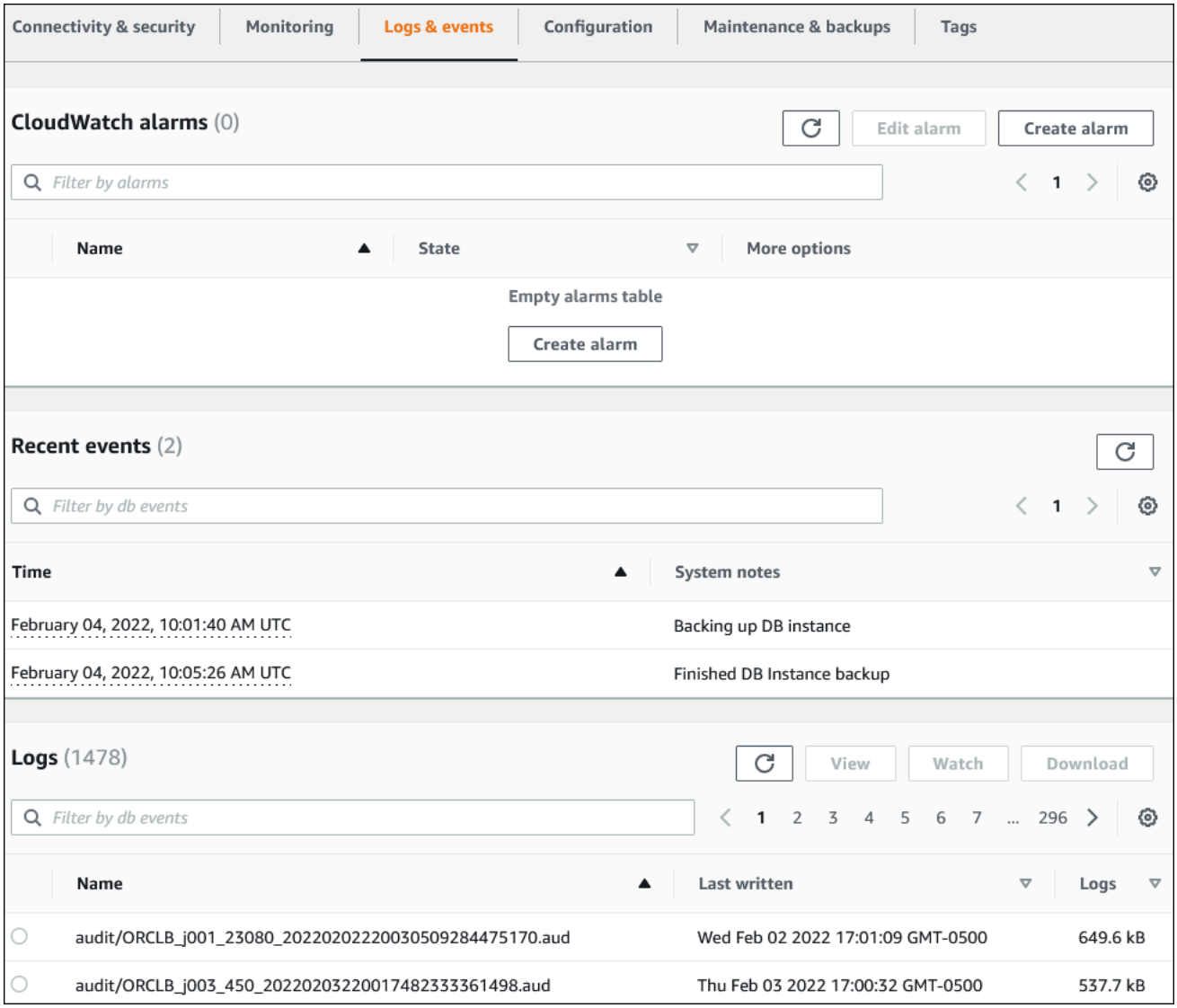
1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data.
3. Pilih nama instans basis data yang ingin Anda pantau.

Halaman basis data akan muncul. Contoh berikut menunjukkan basis data Oracle bernama `orclb`.



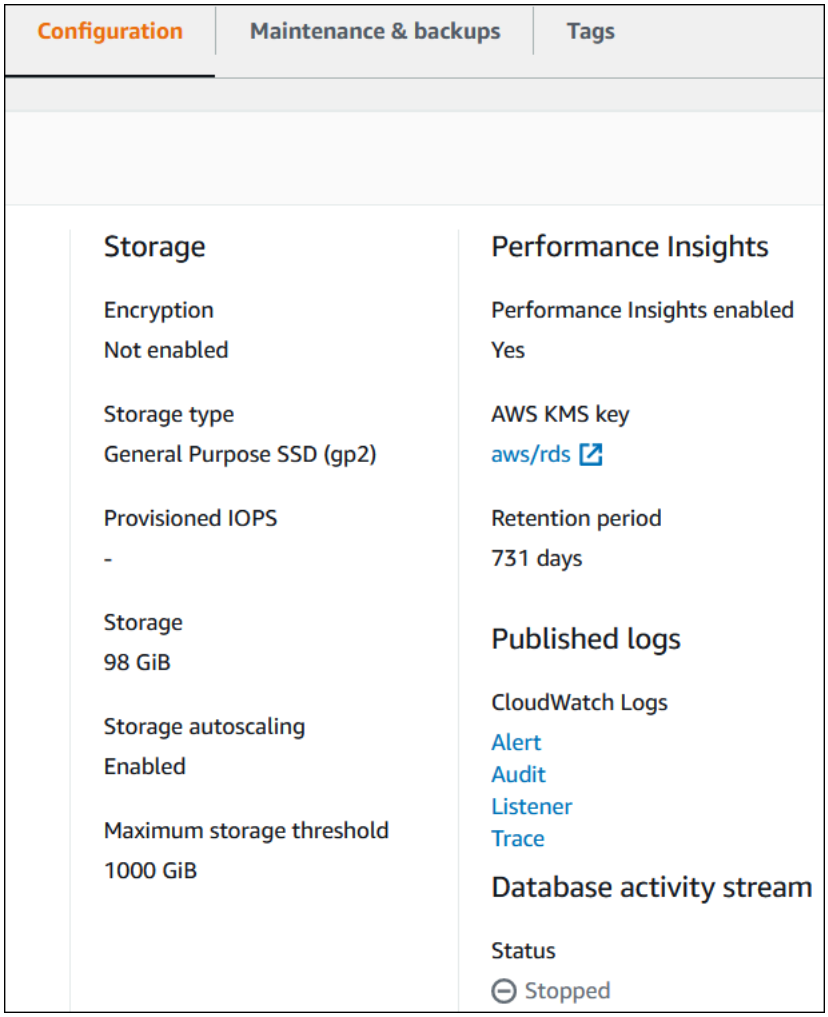
4. Pilih Log dan peristiwa.

   Bagian Log dan peristiwa muncul.



5. Pilih Konfigurasi.

Contoh berikut menunjukkan status aliran aktivitas basis data untuk instans basis data Anda.


Configuration
Maintenance & backups
Tags
Storage
Encryption
Not enabled
Storage type
General Purpose SSD (gp2)
Provisioned IOPS
-
Storage
98 GiB
Storage autoscaling
Enabled
Maximum storage threshold
1000 GiB
Performance Insights
Performance Insights enabled
Yes
AWS KMS key
aws/rds
Retention period
731 days
Published logs
CloudWatch Logs
Alert
Audit
Listener
Trace
Database activity stream
Status
Stopped

Memantau peristiwa Amazon RDS

Peristiwa menunjukkan perubahan dalam lingkungan. Hal ini dapat berupa lingkungan AWS, layanan atau aplikasi partner SaaS, atau aplikasi atau layanan kustom. Untuk mengetahui deskripsi peristiwa RDS, lihat Kategori peristiwa dan pesan peristiwa Amazon RDS.

Topik

Ikhtisar peristiwa untuk Amazon RDS

Peristiwa RDS menunjukkan adanya perubahan di lingkungan Amazon RDS. Misalnya, Amazon RDS menghasilkan peristiwa saat , status instans DB berubah dari tertunda menjadi berjalan. Amazon RDS mengirimkan peristiwa ke CloudWatch Events dan EventBridge hampir secara waktu nyata.

> **Note**
>
> Amazon RDS memancarkan peristiwa semaksimal mungkin. Kami menyarankan Anda menghindari penulisan program yang bergantung pada urutan atau keberadaan peristiwa pemberitahuan, karena program tersebut mungkin tidak berurutan atau hilang.

Amazon RDS mencatat peristiwa yang berhubungan dengan sumber daya berikut:

- Instans DB

  Untuk daftar peristiwa instans DB, lihat Peristiwa instans DB.
- Grup parameter DB

  Untuk daftar peristiwa grup parameter DB, lihat Peristiwa grup parameter DB.
- Grup keamanan DB

  Untuk daftar peristiwa grup keamanan DB, lihat Peristiwa grup keamanan DB.

- Snapshot DB

  Untuk daftar peristiwa snapshot DB, lihat Peristiwa snapshot DB.
- Peristiwa Proksi RDS

  Untuk daftar peristiwa Proksi RDS, lihat Peristiwa Proksi RDS.
- Peristiwa deployment blue/green

  Untuk daftar peristiwa deployment blue/green, lihat Peristiwa deployment blue/green.

Informasi ini mencakup:

- Tanggal dan waktu peristiwa
- Nama sumber dan jenis sumber peristiwa
- Pesan yang terkait dengan peristiwa
- Pemberitahuan peristiwa mencakup tag sejak pesan dikirim dan mungkin tidak mencerminkan tag pada saat peristiwa terjadi

Melihat peristiwa Amazon RDS

Anda dapat mengambil informasi peristiwa berikut untuk sumber daya Amazon RDS:

- Nama sumber daya
- Jenis sumber daya
- Waktu peristiwa
- Ringkasan pesan peristiwa

Akses peristiwa melalui AWS Management Console, yang menampilkan peristiwa selama 24 jam terakhir. Anda juga dapat mengambil peristiwa menggunakan perintah AWS CLI describe-events, atau operasi DescribeEvents RDS API. Jika menggunakan AWS CLI atau RDS API untuk melihat peristiwa, Anda dapat mengambil peristiwa hingga 14 hari terakhir.

> **Note**
>
> Jika perlu menyimpan peristiwa dalam jangka waktu yang lebih lama, Anda dapat mengirimkan peristiwa Amazon RDS ke CloudWatch Events. Untuk mengetahui informasi selengkapnya, lihat Membuat aturan yang memicu peristiwa Amazon RDS

Untuk deskripsi peristiwa Amazon RDS, lihat Kategori peristiwa dan pesan peristiwa Amazon RDS.

Untuk mengakses informasi mendetail tentang peristiwa yang menggunakan AWS CloudTrail, termasuk parameter permintaan, lihat Peristiwa CloudTrail.

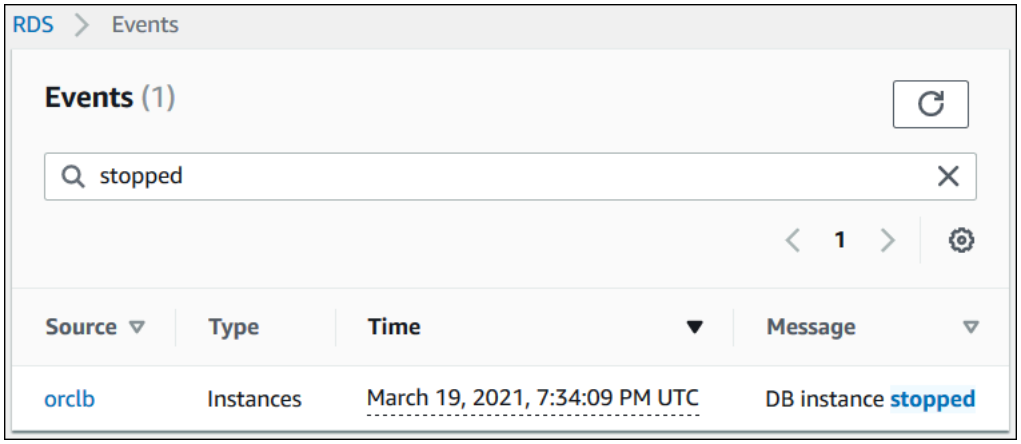
Konsol

Untuk melihat semua peristiwa Amazon RDS selama 24 jam terakhir

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Pada panel navigasi, pilih Peristiwa.

   Peristiwa yang tersedia akan muncul dalam daftar.
3. (Opsional) Masukkan istilah pencarian untuk memfilter hasil.

   Contoh berikut menunjukkan daftar peristiwa yang difilter oleh karakter **`stopped`**.



AWS CLI

Untuk melihat semua peristiwa yang dihasilkan dalam satu jam terakhir, panggil describe-events tanpa parameter.

```
aws rds describe-events
```

Output sampel berikut menunjukkan bahwa instans DB telah dihentikan.

```
{
    "Events": [
        {
            "EventCategories": [
                "notification"
            ],
            "SourceType": "db-instance",
            "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:testinst",
            "Date": "2022-04-22T21:31:00.681Z",
            "Message": "DB instance stopped",
            "SourceIdentifier": "testinst"
        }
    ]
}
```

Untuk melihat semua peristiwa Amazon RDS selama 10.080 menit terakhir (7 hari), panggil perintah describe-events AWS CLI dan atur parameter `--duration` ke `10080`.

```
aws rds describe-events --duration 10080
```

Contoh berikut menunjukkan peristiwa dalam rentang waktu yang ditentukan untuk instans DB *test-instance*.

```
aws rds describe-events \
    --source-identifier test-instance \
    --source-type db-instance \
    --start-time 2022-03-13T22:00Z \
    --end-time 2022-03-13T23:59Z
```

Output sampel berikut menampilkan status cadangan.

```
{
    "Events": [
        {
            "SourceType": "db-instance",
            "SourceIdentifier": "test-instance",
            "EventCategories": [
                "backup"
            ],
            "Message": "Backing up DB instance",
            "Date": "2022-03-13T23:09:23.983Z",
            "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance"
        },
        {
            "SourceType": "db-instance",
            "SourceIdentifier": "test-instance",
            "EventCategories": [
                "backup"
            ],
            "Message": "Finished DB Instance backup",
            "Date": "2022-03-13T23:15:13.049Z",
            "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance"
        }
    ]
}
```

API

Anda dapat melihat semua peristiwa instans Amazon RDS selama 14 hari terakhir dengan memanggil operasi DescribeEvents RDS API dan mengatur parameter `Duration` ke `20160`.

Menggunakan pemberitahuan peristiwa Amazon RDS

Amazon RDS menggunakan Amazon Simple Notification Service (Amazon SNS) untuk memberikan pemberitahuan saat peristiwa Amazon RDS terjadi. Pemberitahuan ini bisa dalam bentuk pemberitahuan apa pun yang didukung oleh Amazon SNS untuk Wilayah AWS, seperti email, pesan teks, atau panggilan ke titik akhir HTTP.

Topik

Ikhtisar pemberitahuan peristiwa Amazon RDS

Amazon RDS mengelompokkan peristiwa ke dalam beberapa kategori langganan sehingga Anda dapat menerima pemberitahuan saat suatu peristiwa dalam kategori tersebut terjadi.

Topik

Sumber daya RDS memenuhi syarat untuk langganan peristiwa

Anda dapat berlangganan kategori peristiwa untuk sumber daya berikut:

- Instans DB
- Snapshot DB
- Grup parameter DB
- Grup keamanan DB
- Proksi RDS
- Versi mesin kustom

Misalnya, jika berlangganan kategori pencadangan untuk instans DB tertentu, Anda akan diberi tahu setiap kali ada peristiwa terkait pencadangan yang memengaruhi instans DB. Jika berlangganan kategori perubahan konfigurasi untuk instans DB, Anda akan diberi tahu saat instans DB diubah. Anda juga menerima pemberitahuan saat langganan pemberitahuan peristiwa berubah.

Sebaiknya Anda membuat beberapa langganan yang berbeda. Misalnya, Anda dapat membuat satu langganan yang menerima semua pemberitahuan peristiwa untuk semua instans DB dan langganan lain yang hanya mencakup peristiwa penting untuk sebagian instans DB. Untuk langganan kedua, tentukan satu atau beberapa instans DB dalam filter.

Proses dasar untuk berlangganan pemberitahuan peristiwa Amazon RDS

Proses untuk berlangganan pemberitahuan peristiwa Amazon RDS adalah sebagai berikut:

1. Anda membuat langganan pemberitahuan peristiwa Amazon RDS dengan menggunakan konsol Amazon RDS, AWS CLI, atau API.

   Amazon RDS menggunakan ARN topik Amazon SNS untuk mengidentifikasi setiap langganan. Konsol Amazon RDS membuat ARN untuk Anda saat Anda membuat langganan. Buat ARN dengan menggunakan konsol Amazon SNS, AWS CLI, atau Amazon SNS API.
2. Amazon RDS mengirimkan email persetujuan atau pesan SMS ke alamat yang Anda kirim bersama langganan Anda.
3. Anda mengonfirmasi langganan Anda dengan memilih tautan di pemberitahuan yang Anda terima.
4. Konsol Amazon RDS memperbarui bagian Langganan Peristiwa Saya dengan status langganan Anda.
5. Amazon RDS mulai mengirim pemberitahuan ke alamat yang Anda berikan saat membuat langganan.

Untuk mempelajari manajemen identitas dan akses saat menggunakan Amazon SNS, lihat Manajemen identitas dan akses di Amazon SNS di Panduan Developer Amazon Simple Notification Service.

Anda dapat menggunakan AWS Lambda untuk memproses pemberitahuan peristiwa dari instans DB. Untuk informasi lebih lanjut, lihat Menggunakan AWS Lambda dengan Amazon RDS di Panduan Developer AWS Lambda.

Pengiriman pemberitahuan peristiwa RDS

Amazon RDS mengirimkan pemberitahuan ke alamat yang Anda berikan saat membuat langganan. Pemberitahuan dapat mencakup atribut pesan yang berisi metadata terstruktur tentang pesan tersebut. Untuk informasi selengkapnya tentang atribut pesan, lihat Kategori peristiwa dan pesan peristiwa Amazon RDS.

Pemberitahuan peristiwa mungkin memerlukan waktu hingga lima menit untuk dikirimkan.

> **Important**
>
> Amazon RDS tidak menjamin urutan peristiwa yang dikirim dalam aliran peristiwa. Urutan peristiwa dapat berubah.

Ketika Amazon SNS mengirimkan pemberitahuan ke titik akhir HTTP atau HTTPS langganan, pesan POST yang dikirim ke titik akhir memiliki isi pesan yang berisi dokumen JSON. Untuk informasi selengkapnya, lihat Pesan Amazon SNS dan format JSON di Panduan Developer Amazon Simple Notification Service.

Anda dapat mengonfigurasi SNS untuk memberi tahu Anda dengan pesan teks. Untuk informasi selengkapnya, lihat Pesan teks seluler (SMS) dalam Panduan Developer Amazon Simple Notification Service.

Untuk menonaktifkan pemberitahuan tanpa menghapus langganan, pilih Tidak untuk Aktif di konsol Amazon RDS. Atau Anda dapat mengatur parameter `Enabled` ke `false` menggunakan AWS CLI atau Amazon RDS API.

Penagihan untuk pemberitahuan peristiwa Amazon RDS

Penagihan untuk pemberitahuan peristiwa Amazon RDS melalui Amazon SNS. Biaya Amazon SNS berlaku saat menggunakan pemberitahuan peristiwa. Untuk informasi selengkapnya tentang penagihan Amazon SNS, lihat Harga Amazon Simple Notification Service.

Contoh peristiwa Amazon RDS

Contoh berikut menggambarkan berbagai jenis peristiwa Amazon RDS dalam format JSON. Untuk tutorial yang menunjukkan cara menangkap dan melihat peristiwa dalam format JSON, lihat Tutorial: Mencatat log perubahan status instans DB menggunakan Amazon EventBridge.

Topik

- Contoh peristiwa instans DB
- Contoh peristiwa grup parameter DB
- Contoh peristiwa snapshot DB

Contoh peristiwa instans DB

Berikut ini adalah contoh peristiwa instans DB dalam format JSON. Peristiwa ini menunjukkan bahwa RDS melakukan failover multi-AZ untuk instans bernama `my-db-instance`. ID peristiwanya adalah RDS-EVENT-0049.

```
{
  "version": "0",
  "id": "68f6e973-1a0c-d37b-f2f2-94a7f62ffd4e",
  "detail-type": "RDS DB Instance Event",
  "source": "aws.rds",
  "account": "123456789012",
  "time": "2018-09-27T22:36:43Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:rds:us-east-1:123456789012:db:my-db-instance"
  ],
  "detail": {
    "EventCategories": [
      "failover"
    ],
    "SourceType": "DB_INSTANCE",
    "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance",
    "Date": "2018-09-27T22:36:43.292Z",
    "Message": "A Multi-AZ failover has completed.",
    "SourceIdentifier": "my-db-instance",
    "EventID": "RDS-EVENT-0049"
  }
}
```

Contoh peristiwa grup parameter DB

Berikut ini adalah contoh peristiwa grup parameter DB dalam format JSON. Peristiwa ini menunjukkan bahwa parameter `time_zone` telah diperbarui dalam grup parameter `my-db-param-group`. ID peristiwanya adalah RDS-EVENT-0037.

```
{
  "version": "0",
  "id": "844e2571-85d4-695f-b930-0153b71dcb42",
  "detail-type": "RDS DB Parameter Group Event",
  "source": "aws.rds",
  "account": "123456789012",
  "time": "2018-10-06T12:26:13Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:rds:us-east-1:123456789012:pg:my-db-param-group"
  ],
  "detail": {
    "EventCategories": [
      "configuration change"
    ],
    "SourceType": "DB_PARAM",
    "SourceArn": "arn:aws:rds:us-east-1:123456789012:pg:my-db-param-group",
    "Date": "2018-10-06T12:26:13.882Z",
    "Message": "Updated parameter time_zone to UTC with apply method immediate",
    "SourceIdentifier": "my-db-param-group",
    "EventID": "RDS-EVENT-0037"
  }
}
```

Contoh peristiwa snapshot DB

Berikut ini adalah contoh peristiwa snapshot DB dalam format JSON. Peristiwa ini menunjukkan penghapusan snapshot bernama `my-db-snapshot`. ID peristiwanya adalah RDS-EVENT-0041.

```
{
  "version": "0",
  "id": "844e2571-85d4-695f-b930-0153b71dcb42",
  "detail-type": "RDS DB Snapshot Event",
  "source": "aws.rds",
  "account": "123456789012",
  "time": "2018-10-06T12:26:13Z",
  "region": "us-east-1",
```

```
  "resources": [
    "arn:aws:rds:us-east-1:123456789012:snapshot:rds:my-db-snapshot"
  ],
  "detail": {
    "EventCategories": [
      "deletion"
    ],
    "SourceType": "SNAPSHOT",
    "SourceArn": "arn:aws:rds:us-east-1:123456789012:snapshot:rds:my-db-snapshot",
    "Date": "2018-10-06T12:26:13.882Z",
    "Message": "Deleted manual snapshot",
    "SourceIdentifier": "my-db-snapshot",
    "EventID": "RDS-EVENT-0041"
  }
}
```

Memberikan izin untuk menerbitkan pemberitahuan ke topik Amazon SNS

Untuk memberikan izin Amazon RDS untuk menerbitkan pemberitahuan ke topik Amazon Simple Notification Service (Amazon SNS), lampirkan kebijakan AWS Identity and Access Management (IAM) ke topik tujuan. Untuk mengetahui informasi selengkapnya tentang izin, lihat Contoh kasus untuk kontrol akses Amazon Simple Notification Service di Panduan Developer Amazon Simple Notification Service.

Secara default, topik Amazon SNS memiliki kebijakan yang mengizinkan semua sumber daya Amazon RDS dalam akun yang sama untuk menerbitkan pemberitahuan ke akun tersebut. Anda dapat melampirkan kebijakan kustom untuk mengizinkan pemberitahuan lintas akun, atau untuk membatasi akses ke sumber daya tertentu.

Berikut ini adalah contoh kebijakan IAM yang Anda lampirkan ke topik Amazon SNS tujuan. Ini membatasi topik ke instans DB dengan nama yang cocok dengan awalan yang ditentukan. Untuk menggunakan kebijakan ini, tentukan nilai berikut:

- `Resource` – Amazon Resource Name (ARN) untuk topik Amazon SNS Anda
- `SourceARN` – ARN sumber data RDS Anda
- `SourceAccount` – ID Akun AWS Anda

Untuk melihat daftar jenis sumber daya dan ARN-nya, lihat Sumber Daya yang Ditentukan oleh Amazon RDS di Referensi Otorisasi Layanan.

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "events.rds.amazonaws.com"
      },
      "Action": [
        "sns:Publish"
      ],
      "Resource": "arn:aws:sns:us-east-1:123456789012:topic_name",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:rds:us-east-1:123456789012:db:prefix-*"
        },
```

```
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        }
      }
    }
  ]
}
```

Berlangganan pemberitahuan peristiwa Amazon RDS

Cara termudah untuk membuat langganan adalah dengan konsol RDS. Jika memilih untuk membuat langganan pemberitahuan peristiwa menggunakan CLI atau API, Anda harus membuat topik Amazon Simple Notification Service dan berlangganan topik tersebut dengan konsol Amazon SNS atau Amazon SNS API. Anda juga akan perlu mempertahankan Amazon Resource Name (ARN) topik tersebut karena digunakan saat mengirim perintah CLI atau operasi API. Untuk informasi tentang cara membuat topik SNS dan berlanggan, lihat Mulai menggunakan Amazon SNS dalam Panduan Developer Amazon Simple Notification Service.

Anda dapat menentukan jenis sumber yang pemberitahuannya ingin dikirim dan sumber Amazon RDS yang memicu peristiwa.

Jenis sumber

Jenis sumber. Misalnya, Jenis sumber mungkin berupa Instans. Anda harus memilih jenis sumber.

Sumber daya yang akan disertakan

Sumber daya Amazon RDS yang menghasilkan peristiwa. Misalnya, Anda dapat memilih Pilih instans tertentu, lalu myDBInstance1.

Tabel berikut menjelaskan hasil saat Anda menentukan atau tidak menentukan ***Sumber Daya*** yang akan disertakan.

Sumber daya yang akan disertakan	Deskripsi	Contoh
Ditentukan	RDS memberi tahu Anda tentang semua peristiwa hanya untuk sumber daya yang ditentukan.	Jika Jenis sumber Anda berupa Instans dan sumber daya Anda adalah MyDBInstance1, RDS akan memberi tahu Anda tentang semua peristiwa hanya untuk myDBInstance1.
Tidak ditentukan	RDS memberi tahu Anda tentang peristiwa untuk jenis sumber yang ditentukan untuk semua sumber daya Amazon RDS.	Jika Jenis sumber Anda berupa Instans, RDS akan memberi tahu

Sumber daya yang akan disertakan	Deskripsi	Contoh
		Anda tentang semua peristiwa terkait instans di akun Anda.

Secara default, pelanggan topik Amazon SNS menerima setiap pesan yang diterbitkan untuk topik tersebut. Untuk menerima subset pesan saja, pelanggan harus menetapkan kebijakan filter untuk langganan topik. Untuk informasi selengkapnya tentang pemfilteran pesan SNS, lihat Pemfilteran pesan Amazon SNS di Panduan Developer Amazon Notification Service

Konsol

Untuk berlangganan pemberitahuan peristiwa RDS

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Langganan peristiwa.
3. Di panel Langganan peristiwa, pilih Buat langganan peristiwa.
4. Masukkan detail langganan Anda sebagai berikut:

   a. Untuk Nama, masukkan nama langganan pemberitahuan peristiwa.

   b. Untuk Kirim pemberitahuan ke, lakukan salah satu langkah berikut:

      - Pilih Topik email baru. Masukkan nama topik email Anda dan daftar penerima. Sebaiknya Anda mengonfigurasi langganan peristiwa ke alamat email yang sama dengan kontak akun utama Anda. Rekomendasi, peristiwa layanan, dan pesan kesehatan pribadi dikirim menggunakan saluran yang berbeda. Langganan ke alamat email yang sama memastikan bahwa semua pesan digabungkan di satu lokasi.
      - Pilih Amazon Resource Name (ARN). Kemudian, pilih Amazon SNS ARN untuk topik Amazon SNS.

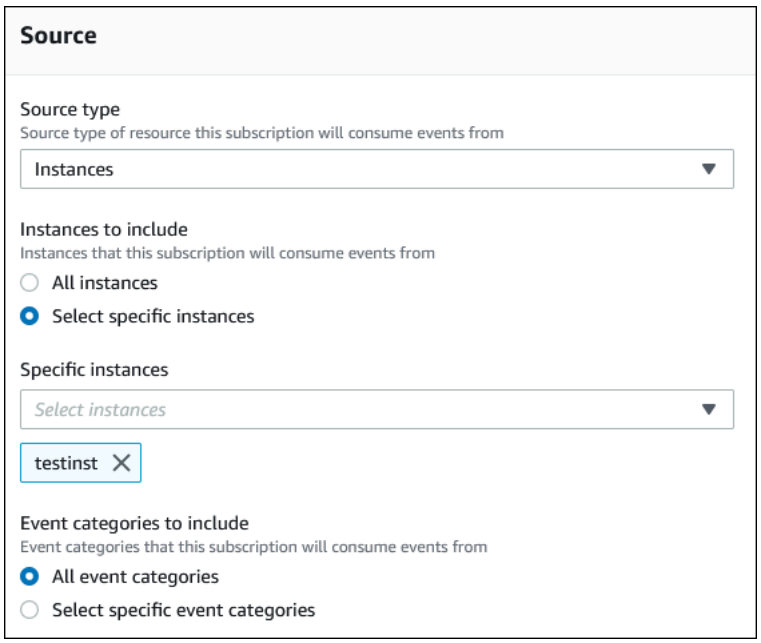
        Jika Anda ingin menggunakan topik yang telah diaktifkan untuk enkripsi di sisi server (SSE), beri Amazon RDS izin yang diperlukan untuk mengakses AWS KMS key. Untuk informasi selengkapnya, lihat Mengaktifkan kompatibilitas antara sumber peristiwa dari

layanan AWS dan topik terenkripsi di Panduan Developer Amazon Simple Notification Service.

c. Untuk Jenis sumber, pilih jenis sumber. Misalnya, pilih Instans atau Grup parameter .

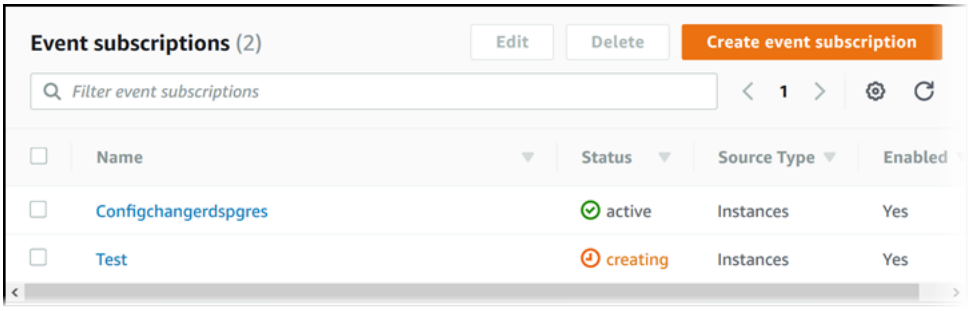
d. Pilih kategori dan sumber daya peristiwa yang pemberitahuan peristiwanya ingin diterima.

Contoh berikut mengonfigurasi pemberitahuan peristiwa untuk instans DB bernama `testinst`.



e. Pilih Buat.

Konsol Amazon RDS menunjukkan bahwa langganan sedang dibuat.



AWS CLI

Untuk berlangganan pemberitahuan peristiwa RDS, gunakan perintah AWS CLI `create-event-subscription`. Sertakan parameter wajib berikut:

- `--subscription-name`

- `--sns-topic-arn`

Example

Untuk Linux, macOS, atau Unix:

```
aws rds create-event-subscription \
    --subscription-name myeventsubscription \
    --sns-topic-arn arn:aws:sns:us-east-1:123456789012:myawsuser-RDS \
    --enabled
```

Untuk Windows:

```
aws rds create-event-subscription ^
    --subscription-name myeventsubscription ^
    --sns-topic-arn arn:aws:sns:us-east-1:123456789012:myawsuser-RDS ^
    --enabled
```

API

Untuk berlangganan pemberitahuan peristiwa Amazon RDS, panggil fungsi Amazon RDS API `CreateEventSubscription`. Sertakan parameter wajib berikut:

- `SubscriptionName`
- `SnsTopicArn`

Tag dan atribut pemberitahuan peristiwa Amazon RDS

Saat Amazon RDS mengirimkan pemberitahuan peristiwa ke Amazon Simple Notification Service (SNS) atau Amazon EventBridge, pemberitahuan tersebut berisi atribut pesan dan tag peristiwa. RDS mengirimkan atribut pesan secara terpisah bersama dengan pesan, sedangkan tag peristiwa berada dalam isi pesan. Gunakan atribut pesan dan tag Amazon RDS untuk menambahkan metadata ke sumber daya Anda. Anda dapat mengubah tag ini dengan notasi Anda sendiri tentang instans DB. Untuk mengetahui informasi selengkapnya tentang cara memberikan tag pada sumber daya Amazon RDS, lihat Memberi tag pada sumber daya Amazon RDS.

Secara default, Amazon SNS dan Amazon EventBridge menerima setiap pesan yang dikirim kepada mereka. SNS dan EventBridge dapat memfilter pesan dan mengirim pemberitahuan ke mode komunikasi yang diinginkan, seperti email, pesan teks, atau panggilan ke titik akhir HTTP.

> **Note**
>
> Pemberitahuan yang dikirim dalam email atau pesan teks tidak akan memiliki tag peristiwa.

Tabel berikut menunjukkan atribut pesan untuk peristiwa RDS yang dikirim ke pelanggan topik.

Atribut peristiwa Amazon RDS	Deskripsi
EventID	ID untuk pesan peristiwa RDS, misalnya, RDS-EVENT-0006.
Sumber daya	Pengidentifikasi ARN untuk sumber daya yang memancarkan peristiwa, misalnya, `arn:aws:rds:ap-southeast-2:123456789012:db:database-1` .

Tag RDS menyediakan data tentang sumber daya yang terpengaruh oleh peristiwa layanan. RDS menambahkan status tag saat ini dalam isi pesan saat pemberitahuan dikirim ke SNS atau EventBridge.

Untuk mengetahui informasi selengkapnya tentang cara memfilter atribut pesan untuk SNS, lihat Pemfilteran pesan Amazon SNS dalam Panduan Developer Amazon Simple Notification Service.

Untuk mengetahui informasi selengkapnya tentang cara memfilter tag peristiwa untuk EventBridge, lihat Pemfilteran konten dalam pola peristiwa Amazon EventBridge dalam Panduan Pengguna Amazon EventBridge.

Untuk mengetahui informasi selengkapnya tentang cara memfilter tag berbasis payload untuk SNS, lihat https://aws.amazon.com/blogs/compute/introducing-payload-based-message-filtering-for-amazon-sns/

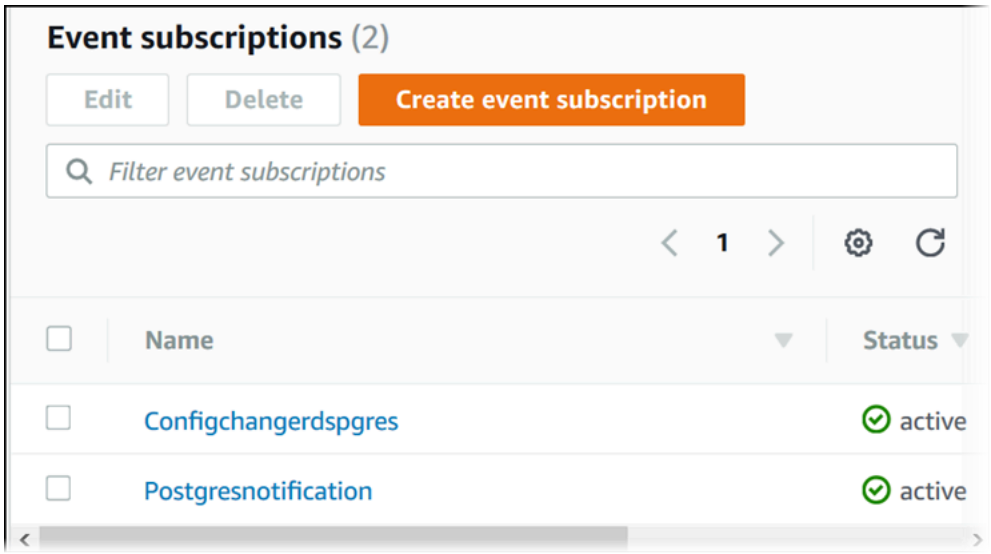
Mencantumkan langganan pemberitahuan peristiwa Amazon RDS

Anda dapat mencantumkan langganan pemberitahuan peristiwa Amazon RDS Anda saat ini.

Konsol

Untuk mencantumkan langganan pemberitahuan peristiwa Amazon RDS Anda saat ini

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Langganan peristiwa. Panel Langganan peristiwa menampilkan semua langganan pemberitahuan peristiwa.



AWS CLI

Untuk mencantumkan langganan pemberitahuan peristiwa Amazon RDS Anda saat ini, gunakan perintah AWS CLI `describe-event-subscriptions`.

Example

Contoh berikut menjelaskan semua langganan peristiwa.

```
aws rds describe-event-subscriptions
```

Contoh berikut menjelaskan `myfirsteventsubscription`.

```
aws rds describe-event-subscriptions --subscription-name myfirsteventsubscription
```

API

Untuk mencantumkan langganan pemberitahuan peristiwa Amazon RDS Anda saat ini, panggil tindakan Amazon RDS API `DescribeEventSubscriptions`.

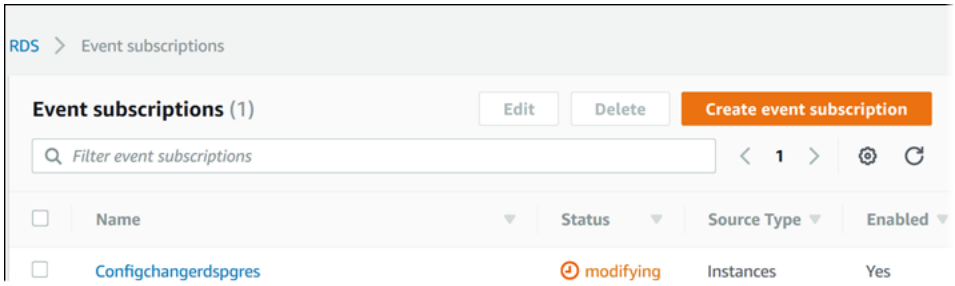
Mengubah langganan pemberitahuan peristiwa Amazon RDS

Setelah membuat langganan, Anda dapat mengubah nama langganan, pengidentifikasi sumber, kategori, atau ARN topik.

Konsol

Untuk mengubah langganan pemberitahuan peristiwa Amazon RDS

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Langganan peristiwa.
3. Di panel Langganan peristiwa, pilih langganan yang ingin diubah, lalu pilih Edit.
4. Buat perubahan pada langganan di bagian Target atau Sumber.
5. Pilih Edit. Konsol Amazon RDS menunjukkan bahwa langganan sedang diubah.



AWS CLI

Untuk mengubah langganan pemberitahuan peristiwa Amazon RDS, gunakan perintah AWS CLI `modify-event-subscription`. Sertakan parameter wajib berikut:

- `--subscription-name`

Example

Kode berikut mengaktifkan `myeventsubscription`.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-event-subscription \
    --subscription-name myeventsubscription \
```

```
    --enabled
```

Untuk Windows:

```
aws rds modify-event-subscription ^
    --subscription-name myeventsubscription ^
    --enabled
```

API

Untuk mengubah peristiwa Amazon RDS, panggil operasi Amazon RDS API `ModifyEventSubscription`. Sertakan parameter wajib berikut:

- `SubscriptionName`

Menambahkan pengidentifikasi sumber ke langganan pemberitahuan peristiwa Amazon RDS

Anda dapat menambahkan pengidentifikasi sumber (sumber Amazon RDS yang menghasilkan peristiwa) ke langganan yang sudah ada.

Konsol

Anda dapat dengan mudah menambahkan atau menghapus ID sumber menggunakan konsol Amazon RDS dengan memilih atau membatalkan pilihan saat memodifikasi langganan. Untuk informasi selengkapnya, lihat Mengubah langganan pemberitahuan peristiwa Amazon RDS.

AWS CLI

Untuk menambahkan pengidentifikasi sumber ke langganan pemberitahuan peristiwa Amazon RDS, gunakan perintah AWS CLI `add-source-identifier-to-subscription`. Sertakan parameter wajib berikut:

- `--subscription-name`
- `--source-identifier`

Example

Contoh berikut menambahkan pengidentifikasi sumber `mysqldb` ke langganan `myrdseventsubscription`.

Untuk Linux, macOS, atau Unix:

```
aws rds add-source-identifier-to-subscription \
    --subscription-name myrdseventsubscription \
    --source-identifier mysqldb
```

Untuk Windows:

```
aws rds add-source-identifier-to-subscription ^
    --subscription-name myrdseventsubscription ^
    --source-identifier mysqldb
```

API

Untuk menambahkan pengidentifikasi sumber ke langganan pemberitahuan peristiwa Amazon RDS, panggil Amazon RDS API `AddSourceIdentifierToSubscription`. Sertakan parameter wajib berikut:

- `SubscriptionName`
- `SourceIdentifier`

Menghapus pengidentifikasi sumber dari langganan pemberitahuan peristiwa Amazon RDS

Anda dapat menghapus pengidentifikasi sumber (sumber Amazon RDS yang menghasilkan peristiwa) dari langganan jika Anda tidak ingin diberi tahu lagi tentang peristiwa sumber tersebut.

Konsol

Anda dapat dengan mudah menambahkan atau menghapus ID sumber menggunakan konsol Amazon RDS dengan memilih atau membatalkan pilihan saat memodifikasi langganan. Untuk informasi selengkapnya, lihat Mengubah langganan pemberitahuan peristiwa Amazon RDS.

AWS CLI

Untuk menghapus pengidentifikasi sumber dari langganan pemberitahuan peristiwa Amazon RDS, gunakan perintah AWS CLI `remove-source-identifier-from-subscription`. Sertakan parameter wajib berikut:

- `--subscription-name`
- `--source-identifier`

Example

Contoh berikut menghapus pengidentifikasi sumber `mysqldb` dari langganan `myrdseventsubscription`.

Untuk Linux, macOS, atau Unix:

```
aws rds remove-source-identifier-from-subscription \
    --subscription-name myrdseventsubscription \
    --source-identifier mysqldb
```

Untuk Windows:

```
aws rds remove-source-identifier-from-subscription ^
    --subscription-name myrdseventsubscription ^
    --source-identifier mysqldb
```

API

Untuk menghapus pengidentifikasi sumber dari langganan pemberitahuan peristiwa Amazon RDS, gunakan perintah Amazon RDS API `RemoveSourceIdentifierFromSubscription`. Sertakan parameter wajib berikut:

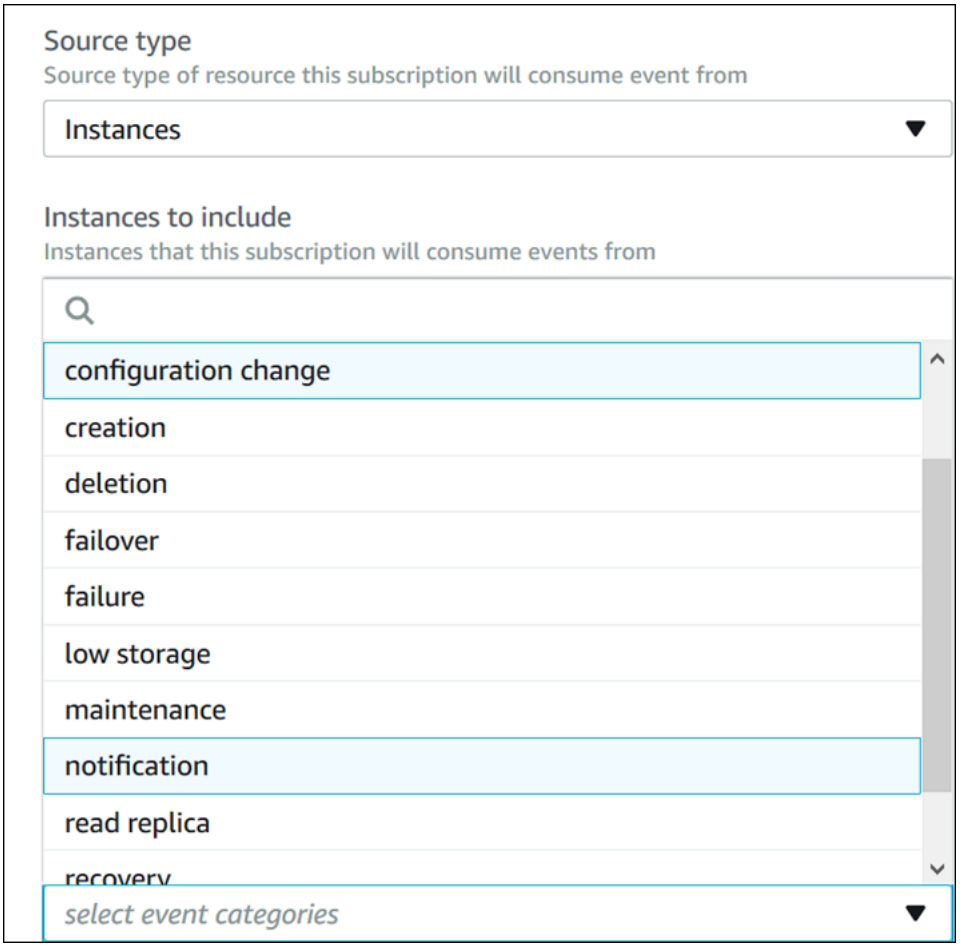
- `SubscriptionName`
- `SourceIdentifier`

Mencantumkan kategori pemberitahuan peristiwa Amazon RDS

Semua peristiwa untuk jenis sumber daya dikelompokkan dalam beberapa kategori. Untuk melihat daftar kategori yang tersedia, gunakan prosedur berikut.

Konsol

Jika Anda membuat atau memodifikasi langganan pemberitahuan peristiwa, kategori peristiwa akan ditampilkan di konsol Amazon RDS. Untuk mengetahui informasi selengkapnya, lihat Mengubah langganan pemberitahuan peristiwa Amazon RDS.



AWS CLI

Untuk mencantumkan kategori pemberitahuan peristiwa Amazon RDS, gunakan perintah AWS CLI `describe-event-categories`. Perintah ini tidak memiliki parameter yang diperlukan.

Example

```
aws rds describe-event-categories
```

API

Untuk mencantumkan kategori pemberitahuan peristiwa Amazon RDS, gunakan perintah Amazon RDS API `DescribeEventCategories`. Perintah ini tidak memiliki parameter yang diperlukan.

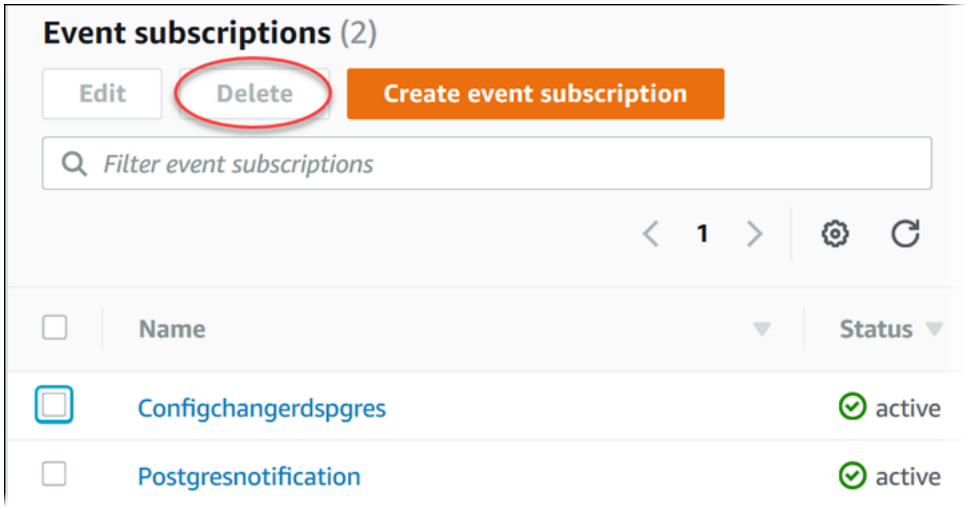
Menghapus langganan pemberitahuan peristiwa Amazon RDS

Anda dapat menghapus langganan jika sudah tidak membutuhkannya lagi. Semua pelanggan topik tidak akan lagi menerima pemberitahuan peristiwa yang ditentukan oleh langganan.

Konsol

Untuk menghapus langganan pemberitahuan peristiwa Amazon RDS

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Langganan Peristiwa DB.
3. Di panel Langganan Peristiwa DB Saya, pilih langganan yang ingin dihapus.
4. Pilih Hapus.
5. Konsol Amazon RDS menunjukkan bahwa langganan sedang diubah.



AWS CLI

Untuk menghapus langganan pemberitahuan peristiwa Amazon RDS, gunakan perintah AWS CLI `delete-event-subscription`. Sertakan parameter wajib berikut:

- `--subscription-name`

Example

Contoh berikut menghapus langganan `myrdssubscription`.

```
aws rds delete-event-subscription --subscription-name myrdssubscription
```

API

Untuk menghapus langganan pemberitahuan peristiwa Amazon RDS, gunakan perintah RDS API `DeleteEventSubscription`. Sertakan parameter wajib berikut:

- `SubscriptionName`

Membuat aturan yang memicu peristiwa Amazon RDS

Menggunakan Amazon CloudWatch Events dan Amazon EventBridge, Anda dapat mengotomatiskan layanan AWS dan merespons peristiwa sistem seperti masalah ketersediaan aplikasi atau perubahan sumber daya.

Topik

- Membuat aturan untuk mengirim peristiwa Amazon RDS ke CloudWatch Events
- Tutorial: Mencatat log perubahan status instans DB menggunakan Amazon EventBridge

Membuat aturan untuk mengirim peristiwa Amazon RDS ke CloudWatch Events

Anda dapat menulis aturan sederhana untuk menunjukkan peristiwa Amazon RDS mana yang menarik bagi Anda, dan tindakan otomatis mana yang diambil ketika ada peristiwa yang cocok dengan aturan. Anda dapat menetapkan berbagai target, seperti fungsi AWS Lambda atau topik Amazon SNS, yang menerima peristiwa dalam format JSON. Misalnya, Anda dapat mengonfigurasi Amazon RDS untuk mengirim peristiwa ke CloudWatch Events atau Amazon EventBridge setiap kali instans DB dibuat atau dihapus. Untuk informasi selengkapnya, lihat Panduan Pengguna Amazon CloudWatch Events dan Panduan Pengguna Amazon EventBridge.

Untuk membuat aturan yang memicu peristiwa RDS:

1. Buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.
2. Di bagian Peristiwa di panel navigasi, pilih Aturan.
3. Pilih Buat aturan.
4. Untuk Sumber Peristiwa, lakukan hal berikut:
    a. Pilih Pola Peristiwa.
    b. Untuk Nama Layanan, pilih Relational Database Service (RDS).
    c. Untuk Jenis Peristiwa, pilih jenis sumber daya Amazon RDS yang memicu peristiwa. Misalnya, jika sebuah instans DB memicu peristiwa, pilih Peristiwa Instans DB RDS.
5. Untuk Target, pilih Tambah Target dan pilih layanan AWS yang akan bertindak ketika peristiwa dengan jenis yang dipilih terdeteksi.
6. Di kolom lain pada bagian ini, masukkan informasi spesifik untuk jenis target ini, jika perlu.

7. Pada sebagian besar jenis target, CloudWatch Events memerlukan izin untuk mengirim peristiwa ke target. Dalam kasus ini, CloudWatch Events dapat membuat peran IAM yang diperlukan agar peristiwa Anda dapat berjalan:

   - Untuk membuat peran IAM secara otomatis, pilih Buat peran baru untuk sumber daya khusus ini.
   - Untuk menggunakan peran IAM yang Anda buat sebelumnya, pilih Gunakan peran yang sudah ada.

8. Secara opsional, ulangi langkah 5-7 untuk menambahkan target lain untuk aturan ini.
9. Pilih Konfigurasikan detail. Untuk Definisi aturan, ketik nama dan deskripsi aturan.

   Nama aturan harus unik dalam Wilayah ini.

10. Pilih Buat aturan.

Untuk informasi selengkapnya, lihat Membuat Aturan CloudWatch Events yang Memicu Peristiwa dalam Panduan Pengguna Amazon CloudWatch.

Tutorial: Mencatat log perubahan status instans DB menggunakan Amazon EventBridge

Tutorial ini menjelaskan cara membuat fungsi AWS Lambda yang mencatat log perubahan status instans Amazon RDS. Kemudian, Anda dapat membuat aturan yang menjalankan fungsi tersebut setiap kali terjadi perubahan status pada instans DB RDS yang ada. Tutorial ini mengasumsikan bahwa Anda memiliki instans uji coba kecil yang sedang berjalan yang dapat Anda hentikan untuk sementara waktu.

> **Important**
>
> Jangan lakukan tutorial ini pada instans DB produksi yang sedang berjalan.

Topik

- Langkah 1: Membuat fungsi AWS Lambda
- Langkah 2: Buat Aturan
- Langkah 3: Uji aturan

Langkah 1: Membuat fungsi AWS Lambda

Buat fungsi Lambda untuk mencatat log peristiwa perubahan status. Anda menetapkan fungsi ini saat membuat aturan.

Untuk membuat fungsi Lambda

1. Buka konsol AWS Lambda di https://console.aws.amazon.com/lambda/.
2. Jika baru menggunakan Lambda, Anda akan melihat halaman selamat datang. Pilih Mulai Sekarang. Atau, pilih Buat fungsi.
3. Pilih Tulis dari awal.
4. Di halaman Buat fungsi, lakukan langkah berikut:
   a. Masukkan nama dan deskripsi fungsi Lambda. Misalnya, beri nama fungsi **`RDSInstanceStateChange`**.
   b. Di bagian Runtime, pilih Node.js 16x.
   c. Untuk Arsitektur, pilih x86_64.
   d. Untuk Peran eksekusi, lakukan salah satu langkah berikut:
      - Pilih Buat peran baru dengan izin Lambda dasar.
      - Untuk Peran yang sudah ada, pilih Gunakan peran yang sudah ada. Pilih peran yang ingin digunakan.
   e. Pilih Buat fungsi.
5. Di halaman RDSInstanceStateChange, lakukan langkah berikut:
   a. Di Sumber kode, pilih index.js.
   b. Di panel index.js, hapus kode yang ada.
   c. Masukkan kode berikut:

```
console.log('Loading function');

exports.handler = async (event, context) => {
    console.log('Received event:', JSON.stringify(event));
};
```

   d. Pilih Deploy.

Langkah 2: Buat Aturan

Buat aturan untuk menjalankan fungsi Lambda setiap kali Anda meluncurkan instans Amazon RDS.

Untuk membuat aturan EventBridge

1. Buka konsol Amazon EventBridge di https://console.aws.amazon.com/events/.
2. Di panel navigasi, pilih Aturan.
3. Pilih Buat aturan.
4. Masukkan nama dan deskripsi qaturan. Misalnya, masukkan **`RDSInstanceStateChangeRule`**.
5. Pilih Aturan dengan pola peristiwa, lalu pilih Berikutnya.
6. Untuk Sumber peristiwa, pilih Peristiwa AWS atau peristiwa mitra EventBridge.
7. Gulir ke bawah ke bagian Pola peristiwa.
8. Untuk Sumber peristiwa, pilih Layanan AWS.
9. Untuk layanan AWS, pilih Relational Database Service (RDS).
10. Untuk Jenis peristiwa, pilih Peristiwa Instans DB RDS.
11. Jangan ubah pola peristiwa default. Lalu pilih Berikutnya.
12. Untuk Jenis Target, pilih Layanan AWS.
13. Untuk Pilih target, pilih Fungsi Lambda.
14. Untuk Fungsi, pilih fungsi Lambda yang Anda buat. Lalu pilih Berikutnya.
15. Di Konfigurasikan tag, pilih Berikutnya.
16. Tinjau langkah-langkah dalam aturan Anda. Kemudian, pilih Buat aturan.

Langkah 3: Uji aturan

Untuk menguji aturan, nonaktifkan instans DB RDS. Setelah menunggu proses penonaktifan instans selama beberapa menit, pastikan fungsi Lambda Anda sudah diinvokasi.

Untuk menguji aturan dengan menghentikan instans DB

1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Hentikan instans DB RDS.
3. Buka konsol Amazon EventBridge di https://console.aws.amazon.com/events/.
4. Di panel navigasi, pilih Aturan, pilih nama aturan yang Anda buat.

5. Di Detail aturan, pilih Pemantauan.

   Anda akan dialihkan ke konsol Amazon CloudWatch. Jika Anda tidak dialihkan, klik Lihat metrik di CloudWatch.

6. Dalam Semua metrik, pilih nama aturan yang Anda buat.

   Grafik harus menunjukkan bahwa aturan telah diinvokasi.

7. Di panel navigasi, pilih Grup log.

8. Pilih nama grup log untuk fungsi Lambda Anda (/aws/lambda/***function-name***).

9. Pilih nama log stream untuk melihat data yang disediakan oleh fungsi untuk instans yang Anda luncurkan. Anda akan melihat peristiwa seperti berikut:

```
{
    "version": "0",
    "id": "12a345b6-78c9-01d2-34e5-123f4ghi5j6k",
    "detail-type": "RDS DB Instance Event",
    "source": "aws.rds",
    "account": "111111111111",
    "time": "2021-03-19T19:34:09Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:rds:us-east-1:111111111111:db:testdb"
    ],
    "detail": {
        "EventCategories": [
            "notification"
        ],
        "SourceType": "DB_INSTANCE",
        "SourceArn": "arn:aws:rds:us-east-1:111111111111:db:testdb",
        "Date": "2021-03-19T19:34:09.293Z",
        "Message": "DB instance stopped",
        "SourceIdentifier": "testdb",
        "EventID": "RDS-EVENT-0087"
    }
}
```

   Untuk melihat contoh lain dari peristiwa RDS dalam format JSON, lihat Ikhtisar peristiwa untuk Amazon RDS.

10. (Opsional) Setelah selesai, Anda dapat membuka konsol Amazon RDS dan memulai instans yang Anda hentikan.

Kategori peristiwa dan pesan peristiwa Amazon RDS

Amazon RDS menghasilkan sejumlah besar peristiwa dalam kategori yang dapat Anda berlangganan menggunakan Konsol Amazon RDS, AWS CLI, atau API.

Topik

Peristiwa klaster DB

Tabel berikut menunjukkan kategori peristiwa dan daftar peristiwa saat klaster DB berupa jenis sumber.

Untuk mengetahui informasi selengkapnya tentang deployment klaster DB Multi-AZ, lihat Deployment klaster basis data Multi-AZ

Kategori	ID peristiwa RDS	Pesan	Catatan
pembuatan	RDS-EVENT-0170	Klaster DB dibuat.	
failover	RDS-EVENT-0069	Failover klaster gagal, periksa kondisi instans klaster dan coba lagi.	
failover	RDS-EVENT-0070	Mendorong primer sebelumnya lagi: *name*.	

Kategori	ID peristiwa RDS	Pesan	Catatan
failover	RDS-EVENT-0071	Menyelesaikan failover ke instans DB: *name*.	
failover	RDS-EVENT-0072	Memulai failover AZ yang sama ke instans DB: *name*.	
failover	RDS-EVENT-0073	Memulai failover lintas AZ ke instans DB: *name*.	
kegagalan	RDS-EVENT-0354	Anda tidak dapat membuat klaster DB karena sumber daya yang tidak kompatibel. *pesan*.	*Pesan* ini mencakup detail tentang kegagalan.
kegagalan	RDS-EVENT-0355	Klaster DB tidak dapat dibuat karena batas sumber daya yang tidak memadai. *pesan*.	*Pesan* ini mencakup detail tentang kegagalan.
failover global	RDS-EVENT-0181	Switchover global ke klaster DB *name* di Wilayah *name* dimulai.	Peristiwa ini untuk operasi switchover (sebelumnya disebut "failover terencana terkelola"). Prosesnya dapat ditunda karena operasi lain berjalan di klaster DB.

Kategori	ID peristiwa RDS	Pesan	Catatan
failover global	RDS-EVENT-0182	Klaster DB primer lama *name* di Wilayah *name* berhasil dimatikan.	Peristiwa ini untuk operasi switchover (sebelumnya disebut "failover terencana terkelola"). Instans primer lama dalam basis data global tidak menerima penulisan. Semua volume disinkron kan.
failover global	RDS-EVENT-0183	Menunggu sinkronisasi data di seluruh anggota klaster global. Arus tertinggal di belakang klaster DB primer: *reason*.	Peristiwa ini untuk operasi switchover (sebelumnya disebut "failover terencana terkelola"). Kelambatan replikasi terjadi selama fase sinkronisasi failover basis data global.
failover global	RDS-EVENT-0184	Klaster DB primer baru *name* di Wilayah *name* berhasil dipromosikan.	Peristiwa ini untuk operasi switchover (sebelumnya disebut "failover terencana terkelola"). Topologi volume basis data global dibangun kembali dengan volume primer baru.

Kategori	ID peristiwa RDS	Pesan	Catatan
failover global	RDS-EVENT-0185	Switchover global ke klaster DB *name* di Wilayah *name* selesai.	Peristiwa ini untuk operasi switchover (sebelumnya disebut "failover terencana terkelola"). Switchover basis data global selesai di klaster DB primer. Replika mungkin membutuhkan waktu lama untuk online setelah failover selesai.
failover global	RDS-EVENT-0186	Switchover global ke klaster DB *name* di Wilayah *name* dibatalkan.	Peristiwa ini untuk operasi switchover (sebelumnya disebut "failover terencana terkelola").
failover global	RDS-EVENT-0187	Switchover global ke klaster DB *name* di Wilayah *name* gagal.	Peristiwa ini untuk operasi switchover (sebelumnya disebut "failover terencana terkelola").
failover global	RDS-EVENT-0238	Failover global ke klaster DB *name* di Wilayah *name* selesai.	
failover global	RDS-EVENT-0239	Failover global ke klaster DB *name* di Wilayah *name* gagal.	
failover global	RDS-EVENT-0240	Memulai sinkronisasi ulang anggota klaster DB *name* di Wilayah *name* setelah failover global.	

Kategori	ID peristiwa RDS	Pesan	Catatan
failover global	RDS-EVENT-0241	Menyelesaikan sinkronisasi ulang anggota klaster DB *name* di Wilayah *name* setelah failover global.	
pemeliharaan	RDS-EVENT-0176	Versi mayor mesin klaster basis data telah ditingkatkan.	
pemeliharaan	RDS-EVENT-0286	Peningkatan versi mesin klaster basis data dimulai.	
pemeliharaan	RDS-EVENT-0287	Persyaratan peningkatan sistem operasi terdeteksi.	
pemeliharaan	RDS-EVENT-0288	Peningkatan sistem operasi klaster dimulai.	
pemeliharaan	RDS-EVENT-0289	Peningkatan sistem operasi klaster selesai.	
pemeliharaan	RDS-EVENT-0290	Klaster basis data telah di-patch: versi sumber *version_number* => *new_version_number* .	
pemberitahuan	RDS-EVENT-0172	Mengganti nama klaster dari *name* menjadi *name*.	

Peristiwa instans DB

Tabel berikut menunjukkan kategori peristiwa dan daftar peristiwa saat instans DB merupakan jenis sumber.

Kategori	ID peristiwa RDS	Pesan	Catatan
ketersediaan	RDS-EVENT-0004	Instans DB dimatikan.	
ketersediaan	RDS-EVENT-0006	Instans DB dimulai ulang.	
ketersediaan	RDS-EVENT-0022	Terjadi kesalahan saat memulai ulang mysql: *pesan*.	Terjadi kesalahan saat memulai ulang MySQL.
ketersediaan	RDS-EVENT-0221	Instans DB telah mencapai ambang batas penyimpan an penuh, dan basis data telah dimatikan. Anda dapat meningkat kan penyimpanan yang dialokasikan untuk mengatasi masalah ini.	
ketersediaan	RDS-EVENT-0222	Kapasitas penyimpan an gratis untuk instans DB *name* rendah pada *percentage* dari penyimpanan yang dialokasikan [Penyimpa nan yang dialokasikan: *amount*, Penyimpanan kosong: *amount*]. Basis data akan dimatikan untuk mencegah kerusakan jika penyimpanan yang kosong lebih rendah dari *amount*. Anda dapat meningkatkan penyimpan an yang dialokasikan untuk mengatasi masalah ini.	Untuk informasi selengkap nya, lihat Penyimpanan instans DB Amazon RDS.

Kategori	ID peristiwa RDS	Pesan	Catatan
pencadangan	RDS-EVENT-0001	Mencadangkan instans DB.	
pencadangan	RDS-EVENT-0002	Menyelesaikan pencadangan instans DB.	
pencadangan	RDS-EVENT-0086	Kami tidak dapat mengaitkan nama grup opsi *name* dengan instans basis data *name*. Konfirmasikan bahwa grup opsi *name* didukung pada kelas dan konfigurasi instans DB Anda. Jika demikian, verifikasi semua pengaturan grup opsi dan coba lagi.	Untuk mengetahui informasi selengkapnya, lihat Menggunakan grup opsi.
perubahan konfigurasi	RDS-EVENT-0011	Diperbarui untuk menggunakan Parameter Group *nama* DB.	
perubahan konfigurasi	RDS-EVENT-0012	Menerapkan modifikasi pada kelas instans basis data.	
perubahan konfigurasi	RDS-EVENT-0014	Menyelesaikan penerapan modifikasi pada kelas instans DB.	
perubahan konfigurasi	RDS-EVENT-0016	Atur ulang kredensial utama.	
perubahan konfigurasi	RDS-EVENT-0017	Menyelesaikan penerapan modifikasi pada penyimpanan yang dialokasikan.	

Kategori	ID peristiwa RDS	Pesan	Catatan
perubahan konfigurasi	RDS-EVENT-0018	Menerapkan modifikasi pada penyimpanan yang dialokasikan.	
perubahan konfigurasi	RDS-EVENT-0024	Menerapkan modifikasi untuk dikonversi menjadi instans DB Multi-AZ.	
perubahan konfigurasi	RDS-EVENT-0025	Menyelesaikan penerapan modifikasi untuk dikonversi menjadi instans DB Multi-AZ.	
perubahan konfigurasi	RDS-EVENT-0028	Menonaktifkan pencadangan otomatis.	
perubahan konfigurasi	RDS-EVENT-0029	Menyelesaikan penerapan modifikasi untuk dikonversi menjadi instans DB standar (AZ Tunggal).	
perubahan konfigurasi	RDS-EVENT-0030	Menerapkan modifikasi untuk dikonversi menjadi instans DB standar (AZ Tunggal).	
perubahan konfigurasi	RDS-EVENT-0032	Mengaktifkan pencadangan otomatis.	
perubahan konfigurasi	RDS-EVENT-0033	Ada *number* pengguna yang cocok dengan nama pengguna utama; hanya mengatur ulang yang tidak terikat dengan host tertentu.	

Kategori	ID peristiwa RDS	Pesan	Catatan
perubahan konfigurasi	RDS-EVENT-0067	Tidak dapat mengatur ulang kata sandi Anda. Informasi kesalahan: *pesan*.	
perubahan konfigurasi	RDS-EVENT-0078	Interval Pemantauan diubah menjadi *number*.	Konfigurasi Pemantaua n yang Ditingkatkan telah diubah.
perubahan konfigurasi	RDS-EVENT-0092	Menyelesaikan pembaruan grup parameter DB.	
perubahan konfigurasi	RDS-EVENT-0217	Menerapkan modifikasi yang diinisiasi penskalaan otomatis ke penyimpanan yang dialokasikan.	
perubahan konfigurasi	RDS-EVENT-0218	Menyelesaikan penerapan modifikasi yang diinisias i penskalaan otomatis ke penyimpanan yang dialokasikan.	
perubahan konfigurasi	RDS-EVENT-0295	Peningkatan konfigurasi penyimpanan dimulai.	
perubahan konfigurasi	RDS-EVENT-0296	Peningkatan konfigurasi penyimpanan selesai.	
pembuatan	RDS-EVENT-0005	Instans DB dibuat.	
penghapusan	RDS-EVENT-0003	Instans DB dihapus.	
failover	RDS-EVENT-0013	Failover instans Multi-AZ dimulai.	Failover Multi-AZ yang menghasilkan promosi instans DB siaga telah dimulai.

Kategori	ID peristiwa RDS	Pesan	Catatan
failover	RDS-EVENT-0015	Failover Multi-AZ ke siaga selesai - Penyebaran DNS mungkin memakan waktu beberapa menit.	Failover Multi-AZ yang menghasilkan promosi instans DB siaga selesai. Mungkin dibutuhkan waktu beberapa menit bagi DNS untuk mentransfer ke instans DB primer baru.
failover	RDS-EVENT-0034	Mengabaikan failover yang diminta pengguna karena failover baru terjadi pada instans basis data.	Amazon RDS tidak mencoba failover yang diminta karena failover baru-baru ini terjadi pada instans DB.
failover	RDS-EVENT-0049	Failover instans Multi-AZ selesai.	
failover	RDS-EVENT-0050	Aktivasi instans Multi-AZ dimulai.	Aktivasi multi-AZ telah dimulai setelah pemulihan instans DB berhasil.
failover	RDS-EVENT-0051	Aktivasi instans Multi-AZ selesai.	Aktivasi Multi-AZ selesai. Basis data Anda seharusny a dapat diakses sekarang.
failover	RDS-EVENT-0065	Dipulihkan dari failover parsial.	
kegagalan	RDS-EVENT-0031	Instans DB dimasukkan dalam status *name*. RDS merekomendasikan agar Anda memulai a. point-in-time-restore	Instans DB gagal karena konfigurasi tidak kompatibe l atau masalah penyimpan an yang mendasarinya. Mulailah a point-in-time-rest ore untuk instance DB.

Kategori	ID peristiwa RDS	Pesan	Catatan
kegagalan	RDS-EVENT-0035	Instans basis data dimasukkan dalam *state*. *pesan*.	Instans DB memiliki parameter yang tidak valid. Misalnya, jika instans DB tidak dapat dimulai karena parameter terkait memori diatur terlalu tinggi untuk kelas instans ini, tindakan Anda adalah memodifik asi parameter memori dan mem-boot ulang instans DB.
kegagalan	RDS-EVENT-0036	Instans basis data dalam *state*. *pesan*.	Instans DB ada di jaringan yang tidak kompatibel. Beberapa ID subnet yang ditentukan tidak valid atau tidak ada.
kegagalan	RDS-EVENT-0058	Penginstalan Statspack gagal. *pesan*.	Terjadi kesalahan saat membuat akun pengguna Oracle Statspack PERFSTAT. Hapus akun sebelum Anda menambahk an opsi STATSPACK .

Kategori	ID peristiwa RDS	Pesan	Catatan
kegagalan	RDS-EVENT-0079	Amazon RDS tidak dapat membuat kredensial untuk pemantauan yang ditingkatkan dan fitur ini telah dinonaktifkan. Ini kemungkinan karena rds-monitoring-role tidak ada dan dikonfigurasi dengan benar di akun Anda. Untuk detail selengkapnya, lihat bagian pemecahan masalah dalam dokumentasi Amazon RDS.	Pemantauan yang Ditingkatkan tidak dapat diaktifkan tanpa peran IAM Pemantauan yang Ditingkatkan. Untuk mengetahui informasi tentang cara membuat peran IAM, lihat Untuk membuat peran IAM untuk pemantauan yang ditingkatkan Amazon RDS.
kegagalan	RDS-EVENT-0080	Amazon RDS tidak dapat mengonfigurasi pemantauan yang ditingkatkan pada instans Anda: *name* dan fitur ini telah dinonaktifkan. Ini kemungkinan karena rds-monitoring-role tidak ada dan dikonfigurasi dengan benar di akun Anda. Untuk detail selengkapnya, lihat bagian pemecahan masalah dalam dokumentasi Amazon RDS.	Pemantauan yang Ditingkatkan dinonaktifkan karena kesalahan terjadi saat perubahan konfigurasi. Sepertinya peran IAM Pemantauan yang Ditingkatkan tidak dikonfigurasi dengan benar. Untuk mengetahui informasi tentang cara membuat peran IAM pemantauan yang ditingkatkan, lihat Untuk membuat peran IAM untuk pemantauan yang ditingkatkan Amazon RDS.

Kategori	ID peristiwa RDS	Pesan	Catatan
kegagalan	RDS-EVENT-0081	Amazon RDS tidak dapat membuat kredensial untuk opsi *name*. Ini karena peran IAM *name* tidak dikonfigurasi dengan benar di akun Anda. Untuk detail selengkapnya, lihat bagian pemecahan masalah dalam dokumentasi Amazon RDS.	Peran IAM yang Anda gunakan untuk mengakses bucket Amazon S3 untuk pencadangan dan pemulihan native SQL Server tidak dikonfigu rasi dengan benar. Untuk informasi selengkap nya, lihat Menyiapka n pencadangan dan pemulihan native.
kegagalan	RDS-EVENT-0165	Instans DB RDS Custom berada di luar perimeter dukungan.	Anda bertanggung jawab untuk memperbaiki masalah konfigurasi yang menempatkan instans DB RDS Custom Anda dalam status `unsupport ed-configuration` . Jika masalahnya ada pada AWS infrastruktur, Anda dapat menggunak an konsol atau AWS CLI untuk memperbaikinya. Jika masalahnya ada pada sistem operasi atau konfigurasi basis data, Anda dapat masuk ke host untuk memperbaikinya. Untuk informasi selengkap nya, lihat Perimeter dukungan RDS Custom.

Kategori	ID peristiwa RDS	Pesan	Catatan
kegagalan	RDS-EVENT-0188	Instans DB berada dalam status yang tidak dapat ditingkatkan. *pesan*	Amazon RDS tidak dapat meningkatkan instans DB MySQL dari versi 5.7 ke versi 8.0 karena tidak kompatibel dengan kamus data. Instans DB diluncurkan kembali ke MySQL versi 5.7. Untuk informasi selengkapnya, lihat Rollback setelah kegagalan untuk meng-upgrade dari MySQL 5.7 ke 8.0.
kegagalan	RDS-EVENT-0219	Instans DB dalam status tidak valid. Tidak ada tindakan yang diperlukan. Penskalaan otomatis akan dicoba lagi nanti.	
kegagalan	RDS-EVENT-0220	Instans DB berada dalam periode pendinginan untuk operasi penyimpanan skala sebelumnya. Kami mengoptimalkan instans DB Anda. Ini membutuhk an waktu minimal 6 jam. Tidak ada tindakan yang diperlukan. Penskalaan otomatis akan dicoba lagi setelah periode pendingin an.	

Kategori	ID peristiwa RDS	Pesan	Catatan
kegagalan	RDS-EVENT-0223	Penskalaan otomatis penyimpanan tidak dapat menskalakan penyimpanan karena alasan ini: *reason*.	
kegagalan	RDS-EVENT-0224	Penskalaan otomatis penyimpanan telah memicu tugas penyimpanan skala tertunda yang akan mencapai atau melebihi ambang batas penyimpan an maksimum. Tingkatka n ambang penyimpanan maksimum.	
kegagalan	RDS-EVENT-0237	Instans DB memiliki jenis penyimpanan yang saat ini tidak tersedia di Zona Ketersediaan. Penskalaan otomatis akan dicoba lagi nanti.	
kegagalan	RDS-EVENT-0254	Kuota penyimpanan yang mendasari untuk akun pelanggan ini telah melampaui batas. Tambah kuota penyimpanan yang diizinkan agar penskalaa n dapat berjalan pada instans.	
kegagalan	RDS-EVENT-0278	Pembuatan instans DB gagal. *pesan*	*Pesan* ini mencakup detail tentang kegagalan.
kegagalan	RDS-EVENT-0279	Promosi replika baca RDS Custom gagal. *pesan*	*Pesan* ini mencakup detail tentang kegagalan.

Kategori	ID peristiwa RDS	Pesan	Catatan
kegagalan	RDS-EVENT-0280	RDS Custom tidak dapat meningkatkan instans DB karena pra-pemeriksaan gagal. *pesan*	*Pesan* ini mencakup detail tentang kegagalan.
kegagalan	RDS-EVENT-0281	RDS Custom tidak dapat memodifikasi instans DB karena pra-pemeriksaan gagal. *pesan*	*Pesan* ini mencakup detail tentang kegagalan.
kegagalan	RDS-EVENT-0282	RDS Custom tidak dapat memodifikasi instans DB karena izin IP Elastis tidak benar. Konfirmasi bahwa alamat IP Elastis ditandai dengan `AWSRDSCustom` .	
kegagalan	RDS-EVENT-0283	RDS Custom tidak dapat memodifikasi instans DB karena batas IP Elastis telah tercapai di akun Anda. Lepaskan IP Elastis yang tidak terpakai atau minta penambahan kuota untuk batas alamat IP Elastis Anda.	
kegagalan	RDS-EVENT-0284	RDS Custom tidak dapat mengonversi instans ke ketersediaan tinggi karena pra-pemeriksaan gagal. *pesan*	*Pesan* ini mencakup detail tentang kegagalan.

Kategori	ID peristiwa RDS	Pesan	Catatan
kegagalan	RDS-EVENT-0285	RDS Custom tidak dapat membuat snapshot akhir untuk instans DB karena *pesan*.	*Pesan* ini mencakup detail tentang kegagalan.
kegagalan	RDS-EVENT-0306	Peningkatan konfigurasi penyimpanan gagal. Coba tingkatkan lagi.	
kegagalan	RDS-EVENT-0315	Tidak dapat memindahkan basis data jaringan yang tidak kompatibel, *name*, ke status tersedia: *pesan*	Konfigurasi jaringan basis data tidak valid. Basis data tidak dapat dipindahkan dari jaringan yang tidak kompatibel ke tersedia.
kegagalan	RDS-EVENT-0328	Gagal bergabung dengan host ke domain. Status keanggotaan domain misalnya nama *instancename* telah disetel ke Gagal.	
kegagalan	RDS-EVENT-0329	Gagal bergabung dengan host ke domain Anda. Selama proses bergabung domain, Microsoft Windows mengembalikan *pesan* kode kesalahan. Verifikasi konfigurasi jaringan dan izin Anda dan keluarkan `modify-db-instance` permintaan untuk mencoba kembali domain join.	Saat menggunakan Direktori Aktif yang dikelola sendiri, lihatPemecahan masalah Directory Active yang dikelola sendiri.

Kategori	ID peristiwa RDS	Pesan	Catatan
kegagalan	RDS-EVENT-0353	Instans DB tidak dapat dibuat karena batas sumber daya tidak mencukupi. *pesan*.	*Pesan* ini mencakup detail tentang kegagalan.
kegagalan	RDS-ACARA-0356	RDS tidak dapat mengonfigurasi titik akhir Kerberos di domain Anda. Ini mungkin mencegah otentikasi Kerberos untuk instans DB Anda. Verifikasi konfigurasi jaringan antara instans DB dan pengontrol domain Anda.	
penyimpanan rendah	RDS-EVENT-0007	Penyimpanan yang dialokasikan telah habis. Alokasikan penyimpanan tambahan untuk mengatasi masalah.	Penyimpanan yang dialokasikan untuk instans DB telah dipakai. Untuk mengatasi masalah ini, alokasikan penyimpanan tambahan untuk instans DB. Untuk mengetahui informasi selengkapnya, lihat Pertanyaan Umum tentang RDS. Anda dapat memantau ruang penyimpanan untuk instans DB menggunakan metrik Ruang Penyimpanan Kosong.

Kategori	ID peristiwa RDS	Pesan	Catatan
penyimpanan rendah	RDS-EVENT-0089	Kapasitas penyimpanan kosong untuk instans DB: *name* rendah pada *percentage* dari penyimpanan yang disediakan [Penyimpanan yang Disediakan: *size*, Ruang Kosong: *size*]. Anda mungkin ingin menambah penyimpanan yang disediakan untuk mengatasi masalah ini.	Instans DB telah menggunakan lebih dari 90% dari penyimpanan yang dialokasikan. Anda dapat memantau ruang penyimpanan untuk instans DB menggunakan metrik Ruang Penyimpanan Kosong.
penyimpanan rendah	RDS-EVENT-0227	Penyimpanan klaster Aurora Anda sangat rendah dengan hanya *amount* terabyte yang tersisa. Ambil tindakan untuk mengurangi muatan penyimpanan pada klaster Anda.	Ruang subsistem penyimpanan Aurora hampir habis.
pemeliharaan	RDS-EVENT-0026	Menerapkan patch off-line ke instans DB.	Pemeliharaan offline instans DB sedang berlangsung. Instans DB saat ini tidak tersedia.
pemeliharaan	RDS-EVENT-0027	Menyelesaikan penerapan patch off-line ke instans DB.	Pemeliharaan offline instans DB selesai. Instans DB kini tersedia.
pemeliharaan	RDS-EVENT-0047	Instans basis data di-patch.	
pemeliharaan	RDS-EVENT-0155	Instans DB memiliki peningkatan versi minor mesin DB yang tersedia.	

Kategori	ID peristiwa RDS	Pesan	Catatan
pemeliharaan	RDS-EVENT-0264	Pra-pemeriksaan dimulai untuk peningkatan versi mesin DB.	
pemeliharaan	RDS-EVENT-0265	Pra-pemeriksaan selesai untuk peningkatan versi mesin DB.	
pemeliharaan	RDS-EVENT-0266	Waktu henti dimulai untuk instans DB.	
pemeliharaan	RDS-EVENT-0267	Peningkatan versi mesin dimulai.	
pemeliharaan	RDS-EVENT-0268	Peningkatan versi mesin selesai.	
pemeliharaan	RDS-EVENT-0269	Tugas pasca-peningkatan sedang berlangsung.	
pemeliharaan	RDS-EVENT-0270	Peningkatan versi mesin DB gagal. Pemulihan peningkatan versi mesin berhasil.	
pemeliharaan, kegagalan	RDS-EVENT-0195	*pesan*	Pembaruan file zona waktu Oracle gagal. Untuk informasi selengkapnya, lihat Pemutakhiran otomatis file zona waktu Oracle.

Kategori	ID peristiwa RDS	Pesan	Catatan
pemeliharaan, pemberitahuan	RDS-EVENT-0191	File zona waktu versi baru tersedia untuk diperbarui.	Jika Anda memperbarui mesin DB RDS for Oracle, Amazon RDS akan menghasilkan peristiwa ini jika Anda belum memilih peningkatan file zona waktu dan basis data tidak menggunakan file zona waktu DST terbaru yang tersedia pada instans. Untuk informasi selengkapnya, lihat Pemutakhiran otomatis file zona waktu Oracle.
pemeliharaan, pemberitahuan	RDS-EVENT-0192	Pembaruan file zona waktu Anda telah dimulai.	Peningkatan file zona waktu Oracle Anda telah dimulai. Untuk informasi selengkapnya, lihat Pemutakhiran otomatis file zona waktu Oracle.

Kategori	ID peristiwa RDS	Pesan	Catatan
pemeliharaan, pemberitahuan	RDS-EVENT-0193	Tidak ada pembaruan yang tersedia untuk versi file zona waktu saat ini.	Instans DB Oracle Anda menggunakan versi file zona waktu terbaru, dan salah satu dari pernyataan berikut terjadi: • Anda baru-baru ini menambahkan opsi `TIMEZONE_FILE_AUTOUPGRADE` . • Mesin DB Oracle sedang ditingkatkan. Untuk informasi selengkapnya, lihat Pemutakhiran otomatis file zona waktu Oracle.
pemeliharaan, pemberitahuan	RDS-EVENT-0194	Pembaruan file zona waktu Anda telah selesai.	Pembaruan file zona waktu Oracle Anda telah selesai. Untuk informasi selengkapnya, lihat Pemutakhiran otomatis file zona waktu Oracle.
pemberitahuan	RDS-EVENT-0044	*pesan*	Ini adalah pemberitahuan yang dikeluarkan operator. Untuk mengetahui informasi selengkapnya, lihat pesan peristiwa.

Kategori	ID peristiwa RDS	Pesan	Catatan
pemberitahuan	RDS-EVENT-0048	Menunda peningkatan mesin basis data karena instans ini telah membaca replika yang perlu ditingkatkan terlebih dahulu.	Patching instans DB telah ditunda.
pemberitahuan	RDS-EVENT-0054	*pesan*	Mesin penyimpanan MySQL yang Anda gunakan bukan InnoDB, yang merupakan mesin penyimpanan MySQL yang direkomendasikan untuk Amazon RDS. Untuk mengetahui informasi tentang mesin penyimpanan MySQL, lihat Mesin penyimpanan yang didukung untuk RDS for MySQL.
pemberitahuan	RDS-EVENT-0055	*pesan*	Jumlah tabel yang Anda miliki untuk instans DB melebihi praktik terbaik yang direkomendasikan untuk Amazon RDS. Kurangi jumlah tabel pada instans DB Anda. Untuk mengetahui informasi tentang praktik terbaik yang direkomendasikan, lihat Pedoman operasional dasar Amazon RDS.

Kategori	ID peristiwa RDS	Pesan	Catatan
pemberitahuan	RDS-EVENT-0056	*pesan*	Jumlah basis data yang Anda miliki untuk instans DB melebihi praktik terbaik yang direkomendasikan untuk Amazon RDS. Kurangi jumlah basis data pada instans DB Anda. Untuk mengetahui informasi tentang praktik terbaik yang direkomendasikan, lihat Pedoman operasional dasar Amazon RDS.
pemberitahuan	RDS-EVENT-0064	Kunci enkripsi TDE berhasil dirotasi.	Untuk mengetahui informasi tentang praktik terbaik yang direkomendasikan, lihat Pedoman operasional dasar Amazon RDS.
pemberitahuan	RDS-EVENT-0084	Tidak dapat mengonversi instans DB ke Multi-AZ: *pesan*.	Anda mencoba mengonversi instans DB ke Multi-AZ, tetapi berisi grup file dalam memori yang tidak didukung untuk Multi-AZ. Untuk informasi selengkapnya, lihat Deployment Multi-AZ untuk Amazon RDS for Microsoft SQL Server.
pemberitahuan	RDS-EVENT-0087	Instans DB dihentikan.	

Kategori	ID peristiwa RDS	Pesan	Catatan
pemberitahuan	RDS-EVENT-0088	Instans DB dimulai.	
pemberitahuan	RDS-EVENT-0154	Instans DB sedang dimulai karena melebihi waktu maksimum penghentian yang diizinkan.	
pemberitahuan	RDS-EVENT-0157	Tidak dapat mengubah kelas instans DB. *pesan*.	RDS tidak dapat memodifikasi kelas instans DB karena kelas instans target tidak dapat mendukung jumlah basis data yang ada pada instans DB sumber. Pesan kesalahan muncul sebagai "Instans memiliki N basis data, tetapi setelah konversi, hanya akan mendukung N". Untuk informasi selengkapnya, lihat Batasan untuk instans DB Microsoft SQL Server.
pemberitahuan	RDS-EVENT-0158	Instans basis data berada dalam status yang tidak dapat ditingkatkan: *pesan*.	
pemberitahuan	RDS-EVENT-0167	*pesan*	Konfigurasi perimeter dukungan RDS Custom telah berubah.

Kategori	ID peristiwa RDS	Pesan	Catatan
pemberitahuan	RDS-EVENT-0189	Kredit saldo burst gp2 untuk instans basis data RDS rendah. Untuk mengatasi masalah ini, kurangi penggunaan IOPS atau ubah pengaturan penyimpanan Anda untuk mengaktifkan performa yang lebih tinggi.	Kredit saldo burst gp2 untuk instans basis data RDS rendah. Untuk mengatasi masalah ini, kurangi penggunaan IOPS atau ubah pengaturan penyimpanan Anda untuk mengaktifkan performa yang lebih tinggi. Untuk mengetahui informasi selengkapnya, lihat Kredit I/O dan performa burst di Panduan Pengguna Amazon Elastic Compute Cloud.
pemberitahuan	RDS-EVENT-0225	Ukuran penyimpanan yang dialokasikan *amount* GB mendekati ambang batas penyimpanan maksimum *amount* GB. Tingkatkan ambang penyimpanan maksimum.	Peristiwa ini diinvokasi saat penyimpanan yang dialokasikan mencapai 80% dari ambang batas penyimpanan maksimum. Untuk menghindari peristiwa ini, tingkatkan ambang batas penyimpanan maksimum.

Kategori	ID peristiwa RDS	Pesan	Catatan
pemberitahuan	RDS-EVENT-0231	Modifikasi penyimpanan instans DB Anda mengalami kesalahan internal. Permintaan modifikasi tertunda dan akan dicoba lagi nanti.	Terjadi kesalahan dalam proses replikasi baca. Untuk mengetahui informasi selengkapnya, lihat pesan peristiwa. Selain itu, lihat bagian pemecahan masalah untuk replika baca untuk mesin DB Anda. • Pemecahan Masalah kendala replika baca MariaDB • Pemecahan Masalah batasan replika baca SQL Server • Pemecahan Masalah batasan replika baca MySQL • Pemecahan masalah replika RDS for Oracle

Kategori	ID peristiwa RDS	Pesan	Catatan
pemberitahuan	RDS-EVENT-0253	Basis data menggunakan buffer doublewrite. *pesan*. Untuk mengetahui informasi selengkapnya, lihat dokumentasi RDS Optimized Writes untuk *name*.	RDS Optimized Writes tidak kompatibel dengan konfigurasi penyimpanan instans. Untuk mengetahui informasi selengkapnya, lihat Meningkatkan performa penulisan dengan RDS Optimized Writes for MySQL dan Meningkatkan performa penulisan dengan Amazon RDS Optimized Writes for MariaDB. Anda dapat melakukan peningkatan konfigurasi penyimpanan untuk mengaktifkan Optimized Writes dengan Membuat deployment blue/green.
pemberitahuan	RDS-EVENT-0297	Konfigurasi penyimpanan untuk instans DB *name* mendukung ukuran maksimum 16.384 GiB. Lakukan peningkatan konfigurasi penyimpanan untuk mendukung ukuran penyimpanan yang lebih besar dari 16384 GiB.	Anda tidak dapat meningkatkan ukuran penyimpanan instans DB yang dialokasikan melebihi 16384 GiB. Untuk mengatasi batasan ini, lakukan peningkatan konfigurasi penyimpanan. Untuk mengetahui informasi selengkapnya, lihat Meningkatkan Konfigurasi Penyimpanan.

Kategori	ID peristiwa RDS	Pesan	Catatan
pemberitahuan	RDS-EVENT-0298	Konfigurasi penyimpanan untuk instans DB *name* mendukung ukuran tabel maksimum 2048 GiB. Lakukan peningkatan konfigurasi penyimpanan untuk mendukung ukuran tabel yang lebih besar dari 2048 GiB.	Instans RDS MySQL dan MariaDB dengan batasan ini tidak dapat memiliki ukuran tabel melebihi 2048 GiB. Untuk mengatasi batasan ini, lakukan peningkatan konfigurasi penyimpanan. Untuk mengetahui informasi selengkapnya, lihat Meningkatkan Konfigurasi Penyimpanan.
pemberitahuan	RDS-EVENT-0327	Amazon RDS tidak dapat menemukan rahasia *RAHASIA ARN*. *pesan.*	
replika baca	RDS-EVENT-0045	Replikasi telah berhenti.	Replikasi pada instans DB Anda telah dihentikan karena penyimpanan tidak mencukupi. Skalakan penyimpanan atau kurangi ukuran maksimum log redo agar replikasi berlanjut. Untuk mengakomodasi log redo ukuran %d MiB Anda memerlukan setidaknya %d MiB penyimpanan kosong.

Kategori	ID peristiwa RDS	Pesan	Catatan
replika baca	RDS-EVENT-0046	Replikasi untuk Replika Baca dilanjutkan.	Pesan ini muncul saat Anda pertama kali membuat replika baca, atau sebagai pesan pemantauan yang mengonfirmasi bahwa replikasi berfungsi dengan benar. Jika pesan ini mengikuti pemberitahuan `RDS-EVENT-0045` , replikasi telah dilanjutkan setelah kesalahan atau setelah replikasi dihentikan.
replika baca	RDS-EVENT-0057	Streaming replikasi telah dihentikan.	
replika baca	RDS-EVENT-0062	Replikasi untuk Replika Baca telah dihentikan secara manual.	
replika baca	RDS-EVENT-0063	Replikasi dari instans Non RDS telah direset.	
replika baca	RDS-EVENT-0202	Pembuatan replika baca gagal.	
replika baca	RDS-ACARA-0357	*Nama* saluran replikasi dimulai.	Untuk informasi tentang saluran replikasi, lihatthe section called “Mengkonfigurasi replikasi multi-sumber”.

Kategori	ID peristiwa RDS	Pesan	Catatan
replika baca	RDS-ACAR-0358	*Nama* saluran replikasi berhenti.	Untuk informasi tentang saluran replikasi, lihatthe section called “Mengkonfigurasi replikasi multi-sumber”.
replika baca	RDS-ACARA-0359	*Nama* saluran replikasi dihentikan secara manual.	Untuk informasi tentang saluran replikasi, lihatthe section called “Mengkonfigurasi replikasi multi-sumber”.
replika baca	RDS-ACARA-0360	*Nama* saluran replikasi diatur ulang.	Untuk informasi tentang saluran replikasi, lihatthe section called “Mengkonfigurasi replikasi multi-sumber”.
pemulihan	RDS-EVENT-0020	Pemulihan instans DB telah dimulai. Waktu pemulihan beragam dengan jumlah data yang akan dipulihkan.	
pemulihan	RDS-EVENT-0021	Pemulihan instans DB selesai.	
pemulihan	RDS-EVENT-0023	Permintaan Snapshot Muncul: *pesan*.	Pencadangan manual telah diminta, tetapi Amazon RDS saat ini dalam proses pembuatan snapshot DB. Kirim permintaan lagi setelah Amazon RDS menyelesaikan snapshot DB.

Kategori	ID peristiwa RDS	Pesan	Catatan
pemulihan	RDS-EVENT-0052	Pemulihan instans Multi-AZ dimulai.	Waktu pemulihan beragam dengan jumlah data yang akan dipulihkan.
pemulihan	RDS-EVENT-0053	Pemulihan instans Multi-AZ selesai. Failover atau aktivasi tertunda.	
pemulihan	RDS-EVENT-0066	Instans akan diturunkan saat pencerminan dibuat kembali: *pesan*.	Instans DB SQL Server membangun ulang cerminnya. Performa akan diturunkan hingga cermin dibangun kembali. Basis data ditemukan dengan model pemulihan non-FULL. Model pemulihan diubah kembali ke FULL dan pemulihan mirroring dimulai. (<dbname>: <recovery model found>[,...])"
pemulihan	RDS-EVENT-0166	*pesan*	Instans DB RDS Custom ada dalam perimeter dukungan.
pemulihan	RDS-EVENT-0019	Dipulihkan dari instans DB *name* ke *name*.	Instans DB telah dipulihkan dari point-in-time cadangan.

Kategori	ID peristiwa RDS	Pesan	Catatan
keamanan	RDS-EVENT-0068	Mendekripsi kata sandi partisi hsm untuk memperbarui instans.	RDS mendekripsi kata sandi AWS CloudHSM partisi untuk membuat pembaruan pada instans DB. Untuk mengetahui informasi selengkapnya, lihat Enkripsi Data Transparan (TDE) Oracle Database dengan AWS CloudHSM di Panduan Pengguna AWS CloudHSM.
patching keamanan	RDS-EVENT-0230	Pembaruan sistem tersedia untuk instans DB Anda. Untuk mengetahui informasi tentang cara menerapkan pembaruan, lihat 'Mempertahankan instans DB' di Panduan Pengguna RDS.	Pembaruan Sistem Operasi baru tersedia. Pembaruan sistem operasi versi minor baru tersedia untuk instans DB Anda. Untuk mengetahui informasi tentang cara menerapkan pembaruan, lihat Bekerja dengan pembaruan sistem operasi.

Peristiwa grup parameter DB

Tabel berikut menunjukkan kategori peristiwa dan daftar peristiwa saat grup parameter DB merupakan jenis sumber.

Kategori	ID peristiwa RDS	Pesan	Catatan
perubahan konfigurasi	RDS-EVENT-0037	Memperbarui parameter *name* ke *value* dengan	

Kategori	ID peristiwa RDS	Pesan	Catatan
		metode penerapan *method*.	

Peristiwa grup keamanan DB

Tabel berikut menunjukkan kategori peristiwa dan daftar peristiwa saat grup keamanan DB merupakan jenis sumber.

> **Note**
>
> Grup keamanan DB merupakan sumber daya untuk EC2-Classic. EC2-Classic sudah tidak digunakan lagi pada 15 Agustus 2022. Jika Anda belum bermigrasi dari EC2-Classic ke VPC, sebaiknya Anda bermigrasi sesegera mungkin. Untuk mengetahui informasi selengkapnya, lihat Migrasi dari EC2-Classic ke VPC di Panduan Pengguna Amazon EC2 dan blog Jaringan EC2-Classic akan Segera Dihentikan – Berikut Cara Mempersiapkannya.

Kategori	ID peristiwa RDS	Pesan	Catatan
perubahan konfigurasi	RDS-EVENT-0038	Menerapkan perubahan ke grup keamanan.	
kegagalan	RDS-EVENT-0039	Mencabut otorisasi sebagai *user*.	Grup keamanan milik *user* tidak ada. Otorisasi untuk grup keamanan telah dicabut karena tidak valid.

Peristiwa snapshot DB

Tabel berikut menunjukkan kategori peristiwa dan daftar peristiwa saat snapshot DB merupakan jenis sumber.

Kategori	ID peristiwa RDS	Pesan	Catatan
pembuatan	RDS-EVENT-0040	Membuat snapshot manual.	
pembuatan	RDS-EVENT-0042	Snapshot manual dibuat.	
pembuatan	RDS-EVENT-0090	Membuat snapshot otomatis.	
pembuatan	RDS-EVENT-0091	Snapshot otomatis dibuat.	
penghapusan	RDS-EVENT-0041	Menghapus snapshot pengguna.	
pemberitahuan	RDS-EVENT-0059	Memulai salinan snapshot *name* dari wilayah *name*.	Ini adalah salinan snapshot lintas-Wilayah.
pemberitahuan	RDS-EVENT-0060	Selesai salinan snapshot *name* dari wilayah *name* dalam *number* menit.	Ini adalah salinan snapshot lintas-Wilayah.
pemberitahuan	RDS-EVENT-0061	Membatalkan permintaan salinan snapshot *name* dari wilayah *name*.	Ini adalah salinan snapshot lintas-Wilayah.
pemberitahuan	RDS-EVENT-0159	Tugas ekspor snapshot gagal.	
pemberitahuan	RDS-EVENT-0160	Tugas ekspor snapshot dibatalkan.	
pemberitahuan	RDS-EVENT-0161	Tugas ekspor snapshot selesai.	
pemberitahuan	RDS-EVENT-0196	Memulai salinan snapshot *name* di wilayah *name*.	Ini adalah salinan snapshot lokal.

Kategori	ID peristiwa RDS	Pesan	Catatan
pemberitahuan	RDS-EVENT-0197	Menyelesaikan salinan snapshot *name* di wilayah *name*.	Ini adalah salinan snapshot lokal.
pemberitahuan	RDS-EVENT-0190	Membatalkan permintaan salinan snapshot *name* di wilayah *name*.	Ini adalah salinan snapshot lokal.
pemulihan	RDS-EVENT-0043	Dipulihkan dari snapshot *name*.	Instans DB sedang dipulihkan dari snapshot DB.

Peristiwa snapshot klaster DB

Tabel berikut menunjukkan kategori peristiwa dan daftar peristiwa saat snapshot klaster DB merupakan jenis sumber.

Kategori	ID peristiwa RDS	Pesan	Catatan
pencadangan	RDS-EVENT-0074	Membuat snapshot klaster manual.	
pencadangan	RDS-EVENT-0075	Snapshot klaster manual dibuat.	
pencadangan	RDS-EVENT-0168	Membuat snapshot klaster otomatis.	
pencadangan	RDS-EVENT-0169	Snapshot klaster otomatis dibuat.	

Peristiwa Proksi RDS

Tabel berikut menunjukkan kategori peristiwa dan daftar peristiwa saat Proksi RDS berupa jenis sumber.

Kategori	ID peristiwa RDS	Pesan	Catatan
perubahan konfigurasi	RDS-EVENT-0204	RDS memodifikasi proksi DB *name*.	
perubahan konfigurasi	RDS-EVENT-0207	RDS memodifikasi titik akhir proksi DB *name*.	
perubahan konfigurasi	RDS-EVENT-0213	RDS mendeteksi penambahan instans DB dan secara otomatis menambahkannya ke grup target proksi DB *name*.	
perubahan konfigurasi	RDS-EVENT-0213	RDS mendeteksi pembuatan instans DB *name* dan secara otomatis menambahkannya ke grup target *name* proksi DB *name*.	
perubahan konfigurasi	RDS-EVENT-0214	RDS mendeteksi penghapusan instans DB *name* dan secara otomatis menghapusnya dari grup target *name* proksi DB *name*.	
perubahan konfigurasi	RDS-EVENT-0215	RDS mendeteksi penghapusan klaster DB *name* dan secara otomatis menghapusnya dari grup target *name* proksi DB *name*.	
pembuatan	RDS-EVENT-0203	RDS membuat proksi DB *name*.	

Kategori	ID peristiwa RDS	Pesan	Catatan
pembuatan	RDS-EVENT-0206	RDS membuat titik akhir *name* untuk proksi DB *name*.	
penghapusan	RDS-EVENT-0205	RDS menghapus proksi DB *name*.	
penghapusan	RDS-EVENT-0208	RDS menghapus titik akhir *name* untuk proksi DB *name*.	
kegagalan	RDS-EVENT-0243	RDS gagal menyediakan kapasitas untuk proxy *name* karena tidak ada alamat IP yang cukup yang tersedia di subnet Anda: *name*. Untuk memperbaiki masalah ini, pastikan subnet Anda memiliki jumlah minimum alamat IP yang tidak digunakan seperti yang direkomendasikan dalam dokumentasi Proksi RDS.	Untuk menentukan jumlah yang direkomendasikan untuk kelas instans Anda, lihat Perencanaan untuk kapasitas alamat IP.
kegagalan	RDS-EVENT-0275	*RDS membatasi beberapa koneksi ke nama proxy DB.* Jumlah permintaan koneksi simultan dari klien ke proxy telah melampaui batas.	

Peristiwa deployment blue/green

Tabel berikut menunjukkan kategori peristiwa dan daftar peristiwa saat deployment blue/green merupakan jenis sumber.

Untuk mengetahui informasi selengkapnya tentang deployment blue/green, lihat Menggunakan Deployment Blue/Green Amazon RDS untuk pembaruan basis data.

Kategori	ID peristiwa Amazon RDS	Pesan	Catatan
pembuatan	RDS-EVENT-0244	Tugas deployment blue/green selesai. Anda dapat membuat lebih banyak modifikasi pada basis data lingkungan hijau atau beralih antar deployment.	
kegagalan	RDS-EVENT-0245	Pembuatan deployment blue/green gagal karena (instans/klaster) DB (sumber/target) tidak ditemukan.	
penghapusan	RDS-EVENT-0246	Deployment blue/green dihapus.	
pemberitahuan	RDS-EVENT-0247	Switchover dari *blue* ke *green* dimulai.	
pemberitahuan	RDS-EVENT-0248	Switchover selesai pada deployment blue/green.	
kegagalan	RDS-EVENT-0249	Switchover dibatalkan pada deployment blue/green.	
pemberitahuan	RDS-EVENT-0250	Switchover dari replika primer/baca *blue* ke *green* dimulai.	

Kategori	ID peristiwa Amazon RDS	Pesan	Catatan
pemberitahuan	RDS-EVENT-0251	Switchover dari replika primer/baca *blue* ke *green* selesai. Mengganti nama *blue* menjadi *blue-old* dan *green* menjadi *blue*.	
kegagalan	RDS-EVENT-0252	Switchover dari replika primer/baca *blue* ke *green* dibatalkan dikarenakan *reason*.	
pemberitahuan	RDS-EVENT-0307	Sinkronisasi urutan untuk switchover *blue* ke *green* telah dimulai. Switchover saat menggunakan urutan dapat menyebabkan waktu henti yang diperpanjang.	
pemberitahuan	RDS-EVENT-0308	Sinkronisasi urutan untuk switchover *blue* ke *green* telah selesai.	
kegagalan	RDS-EVENT-0310	Sinkronisasi urutan untuk switchover *blue* ke *green* telah dibatalkan karena urutan gagal disinkronisasi.	

Peristiwa versi mesin kustom

Tabel berikut menunjukkan kategori peristiwa dan daftar peristiwa saat versi mesin kustom merupakan jenis sumber.

Kategori	ID peristiwa Amazon RDS	Pesan	Catatan
pembuatan	RDS-EVENT-0316	Bersiap untuk membuat versi mesin kustom *name*. Seluruh proses pembuatan dapat memakan waktu hingga empat jam untuk menyelesaikannya.	
pembuatan	RDS-EVENT-0317	Membuat versi mesin kustom *name*.	
pembuatan	RDS-EVENT-0318	Memvalidasi versi mesin kustom *name*.	
pembuatan	RDS-EVENT-0319	Versi mesin kustom *name* telah berhasil dibuat.	
pembuatan	RDS-EVENT-0320	RDS tidak dapat membuat versi mesin kustom *name* karena masalah internal. Kami menangani masalah ini dan akan menghubungi Anda jika perlu. Untuk bantuan lebih lanjut, hubungi AWS Premium Support/.	
kegagalan	RDS-EVENT-0198	Pembuatan gagal versi mesin kustom *name*. *pesan*	*pesan* ini mencakup detail tentang kegagalan, seperti file tidak ada.
kegagalan	RDS-EVENT-0277	Kegagalan selama penghapusan versi mesin kustom *name*. *pesan*	*Pesan* ini mencakup detail tentang kegagalan.

Kategori	ID peristiwa Amazon RDS	Pesan	Catatan
memulihkan	RDS-EVENT-0352	Jumlah basis data maksimum yang didukung untuk point-in-time pemulihan telah berubah.	*pesan* ini mencakup detail tentang peristiwa.

Memantau file log Amazon RDS

Setiap mesin basis RDS menghasilkan log yang dapat diakses untuk audit dan pemecahan masalah. Jenis log ini bergantung pada mesin basis data Anda.

Anda dapat mengakses log basis data menggunakan AWS Management Console, AWS Command Line Interface (AWS CLI), atau Amazon RDS API. Anda tidak dapat menampilkan, melihat, atau mengunduh log transaksi.

Topik

Melihat dan mencantumkan file log basis data

Anda dapat melihat file log basis data untuk mesin DB Amazon RDS Anda dengan menggunakan AWS Management Console. Anda dapat mencantumkan file log apa yang dapat diunduh atau dipantau dengan menggunakan AWS CLI atau Amazon RDS API.

> **Note**
>
> Jika Anda tidak dapat melihat daftar file log untuk intans DB RDS for Oracle yang sudah ada, reboot instans untuk melihat daftar tersebut.

Konsol

Untuk melihat file log basis data

1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data.
3. Pilih nama instans DB yang memiliki file log yang ingin dilihat.
4. Pilih tab Log & peristiwa.
5. Gulir ke bawah ke bagian Log.
6. (Opsional) Masukkan istilah pencarian untuk memfilter hasil.
7. Pilih log yang ingin dilihat, lalu pilih Lihat.

AWS CLI

Untuk mencantumkan file log basis data yang tersedia untuk instans DB, gunakan perintah AWS CLI `describe-db-log-files`.

Contoh berikut menampilkan daftar file log untuk instans DB yang bernama `my-db-instance`.

Example

```
aws rds describe-db-log-files --db-instance-identifier my-db-instance
```

RDS API

Untuk mencantumkan file log basis data yang tersedia untuk instans DB, gunakan tindakan `DescribeDBLogFiles` Amazon RDS API.

Mengunduh file log basis data

Anda dapat menggunakan AWS Management Console, AWS CLI atau API untuk mengunduh file log basis data.

Konsol

Untuk mengunduh file log basis data

1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data.

3. Pilih nama instans DB yang memiliki file log yang ingin dilihat.
4. Pilih tab Log & peristiwa.
5. Gulir ke bawah ke bagian Log.
6. Di bagian Log pilih tombol di samping log yang ingin diunduh, lalu pilih Unduh.
7. Buka menu konteks (klik kanan) untuk tautan yang diberikan, lalu pilih Simpan Tautan Sebagai. Masukkan lokasi tempat file log ingin disimpan, lalu pilih Simpan.

AWS CLI

Untuk mengunduh file log basis data, gunakan perintah AWS CLI `download-db-log-file-portion`. Secara default, perintah ini hanya mengunduh bagian terbaru dari file log. Namun, Anda dapat mengunduh seluruh file dengan menentukan parameter `--starting-token 0`.

Contoh berikut menunjukkan cara mengunduh seluruh konten file log yang disebut log/ERROR.4 dan menyimpannya di dalam file lokal yang disebut errorlog.txt.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds download-db-log-file-portion \
    --db-instance-identifier myexampledb \
    --starting-token 0 --output text \
    --log-file-name log/ERROR.4 > errorlog.txt
```

Untuk Windows:

```
aws rds download-db-log-file-portion ^
    --db-instance-identifier myexampledb ^
    --starting-token 0 --output text ^
    --log-file-name log/ERROR.4 > errorlog.txt
```

RDS API

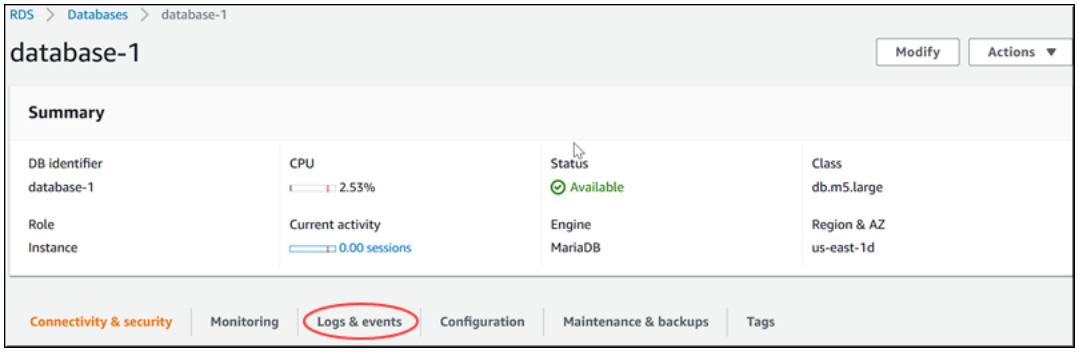
Untuk mengunduh file log basis data, gunakan tindakan `DownloadDBLogFilePortion` Amazon RDS API.

Melihat file log basis data

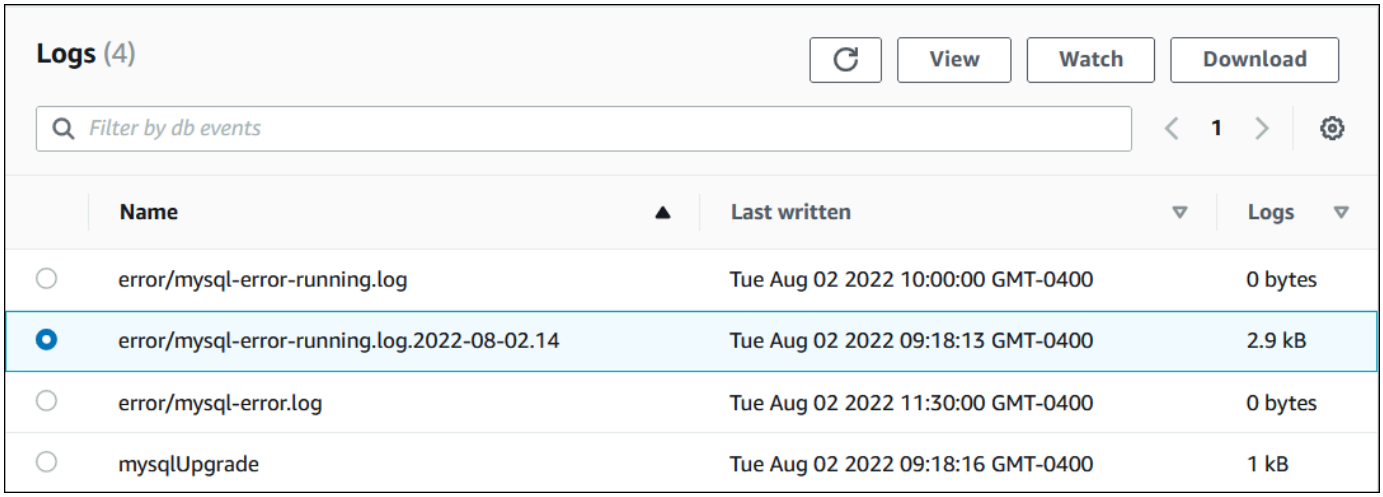
Melihat file log basis data sama seperti membuntuti file pada sistem UNIX atau Linux. Anda dapat melihat file log dengan menggunakan AWS Management Console. RDS menyegarkan ekor log setiap 5 detik.

Untuk melihat file log basis data

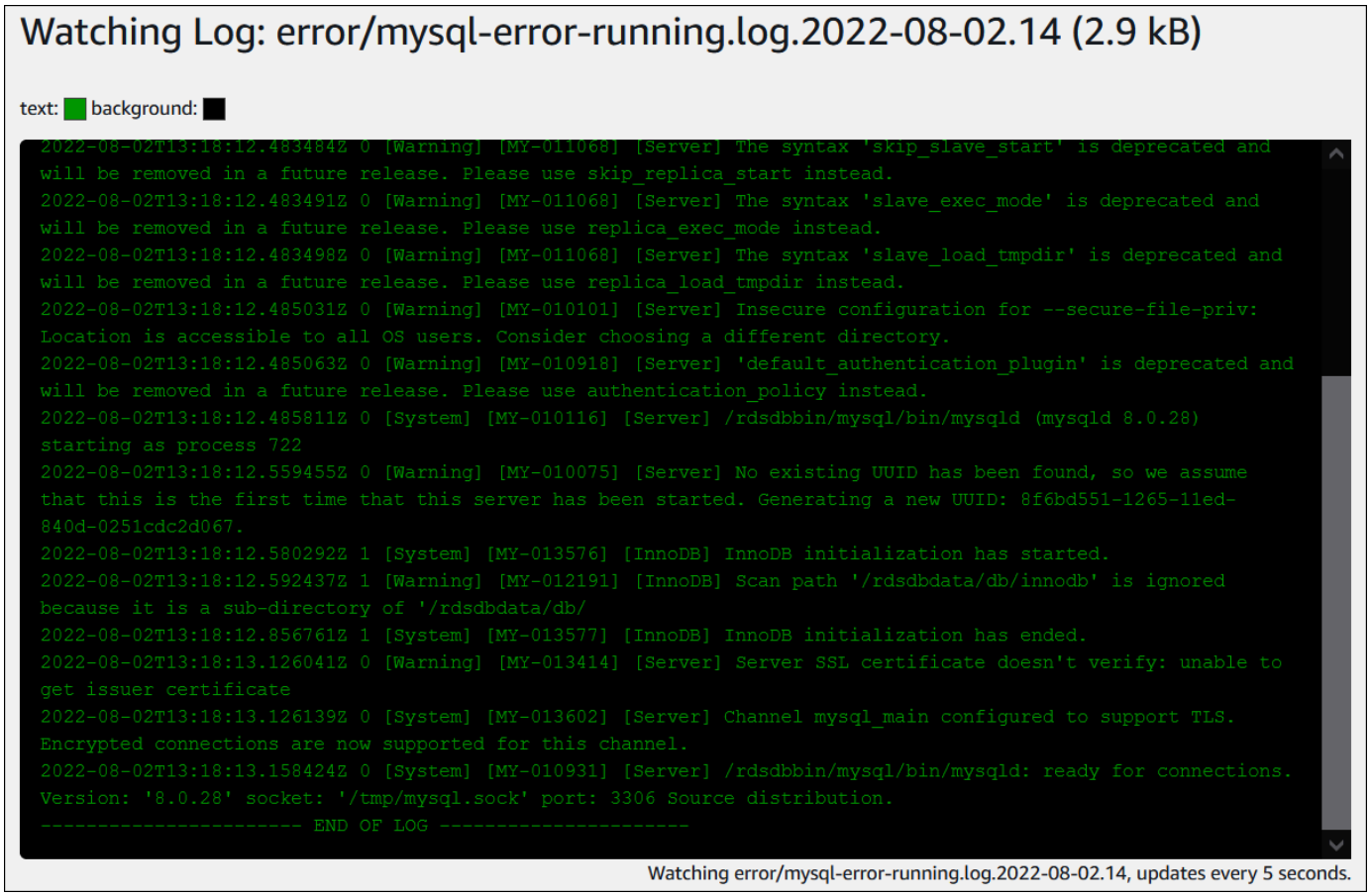
1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data.
3. Pilih nama instans DB yang memiliki file log yang ingin dilihat.
4. Pilih tab Log & peristiwa.



5. Di bagian Log, pilih file log, lalu pilih Lihat.



RDS menunjukkan ekor log, seperti pada contoh MySQL berikut.



Menerbitkan log basis data ke Log Amazon CloudWatch

Dalam basis data on-premise, log basis data berada pada sistem file. Amazon RDS tidak menyediakan akses host ke log basis data pada sistem file instans DB Anda. Untuk karena

itu, Amazon RDS mengizinkan Anda mengekspor log basis data ke Log Amazon CloudWatch. Dengan Log CloudWatch, Anda dapat melakukan analisis data log secara real-time. Anda dapat menyimpan data dalam penyimpanan yang sangat tahan lama dan mengelola data dengan Agen Log CloudWatch.

Topik

- Ringkasan integrasi RDS dengan Log CloudWatch
- Memutuskan log yang akan diterbitkan ke Log CloudWatch
- Menentukan log yang akan diterbitkan ke Log CloudWatch
- Mencari dan memfilter log Anda di Log CloudWatch

Ringkasan integrasi RDS dengan Log CloudWatch

Di Log CloudWatch, log stream adalah urutan log peristiwa yang berbagi sumber yang sama. Setiap sumber log terpisah di Log CloudWatch membentuk log stream terpisah. Grup log adalah grup log stream yang berbagi pengaturan retensi, pemantauan, dan kontrol akses yang sama.

Amazon RDS terus mengalirkan data log instans ke grup log. Misalnya, Anda memiliki grup log `/aws/rds/instance/`*`instance_name`*`/`*`log_type`* untuk setiap jenis log yang Anda terbitkan. Grup log ini berada di Wilayah AWS yang sama dengan instans basis data yang menghasilkan log.

AWS mempertahankan data log yang diterbitkan ke Log CloudWatch selama untuk periode waktu yang tidak ditentukan kecuali Anda menentukan periode retensi. Untuk mengetahui informasi selengkapnya, lihat Mengubah retensi data log di Log CloudWatch.

Memutuskan log yang akan diterbitkan ke Log CloudWatch

Setiap mesin basis data RDS mendukung kumpulan log-nya sendiri. Untuk mempelajari opsi untuk mesin basis data Anda, baca topik berikut:

- the section called “Menerbitkan log MariaDB ke Log Amazon CloudWatch ”
- the section called “Menerbitkan log MySQL ke Amazon Logs CloudWatch ”
- the section called “Menerbitkan log Oracle ke Amazon CloudWatch Logs”
- the section called “Menerbitkan log PostgreSQL ke Amazon Logs CloudWatch ”
- the section called “Menerbitkan log SQL Server ke Amazon CloudWatch Logs”

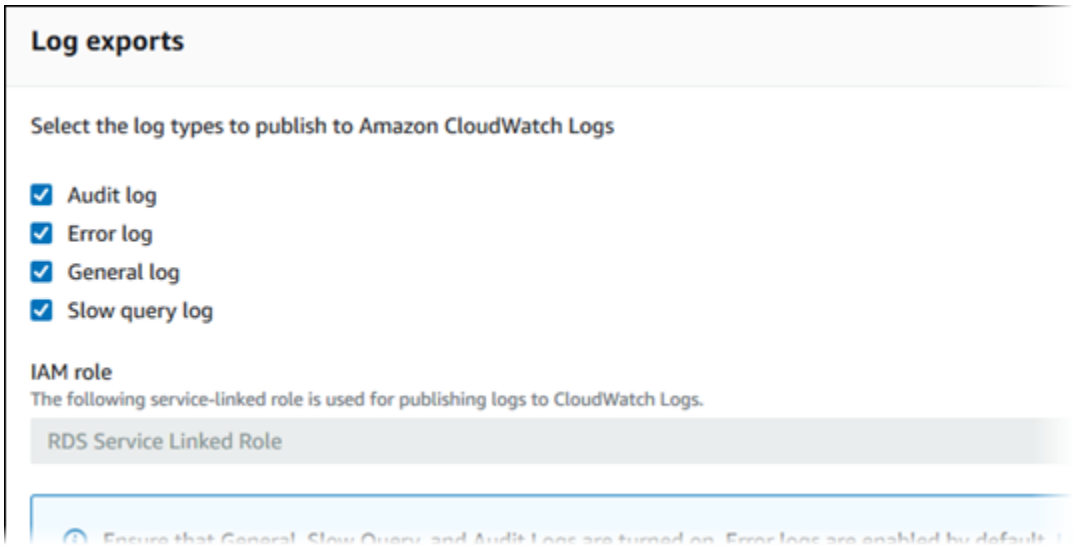
Menentukan log yang akan diterbitkan ke Log CloudWatch

Tentukan log yang akan diterbitkan di konsol. Pastikan Anda memiliki peran terkait layanan di AWS Identity and Access Management (IAM). Untuk mengetahui informasi selengkapnya tentang peran terkait layanan, lihat Menggunakan peran terkait layanan untuk Amazon RDS.

Untuk menentukan log yang akan diterbitkan

1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data.
3. Lakukan salah satu dari langkah berikut:
   - Pilih Buat basis data.
   - Pilih basis data dari daftar, lalu pilih Ubah.
4. Di Ekspor log, pilih log yang akan diterbitkan.

   Contoh berikut menentukan log audit, log kesalahan, log umum, dan log kueri lambat.



Mencari dan memfilter log Anda di Log CloudWatch

Anda dapat mencari entri log yang memenuhi kriteria tertentu menggunakan konsol Log CloudWatch. Anda dapat mengakses log baik melalui konsol RDS, yang mengarahkan Anda ke konsol Log CloudWatch, atau dari konsol Log CloudWatch secara langsung.

Untuk mencari log RDS menggunakan konsol RDS

1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data.
3. Pilih instans DB.
4. Pilih Konfigurasi.
5. Di bagian bawah Log yang diterbitkan, pilih log basis data yang ingin dilihat.

Untuk mencari log RDS menggunakan konsol Log CloudWatch

1. Buka konsol CloudWatch di https://console.aws.amazon.com/cloudwatch/.
2. Pada panel navigasi, pilih Grup log.
3. Di kotak filter, masukkan **`/aws/rds`**.
4. Untuk Grup Log, pilih nama grup log yang berisi log stream yang akan dicari.
5. Untuk Log Stream, pilih nama log stream yang akan dicari.
6. Di bagian Peristiwa log, masukkan sintaksis filter yang akan digunakan.

Untuk mengetahui informasi selengkapnya, lihat Mencari dan memfilter data log di Panduan Pengguna Log Amazon CloudWatch. Untuk tutorial yang menjelaskan cara memantau log RDS, lihat Membangun pemantauan basis data proaktif untuk Amazon RDS dengan Log Amazon CloudWatch, AWS Lambda, dan Amazon SNS.

Membaca isi file log dengan menggunakan REST

Amazon RDS menyediakan titik akhir REST yang memungkinkan akses ke file log instans basis data. Ini berguna jika Anda perlu menulis sebuah aplikasi untuk mengalirkan isi file log Amazon RDS.

Sintaksnya adalah:

```
GET /v13/downloadCompleteLogFile/DBInstanceIdentifier/LogFileName HTTP/1.1
Content-type: application/json
host: rds.region.amazonaws.com
```

Parameter-parameter berikut diperlukan:

- *`DBInstanceIdentifier`*—nama instans basis data yang berisi file log yang ingin Anda unduh.

- *LogFileName*—nama file log yang akan diunduh.

Respons akan mengandung isi file log yang diminta, berupa sebuah aliran.

Contoh berikut mengunduh file log dengan nama log/ERROR.6 untuk instans basis data yang bernama sampel-sql di kawasan us-west-2.

```
GET /v13/downloadCompleteLogFile/sample-sql/log/ERROR.6 HTTP/1.1
host: rds.us-west-2.amazonaws.com
X-Amz-Security-Token: AQoDYXdzEIH//////////
wEa0AIXLhngC5zp9CyB1R6abwKrXHVR5efnAVN3XvR7IwqKYalFSn6UyJuEFTft9nObglx4QJ+GXV9cpACkETq=
X-Amz-Date: 20140903T233749Z
X-Amz-Algorithm: AWS4-HMAC-SHA256
X-Amz-Credential: AKIADQKE4SARGYLE/20140903/us-west-2/rds/aws4_request
X-Amz-SignedHeaders: host
X-Amz-Content-SHA256: e3b0c44298fc1c229afbf4c8996fb92427ae41e4649b934de495991b7852b855
X-Amz-Expires: 86400
X-Amz-Signature: 353a4f14b3f250142d9afc34f9f9948154d46ce7d4ec091d0cdabbcf8b40c558
```

Jika Anda menentukan suatu instans basis data yang tidak ada, respons akan terdiri atas kesalahan berikut:

- `DBInstanceNotFound`—*DBInstanceIdentifier* tidak mengacu ke instans basis data yang ada. (Kode status HTTP: 404)

File log basis data MariaDB

Anda dapat memantau log kesalahan, log kueri lambat, dan log umum MariaDB. Log kesalahan MariaDB dihasilkan secara default; Anda dapat membuat log kueri lambat dan log umum dengan mengatur parameter di grup parameter DB. Amazon RDS merotasi semua file log MariaDB; interval untuk setiap jenis ditentukan sebagai berikut.

Anda dapat memantau log MariaDB secara langsung melalui konsol Amazon RDS, Amazon RDS API, Amazon RDS CLI, atau SDK. AWS Anda juga dapat mengakses log MariaDB dengan mengarahkan log ke tabel basis data di basis data utama dan mengkueri tabel tersebut. Anda dapat menggunakan utilitas mysqlbinlog untuk mengunduh log biner.

Untuk informasi selengkapnya tentang melihat, mengunduh, dan melihat log basis data berbasis file, lihat Memantau file log Amazon RDS.

Topik

Mengakses log kesalahan MariaDB

Log kesalahan MariaDB ditulis ke file `<host-name>.err` Anda. Anda dapat melihat file ini dengan menggunakan konsol Amazon RDS, Anda juga dapat mengambil log menggunakan Amazon RDS API, Amazon RDS CLI, atau SDK. AWS File `<host-name>.err` di-flush setiap 5 menit, dan kontennya ditambahkan ke `mysql-error-running.log`. File `mysql-error-running.log` lalu dirotasi setiap jam dan file per jam yang dihasilkan selama 24 jam terakhir dipertahankan. Setiap file log memiliki jam pembuatan (dalam UTC) yang ditambahkan pada namanya. File log juga memiliki stempel waktu yang membantu Anda menentukan kapan entri log ditulis.

MariaDB menulis ke log kesalahan hanya pada saat dinyalakan, dimatikan, dan saat mengalami kesalahan. Instans DB dapat memakan waktu berjam-jam atau berhari-hari tanpa perlu menulis

entri baru ke log kesalahan. Jika Anda tidak melihat entri terbaru, berarti server tidak mengalami kesalahan yang mengakibatkan entri log.

Mengakses log umum dan kueri lambat MariaDB

Anda dapat menulis log umum dan log kueri lambat MariaDB ke file atau tabel basis data dengan mengatur parameter di grup parameter DB. Untuk informasi tentang pembuatan dan modifikasi grup parameter DB, lihat Bekerja dengan grup parameter. Anda harus mengatur parameter ini sebelum dapat melihat log kueri lambat atau log umum di konsol Amazon RDS atau dengan menggunakan Amazon RDS API, AWS CLI, atau AWS SDK.

Anda dapat mengontrol pengelogan MariaDB dengan menggunakan parameter dalam daftar ini:

- `slow_query_log`atau`log_slow_query`: Untuk membuat log kueri lambat, atur ke 1. Default-nya adalah 0.
- `general_log`: Untuk membuat log umum, atur ke 1. Default-nya adalah 0.
- `long_query_time`atau`log_slow_query_time`: Untuk mencegah kueri yang berjalan cepat agar tidak masuk ke log kueri lambat, tentukan nilai untuk waktu proses kueri terpendek yang akan dicatat, dalam hitungan detik. Nilai default-nya adalah 10 detik; nilai minimumnya adalah 0. Jika log_output = FILE, Anda dapat menentukan nilai titik mengambang yang masuk ke resolusi mikrodetik. Jika log_output = TABLE, Anda harus menentukan nilai integer dengan resolusi kedua. Hanya kueri yang waktu jalannya melebihi `log_slow_query_time` nilai `long_query_time` atau yang dicatat. Misalnya, pengaturan `long_query_time` atau `log_slow_query_time` ke 0,1 mencegah kueri apa pun yang berjalan kurang dari 100 milidetik untuk dicatat.
- `log_queries_not_using_indexes`: Untuk mencatat semua kueri yang tidak menggunakan indeks ke log kueri lambat, atur parameter ini ke 1. Nilai default-nya adalah 0. Kueri yang tidak menggunakan indeks dicatat meskipun runtime-nya kurang dari nilai parameter `long_query_time`.
- `log_output` *`option`*: Anda dapat menentukan salah satu opsi berikut untuk parameter `log_output`:
  - TABLE (default)– Menulis kueri umum ke tabel `mysql.general_log`, dan kueri lambat ke tabel `mysql.slow_log`.
  - FILE– Menulis log umum dan log kueri lambat ke sistem file. File log dirotasi setiap jam.
  - NONE– Menonaktifkan pengelogan.

Saat pengelogan diaktifkan, Amazon RDS merotasi log tabel atau menghapus file log secara berkala. Langkah ini merupakan tindakan pencegahan untuk mengurangi kemungkinan file log besar memblokir penggunaan basis data atau memengaruhi performa. Rotasi dan penghapusan pendekatan pengelogan `FILE` dan `TABLE` sebagai berikut:

- Saat pengelogan `FILE` diaktifkan, file log akan diperiksa setiap jam dan file log yang lebih lama dari 24 jam akan dihapus. Dalam beberapa kasus, ukuran file log gabungan yang tersisa setelah penghapusan mungkin melebihi ambang batas 2 persen dari ruang yang dialokasikan oleh instans DB. Dalam kasus ini, file log paling besar akan dihapus hingga ukuran file log tidak lagi melebihi ambang batas.
- Saat pengelogan `TABLE` diaktifkan, dalam beberapa kasus tabel log dirotasi setiap 24 jam. Rotasi ini terjadi jika ruang yang digunakan oleh log tabel lebih dari 20 persen dari ruang penyimpanan yang dialokasikan. Ini juga terjadi jika ukuran semua log yang digabungkan lebih besar dari 10 GB. Jika jumlah ruang yang digunakan untuk instans DB lebih besar dari 90 persen dari ruang penyimpanan yang dialokasikan untuk instans DB, ambang untuk rotasi log akan berkurang. Tabel log ini kemudian dirotasi jika ruang yang digunakan oleh log tabel lebih dari 10 persen dari ruang penyimpanan yang dialokasikan. Tabel ini juga dirotasi jika ukuran semua log yang digabungkan lebih besar dari 5 GB.

  Saat tabel log dirotasi, tabel log saat ini disalin ke tabel log cadangan dan entri di tabel log saat ini dihapus. Jika sudah ada, tabel log cadangan akan dihapus sebelum tabel log saat ini disalin ke cadangan. Anda dapat meminta tabel log cadangan jika diperlukan. Tabel log cadangan untuk tabel `mysql.general_log` bernama `mysql.general_log_backup`. Tabel log cadangan untuk tabel `mysql.slow_log` bernama `mysql.slow_log_backup`.

  Anda dapat merotasi tabel `mysql.general_log` dengan mengikuti prosedur `mysql.rds_rotate_general_log`. Anda dapat merotasi `mysql.slow_log` tabel dengan mengikuti `mysql.rds_rotate_slow_log` prosedur.

  Log tabel dirotasi selama upgrade versi basis data.

Amazon RDS mencatat rotasi log `TABLE` dan `FILE` dalam peristiwa Amazon RDS dan mengirimkan pemberitahuan kepada Anda.

Untuk bekerja dengan log dari konsol Amazon RDS, Amazon RDS API, Amazon RDS CLI, AWS atau SDK, setel parameter ke FILE. `log_output` Seperti log kesalahan MariaDB, file log ini dirotasi setiap jam. File log yang dihasilkan selama 24 jam sebelumnya akan dipertahankan.

Untuk informasi selengkapnya tentang kueri lambat dan log umum, buka topik berikut di dokumentasi MariaDB:

- Log kueri lambat
- Log kueri umum

Menerbitkan log MariaDB ke Log Amazon CloudWatch

Anda dapat mengonfigurasi instans MariaDB Anda untuk mempublikasikan data log ke grup log di Amazon Logs. CloudWatch Dengan CloudWatch Log, 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 sangat tahan lama.

Amazon RDS menerbitkan masing-masing log basis data MariaDB sebagai aliran basis data terpisah di grup log. Misalnya, misalkan Anda mengonfigurasi fungsi ekspor untuk menyertakan log kueri lambat. Kemudian data kueri lambat disimpan dalam log stream kueri lambat di grup log `/aws/rds/instance/my_instance/slowquery`.

Log kesalahan diaktifkan secara default. Tabel berikut merangkum persyaratan untuk log MariaDB lainnya.

Log	Persyaratan
Log audit	Instans DB harus menggunakan grup opsi kustom dengan opsi `MARIADB_AUDIT_PLUGIN` .
Log umum	Instans DB harus menggunakan grup parameter kustom dengan pengaturan parameter `general_log = 1` untuk mengaktifkan log umum.
Log kueri lambat	Instans DB harus menggunakan grup parameter khusus dengan pengaturan parameter `slow_query_log = 1` atau `log_slow_query = 1` untuk mengaktifkan log kueri lambat.

Log	Persyaratan
Output log	Instans DB harus menggunakan grup parameter khusus dengan pengaturan parameter `log_output = FILE` untuk menulis log ke sistem file dan mempublik asikannya ke CloudWatch Log.

Konsol

Untuk memublikasikan log MariaDB CloudWatch ke Log dari konsol

1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data, lalu pilih instans DB yang ingin diubah.
3. Pilih Ubah.
4. Di bagian Log ekspor, pilih log yang ingin Anda mulai terbitkan ke CloudWatch Log.
5. Pilih Lanjutkan, lalu pilih Ubah Instans DB di halaman ringkasan.

AWS CLI

Anda dapat mempublikasikan log MariaDB dengan file. AWS CLI Anda dapat memanggil perintah `modify-db-instance` dengan parameter berikut:

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

> **Note**
>
> Perubahan pada opsi `--cloudwatch-logs-export-configuration` selalu diterapkan ke instans DB secara langsung. Oleh karena itu, opsi `--apply-immediately` dan `--no-apply-immediately` tidak akan berpengaruh.

Anda juga dapat mempublikasikan log MariaDB dengan memanggil perintah berikut: AWS CLI

- `create-db-instance`

- restore-db-instance-from-db-snapshot
- restore-db-instance-from-s3
- restore-db-instance-to-point-in-time

Jalankan salah satu AWS CLI perintah ini dengan opsi berikut:

- `--db-instance-identifier`
- `--enable-cloudwatch-logs-exports`
- `--db-instance-class`
- `--engine`

Opsi lain mungkin diperlukan tergantung pada AWS CLI perintah yang Anda jalankan.

Example

Contoh berikut memodifikasi instance MariaDB yang ada untuk mempublikasikan file log ke Log. CloudWatch Nilai `--cloudwatch-logs-export-configuration` adalah objek JSON. Kunci untuk objek ini adalah `EnableLogTypes`, dan nilainya adalah serangkaian string dengan setiap kombinasi `audit`, `error`, `general`, dan `slowquery`.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":
["audit","error","general","slowquery"]}'
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":
["audit","error","general","slowquery"]}'
```

Example

Perintah berikut membuat instance MariaDB dan menerbitkan file log ke Log. CloudWatch Nilai `--enable-cloudwatch-logs-exports` adalah rangkaian string JSON. String dapat berupa kombinasi `audit`, `error`, `general`, dan `slowquery`.

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --enable-cloudwatch-logs-exports '["audit","error","general","slowquery"]' \
    --db-instance-class db.m4.large \
    --engine mariadb
```

Untuk Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --enable-cloudwatch-logs-exports '["audit","error","general","slowquery"]' ^
    --db-instance-class db.m4.large ^
    --engine mariadb
```

RDS API

Anda dapat menerbitkan log MariaDB dengan RDS API. Anda dapat memanggil operasi `ModifyDBInstance` dengan parameter berikut:

- `DBInstanceIdentifier`
- `CloudwatchLogsExportConfiguration`

> **Note**
>
> Perubahan pada parameter `CloudwatchLogsExportConfiguration` selalu diterapkan ke instans DB secara langsung. Oleh karena itu, parameter `ApplyImmediately` tidak memiliki dampak.

Anda juga dapat menerbitkan log MariaDB dengan memanggil operasi RDS API berikut:

- `CreateDBInstance`
- `RestoreDBInstanceFromDBSnapshot`
- `RestoreDBInstanceFromS3`
- `RestoreDBInstanceToPointInTime`

Jalankan salah satu operasi RDS API ini dengan parameter berikut:

- `DBInstanceIdentifier`
- `EnableCloudwatchLogsExports`
- `Engine`
- `DBInstanceClass`

Parameter lain mungkin diperlukan tergantung pada AWS CLI perintah yang Anda jalankan.

Ukuran file log

Ukuran file log umum, log kesalahan, dan log kueri lambat MariaDB dibatasi hingga tidak lebih dari 2 persen dari ruang penyimpanan yang dialokasikan untuk instans DB. Untuk mempertahankan ambang batas ini, log secara otomatis dirotasi setiap jam dan file log yang lebih lama dari 24 jam dihapus. Jika ukuran file log gabungan melebihi ambang batas setelah menghapus file log lama, file log paling besar akan dihapus hingga ukuran file log tidak lagi melebihi ambang batas.

Mengelola log MariaDB berbasis tabel

Anda dapat mengarahkan log kueri lambat dan log umum ke tabel di instans DB. Untuk melakukannya, buat grup parameter DB dan atur parameter server `log_output` ke `TABLE`. Kueri umum lalu dicatat ke tabel `mysql.general_log`, dan kueri lambat dicatat ke tabel `mysql.slow_log`. Anda dapat mengueri tabel untuk mengakses informasi log. Mengaktifkan pencatatan ini akan meningkatkan jumlah data yang akan ditulis ke basis data. Hal ini dapat menurunkan performa.

Log umum dan log kueri lambat dinonaktifkan secara default. Untuk mengaktifkan pencatatan ke tabel, Anda juga harus mengatur parameter server berikut ke`1`:

- `general_log`
- `slow_query_log` atau `log_slow_query`

Tabel log terus bertambah hingga aktivitas pencatatan terkait dinonaktifkan dengan mengatur ulang parameter yang sesuai ke `0`. Banyak data yang sering terakumulasi seiring berjalannya waktu, yang dapat menghabiskan banyak persentase ruang penyimpanan yang dialokasikan. Amazon RDS tidak mengizinkan Anda memotong tabel log, tetapi Anda dapat memindahkan kontennya. Merotasi tabel akan menyimpan kontennya ke tabel cadangan, lalu membuat tabel log kosong yang baru. Anda dapat merotasi tabel log secara manual dengan mengikuti prosedur perintah berikut, dengan permintaan perintah ditunjukkan oleh `PROMPT>`:

```
PROMPT> CALL mysql.rds_rotate_slow_log;
PROMPT> CALL mysql.rds_rotate_general_log;
```

Untuk menghapus data lama sepenuhnya dan mengosongkan kembali ruang disk, lakukan prosedur yang sesuai dua kali secara berurutan.

Format pengelogan biner

MariaDB di Amazon RDS mendukung format pengelogan biner berbasis baris, berbasis pernyataan, dan campuran. Format pengelogan biner default adalah campuran. Untuk detail tentang format log biner MariaDB lainnya, lihat Format log biner dalam dokumentasi MariaDB.

Jika Anda berencana menggunakan replikasi, format pengelogan biner itu penting. Hal ini karena menentukan catatan perubahan data yang dicatat di sumber dan dikirim ke target replikasi. Untuk informasi tentang kelebihan dan kelemahan format pengelogan biner lainnya untuk replikasi, lihat Kelebihan dan kelemahan replikasi berbasis pernyataan dan berbasis baris dalam dokumentasi MySQL.

> **Important**
>
> Mengatur format pengelogan biner ke berbasis baris dapat menghasilkan file log biner yang sangat besar. File log biner besar mengurangi jumlah penyimpanan yang tersedia untuk instans DB. File ini juga dapat meningkatkan jumlah waktu untuk melakukan operasi pemulihan instans DB.
>
> Replikasi berbasis pernyataan dapat menyebabkan inkonsistensi antara instans DB sumber dan replika baca. Untuk informasi selengkapnya, lihat Pernyataan yang tidak aman untuk replikasi berbasis pernyataan dalam dokumentasi MariaDB.

Untuk mengatur format pengelogan biner MariaDB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup parameter.
3. Pilih grup parameter yang digunakan oleh instans DB yang ingin diubah.

   Anda tidak dapat mengubah grup parameter default. Jika instans DB menggunakan grup parameter default, buat grup parameter baru dan hubungkan dengan instans DB.

   Untuk informasi selengkapnya tentang grup parameter DB, lihat Bekerja dengan grup parameter.
4. Untuk Tindakan grup parameter, pilih Edit.
5. Atur parameter `binlog_format` ke format pengelogan biner pilihan Anda (ROW, STATEMENT, atau MIXED).
6. Pilih Simpan perubahan untuk menyimpan pembaruan ke grup parameter DB.

Mengakses log biner MariaDB

Anda dapat menggunakan utilitas mysqlbinlog untuk mengunduh log biner dalam format teks dari instans DB MariaDB. Log biner diunduh ke komputer lokal Anda. Untuk informasi selengkapnya tentang penggunaan utilitas mysqlbinlog, buka Menggunakan mysqlbinlog dalam dokumentasi MariaDB.

Untuk menjalankan utilitas mysqlbinlog terhadap instans Amazon RDS, gunakan opsi berikut:

- Tentukan opsi `--read-from-remote-server`.
- `--host`: Tentukan nama DNS dari titik akhir instans.
- `--port`: Tentukan port yang digunakan oleh instans.
- `--user`: Tentukan pengguna MariaDB yang telah diberikan izin slave replikasi.
- `--password`: Tentukan kata sandi untuk pengguna, atau menghapus nilai kata sandi agar utilitas meminta Anda memasukkan kata sandi.
- `--result-file`: Tentukan file lokal yang menerima output.
- Tentukan nama satu atau beberapa file log biner. Untuk mendapatkan daftar log yang tersedia, gunakan perintah SQL SHOW BINARY LOGS.

Untuk informasi selengkapnya tentang opsi mysqlbinlog, buka opsi mysqlbinlog dalam dokumentasi MariaDB.

Berikut ini adalah contohnya:

Untuk Linux, macOS, atau Unix:

```
mysqlbinlog \
    --read-from-remote-server \
    --host=mariadbinstance1.1234abcd.region.rds.amazonaws.com \
    --port=3306  \
    --user ReplUser \
    --password <password> \
    --result-file=/tmp/binlog.txt
```

Untuk Windows:

```
mysqlbinlog ^
    --read-from-remote-server ^
    --host=mariadbinstance1.1234abcd.region.rds.amazonaws.com ^
    --port=3306  ^
    --user ReplUser ^
    --password <password> ^
    --result-file=/tmp/binlog.txt
```

Amazon RDS biasanya membersihkan log biner sesegera mungkin. Namun, log biner harus tetap tersedia di instans untuk diakses oleh mysqlbinlog. Untuk menentukan jumlah jam bagi RDS untuk mempertahankan log biner, gunakan prosedur tersimpan `mysql.rds_set_configuration`. Tentukan periode dengan waktu yang cukup bagi Anda untuk mengunduh log. Setelah Anda mengatur periode retensi, pantau penggunaan penyimpanan untuk instans DB guna memastikan bahwa log biner yang dipertahankan tidak memakan terlalu banyak ruang penyimpanan.

Contoh berikut menetapkan periode retensi ke 1 hari.

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

Untuk menampilkan pengaturan saat ini, gunakan prosedur tersimpan `mysql.rds_show_configuration`.

```
call mysql.rds_show_configuration;
```

Anotasi log biner

Di instans DB MariaDB, Anda dapat menggunakan peristiwa `Annotate_rows` untuk menganotasi peristiwa baris dengan salinan kueri SQL yang menyebabkan peristiwa baris. Pendekatan ini memberikan fungsionalitas serupa untuk mengaktifkan parameter `binlog_rows_query_log_events` di instans DB RDS for MySQL.

Anda dapat mengaktifkan anotasi log biner secara global dengan membuat grup parameter kustom dan mengatur parameter `binlog_annotate_row_events` ke **1**. Anda juga dapat mengaktifkan anotasi pada tingkat sesi, dengan memanggil `SET SESSION binlog_annotate_row_events = 1`. Gunakan `replicate_annotate_row_events` untuk mereplikasi anotasi log biner instans replika jika pengelogan biner diaktifkan pada instans tersebut. Tidak ada hak istimewa khusus yang diperlukan untuk menggunakan pengaturan ini.

Berikut ini adalah contoh dari transaksi berbasis baris di MariaDB. Penggunaan pengelogan berbasis baris dipicu dengan mengatur tingkat isolasi transaksi ke read-committed.

```
CREATE DATABASE IF NOT EXISTS test;
USE test;
CREATE TABLE square(x INT PRIMARY KEY, y INT NOT NULL) ENGINE = InnoDB;
SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED;
BEGIN
INSERT INTO square(x, y) VALUES(5, 5 * 5);
COMMIT;
```

Tanpa anotasi, entri log biner untuk transaksi terlihat seperti berikut:

```
BEGIN
/*!*/;
# at 1163
# at 1209
#150922  7:55:57 server id 1855786460  end_log_pos 1209         Table_map:
 `test`.`square` mapped to number 76
#150922  7:55:57 server id 1855786460  end_log_pos 1247         Write_rows: table id 76
 flags: STMT_END_F
### INSERT INTO `test`.`square`
### SET
###   @1=5
###   @2=25
# at 1247
#150922  7:56:01 server id 1855786460  end_log_pos 1274         Xid = 62
```

```
COMMIT/*!*/;
```

Pernyataan berikut memungkinkan anotasi tingkat sesi untuk transaksi yang sama ini, dan menonaktifkannya setelah melakukan transaksi:

```
CREATE DATABASE IF NOT EXISTS test;
USE test;
CREATE TABLE square(x INT PRIMARY KEY, y INT NOT NULL) ENGINE = InnoDB;
SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED;
SET SESSION binlog_annotate_row_events = 1;
BEGIN;
INSERT INTO square(x, y) VALUES(5, 5 * 5);
COMMIT;
SET SESSION binlog_annotate_row_events = 0;
```

Dengan anotasi, entri log biner untuk transaksi terlihat seperti berikut:

```
BEGIN
/*!*/;
# at 423
# at 483
# at 529
#150922  8:04:24 server id 1855786460  end_log_pos 483  Annotate_rows:
#Q> INSERT INTO square(x, y) VALUES(5, 5 * 5)
#150922  8:04:24 server id 1855786460  end_log_pos 529  Table_map: `test`.`square`
 mapped to number 76
#150922  8:04:24 server id 1855786460  end_log_pos 567  Write_rows: table id 76 flags:
 STMT_END_F
### INSERT INTO `test`.`square`
### SET
###   @1=5
###   @2=25
# at 567
#150922  8:04:26 server id 1855786460  end_log_pos 594  Xid = 88
COMMIT/*!*/;
```

File log basis data Microsoft SQL Server

Anda dapat mengakses log kesalahan, log agen, file jejak, dan file dump Microsoft SQL Server dengan menggunakan konsol Amazon RDS, AWS CLI, atau RDS API. Untuk informasi selengkapnya tentang melihat, mengunduh, dan melihat log basis data berbasis file, lihat Memantau file log Amazon RDS.

Topik

Jadwal retensi

File log dirotasi setiap hari dan setiap kali instans DB Anda dimulai ulang. Berikut ini adalah jadwal retensi untuk log Microsoft SQL Server di Amazon RDS.

Jenis log	Jadwal retensi
Log kesalahan	Maksimal 30 log kesalahan dipertahankan. Amazon RDS dapat menghapus log kesalahan yang lebih lama dari 7 hari.
Log agen	Maksimal 10 log agen dipertahankan. Amazon RDS dapat menghapus log agen yang lebih lama dari 7 hari.
File jejak	File jejak dipertahankan sesuai dengan periode retensi file jejak instans DB Anda. Periode retensi file jejak default adalah 7 hari. Untuk mengubah periode retensi file jejak untuk instans DB, lihat Mengatur periode retensi untuk file pelacakan dan dump.
File dump	File jejak dipertahankan sesuai dengan periode retensi file dump instans DB Anda. Periode retensi file dump default adalah 7 hari. Untuk mengubah periode retensi file dump untuk instans DB, lihat Mengatur periode retensi untuk file pelacakan dan dump.

Melihat log kesalahan SQL Server menggunakan prosedur rds_read_error_log

Anda dapat menggunakan prosedur tersimpan Amazon RDS `rds_read_error_log` untuk melihat log kesalahan dan log agen. Untuk informasi selengkapnya, lihat Melihat log kesalahan dan agen.

Menerbitkan log SQL Server ke Amazon CloudWatch Logs

Dengan Amazon RDS for SQL Server, Anda dapat mempublikasikan kesalahan dan peristiwa log agen langsung ke Amazon Logs. CloudWatch Analisis data log dengan CloudWatch Log, lalu gunakan CloudWatch untuk membuat alarm dan melihat metrik.

Dengan CloudWatch Log, Anda dapat melakukan hal berikut:

- Menyimpan log dalam ruang penyimpanan tahan lama dengan periode retensi yang Anda tentukan.
- Mencari dan memfilter data log.
- Berbagi data log antarakun.
- Mengekspor log ke Amazon S3.
- Streaming data ke OpenSearch Layanan Amazon.
- Memproses data log secara real-time dengan Amazon Kinesis Data Streams. Untuk informasi selengkapnya, lihat Bekerja dengan Amazon CloudWatch Logs di Amazon Managed Service for Apache Flink for SQL Applications Developer Guide.

Amazon RDS menerbitkan setiap log basis data SQL Server sebagai aliran basis data terpisah dalam grup log. Misalnya, jika Anda mempublikasikan log agen dan log kesalahan, data kesalahan disimpan dalam aliran log kesalahan di grup `/aws/rds/instance/`*`my_instance`*`/error` log, dan data log agen disimpan dalam grup `/aws/rds/instance/`*`my_instance`*`/agent` log.

Untuk instans DB Multi-AZ, Amazon RDS menerbitkan log basis data sebagai dua aliran terpisah dalam grup log. Misalnya, jika Anda menerbitkan log kesalahan, data kesalahan akan disimpan dalam log stream kesalahan `/aws/rds/instance/`*`my_instance.node1`*`/error` dan `/aws/rds/instance/`*`my_instance.node2`*`/error` masing-masing. Log stream tidak berubah selama failover dan log stream kesalahan setiap simpul dapat berisi log kesalahan dari instans primer atau sekunder. Dengan Multi-AZ, aliran log secara otomatis dibuat `/aws/rds/instance/`*`my_instance`*`/rds-events` untuk menyimpan data peristiwa seperti kegagalan instans DB.

> **Note**
>
> Menerbitkan log SQL Server ke CloudWatch Log tidak diaktifkan secara default. Menerbitkan file jejak dan dump tidak didukung. Menerbitkan log SQL Server ke CloudWatch Log didukung di semua wilayah, kecuali untuk Asia Pasifik (Hong Kong).

Konsol

Untuk mempublikasikan log SQL Server DB ke CloudWatch Log dari AWS Management Console

1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data, lalu pilih instans DB yang ingin diubah.
3. Pilih Ubah.
4. Di bagian Log ekspor, pilih log yang ingin Anda mulai terbitkan ke CloudWatch Log.

   Anda dapat memilih Log agen, Log kesalahan, atau keduanya.
5. Pilih Lanjutkan, lalu pilih Ubah Instans DB di halaman ringkasan.

AWS CLI

Untuk menerbitkan log SQL Server, Anda dapat menggunakan perintah `modify-db-instance` dengan parameter berikut:

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

> **Note**
>
> Perubahan pada opsi `--cloudwatch-logs-export-configuration` selalu diterapkan ke instans DB secara langsung. Oleh karena itu, opsi `--apply-immediately` dan `--no-apply-immediately` tidak akan berpengaruh.

Anda juga dapat menerbitkan log SQL Server menggunakan perintah berikut:

- `create-db-instance`

- `restore-db-instance-from-db-snapshot`
- `restore-db-instance-to-point-in-time`

Example

Contoh berikut membuat instance SQL Server DB dengan penerbitan CloudWatch Log diaktifkan. Nilai `--enable-cloudwatch-logs-exports` adalah rangkaian JSON dari string yang dapat mencakup `error`, `agent`, atau keduanya.

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --enable-cloudwatch-logs-exports '["error","agent"]' \
    --db-instance-class db.m4.large \
    --engine sqlserver-se
```

Untuk Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --enable-cloudwatch-logs-exports "[\"error\",\"agent\"]" ^
    --db-instance-class db.m4.large ^
    --engine sqlserver-se
```

> **Note**
>
> Saat menggunakan command prompt Windows, Anda harus meng-escape tanda kutip ganda (") dalam kode JSON dengan memberikan garis miring terbalik (\) di depannya.

Example

Contoh berikut memodifikasi instance SQL Server DB yang ada untuk mempublikasikan file log ke Log. CloudWatch Nilai `--cloudwatch-logs-export-configuration` adalah objek JSON. Kunci untuk objek ini adalah `EnableLogTypes`, dan nilainya berupa rangkaian string yang dapat mencakup `error`, `agent`, atau keduanya.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":["error","agent"]}'
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration "{\"EnableLogTypes\":[\"error\",\"agent\"]}"
```

> **Note**
>
> Saat menggunakan command prompt Windows, Anda harus meng-escape tanda kutip ganda (") dalam kode JSON dengan memberikan garis miring terbalik (\) di depannya.

Example

Contoh berikut memodifikasi instance SQL Server DB yang ada untuk menonaktifkan file log agen penerbitan ke CloudWatch Log. Nilai `--cloudwatch-logs-export-configuration` adalah objek JSON. Kunci untuk objek ini adalah `DisableLogTypes`, dan nilainya berupa rangkaian string yang dapat mencakup `error`, `agent`, atau keduanya.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"DisableLogTypes":["agent"]}'
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration "{\"DisableLogTypes\":[\"agent\"]}"
```

> **Note**
>
> Saat menggunakan perintah Windows, Anda harus meng-escape tanda kutip ganda (") dalam kode JSON dengan memberinya awalan garis miring terbalik (\).

File log basis data MySQL

Anda dapat memantau log MySQL secara langsung melalui konsol Amazon RDS, Amazon RDS API, AWS CLI, atau AWS SDK. Anda juga dapat mengakses log MySQL dengan mengarahkan log ke tabel basis data di basis data utama dan mengkueri tabel tersebut. Anda dapat menggunakan utilitas mysqlbinlog untuk mengunduh log biner.

Untuk mengetahui informasi selengkapnya tentang cara melihat, mengunduh, dan melihat log basis data berbasis file, lihat Memantau file log Amazon RDS.

Topik

- Ikhtisar log basis data RDS for MySQL
- Menerbitkan log MySQL ke Amazon Logs CloudWatch
- Mengelola log MySQL berbasis tabel
- Mengkonfigurasi pengelogan biner MySQL
- Mengakses log biner MySQL

Ikhtisar log basis data RDS for MySQL

Anda dapat memantau jenis file log RDS for MySQL berikut:

- Log kesalahan
- Log kueri lambat
- Log umum
- Log audit

Log kesalahan RDS for MySQL dihasilkan secara default. Anda dapat membuat kueri lambat dan log umum dengan mengatur parameter di grup parameter DB Anda.

Topik

- Log kesalahan RDS for MySQL
- Log umum dan kueri lambat RDS for MySQL
- Log audit MySQL
- Rotasi dan retensi log untuk RDS for MySQL
- Batas ukuran pada log redo

Log kesalahan RDS for MySQL

Kesalahan tulis RDS for MySQL dalam file `mysql-error.log`. Setiap file log memiliki jam pembuatan (dalam UTC) yang ditambahkan pada namanya. File log juga memiliki stempel waktu yang membantu Anda menentukan kapan entri log ditulis.

RDS for MySQL ditulis ke log kesalahan hanya saat dinyalakan, dimatikan, dan saat terjadi kesalahan. Instans DB dapat memakan waktu berjam-jam atau berhari-hari tanpa perlu menulis entri baru ke log kesalahan. Jika Anda melihat tidak ada entri terbaru, berarti server tidak mengalami kesalahan yang akan mengakibatkan entri log.

Secara desain, log kesalahan difilter sehingga hanya peristiwa tak terduga seperti kesalahan yang ditampilkan. Namun, log kesalahan juga berisi beberapa informasi basis data tambahan, misalnya kemajuan kueri, yang tidak ditampilkan. Oleh karena itu, bahkan tanpa kesalahan aktual, ukuran log kesalahan mungkin meningkat dikarenakan aktivitas basis data yang sedang berlangsung. Dan meskipun Anda mungkin melihat ukuran tertentu dalam byte atau kilobyte untuk log kesalahan di AWS Management Console, log tersebut mungkin memiliki 0 byte saat Anda mengunduhnya.

RDS for MySQL menulis `mysql-error.log` ke disk setiap 5 menit. Ini menambahkan konten log ke`mysql-error-running.log`.

RDS for MySQL merotasi file `mysql-error-running.log` setiap jam. Ini mempertahankan log yang dihasilkan selama dua minggu terakhir.

> **Note**
>
> Periode retensi log berbeda antara Amazon RDS dan Aurora.

Log umum dan kueri lambat RDS for MySQL

Anda dapat menulis log umum dan log kueri lambat RDS for MySQL ke file atau tabel basis data. Untuk melakukannya, atur parameter di grup parameter DB Anda. Untuk informasi tentang pembuatan dan modifikasi grup parameter DB, lihat Bekerja dengan grup parameter. Anda harus mengatur parameter ini sebelum dapat melihat log kueri lambat atau log umum di konsol Amazon RDS atau dengan menggunakan Amazon RDS API, Amazon RDS CLI, atau AWS SDK.

Anda dapat mengontrol pengelogan RDS for MySQL dengan menggunakan parameter dalam daftar ini:

- `slow_query_log`: Untuk membuat log kueri lambat, atur ke 1. Nilai default-nya adalah 0.

- `general_log`: Untuk membuat log umum, atur ke 1. Nilai default-nya adalah 0.
- `long_query_time`: Untuk mencegah kueri yang berjalan cepat masuk ke log kueri lambat, tentukan nilai untuk runtime kueri terpendek yang akan dicatat, dalam detik. Nilai default-nya adalah 10 detik; nilai minimumnya adalah 0. Jika log_output = FILE, Anda dapat menentukan nilai titik mengambang yang masuk ke resolusi mikrodetik. Jika log_output = TABLE, Anda harus menentukan nilai integer dengan resolusi kedua. Hanya kueri yang runtime-nya melampaui nilai `long_query_time` yang akan dicatat. Misalnya, mengatur `long_query_time` ke 0,1 akan mencegah pengelogan kueri apa pun yang berjalan kurang dari 100 milidetik.
- `log_queries_not_using_indexes`: Untuk mencatat semua kueri yang tidak menggunakan indeks pada log kueri lambat, atur ke 1. Kueri yang tidak menggunakan indeks dicatat meskipun runtime-nya kurang dari nilai parameter `long_query_time`. Nilai default-nya adalah 0.
- `log_output` *`option`*: Anda dapat menentukan salah satu opsi berikut untuk parameter `log_output`.
  - TABLE (default) – Menulis kueri umum ke tabel `mysql.general_log`, dan kueri lambat ke tabel `mysql.slow_log`.
  - FILE – Menulis log umum dan log kueri lambat ke sistem file.
  - NONE – Menonaktifkan pengelogan.

Untuk informasi selengkapnya tentang log umum dan kueri lambat, buka topik berikut di dokumentasi MySQL:

- Log kueri lambat
- Log kueri umum

Log audit MySQL

Untuk mengakses log audit, instans DB harus menggunakan grup opsi kustom dengan opsi `MARIADB_AUDIT_PLUGIN`. Untuk informasi selengkapnya, lihat Dukungan MariaDB Audit Plugin untuk MySQL.

Rotasi dan retensi log untuk RDS for MySQL

Saat pengelogan diaktifkan, Amazon RDS merotasi log tabel atau menghapus file log secara berkala. Langkah ini merupakan tindakan pencegahan untuk mengurangi kemungkinan file log besar memblokir penggunaan basis data atau memengaruhi performa. RDS for MySQL menangani rotasi dan penghapusan sebagai berikut:

- Ukuran file log umum, log kesalahan, dan log kueri lambat MySQL dibatasi hingga tidak lebih dari 2 persen dari ruang penyimpanan yang dialokasikan untuk instans DB. Untuk mempertahankan ambang batas ini, log secara otomatis dirotasi setiap jam. MySQL menghapus file log yang berusia lebih dari dua minggu. Jika ukuran file log gabungan melebihi ambang batas setelah file log lama dihapus, file log paling lama akan dihapus hingga ukuran file log tidak lagi melebihi ambang batas.
- Jika pengelogan `FILE` diaktifkan, file log akan diperiksa setiap jam dan file log yang berusia lebih dari dua minggu akan dihapus. Dalam beberapa kasus, ukuran file log gabungan yang tersisa setelah penghapusan mungkin melebihi ambang batas 2 persen dari ruang yang dialokasikan oleh instans DB. Dalam kasus ini, file log yang paling lama akan dihapus hingga ukuran file log tidak lagi melebihi ambang batas.
- Saat pengelogan `TABLE` diaktifkan, dalam beberapa kasus tabel log dirotasi setiap 24 jam. Rotasi ini terjadi jika ruang yang digunakan oleh log tabel lebih dari 20 persen dari ruang penyimpanan yang dialokasikan. Ini juga terjadi jika ukuran semua log yang digabungkan lebih besar dari 10 GB. Jika jumlah ruang yang digunakan untuk instans DB lebih besar dari 90 persen dari ruang penyimpanan yang dialokasikan untuk instans DB, ambang batas untuk rotasi log akan berkurang. Tabel log ini kemudian dirotasi jika ruang yang digunakan oleh log tabel lebih dari 10 persen dari ruang penyimpanan yang dialokasikan. Tabel ini juga dirotasi jika ukuran semua log yang digabungkan lebih besar dari 5 GB. Anda dapat berlangganan peristiwa `low_free_storage` yang akan diberitahukan saat tabel log dirotasi untuk mengosongkan ruang. Untuk informasi selengkapnya, lihat Menggunakan pemberitahuan peristiwa Amazon RDS.

  Saat tabel log dirotasi, tabel log saat ini akan disalin terlebih dahulu ke tabel log cadangan. Kemudian entri dalam tabel log saat ini dihapus. Jika sudah ada, tabel log cadangan akan dihapus sebelum tabel log saat ini disalin ke cadangan. Anda dapat meminta tabel log cadangan jika diperlukan. Tabel log cadangan untuk tabel `mysql.general_log` bernama `mysql.general_log_backup`. Tabel log cadangan untuk tabel `mysql.slow_log` bernama `mysql.slow_log_backup`.

  Anda dapat merotasi tabel `mysql.general_log` dengan mengikuti prosedur `mysql.rds_rotate_general_log`. Anda dapat merotasi `mysql.slow_log` tabel dengan mengikuti `mysql.rds_rotate_slow_log` prosedur.

  Log tabel dirotasi selama upgrade versi basis data.

Untuk bekerja dengan log dari konsol Amazon RDS, Amazon RDS API, Amazon RDS CLI, atau AWS SDK, atur parameter `log_output` ke FILE. Seperti log kesalahan MySQL, file log ini dirotasi setiap

jam. File log yang dihasilkan selama dua minggu sebelumnya akan dipertahankan. Perhatikan bahwa periode retensi log berbeda antara Amazon RDS dan Aurora.

Batas ukuran pada log redo

Untuk RDS for MySQL versi 8.0.32 dan yang lebih rendah, nilai default parameter ini adalah 256 MB. Jumlah ini diturunkan dengan mengalikan nilai default parameter `innodb_log_file_size` (128 MB) dengan nilai default parameter `innodb_log_files_in_group` (2). Untuk informasi selengkapnya, lihat Praktik terbaik untuk mengonfigurasi parameter untuk Amazon RDS for MySQL, bagian 1: Parameter yang terkait dengan performa.

Dimulai dengan RDS for MySQL versi 8.0.33, Amazon RDS menggunakan parameter `innodb_redo_log_capacity`, bukan parameter `innodb_log_file_size`. Nilai default Amazon RDS parameter `innodb_redo_log_capacity` adalah 2 GB. Untuk informasi selengkapnya, lihat Perubahan pada MySQL 8.0.30 di dokumentasi MySQL.

Menerbitkan log MySQL ke Amazon Logs CloudWatch

Anda dapat mengonfigurasi instans MySQL DB Anda untuk mempublikasikan data log ke grup log di Amazon Logs. CloudWatch Dengan CloudWatch Log, 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 sangat tahan lama.

Amazon RDS menerbitkan setiap log basis data MySQL sebagai aliran basis data terpisah di grup log. Misalnya, jika Anda mengonfigurasi fungsi ekspor untuk menyertakan log kueri lambat, data kueri lambat akan disimpan dalam log stream kueri lambat di grup log `/aws/rds/instance/my_instance/slowquery`.

Log kesalahan diaktifkan secara default. Tabel berikut merangkum persyaratan untuk log MySQL lain.

Log	Persyaratan
Log audit	Instans DB harus menggunakan grup opsi kustom dengan opsi `MARIADB_AUDIT_PLUGIN` .
Log umum	Instans DB harus menggunakan grup parameter kustom dengan pengaturan

Log	Persyaratan
	parameter `general_log = 1` untuk mengaktifkan log umum.
Log kueri lambat	Instans DB harus menggunakan grup parameter kustom dengan pengaturan parameter `slow_query_log = 1` untuk mengaktifkan log kueri lambat.
Output log	DB instance harus menggunakan grup parameter khusus dengan pengaturan parameter `log_output = FILE` untuk mencatat log ke sistem file dan menerbitkannya ke CloudWatch Logs.

Konsol

Untuk mempublikasikan log MySQL CloudWatch ke Log menggunakan konsol

1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data, lalu pilih instans DB yang ingin diubah.
3. Pilih Ubah.
4. Di bagian Log ekspor, pilih log yang ingin Anda mulai terbitkan ke CloudWatch Log.
5. Pilih Lanjutkan, lalu pilih Ubah Instans DB di halaman ringkasan.

AWS CLI

Anda dapat menerbitkan log MySQL dengan AWS CLI. Anda dapat memanggil perintah `modify-db-instance` dengan parameter berikut:

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

> **Note**
>
> Perubahan pada opsi `--cloudwatch-logs-export-configuration` selalu diterapkan ke instans DB secara langsung. Oleh karena itu, opsi `--apply-immediately` dan `--no-apply-immediately` tidak akan berpengaruh.

Anda juga dapat menerbitkan log MySQL dengan memanggil perintah AWS CLI berikut:

- `create-db-instance`
- `restore-db-instance-from-db-snapshot`
- `restore-db-instance-from-s3`
- `restore-db-instance-to-point-in-time`

Jalankan salah satu perintah AWS CLI ini dengan opsi berikut:

- `--db-instance-identifier`
- `--enable-cloudwatch-logs-exports`
- `--db-instance-class`
- `--engine`

Opsi lain mungkin diperlukan bergantung pada perintah AWS CLI yang Anda jalankan.

Example

Contoh berikut memodifikasi instance MySQL DB yang ada untuk mempublikasikan file log ke Log. CloudWatch Nilai `--cloudwatch-logs-export-configuration` adalah objek JSON. Kunci untuk objek ini adalah `EnableLogTypes`, dan nilainya adalah serangkaian string dengan setiap kombinasi `audit`, `error`, `general`, dan `slowquery`.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":
["audit","error","general","slowquery"]}'
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":
["audit","error","general","slowquery"]}'
```

Example

Contoh berikut membuat instance MySQL DB dan menerbitkan file log ke Log. CloudWatch Nilai `--enable-cloudwatch-logs-exports` adalah rangkaian string JSON. String dapat berupa kombinasi `audit`, `error`, `general`, dan `slowquery`.

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --enable-cloudwatch-logs-exports '["audit","error","general","slowquery"]' \
    --db-instance-class db.m4.large \
    --engine MySQL
```

Untuk Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --enable-cloudwatch-logs-exports '["audit","error","general","slowquery"]' ^
    --db-instance-class db.m4.large ^
    --engine MySQL
```

RDS API

Anda dapat menerbitkan log MySQL dengan RDS API. Anda dapat memanggil tindakan `ModifyDBInstance` dengan parameter berikut:

- `DBInstanceIdentifier`
- `CloudwatchLogsExportConfiguration`

> **Note**
>
> Perubahan pada parameter `CloudwatchLogsExportConfiguration` selalu diterapkan ke instans DB secara langsung. Oleh karena itu, parameter `ApplyImmediately` tidak memiliki dampak.

Anda juga dapat menerbitkan log MySQL dengan memanggil operasi RDS API berikut:

- `CreateDBInstance`
- `RestoreDBInstanceFromDBSnapshot`
- `RestoreDBInstanceFromS3`
- `RestoreDBInstanceToPointInTime`

Jalankan salah satu operasi RDS API ini dengan parameter berikut:

- `DBInstanceIdentifier`
- `EnableCloudwatchLogsExports`
- `Engine`
- `DBInstanceClass`

Parameter lain mungkin diperlukan bergantung pada perintah AWS CLI yang Anda jalankan.

Mengelola log MySQL berbasis tabel

Anda dapat mengarahkan log umum dan log kueri lambat ke tabel di instans DB dengan membuat grup parameter DB dan menetapkan parameter server `log_output` ke `TABLE`. Kueri umum lalu dicatat ke tabel `mysql.general_log` dan kueri lambat dicatat ke tabel `mysql.slow_log`. Anda dapat mengueri tabel untuk mengakses informasi log. Mengaktifkan pencatatan ini akan meningkatkan jumlah data yang akan ditulis ke basis data. Hal ini dapat menurunkan performa.

Log umum dan log kueri lambat dinonaktifkan secara default. Untuk mengaktifkan pencatatan ke tabel, Anda juga harus menetapkan parameter server `general_log` dan `slow_query_log` ke `1`.

Tabel log terus bertambah hingga aktivitas pencatatan terkait dinonaktifkan dengan mengatur ulang parameter yang sesuai ke `0`. Banyak data yang sering terakumulasi seiring berjalannya waktu. Hal ini dapat menghabiskan cukup banyak ruang penyimpanan yang dialokasikan. Amazon RDS tidak

memungkinkan Anda memotong tabel log, tetapi Anda dapat memindahkan konten tabel. Merotasi tabel akan menyimpan kontennya ke tabel cadangan dan membuat tabel log kosong yang baru. Anda dapat merotasi tabel log secara manual dengan mengikuti prosedur perintah berikut, dengan permintaan perintah ditunjukkan oleh `PROMPT>`:

```
PROMPT> CALL mysql.rds_rotate_slow_log;
PROMPT> CALL mysql.rds_rotate_general_log;
```

Untuk menghapus data lama sepenuhnya dan mengosongkan kembali ruang disk, panggil prosedur yang sesuai dua kali secara berurutan.

Mengkonfigurasi pengelogan biner MySQL

Log biner adalah sekumpulan file log yang berisi informasi tentang modifikasi data yang dibuat ke instans server MySQL. Log biner berisi informasi seperti berikut:

- Peristiwa yang menggambarkan perubahan basis data seperti pembuatan tabel atau modifikasi baris
- Informasi tentang durasi setiap pernyataan yang memperbarui data
- Peristiwa untuk pernyataan yang bisa saja memperbarui data, tetapi tidak

Log biner mencatat pernyataan yang dikirim selama replikasi. Log ini juga diperlukan untuk beberapa operasi pemulihan. Untuk informasi selengkapnya, lihat Log Biner dan Ikhtisar Log Biner dalam dokumentasi MySQL.

Fitur cadangan otomatis menentukan apakah pengelogan biner diaktifkan atau dinonaktifkan untuk MySQL. Anda memiliki opsi berikut:

Aktifkan logging biner

Mengatur periode retensi cadangan ke nilai non-nol positif.

Nonaktifkan logging biner

Mengatur periode retensi cadangan ke nol.

Untuk informasi selengkapnya, lihat Mengaktifkan pencadangan otomatis.

MySQL di Amazon RDS mendukung format pengelogan biner berbasis baris, berbasis pernyataan, dan campuran. Kami merekomendasikan campuran kecuali Anda memerlukan format binlog

tertentu. Untuk detail tentang format log biner MySQL lainnya, lihat Format pengelogan biner dalam dokumentasi MySQL.

Jika Anda berencana menggunakan replikasi, format pengelogan biner diperlukan karena menentukan catatan perubahan data yang dicatat di sumber dan dikirim ke target replikasi. Untuk informasi tentang kelebihan dan kelemahan format pengelogan biner lainnya untuk replikasi, lihat Kelebihan dan kelemahan replikasi berbasis pernyataan dan berbasis baris dalam dokumentasi MySQL.

> **Important**
>
> Mengatur format pengelogan biner ke berbasis baris dapat menghasilkan file log biner yang sangat besar. File log biner besar mengurangi jumlah penyimpanan yang tersedia untuk klaster DB dan dapat meningkatkan jumlah waktu yang dibutuhkan untuk melakukan operasi pemulihan klaster DB.
>
> Replikasi berbasis pernyataan dapat menyebabkan inkonsistensi antara klaster DB dan replika baca. Untuk informasi selengkapnya, lihat Penentuan pernyataan yang aman dan tidak aman dalam pengelogan biner di dokumentasi MySQL.
>
> Mengaktifkan pengelogan biner akan meningkatkan jumlah operasi I/O disk tulis untuk klaster DB. Anda dapat memantau penggunaan IOPS dengan `WriteIOPS` CloudWatch metrik.

Untuk mengatur format pengelogan biner MySQL

1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup parameter.
3. Pilih grup parameter DB, yang terkait dengan instans DB, yang ingin dimodifikasi.

   Anda tidak dapat mengubah grup parameter default. Jika klaster DB menggunakan grup parameter default, buat grup parameter baru dan hubungkan dengan klaster DB.

   Untuk informasi selengkapnya tentang grup parameter, lihat Bekerja dengan grup parameter.
4. Dari Tindakan, pilih Edit.
5. Atur parameter `binlog_format` ke format pengelogan biner pilihan Anda (`ROW`, `STATEMENT`, atau `MIXED`).

   Anda dapat menonaktifkan pengelogan biner dengan mengatur periode retensi cadangan instans DB ke nol, tetapi tindakan ini akan menonaktifkan cadangan otomatis harian.

Menonaktifkan backup otomatis mematikan atau menonaktifkan variabel sesi. `log_bin` Ini menonaktifkan logging biner pada RDS untuk instance MySQL DB, yang pada gilirannya mengatur ulang variabel `binlog_format` sesi ke nilai default dalam database. `ROW` Sebaiknya jangan menonaktifkan cadangan. Untuk informasi selengkapnya tentang pengaturan Periode penyimpanan cadangan, lihat Pengaturan untuk instans DB.

6. Pilih Simpan perubahan untuk menyimpan pembaruan ke grup parameter DB.

Karena parameter `binlog_format` bersifat dinamis, Anda tidak perlu me-reboot instans DB untuk menerapkan perubahan.

> **Important**
>
> Mengubah grup parameter DB memengaruhi semua instans DB yang menggunakan grup parameter tersebut. Jika Anda ingin menentukan format pengelogan biner lainnya untuk instans DB MySQL di Wilayah AWS, instans DB harus menggunakan grup parameter DB yang berbeda. Grup parameter ini mengidentifikasi format pengelogan yang berbeda. Tetapkan grup parameter DB yang sesuai ke masing-masing instans DB.

Mengakses log biner MySQL

Anda dapat menggunakan utilitas mysqlbinlog untuk mengunduh atau mengalirkan log biner dari instans DB RDS untuk MySQL. Log biner diunduh ke komputer lokal, tempat Anda dapat melakukan tindakan seperti memutar ulang log menggunakan utilitas mysql. Untuk informasi selengkapnya tentang cara menggunakan utilitas mysqlbinlog, lihat Menggunakan mysqlbinlog untuk mencadangkan file log biner dalam dokumentasi MySQL.

Untuk menjalankan utilitas mysqlbinlog terhadap instans Amazon RDS, gunakan opsi berikut:

- `--read-from-remote-server` – Wajib diisi.
- `--host` – Nama DNS dari titik akhir instans.
- `--port` – Port yang digunakan oleh instans.
- `--user` – Pengguna MySQL yang telah diberi izin `REPLICATION SLAVE`.
- `--password` – Kata sandi untuk pengguna MySQL, atau hapus nilai kata sandi agar utilitas meminta Anda memasukkan kata sandi.
- `--raw` – Mengunduh file dalam format biner.

- `--result-file` – File lokal untuk menerima output mentah.
- `--stop-never` – Mengalirkan file log biner.
- `--verbose` – Jika Anda menggunakan format binlog `ROW`, sertakan opsi ini untuk melihat peristiwa baris sebagai pernyataan pseudo-SQL. Untuk informasi selengkapnya tentang opsi `--verbose`, lihat tampilan peristiwa baris mysqlbinlog dalam dokumentasi MySQL.
- Tentukan nama satu atau beberapa file log biner. Untuk mendapatkan daftar log yang tersedia, gunakan perintah SQL `SHOW BINARY LOGS`.

Untuk informasi selengkapnya tentang opsi mysqlbinlog, lihat mysqlbinlog — untuk memperoses file log biner dalam dokumentasi MySQL.

Contoh berikut menunjukkan cara menggunakan utilitas mysqlbinlog.

Untuk Linux, macOS, atau Unix:

```
mysqlbinlog \
    --read-from-remote-server \
    --host=MySQLInstance1.cg034hpkmmjt.region.rds.amazonaws.com \
    --port=3306  \
    --user ReplUser \
    --password \
    --raw \
    --verbose \
    --result-file=/tmp/ \
    binlog.00098
```

Untuk Windows:

```
mysqlbinlog ^
    --read-from-remote-server ^
    --host=MySQLInstance1.cg034hpkmmjt.region.rds.amazonaws.com ^
    --port=3306  ^
    --user ReplUser ^
    --password ^
    --raw ^
    --verbose ^
    --result-file=/tmp/ ^
    binlog.00098
```

Amazon RDS biasanya membersihkan log biner sesegera mungkin, tetapi log biner ini harus tetap tersedia di instans untuk diakses oleh mysqlbinlog. Untuk menentukan jumlah jam yang dibutuhkan RDS untuk mempertahankan log biner, gunakan prosedur tersimpan mysql.rds_set_configuration dan tentukan periode yang cukup agar Anda dapat mengunduh log. Setelah Anda mengatur periode retensi, pantau penggunaan penyimpanan untuk instans DB guna memastikan bahwa log biner yang dipertahankan tidak memakan terlalu banyak ruang penyimpanan.

Contoh berikut menetapkan periode retensi ke 1 hari.

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

Untuk menampilkan pengaturan saat ini, gunakan prosedur tersimpan mysql.rds_show_configuration.

```
call mysql.rds_show_configuration;
```

File log basis data Oracle

Anda dapat mengakses log peringatan, file audit, dan file jejak Oracel dengan menggunakan konsol Amazon RDS atau API. Untuk informasi selengkapnya tentang melihat, mengunduh, dan melihat log basis data berbasis file, lihat Memantau file log Amazon RDS.

File audit Oracle yang disediakan adalah file audit Oracle standar. Amazon RDS mendukung fitur audit mendetail (FGA) Oracle. Namun, akses log tidak menyediakan akses ke peristiwa FGA yang disimpan di tabel `SYS.FGA_LOG$` yang dapat diakses melalui tampilan `DBA_FGA_AUDIT_TRAIL`.

Operasi API `DescribeDBLogFiles` yang mencantumkan file log Oracle yang tersedia untuk instans DB mengabaikan parameter `MaxRecords` dan menampilkan hingga 1.000 data. Panggilan menampilkan `LastWritten` sebagai tanggal POSIX dalam milidetik.

Topik

- Jadwal retensi
- Bekerja dengan file jejak Oracle
- Menerbitkan log Oracle ke Amazon CloudWatch Logs
- Metode sebelumnya untuk mengakses log peringatan dan log pendengar

Jadwal retensi

Mesin basis data Oracle dapat merotasi file log jika menjadi sangat besar. Untuk mempertahankan file audit atau jejak, unduh semuanya. Jika menyimpan file secara lokal, Anda dapat menurunkan biaya penyimpanan Amazon RDS dan menyediakan lebih banyak ruang untuk data Anda.

Tabel berikut menunjukkan jadwal retensi untuk log peringatan, file audit, dan file jejak Oracle di Amazon RDS.

Jenis log	Jadwal retensi
Log peringatan	Log peringatan teks dirotasi setiap hari dengan 30 hari retensi yang dikelola oleh Amazon RDS. Log peringatan XML dipertahankan selama setidaknya tujuh hari. Anda dapat mengakses log ini dengan menggunak an tampilan `ALERTLOG`.
File audit	Periode retensi default untuk file audit adalah tujuh hari. Amazon RDS mungkin menghapus file audit yang lebih lama dari tujuh hari.

Jenis log	Jadwal retensi
File jejak	Periode retensi default untuk file jejak adalah tujuh hari. Amazon RDS mungkin menghapus file jejak yang lebih lama dari tujuh hari.
Log pendengar	Periode retensi default untuk log pendengar adalah tujuh hari. Amazon RDS mungkin menghapus log pendengar yang lebih lama dari tujuh hari.

> **Note**
>
> File audit dan file jejak memiliki konfigurasi penyimpanan yang sama.

Bekerja dengan file jejak Oracle

Selanjutnya, Anda dapat menemukan deskripsi prosedur Amazon RDS untuk membuat, menyegarkan, mengakses, dan menghapus file jejak.

Topik

- Membuat daftar file
- Membuat file jejak dan melacak sesi
- Mengambil file jejak
- Membersihkan file jejak

Membuat daftar file

Anda dapat menggunakan salah satu dari dua prosedur untuk mengizinkan akses ke file apa pun di jalur `background_dump_dest`. Prosedur pertama menyegarkan tampilan yang berisi daftar semua file saat ini ada di `background_dump_dest`.

```
EXEC rdsadmin.manage_tracefiles.refresh_tracefile_listing;
```

Setelah tampilan dimuat ulang, buat kueri tampilan berikut untuk mengakses hasil.

```
SELECT * FROM rdsadmin.tracefile_listing;
```

Alternatif untuk proses sebelumnya adalah menggunakan `FROM table` untuk mengalirkan data non-relasional dalam format seperti tabel untuk mencantumkan konten direktori basis data.

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir('BDUMP'));
```

Kueri berikut menunjukkan teks file log.

```
SELECT text FROM
 TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP','alert_dbname.log.date'));
```

Pada replika baca, dapatkan nama direktori BDUMP dengan mengkueri V `$DATABASE.DB_UNIQUE_NAME`. Jika nama uniknya adalah `DATABASE_B`, maka direktori BDUMP-nya adalah `BDUMP_B`. Contoh berikut mengkueri nama BDUMP pada replika dan kemudian menggunakan nama ini untuk mengkueri konten `alert_DATABASE.log.2020-06-23`.

```
SELECT 'BDUMP' || (SELECT regexp_replace(DB_UNIQUE_NAME,'.*(_[A-Z])', '\1') FROM V
$DATABASE) AS BDUMP_VARIABLE FROM DUAL;

BDUMP_VARIABLE
--------------
BDUMP_B

SELECT TEXT FROM
 table(rdsadmin.rds_file_util.read_text_file('BDUMP_B','alert_DATABASE.log.2020-06-23'));
```

Membuat file jejak dan melacak sesi

Karena tidak ada batasan pada `ALTER SESSION`, banyak metode standar untuk menghasilkan file jejak dalam Oracle yang tetap tersedia untuk instans DB Amazon RDS. Prosedur berikut disediakan untuk file jejak yang memerlukan akses lebih besar.

Metode Oracle	Metode Amazon RDS
`oradebug hanganalyze 3`	`EXEC rdsadmin.manage_tracefiles.hanganalyze;`
`oradebug dump systemstate 266`	`EXEC rdsadmin.manage_tracefiles.dump_systemstate;`

Anda dapat menggunakan berbagai metode standar untuk melacak sesi individu yang terhubung ke instans DB Oracle di Amazon RDS. Untuk memungkinkan pelacakan sesi, Anda dapat menjalankan subprogram dalam paket PL/SQL yang disediakan oleh Oracle, seperti `DBMS_SESSION` dan `DBMS_MONITOR`. Untuk informasi selengkapnya, lihat Mengaktifkan pelacakan untuk sesi dalam dokumentasi Oracle.

Mengambil file jejak

Anda dapat mengambil file jejak apa pun dalam `background_dump_dest` menggunakan kueri SQL standar di tabel eksternal yang dikelola Amazon RDS. Untuk menggunakan metode ini, Anda harus menjalankan prosedur untuk menetapkan lokasi untuk tabel ini ke file jejak spesifik.

Misalnya, Anda dapat menggunakan tampilan `rdsadmin.tracefile_listing` yang disebutkan sebelumnya untuk mencantumkan semua file jejak pada sistem. Anda kemudian dapat mengatur tampilan `tracefile_table` untuk diarahkan ke file ke jejak yang dimaksud dengan menggunakan prosedur berikut.

```
EXEC
 rdsadmin.manage_tracefiles.set_tracefile_table_location('CUST01_ora_3260_SYSTEMSTATE.trc');
```

Contoh berikut membuat tabel eksternal dalam skema saat ini dengan lokasi yang diatur ke file yang disediakan. Anda dapat mengambil konten ke dalam file lokal menggunakan kueri SQL.

```
SPOOL /tmp/tracefile.txt
SELECT * FROM tracefile_table;
SPOOL OFF;
```

Membersihkan file jejak

File jejak dapat terakumulasi dan menghabiskan ruang disk. Amazon RDS menghapus file jejak secara default dan file log yang lebih lama dari tujuh hari. Anda dapat melihat dan mengatur periode retensi file jejak menggunakan prosedur `show_configuration`. Anda harus menjalankan perintah `SET SERVEROUTPUT ON` agar dapat melihat hasil konfigurasi.

Contoh berikut menunjukkan periode retensi file jejak saat ini, lalu mengatur periode retensi file jejak baru.

```
# Show the current tracefile retention
SQL> EXEC rdsadmin.rdsadmin_util.show_configuration;
NAME:tracefile retention
VALUE:10080
```

```
DESCRIPTION:tracefile expiration specifies the duration in minutes before tracefiles in
 bdump are automatically deleted.

# Set the tracefile retention to 24 hours:
SQL> EXEC rdsadmin.rdsadmin_util.set_configuration('tracefile retention',1440);
SQL> commit;

#show the new tracefile retention
SQL> EXEC rdsadmin.rdsadmin_util.show_configuration;
NAME:tracefile retention
VALUE:1440
DESCRIPTION:tracefile expiration specifies the duration in minutes before tracefiles in
 bdump are automatically deleted.
```

Selain proses pembersihan berkala, Anda dapat menghapus file secara manual dari `background_dump_dest`. Contoh berikut menunjukkan cara membersihkan semua file yang lebih lama dari lima menit.

```
EXEC rdsadmin.manage_tracefiles.purge_tracefiles(5);
```

Anda juga dapat membersihkan semua file yang cocok dengan pola tertentu (jika Anda melakukannya, jangan sertakan ekstensi file, seperti .trc). Contoh berikut menunjukkan cara membersihkan semua file yang dimulai dengan `SCHPOC1_ora_5935`.

```
EXEC rdsadmin.manage_tracefiles.purge_tracefiles('SCHPOC1_ora_5935');
```

Menerbitkan log Oracle ke Amazon CloudWatch Logs

Anda dapat mengonfigurasi RDS untuk instans Oracle DB untuk mempublikasikan data log ke grup log di Amazon CloudWatch Logs. Dengan CloudWatch Log, Anda dapat menganalisis data log, dan menggunakannya CloudWatch untuk membuat alarm dan melihat metrik. Anda dapat menggunakan CloudWatch Log untuk menyimpan catatan log Anda dalam penyimpanan yang sangat tahan lama.

Amazon RDS menerbitkan setiap log basis data Oracle sebagai aliran basis data terpisah di grup log. Misalnya, jika Anda mengonfigurasi fungsi ekspor untuk menyertakan log audit, data audit akan disimpan dalam log stream audit di grup log `/aws/rds/instance/my_instance/audit`. Tabel berikut merangkum persyaratan untuk RDS untuk Oracle untuk menerbitkan log ke Amazon Logs. CloudWatch

Nama log	Persyaratan	Default
Log peringatan	Tidak ada. Anda tidak dapat menonaktifkan log ini.	Aktif
Log jejak	Atur parameter `trace_enabled` ke `TRUE` atau biarkan diatur ke setelan default.	`TRUE`
Log audit	Atur parameter `audit_trail` ke `OS`, `XML`, atau `EXTENDED`.	`NONE`
Log pendengar	Tidak ada. Anda tidak dapat menonaktifkan log ini.	Aktif
Log Agen Manajemen Oracle	Tidak ada. Anda tidak dapat menonaktifkan log ini.	Aktif

Log Agen Manajemen Oracle ini terdiri dari grup log yang ditunjukkan pada tabel berikut.

Nama log	CloudWatch grup log
emctl.log	oemagent-emctl
emdctlj.log	oemagent-emdctlj
gcagent.log	oemagent-gcagent
gcagent_errors.log	oemagent-gcagent-errors
emagent.nohup	oemagent-emagent-nohup
secure.log	oemagent-secure

Untuk informasi selengkapnya, lihat Menemukan Log Agen Manajemen dan File Jejak dalam dokumentasi Oracle.

Konsol

Untuk mempublikasikan log Oracle DB ke CloudWatch Log dari AWS Management Console

1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data, lalu pilih instans DB yang ingin diubah.
3. Pilih Ubah.
4. Di bagian Log ekspor, pilih log yang ingin Anda mulai terbitkan ke CloudWatch Log.
5. Pilih Lanjutkan, lalu pilih Ubah Instans DB di halaman ringkasan.

AWS CLI

Untuk menerbitkan log Oracle, Anda dapat menggunakan perintah `modify-db-instance` dengan parameter berikut:

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

> **Note**
>
> Perubahan pada opsi `--cloudwatch-logs-export-configuration` selalu diterapkan ke instans DB secara langsung. Oleh karena itu, opsi `--apply-immediately` dan `--no-apply-immediately` tidak akan berpengaruh.

Anda juga dapat menerbitkan log Oracle menggunakan perintah berikut:

- `create-db-instance`
- `restore-db-instance-from-db-snapshot`
- `restore-db-instance-from-s3`
- `restore-db-instance-to-point-in-time`

Example

Contoh berikut membuat instance Oracle DB dengan penerbitan CloudWatch Log diaktifkan. Nilai `--cloudwatch-logs-export-configuration` adalah rangkaian string JSON. String dapat berupa kombinasi `alert`, `audit`, `listener`, dan `trace`.

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration
 '["trace","audit","alert","listener","oemagent"]' \
    --db-instance-class db.m5.large \
    --allocated-storage 20 \
    --engine oracle-ee \
    --engine-version 12.1.0.2.v18 \
    --license-model bring-your-own-license \
    --master-username myadmin \
    --manage-master-user-password
```

Untuk Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration trace alert audit listener oemagent ^
    --db-instance-class db.m5.large ^
    --allocated-storage 20 ^
    --engine oracle-ee ^
    --engine-version 12.1.0.2.v18 ^
    --license-model bring-your-own-license ^
    --master-username myadmin ^
    --manage-master-user-password
```

Example

Contoh berikut memodifikasi instance Oracle DB yang ada untuk mempublikasikan file log ke Log. CloudWatch Nilai `--cloudwatch-logs-export-configuration` adalah objek JSON. Kunci untuk objek ini adalah `EnableLogTypes`, dan nilainya adalah serangkaian string dengan setiap kombinasi `alert`, `audit`, `listener`, dan `trace`.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":
["trace","alert","audit","listener","oemagent"]}'
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration EnableLogTypes=\"trace\",\"alert\",\"audit
\",\"listener\",\"oemagent\"
```

Example

Contoh berikut memodifikasi instans Oracle DB yang ada untuk menonaktifkan audit penerbitan dan file log pendengar ke Log. CloudWatch Nilai `--cloudwatch-logs-export-configuration` adalah objek JSON. Kunci untuk objek ini adalah `DisableLogTypes`, dan nilainya adalah serangkaian string dengan setiap kombinasi `alert`, `audit`, `listener`, dan `trace`.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"DisableLogTypes":["audit","listener"]}'
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration DisableLogTypes=\"audit\",\"listener\"
```

RDS API

Anda dapat menerbitkan log DB Oracle dengan RDS API. Anda dapat memanggil tindakan `ModifyDBInstance` dengan parameter berikut:

- `DBInstanceIdentifier`
- `CloudwatchLogsExportConfiguration`

> **Note**
>
> Perubahan pada parameter `CloudwatchLogsExportConfiguration` selalu diterapkan ke instans DB secara langsung. Oleh karena itu, parameter `ApplyImmediately` tidak memiliki dampak.

Anda juga dapat menerbitkan log Oracle dengan memanggil operasi RDS API berikut:

- `CreateDBInstance`
- `RestoreDBInstanceFromDBSnapshot`
- `RestoreDBInstanceFromS3`
- `RestoreDBInstanceToPointInTime`

Jalankan salah satu operasi RDS API ini dengan parameter berikut:

- `DBInstanceIdentifier`
- `EnableCloudwatchLogsExports`
- `Engine`
- `DBInstanceClass`

Parameter lain mungkin diperlukan bergantung pada operasi RDS yang Anda jalankan.

Metode sebelumnya untuk mengakses log peringatan dan log pendengar

Anda dapat melihat log peringatan menggunakan konsol Amazon RDS. Anda juga dapat menggunakan pernyataan SQL berikut untuk mengakses log peringatan.

```
SELECT message_text FROM alertlog;
```

Tampilan `listenerlog` berisi entri untuk Oracle Database versi 12.1.0.2 dan versi yang lebih lama. Untuk mengakses log pendengar untuk versi basis data ini, gunakan kueri berikut.

```
SELECT message_text FROM listenerlog;
```

Untuk Oracle Database versi 12.2.0.1 dan yang lebih baru, akses log listener menggunakan Amazon Logs. CloudWatch

> **Note**
>
> Oracle merotasi log peringatan dan pendengar jika melebihi 10 MB. Pada saat itu, log ini tidak tersedia dari tampilan Amazon RDS.

File log basis data RDS for PostgreSQL

RDS for PostgreSQL mencatat aktivitas basis data ke file log PostgreSQL default. Untuk instans DB PostgreSQL on-premise, pesan ini disimpan secara lokal di `log/postgresql.log`. Untuk instans DB RDS for PostgreSQL, file log tersedia di instans Amazon RDS. Selain itu, Anda harus menggunakan Konsol Amazon RDS untuk melihat atau mengunduh kontennya. Tingkat penglogan default menangkap kegagalan masuk, kesalahan server fatal, kebuntuan, dan kegagalan kueri.

Untuk informasi selengkapnya tentang cara menampilkan, mengunduh, dan melihat log basis data berbasis file, lihat Memantau file log Amazon RDS. Untuk mempelajari selengkapnya tentang log PostgreSQL, lihat Menggunakan log Amazon RDS dan Aurora PostgreSQL: Bagian 1 dan Menggunakan log Amazon RDS dan Aurora PostgreSQL: Bagian 2.

Selain log PostgreSQL standar yang dibahas dalam topik ini, RDS for PostgreSQL juga mendukung ekstensi PostgreSQL Audit (`pgAudit`). Sebagian besar industri dan lembaga pemerintah yang teregulasi perlu mempertahankan log audit atau jejak audit perubahan yang dibuat pada data untuk memenuhi persyaratan hukum. Untuk informasi tentang cara menginstal dan menggunakan pgAudit, lihat Menggunakan pgAudit untuk membuat log aktivitas basis data.

Topik

- Parameter yang memengaruhi perilaku pengelogan
- Mengaktifkan pengelogan kueri untuk instans DB RDS for PostgreSQL
- Menerbitkan log PostgreSQL ke Amazon Logs CloudWatch

Parameter yang memengaruhi perilaku pengelogan

Anda dapat menyesuaikan perilaku pengelogan untuk instans DB RDS for PostgreSQL dengan mengubah berbagai parameter. Dalam tabel berikut, Anda dapat menemukan parameter yang memengaruhi durasi penyimpanan log, waktu untuk memutar log, dan apakah akan menampilkan log sebagai format CSV (nilai yang dipisahkan koma). Anda juga dapat menemukan output teks yang dikirim ke STDERR, di antara pengaturan lainnya. Untuk mengubah pengaturan parameter yang dapat dimodifikasi, gunakan grup parameter DB kustom untuk Instans DB RDS for PostgreSQL. Untuk informasi selengkapnya, lihat Bekerja dengan grup parameter DB dalam instance DB. Seperti disebutkan dalam tabel, `log_line_prefix` tidak dapat diubah.

Parameter	Default	Deskripsi
log_destination	stderr	Menetapkan format output untuk log. Defaultnya adalah `stderr`, tetapi Anda juga dapat menentukan nilai dipisahkan koma (CSV) dengan menambahkan `csvlog` ke pengaturan. Lihat informasi yang lebih lengkap di Mengatur tujuan log (`stderr, csvlog`)
log_filename	postgresql.log.%Y-%m-%d-%H	Menentukan pola untuk nama file log. Selain default, parameter ini mendukung `postgresql.log.%Y-%m-%d` untuk pola nama file.
log_line_prefix	%t:%r:%u@%d:[%p]:	Mendefinisikan awalan untuk setiap baris log yang akan ditulis ke `stderr`, untuk mencatat waktu (%t), host jarak jauh (%r), pengguna (%u), basis data (%d), dan ID proses (%p). Anda tidak dapat mengubah parameter ini.
log_rotation_age	60	Menit setelah file log dirotasi secara otomatis. Anda dapat mengubah nilai ini menjadi antara 1 dan 1.440 menit. Untuk informasi selengkapnya, lihat Mengatur rotasi file log.
log_rotation_size	–	Ukuran (kB) di mana log dirotasi secara otomatis. Secara default, parameter ini tidak digunakan karena log dirotasi berdasarkan parameter `log_rotation_age` . Untuk mempelajari selengkapnya, lihat Mengatur rotasi file log.
rds.log_retention_period	4320	Log PostgreSQL yang lebih lama dari jumlah menit yang ditentukan dihapus. Nilai default 4320 menit menghapus file log setelah 3 hari. Untuk informasi selengkapnya, lihat Mengatur periode retensi log.

Untuk mengidentifikasi masalah aplikasi, Anda dapat mencari kegagalan kueri, kegagalan masuk, kebuntuan, dan kesalahan server fatal di log. Misalnya, anggaplah Anda mengonversi aplikasi lama dari Oracle ke Amazon RDS PostgreSQL, tetapi tidak semua kueri dikonversi dengan benar. Kueri yang salah format ini menghasilkan pesan kesalahan yang dapat Anda temukan di log untuk membantu mengidentifikasi masalah. Untuk informasi selengkapnya tentang pengelogan kueri, lihat Mengaktifkan pengelogan kueri untuk instans DB RDS for PostgreSQL.

Dalam topik berikut, Anda dapat menemukan informasi tentang cara mengatur berbagai parameter yang mengontrol detail dasar untuk log PostgreSQL Anda.

Topik

- Mengatur periode retensi log
- Mengatur rotasi file log
- Mengatur tujuan log (stderr, csvlog)
- Memahami parameter log_line_prefix

Mengatur periode retensi log

Parameter `rds.log_retention_period` menentukan berapa lama instans DB RDS for PostgreSQL menyimpan file log-nya. Pengaturan default-nya adalah 3 hari (4.320 menit), tetapi Anda dapat mengatur nilai ini ke mana saja dari 1 hari (1.440 menit) hingga 7 hari (10.080 menit). Pastikan bahwa instans DB RDS for PostgreSQL Anda memiliki penyimpanan yang cukup untuk menyimpan file log selama periode waktu tertentu.

Kami menyarankan agar log Anda dipublikasikan secara rutin ke Amazon CloudWatch Logs sehingga Anda dapat melihat dan menganalisis data sistem lama setelah log dihapus dari cluster DB Aurora Anda. Instans DB RDS for PostgreSQL. Untuk informasi selengkapnya, lihat Menerbitkan log PostgreSQL ke Amazon Logs CloudWatch . .

Mengatur rotasi file log

Amazon RDS membuat file log baru setiap jam secara default. Waktu dikendalikan oleh parameter `log_rotation_age`. Parameter ini memiliki nilai default 60 (menit), tetapi Anda dapat mengaturnya ke mana saja dari 1 menit hingga 24 jam (1.440 menit). Ketika tiba waktunya rotasi, file log baru yang berbeda akan dibuat. File ini diberi nama sesuai dengan pola yang ditentukan oleh parameter `log_filename`.

File log juga dapat dirotasi sesuai dengan ukurannya, seperti yang ditentukan dalam parameter `log_rotation_size`. Parameter ini menentukan bahwa log harus dirotasi saat mencapai ukuran yang ditentukan (dalam kilobyte). Untuk instans DB RDS for PostgreSQL, `log_rotation_size` tidak diatur, yaitu tidak ada nilai yang ditentukan. Namun, Anda dapat mengatur parameter dari 0-2097151 kB (kilobyte).

Nama file log didasarkan pada pola nama file yang ditentukan dalam parameter `log_filename`. Pengaturan yang tersedia untuk parameter ini adalah sebagai berikut:

- `postgresql.log.%Y-%m-%d` – Format default untuk nama file log. Termasuk tahun, bulan, dan tanggal dalam nama file log.
- `postgresql.log.%Y-%m-%d-%H` – Termasuk jam dalam format nama file log.

Untuk informasi selengkapnya, lihat `log_rotation_age` dan `log_rotation_size` dalam dokumentasi PostgreSQL.

Mengatur tujuan log (**`stderr`**, **`csvlog`**)

Secara default, Amazon RDS PostgreSQL menghasilkan log dalam format kesalahan standar (stderr). Format ini adalah pengaturan default untuk parameter `log_destination`. Setiap pesan diawali menggunakan pola yang ditentukan dalam parameter `log_line_prefix`. Untuk mengetahui informasi selengkapnya, lihat Memahami parameter log_line_prefix.

RDS for PostgreSQL juga dapat menghasilkan log dalam format `csvlog`. `csvlog` berguna untuk menganalisis data log sebagai data nilai yang dipisahkan koma (CSV). Misalnya, anggaplah Anda menggunakan ekstensi `log_fdw` untuk bekerja dengan log Anda sebagai tabel asing. Tabel asing yang dibuat pada file log `stderr` berisi satu kolom dengan data peristiwa log. Dengan menambahkan `csvlog` ke parameter `log_destination`, Anda mendapatkan file log dalam format CSV dengan demarkasi untuk beberapa kolom tabel asing. Anda kini dapat mengurutkan dan menganalisis log dengan lebih mudah. Untuk mempelajari cara menggunakan `log_fdw` dengan `csvlog`, lihat Menggunakan ekstensi log_fdw untuk mengakses log DB menggunakan SQL.

Jika Anda menentukan `csvlog` untuk parameter ini, perhatikan bahwa file `stderr` dan `csvlog` dihasilkan. Pastikan untuk memantau penyimpanan yang digunakan oleh log, dengan mempertimbangkan `rds.log_retention_period` dan pengaturan lain yang memengaruhi penyimpanan log dan omset. Menggunakan `stderr` dan `csvlog` lebih dari dua kali lipat penyimpanan yang digunakan oleh log.

Jika Anda menambahkan `csvlog` ke `log_destination` dan ingin kembali ke `stderr` sendiri, Anda perlu mengatur ulang parameter. Untuk melakukannya, buka Konsol Amazon RDS, lalu buka grup parameter DB untuk instans Anda. Pilih parameter `log_destination`, pilih Edit parameter, lalu pilih Atur ulang.

Untuk informasi selengkapnya tentang cara mengonfigurasi pengelogan, lihat Menggunakan log Amazon RDS dan Aurora PostgreSQL: Bagian 1.

Memahami parameter log_line_prefix

Format log `stderr` mengawali setiap pesan log dengan detail yang ditentukan oleh parameter `log_line_prefix`, sebagai berikut.

```
%t:%r:%u@%d:[%p]:t
```

Anda dapat mengubah pengaturan ini: Setiap entri log yang dikirim ke `stderr` berisi informasi berikut.

- `%t` – Waktu entri log
- `%r` – Alamat host jarak jauh
- `%u@%d` – Nama pengguna @ nama basis data
- `[%p]` – ID Proses jika tersedia

Mengaktifkan pengelogan kueri untuk instans DB RDS for PostgreSQL

Anda dapat mengumpulkan informasi yang lebih mendetail tentang aktivitas basis data, termasuk kueri, kueri yang menunggu kunci, titik pemeriksaan, dan banyak detail lainnya dengan mengatur beberapa parameter yang tercantum dalam tabel berikut. Topik ini berfokus pada kueri pengelogan.

Parameter	Default	Deskripsi
log_connections	–	Mencatat setiap koneksi yang berhasil.
log_disconnections	–	Mencatat akhir setiap sesi dan durasinya.
log_checkpoints	1	Mencatat setiap titik pemeriksaan.

Parameter	Default	Deskripsi
log_lock_waits	–	Mencatat waktu tunggu kunci yang panjang. Secara default, parameter ini tidak diatur.
log_min_duration_sample	–	(md) Menetapkan waktu eksekusi minimum yang jika terlampaui akan membuat sampel pernyataan dicatat. Ukuran sampel diatur menggunakan parameter `log_statement_sample_rate` .
log_min_duration_statement	–	Setiap pernyataan SQL yang berjalan setidaknya selama periode waktu tertentu atau lebih lama akan dicatat. Secara default, parameter ini tidak diatur. Mengaktifkan parameter ini dapat membantu Anda menemukan kueri yang belum dioptimalkan.
log_statement	–	Menetapkan jenis pernyataan yang dicatat. Secara default, parameter ini tidak diatur, tetapi Anda dapat mengubahnya ke `all`, `ddl`, atau `mod` untuk menentukan jenis pernyataan SQL yang ingin Anda catat. Jika menentukan apa pun selain `none` untuk parameter ini, Anda juga harus mengambil langkah-langkah tambahan untuk mencegah eksposur kata sandi dalam file log. Untuk informasi selengkapnya, lihat Mengurangi risiko eksposur kata sandi saat menggunakan pengelogan kueri.
log_statement_sample_rate	–	Persentase pernyataan melebihi waktu yang ditentukan dalam `log_min_duration_sample` untuk dicatat, yang dinyatakan sebagai nilai titik mengambang antara 0,0 dan 1,0.
log_statement_stats	–	Menulis statistik performa kumulatif ke log server.

Menggunakan pengelogan untuk menemukan kueri performa lambat

Anda dapat mencatat pernyataan dan kueri SQL untuk membantu menemukan kueri performa lambat. Anda mengaktifkan kemampuan ini dengan memodifikasi pengaturan di `log_statement` dan parameter `log_min_duration` seperti yang diuraikan dalam bagian ini. Sebelum mengaktifkan pengelogan kueri untuk instans DB RDS for PostgreSQL, Anda harus mengetahui kemungkinan eksposur kata sandi di dalam log dan cara mengurangi risiko ini. Untuk informasi selengkapnya, lihat Mengurangi risiko eksposur kata sandi saat menggunakan pengelogan kueri.

Berikut ini, Anda dapat menemukan informasi referensi tentang parameter `log_statement` dan `log_min_duration`.

log_statement

Parameter ini menentukan jenis pernyataan SQL yang harus dikirim ke log. Nilai default-nya adalah `none`. Jika Anda mengubah parameter ini ke `all`, `ddl`, atau `mod`, pastikan untuk menerapkan tindakan yang disarankan untuk mengurangi risiko eksposur kata sandi di dalam log. Untuk informasi selengkapnya, lihat Mengurangi risiko eksposur kata sandi saat menggunakan pengelogan kueri.

all

Mencatat semua pernyataan. Pengaturan ini direkomendasikan untuk tujuan debugging.

ddl

Mencatat semua pernyataan bahasa definisi data (DDL), seperti CREATE, ALTER, DROP, dan seterusnya.

mod

Mencatat semua pernyataan DDL dan pernyataan bahasa manipulasi data (DML), seperti INSERT, UPDATE, dan DELETE, yang mengubah data.

none

Tidak ada pernyataan SQL yang dicatat. Kami merekomendasikan pengaturan ini untuk menghindari risiko eksposur kata sandi di dalam log.

log_min_duration_statement

Setiap pernyataan SQL yang berjalan setidaknya selama periode waktu tertentu atau lebih lama akan dicatat. Secara default, parameter ini tidak diatur. Mengaktifkan parameter ini dapat membantu Anda menemukan kueri yang belum dioptimalkan.

–1–2147483647

Jumlah milidetik (md) runtime di mana pernyataan dicatat.

Untuk menyiapkan pengelogan kueri

Langkah-langkah ini mengasumsikan bahwa Instans DB RDS for PostgreSQL menggunakan grup parameter DB kustom.

1. Atur parameter `log_statement` ke `all`. Contoh berikut ini menunjukkan informasi yang ditulis ke file `postgresql.log` dengan pengaturan parameter ini.

```
2022-10-05 22:05:52 UTC:52.95.4.1(11335):postgres@labdb:[3639]:LOG: statement:
 SELECT feedback, s.sentiment,s.confidence
FROM support,aws_comprehend.detect_sentiment(feedback, 'en') s
ORDER BY s.confidence DESC;
2022-10-05 22:05:52 UTC:52.95.4.1(11335):postgres@labdb:[3639]:LOG: QUERY
 STATISTICS
2022-10-05 22:05:52 UTC:52.95.4.1(11335):postgres@labdb:[3639]:DETAIL: ! system
 usage stats:
! 0.017355 s user, 0.000000 s system, 0.168593 s elapsed
! [0.025146 s user, 0.000000 s system total]
! 36644 kB max resident size
! 0/8 [0/8] filesystem blocks in/out
! 0/733 [0/1364] page faults/reclaims, 0 [0] swaps
! 0 [0] signals rcvd, 0/0 [0/0] messages rcvd/sent
! 19/0 [27/0] voluntary/involuntary context switches
2022-10-05 22:05:52 UTC:52.95.4.1(11335):postgres@labdb:[3639]:STATEMENT: SELECT
 feedback, s.sentiment,s.confidence
FROM support,aws_comprehend.detect_sentiment(feedback, 'en') s
ORDER BY s.confidence DESC;
2022-10-05 22:05:56 UTC:52.95.4.1(11335):postgres@labdb:[3639]:ERROR: syntax error
 at or near "ORDER" at character 1
2022-10-05 22:05:56 UTC:52.95.4.1(11335):postgres@labdb:[3639]:STATEMENT: ORDER BY
 s.confidence DESC;
---------------------- END OF LOG ----------------------
```

2. Atur parameter `log_min_duration_statement`. Contoh berikut ini menunjukkan informasi yang ditulis ke file `postgresql.log` saat pengaturan parameter ini diatur ke `1`.

Kueri yang melebihi durasi yang ditentukan dalam parameter `log_min_duration_statement` dicatat. Bagian berikut menunjukkan satu contoh. Anda dapat melihat file log untuk instans DB RDS for PostgreSQL di Konsol Amazon RDS.

```
2022-10-05 19:05:19 UTC:52.95.4.1(6461):postgres@labdb:[6144]:LOG: statement: DROP
 table comments;
2022-10-05 19:05:19 UTC:52.95.4.1(6461):postgres@labdb:[6144]:LOG: duration:
 167.754 ms
2022-10-05 19:08:07 UTC::@:[355]:LOG: checkpoint starting: time
2022-10-05 19:08:08 UTC::@:[355]:LOG: checkpoint complete: wrote 11 buffers
 (0.0%); 0 WAL file(s) added, 0 removed, 0 recycled; write=1.013 s, sync=0.006 s,
 total=1.033 s; sync files=8, longest=0.004 s, average=0.001 s; distance=131028 kB,
 estimate=131028 kB
----------------------- END OF LOG ----------------------
```

Mengurangi risiko eksposur kata sandi saat menggunakan pengelogan kueri

Kami menyarankan agar Anda tetap mengatur `log_statement` ke `none` agar tidak mengekspos kata sandi. Jika Anda mengatur `log_statement` ke `all`, `ddl`, atau `mod`, sebaiknya Anda mengambil satu atau beberapa langkah berikut.

- Untuk klien, enkripsi informasi sensitif. Untuk informasi selengkapnya, lihat Opsi Enkripsi dalam dokumentasi PostgreSQL. Gunakan opsi `ENCRYPTED` (dan `UNENCRYPTED`) dari pernyataan `CREATE` dan `ALTER`. Untuk informasi selengkapnya, lihat CREATE USER dalam dokumentasi PostgreSQL.
- Untuk instans DB RDS for PostgreSQL, siapkan dan gunakan ekstensi PostgreSQL Audit (pgAudit). Ekstensi ini menyunting informasi sensitif dalam pernyataan CREATE dan ALTER yang dikirim ke log. Untuk informasi selengkapnya, lihat Menggunakan pgAudit untuk membuat log aktivitas basis data.
- Batasi akses ke CloudWatch log.
- Gunakan mekanisme autentikasi yang lebih kuat seperti IAM.

Menerbitkan log PostgreSQL ke Amazon Logs CloudWatch

Untuk menyimpan catatan log PostgreSQL Anda dalam penyimpanan yang sangat tahan lama, Anda dapat menggunakan Amazon Logs. CloudWatch Dengan CloudWatch Log, Anda juga dapat melakukan analisis real-time data log dan digunakan CloudWatch untuk melihat metrik dan membuat

alarm. Misalnya, jika Anda mengatur `log_statement` ke `ddl`, Anda dapat menyiapkan alarm untuk memperingatkan Anda setiap kali pernyataan DDL dijalankan. Anda dapat memilih agar log PostgreSQL Anda diunggah ke Log selama proses pembuatan RDS CloudWatch untuk instance PostgreSQL DB. Jika memilih untuk tidak mengunggah log pada saat itu, Anda nantinya dapat mengubah instans untuk mulai mengunggah log sejak saat itu. Dengan kata lain, log yang ada tidak diunggah. Hanya log baru yang diunggah saat dibuat pada RDS Anda yang dimodifikasi untuk instans DB PostgreSQL.

Semua RDS yang tersedia saat ini untuk versi PostgreSQL mendukung penerbitan file log ke Log. CloudWatch Untuk informasi selengkapnya, lihat Pembaruan Amazon RDS for PostgreSQL di Catatan Rilis Amazon RDS for PostgreSQL..

Untuk bekerja dengan CloudWatch Log, konfigurasikan RDS Anda untuk instans PostgreSQL DB untuk mempublikasikan data log ke grup log.

Anda dapat mempublikasikan jenis log berikut ke CloudWatch Log untuk RDS untuk PostgreSQL:

- Postgresql log
- Mengupgrade log

Setelah Anda menyelesaikan konfigurasi, Amazon RDS menerbitkan peristiwa log untuk mencatat aliran dalam grup log. CloudWatch Sebagai contoh, data log PostgreSQL disimpan dalam grup log `/aws/rds/instance/my_instance/postgresql`. Untuk melihat log Anda, buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.

Konsol

Untuk mempublikasikan log PostgreSQL ke Log menggunakan konsol CloudWatch

1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data.
3. Pilih instans DB yang ingin diubah, lalu pilih Ubah.
4. Di bagian Log ekspor, pilih log yang ingin Anda mulai terbitkan ke CloudWatch Log.

   Bagian ekspor Log hanya tersedia untuk versi PostgreSQL yang mendukung penerbitan ke Log. CloudWatch
5. Pilih Lanjutkan, lalu pilih Ubah Instans DB di halaman ringkasan.

AWS CLI

Anda dapat menerbitkan log PostgreSQL dengan AWS CLI. Anda dapat memanggil perintah `modify-db-instance` dengan parameter berikut:

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

> **Note**
>
> Perubahan pada opsi `--cloudwatch-logs-export-configuration` selalu diterapkan ke instans DB secara langsung. Oleh karena itu, opsi `--apply-immediately` dan `--no-apply-immediately` tidak akan berpengaruh.

Anda juga dapat menerbitkan log PostgreSQL dengan memanggil perintah CLI berikut:

- `create-db-instance`
- `restore-db-instance-from-db-snapshot`
- `restore-db-instance-to-point-in-time`

Jalankan salah satu perintah CLI ini dengan opsi berikut:

- `--db-instance-identifier`
- `--enable-cloudwatch-logs-exports`
- `--db-instance-class`
- `--engine`

Opsi lain mungkin diperlukan bergantung pada perintah CLI yang Anda jalankan.

Example Memodifikasi instance untuk mempublikasikan log ke CloudWatch Log

Contoh berikut memodifikasi instance PostgreSQL DB yang ada untuk mempublikasikan file log ke Log. CloudWatch Nilai `--cloudwatch-logs-export-configuration` adalah objek JSON. Kunci untuk objek ini adalah `EnableLogTypes`, dan nilainya adalah serangkaian string dengan kombinasi `upgrade` dan `postgresql` apa pun.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":["postgresql",
 "upgrade"]}'
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":
["postgresql","upgrade"]}'
```

Example Buat instance untuk mempublikasikan log ke CloudWatch Log

Contoh berikut membuat instance PostgreSQL DB dan menerbitkan file log ke Log. CloudWatch Nilai `--enable-cloudwatch-logs-exports` adalah rangkaian string JSON. String dapat berupa kombinasi `postgresql` dan `upgrade`.

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --enable-cloudwatch-logs-exports '["postgresql","upgrade"]' \
    --db-instance-class db.m4.large \
    --engine postgres
```

Untuk Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --enable-cloudwatch-logs-exports '["postgresql","upgrade"]' ^
    --db-instance-class db.m4.large ^
    --engine postgres
```

RDS API

Anda dapat menerbitkan log PostgreSQL dengan RDS API. Anda dapat memanggil tindakan `ModifyDBInstance` dengan parameter berikut:

- `DBInstanceIdentifier`
- `CloudwatchLogsExportConfiguration`

> **Note**
>
> Perubahan pada parameter `CloudwatchLogsExportConfiguration` selalu diterapkan ke instans DB secara langsung. Oleh karena itu, parameter `ApplyImmediately` tidak memiliki dampak.

Anda juga dapat menerbitkan log PostgreSQL dengan memanggil operasi RDS API berikut:

- `CreateDBInstance`
- `RestoreDBInstanceFromDBSnapshot`
- `RestoreDBInstanceToPointInTime`

Jalankan salah satu operasi RDS API ini dengan parameter berikut:

- `DBInstanceIdentifier`
- `EnableCloudwatchLogsExports`
- `Engine`
- `DBInstanceClass`

Parameter lain mungkin diperlukan bergantung pada operasi yang Anda jalankan.

Memantau panggilan API Amazon RDS di AWS CloudTrail

AWS CloudTrail adalah layanan AWS yang membantu Anda mengaudit akun AWS Anda. AWS CloudTraildiaktifkan untuk akun AWS Anda saat Anda membuatnya. Lihat informasi yang lebih lengkap tentang CloudTrail di Panduan Pengguna AWS CloudTrail.

Topik

Integrasi CloudTrail dengan Amazon RDS

Semua tindakan Amazon RDS dilog oleh CloudTrail. CloudTrail menyediakan rekam tindakan yang diambil oleh pengguna, peran, atau layanan AWS di Amazon RDS.

Peristiwa CloudTrail

CloudTrail menangkap panggilan API untuk Amazon RDS sebagai peristiwa. Sebuah peristiwa mewakili satu permintaan dari sumber apa pun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. Peristiwa mencakup panggilan dari konsol Amazon RDS dan dari panggilan kode ke operasi API Amazon RDS.

Aktivitas Amazon RDS direkam dalam peristiwa CloudTrail di Riwayat peristiwa. Anda dapat menggunakan konsol CloudTrail untuk melihat aktivitas dan peristiwa API yang direkam selama 90 hari terakhir di Kawasan AWS. Lihat informasi yang lebih lengkap di Melihat peristiwa dengan riwayat peristiwa CloudTrail.

Jejak CloudTrail

Untuk rekam berlanjut peristiwa di AWS akun Anda, yang meliputi peristiwa-peristiwa untuk Amazon RDS, buatlah jejak. Sebuah jejak adalah konfigurasi yang memungkinkan pengiriman peristiwa ke bucket Amazon S3 yang ditentukan. Biasanya, CloudTrail mengirimkan file log dalam waktu 15 menit dari aktivitas akun.

> **Note**
>
> Jika Anda tidak mengonfigurasikan jejak, Anda masih dapat melihat peristiwa terbaru dalam konsol CloudTrail dalam Riwayat peristiwa.

Anda dapat membuat dua jenis jejak untuk akun AWS: jejak yang berlaku untuk semua Kawasan, atau jejak yang berlaku untuk satu Kawasan. Secara bawaan, ketika Anda membuat jejak di konsol, jejak itu berlaku untuk semua Kawasan.

Selain itu, Anda dapat mengonfigurasikan layanan AWS lainnya untuk menganalisis lebih lanjut dan menindaki data peristiwa yang terkumpul di log CloudTrail. Lihat informasi yang lebih lengkap di:

- Ikhtisar untuk membuat jejak
- Layanan dan integrasi yang didukung CloudTrail
- Mengonfigurasikan notifikasi Amazon SNS untuk CloudTrail
- Menerima file log CloudTrail dari beberapa Kawasan dan Menerima file log CloudTrail dari beberapa akun

Entri file log Amazon RDS

File log CloudTrail berisi satu atau beberapa entri log. File log CloudTrail bukan sebuah jejak tumpukan terurut panggilan API publik, sehingga file itu tidak muncul dengan urutan tertentu.

Contoh berikut menunjukkan entri log CloudTrail yang memperlihatkan tindakan `CreateDBInstance`.

```
{
    "eventVersion": "1.04",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AKIAIOSFODNN7EXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/johndoe",
        "accountId": "123456789012",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "userName": "johndoe"
    },
    "eventTime": "2018-07-30T22:14:06Z",
    "eventSource": "rds.amazonaws.com",
    "eventName": "CreateDBInstance",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "aws-cli/1.15.42 Python/3.6.1 Darwin/17.7.0 botocore/1.10.42",
    "requestParameters": {
        "enableCloudwatchLogsExports": [
```

```
            "audit",
            "error",
            "general",
            "slowquery"
        ],
        "dBInstanceIdentifier": "test-instance",
        "engine": "mysql",
        "masterUsername": "myawsuser",
        "allocatedStorage": 20,
        "dBInstanceClass": "db.m1.small",
        "masterUserPassword": "****"
    },
    "responseElements": {
        "dBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance",
        "storageEncrypted": false,
        "preferredBackupWindow": "10:27-10:57",
        "preferredMaintenanceWindow": "sat:05:47-sat:06:17",
        "backupRetentionPeriod": 1,
        "allocatedStorage": 20,
        "storageType": "standard",
        "engineVersion": "8.0.28",
        "dbInstancePort": 0,
        "optionGroupMemberships": [
            {
                "status": "in-sync",
                "optionGroupName": "default:mysql-8-0"
            }
        ],
        "dBParameterGroups": [
            {
                "dBParameterGroupName": "default.mysql8.0",
                "parameterApplyStatus": "in-sync"
            }
        ],
        "monitoringInterval": 0,
        "dBInstanceClass": "db.m1.small",
        "readReplicaDBInstanceIdentifiers": [],
        "dBSubnetGroup": {
            "dBSubnetGroupName": "default",
            "dBSubnetGroupDescription": "default",
            "subnets": [
                {
                    "subnetAvailabilityZone": {"name": "us-east-1b"},
                    "subnetIdentifier": "subnet-cbfff283",
```

```
                    "subnetStatus": "Active"
                },
                {
                    "subnetAvailabilityZone": {"name": "us-east-1e"},
                    "subnetIdentifier": "subnet-d7c825e8",
                    "subnetStatus": "Active"
                },
                {
                    "subnetAvailabilityZone": {"name": "us-east-1f"},
                    "subnetIdentifier": "subnet-6746046b",
                    "subnetStatus": "Active"
                },
                {
                    "subnetAvailabilityZone": {"name": "us-east-1c"},
                    "subnetIdentifier": "subnet-bac383e0",
                    "subnetStatus": "Active"
                },
                {
                    "subnetAvailabilityZone": {"name": "us-east-1d"},
                    "subnetIdentifier": "subnet-42599426",
                    "subnetStatus": "Active"
                },
                {
                    "subnetAvailabilityZone": {"name": "us-east-1a"},
                    "subnetIdentifier": "subnet-da327bf6",
                    "subnetStatus": "Active"
                }
            ],
            "vpcId": "vpc-136a4c6a",
            "subnetGroupStatus": "Complete"
        },
        "masterUsername": "myawsuser",
        "multiAZ": false,
        "autoMinorVersionUpgrade": true,
        "engine": "mysql",
        "cACertificateIdentifier": "rds-ca-2015",
        "dbiResourceId": "db-ETDZIIXHEWY5N7GXVC4SH7H5IA",
        "dBSecurityGroups": [],
        "pendingModifiedValues": {
            "masterUserPassword": "****",
            "pendingCloudwatchLogsExports": {
                "logTypesToEnable": [
                    "audit",
                    "error",
```

```
                    "general",
                    "slowquery"
                ]
            }
        },
        "dBInstanceStatus": "creating",
        "publiclyAccessible": true,
        "domainMemberships": [],
        "copyTagsToSnapshot": false,
        "dBInstanceIdentifier": "test-instance",
        "licenseModel": "general-public-license",
        "iAMDatabaseAuthenticationEnabled": false,
        "performanceInsightsEnabled": false,
        "vpcSecurityGroups": [
            {
                "status": "active",
                "vpcSecurityGroupId": "sg-f839b688"
            }
        ]
    },
    "requestID": "daf2e3f5-96a3-4df7-a026-863f96db793e",
    "eventID": "797163d3-5726-441d-80a7-6eeb7464acd4",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}
```

Seperti ditunjukkan pada elemen `userIdentity` dalam contoh sebelumnya, setiap entri peristiwa atau log berisi informasi tentang siapa yang membuat permintaan. Informasi identitas membantu Anda menentukan hal-hal berikut:

- Apakah permintaan dibuat dengan kredensial akar atau pengguna IAM.
- Apakah permintaan dibuat dengan kredensial keamanan sementara untuk suatu peran atau pengguna gabungan.
- Apakah permintaan dibuat oleh layanan AWS lain.

Lihat informasi yang lebih lengkap tentang `userIdentity` di Elemen userIdentity CloudTrail. Lihat informasi yang lebih lengkap tentang `CreateDBInstance` dan tindakan Amazon RDS lain di Referensi API Amazon RDS.

Memantau Amazon RDS dengan Aliran Aktivitas Basis Data

Dengan menggunakan Aliran Aktivitas Basis Data, Anda dapat memantau aliran aktivitas basis data secara waktu nyaris nyata.

Topik

Ikhtisar Aliran Aktivitas Basis Data

Sebagai seorang administrator basis data Amazon RDS, Anda perlu melindungi basis data Anda dan memenuhi persyaratan kepatuhan dan peraturan. Satu strateginya adalah mengintegrasikan aliran aktivitas basis data dengan alat pemantauan. Dengan cara ini, Anda memantau dan mengatur alarm untuk aktivitas pengauditan basis data klaster Anda.

Ancaman keamanan bersifat eksternal dan internal. Untuk melindungi terhadap ancaman internal, Anda dapat mengendalikan akses administrator ke aliran data dengan mengonfigurasikan fitur Aliran Aktivitas Basis Data. DBA Amazon RDS tidak memiliki akses ke pengumpulan, pengiriman, penyimpanan, dan pengolahan aliran.

Topik

- Kelas-kelas instans basis data yang didukung untuk aliran aktivitas basis data

Cara kerja aliran aktivitas basis data

Amazon RDS Anda mendorong aktivitas ke aliran data Amazon Kinesis dalam waktu nyaris nyata. Aliran Kinesis dibuat secara otomatis. Dari Kinesis, Anda dapat mengonfigurasi AWS layanan seperti Amazon Data Firehose dan AWS Lambda menggunakan aliran dan menyimpan data.

> **Important**
>
> Penggunaan fitur aliran aktivitas basis data di Amazon Aurora adalah gratis, tetapi Amazon Kinesis mengenakan biaya untuk aliran data. Lihat informasi yang lebih lengkap di struktur harga Amazon Kinesis Data Streams.

Anda dapat mengonfigurasikan aplikasi untuk pengelolaan kepatuhan agar menggunakan aliran aktivitas basis data. Aplikasi-aplikasi ini dapat menggunakan aliran untuk menghasilkan peringatan dan aktivitas audit pada basis data.

Amazon RDS mendukung aliran aktivitas basis data dalam deployment multi-AZ. Dalam hal ini, aliran aktivitas basis data mengaudit baik instans utama maupun instans siaga.

Pengauditan di Oracle Database dan Microsoft SQL Server Database

Pengauditan adalah pemantauan dan perekaman tindakan basis data yang dikonfigurasikan. Amazon RDS tidak menangkap aktivitas basis data secara bawaan. Anda membuat dan mengelola sendiri kebijakan audit di basis data Anda.

Topik

- Pengauditan terpadu di Oracle Database
- Pengauditan di Microsoft SQL Server
- Bidang-bidang audit non-asli untuk Oracle Database dan SQL Server
- Penindasan grup parameter basis data

Pengauditan terpadu di Oracle Database

Di sebuah basis data Oracle, kebijakan audit terpadu adalah grup setelan audit bernama yang dapat Anda gunakan untuk mengaudit sebuah aspek perilaku pengguna. Kebijakan dapat sesederhana

mengaudit aktivitas seorang pengguna. Anda juga dapat membuat kebijakan audit yang rumit yang bersyarat.

Basis data Oracle menulis rekam audit, yang meliputi rekam audit SYS, ke Jejak audit terpadu. Misalnya, jika terjadi kesalahan selama pernyataan INSERT, pengauditan standar menunjukkan nomor kesalahan dan SQL yang dijalankan. Jejak audit berada dalam sebuah tabel hanya baca dalam skema AUDSYS. Untuk mengakses rekam ini, lakukan kueri tampilan kamus data UNIFIED_AUDIT_TRAIL.

Biasanya, Anda mengonfigurasikan aliran aktivitas basis data sebagai berikut:

1. Buat kebijakan audit Oracle Database dengan menggunakan perintah CREATE AUDIT POLICY.

   Oracle Database menghasilkan rekam audit.
2. Aktifkan kebijakan audit dengan menggunakan perintah AUDIT POLICY.
3. Konfigurasikan aliran aktivitas basis data.

   Hanya aktivitas yang sesuai dengan kebijakan audit Oracle Database yang ditangkap dan dikirim ke aliran data Amazon Kinesis. Ketika aliran aktivitas basis data diaktifkan, administrator basis data Oracle tidak dapat mengubah kebijakan audit atau menghapus log audit.

Untuk mempelajari lebih lanjut kebijakan audit terpadu, lihat Tentang Aktivitas Pengauditan dengan Kebijakan Audit Terpadu dan AUDIT dalam Panduan Keamanan Oracle Database.

Pengauditan di Microsoft SQL Server

Aliran Aktivitas Basis Data menggunakan fitur SQLAudit untuk mengaudit basis data SQL Server.

Instans RDS for SQL Server berisi unsur-unsur berikut:

- Audit server – Audit server SQL mengumpulkan satu kejadian tindakan tingkat server atau basis data, dan sekelompok tindakan untuk dipantau. Audit-audit tingkat server RDS_DAS_AUDIT dan RDS_DAS_AUDIT_CHANGES dikelola oleh RDS.
- Spesifikasi audit server – Spesifikasi audit server merekam peristiwa tingkat server. Anda dapat mengubah spesifikasi RDS_DAS_SERVER_AUDIT_SPEC. Spesifikasi ini terkait dengan audit serverRDS_DAS_AUDIT. Spesifikasi RDS_DAS_CHANGES_AUDIT_SPEC dikelola oleh RDS.
- Spesifikasi audit basis data – Spesifikasi audit basis data merekam peristiwa tingkat basis data. Anda dapat membuat spesifikasi audit basis data RDS_DAS_DB_<name> dan menautkannya dengan audit server RDS_DAS_AUDIT.

Anda dapat mengonfigurasikan aliran aktivitas basis data dengan menggunakan konsol atau CLI. Biasanya, Anda mengonfigurasikan aliran aktivitas basis data sebagai berikut:

1. (Opsional) Buat spesifikasi audit basis data dengan perintah `CREATE DATABASE AUDIT SPECIFICATION` dan tautkan dengan audit server `RDS_DAS_AUDIT`.
2. (Opsional) Ubah spesifikasi audit server dengan perintah `ALTER SERVER AUDIT SPECIFICATION` dan tentukan kebijakan.
3. Aktifkan kebijakan audit basis data dan server. Sebagai contoh:

   ```
   ALTER DATABASE AUDIT SPECIFICATION [<Your database specification>] WITH (STATE=ON)

   ALTER SERVER AUDIT SPECIFICATION [RDS_DAS_SERVER_AUDIT_SPEC] WITH (STATE=ON)
   ```
4. Konfigurasikan aliran aktivitas basis data.

   Hanya aktivitas-aktivitas yang sesuai dengan kebijakan audit server dan basis data yang ditangkap dan dikirim ke aliran data Amazon Kinesis. Ketika aliran aktivitas basis data diaktifkan dan kebijakan dikunci, administrator basis data tidak dapat mengubah kebijakan audit atau menghapus log audit.

   > **Important**
   >
   > Jika spesifikasi audit basis data untuk basis data tertentu diaktifkan dan kebijakan dalam keadaan terkunci, maka basis data tidak dapat didrop.

Lihat informasi yang lebih lengkap tentang audit SQL Server di Komponen-Komponen Audit SQL Server dalam dokumentasi Microsoft SQL Server.

Bidang-bidang audit non-asli untuk Oracle Database dan SQL Server

Ketika Anda memulai aliran aktivitas basis data, setiap peristiwa basis data menghasilkan peristiwa aliran aktivitas yang terkait. Misalnya, pengguna basis data dapat menjalankan pernyataan-pernyataan `SELECT` dan `INSERT`. Basis data mengaudit kedua peristiwa ini dan mengirimkannya ke aliran data Amazon Kinesis.

Peristiwa-peristiwa disajikan dalam aliran sebagai objek JSON. Sebuah objek JSON berisi `DatabaseActivityMonitoringRecord`, yang berisi larik `databaseActivityEventList`. Bidang-bidang yang telah ditetapkan dalam larik mencakup `class`, `clientApplication`, dan`command`.

Secara bawaan, aliran aktivitas tidak mencakup bidang-bidang audit asli mesin. Anda dapat mengonfigurasikan Amazon RDS for Oracle dan SQL Server agar menyertakan bidang-bidang tambahan ini di dalam objek JSON `engineNativeAuditFields`.

Di Oracle Database, sebagian besar peristiwa dalam jejak audit terpadu memeta ke bidang-bidang dalam aliran aktivitas data RDS. Misalnya, bidang `UNIFIED_AUDIT_TRAIL.SQL_TEXT` dalam pengauditan terpadu memeta ke bidang `commandText` dalam aliran aktivitas basis data. Namun, bidang-bidang audit Oracle Database seperti `OS_USERNAME` tidak memeta ke bidang-bidang yang telah ditentukan dalam aliran aktivitas basis data.

Di SQL Server, sebagian besar bidang peristiwa yang direkam oleh SQLAudit memeta ke bidang-bidang dalam aliran aktivitas basis data RDS. Misalnya, bidang `code` dari `sys.fn_get_audit_file` dalam audit memeta ke bidang `commandText` dalam aliran aktivitas basis data. Namun, bidang-bidang audit basis data SQL Server seperti `permission_bitmask` tidak memeta ke bidang-bidang yang telah ditentukan dalam aliran aktivitas basis data.

Untuk informasi selengkapnya tentang databaseActivityEvent Daftar, lihatdatabaseActivityEventDaftar array JSON.

Penindasan grup parameter basis data

Biasanya, Anda mengaktifkan pengauditan terpadu di RDS for Oracle dengan melampirkan grup parameter. Namun, Aliran Aktivitas Basis Data memerlukan konfigurasi tambahan. Untuk meningkatkan pengalaman pelanggan Anda, Amazon RDS melakukan hal-hal berikut:

- Jika Anda mengaktifkan aliran aktivitas, RDS for Oracle mengabaikan parameter-parameter pengauditan dalam grup parameter.
- Jika Anda menonaktifkan aliran aktivitas, RDS for Oracle berhenti mengabaikan parameter-parameter pengauditan.

Aliran aktivitas basis data untuk SQL Server benar-benar independen terhadap parameter-parameter yang Anda tetapkan dalam opsi SQLAudit.

Mode asinkron untuk aliran aktivitas basis data

Aliran aktivitas di Amazon RDS selalu asinkron. Ketika sesi basis data menghasilkan peristiwa aliran aktivitas, sesi akan kembali dengan seketika ke aktivitas normal. Di latar belakang, Amazon RDS menjadikan peristiwa aliran aktivitas sebuah rekam yang awet.

Jika kesalahan terjadi dalam tugas latar belakang, Amazon RDS akan menghasilkan peristiwa. Peristiwa ini menunjukkan awal dan akhir segala jendela waktu ketika rekam peristiwa aliran aktivitas mungkin telah hilang. Mode asinkron lebih memprioritaskan kinerja basis data daripada akurasi aliran aktivitas.

Persyaratan dan keterbatasan untuk aliran aktivitas basis data

Dalam RDS, aliran aktivitas basis data memiliki persyaratan dan keterbatasan berikut:

- Amazon Kinesis diharuskan untuk aliran aktivitas basis data.
- AWS Key Management Service (AWS KMS) diperlukan untuk aliran aktivitas database karena selalu dienkripsi.
- Menerapkan enkripsi tambahan ke aliran data Amazon Kinesis Anda tidak kompatibel dengan aliran aktivitas database, yang sudah dienkripsi dengan kunci Anda. AWS KMS
- Anda membuat dan mengelola sendiri kebijakan audit. Berbeda dengan Amazon Aurora, RDS for Oracle tidak menangkap aktivitas basis data secara bawaan.
- Anda membuat dan mengelola sendiri kebijakan atau spesifikasi audit. Berbeda dengan Amazon Aurora, Amazon RDS tidak menangkap aktivitas basis data secara bawaan.
- Dalam deployment Multi-AZ, mulai aliran aktivitas basis data hanya pada instans basis data utama. Aliran aktivitas mengaudit secara otomatis baik instans basis data utama maupun instans basis data siaga. Tidak ada langkah tambahan yang diperlukan selama pindah saat gagal (failover).
- Mengganti nama instans basis data tidak membuat aliran Kinesis baru.
- CDB tidak didukung untuk RDS for Oracle.
- Replika baca tidak didukung.

Kawasan dan ketersediaan versi

Ketersediaan dan dukungan fitur bervariasi di berbagai versi khusus dari setiap mesin basis data, dan di seluruh Wilayah AWS. Lihat informasi yang lebih lengkap tentang ketersediaan versi dan Kawasan dengan aliran aktivitas basis data di Aliran aktivitas basis data.

Kelas-kelas instans basis data yang didukung untuk aliran aktivitas basis data

Untuk RDS for Oracle, Anda dapat menggunakan aliran aktivitas basis data dengan kelas-kelas instans basis data berikut:

- db.m4.*large
- db.m5.*large
- db.m5d.*large
- db.m6i.*large
- db.r4.*large
- db.r5.*large
- db.r5.*large.tpc*.mem*x
- db.r5b.*large
- db.r5b.*large.tpc*.mem*x
- db.r5d.*large
- db.r6i.*large
- db.x2idn.*large
- db.x2iedn.*large
- db.x2iezn.*large
- db.z1d.*large

Untuk RDS for SQL Server, Anda dapat menggunakan aliran aktivitas basis data dengan kelas-kelas instans basis data berikut:

- db.m4.*large
- db.m5.*large
- db.m5d.*large
- db.m6i.*large
- db.r4.*large
- db.r5.*large
- db.r5b.*large
- db.r5d.*large
- db.r6i.*large

- db.x1e.*large
- db.z1d.*large

Lihat informasi yang lebih lengkap tentang jenis-jenis kelas instans di Kelas instans DB .

Mengonfigurasikan pengauditan terpadu untuk basis data Oracle

Ketika Anda mengonfigurasikan pengauditan terpadu untuk digunakan dengan aliran aktivitas basis data, situasi-situasi berikut mungkin terjadi:

- Pengauditan terpadu tidak dikonfigurasikan untuk basis data Oracle Anda.

  Dalam hal ini, buat kebijakan-kebijakan baru dengan perintah `CREATE AUDIT POLICY`, lalu aktifkan kebijakan dengan perintah `AUDIT POLICY`. Contoh berikut membuat dan mengaktifkan sebuah kebijakan untuk memantau pengguna dengan privilese dan peran yang spesifik.

  ```
  CREATE AUDIT POLICY table_pol
  PRIVILEGES CREATE ANY TABLE, DROP ANY TABLE
  ROLES emp_admin, sales_admin;

  AUDIT POLICY table_pol;
  ```

  Lihat petunjuk lengkapnya di Mengonfigurasikan Kebijakan Audit dalam dokumentasi Oracle Database.
- Pengauditan terpadu dikonfigurasikan untuk basis data Oracle Anda.

  Bila Anda mengaktifkan suatu aliran aktivitas basis data, RDS for Oracle membersihkan secara otomatis data audit yang ada. Sistem juga mencabut privilese jejak audit. RDS for Oracle tidak dapat lagi melakukan hal-hal berikut:
  - Membersihkan rekam jejak audit terpadu.
  - Menambah, menghapus, atau mengubah kebijakan audit terpadu.
  - Memperbarui stempel waktu yang terakhir diarsipkan.

  > **Important**
  >
  > Kami sangat menyarankan supaya Anda membuat cadangan data audit sebelum mengaktifkan aliran aktivitas basis data.

Lihat deskripsi tampilan `UNIFIED_AUDIT_TRAIL` di UNIFIED_AUDIT_TRAIL. Jika Anda memiliki akun dengan Oracle Support, lihat Cara Membersihkan JEJAK AUDIT TERPADU.

Mengonfigurasikan kebijakan pengauditan untuk Microsoft SQL Server

Sebuah instans basis data SQL Server memiliki audit server `RDS_DAS_AUDIT`, yang dikelola oleh Amazon RDS. Anda dapat menentukan kebijakan untuk merekam peristiwa server dalam spesifikasi audit server `RDS_DAS_SERVER_AUDIT_SPEC`. Anda dapat membuat spesifikasi audit basis data, seperti `RDS_DAS_DB_<name>`, dan menentukan kebijakan untuk merekam peristiwa basis data. Lihat daftar grup tindakan audit tingkat server dan basis data di Grup Tindakan dan Tindakan Audit SQL Server dalam dokumentasi Microsoft SQL Server.

Kebijakan server bawaan memantau hanya upaya masuk yang gagal dan perubahan pada semua spesifikasi audit basis data atau server untuk aliran aktivitas basis data.

Keterbatasan untuk audit dan spesifikasi audit meliputi:

- Anda tidak dapat mengubah spesifikasi audit server atau basis data saat aliran aktivitas basis data dalam keadaan terkunci.
- Anda tidak dapat mengubah spesifikasi audit server `RDS_DAS_AUDIT`.
- Anda tidak dapat mengubah audit SQL Server `RDS_DAS_CHANGES` atau spesifikasi audit server terkaitnya `RDS_DAS_CHANGES_AUDIT_SPEC`.
- Saat membuat spesifikasi audit basis data, Anda harus menggunakan format `RDS_DAS_DB_<name>`, misalnya, `RDS_DAS_DB_databaseActions`.

> **Important**
>
> Untuk kelas instans yang lebih kecil, sebaiknya jangan audit semua melainkan hanya data yang diperlukan. Hal ini membantu mengurangi dampak kinerja Aliran Aktivitas Basis Data pada kelas instans ini.

Kode contoh berikut mengubah spesifikasi audit server `RDS_DAS_SERVER_AUDIT_SPEC` dan mengaudit semua tindakan masuk yang berhasil dan tindakan keluar:

```
ALTER SERVER AUDIT SPECIFICATION [RDS_DAS_SERVER_AUDIT_SPEC]
```

```
      WITH (STATE=OFF);
ALTER SERVER AUDIT SPECIFICATION [RDS_DAS_SERVER_AUDIT_SPEC]
      ADD (LOGOUT_GROUP),
      ADD (SUCCESSFUL_LOGIN_GROUP)
      WITH (STATE = ON );
```

Kode contoh berikut membuat spesifikasi audit basis data `RDS_DAS_DB_database_spec` dan melampirkannya pada audit server `RDS_DAS_AUDIT`:

```
USE testDB;
CREATE DATABASE AUDIT SPECIFICATION [RDS_DAS_DB_database_spec]
     FOR SERVER AUDIT [RDS_DAS_AUDIT]
     ADD ( INSERT, UPDATE, DELETE
          ON testTable BY testUser )
     WITH (STATE = ON);
```

Setelah spesifikasi audit dikonfigurasikan, pastikan bahwa spesifikasi-spesifikasi `RDS_DAS_SERVER_AUDIT_SPEC` dan `RDS_DAS_DB_<name>` diatur ke keadaan `ON`. Kini, keduanya dapat mengirim data audit ke aliran aktivitas basis data Anda.

Memulai aliran aktivitas basis data

Ketika Anda memulai aliran aktivitas untuk instans basis data, setiap peristiwa aktivitas basis data yang Anda konfigurasikan dalam kebijakan audit akan menghasilkan peristiwa aliran aktivitas. Perintah-perintah SQL seperti `CONNECT` dan `SELECT` menghasilkan peristiwa akses. Perintah-perintah SQL seperti `CREATE` dan `INSERT` menghasilkan peristiwa perubahan.

> **Important**
>
> Mengaktifkan aliran aktivitas untuk instans basis data Oracle akan membersihkan data audit yang ada. Hal ini juga mencabut privilese jejak audit. Ketika aliran diaktifkan, RDS for Oracle tidak dapat lagi melakukan hal-hal berikut:
>
> - Membersihkan rekam jejak audit terpadu.
> - Menambah, menghapus, atau mengubah kebijakan audit terpadu.
> - Memperbarui stempel waktu yang terakhir diarsipkan.

Konsol

Untuk memulai aliran aktivitas basis data

1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data.
3. Pilih instans basis data Amazon RDS tempat Anda ingin memulai aliran aktivitas. Dalam deployment Multi-AZ, mulai aliran hanya pada instans utama. Aliran aktivitas mengaudit instans-instans utama dan siaga.
4. Untuk Tindakan, pilih Mulai aliran aktivitas.

   Jendela Mulai aliran aktivitas basis data activity: *nama* muncul, dengan *nama* adalah instans RDS Anda.
5. Masukkan setelan berikut:

   - Untuk AWS KMS key, pilih sebuah kunci dari daftar AWS KMS keys.

     Amazon RDS menggunakan kunci KMS untuk mengenkripsi kunci yang pada gilirannya mengenkripsi aktivitas basis data. Pilih kunci KMS selain kunci bawaan. Lihat informasi yang lebih lengkap tentang kunci enkripsi dan AWS KMS di Apakah AWS Key Management Service? dalam Panduan Pengembang AWS Key Management Service.
   - Untuk Peristiwa aktivitas basis data, pilih Aktifkan bidang-bidang audit asli mesin untuk menyertakan bidang-bidang audit khusus mesin.
   - Pilih Seketika.

     Bila Anda memilih Segera, instans RDS memulai ulang dengan seketika. Jika Anda memilih Selama jendela pemeliharaan berikutnya, instans RDS tidak seketika memulai ulang. Dalam hal ini, aliran aktivitas basis data tidak dimulai hingga jendela pemeliharaan berikutnya.
6. Pilih Mulai aliran aktivitas basis data.

   Status untuk basis data menunjukkan bahwa aliran aktivitas dimulai.

> **Note**
>
> Jika Anda mendapatkan kesalahan `You can't start a database activity stream in this configuration`, periksa Kelas-kelas instans basis data yang

didukung untuk aliran aktivitas basis data untuk melihat apakah instans RDS Anda menggunakan kelas instans yang didukung.

AWS CLI

Untuk memulai aliran aktivitas database untuk instans DB, database menggunakan start-activity-streamAWS CLIperintah.

- `--resource-arn` *arn* – Menentukan Amazon Resource Name (ARN) instans basis data.
- `--kms-key-id` *key* – Menentukan pengidentifikasi kunci KMS untuk mengenkripsi pesan dalam aliran aktivitas basis data. Pengidentifikasi kunci KMS AWS adalah ARN kunci, ID kunci, ARN alias, atau nama alias bagi AWS KMS key.
- `--engine-native-audit-fields-included` – Menyertakan bidang-bidang pengauditan khusus mesin dalam aliran data. Untuk mengecualikan bidang-bidang ini, pilih `--no-engine-native-audit-fields-included` (bawaan).

Contoh berikut memulai aliran aktivitas basis data untuk sebuah instans basis data dalam mode asinkron.

Untuk Linux, macOS, atau Unix:

```
aws rds start-activity-stream \
    --mode async \
    --kms-key-id my-kms-key-arn \
    --resource-arn my-instance-arn \
    --engine-native-audit-fields-included \
    --apply-immediately
```

Untuk Windows:

```
aws rds start-activity-stream ^
    --mode async ^
    --kms-key-id my-kms-key-arn ^
    --resource-arn my-instance-arn ^
    --engine-native-audit-fields-included ^
    --apply-immediately
```

API RDS

Untuk memulai aliran aktivitas database untuk , instans DB, instance menggunakan StartActivityStreamoperasi.

Panggil tindakan dengan parameter-parameter di bawah:

- `Region`
- `KmsKeyId`
- `ResourceArn`
- `Mode`
- `EngineNativeAuditFieldsIncluded`

Mengubah aliran aktivitas basis data

Anda mungkin ingin menyesuaikan kebijakan audit Amazon RDS saat aliran aktivitas dimulai. Jika tidak ingin kehilangan waktu dan data dengan menghentikan aliran aktivitas, Anda dapat mengubah keadaan kebijakan audit ke salah satu setelan berikut:

Terkunci (bawaan)

Kebijakan audit di basis data Anda bersifat hanya baca.

Tidak Terkunci

Kebijakan audit di basis data Anda bersifat baca/tulis.

Langkah-langkah dasarnya adalah sebagai berikut:

1. Ubah keadaan kebijakan audit ke tidak terkunci.
2. Sesuaikan kebijakan audit Anda.
3. Ubah keadaan kebijakan audit ke terkunci.

Konsol

Untuk mengubah keadaan kebijakan audit aliran aktivitas Anda

1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.

2. Di panel navigasi, pilih Basis data.
3. Untuk Tindakan, pilih Ubah aliran aktivitas basis data.

   Jendela Ubah aliran aktivitas basis data: ***nama*** muncul, dengan *nama* adalah instans RDS Anda.
4. Pilih salah satu opsi berikut:

   Terkunci

   Ketika kebijakan audit Anda kunci, kebijakan itu menjadi hanya baca. Anda tidak dapat mengedit kebijakan audit, kecuali kuncinya Anda buka atau aliran aktivitas Anda hentikan.

   Tidak Terkunci

   Ketika kebijakan audit Anda buka kuncinya, kebijakan itu menjadi baca/tulis. Anda dapat mengedit kebijakan audit selagi aliran aktivitas dimulai.
5. Pilih Ubah aliran aktivitas basis data.

   Status untuk basis data Amazon RDS menunjukkan Mengonfigurasikan aliran aktivitas.
6. (Opsional) Pilih penaut instans basis data. Lalu pilih tab Konfigurasi.

   Bidang Status kebijakan Audit menunjukkan salah satu nilai berikut:

   - Terkunci
   - Tidak Terkunci
   - Kebijakan penguncian
   - Kebijakan pembukaan kunci

AWS CLI

Untuk mengubah status aliran aktivitas untuk instance database, gunakan modify-activity-streamAWS CLIperintah.

Opsi	Wajib?	Deskripsi
`--resource-arn` *my-instance-ARN*	Ya	Amazon Resource Name (ARN) instans basis data RDS Anda.

Opsi	Wajib?	Deskripsi
`--audit-policy-state`	Tidak	Keadaan baru kebijakan audit untuk aliran aktivitas basis data pada instans Anda: `locked` atau `unlocked`.

Contoh berikut membuka kebijakan audit untuk aliran aktivitas yang dimulai pada *my-instance-ARN*.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-activity-stream \
    --resource-arn my-instance-ARN \
    --audit-policy-state unlocked
```

Untuk Windows:

```
aws rds modify-activity-stream ^
    --resource-arn my-instance-ARN ^
    --audit-policy-state unlocked
```

Contoh berikut menjelaskan instans *my-instance*. Contoh output sebagian menunjukkan bahwa kebijakan audit tidak terkunci.

```
aws rds describe-db-instances --db-instance-identifier my-instance

{
    "DBInstances": [
        {
            ...
            "Engine": "oracle-ee",
            ...
            "ActivityStreamStatus": "started",
            "ActivityStreamKmsKeyId": "ab12345e-1111-2bc3-12a3-ab1cd12345e",
            "ActivityStreamKinesisStreamName": "aws-rds-das-db-
AB1CDEFG23GHIJK4LMNOPQRST",
            "ActivityStreamMode": "async",
            "ActivityStreamEngineNativeAuditFieldsIncluded": true,
            "ActivityStreamPolicyStatus": "unlocked",
            ...
```

```
        }
    ]
}
```

API RDS

Untuk mengubah status kebijakan aliran aktivitas database Anda, gunakan ModifyActivityStreamoperasi.

Panggil tindakan dengan parameter-parameter di bawah:

- `AuditPolicyState`
- `ResourceArn`

Mendapatkan status aliran aktivitas basis data

Anda bisa mendapatkan status aliran aktivitas untuk instans basis data Amazon RDS Anda dengan menggunakan konsol atau AWS CLI.

Konsol

Untuk mendapatkan status aliran aktivitas basis data

1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data, lalu pilih penaut Instans basis data.
3. Pilih tab Konfigurasi, dan periksa Aliran aktivitas basis data untuk statusnya.

AWS CLI

Anda bisa mendapatkan konfigurasi aliran aktivitas untuk instans basis data sebagai respons terhadap permintaan CLI describe-db-instance.

Contoh berikut menjelaskan *`my-instance`*.

```
aws rds --region my-region describe-db-instances --db-instance-identifier my-db
```

Contoh berikut menunjukkan respons JSON. Bidang-bidang berikut ditampilkan:

- `ActivityStreamKinesisStreamName`

- `ActivityStreamKmsKeyId`
- `ActivityStreamStatus`
- `ActivityStreamMode`
- `ActivityStreamPolicyStatus`

```
{
    "DBInstances": [
        {
            ...
            "Engine": "oracle-ee",
            ...
            "ActivityStreamStatus": "starting",
            "ActivityStreamKmsKeyId": "ab12345e-1111-2bc3-12a3-ab1cd12345e",
            "ActivityStreamKinesisStreamName": "aws-rds-das-db-
AB1CDEFG23GHIJK4LMNOPQRST",
            "ActivityStreamMode": "async",
            "ActivityStreamEngineNativeAuditFieldsIncluded": true,
            "ActivityStreamPolicyStatus": locked",
            ...
        }
    ]
}
```

API RDS

Anda bisa mendapatkan konfigurasi aliran aktivitas untuk instans basis data sebagai respons operasi DescribeDBInstances.

Menghentikan aliran aktivitas basis data

Anda dapat menghentikan aliran aktivitas menggunakan konsol atau AWS CLI.

Jika Anda menghapus instans basis data Amazon RDS, aliran aktivitas dihentikan dan aliran Amazon Kinesis yang mendasari dihapus secara otomatis.

Konsol

Untuk mematikan aliran aktivitas

1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.

2. Di panel navigasi, pilih Database.
3. Pilih basis data yang ingin Anda hentikan aliran aktivitas basis datanya.
4. Untuk Tindakan, pilih Hentikan aliran aktivitas. Jendela Aliran Aktivitas Basis Data akan muncul.
    a. Pilih Seketika.

        Bila Anda memilih Segera, instans RDS memulai ulang dengan seketika. Jika Anda memilih Selama jendela pemeliharaan berikutnya, instans RDS tidak seketika memulai ulang. Dalam hal ini, aliran aktivitas basis data tidak berhenti hingga jendela pemeliharaan berikutnya.

    b. Pilih Lanjutkan.

AWS CLI

Untuk menghentikan aliran aktivitas database untuk database Anda, konfigurasikan instans DB menggunakan AWS CLI perintah stop-activity-stream. Tandai Kawasan AWS untuk instans basis data dengan menggunakan parameter `--region`. Parameter `--apply-immediately` bersifat opsional.

Untuk Linux, macOS, atau Unix:

```
aws rds --region MY_REGION \
    stop-activity-stream \
    --resource-arn MY_DB_ARN \
    --apply-immediately
```

Untuk Windows:

```
aws rds --region MY_REGION ^
    stop-activity-stream ^
    --resource-arn MY_DB_ARN ^
    --apply-immediately
```

API RDS

Untuk menghentikan aliran aktivitas database untuk database Anda, konfigurasikan instans DB menggunakan StopActivityStreamoperasi. Tandai Kawasan AWS untuk instans basis data dengan menggunakan parameter `Region`. Parameter `ApplyImmediately` bersifat opsional.

Memantau aliran aktivitas basis data

Aliran aktivitas basis data memantau dan melaporkan aktivitas. Aliran aktivitas dikumpulkan dan dikirim ke Amazon Kinesis. Dari Kinesis, Anda dapat memantau aliran aktivitas, atau layanan dan aplikasi lain dapat menggunakan aliran aktivitas untuk analisis lebih lanjut. Anda dapat menemukan nama aliran Kinesis yang mendasarinya dengan menggunakan AWS CLI perintah atau operasi API RDS.

Amazon RDS mengelola aliran Kinesis untuk Anda sebagai berikut:

- Amazon RDS membuat aliran Kinesis secara otomatis dengan periode retensi 24 jam.
- Amazon RDS menskalakan aliran Kinesis jika perlu.
- Jika Anda menghentikan aliran aktivitas basis data atau menghapus instans basis data, Amazon RDS menghapus aliran Kinesis.

Kategori-kategori aktivitas berikut dipantau dan dimasukkan ke dalam log audit aliran aktivitas:

- Perintah SQL – Semua perintah SQL diaudit, begitu pula dengan pernyataan yang disiapkan, fungsi bawaan, dan fungsi dalam PL/SQL. Panggilan ke prosedur tersimpan akan diaudit. Setiap pernyataan SQL yang diterbitkan di dalam prosedur atau fungsi tersimpan juga diaudit.
- Informasi basis data lainnya – Aktivitas yang dipantau mencakup pernyataan SQL lengkap, hitungan baris yang terpengaruh dari perintah DML, objek yang diakses, dan nama basis data unik. Aliran aktivitas basis data juga memantau variabel pengikatan dan parameter prosedur tersimpan.

  Important

  Teks SQL lengkap setiap pernyataan, yang meliputi semua data sensitif, dapat dilihat di log audit aliran aktivitas. Namun, kata sandi pengguna basis data disensor jika Oracle dapat memastikannya dari konteks, seperti dalam pernyataan SQL berikut.

  ```
  ALTER ROLE role-name WITH password
  ```

- Informasi koneksi – Aktivitas yang dipantau mencakup sesi dan informasi jaringan, ID proses server, dan kode keluar.

Jika aliran aktivitas mengalami kegagalan saat memantau instans basis data, Anda akan diberi tahu melalui peristiwa RDS.

Topik

Mengakses aliran aktivitas dari Kinesis

Saat Anda mengaktifkan aliran aktivitas untuk basis data, aliran Kinesis dibuat untuk Anda. Dari Kinesis, Anda dapat memantau aktivitas basis data Anda secara waktu nyata. Untuk menganalisis lebih lanjut aktivitas basis data, Anda dapat menghubungkan aliran Kinesis dengan aplikasi konsumen. Anda juga dapat menghubungkan aliran ke aplikasi manajemen kepatuhan seperti Database Imperva. SecureSphere

Anda dapat mengakses aliran Kinesis dari konsol RDS atau konsol Kinesis.

Untuk mengakses aliran aktivitas dari Kinesis dengan menggunakan konsol RDS

1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data.
3. Pilih instans basis data Amazon RDS tempat Anda memulai aliran aktivitas.
4. Pilih Konfigurasi.
5. Di bawah Aliran aktivitas basis data, pilih penaut di bawah Aliran Kinesis.
6. Di konsol Kinesis, pilih Pemantauan untuk mulai mengamati aktivitas basis data.

Untuk mengakses aliran aktivitas dari Kinesis dengan menggunakan konsol Kinesis

1. Buka konsol Kinesis di https://console.aws.amazon.com/kinesis.
2. Pilih aliran aktivitas Anda dari daftar aliran Kinesis.

   Nama aliran aktivitas mencakup awalan `aws-rds-das-db-` diikuti dengan ID sumber daya basis data. Berikut sebuah contohnya.

   ```
   aws-rds-das-db-NHVOV4PCLWHGF52NP
   ```

Untuk memakai konsol Amazon RDS guna menemukan ID sumber daya bagi basis data, pilih instans basis data Anda dari daftar basis data, lalu pilih tab Konfigurasi.

Untuk menggunakan AWS CLI untuk menemukan nama aliran Kinesis lengkap untuk aliran aktivitas, gunakan permintaan CLI dan catat nilai dalam respons. `ActivityStreamKinesisStreamName`

3. Pilih Pemantauan untuk mulai mengamati aktivitas basis data.

Lihat informasi yang lebih lengkap tentang penggunaan Amazon Kinesis di Apakah Amazon Kinesis Data Streams?.

Isi dan contoh log Audit

Peristiwa-peristiwa yang dipantau disajikan dalam aliran aktivitas basis data berupa string JSON. Strukturnya terdiri atas objek JSON yang berisi `DatabaseActivityMonitoringRecord`, yang selanjutnya berisi sebuah larik peristiwa aktivitas `databaseActivityEventList`.

Topik

- Contoh-contoh log audit untuk aliran aktivitas
- Objek JSON DatabaseActivityMonitoringRecords
- databaseActivityEvents Objek JSON

Contoh-contoh log audit untuk aliran aktivitas

Berikut adalah contoh log audit JSON terdekripsi dari rekam peristiwa aktivitas.

Example Rekam peristiwa aktivitas dari pernyataan SQL CONNECT

Rekam peristiwa aktivitas berikut menunjukkan upaya masuk dengan penggunaan pernyataan SQL `CONNECT` (`command`) oleh JDBC Thin Client (`clientApplication`) untuk basis data Oracle Anda.

```
{
    "class": "Standard",
    "clientApplication": "JDBC Thin Client",
    "command": "LOGON",
    "commandText": null,
    "dbid": "0123456789",
    "databaseName": "ORCL",
    "dbProtocol": "oracle",
```

```
    "dbUserName": "TEST",
    "endTime": null,
    "errorMessage": null,
    "exitCode": 0,
    "logTime": "2021-01-15 00:15:36.233787",
    "netProtocol": "tcp",
    "objectName": null,
    "objectType": null,
    "paramList": [],
    "pid": 17904,
    "remoteHost": "123.456.789.012",
    "remotePort": "25440",
    "rowCount": null,
    "serverHost": "987.654.321.098",
    "serverType": "oracle",
    "serverVersion": "19.0.0.0.ru-2020-01.rur-2020-01.r1.EE.3",
    "serviceName": "oracle-ee",
    "sessionId": 987654321,
    "startTime": null,
    "statementId": 1,
    "substatementId": null,
    "transactionId": "0000000000000000",
    "engineNativeAuditFields": {
        "UNIFIED_AUDIT_POLICIES": "TEST_POL_EVERYTHING",
        "FGA_POLICY_NAME": null,
        "DV_OBJECT_STATUS": null,
        "SYSTEM_PRIVILEGE_USED": "CREATE SESSION",
        "OLS_LABEL_COMPONENT_TYPE": null,
        "XS_SESSIONID": null,
        "ADDITIONAL_INFO": null,
        "INSTANCE_ID": 1,
        "DBID": 123456789
        "DV_COMMENT": null,
        "RMAN_SESSION_STAMP": null,
        "NEW_NAME": null,
        "DV_ACTION_NAME": null,
        "OLS_PROGRAM_UNIT_NAME": null,
        "OLS_STRING_LABEL": null,
        "RMAN_SESSION_RECID": null,
        "OBJECT_PRIVILEGES": null,
        "OLS_OLD_VALUE": null,
        "XS_TARGET_PRINCIPAL_NAME": null,
        "XS_NS_ATTRIBUTE": null,
        "XS_NS_NAME": null,
```

```
        "DBLINK_INFO": null,
        "AUTHENTICATION_TYPE": "(TYPE\u003d(DATABASE));(CLIENT ADDRESS\u003d((ADDRESS
\u003d(PROTOCOL\u003dtcp)(HOST\u003d205.251.233.183)(PORT\u003d25440))));",
        "OBJECT_EDITION": null,
        "OLS_PRIVILEGES_GRANTED": null,
        "EXCLUDED_USER": null,
        "DV_ACTION_OBJECT_NAME": null,
        "OLS_LABEL_COMPONENT_NAME": null,
        "EXCLUDED_SCHEMA": null,
        "DP_TEXT_PARAMETERS1": null,
        "XS_USER_NAME": null,
        "XS_ENABLED_ROLE": null,
        "XS_NS_ATTRIBUTE_NEW_VAL": null,
        "DIRECT_PATH_NUM_COLUMNS_LOADED": null,
        "AUDIT_OPTION": null,
        "DV_EXTENDED_ACTION_CODE": null,
        "XS_PACKAGE_NAME": null,
        "OLS_NEW_VALUE": null,
        "DV_RETURN_CODE": null,
        "XS_CALLBACK_EVENT_TYPE": null,
        "USERHOST": "a1b2c3d4e5f6.amazon.com",
        "GLOBAL_USERID": null,
        "CLIENT_IDENTIFIER": null,
        "RMAN_OPERATION": null,
        "TERMINAL": "unknown",
        "OS_USERNAME": "sumepate",
        "OLS_MAX_READ_LABEL": null,
        "XS_PROXY_USER_NAME": null,
        "XS_DATASEC_POLICY_NAME": null,
        "DV_FACTOR_CONTEXT": null,
        "OLS_MAX_WRITE_LABEL": null,
        "OLS_PARENT_GROUP_NAME": null,
        "EXCLUDED_OBJECT": null,
        "DV_RULE_SET_NAME": null,
        "EXTERNAL_USERID": null,
        "EXECUTION_ID": null,
        "ROLE": null,
        "PROXY_SESSIONID": 0,
        "DP_BOOLEAN_PARAMETERS1": null,
        "OLS_POLICY_NAME": null,
        "OLS_GRANTEE": null,
        "OLS_MIN_WRITE_LABEL": null,
        "APPLICATION_CONTEXTS": null,
        "XS_SCHEMA_NAME": null,
```

```
        "DV_GRANTEE": null,
        "XS_COOKIE": null,
        "DBPROXY_USERNAME": null,
        "DV_ACTION_CODE": null,
        "OLS_PRIVILEGES_USED": null,
        "RMAN_DEVICE_TYPE": null,
        "XS_NS_ATTRIBUTE_OLD_VAL": null,
        "TARGET_USER": null,
        "XS_ENTITY_TYPE": null,
        "ENTRY_ID": 1,
        "XS_PROCEDURE_NAME": null,
        "XS_INACTIVITY_TIMEOUT": null,
        "RMAN_OBJECT_TYPE": null,
        "SYSTEM_PRIVILEGE": null,
        "NEW_SCHEMA": null,
        "SCN": 5124715
    }
}
```

Rekam peristiwa aktivitas berikut menunjukkan kegagalan upaya masuk untuk basis data SQL Server Anda.

```
{
    "type": "DatabaseActivityMonitoringRecord",
    "clusterId": "",
    "instanceId": "db-4JCWQLUZVFYP7DIWP6JVQ77O3Q",
    "databaseActivityEventList": [
        {
            "class": "LOGIN",
            "clientApplication": "Microsoft SQL Server Management Studio",
            "command": "LOGIN FAILED",
            "commandText": "Login failed for user 'test'. Reason: Password did not
 match that for the login provided. [CLIENT: local-machine]",
            "databaseName": "",
            "dbProtocol": "SQLSERVER",
            "dbUserName": "test",
            "endTime": null,
            "errorMessage": null,
            "exitCode": 0,
            "logTime": "2022-10-06 21:34:42.7113072+00",
            "netProtocol": null,
            "objectName": "",
            "objectType": "LOGIN",
```

```
            "paramList": null,
            "pid": null,
            "remoteHost": "local machine",
            "remotePort": null,
            "rowCount": 0,
            "serverHost": "172.31.30.159",
            "serverType": "SQLSERVER",
            "serverVersion": "15.00.4073.23.v1.R1",
            "serviceName": "sqlserver-ee",
            "sessionId": 0,
            "startTime": null,
            "statementId": "0x1eb0d1808d34a94b9d3dcf5432750f02",
            "substatementId": 1,
            "transactionId": "0",
            "type": "record",
            "engineNativeAuditFields": {
                "target_database_principal_id": 0,
                "target_server_principal_id": 0,
                "target_database_principal_name": "",
                "server_principal_id": 0,
                "user_defined_information": "",
                "response_rows": 0,
                "database_principal_name": "",
                "target_server_principal_name": "",
                "schema_name": "",
                "is_column_permission": false,
                "object_id": 0,
                "server_instance_name": "EC2AMAZ-NFUJJNO",
                "target_server_principal_sid": null,
                "additional_information": "<action_info "xmlns=\"http://
schemas.microsoft.com/sqlserver/2008/sqlaudit_data\"><pooled_connection>0</
pooled_connection><error>0x00004818</error><state>8</state><address>local machine</
address><PasswordFirstNibbleHash>B</PasswordFirstNibbleHash></action_info>"-->,
                "duration_milliseconds": 0,
                "permission_bitmask": "0x00000000000000000000000000000000",
                "data_sensitivity_information": "",
                "session_server_principal_name": "",
                "connection_id": "98B4F537-0F82-49E3-AB08-B9D33B5893EF",
                "audit_schema_version": 1,
                "database_principal_id": 0,
                "server_principal_sid": null,
                "user_defined_event_id": 0,
                "host_name": "EC2AMAZ-NFUJJNO"
            }
```

```
        }
    ]
}
```

> **Note**
>
> Jika aliran aktivitas basis data tidak diaktifkan, maka bidang terakhir dalam dokumen JSON adalah `"engineNativeAuditFields": { }`.

Example Rekam peristiwa aktivitas pernyataan CREATE TABLE

Contoh berikut menunjukkan peristiwa `CREATE TABLE` untuk basis data Oracle Anda.

```
{
    "class": "Standard",
    "clientApplication": "sqlplus@ip-12-34-5-678 (TNS V1-V3)",
    "command": "CREATE TABLE",
    "commandText": "CREATE TABLE persons(\n    person_id NUMBER GENERATED BY DEFAULT AS
 IDENTITY,\n    first_name VARCHAR2(50) NOT NULL,\n    last_name VARCHAR2(50) NOT NULL,
\n    PRIMARY KEY(person_id)\n)",
    "dbid": "0123456789",
    "databaseName": "ORCL",
    "dbProtocol": "oracle",
    "dbUserName": "TEST",
    "endTime": null,
    "errorMessage": null,
    "exitCode": 0,
    "logTime": "2021-01-15 00:22:49.535239",
    "netProtocol": "beq",
    "objectName": "PERSONS",
    "objectType": "TEST",
    "paramList": [],
    "pid": 17687,
    "remoteHost": "123.456.789.0",
    "remotePort": null,
    "rowCount": null,
    "serverHost": "987.654.321.01",
    "serverType": "oracle",
    "serverVersion": "19.0.0.0.ru-2020-01.rur-2020-01.r1.EE.3",
    "serviceName": "oracle-ee",
    "sessionId": 1234567890,
```

```
    "startTime": null,
    "statementId": 43,
    "substatementId": null,
    "transactionId": "090011007F0D0000",
    "engineNativeAuditFields": {
        "UNIFIED_AUDIT_POLICIES": "TEST_POL_EVERYTHING",
        "FGA_POLICY_NAME": null,
        "DV_OBJECT_STATUS": null,
        "SYSTEM_PRIVILEGE_USED": "CREATE SEQUENCE, CREATE TABLE",
        "OLS_LABEL_COMPONENT_TYPE": null,
        "XS_SESSIONID": null,
        "ADDITIONAL_INFO": null,
        "INSTANCE_ID": 1,
        "DV_COMMENT": null,
        "RMAN_SESSION_STAMP": null,
        "NEW_NAME": null,
        "DV_ACTION_NAME": null,
        "OLS_PROGRAM_UNIT_NAME": null,
        "OLS_STRING_LABEL": null,
        "RMAN_SESSION_RECID": null,
        "OBJECT_PRIVILEGES": null,
        "OLS_OLD_VALUE": null,
        "XS_TARGET_PRINCIPAL_NAME": null,
        "XS_NS_ATTRIBUTE": null,
        "XS_NS_NAME": null,
        "DBLINK_INFO": null,
        "AUTHENTICATION_TYPE": "(TYPE\u003d(DATABASE));(CLIENT ADDRESS\u003d((PROTOCOL
\u003dbeq)(HOST\u003d123.456.789.0)));",
        "OBJECT_EDITION": null,
        "OLS_PRIVILEGES_GRANTED": null,
        "EXCLUDED_USER": null,
        "DV_ACTION_OBJECT_NAME": null,
        "OLS_LABEL_COMPONENT_NAME": null,
        "EXCLUDED_SCHEMA": null,
        "DP_TEXT_PARAMETERS1": null,
        "XS_USER_NAME": null,
        "XS_ENABLED_ROLE": null,
        "XS_NS_ATTRIBUTE_NEW_VAL": null,
        "DIRECT_PATH_NUM_COLUMNS_LOADED": null,
        "AUDIT_OPTION": null,
        "DV_EXTENDED_ACTION_CODE": null,
        "XS_PACKAGE_NAME": null,
        "OLS_NEW_VALUE": null,
        "DV_RETURN_CODE": null,
```

```
        "XS_CALLBACK_EVENT_TYPE": null,
        "USERHOST": "ip-10-13-0-122",
        "GLOBAL_USERID": null,
        "CLIENT_IDENTIFIER": null,
        "RMAN_OPERATION": null,
        "TERMINAL": "pts/1",
        "OS_USERNAME": "rdsdb",
        "OLS_MAX_READ_LABEL": null,
        "XS_PROXY_USER_NAME": null,
        "XS_DATASEC_POLICY_NAME": null,
        "DV_FACTOR_CONTEXT": null,
        "OLS_MAX_WRITE_LABEL": null,
        "OLS_PARENT_GROUP_NAME": null,
        "EXCLUDED_OBJECT": null,
        "DV_RULE_SET_NAME": null,
        "EXTERNAL_USERID": null,
        "EXECUTION_ID": null,
        "ROLE": null,
        "PROXY_SESSIONID": 0,
        "DP_BOOLEAN_PARAMETERS1": null,
        "OLS_POLICY_NAME": null,
        "OLS_GRANTEE": null,
        "OLS_MIN_WRITE_LABEL": null,
        "APPLICATION_CONTEXTS": null,
        "XS_SCHEMA_NAME": null,
        "DV_GRANTEE": null,
        "XS_COOKIE": null,
        "DBPROXY_USERNAME": null,
        "DV_ACTION_CODE": null,
        "OLS_PRIVILEGES_USED": null,
        "RMAN_DEVICE_TYPE": null,
        "XS_NS_ATTRIBUTE_OLD_VAL": null,
        "TARGET_USER": null,
        "XS_ENTITY_TYPE": null,
        "ENTRY_ID": 12,
        "XS_PROCEDURE_NAME": null,
        "XS_INACTIVITY_TIMEOUT": null,
        "RMAN_OBJECT_TYPE": null,
        "SYSTEM_PRIVILEGE": null,
        "NEW_SCHEMA": null,
        "SCN": 5133083
    }
}
```

Contoh berikut menunjukkan peristiwa `CREATE TABLE` untuk basis data SQL Server Anda.


```
{
    "type": "DatabaseActivityMonitoringRecord",
    "clusterId": "",
    "instanceId": "db-4JCWQLUZVFYP7DIWP6JVQ77O3Q",
    "databaseActivityEventList": [
        {
            "class": "SCHEMA",
            "clientApplication": "Microsoft SQL Server Management Studio - Query",
            "command": "ALTER",
            "commandText": "Create table [testDB].[dbo].[TestTable2](\r\ntextA
 varchar(6000),\r\n    textB varchar(6000)\r\n)",
            "databaseName": "testDB",
            "dbProtocol": "SQLSERVER",
            "dbUserName": "test",
            "endTime": null,
            "errorMessage": null,
            "exitCode": 1,
            "logTime": "2022-10-06 21:44:38.4120677+00",
            "netProtocol": null,
            "objectName": "dbo",
            "objectType": "SCHEMA",
            "paramList": null,
            "pid": null,
            "remoteHost": "local machine",
            "remotePort": null,
            "rowCount": 0,
            "serverHost": "172.31.30.159",
            "serverType": "SQLSERVER",
            "serverVersion": "15.00.4073.23.v1.R1",
            "serviceName": "sqlserver-ee",
            "sessionId": 84,
            "startTime": null,
            "statementId": "0x5178d33d56e95e419558b9607158a5bd",
            "substatementId": 1,
            "transactionId": "4561864",
            "type": "record",
            "engineNativeAuditFields": {
                "target_database_principal_id": 0,
                "target_server_principal_id": 0,
                "target_database_principal_name": "",
                "server_principal_id": 2,
                "user_defined_information": "",
```

```
                "response_rows": 0,
                "database_principal_name": "dbo",
                "target_server_principal_name": "",
                "schema_name": "",
                "is_column_permission": false,
                "object_id": 1,
                "server_instance_name": "EC2AMAZ-NFUJJNO",
                "target_server_principal_sid": null,
                "additional_information": "",
                "duration_milliseconds": 0,
                "permission_bitmask": "0x00000000000000000000000000000000",
                "data_sensitivity_information": "",
                "session_server_principal_name": "test",
                "connection_id": "EE1FE3FD-EF2C-41FD-AF45-9051E0CD983A",
                "audit_schema_version": 1,
                "database_principal_id": 1,
                "server_principal_sid":
 "0x0105000000000005150000000bdc2795e2d0717901ba6998cf4010000",
                "user_defined_event_id": 0,
                "host_name": "EC2AMAZ-NFUJJNO"
            }
        }
    ]
}
```

Example Rekam peristiwa aktivitas pernyataan SELECT

Contoh berikut menunjukkan peristiwa SELECT untuk basis data Oracle Anda.

```
{
    "class": "Standard",
    "clientApplication": "sqlplus@ip-12-34-5-678 (TNS V1-V3)",
    "command": "SELECT",
    "commandText": "select count(*) from persons",
    "databaseName": "1234567890",
    "dbProtocol": "oracle",
    "dbUserName": "TEST",
    "endTime": null,
    "errorMessage": null,
    "exitCode": 0,
    "logTime": "2021-01-15 00:25:18.850375",
    "netProtocol": "beq",
    "objectName": "PERSONS",
    "objectType": "TEST",
```

```
    "paramList": [],
    "pid": 17687,
    "remoteHost": "123.456.789.0",
    "remotePort": null,
    "rowCount": null,
    "serverHost": "987.654.321.09",
    "serverType": "oracle",
    "serverVersion": "19.0.0.0.ru-2020-01.rur-2020-01.r1.EE.3",
    "serviceName": "oracle-ee",
    "sessionId": 1080639707,
    "startTime": null,
    "statementId": 44,
    "substatementId": null,
    "transactionId": null,
    "engineNativeAuditFields": {
        "UNIFIED_AUDIT_POLICIES": "TEST_POL_EVERYTHING",
        "FGA_POLICY_NAME": null,
        "DV_OBJECT_STATUS": null,
        "SYSTEM_PRIVILEGE_USED": null,
        "OLS_LABEL_COMPONENT_TYPE": null,
        "XS_SESSIONID": null,
        "ADDITIONAL_INFO": null,
        "INSTANCE_ID": 1,
        "DV_COMMENT": null,
        "RMAN_SESSION_STAMP": null,
        "NEW_NAME": null,
        "DV_ACTION_NAME": null,
        "OLS_PROGRAM_UNIT_NAME": null,
        "OLS_STRING_LABEL": null,
        "RMAN_SESSION_RECID": null,
        "OBJECT_PRIVILEGES": null,
        "OLS_OLD_VALUE": null,
        "XS_TARGET_PRINCIPAL_NAME": null,
        "XS_NS_ATTRIBUTE": null,
        "XS_NS_NAME": null,
        "DBLINK_INFO": null,
        "AUTHENTICATION_TYPE": "(TYPE\u003d(DATABASE));(CLIENT ADDRESS\u003d((PROTOCOL
\u003dbeq)(HOST\u003d123.456.789.0)));",
        "OBJECT_EDITION": null,
        "OLS_PRIVILEGES_GRANTED": null,
        "EXCLUDED_USER": null,
        "DV_ACTION_OBJECT_NAME": null,
        "OLS_LABEL_COMPONENT_NAME": null,
        "EXCLUDED_SCHEMA": null,
```

```
        "DP_TEXT_PARAMETERS1": null,
        "XS_USER_NAME": null,
        "XS_ENABLED_ROLE": null,
        "XS_NS_ATTRIBUTE_NEW_VAL": null,
        "DIRECT_PATH_NUM_COLUMNS_LOADED": null,
        "AUDIT_OPTION": null,
        "DV_EXTENDED_ACTION_CODE": null,
        "XS_PACKAGE_NAME": null,
        "OLS_NEW_VALUE": null,
        "DV_RETURN_CODE": null,
        "XS_CALLBACK_EVENT_TYPE": null,
        "USERHOST": "ip-12-34-5-678",
        "GLOBAL_USERID": null,
        "CLIENT_IDENTIFIER": null,
        "RMAN_OPERATION": null,
        "TERMINAL": "pts/1",
        "OS_USERNAME": "rdsdb",
        "OLS_MAX_READ_LABEL": null,
        "XS_PROXY_USER_NAME": null,
        "XS_DATASEC_POLICY_NAME": null,
        "DV_FACTOR_CONTEXT": null,
        "OLS_MAX_WRITE_LABEL": null,
        "OLS_PARENT_GROUP_NAME": null,
        "EXCLUDED_OBJECT": null,
        "DV_RULE_SET_NAME": null,
        "EXTERNAL_USERID": null,
        "EXECUTION_ID": null,
        "ROLE": null,
        "PROXY_SESSIONID": 0,
        "DP_BOOLEAN_PARAMETERS1": null,
        "OLS_POLICY_NAME": null,
        "OLS_GRANTEE": null,
        "OLS_MIN_WRITE_LABEL": null,
        "APPLICATION_CONTEXTS": null,
        "XS_SCHEMA_NAME": null,
        "DV_GRANTEE": null,
        "XS_COOKIE": null,
        "DBPROXY_USERNAME": null,
        "DV_ACTION_CODE": null,
        "OLS_PRIVILEGES_USED": null,
        "RMAN_DEVICE_TYPE": null,
        "XS_NS_ATTRIBUTE_OLD_VAL": null,
        "TARGET_USER": null,
        "XS_ENTITY_TYPE": null,
```

```
        "ENTRY_ID": 13,
        "XS_PROCEDURE_NAME": null,
        "XS_INACTIVITY_TIMEOUT": null,
        "RMAN_OBJECT_TYPE": null,
        "SYSTEM_PRIVILEGE": null,
        "NEW_SCHEMA": null,
        "SCN": 5136972
    }
}
```

Contoh berikut menunjukkan peristiwa SELECT untuk basis data SQL Server Anda.

```
{
    "type": "DatabaseActivityMonitoringRecord",
    "clusterId": "",
    "instanceId": "db-4JCWQLUZVFYP7DIWP6JVQ77O3Q",
    "databaseActivityEventList": [
        {
            "class": "TABLE",
            "clientApplication": "Microsoft SQL Server Management Studio - Query",
            "command": "SELECT",
            "commandText": "select * from [testDB].[dbo].[TestTable]",
            "databaseName": "testDB",
            "dbProtocol": "SQLSERVER",
            "dbUserName": "test",
            "endTime": null,
            "errorMessage": null,
            "exitCode": 1,
            "logTime": "2022-10-06 21:24:59.9422268+00",
            "netProtocol": null,
            "objectName": "TestTable",
            "objectType": "TABLE",
            "paramList": null,
            "pid": null,
            "remoteHost": "local machine",
            "remotePort": null,
            "rowCount": 0,
            "serverHost": "172.31.30.159",
            "serverType": "SQLSERVER",
            "serverVersion": "15.00.4073.23.v1.R1",
            "serviceName": "sqlserver-ee",
            "sessionId": 62,
            "startTime": null,
```

```
            "statementId": "0x03baed90412f564fad640ebe51f89b99",
            "substatementId": 1,
            "transactionId": "4532935",
            "type": "record",
            "engineNativeAuditFields": {
                "target_database_principal_id": 0,
                "target_server_principal_id": 0,
                "target_database_principal_name": "",
                "server_principal_id": 2,
                "user_defined_information": "",
                "response_rows": 0,
                "database_principal_name": "dbo",
                "target_server_principal_name": "",
                "schema_name": "dbo",
                "is_column_permission": true,
                "object_id": 581577110,
                "server_instance_name": "EC2AMAZ-NFUJJNO",
                "target_server_principal_sid": null,
                "additional_information": "",
                "duration_milliseconds": 0,
                "permission_bitmask": "0x00000000000000000000000000000001",
                "data_sensitivity_information": "",
                "session_server_principal_name": "test",
                "connection_id": "AD3A5084-FB83-45C1-8334-E923459A8109",
                "audit_schema_version": 1,
                "database_principal_id": 1,
                "server_principal_sid":
 "0x0105000000000005150000000bdc2795e2d0717901ba6998cf4010000",
                "user_defined_event_id": 0,
                "host_name": "EC2AMAZ-NFUJJNO"
            }
        }
    ]
}
```


Objek JSON DatabaseActivityMonitoringRecords

Rekam peristiwa aktivitas basis data berada dalam objek JSON yang berisi informasi berikut.

Bidang JSON	Tipe data	Deskripsi
`type`	string	Jenis rekam JSON. Nilainya adalah `DatabaseActivityMonitoringRecords` .
`version`	string	Versi rekam pemantauan aktivitas basis data. Basis data Oracle menggunakan versi 1.3 dan SQL Server menggunakan versi 1.4. Versi-versi mesin ini memperkenalkan objek JSON `engineNativeAuditFields` .
databaseActivityEvents	string	Objek JSON yang berisi peristiwa aktivitas.
kunci	string	Kunci enkripsi yang Anda gunakan untuk mendekripsi databaseActivityEventDaftar

databaseActivityEvents Objek JSON

Objek JSON `databaseActivityEvents` berisi informasi berikut.

Bidang-bidang tingkat atas dalam rekam JSON

Setiap peristiwa dalam log audit dibungkus dalam sebuah rekam dalam format JSON. Rekam ini berisi bidang-bidang berikut.

tipe

Bidang ini selalu memiliki nilai `DatabaseActivityMonitoringRecords`.

versi

Bidang ini mewakili versi protokol atau kontrak data aliran aktivitas basis data. Versi menentukan bidang-bidang yang tersedia.

databaseActivityEvents

String terenkripsi yang mewakili satu atau beberapa peristiwa aktivitas. String disajikan berupa larik byte base64. Saat Anda mendekripsi string, hasilnya adalah rekam dalam format JSON dengan bidang-bidang seperti ditunjukkan dalam contoh di bagian ini.

kunci

Kunci data terenkripsi yang digunakan untuk mengenkripsi string `databaseActivityEvents`. Ini sama dengan AWS KMS key yang Anda berikan saat memulai aliran aktivitas database.

Contoh berikut menunjukkan format rekam ini.

```
{
  "type":"DatabaseActivityMonitoringRecords",
  "version":"1.3",
  "databaseActivityEvents":"encrypted audit records",
  "key":"encrypted key"
}
```

```
          "type":"DatabaseActivityMonitoringRecords",
          "version":"1.4",
          "databaseActivityEvents":"encrypted audit records",
          "key":"encrypted key"
```

Lakukan langkah-langkah berikut untuk mendekripsi isi bidang `databaseActivityEvents`:

1. Lakukan dekripsi nilai dalam bidang JSON `key` dengan menggunakan kunci KMS yang Anda sediakan ketika memulai aliran aktivitas basis data. Melakukan hal itu akan menghasilkan kunci enkripsi data berupa teks jelas.
2. Base64 mendekode nilai dalam bidang JSON `databaseActivityEvents` untuk mendapatkan teks sandi, dalam format biner, dari muatan audit.
3. Lakukan dekripsi teks sandi biner dengan kunci enkripsi data yang Anda dekode pada langkah pertama.
4. Lakukan dekompresi muatan yang terdekripsi.
   - Muatan terenkripsi ada di bidang `databaseActivityEvents`.
   - Bidang `databaseActivityEventList` berisi larik rekam audit. Bidang `type` dalam larik dapat berupa `record` atau `heartbeat`.

Rekam peristiwa aktivitas log audit adalah objek JSON yang berisi informasi berikut.

Bidang JSON	Tipe data	Deskripsi
`type`	string	Jenis rekam JSON. Nilainya adalah `DatabaseActivityMonitoringRecord` .
`instanceId`	string	Pengidentifikasi sumber daya instans basis data. Pengidentifikasi ini berkaitan dengan atribut instans basis data `DbiResourceId` .
databaseActivityEventDaftar	string	Larik rekam audit aktivitas atau pesan denyut jantung.

databaseActivityEventDaftar array JSON

Muatan log audit adalah larik JSON `databaseActivityEventList` terenkripsi. tabel berikut memerinci secara alfabetis bidang-bidang untuk setiap peristiwa aktivitas dalam larik `DatabaseActivityEventList` terdekripsi sebuah log audit.

Ketika pengauditan terpadu diaktifkan di Oracle Database, rekam audit diisi dalam jejak audit baru ini. Tampilan `UNIFIED_AUDIT_TRAIL` memperlihatkan rekam audit dalam bentuk tabel dengan mengambil rekam audit dari jejak audit. Ketika Anda memulai aliran aktivitas basis data, kolom di `UNIFIED_AUDIT_TRAIL` memeta ke sebuah bidang di dalam larik `databaseActivityEventList`.

> **Important**
>
> Struktur peristiwa dapat berubah sewaktu-waktu. Amazon RDS mungkin menambahkan bidang-bidang baru ke peristiwa aktivitas di masa mendatang. Dalam aplikasi yang menguraikan data JSON, pastikan bahwa kode Anda dapat mengabaikan atau mengambil tindakan yang tepat untuk nama-nama bidang yang tidak dikenal.

databaseActivityEventDaftar bidang untuk Amazon RDS for Oracle

Field	Tipe data	Sumber	Deskripsi
`class`	string	Kolom `AUDIT_TYPE` dalam `UNIFIED_AUDIT_TRAIL`	Kelas peristiwa aktivitas . Hal berkaitan dengan kolom `AUDIT_TYPE` dalam tampilan `UNIFIED_AUDIT_TRAIL` . Nilai valid untuk Amazon RDS for Oracle adalah sebagai berikut: • `Standard` • `FineGrainedAudit` • `XS` • `Database Vault` • `Label Security` • `RMAN_AUDIT` • `Datapump` • `Direct path API` Lihat informasi yang lebih lengkap di UNIFIED_AUDIT_TRAIL dalam dokumentasi Oracle.
`clientApplication`	string	`CLIENT_PROGRAM_NAME` di `UNIFIED_AUDIT_TRAIL`	Aplikasi yang digunakan klien untuk menghubungi seperti dilaporkan oleh klien. Klien tidak wajib memberikan informasi ini, sehingga nilainya dapat null. Nilai sampel adalah `JDBC Thin Client`.

Field	Tipe data	Sumber	Deskripsi
command	string	Kolom ACTION_NAME dalam UNIFIED_AUDIT_TRAIL	Nama tindakan yang dieksekusi oleh pengguna. Untuk memahami tindakan lengkap, baca baik nama perintah maupun nilai AUDIT_TYPE . Nilai sampel adalah ALTER DATABASE.
commandText	string	Kolom SQL_TEXT dalam UNIFIED_AUDIT_TRAIL	Pernyataan SQL yang terkait dengan peristiwa . Nilai sampel adalah ALTER DATABASE BEGIN BACKUP.
databaseName	string	Kolom NAME dalam V $DATABASE	Nama basis data.
dbid	nomo	Kolom DBID dalam UNIFIED_AUDIT_TRAIL	Pengidentifikasi numerik untuk basis data. Nilai sampel adalah 1559204751 .
dbProtocol	string	N/A	Protokol basis data. Dalam beta ini, nilainya adalah oracle.
dbUserName	string	Kolom DBUSERNAME dalam UNIFIED_AUDIT_TRAIL	Nama pengguna basis data yang tindakann ya diaudit. Nilai sampel adalah RDSADMIN.

Field	Tipe data	Sumber	Deskripsi
`endTime`	string	N/A	Bidang ini tidak digunakan untuk RDS for Oracle dan selalu null.

Field	Tipe data	Sumber	Deskripsi

`engineNativeAuditFields`	objek	`UNIFIED_AUDIT_TRAIL`	Secara bawaan, objek ini kosong. Ketika Anda memulai aliran aktivitas dengan opsi `--engine-native-audit-fields-included` , objek ini meliputi kolom-kolom berikut dan nilai-nil ainya:
			
			`ADDITIONAL_INFO`
			`APPLICATION_CONTEXTS`
			`AUDIT_OPTION`
			`AUTHENTICATION_TYPE`
			`CLIENT_IDENTIFIER`
			`CURRENT_USER`
			`DBLINK_INFO`
			`DBPROXY_USERNAME`
			`DIRECT_PATH_NUM_COLUMNS_LOADED`
			`DP_BOOLEAN_PARAMETERS1`
			`DP_TEXT_PARAMETERS1`
			`DV_ACTION_CODE`
			`DV_ACTION_NAME`
			`DV_ACTION_OBJECT_NAME`
			`DV_COMMENT`
			`DV_EXTENDED_ACTION_CODE`
			`DV_FACTOR_CONTEXT`
			`DV_GRANTEE`
			`DV_OBJECT_STATUS`
			`DV_RETURN_CODE`
			`DV_RULE_SET_NAME`

Field	Tipe data	Sumber	Deskripsi
			ENTRY_ID EXCLUDED_OBJECT EXCLUDED_SCHEMA EXCLUDED_USER EXECUTION_ID EXTERNAL_USERID FGA_POLICY_NAME GLOBAL_USERID INSTANCE_ID KSACL_SER VICE_NAME KSACL_SOURCE_LOCA TION KSACL_USER_NAME NEW_NAME NEW_SCHEMA OBJECT_EDITION OBJECT_PRIVILEGES OLS_GRANTEE OLS_LABEL_COM PONENT_NAME OLS_LABEL_COMPO NENT_TYPE OLS_MAX_READ_LABEL OLS_MAX_WRITE_LA BEL OLS_MIN_WRITE_ LABEL OLS_NEW_VALUE OLS_OLD_VALUE OLS_PARENT_ GROUP_NAME OLS_POLICY_NAME OLS_PRIVILEGES_GRA NTED OLS_PRIVILEGE S_USED OLS_PROGRAM _UNIT_NAME OLS_STRING_LABEL OS_USERNAME

Field	Tipe data	Sumber	Deskripsi
			PROTOCOL_ACTIO N_NAME PROTOCOL_MESSAGE PROTOCOL_RET URN_CODE PROTOCOL_SESSION_I D PROTOCOL_USERHOST PROXY_SESSIONID RLS_INFO RMAN_DEVICE_TYPE RMAN_OBJECT_TYPE RMAN_OPERATION RMAN_SESSION_RECID RMAN_SESSION_STAMP ROLE SCN SYSTEM_PRIVILEGE SYSTEM_PRIVIL EGE_USED TARGET_USER TERMINAL UNIFIED_AUDIT_P OLICIES USERHOST XS_CALLBAC K_EVENT_TYPE XS_COOKIE XS_DATASEC_PO LICY_NAME XS_ENABLED_ROLE XS_ENTITY_TYPE XS_INACTIVITY _TIMEOUT XS_NS_ATTRIBUTE XS_NS_ATTRI BUTE_NEW_VAL XS_NS_ATTRIBUT E_OLD_VAL XS_NS_NAME XS_PACKAGE_NAME

Field	Tipe data	Sumber	Deskripsi

			`XS_PROCEDURE_NAME`
			`XS_PROXY_USER_NAME`
			`XS_SCHEMA_NAME`
			`XS_SESSIONID`
			`XS_TARGET_PRINC IPAL_NAME`
			`XS_USER_NAME`
			
			Lihat informasi yang lebih lengkap di UNIFIED_AUDIT_TRAIL dalam dokumentasi Oracle Database.

`errorMessage`	string	N/A	Bidang ini tidak digunakan untuk RDS for Oracle dan selalu null.
`exitCode`	nomo	Kolom `RETURN_CODE` dalam `UNIFIED_AUDIT_TRAIL`	Kode kesalahan Oracle Database yang dihasilkan oleh tindakan. Jika tindakan berhasil, nilainya adalah `0`.
`logTime`	string	Kolom `EVENT_TIMESTAMP_UTC` dalam `UNIFIED_AUDIT_TRAIL`	Stempel waktu pembuatan entri jejak audit. Nilai sampel adalah `2020-11-27 06:56:14.981404` .
`netProtocol`	string	Kolom `AUTHENTICATION_TYPE` dalam `UNIFIED_AUDIT_TRAIL`	Protokol komunikasi jaringan. Nilai sampel adalah `TCP`.

Field	Tipe data	Sumber	Deskripsi
objectName	string	Kolom OBJECT_NAME dalam UNIFIED_AUDIT_TRAIL	Nama objek yang terpengaruh oleh tindakan. Nilai sampel adalah employees .
objectType	string	Kolom OBJECT_SCHEMA dalam UNIFIED_AUDIT_TRAIL	Nama skema dari objek yang terpengaruh oleh tindakan. Nilai sampel adalah hr.
paramList	daftar	Kolom SQL_BINDS dalam UNIFIED_AUDIT_TRAIL	Daftar variabel pengikat, jika ada, yang terkait dengan SQL_TEXT. Nilai sampel adalah parameter_1,parameter_2 .
pid	nomo	Kolom OS_PROCESS dalam UNIFIED_AUDIT_TRAIL	Pengidentifikasi proses sistem operasi dari proses basis data Oracle. Nilai sampel adalah 22396.
remoteHost	string	Kolom AUTHENTICATION_TYPE dalam UNIFIED_AUDIT_TRAIL	Alamat IP klien atau nama host tempat sesi dibangkitkan. Nilai sampel adalah 123.456.789.123 .
remotePort	string	Kolom AUTHENTICATION_TYPE dalam UNIFIED_AUDIT_TRAIL	Nomor porta klien. Nilai yang lazim di lingkungan Oracle Database adalah 1521.

Field	Tipe data	Sumber	Deskripsi
`rowCount`	nomo	N/A	Bidang ini tidak digunakan untuk RDS for Oracle dan selalu null.
`serverHost`	string	Host basis data	Alamat IP host server basis data. Nilai sampel adalah `123.456.789.123` .
`serverType`	string	N/A	Jenis server basis data. Nilainya selalu `ORACLE`.
`serverVersion`	string	Host basis data	Versi Amazon RDS for Oracle, Pembaruan Rilis (RU), dan Revisi Pembaruan Rilis (RUR). Nilai sampel adalah `19.0.0.0.ru-2020-01.rur-2020-01.r1.EE.3` .
`serviceName`	string	Host basis data	Nama layanan. Nilai sampel adalah `oracle-ee` .
`sessionId`	nomo	Kolom `SESSIONID` dalam `UNIFIED_AUDIT_TRAIL`	Pengidentifikasi sesi audit. Contohnya adalah `1894327130` .
`startTime`	string	N/A	Bidang ini tidak digunakan untuk RDS for Oracle dan selalu null.

Field	Tipe data	Sumber	Deskripsi
`statementId`	nomo	Kolom `STATEMENT_ID` dalam `UNIFIED_AUDIT_TRAIL`	ID numerik untuk setiap eksekusi pernyataan. Sebuah pernyataan dapat menyebabkan banyak tindakan. Nilai sampel adalah `142197`.
`substatementId`	N/A	N/A	Bidang ini tidak digunakan untuk RDS for Oracle dan selalu null.
`transactionId`	string	Kolom `TRANSACTION_ID` dalam `UNIFIED_AUDIT_TRAIL`	Pengidentifikasi transaksi yang mengubah objek. Nilai sampel adalah `02000800D5030000` .

databaseActivityEventDaftar bidang untuk Amazon RDS for SQL Server

Field	Tipe data	Sumber	Deskripsi
`class`	string	`sys.fn_get_audit_file.class_type` dipetakan ke `sys.dm_audit_class_type_map.class_type_desc`	Kelas peristiwa aktivitas. Lihat informasi yang lebih lengkap di Audit SQL Server (Mesin Basis Data) dalam dokumentasi Microsoft.
`clientApplication`	string	`sys.fn_get_audit_file.application_name`	Aplikasi yang dihubung klien seperti dilaporkan oleh klien (SQL Server versi 14 dan lebih tinggi). Bidang ini null di SQL Server versi 13.

Field	Tipe data	Sumber	Deskripsi
command	string	sys.fn_get_audit_file.action_id dipetakan ke sys.dm_audit_actions.name	Kategori umum pernyataan SQL. Nilai untuk bidang ini bergantung pada nilai kelas.
commandText	string	sys.fn_get_audit_file.statement	Bidang ini menunjukkan pernyataan SQL.
databaseName	string	sys.fn_get_audit_file.database_name	Nama basis data.
dbProtocol	string	N/A	Protokol basis data. Nilai ini adalah SQLSERVER .
dbUserName	string	sys.fn_get_audit_file.server_principal_name	Pengguna basis data untuk autentikasi klien.
endTime	string	N/A	Bidang ini tidak digunakan oleh Amazon RDS for SQL Server dan nilainya null.
engineNativeAuditFields	objek	Setiap bidang sys.fn_get_audit_file yang tidak tercantum dalam kolom ini.	Secara bawaan, objek ini kosong. Ketika Anda memulai aliran aktivitas dengan opsi --engine-native-audit-fields-included , objek ini menyertakan bidang-bidang audit mesin asli lainnya, yang tidak dihasilkan oleh peta JSON ini.
errorMessage	string	N/A	Bidang ini tidak digunakan oleh Amazon RDS for SQL Server dan nilainya null.

Field	Tipe data	Sumber	Deskripsi
`exitCode`	integer	`sys.fn_get_audit_file.succeeded`	Menunjukkan apakah tindakan yang memulai peristiwa berhasil. Bidang ini tidak boleh bernilai null. Untuk semua peristiwa selain upaya masuk, bidang ini melaporkan apakah pemeriksaan izin berhasil atau gagal, tetapi tidak apakah operasi berhasil atau gagal. Nilai-nilai meliputi: • `0` – Gagal • 1 – Berhasil
`logTime`	string	`sys.fn_get_audit_file.event_time`	Stempel waktu peristiwa yang direkam oleh SQL Server.
`netProtocol`	string	N/A	Bidang ini tidak digunakan oleh Amazon RDS for SQL Server dan nilainya null.
`objectName`	string	`sys.fn_get_audit_file.object_name`	Nama objek basis data jika pernyataan SQL beroperasi pada objek.
`objectType`	string	`sys.fn_get_audit_file.class_type` dipetakan ke `sys.dm_audit_class_type_map.class_type_desc`	Jenis objek basis data jika pernyataan SQL beroperasi pada jenis objek.
`paramList`	string	N/A	Bidang ini tidak digunakan oleh Amazon RDS for SQL Server dan nilainya null.

Field	Tipe data	Sumber	Deskripsi
pid	integer	N/A	Bidang ini tidak digunakan oleh Amazon RDS for SQL Server dan nilainya null.
remoteHost	string	sys.fn_get_audit_file.client_ip	Alamat IP atau nama host klien yang menerbitkan pernyataan SQL (SQL Server versi 14 dan lebih tinggi). Bidang ini null di SQL Server versi 13.
remotePort	integer	N/A	Bidang ini tidak digunakan oleh Amazon RDS for SQL Server dan nilainya null.
rowCount	integer	sys.fn_get_audit_file.affected_rows	Jumlah baris tabel yang terpengaruh oleh pernyataan SQL (SQL Server versi 14 dan lebih tinggi). Bidang ini ada di SQL Server versi 13.
serverHost	string	Host Basis Data	Alamat IP server basis data host.
serverType	string	N/A	Jenis server basis data. Nilainya adalah SQLSERVER .
serverVersion	string	Host Basis Data	Versi server basis data, misalnya, 15.00.4073.23.v1.r1 untuk SQL Server 2017.
serviceName	string	Host Basis Data	Nama layanan. Contoh nilai adalah sqlserver-ee .
sessionId	integer	sys.fn_get_audit_file.session_id	Pengidentifikasi unik sesi.

Field	Tipe data	Sumber	Deskripsi
`startTime`	string	N/A	Bidang ini tidak digunakan oleh Amazon RDS for SQL Server dan nilainya null.
`statementId`	string	`sys.fn_get_audit_file.sequence_group_id`	Pengidentifikasi unik untuk pernyataan SQL klien. Pengidentifikasi berbeda untuk setiap peristiwa yang dihasilkan. Nilai sampel adalah `0x38eaf4156267184094bb82071aaab644` .
`substatementId`	integer	`sys.fn_get_audit_file.sequence_number`	Pengidentifikasi untuk menentukan nomor urut untuk pernyataan. Pengidentifikasi ini membantu ketika rekam yang besar dibagi menjadi beberapa rekam.
`transactionId`	integer	`sys.fn_get_audit_file.transaction_id`	Pengidentifikasi transaksi. Jika tidak ada transaksi aktif, nilainya nol.
`type`	string	Aliran aktivitas basis data yang dihasilkan	Jenis peristiwa. Nilai-nilainya adalah `record` atau `heartbeat` .

Memproses aliran aktivitas database menggunakan AWS SDK

Anda dapat memproses aliran aktivitas secara terprogram menggunakan SDK AWS . Berikut adalah contoh Java dan Python yang berfungsi penuh tentang penggunaan rekam Aliran Aktivitas Basis Data untuk pengaktifan berbasis instans.

Java

```
import java.io.ByteArrayInputStream;
```

```
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.net.InetAddress;
import java.nio.ByteBuffer;
import java.nio.charset.StandardCharsets;
import java.security.NoSuchAlgorithmException;
import java.security.NoSuchProviderException;
import java.security.Security;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.UUID;
import java.util.zip.GZIPInputStream;

import javax.crypto.Cipher;
import javax.crypto.NoSuchPaddingException;
import javax.crypto.spec.SecretKeySpec;

import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.BasicAWSCredentials;
import com.amazonaws.encryptionsdk.AwsCrypto;
import com.amazonaws.encryptionsdk.CryptoInputStream;
import com.amazonaws.encryptionsdk.jce.JceMasterKey;
import
 com.amazonaws.services.kinesis.clientlibrary.exceptions.InvalidStateException;
import com.amazonaws.services.kinesis.clientlibrary.exceptions.ShutdownException;
import com.amazonaws.services.kinesis.clientlibrary.exceptions.ThrottlingException;
import com.amazonaws.services.kinesis.clientlibrary.interfaces.IRecordProcessor;
import
 com.amazonaws.services.kinesis.clientlibrary.interfaces.IRecordProcessorCheckpointer;
import
 com.amazonaws.services.kinesis.clientlibrary.interfaces.IRecordProcessorFactory;
import
 com.amazonaws.services.kinesis.clientlibrary.lib.worker.InitialPositionInStream;
import
 com.amazonaws.services.kinesis.clientlibrary.lib.worker.KinesisClientLibConfiguration;
import com.amazonaws.services.kinesis.clientlibrary.lib.worker.ShutdownReason;
import com.amazonaws.services.kinesis.clientlibrary.lib.worker.Worker;
import com.amazonaws.services.kinesis.clientlibrary.lib.worker.Worker.Builder;
import com.amazonaws.services.kinesis.model.Record;
import com.amazonaws.services.kms.AWSKMS;
import com.amazonaws.services.kms.AWSKMSClientBuilder;
import com.amazonaws.services.kms.model.DecryptRequest;
import com.amazonaws.services.kms.model.DecryptResult;
```

```
import com.amazonaws.util.Base64;
import com.amazonaws.util.IOUtils;
import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
import com.google.gson.annotations.SerializedName;
import org.bouncycastle.jce.provider.BouncyCastleProvider;

public class DemoConsumer {

    private static final String STREAM_NAME = "aws-rds-das-[instance-external-
resource-id]"; // aws-rds-das-db-ABCD123456
    private static final String APPLICATION_NAME = "AnyApplication"; //unique
 application name for dynamo table generation that holds kinesis shard tracking
    private static final String AWS_ACCESS_KEY =
 "[AWS_ACCESS_KEY_TO_ACCESS_KINESIS]";
    private static final String AWS_SECRET_KEY =
 "[AWS_SECRET_KEY_TO_ACCESS_KINESIS]";
    private static final String RESOURCE_ID = "[external-resource-id]"; // db-
ABCD123456
    private static final String REGION_NAME = "[region-name]"; //us-east-1, us-
east-2...
    private static final BasicAWSCredentials CREDENTIALS = new
 BasicAWSCredentials(AWS_ACCESS_KEY, AWS_SECRET_KEY);
    private static final AWSStaticCredentialsProvider CREDENTIALS_PROVIDER = new
 AWSStaticCredentialsProvider(CREDENTIALS);

    private static final AwsCrypto CRYPTO = new AwsCrypto();
    private static final AWSKMS KMS = AWSKMSClientBuilder.standard()
            .withRegion(REGION_NAME)
            .withCredentials(CREDENTIALS_PROVIDER).build();

    class Activity {
        String type;
        String version;
        String databaseActivityEvents;
        String key;
    }

    class ActivityEvent {
        @SerializedName("class") String _class;
        String clientApplication;
        String command;
        String commandText;
        String databaseName;
```

```
        String dbProtocol;
        String dbUserName;
        String endTime;
        String errorMessage;
        String exitCode;
        String logTime;
        String netProtocol;
        String objectName;
        String objectType;
        List<String> paramList;
        String pid;
        String remoteHost;
        String remotePort;
        String rowCount;
        String serverHost;
        String serverType;
        String serverVersion;
        String serviceName;
        String sessionId;
        String startTime;
        String statementId;
        String substatementId;
        String transactionId;
        String type;
    }

    class ActivityRecords {
        String type;
        String clusterId; // note that clusterId will contain an empty string on RDS
Oracle and RDS SQL Server
        String instanceId;
        List<ActivityEvent> databaseActivityEventList;
    }

    static class RecordProcessorFactory implements IRecordProcessorFactory {
        @Override
        public IRecordProcessor createProcessor() {
            return new RecordProcessor();
        }
    }

    static class RecordProcessor implements IRecordProcessor {

        private static final long BACKOFF_TIME_IN_MILLIS = 3000L;
```

```
        private static final int PROCESSING_RETRIES_MAX = 10;
        private static final long CHECKPOINT_INTERVAL_MILLIS = 60000L;
        private static final Gson GSON = new
GsonBuilder().serializeNulls().create();

        private static final Cipher CIPHER;
        static {
            Security.insertProviderAt(new BouncyCastleProvider(), 1);
            try {
                CIPHER = Cipher.getInstance("AES/GCM/NoPadding", "BC");
            } catch (NoSuchAlgorithmException | NoSuchPaddingException |
NoSuchProviderException e) {
                throw new ExceptionInInitializerError(e);
            }
        }

        private long nextCheckpointTimeInMillis;

        @Override
        public void initialize(String shardId) {
        }

        @Override
        public void processRecords(final List<Record> records, final
IRecordProcessorCheckpointer checkpointer) {
            for (final Record record : records) {
                processSingleBlob(record.getData());
            }

            if (System.currentTimeMillis() > nextCheckpointTimeInMillis) {
                checkpoint(checkpointer);
                nextCheckpointTimeInMillis = System.currentTimeMillis() +
CHECKPOINT_INTERVAL_MILLIS;
            }
        }

        @Override
        public void shutdown(IRecordProcessorCheckpointer checkpointer,
ShutdownReason reason) {
            if (reason == ShutdownReason.TERMINATE) {
                checkpoint(checkpointer);
            }
        }
```

```
        private void processSingleBlob(final ByteBuffer bytes) {
            try {
                // JSON $Activity
                final Activity activity = GSON.fromJson(new String(bytes.array(),
StandardCharsets.UTF_8), Activity.class);

                // Base64.Decode
                final byte[] decoded =
Base64.decode(activity.databaseActivityEvents);
                final byte[] decodedDataKey = Base64.decode(activity.key);

                Map<String, String> context = new HashMap<>();
                context.put("aws:rds:db-id", RESOURCE_ID);

                // Decrypt
                final DecryptRequest decryptRequest = new DecryptRequest()

.withCiphertextBlob(ByteBuffer.wrap(decodedDataKey)).withEncryptionContext(context);
                final DecryptResult decryptResult = KMS.decrypt(decryptRequest);
                final byte[] decrypted = decrypt(decoded,
getByteArray(decryptResult.getPlaintext()));

                // GZip Decompress
                final byte[] decompressed = decompress(decrypted);
                // JSON $ActivityRecords
                final ActivityRecords activityRecords = GSON.fromJson(new
String(decompressed, StandardCharsets.UTF_8), ActivityRecords.class);

                // Iterate throught $ActivityEvents
                for (final ActivityEvent event :
activityRecords.databaseActivityEventList) {
                    System.out.println(GSON.toJson(event));
                }
            } catch (Exception e) {
                // Handle error.
                e.printStackTrace();
            }
        }

        private static byte[] decompress(final byte[] src) throws IOException {
            ByteArrayInputStream byteArrayInputStream = new
ByteArrayInputStream(src);
            GZIPInputStream gzipInputStream = new
GZIPInputStream(byteArrayInputStream);
```

```
            return IOUtils.toByteArray(gzipInputStream);
        }

        private void checkpoint(IRecordProcessorCheckpointer checkpointer) {
            for (int i = 0; i < PROCESSING_RETRIES_MAX; i++) {
                try {
                    checkpointer.checkpoint();
                    break;
                } catch (ShutdownException se) {
                    // Ignore checkpoint if the processor instance has been shutdown
(fail over).
                    System.out.println("Caught shutdown exception, skipping
checkpoint." + se);
                    break;
                } catch (ThrottlingException e) {
                    // Backoff and re-attempt checkpoint upon transient failures
                    if (i >= (PROCESSING_RETRIES_MAX - 1)) {
                        System.out.println("Checkpoint failed after " + (i + 1) +
"attempts." + e);
                        break;
                    } else {
                        System.out.println("Transient issue when checkpointing -
attempt " + (i + 1) + " of " + PROCESSING_RETRIES_MAX + e);
                    }
                } catch (InvalidStateException e) {
                    // This indicates an issue with the DynamoDB table (check for
table, provisioned IOPS).
                    System.out.println("Cannot save checkpoint to the DynamoDB table
used by the Amazon Kinesis Client Library." + e);
                    break;
                }
                try {
                    Thread.sleep(BACKOFF_TIME_IN_MILLIS);
                } catch (InterruptedException e) {
                    System.out.println("Interrupted sleep" + e);
                }
            }
        }
    }

    private static byte[] decrypt(final byte[] decoded, final byte[] decodedDataKey)
throws IOException {
        // Create a JCE master key provider using the random key and an AES-GCM
encryption algorithm
```

```
        final JceMasterKey masterKey = JceMasterKey.getInstance(new
SecretKeySpec(decodedDataKey, "AES"),
                "BC", "DataKey", "AES/GCM/NoPadding");
        try (final CryptoInputStream<JceMasterKey> decryptingStream =
CRYPTO.createDecryptingStream(masterKey, new ByteArrayInputStream(decoded));
             final ByteArrayOutputStream out = new ByteArrayOutputStream()) {
            IOUtils.copy(decryptingStream, out);
            return out.toByteArray();
        }
    }

    public static void main(String[] args) throws Exception {
        final String workerId = InetAddress.getLocalHost().getCanonicalHostName() +
":" + UUID.randomUUID();
        final KinesisClientLibConfiguration kinesisClientLibConfiguration =
                new KinesisClientLibConfiguration(APPLICATION_NAME, STREAM_NAME,
CREDENTIALS_PROVIDER, workerId);

kinesisClientLibConfiguration.withInitialPositionInStream(InitialPositionInStream.LATEST);
        kinesisClientLibConfiguration.withRegionName(REGION_NAME);
        final Worker worker = new Builder()
                .recordProcessorFactory(new RecordProcessorFactory())
                .config(kinesisClientLibConfiguration)
                .build();

        System.out.printf("Running %s to process stream %s as worker %s...\n",
APPLICATION_NAME, STREAM_NAME, workerId);

        try {
            worker.run();
        } catch (Throwable t) {
            System.err.println("Caught throwable while processing data.");
            t.printStackTrace();
            System.exit(1);
        }
        System.exit(0);
    }

    private static byte[] getByteArray(final ByteBuffer b) {
        byte[] byteArray = new byte[b.remaining()];
        b.get(byteArray);
        return byteArray;
    }
```

```
}
```

Python

```
import base64
import json
import zlib
import aws_encryption_sdk
from aws_encryption_sdk import CommitmentPolicy
from aws_encryption_sdk.internal.crypto import WrappingKey
from aws_encryption_sdk.key_providers.raw import RawMasterKeyProvider
from aws_encryption_sdk.identifiers import WrappingAlgorithm, EncryptionKeyType
import boto3

REGION_NAME = '<region>'                    # us-east-1
RESOURCE_ID = '<external-resource-id>'      # db-ABCD123456
STREAM_NAME = 'aws-rds-das-' + RESOURCE_ID  # aws-rds-das-db-ABCD123456

enc_client =
 aws_encryption_sdk.EncryptionSDKClient(commitment_policy=CommitmentPolicy.FORBID_ENCRYPT_AL

class MyRawMasterKeyProvider(RawMasterKeyProvider):
    provider_id = "BC"

    def __new__(cls, *args, **kwargs):
        obj = super(RawMasterKeyProvider, cls).__new__(cls)
        return obj

    def __init__(self, plain_key):
        RawMasterKeyProvider.__init__(self)
        self.wrapping_key =
 WrappingKey(wrapping_algorithm=WrappingAlgorithm.AES_256_GCM_IV12_TAG16_NO_PADDING,
                                        wrapping_key=plain_key,
 wrapping_key_type=EncryptionKeyType.SYMMETRIC)

    def _get_raw_key(self, key_id):
        return self.wrapping_key


def decrypt_payload(payload, data_key):
    my_key_provider = MyRawMasterKeyProvider(data_key)
    my_key_provider.add_master_key("DataKey")
    decrypted_plaintext, header = enc_client.decrypt(
```

```
        source=payload,

 materials_manager=aws_encryption_sdk.materials_managers.default.DefaultCryptoMaterialsManag
    return decrypted_plaintext


def decrypt_decompress(payload, key):
    decrypted = decrypt_payload(payload, key)
    return zlib.decompress(decrypted, zlib.MAX_WBITS + 16)


def main():
    session = boto3.session.Session()
    kms = session.client('kms', region_name=REGION_NAME)
    kinesis = session.client('kinesis', region_name=REGION_NAME)

    response = kinesis.describe_stream(StreamName=STREAM_NAME)
    shard_iters = []
    for shard in response['StreamDescription']['Shards']:
        shard_iter_response = kinesis.get_shard_iterator(StreamName=STREAM_NAME,
 ShardId=shard['ShardId'],

 ShardIteratorType='LATEST')
        shard_iters.append(shard_iter_response['ShardIterator'])

    while len(shard_iters) > 0:
        next_shard_iters = []
        for shard_iter in shard_iters:
            response = kinesis.get_records(ShardIterator=shard_iter, Limit=10000)
            for record in response['Records']:
                record_data = record['Data']
                record_data = json.loads(record_data)
                payload_decoded =
 base64.b64decode(record_data['databaseActivityEvents'])
                data_key_decoded = base64.b64decode(record_data['key'])
                data_key_decrypt_result =
 kms.decrypt(CiphertextBlob=data_key_decoded,

 EncryptionContext={'aws:rds:db-id': RESOURCE_ID})
                print (decrypt_decompress(payload_decoded,
 data_key_decrypt_result['Plaintext']))
            if 'NextShardIterator' in response:
                next_shard_iters.append(response['NextShardIterator'])
        shard_iters = next_shard_iters
```

```
if __name__ == '__main__':
    main()
```

Mengelola akses ke aliran aktivitas basis data

Setiap pengguna dengan privilese peran AWS Identity and Access Management (IAM) yang sesuai untuk aliran aktivitas basis data dapat membuat, memulai, menghentikan, dan mengubah setelan aliran aktivitas untuk instans basis data. Semua tindakan ini dimasukkan ke dalam log audit aliran. Untuk praktik kepatuhan terbaik, sebaiknya jangan berikan privilese ini kepada DBA.

Anda mengatur akses ke aliran aktivitas basis data dengan menggunakan kebijakan IAM. Lihat informasi yang lebih lengkap tentang autentikasi Amazon RDS di Manajemen identitas dan akses untuk Amazon RDS. Lihat informasi yang lebih lengkap tentang pembuatan kebijakan IAM di Membuat dan menggunakan kebijakan IAM untuk akses basis data IAM.

Example Kebijakan untuk memungkinkan pengonfigurasian aliran aktivitas basis data

Untuk memberi pengguna akses terperinci untuk mengubah aliran aktivitas, gunakan kunci-kunci konteks operasi khusus layanan `rds:StartActivityStream` dan `rds:StopActivityStream` dalam kebijakan IAM. Contoh kebijakan IAM berikut memungkinkan pengguna atau peran mengonfigurasikan aliran aktivitas.

```
{
   "Version":"2012-10-17",
   "Statement":[
        {
            "Sid":"ConfigureActivityStreams",
            "Effect":"Allow",
            "Action": [
                "rds:StartActivityStream",
                "rds:StopActivityStream"
            ],
            "Resource":"*",
          }
     ]
}
```

Example Kebijakan untuk memungkinkan pemulaian aliran aktivitas basis data

Contoh kebijakan IAM berikut memungkinkan pengguna atau peran memulai aliran aktivitas.

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"AllowStartActivityStreams",
            "Effect":"Allow",
            "Action":"rds:StartActivityStream",
            "Resource":"*"
        }
    ]
}
```

Example Kebijakan untuk memungkinkan penghentian aliran aktivitas basis data

Contoh kebijakan IAM berikut memungkinkan pengguna atau peran menghentikan aliran aktivitas.

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"AllowStopActivityStreams",
            "Effect":"Allow",
            "Action":"rds:StopActivityStream",
            "Resource":"*"
        }
     ]
}
```

Example Kebijakan untuk menolak pemulaian aliran aktivitas basis data

Contoh kebijakan IAM berikut mencegah pengguna atau peran dari memulai aliran aktivitas.

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"DenyStartActivityStreams",
            "Effect":"Deny",
```

```
            "Action":"rds:StartActivityStream",
            "Resource":"*"
        }
     ]
}
```

Example Kebijakan untuk menolak penghentian aliran aktivitas basis data

Contoh kebijakan IAM berikut mencegah pengguna atau peran dari menghentikan aliran aktivitas.

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"DenyStopActivityStreams",
            "Effect":"Deny",
            "Action":"rds:StopActivityStream",
            "Resource":"*"
        }
    ]
}
```

Menggunakan Amazon RDS Custom

Amazon RDS Custom mengotomatiskan tugas dan operasi administrasi basis data. Dengan RDS Custom, Anda sebagai administrator basis data dapat mengakses dan menyesuaikan lingkungan basis data dan sistem operasi Anda. Dengan RDS Custom, Anda dapat melakukan penyesuaian untuk memenuhi persyaratan aplikasi lama, kustom, dan paket.

Untuk mengetahui seminar web dan blog terbaru tentang RDS Custom, lihat Sumber daya Amazon RDS Custom.

Topik

Mengatasi tantangan penyesuaian basis data

Amazon RDS Custom menghadirkan manfaat Amazon RDS ke pasar yang tidak dapat dengan mudah berpindah ke layanan terkelola penuh karena adanya penyesuaian yang perlu dilakukan dengan aplikasi pihak ketiga. Amazon RDS Custom menghemat waktu administratif, tahan lama, dan dapat diskalakan sesuai bisnis Anda.

Jika Anda memerlukan seluruh database dan sistem operasi untuk dikelola sepenuhnya oleh AWS, kami merekomendasikan Amazon RDS. Jika Anda memerlukan hak administratif ke basis data dan sistem operasi yang mendasari untuk membuat aplikasi dependen tersedia, Amazon RDS Custom adalah opsi yang lebih tepat. Jika Anda masih ingin memegang penuh tanggung jawab manajemen dan hanya memerlukan layanan komputas terkelola, opsi terbaiknya adalah mengelola basis data komersial Anda secara mandiri di Amazon EC2.

Untuk memberikan pengalaman layanan terkelola, Amazon RDS tidak mengizinkan Anda mengakses host yang mendasarinya. Amazon RDS juga melarang akses ke beberapa prosedur dan objek yang memerlukan hak istimewa tingkat tinggi. Namun, untuk beberapa aplikasi, Anda mungkin perlu melakukan operasi sebagai pengguna sistem operasi (OS) yang memiliki hak istimewa.

Misalnya, Anda mungkin perlu melakukan hal berikut:

- Menginstal basis data kustom serta patch dan paket OS.
- Mengonfigurasi pengaturan basis data tertentu.
- Mengonfigurasi sistem file untuk berbagi file secara langsung dengan aplikasinya.

Jika aplikasi Anda perlu disesuaikan, sebelumnya Anda harus melakukan deployment basis data on-premise atau di Amazon EC2. Dalam hal ini, Anda memegang sebagian besar atau seluruh tanggung jawab untuk manajemen basis data, seperti yang dirangkum dalam tabel berikut.

Fitur	Tanggung jawab on-premise	Tanggung jawab Amazon EC2	Tanggung jawab Amazon RDS
Pengoptimalan aplikasi	Pelanggan	Pelanggan	Pelanggan
Penskalaan	Pelanggan	Pelanggan	AWS
Ketersediaan tinggi	Pelanggan	Pelanggan	AWS
Pencadangan basis data	Pelanggan	Pelanggan	AWS
Patching perangkat lunak basis data	Pelanggan	Pelanggan	AWS
Instalasi perangkat lunak basis data	Pelanggan	Pelanggan	AWS
Patching OS	Pelanggan	Pelanggan	AWS
Instalasi OS	Pelanggan	Pelanggan	AWS
Pemeliharaan server	Pelanggan	AWS	AWS
Siklus hidup perangkat keras	Pelanggan	AWS	AWS

Fitur	Tanggung jawab on-premise	Tanggung jawab Amazon EC2	Tanggung jawab Amazon RDS
Daya, jaringan, dan pendinginan	Pelanggan	AWS	AWS

Ketika Anda mengelola perangkat lunak basis data secara mandiri, Anda akan memiliki kontrol yang lebih besar, tetapi juga lebih rentan terhadap kesalahan pengguna. Misalnya, ketika melakukan perubahan secara manual, Anda mungkin secara tidak sengaja menyebabkan waktu henti pada aplikasi. Anda mungkin perlu waktu berjam-jam untuk memeriksa setiap perubahan untuk mengidentifikasi dan memperbaiki masalah. Idealnya, Anda menginginkan layanan basis data terkelola yang mengotomatiskan tugas umum DBA, tetapi juga mendukung akses istimewa ke basis data dan sistem operasi yang mendasarinya.

Model manajemen dan manfaat Amazon RDS Custom

Amazon RDS Custom adalah layanan basis data terkelola untuk aplikasi lama, kustom, dan paket yang memerlukan akses ke sistem operasi dan lingkungan basis data yang mendasarinya. RDS Custom mengotomatiskan pengaturan, operasi, dan penskalaan database AWS Cloud sekaligus memberi Anda akses ke database dan sistem operasi yang mendasarinya. Dengan akses ini, Anda dapat mengonfigurasi pengaturan, menginstal patch, dan mengaktifkan fitur asli untuk memenuhi persyaratan aplikasi dependen. Dengan RDS Custom, Anda dapat menjalankan beban kerja database Anda menggunakan AWS Management Console atau file. AWS CLI

RDS Custom hanya mendukung mesin DB Oracle Database dan Microsoft SQL Server.

Topik

- Model tanggung jawab bersama dalam RDS Custom
- Perimeter dukungan dan konfigurasi yang tidak didukung di RDS Custom
- Manfaat utama RDS Custom

Model tanggung jawab bersama dalam RDS Custom

Dengan RDS Custom, Anda menggunakan fitur terkelola dari Amazon RDS, tetapi mengelola host dan menyesuaikan OS seperti di Amazon EC2. Anda memiliki tanggung jawab manajemen basis

data tambahan selain yang Anda lakukan di Amazon RDS. Hasilnya, kontrol Anda terhadap basis data dan manajemen instans DB lebih besar dibandingkan dengan yang Anda lakukan di Amazon RDS, dan masih bisa mendapatkan manfaat dari otomatisasi RDS.

Tanggung jawab bersama berarti:

1. Anda memegang tanggung jawab atas proses saat menggunakan fitur RDS Custom.

   Misalnya, di RDS Custom for Oracle, Anda mengontrol patch basis data Oracle mana yang akan digunakan dan menentukan waktu penerapannya ke instans DB Anda.
2. Anda bertanggung jawab untuk memastikan bahwa setiap penyesuaian ke fitur RDS Custom berfungsi dengan benar.

   Untuk mencegah penyesuaian yang tidak valid, RDS Custom memiliki perangkat lunak otomatisasi yang berjalan di luar instans DB Anda. Jika instans Amazon EC2 yang mendasari Anda mengalami gangguan, RDS Custom berupaya menyelesaikan masalah ini secara otomatis dengan melakukan reboot atau mengganti instans EC2. Satu-satunya perubahan yang dapat dilihat oleh pengguna adalah alamat IP baru. Untuk informasi selengkapnya, lihat Penggantian host Amazon RDS Custom.

Tabel berikut menjelaskan model tanggung jawab bersama untuk berbagai fitur RDS Custom.

Fitur	Tanggung jawab Amazon EC2	Tanggung jawab Amazon RDS	Tanggung jawab RDS Custom for Oracle	Tanggung jawab RDS Custom for SQL Server
Pengoptimalan aplikasi	Pelanggan	Pelanggan	Pelanggan	Pelanggan
Penskalaan	Pelanggan	AWS	Bersama	Bersama
Ketersediaan tinggi	Pelanggan	AWS	Pelanggan	AWS
Pencadangan basis data	Pelanggan	AWS	Bersama	AWS

Fitur	Tanggung jawab Amazon EC2	Tanggung jawab Amazon RDS	Tanggung jawab RDS Custom for Oracle	Tanggung jawab RDS Custom for SQL Server
Patching perangkat lunak basis data	Pelanggan	AWS	Bersama	AWS
Instalasi perangkat lunak basis data	Pelanggan	AWS	Bersama	AWS [1] untuk RPEV, Pelanggan untuk CEV
Patching OS	Pelanggan	AWS	Pelanggan	AWS [1] untuk RPEV, Pelanggan untuk CEV
Instalasi OS	Pelanggan	AWS	Bersama	AWS
Pemeliharaan server	AWS	AWS	AWS	AWS
Siklus hidup perangkat keras	AWS	AWS	AWS	AWS
Daya, jaringan, dan pendinginan	AWS	AWS	AWS	AWS

[1] Versi mesin khusus (CEV) adalah snapshot volume biner dari versi database dan Amazon Machine Image (AMI). Versi mesin yang disediakan RDS (RPEV) adalah default Amazon Machine Image (AMI) dan instalasi Microsoft SQL Server.

Anda dapat membuat instans DB RDS Custom menggunakan Microsoft SQL Server. Dalam kasus ini:

- Anda dapat memilih dari dua model lisensi: Lisensi Termasuk (LI) dan Bring Your Own Media (BYOM).
- Dengan LI, Anda tidak perlu membeli lisensi SQL Server secara terpisah. AWS memegang lisensi untuk perangkat lunak database SQL Server.

- Dengan BYOM, Anda menyediakan dan menginstal binari dan lisensi Microsoft SQL Server Anda sendiri.

Anda dapat membuat instans DB RDS Custom menggunakan Oracle Database. Dalam kasus ini, lakukan hal berikut:

- Kelola media Anda sendiri.

  Ketika menggunakan RDS Custom, Anda mengunggah patch dan file instalasi basis data Anda sendiri. Anda membuat versi mesin kustom (CEV) dari file-file ini. Kemudian Anda dapat membuat instans DB RDS Custom menggunakan CEV ini.
- Kelola lisensi Anda sendiri.

  Anda membawa lisensi Oracle Database Anda sendiri dan mengelola lisensi secara mandiri.

Perimeter dukungan dan konfigurasi yang tidak didukung di RDS Custom

RDS Custom menyediakan kemampuan pemantauan yang disebut perimeter dukungan. Fitur ini memastikan konfigurasi host dan lingkungan basis data Anda sudah benar. Jika Anda melakukan perubahan yang membuat instans DB Anda berada di luar perimeter dukungan, RDS Custom akan mengubah status instans menjadi `unsupported-configuration` hingga Anda memperbaiki masalah konfigurasi ini secara manual. Untuk informasi selengkapnya, lihat Perimeter dukungan RDS Custom.

Manfaat utama RDS Custom

Dengan RDS Custom, Anda dapat melakukan hal berikut:

- Mengotomatiskan banyak tugas administratif yang sama dengan Amazon RDS, termasuk tugas berikut:
  - Manajemen siklus hidup basis data
  - Pencadangan dan point-in-time pemulihan otomatis (PITR)
  - Memantau kesehatan instans RDS Custom DB dan mengamati perubahan pada infrastruktur, sistem operasi, dan proses database.
  - Memberikan notifikasi atau mengambil tindakan untuk memperbaiki masalah sesuai gangguan pada instans DB
- Menginstal aplikasi pihak ketiga.

Anda dapat menginstal perangkat lunak untuk menjalankan aplikasi dan agen kustom. Karena Anda memiliki akses istimewa ke host, Anda dapat memodifikasi sistem file untuk mendukung aplikasi lama.

- Menginstal patch kustom.

  Anda dapat menerapkan patch basis data kustom atau memodifikasi paket OS pada instans DB RDS Custom Anda.

- Menyiapkan basis data on-premise sebelum memindahkannya ke layanan terkelola penuh.

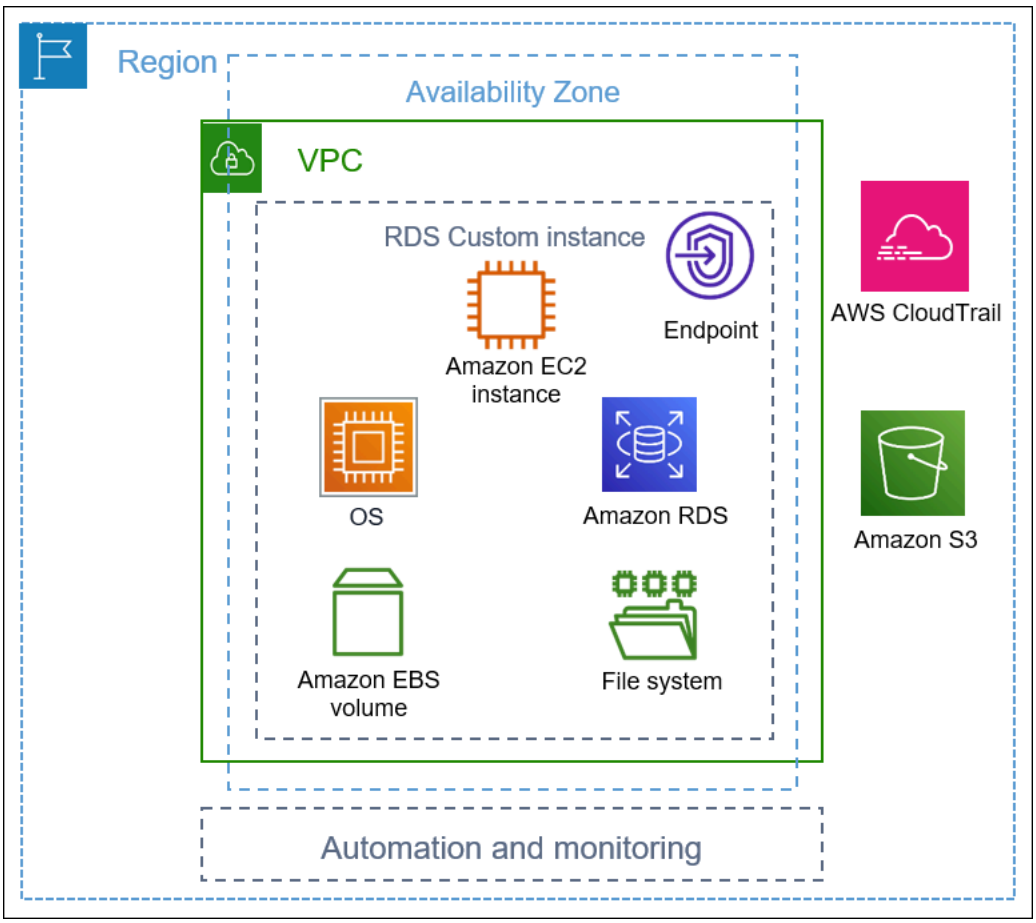
  Jika Anda mengelola basis data on-premise Anda sendiri, Anda dapat mengatur basis data ke RDS Custom sebagaimana adanya. Setelah Anda memahami lingkungan cloud, Anda bisa memigrasikan basis data Anda ke instans DB Amazon RDS yang terkelola sepenuhnya.

- Membuat otomatisasi Anda sendiri.

  Anda dapat membuat, menjadwalkan, dan menjalankan skrip otomatisasi kustom untuk alat bantu pelaporan, manajemen, atau diagnostik.

Arsitektur Amazon RDS Custom

Arsitektur Amazon RDS Custom didasarkan pada Amazon RDS, dengan perbedaan penting. Diagram berikut menunjukkan komponen utama arsitektur RDS Custom.



Topik

- VPC
- Otomatisasi dan pemantauan RDS Custom
- Amazon S3
- AWS CloudTrail

VPC

Seperti di Amazon RDS, instans DB RDS Custom Anda berada di cloud privat virtual (VPC).

Instans DB RDS Custom Anda terdiri dari komponen utama berikut:

- Instans Amazon EC2
- Titik akhir instans
- Sistem operasi diinstal pada instans Amazon EC2
- Penyimpanan Amazon EBS yang berisi sistem file tambahan

Otomatisasi dan pemantauan RDS Custom

RDS Custom memiliki perangkat lunak otomatisasi yang berjalan di luar instans DB. Perangkat lunak ini berkomunikasi dengan agen pada instans DB dan dengan komponen lain dalam lingkungan RDS Custom secara keseluruhan.

Fitur pemantauan dan pemulihan RDS Custom menawarkan fungsionalitas yang mirip dengan Amazon RDS. Secara default, RDS Custom dalam mode otomatisasi penuh. Perangkat lunak otomatisasi memiliki tanggung jawab utama sebagai berikut:

- Mengumpulkan metrik dan mengirimkan pemberitahuan
- Melakukan pemulihan instans otomatis

Tanggung jawab penting otomatisasi RDS Custom adalah merespons masalah instans Amazon EC2 Anda. Karena berbagai alasan, host mungkin terganggu atau tidak dapat dijangkau. RDS Custom menyelesaikan masalah ini dengan boot ulang atau penggantian instans Amazon EC2.

Topik

Penggantian host Amazon RDS Custom

Jika host Amazon EC2 mengalami gangguan, RDS Custom mencoba untuk boot ulang. Jika upaya ini gagal, RDS Custom menggunakan fitur hentikan dan mulai yang sama yang disertakan dalam Amazon EC2. Satu-satunya perubahan yang terlihat oleh pelanggan ketika host diganti adalah alamat IP publik baru.

Topik

Menghentikan dan memulai host

RDS Custom mengambil langkah-langkah berikut secara otomatis, tanpa memerlukan intervensi pengguna:

1. Menghentikan host Amazon EC2.

   Instans EC2 melakukan pematian normal dan berhenti berjalan. Semua volume Amazon EBS tetap terlampir pada instans dan datanya tetap ada. Setiap data yang disimpan dalam volume penyimpanan instans (tidak didukung pada RDS Custom) atau RAM komputer host hilang.

   Untuk informasi selengkapnya, lihat Menghentikan dan memulai instans Anda dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.
2. Memulai host Amazon EC2.

   Instans EC2 bermigrasi ke perangkat keras host baru yang mendasarinya. Dalam beberapa kasus, instans DB RDS Custom tetap berada di host asli.

Efek penggantian host

Di RDS Custom, Anda memiliki kontrol penuh atas volume perangkat root dan volume penyimpanan Amazon EBS. Volume root dapat berisi data dan konfigurasi penting yang tidak ingin Anda hilangkan.

RDS Custom for Oracle mempertahankan semua basis data dan data pelanggan setelah operasi, termasuk data volume root. Tidak perlu ada intervensi pengguna. Pada RDS Custom for SQL Server, data basis data dipertahankan, tetapi semua data pada drive C:, termasuk sistem operasi dan data pelanggan, hilang.

Setelah proses penggantian, host Amazon EC2 memiliki alamat IP publik baru. Host mempertahankan hal berikut:

- ID Instans
- Alamat IP privat
- Alamat IP elastis
- Metadata instans
- Data volume penyimpanan data
- Data volume root (pada RDS Custom for Oracle)

Praktik terbaik untuk host Amazon EC2

Fitur penggantian host Amazon EC2 mencakup sebagian besar skenario gangguan Amazon EC2. Sebaiknya Anda untuk mematuhi praktik terbaik berikut:

- Sebelum mengubah konfigurasi atau sistem operasi, cadangkan data Anda. Jika volume root atau sistem operasi rusak, penggantian host tidak dapat memperbaikinya. Satu-satunya pilihan Anda adalah pemulihan dari snapshot DB atau pemulihan titik waktu.
- Jangan menghentikan atau mengakhiri host Amazon EC2 fisik secara manual. Kedua tindakan tersebut mengakibatkan instans ditempatkan di luar perimeter dukungan RDS Custom.
- (RDS Custom for SQL Server) Jika Anda melampirkan volume tambahan ke host Amazon EC2, konfigurasikan untuk dipasang kembali saat memulai ulang. Jika host mengalami gangguan, RDS Custom mungkin menghentikan dan memulai host secara otomatis.

Perimeter dukungan RDS Custom

RDS Custom menyediakan kemampuan pemantauan tambahan yang disebut perimeter dukungan. Pemantauan tambahan ini memastikan bahwa instans DB RDS Custom Anda menggunakan infrastruktur AWS, sistem operasi, dan basis data yang didukung.

Perimeter dukungan memeriksa apakah instans DB Anda sesuai dengan persyaratan yang tercantum dalam Memperbaiki konfigurasi yang tidak didukung di RDS Custom for Oracle dan Memperbaiki

konfigurasi yang tidak didukung di RDS Custom for SQL Server. Jika salah satu persyaratan ini tidak terpenuhi, RDS Custom menganggap instans DB Anda berada di luar perimeter dukungan.

Topik

- Konfigurasi yang tidak didukung di RDS Custom
- Memecahkan masalah konfigurasi yang tidak didukung

Konfigurasi yang tidak didukung di RDS Custom

Saat instans DB Anda berada di luar perimeter dukungan, RDS Custom mengubah status instans DB menjadi `unsupported-configuration` dan mengirimkan pemberitahuan peristiwa. Setelah Anda memperbaiki masalah konfigurasi, RDS Custom mengubah status instans DB kembali ke `available`.

Saat instans DB Anda dalam status `unsupported-configuration`, pernyataan berikut benar:

- Basis data Anda dapat dijangkau. Terdapat pengecualian ketika instans DB berstatus `unsupported-configuration` karena basis data mati secara tidak terduga.
- Anda tidak dapat memodifikasi instans DB Anda.
- Anda tidak dapat mengambil snapshot DB.
- Pencadangan otomatis tidak dibuat.
- Khusus untuk instans DB RDS Custom for SQL Server, RDS Custom tidak mengganti instans Amazon EC2 yang mendasarinya jika mengalami gangguan. Untuk informasi selengkapnya tentang penggantian host, lihat Penggantian host Amazon RDS Custom.
- Anda dapat menghapus instans DB, tetapi sebagian besar operasi API RDS Custom lainnya tidak tersedia.
- RDS Custom terus mendukung pemulihan titik waktu (PITR) dengan mengarsipkan file log pengulangan dan mengunggahnya ke Amazon S3. PITR dalam status `unsupported-configuration` berbeda dalam hal berikut:
  - PITR mungkin memerlukan waktu lama untuk memulihkan sepenuhnya ke instans DB RDS Custom baru. Situasi ini terjadi karena Anda tidak dapat mengambil snapshot otomatis atau manual saat instans dalam status `unsupported-configuration`.
  - PITR harus memutar ulang lebih banyak log pengulangan mulai dari snapshot terbaru yang diambil sebelum instans memasuki status `unsupported-configuration`.

- Dalam beberapa kasus, instans DB berada dalam status `unsupported-configuration` karena Anda membuat perubahan yang mencegah pengunggahan file log pengulangan yang diarsipkan. Contohnya termasuk menghentikan instans EC2, menghentikan agen RDS Custom, dan melepaskan volume EBS. Dalam kasus seperti itu, PITR tidak dapat memulihkan instans DB ke waktu pemulihan terbaru.

Memecahkan masalah konfigurasi yang tidak didukung

RDS Custom menyediakan panduan pemecahan masalah untuk status `unsupported-configuration`. Meskipun beberapa panduan berlaku untuk RDS Custom for Oracle dan RDS Custom for SQL Server, panduan lain bergantung pada mesin DB Anda. Untuk informasi pemecahan masalah khusus mesin, lihat topik berikut:

- Memperbaiki konfigurasi yang tidak didukung di RDS Custom for Oracle
- Memperbaiki konfigurasi yang tidak didukung di RDS Custom for SQL Server

Amazon S3

Jika menggunakan RDS Custom for Oracle, Anda mengunggah media instalasi ke bucket Amazon S3 buatan pengguna. RDS Custom for Oracle menggunakan media di bucket ini untuk membuat versi mesin kustom (CEV). CEV adalah snapshot volume biner dari versi basis data dan Amazon Machine Image (AMI). Dari CEV, Anda dapat membuat instans DB RDS Custom. Untuk informasi selengkapnya, lihat Menggunakan versi mesin kustom untuk Amazon RDS Custom for Oracle.

Untuk RDS Custom for Oracle dan RDS Custom for SQL Server, RDS Custom otomatis membuat bucket Amazon S3 yang diawali dengan string `do-not-delete-rds-custom-`. RDS Custom menggunakan bucket S3 `do-not-delete-rds-custom-` untuk menyimpan jenis file berikut:

- Log AWS CloudTrail untuk jejak yang dibuat oleh RDS Custom
- Artefak perimeter dukungan (lihat Perimeter dukungan RDS Custom)
- File log pengulangan basis data (khusus RDS Custom for Oracle)
- Log transaksi (khusus RDS Custom for SQL Server)
- Artefak versi mesin kustom (khusus RDS Custom for Oracle)

RDS Custom membuat bucket S3 `do-not-delete-rds-custom-` saat Anda membuat salah satu sumber daya berikut:

- CEV pertama untuk RDS Custom for Oracle
- Instans DB pertama untuk RDS Custom for SQL Server

RDS Custom membuat satu bucket untuk setiap kombinasi berikut:

- ID Akun AWS
- Jenis mesin (baik RDS Custom for Oracle atau RDS Custom for SQL Server)
- Wilayah AWS

Misalnya, jika Anda membuat CEV RDS Custom for Oracle dalam satu Wilayah AWS, ada satu bucket `do-not-delete-rds-custom-`. Jika Anda membuat beberapa instans RDS Custom for SQL Server dan instans tersebut berada di Wilayah AWS yang berbeda, satu bucket `do-not-delete-rds-custom-` ada di masing-masing Wilayah AWS. Jika Anda membuat satu instans RDS Custom for Oracle dan dua instans RDS Custom for SQL Server dalam satu Wilayah AWS, akan ada dua bucket `do-not-delete-rds-custom-`.

AWS CloudTrail

RDS Custom secara otomatis membuat jejak AWS CloudTrail yang namanya dimulai dengan `do-not-delete-rds-custom-`. Perimeter dukungan RDS Custom bergantung pada peristiwa dari CloudTrail untuk menentukan apakah tindakan Anda memengaruhi otomatisasi RDS Custom. Untuk informasi selengkapnya, lihat Memecahkan masalah konfigurasi yang tidak didukung.

RDS Custom membuat jejak saat Anda membuat instans DB pertama. RDS Custom membuat satu jejak untuk setiap kombinasi berikut:

- ID Akun AWS
- Jenis mesin (baik RDS Custom for Oracle atau RDS Custom for SQL Server)
- Wilayah AWS

Saat Anda menghapus instans DB RDS Custom, CloudTrail untuk instans ini tidak dihapus secara otomatis. Dalam hal ini, Akun AWS Anda terus ditagih untuk CloudTrail yang tidak dihapus. RDS Custom tidak bertanggung jawab atas penghapusan sumber daya ini. Untuk mempelajari cara menghapus CloudTrail secara manual, lihat Menghapus jejak di Panduan Pengguna AWS CloudTrail.

Keamanan di Amazon RDS Custom

Pahami pertimbangan keamanan untuk RDS Custom.

Topik

Cara RDS Custom mengelola tugas dengan aman untuk Anda

RDS Custom menggunakan alat dan teknik berikut untuk menjalankan operasi dengan aman untuk Anda:

AWSServiceRoleForRDSCustom peran terkait layanan

Peran terkait layanan ditentukan sebelumnya oleh layanan dan mencakup semua izin yang diperlukan layanan untuk memanggil Layanan AWS lainnya untuk Anda. Untuk RDS Custom, `AWSServiceRoleForRDSCustom` adalah peran terkait layanan yang didefinisikan sesuai dengan prinsip hak akses paling rendah. RDS Custom menggunakan izin di `AmazonRDSCustomServiceRolePolicy`, yang merupakan kebijakan yang dilampirkan pada peran ini, untuk melakukan sebagian besar penyediaan dan semua tugas manajemen di luar host. Untuk informasi lebih lanjut, lihat CustomServiceRolePolicyAmazonRDS.

Saat melakukan tugas di host, otomatisasi Kustom RDS menggunakan kredenal dari peran terkait layanan untuk menjalankan perintah menggunakan. AWS Systems Manager Anda dapat mengaudit riwayat perintah melalui riwayat perintah Systems Manager dan AWS CloudTrail. Systems Manager terhubung ke instans DB RDS Custom Anda menggunakan pengaturan jaringan Anda. Untuk informasi selengkapnya, lihat Langkah 4: Konfigurasikan IAM untuk RDS Custom for Oracle.

Kredensial IAM sementara

Saat menyediakan atau menghapus sumber daya, RDS Custom terkadang menggunakan kredensial sementara yang berasal dari kredensial prinsipal IAM pemanggil. Kredensial IAM ini dibatasi oleh kebijakan IAM yang dilampirkan pada prinsipal tersebut dan kedaluwarsa setelah operasi selesai. Untuk mempelajari tentang izin yang diperlukan untuk prinsipal IAM yang

menggunakan RDS Custom, lihat Langkah 5: Berikan izin yang diperlukan ke pengguna atau peran IAM Anda.

Profil instans Amazon EC2

Profil instans EC2 adalah kontainer untuk peran IAM yang dapat Anda gunakan untuk meneruskan informasi peran ke instans EC2. Instans EC2 mendasari instans DB RDS Custom. Anda memberikan profil instans saat membuat instans DB RDS Custom. RDS Custom menggunakan kredensial profil instans EC2 saat melakukan tugas manajemen berbasis host seperti pencadangan. Untuk informasi selengkapnya, lihat Buat peran IAM dan profil instans secara manual.

Pasangan kunci SSH

Ketika RDS Custom membuat instans EC2 yang mendasari instans DB, layanan ini membuat pasangan kunci SSH untuk Anda. Kuncinya menggunakan awalan `do-not-delete-rds-custom-ssh-privatekey-db-` penamaan. AWS Secrets Manager menyimpan kunci pribadi SSH ini sebagai rahasia di Anda Akun AWS. Amazon RDS tidak menyimpan, mengakses, atau menggunakan kredensial ini. Untuk informasi selengkapnya, lihat Pasangan kunci Amazon EC2 dan instans Linux.

Sertifikat SSL

Instans DB RDS Custom tidak mendukung sertifikat SSL terkelola. Jika Anda ingin men-deploy SSL, Anda dapat mengelola sendiri sertifikat SSL di wallet Anda sendiri dan membuat pendengar SSL untuk mengamankan koneksi antara basis data klien atau untuk replikasi basis data. Untuk informasi selengkapnya, lihat Configuring Transport Layer Security Authentication dalam dokumentasi Oracle Database.

Mengamankan bucket Amazon S3 Anda dari masalah "confused deputy"

Saat Anda membuat versi mesin kustom (CEV) Amazon RDS Custom for Oracle atau instans DB RDS Custom for SQL Server, RDS Custom membuat bucket Amazon S3. Bucket S3 menyimpan file seperti artefak CEV, log redo (transaksi), item konfigurasi untuk perimeter dukungan, dan log AWS CloudTrail .

Anda dapat membuat bucket S3 ini lebih aman dengan menggunakan kunci konteks kondisi global untuk mencegah masalah confused deputy. Untuk informasi selengkapnya, lihat Pencegahan masalah confused deputy lintas layanan.

Contoh RDS Custom for Oracle berikut menunjukkan penggunaan kunci konteks kondisi global `aws:SourceArn` dan `aws:SourceAccount` dalam kebijakan bucket S3. Untuk RDS Custom for Oracle, pastikan untuk menyertakan Amazon Resource Names (ARN) untuk CEV dan instans DB. Untuk RDS Custom for SQL Server, pastikan untuk menyertakan ARN untuk instans DB.

```
...
{
  "Sid": "AWSRDSCustomForOracleInstancesObjectLevelAccess",
  "Effect": "Allow",
  "Principal": {
     "Service": "custom.rds.amazonaws.com"
  },
  "Action": [
     "s3:GetObject",
     "s3:GetObjectVersion",
     "s3:DeleteObject",
     "s3:DeleteObjectVersion",
     "s3:GetObjectRetention",
     "s3:BypassGovernanceRetention"
  ],
  "Resource": "arn:aws:s3:::do-not-delete-rds-custom-123456789012-us-east-2-c8a6f7/
RDSCustomForOracle/Instances/*",
  "Condition": {
     "ArnLike": {
        "aws:SourceArn": [
            "arn:aws:rds:us-east-2:123456789012:db:*",
            "arn:aws:rds:us-east-2:123456789012:cev:*/*"
        ]
     },
     "StringEquals": {
        "aws:SourceAccount": "123456789012"
    }
  }
},
...
```

Merotasi kredensial RDS Custom for Oracle untuk program kepatuhan

Beberapa program kepatuhan mewajibkan kredensial pengguna basis data diubah secara berkala, misalnya, setiap 90 hari. RDS Custom for Oracle secara otomatis merotasi kredensial untuk beberapa pengguna basis data yang telah ditentukan.

Topik

Rotasi kredensial otomatis untuk pengguna yang telah ditentukan

Jika instans DB RDS Custom for Oracle Anda di-host di Amazon RDS, kredensial untuk pengguna Oracle yang telah ditentukan sebelumnya akan dirotasi setiap 30 hari secara otomatis. Kredensi untuk pengguna sebelumnya berada di. AWS Secrets Manager

Pengguna Oracle yang telah ditetapkan

Pengguna basis data	Dibuat oleh	Versi mesin yang didukung	Catatan
`SYS`	Oracle	custom-oracle-ee dan custom-oracle-ee-cdb	
`SYSTEM`	Oracle	custom-oracle-ee dan custom-oracle-ee-cdb	
`RDSADMIN`	RDS	custom-oracle-ee	
`C##RDSADMIN`	RDS	custom-oracle-ee-cdb	Nama pengguna dengan awalan `C##` hanya ada di CDB. Untuk informasi selengkapnya tentang CDB, lihat Gambaran umum arsitektur Amazon RDS Custom for Oracle.
`RDS_DATAGUARD`	RDS	custom-oracle-ee	Pengguna ini hanya ada di replika baca, basis data sumber untuk replika baca, dan basis data yang telah Anda migrasikan secara fisik ke RDS Custom menggunakan Oracle Data Guard.

Pengguna basis data	Dibuat oleh	Versi mesin yang didukung	Catatan
`C##RDS_DATAGUARD`	RDS	custom-oracle-ee-cdb	Pengguna ini hanya ada di replika baca, basis data sumber untuk replika baca, dan basis data yang telah Anda migrasikan secara fisik ke RDS Custom menggunakan Oracle Data Guard. Nama pengguna dengan awalan `C##` hanya ada di CDB. Untuk informasi selengkapnya tentang CDB, lihat Gambaran umum arsitektur Amazon RDS Custom for Oracle.

Pengecualian untuk rotasi kredensial otomatis adalah instans DB RDS Custom for Oracle yang telah Anda konfigurasi secara manual sebagai basis data siaga. RDS hanya merotasi kredensial untuk replika baca yang telah Anda buat menggunakan perintah CLI `create-db-instance-read-replica` atau API `CreateDBInstanceReadReplica`.

Pedoman untuk merotasi kredensial pengguna

Untuk memastikan bahwa kredensial Anda dirotasi sesuai dengan program kepatuhan Anda, perhatikan pedoman berikut:

- Jika instans DB Anda merotasi kredensial secara otomatis, jangan mengubah atau menghapus rahasia, file kata sandi, atau kata sandi secara manual untuk pengguna yang tercantum di Pengguna Oracle yang telah ditentukan sebelumnya. Jika tidak, RDS Custom dapat menempatkan instans DB Anda di luar perimeter dukungan, yang menangguhkan rotasi otomatis.
- Pengguna master RDS tidak ditentukan sebelumnya, jadi Anda bertanggung jawab untuk mengubah kata sandi secara manual atau mengatur rotasi otomatis di Secrets Manager. Untuk informasi selengkapnya, lihat Memutar AWS Secrets Manager rahasia.

Kredensial pengguna yang dirotasi secara manual

Untuk kategori basis data berikut, RDS tidak secara otomatis merotasi kredensial untuk pengguna yang tercantum di Pengguna Oracle yang telah ditentukan sebelumnya:

- Basis data yang Anda konfigurasi secara manual untuk berfungsi sebagai basis data siaga.
- Basis data on-premise.
- Instans DB yang berada di luar perimeter dukungan atau dalam status yang tidak memungkinkan otomatisasi RDS Custom berjalan. Dalam hal ini, RDS Custom juga tidak merotasi kunci.

Jika basis data Anda berada di salah satu kategori di atas, Anda harus merotasi kredensial pengguna Anda secara manual.

Untuk merotasi kredensial pengguna secara manual untuk instans DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di Basis Data, pastikan RDS saat ini tidak mencadangkan instans DB Anda atau melakukan operasi seperti mengonfigurasi ketersediaan tinggi.
3. Di halaman detail basis data, pilih Konfigurasi dan catat ID Sumber Daya untuk instans DB. Atau Anda dapat menggunakan AWS CLI perintah`describe-db-instances`.
4. Buka konsol Secrets Manager di https://console.aws.amazon.com/secretsmanager/.
5. Di kotak pencarian, masukkan ID Sumber Daya DB Anda dan temukan rahasianya dalam form berikut:

   ```
   do-not-delete-rds-custom-db-resource-id-numeric-string
   ```

   Rahasia ini menyimpan kata sandi untuk `RDSADMIN`, `SYS`, dan `SYSTEM`. Contoh kunci berikut ditujukan untuk instans DB dengan ID sumber daya DB `db-ABCDEFG12HIJKLNMNOPQRS3TUVWX`:

   ```
   do-not-delete-rds-custom-db-ABCDEFG12HIJKLNMNOPQRS3TUVWX-123456
   ```

> **Important**
>
> Jika instans DB Anda adalah replika baca dan menggunakan mesin `custom-oracle-ee-cdb`, ada dua rahasia dengan akhiran *`db-resource-id-numeric-string`*, satu untuk pengguna master dan yang lainnya untuk `RDSADMIN`, `SYS`, dan `SYSTEM`. Untuk menemukan rahasia yang benar, jalankan perintah berikut pada host:
>
> ```
> cat /opt/aws/rdscustomagent/config/database_metadata.json | python3 -c
>  "import sys,json; print(json.load(sys.stdin)['dbMonitoringUserPassword'])"
> ```
>
> Atribut `dbMonitoringUserPassword` menunjukkan rahasia untuk `RDSADMIN`, `SYS`, dan `SYSTEM`.

6. Jika instans DB Anda ada dalam konfigurasi Oracle Data Guard, temukan rahasianya dalam form berikut:

   ```
   do-not-delete-rds-custom-db-resource-id-numeric-string-dg
   ```

   Rahasia ini menyimpan kata sandi untuk `RDS_DATAGUARD`. Contoh kunci berikut ditujukan untuk instans DB dengan ID sumber daya DB `db-ABCDEFG12HIJKLNMNOPQRS3TUVWX`:

   ```
   do-not-delete-rds-custom-db-ABCDEFG12HIJKLNMNOPQRS3TUVWX-789012-dg
   ```

7. Untuk semua pengguna database yang tercantum dalam pengguna Oracle yang telah ditentukan sebelumnya, perbarui kata sandi dengan mengikuti petunjuk di Ubah rahasia. AWS Secrets Manager

8. Jika basis data Anda adalah basis data mandiri atau basis data sumber dalam konfigurasi Oracle Data Guard:

   a. Mulai klien Oracle SQL Anda dan masuk sebagai `SYS`.

   b. Jalankan pernyataan SQL dalam form berikut untuk setiap pengguna basis data yang tercantum di Pengguna Oracle yang telah ditentukan sebelumnya:

      ```
      ALTER USER user-name IDENTIFIED BY pwd-from-secrets-manager ACCOUNT UNLOCK;
      ```

      Misalnya, jika kata sandi baru untuk `RDSADMIN` yang disimpan di Secrets Manager adalah `pwd-123`, jalankan pernyataan berikut:

```
ALTER USER RDSADMIN IDENTIFIED BY pwd-123 ACCOUNT UNLOCK;
```

9. Jika instans DB Anda menjalankan Oracle Database 12c Release 1 (12.1) dan dikelola oleh Oracle Data Guard, salin file kata sandi (`orapw`) secara manual dari instans DB primer ke setiap instans DB siaga.

   Jika instans DB Anda di-host di Amazon RDS, lokasi file kata sandi adalah `/rdsdbdata/config/orapw`. Untuk basis data yang tidak di-host di Amazon RDS, lokasi default-nya adalah `$ORACLE_HOME/dbs/orapw$ORACLE_SID` di Linux dan UNIX serta `%ORACLE_HOME%\database\PWD%ORACLE_SID%.ora` di Windows.

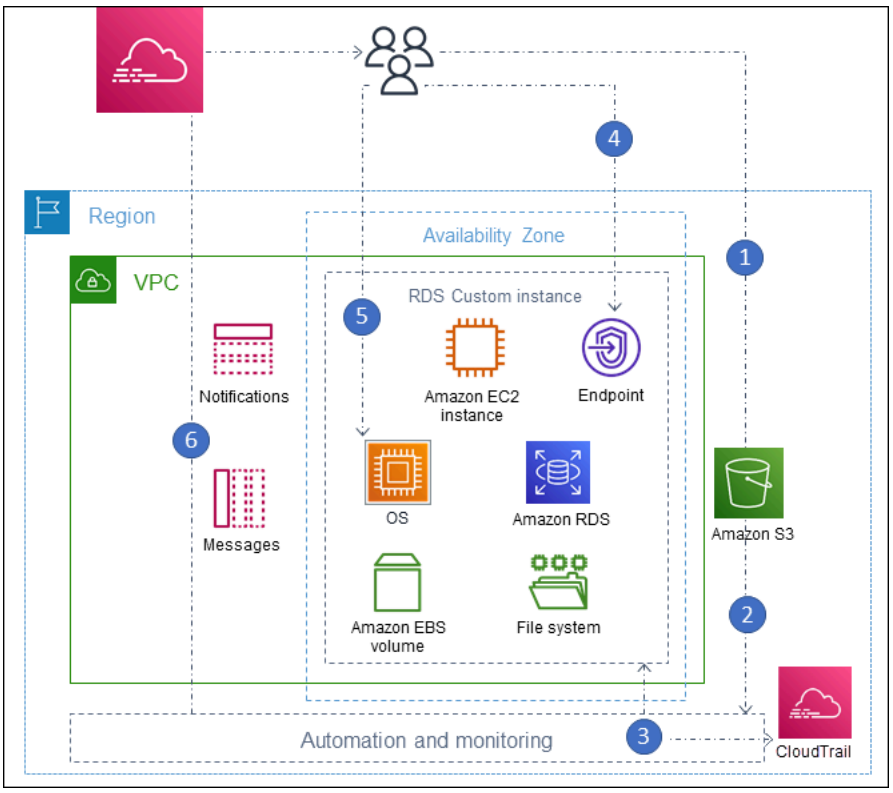
Menggunakan RDS Custom for Oracle

Di bagian berikut ini, Anda dapat menemukan petunjuk untuk membuat, mengelola, dan memelihara instans DB RDS Custom for Oracle.

Topik

Alur kerja RDS Custom for Oracle

Diagram berikut menunjukkan alur kerja umum untuk RDS Custom for Oracle.



Langkah-langkahnya adalah sebagai berikut:

1. Unggah perangkat lunak basis data Anda ke bucket Amazon S3.

   Untuk informasi selengkapnya, lihat Langkah 3: Unggah file instalasi Anda ke Amazon S3.

2. Buat versi mesin kustom (CEV) RDS Custom for Oracle dari media Anda.

   Pilih arsitektur multi-penghuni Oracle atau non-CDB tradisional. Untuk informasi selengkapnya, lihat Membuat CEV.

3. Buat instans DB RDS Custom for Oracle dari CEV.

   Untuk informasi selengkapnya, lihat Membuat instans DB RDS Custom for Oracle.

4. Hubungkan aplikasi Anda ke titik akhir instans DB.

   Untuk informasi selengkapnya, lihat Menghubungkan ke instans DB RDS Custom Anda menggunakan SSH dan Menghubungkan ke instans DB RDS Custom Anda menggunakan Session Manager.

5. (Opsional) Akses host untuk menyesuaikan perangkat lunak Anda.

6. Pantau pemberitahuan dan pesan yang dihasilkan oleh otomatisasi RDS Custom.

File instalasi basis data

Tanggung jawab Anda terhadap media adalah perbedaan utama antara Amazon RDS dan RDS Custom. Amazon RDS, yang merupakan layanan terkelola penuh, menyediakan Amazon Machine Image (AMI) dan perangkat lunak basis data. Perangkat lunak basis data Amazon RDS sudah diinstal sebelumnya, jadi Anda hanya perlu memilih mesin dan versi basis data, serta membuat basis data Anda.

Untuk RDS Custom, Anda menyediakan media Anda sendiri. Saat Anda membuat versi mesin kustom, RDS Custom menginstal media yang Anda berikan. Media RDS Custom berisi file instalasi dan patch basis data Anda. Model layanan ini disebut Bawa Media Anda Sendiri (BYOM).

Versi mesin kustom untuk RDS Custom for Oracle

Versi mesin kustom (CEV) RDS Custom for Oracle adalah snapshot volume biner dari versi basis data dan AMI. Secara default, RDS Custom for Oracle menggunakan AMI terbaru yang disediakan Amazon EC2. Anda juga dapat memilih untuk menggunakan kembali AMI yang ada.

Manifes CEV

Setelah mengunduh file instalasi basis data Oracle dari Oracle, Anda mengunggahnya ke bucket Amazon S3. Saat membuat CEV, Anda menentukan nama file dalam dokumen JSON yang disebut manifes CEV. RDS Custom for Oracle menggunakan file yang ditentukan dan AMI untuk membuat CEV Anda.

RDS Custom for Oracle menyediakan templat manifes JSON dengan file .zip yang kami rekomendasikan untuk setiap rilis Oracle Database yang didukung. Misalnya, templat berikut adalah untuk 19.17.0.0.0 RU.

```
{
    "mediaImportTemplateVersion": "2020-08-14",
    "databaseInstallationFileNames": [
        "V982063-01.zip"
    ],
    "opatchFileNames": [
        "p6880880_190000_Linux-x86-64.zip"
    ],
    "psuRuPatchFileNames": [
        "p34419443_190000_Linux-x86-64.zip",
        "p34411846_190000_Linux-x86-64.zip"
```

```
    ],
    "otherPatchFileNames": [
        "p28852325_190000_Linux-x86-64.zip",
        "p29997937_190000_Linux-x86-64.zip",
        "p31335037_190000_Linux-x86-64.zip",
        "p32327201_190000_Linux-x86-64.zip",
        "p33613829_190000_Linux-x86-64.zip",
        "p34006614_190000_Linux-x86-64.zip",
        "p34533061_190000_Linux-x86-64.zip",
        "p34533150_190000_Generic.zip",
        "p28730253_190000_Linux-x86-64.zip",
        "p29213893_1917000DBRU_Generic.zip",
        "p33125873_1917000DBRU_Linux-x86-64.zip",
        "p34446152_1917000DBRU_Linux-x86-64.zip"
    ]
}
```

Anda juga dapat menentukan parameter instalasi dalam manifes JSON. Misalnya, Anda dapat mengatur nilai nondefault untuk basis Oracle, beranda Oracle, serta ID dan nama pengguna dan grup UNIX/Linux. Untuk informasi selengkapnya, lihat Bidang JSON di manifes CEV.

Format penamaan CEV

Beri nama CEV RDS Custom for Oracle Anda menggunakan string yang ditentukan pelanggan. Format nama adalah sebagai berikut, bergantung pada rilis Oracle Database Anda:

- `19.customized_string`
- `18.customized_string`
- `12.2.customized_string`
- `12.1.customized_string`

Anda dapat menggunakan 1-50 karakter alfanumerik, garis bawah, tanda hubung, dan titik. Misalnya, Anda dapat memberi nama CEV `19.my_cev1`.

Arsitektur multi-penghuni Oracle di RDS Custom for Oracle

Arsitektur multi-penghuni Oracle memungkinkan basis data Oracle berfungsi sebagai basis data kontainer (CDB). Sebuah CDB mencakup nol, satu, atau banyak basis data pluggable (PDB) yang dibuat pelanggan. PDB adalah kumpulan skema dan objek portabel yang muncul ke aplikasi sebagai non-CDB tradisional. Mulai dari Oracle Database 21c, semua basis data Oracle adalah CDB.

Saat membuat RDS Custom for Oracle CEV, Anda menentukan arsitektur CDB atau non-CDB. Anda dapat membuat RDS Custom for Oracle CDB hanya ketika CEV yang digunakan untuk membuatnya menggunakan arsitektur multi-penghuni Oracle. Untuk informasi selengkapnya, lihat Menggunakan versi mesin kustom untuk Amazon RDS Custom for Oracle.

Membuat instans DB untuk RDS Custom for Oracle

Setelah Anda membuat CEV, CEV tersebut tersedia untuk digunakan. Anda dapat membuat beberapa CEV dan dapat membuat beberapa instans DB RDS Custom for Oracle dari CEV apa pun. Anda juga dapat mengubah status CEV agar tersedia atau tidak aktif.

Anda dapat membuat instans DB RDS Custom for Oracle dengan arsitektur Multipenyewa Oracle (jenis mesin `custom-oracle-ee-cdb`) atau dengan arsitektur non-CDB tradisional (jenis mesin `custom-oracle-ee`). Saat Anda membuat basis data kontainer (CDB), basis data tersebut berisi satu basis data pluggable (PDB) dan satu seed PDB. Anda dapat membuat PDB tambahan secara manual menggunakan Oracle SQL.

Untuk membuat instans DB RDS Custom for Oracle Anda, gunakan perintah `create-db-instance`. Dalam perintah ini, tentukan CEV yang akan digunakan. Prosedurnya mirip dengan membuat instans DB Amazon RDS. Namun, ada beberapa parameter yang berbeda. Untuk informasi selengkapnya, lihat Mengonfigurasi instans DB untuk Amazon RDS Custom for Oracle.

Koneksi basis data

Seperti instans DB Amazon RDS, instans DB RDS Custom berada di cloud privat virtual (VPC). Aplikasi Anda terhubung ke basis data Oracle menggunakan pendengar Oracle.

Jika basis data Anda adalah CDB, Anda dapat menggunakan pendengar `L_RDSCDB_001` untuk terhubung ke root CDB dan ke PDB. Jika Anda menghubungkan non-CDB ke CDB, pastikan untuk mengatur `USE_SID_AS_SERVICE_LISTENER = ON` agar aplikasi yang dimigrasikan tetap memiliki pengaturan yang sama.

Ketika Anda terhubung ke non-CDB, pengguna master adalah pengguna untuk non-CDB. Ketika Anda terhubung ke CDB, pengguna master adalah pengguna untuk CDB. Untuk terhubung ke root CDB, masuk ke host, mulai klien SQL, dan buat pengguna administratif dengan perintah SQL.

Kustomisasi RDS Custom

Anda dapat mengakses host RDS Custom untuk menginstal atau menyesuaikan perangkat lunak. Untuk menghindari konflik antara perubahan dan otomatisasi RDS Custom, Anda dapat menjeda

otomatisasi selama jangka waktu tertentu. Selama periode ini, RDS Custom tidak melakukan pemantauan atau pemulihan instans. Pada akhir periode, RDS Custom melanjutkan otomatisasi penuh. Untuk informasi selengkapnya, lihat Menjeda dan melanjutkan instans RDS Custom DB.

Arsitektur basis data untuk Amazon RDS Custom for Oracle

RDS Custom for Oracle mendukung arsitektur Oracle multi-penghuni dan nonmulti-penghuni.

Topik

Arsitektur basis data Oracle yang didukung

Oracle Database 19c mendukung arsitektur Oracle multi-penghuni dan nonmulti-penghuni. Arsitektur multi-penghuni Oracle, disebut juga arsitektur CDB, memungkinkan basis data Oracle untuk berfungsi sebagai basis data kontainer (CDB). CDB mencakup basis data pluggable (PDB). PDB adalah kumpulan skema dan objek yang muncul ke aplikasi sebagai basis data Oracle tradisional. Untuk informasi selengkapnya, lihat Introduction to the Multitenant Architecture di Oracle Multitenant Administrator's Guide.

Arsitektur CDB dan non-CDB tidak dapat digunakan bersamaan. Jika bukan CDB, basis data Oracle adalah non-CDB sehingga tidak dapat berisi PDB. Di RDS Custom for Oracle, hanya Oracle Database 19c yang mendukung arsitektur CDB. Jadi, jika Anda membuat instans DB menggunakan rilis basis data Oracle sebelumnya, Anda hanya dapat membuat non-CDB. Untuk informasi selengkapnya, lihat Pertimbangan arsitektur multi-penghuni.

Jenis mesin yang didukung

Saat Anda membuat CEV atau instans DB Amazon RDS Custom for Oracle, pilih salah satu dari jenis mesin berikut:

- `custom-oracle-ee-cdb`

  Jenis mesin ini menentukan arsitektur multi-penghuni. Opsi ini hanya tersedia untuk Oracle Database 19c. Saat Anda membuat instans RDS untuk Oracle DB menggunakan arsitektur multi-penghuni, CDB Anda menyertakan kontainer berikut:

  - Root CDB (`CDB$ROOT`)
  - Seed PDB (`PDB$SEED`)
  - PDB awal

Anda dapat membuat lebih banyak PDB menggunakan perintah Oracle SQL `CREATE PLUGGABLE DATABASE`. Anda tidak dapat menggunakan API RDS untuk membuat atau menghapus PDB.

- `custom-oracle-ee`

  Jenis mesin ini menentukan arsitektur non-CDB tradisional. Non-CDB tidak bisa berisi basis data pluggable (PDB).

Untuk informasi selengkapnya, lihat Pertimbangan arsitektur multi-penghuni.

Fitur yang didukung dalam arsitektur multi-penghuni

Instans CDB RDS Custom for Oracle mendukung fitur berikut:

- Cadangan
- Melakukan pemulihan dan pemulihan titik waktu (PITR) dari cadangan
- Replika baca
- Peningkatan versi minor

Ketersediaan fitur dan dukungan untuk RDS Custom for Oracle

Dalam topik ini, Anda dapat menemukan ringkasan ketersediaan fitur RDS Custom for Oracle dan dukungan untuk referensi cepat.

Topik

Wilayah AWS dan dukungan versi database untuk RDS Custom untuk Oracle

Ketersediaan dan dukungan fitur bervariasi di antara versi-versi spesifik setiap mesin basis data, dan di seluruh Wilayah AWS. Untuk informasi selengkapnya tentang ketersediaan versi dan Wilayah RDS Custom for Oracle, lihat RDS Custom.

Dukungan versi basis data untuk RDS Custom for Oracle

RDS Custom for Oracle mendukung versi basis data Oracle berikut:

- Oracle Database 19c
- Oracle Database 18c
- Oracle Database 12c Rilis 2 (12.2)
- Oracle Database 12c Rilis 1 (12.1)

Edisi dan dukungan lisensi untuk RDS Custom for Oracle

RDS Custom for Oracle hanya mendukung Enterprise Edition pada model BYOL.

Dukungan kelas instans DB untuk RDS Custom for Oracle

RDS Custom for Oracle mendukung kelas instans DB berikut.

Tipe	Ukuran
db.r6i	db.r6i.large |db.r6i.xlarge |db.r6i.2xlarge |db.r6i.4xlarge |db.r6i.8xlarge |db.r6i.12xlarge |db.r6i.16xlarge |db.r6i.24xlarge |db.r6i.32xlarge
db.r5b	db.r5b.large |db.r5b.xlarge |db.r5b.2xlarge |db.r5b.4xlarge |db.r5b.8xlarge |db.r5b.12xlarge |db.r5b.16xlarge |db.r5b.24xlarge
db.r5	db.r5.large |db.r5.xlarge |db.r5.2xlarge |db.r5.4xlarge |db.r5.8xlarge |db.r5.12xlarge |db.r5.16xlarge |db.r5.24xlarge
db.x2iedn	db.x2iedn.xlarge |db.x2iedn.2xlarge |db.x2iedn.4xlarge | db.x2iedn.8xlarge |db.x2iedn.16xlarge |db.x2iedn.24xlarge | db.x2iedn.32xlarge
db.x2iezn	db.x2iezn.2xlarge |db.x2iezn.4xlarge |db.x2iezn.6xlarge | db.x2iezn.8xlarge |db.x2iezn.12xlarge
db.m6i	db.m6i.large |db.m6i.xlarge |db.m6i.2xlarge |db.m6i.4xlarge |db.m6i.8xlarge |db.m6i.12xlarge |db.m6i.16xlarge |db.m6i.24xlarge |db.m6i.32xlarge
db.m5	db.m5.large |db.m5.xlarge |db.m5.2xlarge |db.m5.4xlarge |db.m5.8xlarge |db.m5.12xlarge |db.m5.16xlarge |db.m5.24xlarge
db.t3	db.t3.medium |db.t3.large |db.t3.xlarge |db.t3.2xlarge

Dukungan grup opsi untuk RDS Custom for Oracle

Anda dapat menentukan grup opsi saat membuat atau memodifikasi instans DB RDS Custom for Oracle. Untuk informasi selengkapnya, lihat Menggunakan grup opsi di RDS Custom for Oracle.

Persyaratan dan batasan RDS Custom for Oracle

Dalam topik ini, Anda dapat menemukan ringkasan ketersediaan dan persyaratan fitur Amazon RDS Custom for Oracle untuk referensi cepat.

Topik

Persyaratan umum untuk RDS Custom for Oracle

Pastikan untuk memenuhi persyaratan berikut untuk Amazon RDS Custom for Oracle:

- Anda memiliki akses ke My Oracle Support dan Oracle Software Delivery Cloud untuk mengunduh daftar file penginstalan dan patch yang didukung untuk RDS Custom for Oracle. Jika Anda menggunakan patch yang tidak dikenal, pembuatan versi mesin kustom (CEV) gagal. Dalam hal ini, hubungi tim dukungan RDS Custom dan minta untuk menambahkan patch yang hilang. Untuk informasi selengkapnya, lihat Langkah 2: Unduh file dan patch instalasi basis data Anda dari Oracle Software Delivery Cloud.
- Anda memiliki akses ke Amazon S3. Anda memerlukan layanan ini karena alasan berikut:
  - Anda mengunggah file penginstalan Oracle Anda ke bucket S3. Anda menggunakan file penginstalan yang diunggah untuk membuat CEV RDS Custom Anda.
  - RDS Custom for Oracle menggunakan skrip yang diunduh dari bucket S3 yang didefinisikan secara internal untuk melakukan tindakan pada instans DB Anda. Skrip ini diperlukan untuk onboarding dan otomatisasi RDS Custom.
  - RDS Custom for Oracle mengunggah file tertentu ke bucket S3 yang terletak di akun pelanggan Anda. Bucket ini menggunakan format penamaan berikut: `do-not-delete-rds-custom-`*`account_id-region-six_character_alphanumeric_string`*. Misalnya, Anda mungkin memiliki bucket bernama `do-not-delete-rds-custom-123456789012-us-east-1-12a3b4`.

  Lihat informasi yang lebih lengkap di Langkah 3: Unggah file instalasi Anda ke Amazon S3 dan Membuat CEV.

- Anda menggunakan kelas instans DB yang tercantum di Dukungan kelas instans DB untuk RDS Custom for Oracle untuk membuat instans DB RDS Custom for Oracle.
- Instans DB RDS Custom for Oracle Anda menjalankan Oracle Linux 7 Update 9 atau lebih tinggi.
- Anda menentukan solid state drive gp2, gp3, atau io1 untuk penyimpanan Amazon EBS. Ukuran penyimpanan maksimum adalah 64 TiB.
- Anda memiliki kunci AWS KMS untuk membuat instans DB RDS Custom for Oracle. Untuk informasi selengkapnya, lihat Langkah 1: Buat atau gunakan kembali kunci enkripsi simetris AWS KMS.
- Anda memiliki peran AWS Identity and Access Management (IAM) dan profil instans yang diperlukan untuk membuat instans DB RDS Custom for Oracle. Untuk informasi selengkapnya, lihat Langkah 4: Konfigurasikan IAM untuk RDS Custom for Oracle.
- Pengguna AWS Identity and Access Management (IAM) yang membuat instans CEV atau RDS Custom DB memiliki izin yang diperlukan untuk IAM,, dan CloudTrail Amazon S3.

  Untuk informasi selengkapnya, lihat Langkah 5: Berikan izin yang diperlukan ke pengguna atau peran IAM Anda.
- Anda menyediakan cloud privat virtual (VPC) dan konfigurasi grup keamanan Anda sendiri. Untuk informasi selengkapnya, lihat Langkah 6: Konfigurasikan VPC Anda untuk RDS Custom for Oracle.
- Anda menyediakan konfigurasi jaringan yang dapat digunakan RDS Custom for Oracle untuk mengakses Layanan AWS lainnya. Untuk persyaratan spesifik, lihat Langkah 4: Konfigurasikan IAM untuk RDS Custom for Oracle.

Batasan umum untuk RDS Custom for Oracle

Batasan berikut berlaku untuk RDS Custom for Oracle:

- Anda tidak dapat mengubah pengidentifikasi instans DB milik Instans DB RDS Custom for Oracle yang sudah ada.
- Anda tidak dapat menentukan arsitektur multipenghuni Oracle untuk rilis basis data selain Oracle Database 19c.
- Anda tidak dapat membuat beberapa basis data Oracle pada satu instans DB RDS Custom for Oracle.
- Anda tidak dapat menghentikan instans DB RDS Custom for Oracle atau instans Amazon EC2 yang mendasarinya. Penagihan untuk instans DB RDS Custom for Oracle tidak dapat dihentikan.

- Anda tidak dapat menggunakan manajemen memori bersama otomatis. RDS Custom for Oracle hanya mendukung manajemen memori otomatis. Untuk informasi selengkapnya, lihat Manajemen Memori Otomatis dalam Panduan Administrator Oracle Database.
- Pastikan untuk tidak mengubah `DB_UNIQUE_NAME` untuk instans DB primer. Mengubah nama ini akan menyebabkan operasi pemulihan menjadi macet.

Untuk batasan khusus terkait memodifikasi instans DB RDS Custom for Oracle, lihat Memodifikasi instans DB RDS Custom for Oracle. Untuk batasan replikasi, lihat Batasan umum untuk replikasi RDS Custom for Oracle.

Batasan CEV dan AMI untuk RDS Custom for Oracle

Batasan berikut berlaku untuk CEV dan AMI RDS Custom for Oracle:

- Anda tidak dapat menyediakan AMI Anda sendiri untuk digunakan dalam CEV RDS Custom for Oracle. Anda dapat menentukan AMI default atau AMI yang sebelumnya telah digunakan oleh CEV RDS Custom for Oracle.

  > **Note**
  >
  > RDS Custom for Oracle merilis AMI default baru ketika kerentanan dan eksposur umum ditemukan. Tidak ada jadwal tetap yang tersedia atau dijamin. RDS Custom for Oracle cenderung menerbitkan AMI default baru setiap 30 hari.

- Anda tidak dapat memodifikasi CEV untuk menggunakan AMI yang berbeda.
- Anda tidak dapat membuat instans CDB dari CEV yang menggunakan mesin `custom-oracle-ee`. CEV harus menggunakan `custom-oracle-ee-cdb`.
- RDS Custom for Oracle saat ini tidak memungkinkan Anda meng-upgrade OS instans DB RDS Custom for Oracle Anda dengan panggilan API RDS. Sebagai solusinya, Anda dapat memperbarui OS Anda secara manual dengan perintah berikut: `sudo yum update --security`.

Setelan yang tidak didukung untuk membuat dan memodifikasi alur kerja

Saat membuat atau memodifikasi instans DB RDS Custom for Oracle, Anda tidak dapat melakukan hal berikut:

- Mengubah jumlah inti dan thread per inti CPU pada kelas instans DB.

- Mengaktifkan penskalaan otomatis penyimpanan.
- Membuat deployment Multi-AZ.

> **Note**
>
> Untuk solusi HA alternatif, lihat artikel blog AWS, Membangun ketersediaan tinggi untuk Amazon RDS Custom for Oracle menggunakan replika baca.

- Mengatur retensi cadangan ke `0`.
- Mengonfigurasi autentikasi Kerberos.
- Menentukan grup parameter atau grup opsi DB Anda sendiri.
- Mengaktifkan Wawasan Performa.
- Mengaktifkan upgrade versi minor otomatis.

Kuota instans DB untuk Akun AWS Anda

Pastikan jumlah gabungan instans DB RDS Custom dan Amazon RDS tidak melebihi batas kuota Anda. Misalnya, jika kuota Amazon RDS adalah 40 instans DB, Anda dapat memiliki 20 instans DB RDS Custom for Oracle dan 20 instans DB Amazon RDS.

Menyiapkan lingkungan Anda untuk Amazon RDS Custom for Oracle

Sebelum Anda membuat instans DB Amazon RDS Custom for Oracle, lakukan tugas-tugas berikut.

Topik

Langkah 1: Buat atau gunakan kembali kunci enkripsi simetris AWS KMS

Kunci yang dikelola pelanggan adalah AWS KMS keys di akun AWS Anda yang Anda buat, miliki, dan kelola. Kunci KMS enkripsi simetris yang dikelola pelanggan diperlukan untuk RDS Custom. Saat Anda membuat instans DB RDS Custom for Oracle, Anda menyediakan pengidentifikasi kunci KMS. Untuk informasi selengkapnya, lihat Mengonfigurasi instans DB untuk Amazon RDS Custom for Oracle.

Anda memiliki opsi berikut:

- Jika Anda memiliki kunci KMS yang dikelola pelanggan yang sudah ada di Akun AWS Anda, Anda dapat menggunakannya dengan RDS Custom. Tidak ada tindakan lebih lanjut yang diperlukan.
- Jika Anda telah membuat kunci KMS enkripsi simetris yang dikelola pelanggan untuk mesin RDS Custom yang berbeda, Anda dapat menggunakan kembali kunci KMS yang sama. Tidak ada tindakan lebih lanjut yang diperlukan.
- Jika Anda tidak memiliki kunci KMS enkripsi simetris yang dikelola pelanggan yang sudah ada di akun Anda, buat kunci KMS dengan mengikuti petunjuk dalam Membuat kunci dalam Panduan Developer AWS Key Management Service.
- Jika Anda membuat CEV atau instans DB RDS Custom, dan kunci KMS Anda berada di Akun AWS yang berbeda, pastikan untuk menggunakan AWS CLI. Anda tidak dapat menggunakan konsol AWS dengan kunci KMS lintas akun.

> **Important**
>
> RDS Custom tidak mendukung kunci KMS yang dikelola AWS.

Pastikan kunci enkripsi simetris Anda memberikan akses ke operasi `kms:Decrypt` dan `kms:GenerateDataKey` ke peran AWS Identity and Access Management (IAM) di profil instans IAM Anda. Jika Anda memiliki kunci enkripsi simetris baru di akun Anda, perubahan tidak diperlukan. Jika tidak, pastikan kebijakan kunci enkripsi simetris Anda memberikan akses ke operasi ini.

Untuk informasi selengkapnya, lihat Langkah 4: Konfigurasikan IAM untuk RDS Custom for Oracle.

Untuk informasi selengkapnya tentang mengonfigurasi IAM untuk RDS Custom for Oracle, lihat Langkah 4: Konfigurasikan IAM untuk RDS Custom for Oracle.

Langkah 2: Mengunduh dan menginstal AWS CLI

AWS memberi Anda antarmuka baris perintah untuk menggunakan fitur RDS Custom. Anda dapat menggunakan AWS CLI versi 1 atau versi 2.

Untuk informasi tentang mengunduh dan menginstal AWS CLI, lihat Menginstal atau memperbarui versi terbaru AWS CLI.

Lewati langkah ini jika salah satu hal berikut ini berlaku:

- Anda berencana untuk mengakses RDS Custom hanya dari AWS Management Console.
- Anda telah mengunduh AWS CLI untuk Amazon RDS atau mesin DB RDS Custom yang berbeda.

Langkah 3: Ekstrak CloudFormation template untuk RDS Custom untuk Oracle

Untuk menyederhanakan pengaturan, kami sangat menyarankan Anda menggunakan AWS CloudFormation template untuk membuat CloudFormation tumpukan. Jika Anda berencana untuk mengonfigurasi IAM dan VPC Anda secara manual, lewati langkah ini.

Topik

- Langkah 3a: Unduh file CloudFormation template
- Langkah 3b: custom-oracle-iam Ekstrak.json
- Langkah 3c: Ekstrak custom-vpc.json

Langkah 3a: Unduh file CloudFormation template

CloudFormation Template adalah deklarasi AWS sumber daya yang membentuk tumpukan. Templat ini disimpan sebagai file JSON.

Untuk mengunduh file CloudFormation template

1. Buka menu konteks (klik kanan) untuk custom-oracle-iamtautan.zip dan pilih Simpan Tautan Sebagai.
2. Simpan file tersebut ke komputer Anda.
3. Ulangi langkah-langkah sebelumnya untuk tautan custom-vpc.zip.

   Jika Anda telah mengonfigurasi VPC untuk RDS Custom, lewati langkah ini.

Langkah 3b: custom-oracle-iam Ekstrak.json

Buka file `custom-oracle-iam.zip` yang Anda unduh, lalu ekstrak `custom-oracle-iam.json` file tersebut. Bagian awal file terlihat seperti berikut ini.

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "EncryptionKey": {
      "Type": "String",
      "Default": "*",
      "Description": "KMS Key ARN for encryption of data managed by RDS Custom and by
 DB Instances."
    }
  },
  "Resources": {
    "RDSCustomInstanceServiceRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "RoleName": { "Fn::Sub": "AWSRDSCustomInstanceRole-${AWS::Region}" },
        "AssumeRolePolicyDocument": {
          "Version": "2012-10-17",
          "Statement": [
            {
              "Action": "sts:AssumeRole",
              "Effect": "Allow",
              "Principal": {
                "Service": "ec2.amazonaws.com"
```

```
            }
          }
        ]
      },...
```

Langkah 3c: Ekstrak custom-vpc.json

> **Note**
>
> Jika Anda sudah mengonfigurasi VPC yang ada untuk RDS Custom for Oracle, lewati langkah ini. Untuk informasi selengkapnya, lihat Konfigurasikan VPC Anda secara manual untuk RDS Custom for Oracle.

Buka file `custom-vpc.zip` yang Anda unduh, lalu ekstrak `custom-vpc.json` file tersebut. Bagian awal file terlihat seperti berikut ini.

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "PrivateVpc": {
      "Type": "AWS::EC2::VPC::Id",
      "Description": "Private VPC Id to use for RDS Custom DB Instances"
    },
    "PrivateSubnets": {
      "Type": "List<AWS::EC2::Subnet::Id>",
      "Description": "Private Subnets to use for RDS Custom DB Instances"
    },
    "RouteTable": {
      "Type": "String",
      "Description": "Route Table that must be associated with the PrivateSubnets and
 used by S3 VPC Endpoint",
      "AllowedPattern": "rtb-[0-9a-z]+"
    }
  },
  "Resources": {
    "DBSubnetGroup": {
      "Type": "AWS::RDS::DBSubnetGroup",
      "Properties": {
        "DBSubnetGroupName": "rds-custom-private",
        "DBSubnetGroupDescription": "RDS Custom Private Network",
        "SubnetIds": {
```

```
          "Ref": "PrivateSubnets"
        }
      }
    },...
```

Langkah 4: Konfigurasikan IAM untuk RDS Custom for Oracle

Anda menggunakan peran IAM atau pengguna IAM (dikenal sebagai entitas IAM) untuk membuat instans DB RDS Custom menggunakan konsol atau AWS CLI. Entitas IAM ini harus memiliki izin yang diperlukan untuk pembuatan instans.

Anda dapat mengonfigurasi IAM menggunakan salah satu CloudFormation atau langkah manual.

> **Important**
>
> Kami sangat menyarankan agar Anda mengonfigurasi RDS Custom for Oracle menggunakan lingkungan AWS CloudFormation. Teknik ini adalah yang termudah dan paling tidak rentan kesalahan.

Topik

- Konfigurasikan IAM menggunakan CloudFormation
- Buat peran IAM dan profil instans secara manual

Konfigurasikan IAM menggunakan CloudFormation

Ketika Anda menggunakan CloudFormation template untuk IAM, itu menciptakan sumber daya yang diperlukan berikut:

- Profil instans bernama `AWSRDSCustomInstanceProfile-`*`region`*
- Peran layanan bernama `AWSRDSCustomInstanceRole-`*`region`*
- Kebijakan akses bernama `AWSRDSCustomIamRolePolicy` yang dilampirkan ke peran layanan

Untuk mengkonfigurasi IAM menggunakan CloudFormation

1. Buka CloudFormation konsol di https://console.aws.amazon.com/cloudformation.
2. Mulai wizard Buat Tumpukan, dan pilih Buat Tumpukan.
3. Pada halaman Buat tumpukan, lakukan hal berikut:

a. Untuk Siapkan templat, pilih Templat sudah siap.
b. Untuk Sumber templat, pilih Unggah file templat.
c. Untuk Pilih file, navigasikan ke, lalu pilih custom-oracle-iam.json.
d. Pilih Berikutnya.

4. Pada halaman Tentukan detail tumpukan, lakukan hal berikut:

a. Untuk Nama tumpukan, masukkan **`custom-oracle-iam`**.
b. Pilih Berikutnya.

5. Pada halaman Konfigurasikan opsi tumpukan, pilih Berikutnya.
6. Pada halaman Tinjauan custom-oracle-iam, lakukan hal berikut:

a. Pilih kotak centang Saya memahami bahwa AWS CloudFormation dapat membuat sumber daya IAM dengan nama kustom.
b. Pilih Kirim.

CloudFormation membuat peran IAM yang dibutuhkan RDS Custom for Oracle. Di panel kiri, saat custom-oracle-iammenampilkan CREATE_COMPLETE, lanjutkan ke langkah berikutnya.

7. Di panel kiri, pilih custom-oracle-iam. Di panel kanan, lakukan hal berikut:

a. Pilih Info tumpukan. *`Tumpukan Anda memiliki ID dalam format`* ***`arn:aws:cloudformation: region: account-no:stack/`*** *`identifier. custom-oracle-iam`*
b. Pilih Sumber daya. Anda akan melihat yang berikut ini:

- Sebuah contoh profil bernama AWSRDSCustomInstanceProfile- ***`region`***
- Peran layanan bernama AWSRDSCustomInstanceRole- ***`wilayah`***

Saat membuat instans DB RDS Custom, Anda harus menyediakan ID profil instans.

Buat peran IAM dan profil instans secara manual

Konfigurasi paling mudah saat Anda gunakan CloudFormation. Namun, Anda juga dapat mengonfigurasi IAM secara manual. Untuk pengaturan manual, lakukan hal berikut:

- Langkah 1: Buat peran IAM AWSRDSCustomInstanceRoleForRdsCustomInstance.

- Langkah 2: Tambahkan kebijakan akses ke AWSRDSCustomInstanceRoleForRdsCustomInstance.
- Langkah 2: Tambahkan kebijakan akses ke AWSRDSCustomInstanceRoleForRdsCustomInstance.
- Langkah 4: Tambahkan AWSRDSCustomInstanceRoleForRdsCustomInstance ke AWSRDSCustomInstanceProfile.

Langkah 1: Buat peran IAM AWSRDSCustomInstanceRoleForRdsCustomInstance

Pada langkah ini, Anda membuat peran menggunakan format penamaan `AWSRDSCustomInstanceRole-region`. Dengan menggunakan kebijakan kepercayaan, Amazon EC2 dapat mengambil peran tersebut. Contoh berikut mengasumsikan bahwa Anda telah mengatur variabel lingkungan `$REGION` ke Wilayah AWS tempat Anda ingin membuat instans DB Anda.

```
aws iam create-role \
  --role-name AWSRDSCustomInstanceRole-$REGION \
  --assume-role-policy-document '{
    "Version": "2012-10-17",
      "Statement": [
        {
          "Action": "sts:AssumeRole",
          "Effect": "Allow",
          "Principal": {
              "Service": "ec2.amazonaws.com"
          }
        }
      ]
    }'
```

Langkah 2: Tambahkan kebijakan akses ke AWSRDSCustomInstanceRoleForRdsCustomInstance

Saat Anda menyematkan kebijakan inline dalam peran IAM, kebijakan inline ini digunakan sebagai bagian dari kebijakan akses (izin) peran. Anda membuat kebijakan `AWSRDSCustomIamRolePolicy` yang mengizinkan Amazon EC2 mengirim dan menerima pesan serta melakukan berbagai tindakan.

Contoh berikut membuat kebijakan akses bernama `AWSRDSCustomIamRolePolicy`, dan menambahkannya ke peran IAM `AWSRDSCustomInstanceRole-region`. Contoh ini mengasumsikan bahwa Anda telah menetapkan variabel lingkungan berikut:

`$REGION`

Tetapkan variabel ini ke Wilayah AWS tempat Anda berencana untuk membuat instans DB Anda.

`$ACCOUNT_ID`

Tetapkan variabel ini ke nomor Akun AWS Anda.

`$KMS_KEY`

Atur variabel ini ke Amazon Resource Name (ARN) milik AWS KMS key yang ingin Anda gunakan untuk instans DB RDS Custom. Untuk menentukan lebih dari satu kunci KMS, tambahkan ke bagian `Resources` dalam ID pernyataan (Sid) 11.

```
aws iam put-role-policy \
  --role-name AWSRDSCustomInstanceRole-$REGION \
  --policy-name AWSRDSCustomIamRolePolicy \
  --policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "1",
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeAssociation",
                "ssm:GetDeployablePatchSnapshotForInstance",
                "ssm:GetDocument",
                "ssm:DescribeDocument",
                "ssm:GetManifest",
                "ssm:GetParameter",
                "ssm:GetParameters",
                "ssm:ListAssociations",
                "ssm:ListInstanceAssociations",
                "ssm:PutInventory",
                "ssm:PutComplianceItems",
                "ssm:PutConfigurePackageResult",
                "ssm:UpdateAssociationStatus",
                "ssm:UpdateInstanceAssociationStatus",
                "ssm:UpdateInstanceInformation",
                "ssm:GetConnectionStatus",
                "ssm:DescribeInstanceInformation",
                "ssmmessages:CreateControlChannel",
                "ssmmessages:CreateDataChannel",
                "ssmmessages:OpenControlChannel",
                "ssmmessages:OpenDataChannel"
            ],
            "Resource": [
```

```
                "*"
            ]
        },
        {
            "Sid": "2",
            "Effect": "Allow",
            "Action": [
                "ec2messages:AcknowledgeMessage",
                "ec2messages:DeleteMessage",
                "ec2messages:FailMessage",
                "ec2messages:GetEndpoint",
                "ec2messages:GetMessages",
                "ec2messages:SendReply"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "3",
            "Effect": "Allow",
            "Action": [
                "logs:PutRetentionPolicy",
                "logs:PutLogEvents",
                "logs:DescribeLogStreams",
                "logs:DescribeLogGroups",
                "logs:CreateLogStream",
                "logs:CreateLogGroup"
            ],
            "Resource": [
                "arn:aws:logs:'$REGION':'$ACCOUNT_ID':log-group:rds-custom-instance*"
            ]
        },
        {
            "Sid": "4",
            "Effect": "Allow",
            "Action": [
                "s3:putObject",
                "s3:getObject",
                "s3:getObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::do-not-delete-rds-custom-*/*"
            ]
```

```
        },
        {
            "Sid": "5",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": [
                        "RDSCustomForOracle/Agent"
                    ]
                }
            }
        },
        {
            "Sid": "6",
            "Effect": "Allow",
            "Action": [
                "events:PutEvents"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "7",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue",
                "secretsmanager:DescribeSecret"
            ],
            "Resource": [
                "arn:aws:secretsmanager:'$REGION':'$ACCOUNT_ID':secret:do-not-delete-
rds-custom-*"
            ]
        },
        {
           "Sid": "8",
           "Effect": "Allow",
           "Action": [
```

```
            "s3:ListBucketVersions"
          ],
          "Resource": [
            "arn:aws:s3:::do-not-delete-rds-custom-*"
          ]
        },
        {
           "Sid": "9",
           "Effect": "Allow",
           "Action": "ec2:CreateSnapshots",
           "Resource": [
               "arn:aws:ec2:*:*:instance/*",
               "arn:aws:ec2:*:*:volume/*"
           ],
           "Condition": {
               "StringEquals": {
                   "ec2:ResourceTag/AWSRDSCustom": "custom-oracle"
               }
           }
         },
         {
           "Sid": "10",
           "Effect": "Allow",
           "Action": "ec2:CreateSnapshots",
           "Resource": [
               "arn:aws:ec2:*::snapshot/*"
           ]
         },
         {
           "Sid": "11",
           "Effect": "Allow",
           "Action": [
             "kms:Decrypt",
             "kms:GenerateDataKey"
           ],
           "Resource": [
             "arn:aws:kms:'$REGION':'$ACCOUNT_ID':key/'$KMS_KEY'"
           ]
         },
         {
           "Sid": "12",
           "Effect": "Allow",
           "Action": "ec2:CreateTags",
           "Resource": "*",
```

```
            "Condition": {
                "StringLike": {
                    "ec2:CreateAction": [
                        "CreateSnapshots"
                    ]
                }
            }
        }
    ]
}'
```

Langkah 3: Buat profil contoh Kustom RDS AWSRDSCustomInstanceProfile

Profil instans adalah kontainer yang menyertakan peran IAM tunggal. RDS Custom menggunakan profil instans untuk meneruskan peran ke instans.

Jika Anda menggunakan CLI untuk membuat peran, Anda dapat membuat peran dan profil instans sebagai tindakan terpisah, dengan nama yang mungkin berbeda. Buat profil instans IAM Anda sebagai berikut, dengan memberinya nama menggunakan format `AWSRDSCustomInstanceProfile-`*`region`*. Contoh berikut mengasumsikan bahwa Anda telah mengatur variabel lingkungan `$REGION` ke Wilayah AWS tempat Anda ingin membuat instans DB Anda.

```
aws iam create-instance-profile \
    --instance-profile-name AWSRDSCustomInstanceProfile-$REGION
```

Langkah 4: Tambahkan AWSRDSCustomInstanceRoleForRdsCustomInstance ke AWSRDSCustomInstanceProfile

Tambahkan peran IAM Anda ke profil instans yang sebelumnya Anda buat. Contoh berikut mengasumsikan bahwa Anda telah mengatur variabel lingkungan `$REGION` ke Wilayah AWS tempat Anda ingin membuat instans DB Anda.

```
aws iam add-role-to-instance-profile \
    --instance-profile-name AWSRDSCustomInstanceProfile-$REGION \
    --role-name AWSRDSCustomInstanceRole-$REGION
```

Langkah 5: Berikan izin yang diperlukan ke pengguna atau peran IAM Anda

Pastikan bahwa prinsipal IAM (pengguna atau peran) yang membuat CEV atau instans DB RDS Custom memiliki salah satu kebijakan berikut:

- Kebijakan `AdministratorAccess`
- Kebijakan `AmazonRDSFullAccess` dengan izin yang diperlukan untuk Amazon S3 dan AWS KMS, pembuatan CEV, dan pembuatan instans DB

Topik

- Izin IAM yang diperlukan untuk Amazon S3 dan AWS KMS
- Izin IAM yang diperlukan untuk membuat CEV
- Izin IAM yang diperlukan untuk membuat instans DB dari CEV

Izin IAM yang diperlukan untuk Amazon S3 dan AWS KMS

Untuk membuat CEV atau instans DB RDS Custom for Oracle, prinsipal IAM Anda perlu mengakses Amazon S3 dan AWS KMS. Contoh kebijakan JSON berikut memberikan izin yang diperlukan.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CreateS3Bucket",
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:PutBucketPolicy",
                "s3:PutBucketObjectLockConfiguration",
                "s3:PutBucketVersioning"
            ],
            "Resource": "arn:aws:s3:::do-not-delete-rds-custom-*"
        },
        {
            "Sid": "CreateKmsGrant",
            "Effect": "Allow",
            "Action": [
                "kms:CreateGrant",
                "kms:DescribeKey"
            ],
            "Resource": "*"
        }
    ]
}
```

Untuk informasi selengkapnya tentang izin `kms:CreateGrant`, lihat Manajemen AWS KMS key.

Izin IAM yang diperlukan untuk membuat CEV

Untuk membuat CEV, prinsipal IAM Anda memerlukan izin tambahan berikut:

```
s3:GetObjectAcl
s3:GetObject
s3:GetObjectTagging
s3:ListBucket
mediaimport:CreateDatabaseBinarySnapshot
```

Contoh kebijakan JSON berikut memberikan izin tambahan yang diperlukan untuk mengakses bucket *my-custom-installation-files*dan isinya.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AccessToS3MediaBucket",
            "Effect": "Allow",
            "Action": [
                "s3:GetObjectAcl",
                "s3:GetObject",
                "s3:GetObjectTagging",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::my-custom-installation-files",
                "arn:aws:s3:::my-custom-installation-files/*"
            ]
        },
        {
            "Sid": "PermissionForByom",
            "Effect": "Allow",
            "Action": [
                "mediaimport:CreateDatabaseBinarySnapshot"
            ],
            "Resource": "*"
        }
    ]
}
```

Anda dapat memberikan izin serupa untuk Amazon S3 ke akun pemanggil menggunakan kebijakan bucket S3.

Izin IAM yang diperlukan untuk membuat instans DB dari CEV

Untuk membuat instans DB RDS Custom for Oracle dari CEV yang ada, prinsipal IAM memerlukan izin tambahan berikut.

```
iam:SimulatePrincipalPolicy
cloudtrail:CreateTrail
cloudtrail:StartLogging
```

Contoh kebijakan JSON berikut memberikan izin yang diperlukan untuk memvalidasi peran IAM dan mencatat log informasi ke AWS CloudTrail.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ValidateIamRole",
            "Effect": "Allow",
            "Action": "iam:SimulatePrincipalPolicy",
            "Resource": "*"
        },
        {
            "Sid": "CreateCloudTrail",
            "Effect": "Allow",
            "Action": [
                "cloudtrail:CreateTrail",
                "cloudtrail:StartLogging"
            ],
            "Resource": "arn:aws:cloudtrail:*:*:trail/do-not-delete-rds-custom-*"
        }
    ]
}
```

Langkah 6: Konfigurasikan VPC Anda untuk RDS Custom for Oracle

Instans DB RDS Custom Anda berada di cloud privat virtual (VPC) yang didasarkan pada layanan Amazon VPC, seperti instans Amazon EC2 atau instans Amazon RDS. Anda menyediakan dan mengonfigurasi VPC Anda sendiri. Tidak seperti RDS Custom for SQL Server, RDS Custom for

Oracle tidak membuat daftar kontrol akses atau grup keamanan. Anda harus melampirkan grup keamanan, subnet, dan tabel rute Anda sendiri.

Anda dapat mengonfigurasi virtual private cloud (VPC) menggunakan salah satu CloudFormation atau proses manual.

> **Important**
>
> Kami sangat menyarankan agar Anda mengonfigurasi RDS Custom for Oracle menggunakan lingkungan AWS CloudFormation. Teknik ini adalah yang termudah dan paling tidak rentan kesalahan.

Topik

- Konfigurasikan VPC Anda menggunakan CloudFormation (disarankan)
- Konfigurasikan VPC Anda secara manual untuk RDS Custom for Oracle

Konfigurasikan VPC Anda menggunakan CloudFormation (disarankan)

Jika Anda telah mengonfigurasi VPC Anda untuk mesin RDS Custom yang berbeda, dan ingin menggunakan kembali VPC yang ada, lewati langkah ini. Bagian ini mengasumsikan hal berikut:

- Anda telah digunakan CloudFormation untuk membuat profil dan peran instans IAM Anda.
- Anda mengetahui ID tabel rute Anda.

  Agar menjadi privat, instans DB harus berada dalam subnet privat. Agar menjadi privat, subnet tidak boleh dikaitkan dengan tabel rute yang memiliki gateway internet default. Untuk informasi selengkapnya, lihat Mengonfigurasi tabel rute dalam Panduan Pengguna Amazon VPC.

Ketika Anda menggunakan CloudFormation template untuk VPC Anda, itu menciptakan sumber daya berikut:

- VPC privat
- Grup subnet bernama `rds-custom-private`
- Titik akhir VPC berikut, yang digunakan instans DB Anda untuk berkomunikasi dengan Layanan AWS dependen:
  - `com.amazonaws.region.ec2messages`

- `com.amazonaws.region.events`
- `com.amazonaws.region.logs`
- `com.amazonaws.region.monitoring`
- `com.amazonaws.region.s3`
- `com.amazonaws.region.secretsmanager`
- `com.amazonaws.region.ssm`
- `com.amazonaws.region.ssmmessages`

> **Note**
>
> Untuk pengaturan jaringan yang kompleks dengan akun yang ada, sebaiknya Anda mengonfigurasi akses ke layanan dependen secara manual jika akses belum ada. Untuk informasi selengkapnya, lihat Pastikan VPC Anda dapat mengakses Layanan AWS dependen.

Untuk mengkonfigurasi VPC Anda menggunakan CloudFormation

1. Buka CloudFormation konsol di https://console.aws.amazon.com/cloudformation.
2. Mulai wizard Buat Tumpukan, dan pilih Buat Tumpukan lalu Dengan sumber daya baru (standar).
3. Pada halaman Buat tumpukan, lakukan hal berikut:
   a. Untuk Siapkan templat, pilih Templat sudah siap.
   b. Untuk Sumber templat, pilih Unggah file templat.
   c. Untuk Pilih file, navigasikan ke, lalu pilih `custom-vpc.json`.
   d. Pilih Berikutnya.
4. Pada halaman Tentukan detail tumpukan, lakukan hal berikut:
   a. Untuk Nama tumpukan, masukkan **`custom-vpc`**.
   b. Untuk Parameter, pilih subnet privat yang akan digunakan untuk instans DB RDS Custom.
   c. Pilih ID VPC privat yang akan digunakan untuk instans DB RDS Custom.
   d. Masukkan tabel rute yang dikaitkan dengan subnet privat.
   e. Pilih Berikutnya.
5. Pada halaman Konfigurasikan opsi tumpukan, pilih Berikutnya.

6. Pada halaman Tinjau custom-vpc, pilih Kirim.

   CloudFormation mengkonfigurasi VPC pribadi Anda. Di panel kiri, ketika custom-vpc menunjukkan CREATE_COMPLETE, lanjutkan ke langkah berikutnya.

7. (Opsional) Tinjau detail VPC Anda. Di panel Tumpukan, pilih custom-vpc. Di panel kanan, lakukan hal berikut:

   a. Pilih Info tumpukan. Tumpukan Anda memiliki ID dalam format arn:aws:cloudformation:***region***:***account-no***:stack/custom-vpc/***identifier***.

   b. Pilih Sumber daya. *Anda akan melihat grup subnet bernama* ***rds-custom-private****dan beberapa titik akhir VPC yang menggunakan* ***format*** *penamaan vpce- string.* Setiap titik akhir sesuai dengan Layanan AWS yang akan berkomunikasi dengan RDS Custom. Untuk informasi selengkapnya, lihat Pastikan VPC Anda dapat mengakses Layanan AWS dependen.

   c. Pilih Parameter. Anda akan melihat subnet privat, VPC privat, dan tabel rute yang Anda tentukan saat Anda membuat tumpukan. Saat Anda membuat instans DB, Anda harus menyediakan ID VPC dan grup subnet.

Konfigurasikan VPC Anda secara manual untuk RDS Custom for Oracle

Sebagai alternatif untuk mengotomatiskan pembuatan VPC dengan AWS CloudFormation, Anda dapat mengonfigurasi VPC Anda secara manual. Opsi ini mungkin yang terbaik jika Anda memiliki pengaturan jaringan kompleks yang menggunakan sumber daya yang ada.

Topik

- Pastikan VPC Anda dapat mengakses Layanan AWS dependen
- Konfigurasikan layanan metadata instans

Pastikan VPC Anda dapat mengakses Layanan AWS dependen

RDS Custom mengirimkan komunikasi dari instans DB Anda ke Layanan AWS lain. Pastikan layanan berikut dapat diakses dari subnet tempat Anda membuat instans DB RDS Custom for Oracle:

- Amazon CloudWatch
- CloudWatch Log Amazon
- CloudWatch Acara Amazon

- Amazon EC2
- Amazon EventBridge
- Amazon S3
- AWS Secrets Manager
- AWS Systems Manager

Jika akses ke Layanan AWS di atas saat ini tidak ada, konfigurasikan titik akhir VPC berikut:

- `com.amazonaws.region.ec2messages`
- `com.amazonaws.region.events`
- `com.amazonaws.region.logs`
- `com.amazonaws.region.monitoring`
- `com.amazonaws.region.s3`
- `com.amazonaws.region.secretsmanager`
- `com.amazonaws.region.ssmmessages`

Jika RDS Custom tidak dapat berkomunikasi dengan layanan yang diperlukan, RDS Custom akan menerbitkan peristiwa berikut:

```
Database instance in incompatible-network. SSM Agent connection not available. Amazon
 RDS can't connect to the dependent AWS services.
```

Untuk menghindari kesalahan `incompatible-network`, pastikan komponen VPC yang diperlukan dalam komunikasi antara instans DB RDS Custom Anda dan Layanan AWS memenuhi persyaratan berikut:

- Instans DB dapat membuat koneksi keluar pada port 443 ke Layanan AWS lainnya.
- VPC mengizinkan respons masuk untuk permintaan yang berasal dari instans DB RDS Custom Anda.
- RDS Custom dapat secara tepat me-resolve nama domain titik akhir untuk masing-masing Layanan AWS.

RDS Custom mengandalkan konektivitas AWS Systems Manager untuk otomatisasi. Untuk informasi tentang cara mengonfigurasi titik akhir VPC, lihat Membuat titik akhir VPC untuk Systems Manager.

Untuk daftar titik akhir di setiap Wilayah, lihat Titik akhir dan kuota AWS Systems Manager dalam Referensi Umum Amazon Web Services.

Jika Anda sudah mengonfigurasi VPC untuk mesin DB RDS Custom yang berbeda, Anda dapat menggunakan kembali VPC tersebut dan melewati proses ini.

Konfigurasikan layanan metadata instans

Pastikan instans Anda dapat melakukan hal berikut:

- Mengakses layanan metadata instans menggunakan Instance Metadata Service Version 2 (IMDSv2).
- Memungkinkan komunikasi keluar melalui port 80 (HTTP) ke alamat IP tautan IMDS.
- Meminta metadata instans dari `http://169.254.169.254`, tautan IMDSv2.

Untuk informasi selengkapnya, lihat Gunakan IMDSv2 dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Otomatisasi RDS Custom for Oracle menggunakan IMDSv2 secara default, dengan mengatur `HttpTokens=enabled` di instans Amazon EC2 yang mendasarinya. Namun, Anda dapat menggunakan IMDSv1 jika ingin. Untuk informasi selengkapnya, lihat Konfigurasikan opsi metadata instans dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Menggunakan versi mesin kustom untuk Amazon RDS Custom for Oracle

Versi mesin kustom (CEV) untuk Amazon RDS Custom for Oracle adalah snapshot volume biner dari mesin basis data dan Amazon Machine Image (AMI) tertentu. Secara default, RDS Custom for Oracle menggunakan AMI terbaru yang tersedia yang dikelola oleh RDS Custom, tetapi Anda dapat menentukan AMI yang digunakan dalam CEV sebelumnya. Anda menyimpan file instalasi basis data di Amazon S3. RDS Custom menggunakan file instalasi dan AMI untuk membuat CEV Anda.

Topik

Persiapan membuat CEV

Untuk membuat CEV, akses file dan patch instalasi yang disimpan di bucket Amazon S3 Anda untuk salah satu rilis berikut:

- Oracle Database 19c
- Oracle Database 18c
- Oracle Database 12c Rilis 2 (12.2)
- Oracle Database 12c Rilis 1 (12.1)

Misalnya, Anda dapat menggunakan RU/RUR April 2021 untuk Oracle Database 19c atau kombinasi file dan patch instalasi yang valid. Untuk informasi selengkapnya tentang versi dan Wilayah yang didukung oleh RDS Custom for Oracle, lihat RDS Custom dengan RDS for Oracle.

Topik

Langkah 1 (Opsional): Unduh templat manifes

Manifes CEV adalah dokumen JSON yang menyertakan daftar file .zip instalasi basis data untuk CEV Anda. Untuk membuat CEV, lakukan hal berikut:

1. Identifikasi file instalasi basis data Oracle yang ingin Anda sertakan dalam CEV Anda.
2. Unduh file instalasi.
3. Buat manifes JSON yang mencantumkan file instalasi.

RDS Custom for Oracle menyediakan templat manifes JSON dengan file .zip yang kami rekomendasikan untuk setiap rilis Oracle Database yang didukung. Misalnya, templat berikut adalah untuk 19.17.0.0.0 RU.

```
{
    "mediaImportTemplateVersion": "2020-08-14",
    "databaseInstallationFileNames": [
        "V982063-01.zip"
    ],
    "opatchFileNames": [
        "p6880880_190000_Linux-x86-64.zip"
    ],
    "psuRuPatchFileNames": [
        "p34419443_190000_Linux-x86-64.zip",
        "p34411846_190000_Linux-x86-64.zip"
    ],
    "otherPatchFileNames": [
        "p28852325_190000_Linux-x86-64.zip",
        "p29997937_190000_Linux-x86-64.zip",
        "p31335037_190000_Linux-x86-64.zip",
        "p32327201_190000_Linux-x86-64.zip",
        "p33613829_190000_Linux-x86-64.zip",
        "p34006614_190000_Linux-x86-64.zip",
        "p34533061_190000_Linux-x86-64.zip",
        "p34533150_190000_Generic.zip",
        "p28730253_190000_Linux-x86-64.zip",
        "p29213893_1917000DBRU_Generic.zip",
        "p33125873_1917000DBRU_Linux-x86-64.zip",
```

```
        "p34446152_1917000DBRU_Linux-x86-64.zip"
    ]
}
```

Setiap templat memiliki readme terkait yang mencakup instruksi untuk mengunduh patch, URL untuk file .zip, dan checksum file. Anda dapat menggunakan templat ini apa adanya atau memodifikasinya dengan patch Anda sendiri. Untuk meninjau template, unduh custom-oracle-manifest.zip ke disk lokal Anda dan kemudian buka dengan aplikasi pengarsipan file. Untuk informasi selengkapnya, lihat Langkah 5: Siapkan manifes CEV.

Langkah 2: Unduh file dan patch instalasi basis data Anda dari Oracle Software Delivery Cloud

Ketika telah mengidentifikasi file instalasi yang diinginkan untuk CEV Anda, unduh ke sistem lokal Anda. File dan patch instalasi Oracle Database di-host di Oracle Software Delivery Cloud. Setiap CEV membutuhkan rilis dasar, seperti Oracle Database 19c atau Oracle Database 12c Rilis 2 (12.2), dan daftar patch opsional.

Cara mengunduh file instalasi basis data untuk Oracle Database

1. Buka https://edelivery.oracle.com/ dan masuk.
2. Masukkan **`Oracle Database Enterprise Edition`** di kotak dan pilih Search.
3. Pilih salah satu rilis dasar berikut:
   - DLP: Oracle Database Enterprise Edition 19.3.0.0.0 (Oracle Database Enterprise Edition).
   - Pilih DLP: Oracle Database 12c Enterprise Edition 18.0.0.0.0 (Oracle Database Enterprise Edition).
   - Pilih DLP: Oracle Database 12c Enterprise Edition 12.2.0.1.0 (Oracle Database Enterprise Edition).
   - Pilih DLP: Oracle Database 12c Enterprise Edition 12.1.0.2.0 (Oracle Database Enterprise Edition).
4. Pilih Continue.
5. Hapus kotak centang Download Queue.
6. Pilih opsi yang sesuai dengan rilis dasar Anda:
   - Oracle Database 19.3.0.0.0 - Long Term Release.
   - Oracle Database 18.0.0.0.0
   - Oracle Database 12.2.0.1.0.

- Oracle Database 12.1.0.2.0.

7. Pilih Linux x86-64 di Platform/Languages.
8. Pilih Continue, lalu tandatangani pengabaian.
9. Pilih file .zip yang sesuai dengan rilis basis data Anda:

Rilis basis data	File zip	SHA-256 hash
19c	V982063-01.zip	`BA8329C757133DA313ED3B6D7F86C5AC42CD 9970A28BF2E6233F3235233AA8D8`
18c	V978967-01.zip	`C96A4FD768787AF98272008833FE10B17269 1CF84E42816B138C12D4DE63AB96`
12.2	V839960-01.zip	`96ED97D21F15C1AC0CCE3749DA6C3DAC7059 BB60672D76B008103FC754D22DDE`
12.1	V46095-01 _1of2.zip V46095-01 _2of2.zip	`31FDC2AF41687B4E547A3A18F796424D8C1A F36406D2160F65B0AF6A9CD47355` untuk V46095-01 _1of2.zip `03DA14F5E875304B28F0F3BB02AF0EC33227 885B99C9865DF70749D1E220ACCD` untuk V46095-01 _2of2.zip

10. Unduh patch Oracle yang diinginkan dari `updates.oracle.com` atau `support.oracle.com` ke sistem lokal Anda. Anda dapat menemukan URL untuk patch di lokasi berikut:

    - File readme dalam file.zip yang Anda unduh di Langkah 1 (Opsional): Unduh templat manifes
    - Patch yang tercantum di setiap Pembaruan Rilis (RU) di Catatan rilis untuk Amazon Relational Database Service (Amazon RDS) for Oracle

Langkah 3: Unggah file instalasi Anda ke Amazon S3

Unggah file instalasi dan patch Oracle ke Amazon S3 menggunakan AWS CLI. Bucket S3 yang berisi file instalasi harus berada di Wilayah AWS yang sama dengan CEV Anda.

Contoh di bagian ini menggunakan placeholder berikut:

- *install-or-patch-file.zip* – File media instalasi Oracle. Misalnya, p32126828_190000_Linux-x86-64.zip adalah patch.
- *my-custom-installation-files* – Bucket Amazon S3 yang ditunjuk untuk file instalasi yang Anda unggah.
- *123456789012/cev1* – Prefiks opsional di bucket Amazon S3 Anda.
- *source-bucket* – Bucket Amazon S3 tempat Anda dapat mengatur file secara opsional.

Topik

Langkah 3a: Verifikasi bahwa bucket S3 Anda berada di Wilayah AWS yang benar

Verifikasi bahwa bucket S3 Anda berada di AWS Wilayah tempat Anda berencana untuk menjalankan perintah `create-custom-db-engine-version`.

```
aws s3api get-bucket-location --bucket my-custom-installation-files
```

Langkah 3b: Pastikan kebijakan bucket S3 Anda memiliki izin yang benar

Anda dapat membuat CEV dari awal atau dari CEV sumber. Jika Anda berencana membuat CEV baru dari CEV sumber, pastikan kebijakan bucket S3 Anda memiliki izin yang benar:

1. Identifikasi bucket S3 yang dipesan oleh RDS Custom. Nama bucket memiliki format `do-not-delete-rds-custom-account-region-string`. Misalnya, bucket mungkin diberi nama `do-not-delete-rds-custom-123456789012-us-east-1-abc123EXAMPLE`.
2. Pastikan izin berikut ditambahkan ke kebijakan bucket S3 Anda. Ganti `do-not-delete-rds-custom-123456789012-us-east-1-abc123EXAMPLE` dengan nama bucket Anda.

```
{
    "Sid": "AWSRDSCustomForOracleCustomEngineVersionGetObject",
    "Effect": "Allow",
```

```
    "Principal": {
        "Service": "custom.rds.amazonaws.com"
    },
    "Action": [
        "s3:GetObject",
        "s3:GetObjectTagging"
    ],
    "Resource": "arn:aws:s3:::do-not-delete-rds-custom-123456789012-us-
east-1-abc123EXAMPLE/CustomEngineVersions/*"
}, ...
```

Langkah 3c: Unggah file Anda menggunakan perintah cp atau sinkronisasi

Pilih salah satu opsi berikut:

- Gunakan `aws s3 cp` untuk mengunggah satu file .zip.

  Unggah setiap file.zip instalasi secara terpisah. Jangan gabungkan file.zip menjadi satu file.zip.
- Gunakan `aws s3 sync` untuk mengunggah direktori.

Example

Contoh berikut mengunggah *install-or-patch-file.zip* ke folder *123456789012/cev1* di bucket Amazon S3 RDS Custom. Jalankan perintah `aws s3` terpisah untuk setiap .zip yang ingin Anda unggah.

Untuk Linux, macOS, atau Unix:

```
aws s3 cp install-or-patch-file.zip \
    s3://my-custom-installation-files/123456789012/cev1/
```

Untuk Windows:

```
aws s3 cp install-or-patch-file.zip ^
    s3://my-custom-installation-files/123456789012/cev1/
```

Example

Contoh berikut mengunggah file di folder *cev1* lokal Anda ke folder *123456789012/cev1* di bucket Amazon S3 Anda.

Untuk Linux, macOS, atau Unix:

```
aws s3 sync cev1 \
    s3://my-custom-installation-files/123456789012/cev1/
```

Untuk Windows:

```
aws s3 sync cev1 ^
    s3://my-custom-installation-files/123456789012/cev1/
```

Example

Contoh berikut mengunggah semua file *source-bucket* ke folder *123456789012/cev1* di bucket Amazon S3 Anda.

Untuk Linux, macOS, atau Unix:

```
aws s3 sync s3://source-bucket/ \
    s3://my-custom-installation-files/123456789012/cev1/
```

Untuk Windows:

```
aws s3 sync s3://source-bucket/ ^
    s3://my-custom-installation-files/123456789012/cev1/
```

Langkah 3d: Buat daftar file di bucket S3

Contoh berikut menggunakan perintah `s3 ls` untuk membuat daftar file di bucket Amazon S3 RDS Custom Anda.

```
aws s3 ls \
    s3://my-custom-installation-files/123456789012/cev1/
```

Langkah 4 (Opsional): Bagikan media instalasi Anda di S3 di seluruh Akun AWS

Untuk bagian ini, bucket Amazon S3 berisi file instalasi Oracle yang Anda unggah adalah bucket media Anda. Organisasi Anda mungkin menggunakan beberapa Akun AWS dalam satu Wilayah AWS. Jika demikian, Anda mungkin ingin menggunakan satu Akun AWS untuk mengisi bucket media dan Akun AWS yang lain untuk membuat CEV. Jika Anda tidak ingin membagikan bucket media, lewati ke bagian berikutnya.

Bagian ini mengasumsikan hal berikut:

- Anda dapat mengakses akun yang membuat bucket media Anda dan akun lain tempat Anda ingin membuat CEV.
- Anda bermaksud membuat CEV hanya dalam satu Wilayah AWS. Jika Anda ingin menggunakan beberapa Wilayah, buat bucket media di setiap Wilayah.
- Anda menggunakan CLI. Jika Anda menggunakan konsol Amazon S3, sesuaikan langkah-langkah berikut.

Cara mengonfigurasi bucket media Anda untuk dibagikan ke seluruh Akun AWS

1. Masuk ke Akun AWS yang berisi bucket S3 tempat Anda mengunggah media instalasi Anda.
2. Mulai dengan templat kebijakan JSON kosong atau kebijakan yang sudah ada yang dapat Anda sesuaikan.

   Perintah berikut mengambil kebijakan yang ada dan menyimpannya sebagai *my-policy.json*. Dalam contoh ini, bucket S3 yang berisi file instalasi Anda diberi nama *oracle-media-bucket*.

   ```
   aws s3api get-bucket-policy \
       --bucket oracle-media-bucket \
       --query Policy \
       --output text > my-policy.json
   ```

3. Edit izin bucket media sebagai berikut:

   - Dalam elemen `Resource` templat Anda, tentukan bucket S3 tempat Anda mengunggah file instalasi Oracle Database.
   - Dalam elemen `Principal`, tentukan ARN untuk semua Akun AWS yang ingin Anda gunakan untuk membuat CEV. Anda dapat menambahkan root, pengguna, atau peran ke daftar izin bucket S3. Untuk informasi selengkapnya, lihat Pengidentifikasi IAM di Panduan Pengguna AWS Identity and Access Management.

   ```
   {
       "Version": "2008-10-17",
       "Statement": [
           {
               "Sid": "GrantAccountsAccess",
   ```

```
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::account-1:root",
                    "arn:aws:iam::account-2:user/user-name-with-path",
                    "arn:aws:iam::account-3:role/role-name-with-path",
                    ...
                ]
            },
            "Action": [
                "s3:GetObject",
                "s3:GetObjectAcl",
                "s3:GetObjectTagging",
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::oracle-media-bucket",
                "arn:aws:s3:::oracle-media-bucket/*"
            ]
        }
    ]
}
```

4. Lampirkan kebijakan tersebut ke bucket media Anda.

   Dalam contoh berikut, *oracle-media-bucket* adalah nama bucket S3 yang berisi file instalasi Anda, dan *my-policy.json adalah nama file JSON* Anda.

   ```
   aws s3api put-bucket-policy \
       --bucket oracle-media-bucket \
       --policy file://my-policy.json
   ```

5. Masuk ke Akun AWS tempat Anda ingin membuat CEV.

6. Verifikasi bahwa akun ini dapat mengakses bucket media di Akun AWS yang membuatnya.

   ```
   aws s3 ls --query "Buckets[].Name"
   ```

   Untuk informasi selengkapnya, lihat aws s3 ls di Referensi Perintah AWS CLI.

7. Buat CEV dengan mengikuti langkah-langkah di Membuat CEV.

Langkah 5: Siapkan manifes CEV

Sebuah manifes CEV adalah dokumen JSON yang mencakup hal-hal berikut ini:

- (Wajib) Daftar file .zip instalasi yang Anda unggah ke Amazon S3. RDS Custom menerapkan patch sesuai urutan yang dicantumkan di manifes.
- (Opsional) Parameter instalasi yang menetapkan nilai nondefault untuk basis Oracle, beranda Oracle, serta ID dan nama pengguna dan grup UNIX/Linux. Ketahui bahwa Anda tidak dapat memodifikasi parameter instalasi untuk CEV yang ada atau instans DB yang ada. Anda juga tidak dapat meningkatkan dari satu CEV ke CEV lain ketika parameter instalasi memiliki pengaturan yang berbeda.

Untuk contoh manifes CEV, lihat templat JSON yang Anda unduh di Langkah 1 (Opsional): Unduh templat manifes. Anda juga dapat meninjau sampel di Contoh manifes CEV.

Topik

- Bidang JSON di manifes CEV
- Membuat manifes CEV
- Contoh manifes CEV

Bidang JSON di manifes CEV

Tabel berikut menjelaskan bidang JSON dalam manifes.

Bidang JSON di manifes CEV

Bidang JSON	Deskripsi
`MediaImportTemplateVersion`	Versi manifes CEV. Format tanggal adalah `YYYY-MM-DD` .
`databaseInstallationFileNames`	Daftar file instalasi yang dipesan untuk basis data.
`opatchFileNames`	Daftar penginstal OPatch yang digunakan untuk mesin DB Oracle. Hanya satu nilai yang valid. Nilai untuk `opatchFileNames` harus dimulai dengan `p6880880_` .
`psuRuPatchFileNames`	Patch PSU dan RU untuk basis data ini.

Bidang JSON	Deskripsi
	Important Jika Anda menyertakan `psuRuPatchFileNames` , `opatchFileNames` diperlukan. Nilai untuk `opatchFileNames` harus dimulai dengan `p6880880_` .
`OtherPatchFileNames`	Patch yang tidak ada dalam daftar patch PSU dan RU. RDS Custom menerapkan patch ini setelah menerapkan patch PSU dan RU. **Important** Jika Anda menyertakan `OtherPatchFileNames` , `opatchFileNames` diperlukan. Nilai untuk `opatchFileNames` harus dimulai dengan `p6880880_` .

Bidang JSON	Deskripsi
`installationParameters`	Pengaturan nondefault untuk basis Oracle, beranda Oracle, serta ID dan nama pengguna dan grup UNIX/Linux. Anda dapat mengatur parameter berikut: oracleBase Direktori tempat biner Oracle Anda diinstal. Ini adalah titik pemasangan volume biner yang menyimpan file Anda. Direktori basis Oracle dapat mencakup beberapa beranda Oracle. Misalnya, jika `/home/oracle/oracle.19.0.0.0.ru-2020-04.rur-2020-04.r1.EE.1` adalah salah satu direktori beranda Oracle Anda, `/home/oracle` adalah direktori basis Oracle. Direktori basis Oracle yang ditentukan pengguna bukan tautan simbolis. Jika Anda tidak menentukan basis Oracle, direktori default adalah `/rdsdbbin` . oracleHome Direktori tempat biner basis data Oracle Anda diinstal. Misalnya, jika Anda menentukan `/home/oracle/` sebagai basis Oracle, Anda dapat menentukan `/home/oracle/oracle.19.0.0.0.ru-2020-04.rur-2020-04.r1.EE.1/` sebagai beranda Oracle Anda. Direktori beranda Oracle yang ditentukan pengguna bukan tautan simbolis. Nilai beranda Oracle direferensikan oleh variabel lingkungan `$ORACLE_HOME` . Jika Anda tidak menentukan beranda Oracle, format penamaan default adalah `/rdsdbbin/oracle.` *`major-engine-version`* `.custom.r1.` *`engine-edition`* `.1`. unixUname Nama pengguna UNIX yang memiliki perangkat lunak Oracle. RDS Custom mengasumsikan pengguna ini saat menjalankan perintah basis data lokal. Jika Anda menentukan `unixUid`

Bidang JSON	Deskripsi
	dan `unixUname` , RDS Custom membuat pengguna jika tidak ada, kemudian menetapkan UID ke pengguna jika tidak sama dengan UID awal. Nama pengguna default adalah `rdsdb`. unixUid ID (UID) pengguna UNIX yang memiliki perangkat lunak Oracle. Jika Anda menentukan `unixUid` dan `unixUname` , RDS Custom membuat pengguna jika tidak ada, kemudian menetapkan UID ke pengguna jika tidak sama dengan UID awal. UID default adalah `61001`. Ini adalah UID pengguna `rdsdb`. unixGroupName Nama grup UNIX. Pengguna UNIX yang memiliki perangkat lunak Oracle termasuk dalam grup ini. Nama grup default adalah `rdsdb`. unixGroupId ID grup UNIX tempat pengguna UNIX berada. ID grup default adalah `1000`. Ini adalah ID grup `rdsdb`.

Setiap rilis Oracle Database memiliki daftar file instalasi yang didukung yang berbeda. Saat membuat manifes CEV Anda, pastikan Anda hanya menentukan file yang didukung oleh RDS Custom for Oracle. Jika tidak, pembuatan CEV gagal dengan kesalahan. Semua patch yang tercantum dalam Catatan rilis untuk Amazon Relational Database Service (Amazon RDS) for Oracle didukung.

Membuat manifes CEV

Cara membuat manifes CEV

1. Buat daftar semua file instalasi yang berencana Anda terapkan, dalam urutan yang ingin Anda terapkan.

2. Korelasikan file instalasi dengan bidang JSON yang dijelaskan dalam Bidang JSON di manifes CEV.
3. Lakukan salah satu dari langkah berikut:
   - Buat manifes CEV sebagai file teks JSON.
   - Edit templat manifes CEV saat Anda membuat CEV di konsol. Untuk informasi selengkapnya, lihat Membuat CEV.

Contoh manifes CEV

Contoh berikut menunjukkan file manifes CEV untuk rilis Oracle Database yang berbeda. Jika Anda menyertakan bidang JSON dalam manifes, pastikan tidak kosong. Misalnya, manifes CEV berikut ini tidak valid karena `otherPatchFileNames` kosong.

```
{
    "mediaImportTemplateVersion": "2020-08-14",
    "databaseInstallationFileNames": [
        "V982063-01.zip"
    ],
    "opatchFileNames": [
        "p6880880_190000_Linux-x86-64.zip"
    ],
    "psuRuPatchFileNames": [
        "p32126828_190000_Linux-x86-64.zip"
    ],
    "otherPatchFileNames": [
    ]
}
```

Topik

- Sample CEV manifest for Oracle Database 12c Release 1 (12.1)
- Sample CEV manifest for Oracle Database 12c Release 2 (12.2)
- Sample CEV manifest for Oracle Database 18c
- Sample CEV manifest for Oracle Database 19c

Example Contoh manifes CEV untuk Oracle Database 12c Rilis 1 (12.1)

Dalam contoh PSU Juli 2021 untuk Oracle Database 12c Rilis 1 (12.1) berikut, RDS Custom menerapkan patch sesuai urutan yang ditentukan. Dengan demikian, RDS Custom menerapkan p32768233, lalu p32876425, lalu p18759211, dan seterusnya. Contoh ini menetapkan nilai baru untuk pengguna dan grup UNIX, serta beranda Oracle dan basis Oracle.

```
{
    "mediaImportTemplateVersion":"2020-08-14",
    "databaseInstallationFileNames":[
        "V46095-01_1of2.zip",
        "V46095-01_2of2.zip"
    ],
    "opatchFileNames":[
        "p6880880_121010_Linux-x86-64.zip"
    ],
    "psuRuPatchFileNames":[
        "p32768233_121020_Linux-x86-64.zip"
    ],
    "otherPatchFileNames":[
        "p32876425_121020_Linux-x86-64.zip",
        "p18759211_121020_Linux-x86-64.zip",
        "p19396455_121020_Linux-x86-64.zip",
        "p20875898_121020_Linux-x86-64.zip",
        "p22037014_121020_Linux-x86-64.zip",
        "p22873635_121020_Linux-x86-64.zip",
        "p23614158_121020_Linux-x86-64.zip",
        "p24701840_121020_Linux-x86-64.zip",
        "p25881255_121020_Linux-x86-64.zip",
        "p27015449_121020_Linux-x86-64.zip",
        "p28125601_121020_Linux-x86-64.zip",
        "p28852325_121020_Linux-x86-64.zip",
        "p29997937_121020_Linux-x86-64.zip",
        "p31335037_121020_Linux-x86-64.zip",
        "p32327201_121020_Linux-x86-64.zip",
        "p32327208_121020_Generic.zip",
        "p17969866_12102210119_Linux-x86-64.zip",
        "p20394750_12102210119_Linux-x86-64.zip",
        "p24835919_121020_Linux-x86-64.zip",
        "p23262847_12102201020_Linux-x86-64.zip",
        "p21171382_12102201020_Generic.zip",
        "p21091901_12102210720_Linux-x86-64.zip",
        "p33013352_12102210720_Linux-x86-64.zip",
```

```
        "p25031502_12102210720_Linux-x86-64.zip",
        "p23711335_12102191015_Generic.zip",
        "p19504946_121020_Linux-x86-64.zip"
    ],
    "installationParameters": {
        "unixGroupName": "dba",
        "unixGroupId": 12345,
        "unixUname": "oracle",
        "unixUid": 12345,
        "oracleHome": "/home/oracle/oracle.12.1.0.2",
        "oracleBase": "/home/oracle"
    }
}
```

Example Contoh manifes CEV untuk Oracle Database 12c Rilis 2 (12.2)

Dalam contoh PSU Oktober 2021 untuk Oracle Database 12c Rilis 2 (12.2) berikut, RDS Custom menerapkan p33261817, lalu p33192662, lalu p29213893, dan seterusnya. Contoh ini menetapkan nilai baru untuk pengguna dan grup UNIX, serta beranda Oracle dan basis Oracle.

```
{
    "mediaImportTemplateVersion":"2020-08-14",
    "databaseInstallationFileNames":[
        "V839960-01.zip"
    ],
    "opatchFileNames":[
        "p6880880_122010_Linux-x86-64.zip"
    ],
    "psuRuPatchFileNames":[
        "p33261817_122010_Linux-x86-64.zip"
    ],
    "otherPatchFileNames":[
        "p33192662_122010_Linux-x86-64.zip",
        "p29213893_122010_Generic.zip",
        "p28730253_122010_Linux-x86-64.zip",
        "p26352615_12201211019DBOCT2021RU_Linux-x86-64.zip",
        "p23614158_122010_Linux-x86-64.zip",
        "p24701840_122010_Linux-x86-64.zip",
        "p25173124_122010_Linux-x86-64.zip",
        "p25881255_122010_Linux-x86-64.zip",
        "p27015449_122010_Linux-x86-64.zip",
        "p28125601_122010_Linux-x86-64.zip",
        "p28852325_122010_Linux-x86-64.zip",
```

```
        "p29997937_122010_Linux-x86-64.zip",
        "p31335037_122010_Linux-x86-64.zip",
        "p32327201_122010_Linux-x86-64.zip",
        "p32327208_122010_Generic.zip"
    ],
    "installationParameters": {
        "unixGroupName": "dba",
        "unixGroupId": 12345,
        "unixUname": "oracle",
        "unixUid": 12345,
        "oracleHome": "/home/oracle/oracle.12.2.0.1",
        "oracleBase": "/home/oracle"
    }
}
```

Example Contoh manifes CEV untuk Oracle Database 18c

Dalam contoh PSU Oktober 2021 untuk Oracle Database 18c berikut, RDS Custom menerapkan p32126855, lalu p28730253, lalu p27539475, dan seterusnya. Contoh ini menetapkan nilai baru untuk pengguna dan grup UNIX, serta beranda Oracle dan basis Oracle.

```
{
    "mediaImportTemplateVersion":"2020-08-14",
    "databaseInstallationFileNames":[
        "V978967-01.zip"
    ],
    "opatchFileNames":[
        "p6880880_180000_Linux-x86-64.zip"
    ],
    "psuRuPatchFileNames":[
        "p32126855_180000_Linux-x86-64.zip"
    ],
    "otherPatchFileNames":[
        "p28730253_180000_Linux-x86-64.zip",
        "p27539475_1813000DBRU_Linux-x86-64.zip",
        "p29213893_180000_Generic.zip",
        "p29374604_1813000DBRU_Linux-x86-64.zip",
        "p29782284_180000_Generic.zip",
        "p28125601_180000_Linux-x86-64.zip",
        "p28852325_180000_Linux-x86-64.zip",
        "p29997937_180000_Linux-x86-64.zip",
        "p31335037_180000_Linux-x86-64.zip",
        "p31335142_180000_Generic.zip"
```

```
    ]
    "installationParameters": {
        "unixGroupName": "dba",
        "unixGroupId": 12345,
        "unixUname": "oracle",
        "unixUid": 12345,
        "oracleHome": "/home/oracle/18.0.0.0.ru-2020-10.rur-2020-10.r1",
        "oracleBase": "/home/oracle/"
    }
}
```

Example Contoh manifes CEV untuk Oracle Database 19c

Dalam contoh untuk Oracle Database 19c berikut, RDS Custom menerapkan p32126828, lalu p29213893, lalu p29782284, dan seterusnya. Contoh ini menetapkan nilai baru untuk pengguna dan grup UNIX, serta beranda Oracle dan basis Oracle.

```
{
    "mediaImportTemplateVersion": "2020-08-14",
    "databaseInstallationFileNames": [
        "V982063-01.zip"
    ],
    "opatchFileNames": [
        "p6880880_190000_Linux-x86-64.zip"
    ],
    "psuRuPatchFileNames": [
        "p32126828_190000_Linux-x86-64.zip"
    ],
    "otherPatchFileNames": [
        "p29213893_1910000DBRU_Generic.zip",
        "p29782284_1910000DBRU_Generic.zip",
        "p28730253_190000_Linux-x86-64.zip",
        "p29374604_1910000DBRU_Linux-x86-64.zip",
        "p28852325_190000_Linux-x86-64.zip",
        "p29997937_190000_Linux-x86-64.zip",
        "p31335037_190000_Linux-x86-64.zip",
        "p31335142_190000_Generic.zip"
    ],
    "installationParameters": {
        "unixGroupName": "dba",
        "unixGroupId": 12345,
        "unixUname": "oracle",
        "unixUid": 12345,
```

```
        "oracleHome": "/home/oracle/oracle.19.0.0.0.ru-2020-04.rur-2020-04.r1.EE.1",
        "oracleBase": "/home/oracle"
    }
}
```

Langkah 6 (Opsional): Validasi manifes CEV

Secara opsional, verifikasi bahwa manifes adalah file JSON yang valid dengan menjalankan skrip Python `json.tool`. Misalnya, jika Anda mengubah ke direktori yang berisi manifes CEV bernama `manifest.json`, jalankan perintah berikut.

```
python -m json.tool < manifest.json
```

Langkah 7: Tambahkan izin IAM yang diperlukan

Pastikan bahwa pengguna utama IAM yang membuat CEV memiliki kebijakan yang diperlukan yang dijelaskan dalam Langkah 5: Berikan izin yang diperlukan ke pengguna atau peran IAM Anda.

Membuat CEV

Anda dapat membuat CEV menggunakan AWS Management Console atau AWS CLI. Tentukan arsitektur multi-penghuni atau nonmulti-penghuni. Untuk informasi selengkapnya, lihat Pertimbangan arsitektur multi-penghuni.

Biasanya, membuat CEV membutuhkan waktu sekitar dua jam. Setelah CEV dibuat, Anda dapat menggunakannya untuk membuat instans DB RDS Custom. Untuk informasi selengkapnya, lihat Membuat instans DB RDS Custom for Oracle.

Perhatikan persyaratan dan batasan berikut untuk membuat CEV:

- Bucket Amazon S3 yang berisi file instalasi Anda harus Wilayah AWS sama dengan CEV Anda. Jika tidak, proses pembuatan gagal.
- Nama CEV harus dalam format*`major-engine-version.customized_string`*, seperti dalam`19.cdb_cev1`.
- Nama CEV harus berisi 1-50 karakter alfanumerik, garis bawah, tanda hubung, atau titik.
- Nama CEV tidak dapat berisi periode berturut-turut, seperti pada. `19..cdb_cev1`

Konsol

Cara membuat CEV

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Versi mesin kustom.

   Halaman Versi mesin kustom menampilkan semua CEV yang ada saat ini. Jika Anda belum membuat CEV, halaman tersebut kosong.
3. Pilih Buat versi mesin kustom.
4. Di bagian Jenis mesin, lakukan hal berikut:

   a. Untuk Jenis mesin, pilih Oracle.

   b. Untuk Pengaturan arsitektur, Anda dapat memilih Arsitektur multi-penghuni jika ingin, untuk membuat CEV Multi-penghuni yang menggunakan mesin `custom-oracle-ee-cdb`. Anda dapat membuat CDB RDS Custom for Oracle hanya dengan CEV Multi-penghuni. Jika Anda tidak memilih opsi ini, CEV Anda adalah non-CDB, yang menggunakan mesin `custom-oracle-ee`.

   > **Note**
   >
   > Arsitektur yang Anda pilih adalah karakteristik permanen CEV Anda. Anda tidak dapat memodifikasi CEV Anda untuk menggunakan arsitektur yang berbeda nanti.

   c. Pilih salah satu opsi berikut:

   - Buat CEV baru – Buat CEV dari awal. Dalam hal ini, Anda harus menentukan manifes JSON yang menentukan biner basis data.
   - Buat CEV dari sumber – Dalam Tentukan CEV yang ingin Anda salin, pilih CEV yang ada untuk digunakan sebagai sumber CEV. Dalam hal ini, Anda dapat menentukan Amazon Machine Image (AMI) baru, tetapi tidak dapat menentukan biner basis data yang berbeda.

   d. Untuk Versi mesin, pilih versi mesin utama.
5. Dalam Detail versi, lakukan hal berikut:

   a. Masukkan nama yang valid dalam Nama versi mesin kustom. Misalnya, Anda dapat memasukkan nama **`19.cdb_cev1`**.

b. (Opsional) Masukkan deskripsi untuk CEV Anda.

6. Di Media instalasi, lakukan hal berikut:

   a. (Opsional) Untuk ID AMI, biarkan bidang kosong untuk menggunakan AMI terbaru yang disediakan layanan, atau masukkan AMI yang sebelumnya Anda gunakan untuk membuat CEV. Untuk mendapatkan ID AMI yang valid, gunakan salah satu teknik berikut:

      - Di konsol, pilih Versi mesin kustom di panel navigasi kiri, dan pilih nama CEV. ID AMI yang digunakan oleh CEV muncul di tab Konfigurasi.
      - Di AWS CLI, gunakan perintah `describe-db-engine-versions`. Cari output untuk `ImageID`.

   b. Untuk Lokasi file manifes S3, masukkan lokasi bucket Amazon S3 yang Anda tentukan di Langkah 3: Unggah file instalasi Anda ke Amazon S3. Misalnya, masukkan **`s3://my-custom-installation-files/123456789012/cev1/`**.

      > **Note**
      >
      > Wilayah AWS tempat Anda membuat CEV harus berada di Wilayah yang sama dengan bucket S3.

   c. (Khusus buat CEV baru) Untuk manifes CEV, masukkan manifes JSON yang Anda buat di Membuat manifes CEV.

7. Di bagian Kunci KMS, pilih Masukkan ARN kunci untuk membuat daftar kunci AWS KMS yang tersedia. Lalu pilih kunci KMS Anda dari daftar.

   Kunci AWS KMS diperlukan untuk RDS Custom. Untuk informasi selengkapnya, lihat Langkah 1: Buat atau gunakan kembali kunci enkripsi simetris AWS KMS.

8. (Opsional) Pilih Tambahkan tag baru guna membuat pasangan nilai kunci untuk CEV Anda.

9. Pilih Buat versi mesin kustom.

   Jika format manifes JSON tidak valid, konsol akan menampilkan Terjadi kesalahan saat memvalidasi manifes CEV. Perbaiki masalah dan coba lagi.

Halaman Versi mesin kustom muncul. CEV Anda ditampilkan dengan status Membuat. Proses untuk membuat CEV membutuhkan waktu sekitar dua jam.

AWS CLI

Untuk membuat CEV dengan menggunakanAWS CLI, jalankan perintah create-custom-db-engine-version.

Opsi berikut diperlukan:

- `--engine` – Tentukan jenis mesin, baik `custom-oracle-ee-cdb` untuk CDB atau `custom-oracle-ee` untuk non-CDB. Anda dapat membuat CDB hanya dari CEV yang dibuat dengan `custom-oracle-ee-cdb`. Anda dapat membuat non-CDB hanya dari CEV yang dibuat dengan `custom-oracle-ee`.
- `--engine-version` – Tentukan versi mesin. Formatnya adalah *`major-engine-version`*.*`customized_string`*. Nama CEV harus berisi 1-50 karakter alfanumerik, garis bawah, tanda hubung, atau titik. Nama CEV tidak dapat berisi periode berturut-turut, seperti pada. `19..cdb_cev1`
- `--kms-key-id` – Tentukan AWS KMS key.
- `--manifest` – Tentukan *`manifest_json_string`* atau `--manifest file:`*`file_name`*. Karakter baris baru tidak diizinkan di *`manifest_json_string`*. Pastikan tidak ada kutipan ganda (") dalam kode JSON dengan memberikan awalan garis miring terbalik (\).

  Contoh berikut menunjukkan *`manifest_json_string`* untuk 19c dari Langkah 5: Siapkan manifes CEV. Contoh tersebut menetapkan nilai baru untuk basis Oracle, beranda Oracle, serta ID dan nama pengguna dan grup UNIX/Linux. Jika Anda menyalin string ini, hapus semua karakter baris baru sebelum menempelkannya ke perintah Anda.

```
"{\"mediaImportTemplateVersion\": \"2020-08-14\",
\"databaseInstallationFileNames\": [\"V982063-01.zip\"],
\"opatchFileNames\": [\"p6880880_190000_Linux-x86-64.zip\"],
\"psuRuPatchFileNames\": [\"p32126828_190000_Linux-x86-64.zip\"],
\"otherPatchFileNames\": [\"p29213893_1910000DBRU_Generic.zip\",
\"p29782284_1910000DBRU_Generic.zip\",\"p28730253_190000_Linux-
x86-64.zip\",\"p29374604_1910000DBRU_Linux-x86-64.zip\",
\"p28852325_190000_Linux-x86-64.zip\",\"p29997937_190000_Linux-x86-64.zip
\",\"p31335037_190000_Linux-x86-64.zip\",\"p31335142_190000_Generic.zip
\"]\"installationParameters\":{ \"unixGroupName\":\"dba\",
\ \"unixUname\":\"oracle\", \ \"oracleHome\":\"/home/oracle/
oracle.19.0.0.0.ru-2020-04.rur-2020-04.r1.EE.1\", \ \"oracleBase\":\"/
home/oracle/\"}}"
```

- `--database-installation-files-s3-bucket-name` – Tentukan nama bucket yang sama dengan yang Anda tentukan di Langkah 3: Unggah file instalasi Anda ke Amazon S3. Wilayah AWS tempat Anda menjalankan `create-custom-db-engine-version` harus merupakan Wilayah yang sama dengan bucket Amazon S3.

Anda juga dapat menentukan parameter berikut:

- `--description` – Tentukan deskripsi CEV Anda.
- `--database-installation-files-s3-prefix` – Tentukan nama folder yang Anda tentukan di Langkah 3: Unggah file instalasi Anda ke Amazon S3.
- `--image-id` – Tentukan ID AMI yang ingin digunakan kembali. Untuk menemukan ID yang valid, jalankan perintah `describe-db-engine-versions`, lalu cari output untuk `ImageID`. Secara default, RDS Custom for Oracle menggunakan AMI terbaru yang tersedia.

Contoh berikut membuat CEV multi-penghuni Oracle bernama `19.cdb_cev1`. Contoh menggunakan kembali AMI yang ada, bukan menggunakan AMI terbaru yang tersedia. Pastikan bahwa nama CEV Anda dimulai dengan nomor versi mesin utama.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds create-custom-db-engine-version \
    --engine custom-oracle-ee-cdb \
    --engine-version 19.cdb_cev1 \
    --database-installation-files-s3-bucket-name us-east-1-123456789012-custom-
installation-files \
    --database-installation-files-s3-prefix 123456789012/cev1 \
    --kms-key-id my-kms-key \
    --description "test cev" \
    --manifest manifest_string \
    --image-id ami-012a345678901bcde
```

Untuk Windows:

```
aws rds create-custom-db-engine-version ^
    --engine custom-oracle-ee-cdb ^
    --engine-version 19.cdb_cev1 ^
```

```
    --database-installation-files-s3-bucket-name us-east-1-123456789012-custom-
installation-files ^
    --database-installation-files-s3-prefix 123456789012/cev1 ^
    --kms-key-id my-kms-key ^
    --description "test cev" ^
    --manifest manifest_string ^
    --image-id ami-012a345678901bcde
```

Example

Dapatkan detail tentang CEV Anda dengan menggunakan perintah `describe-db-engine-versions`.

```
aws rds describe-db-engine-versions \
    --engine custom-oracle-ee-cdb \
    --include-all
```

Output contoh parsial berikut menunjukkan mesin, grup parameter, manifes, dan informasi lainnya.

```
{
    "DBEngineVersions": [
        {
            "Engine": "custom-oracle-ee-cdb",
            "EngineVersion": "19.cdb_cev1",
            "DBParameterGroupFamily": "custom-oracle-ee-cdb-19",
            "DBEngineDescription": "Containerized Database for Oracle Custom EE",
            "DBEngineVersionDescription": "test cev",
            "Image": {
                "ImageId": "ami-012a345678901bcde",
                "Status": "active"
            },
            "ValidUpgradeTarget": [],
            "SupportsLogExportsToCloudwatchLogs": false,
            "SupportsReadReplica": true,
            "SupportedFeatureNames": [],
            "Status": "available",
            "SupportsParallelQuery": false,
            "SupportsGlobalDatabases": false,
            "MajorEngineVersion": "19",
            "DatabaseInstallationFilesS3BucketName": "us-east-1-123456789012-custom-
installation-files",
            "DatabaseInstallationFilesS3Prefix": "123456789012/cev1",
```

```
            "DBEngineVersionArn": "arn:aws:rds:us-east-1:123456789012:cev:custom-
oracle-ee-cdb/19.cdb_cev1/abcd12e3-4f5g-67h8-i9j0-k1234l56m789",
            "KMSKeyId": "arn:aws:kms:us-
east-1:732027699161:key/1ab2345c-6d78-9ef0-1gh2-3456i7j89k01",
            "CreateTime": "2023-03-07T19:47:58.131000+00:00",
            "TagList": [],
            "SupportsBabelfish": false,
...
```

Kegagalan dalam membuat CEV

Jika proses untuk membuat CEV gagal, RDS Custom mengeluarkan `RDS-EVENT-0198` dengan pesan `Creation failed for custom engine version` *`major-engine-version.cev_name`* yang menyertakan detail tentang kegagalan. Misalnya, peristiwa mencetak file yang tidak ada.

Anda tidak dapat memodifikasi CEV yang gagal. Anda hanya dapat menghapusnya, lalu mencoba membuat CEV lagi setelah memperbaiki penyebab kegagalan. Untuk informasi tentang pemecahan masalah alasan kegagalan pembuatan CEV, lihat Memecahkan masalah pembuatan versi mesin kustom untuk RDS Custom for Oracle.

Mengubah status CEV

Anda dapat mengubah CEV menggunakan AWS Management Console atau AWS CLI. Anda dapat mengubah deskripsi CEV atau status ketersediaannya. CEV Anda memiliki salah satu dari nilai status berikut:

- `available` – Anda dapat menggunakan CEV ini untuk membuat instans DB RDS Custom baru atau meningkatkan instans DB. Ini adalah status default untuk CEV yang baru dibuat.
- `inactive` – Anda tidak dapat membuat atau meningkatkan instans RDS Custom dengan CEV ini. Anda tidak dapat memulihkan snapshot DB untuk membuat instans DB RDS Custom baru dengan CEV ini.

Anda dapat mengubah CEV dari status apa pun yang didukung ke status lain yang didukung. Anda dapat mengubah status untuk mencegah penggunaan CEV yang tidak disengaja atau membuat CEV yang dihentikan memenuhi syarat untuk digunakan lagi. Misalnya, Anda dapat mengubah status CEV Anda dari `available` ke `inactive`, dan dari `inactive` kembali ke `available`.

Konsol

Cara mengubah CEV

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Versi mesin kustom.
3. Pilih CEV yang deskripsi atau statusnya ingin Anda ubah.
4. Untuk Tindakan, pilih Ubah.
5. Lakukan salah satu dari perubahan berikut:
   - Untuk Pengaturan status CEV, pilih status ketersediaan baru.
   - Untuk Deskripsi versi, masukkan deskripsi baru.
6. Pilih Ubah CEV.

   Jika CEV sedang digunakan, konsol akan menampilkan Anda tidak dapat mengubah status CEV. Perbaiki masalah dan coba lagi.

Halaman Versi mesin kustom muncul.

AWS CLI

Untuk memodifikasi CEV dengan menggunakanAWS CLI, jalankan perintah modify-custom-db-engine-version. Anda dapat menemukan CEV untuk dimodifikasi dengan menjalankan perintah. describe-db-engine-versions

Opsi berikut diperlukan:

- `--engine custom-oracle-ee`
- `--engine-version` *`cev`*, dengan *`cev`* adalah nama versi mesin kustom yang ingin Anda modifikasi
- `--status` *`status`*, dengan *`status`* adalah status ketersediaan yang ingin Anda tetapkan ke CEV

Contoh berikut mengubah CEV bernama `19.my_cev1` dari statusnya saat ini menjadi`inactive`.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds modify-custom-db-engine-version \
    --engine custom-oracle-ee \
    --engine-version 19.my_cev1 \
    --status inactive
```

Untuk Windows:

```
aws rds modify-custom-db-engine-version ^
    --engine custom-oracle-ee ^
    --engine-version 19.my_cev1 ^
    --status inactive
```

Melihat detail CEV

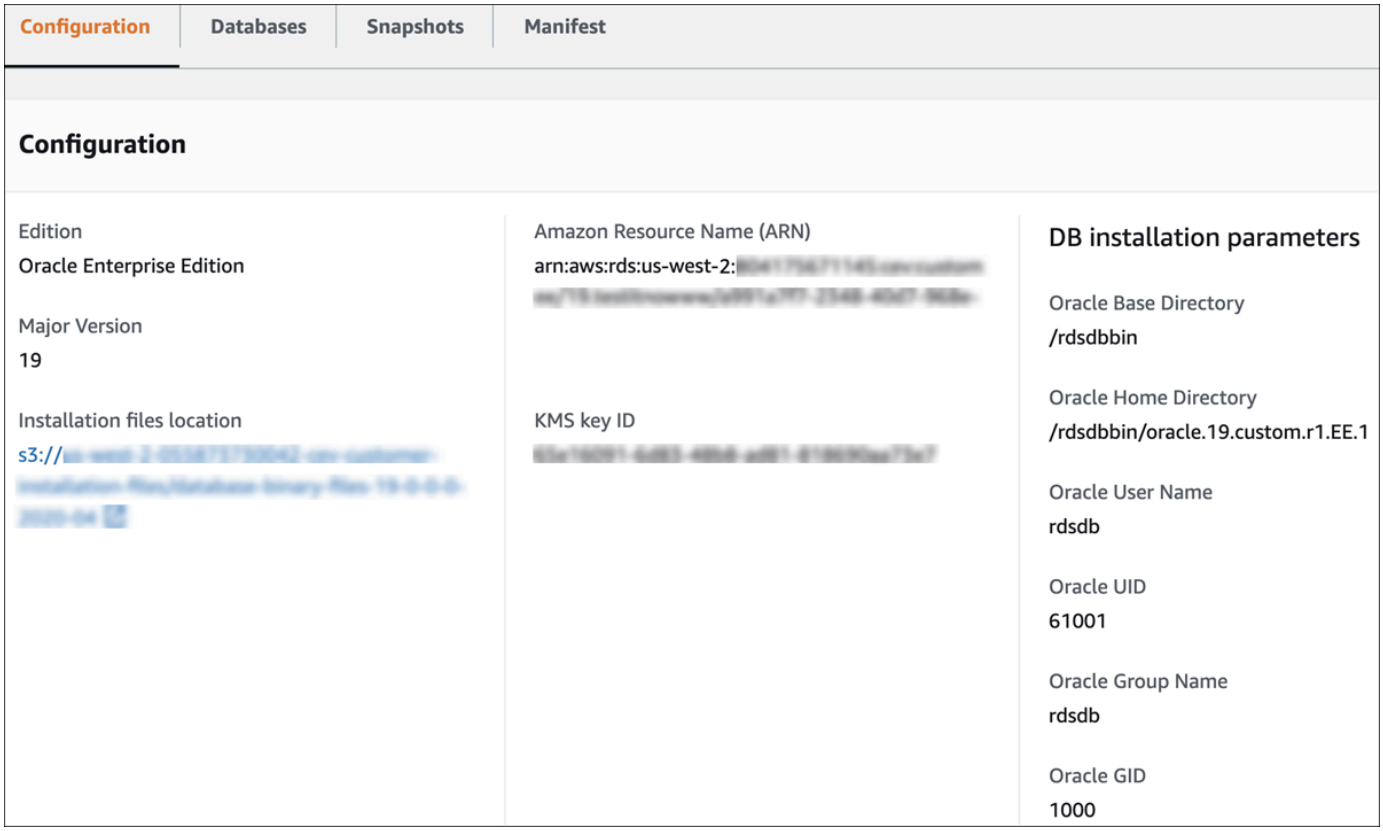
Anda dapat melihat detail tentang manifes CEV dan perintah yang digunakan untuk membuat CEV Anda dengan menggunakan AWS Management Console atau AWS CLI.

Konsol

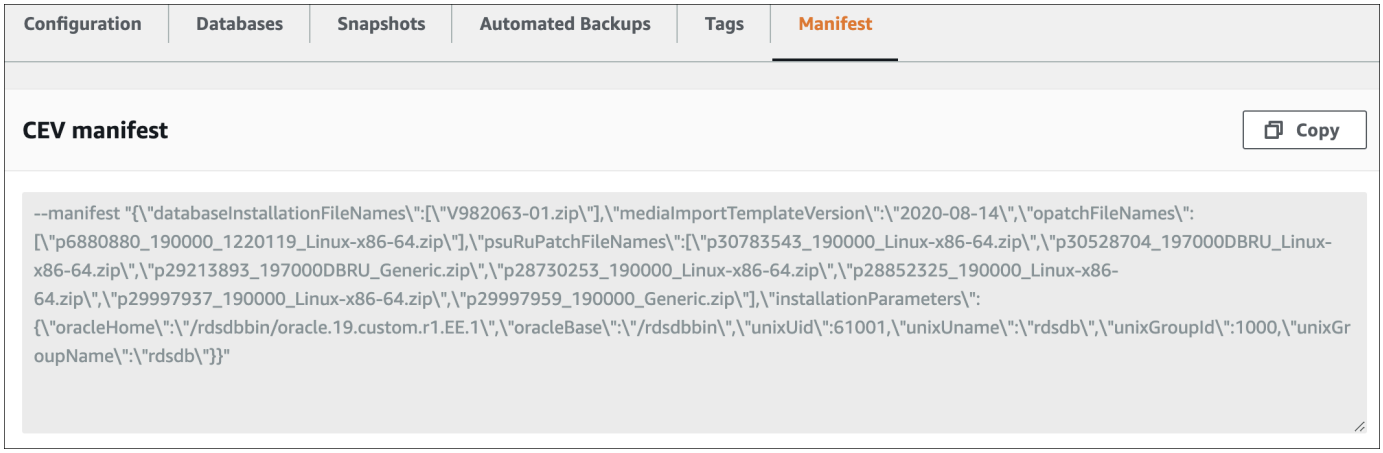
Cara melihat detail CEV

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Versi mesin kustom.

   Halaman Versi mesin kustom menampilkan semua CEV yang ada saat ini. Jika Anda belum membuat CEV, halaman tersebut kosong.
3. Pilih nama CEV yang ingin Anda lihat.
4. Pilih Konfigurasi untuk melihat parameter instalasi yang ditentukan dalam manifes Anda.



5. Pilih Manifes untuk melihat parameter instalasi yang ditentukan dalam opsi `--manifest` perintah `create-custom-db-engine-version`. Anda dapat menyalin teks ini, mengganti nilai sesuai kebutuhan, dan menggunakannya dalam perintah baru.



AWS CLI

Untuk melihat detail tentang CEV dengan menggunakanAWS CLI, jalankan describe-db-engine-versionsperintah.

Opsi berikut diperlukan:

- `--engine custom-oracle-ee`
- `--engine-version` *`major-engine-version.customized_string`*

Contoh berikut membuat CEV bernama `19.my_cev1`. Pastikan bahwa nama CEV Anda dimulai dengan nomor versi mesin utama.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds describe-db-engine-versions \
    --engine custom-oracle-ee \
    --engine-version 19.my_cev1
```

Untuk Windows:

```
aws rds describe-db-engine-versions ^
    --engine custom-oracle-ee ^
    --engine-version 19.my_cev1
```

Output contoh parsial berikut menunjukkan mesin, grup parameter, manifes, dan informasi lainnya.

```
"DBEngineVersions": [
    {
        "Engine": "custom-oracle-ee",
        "MajorEngineVersion": "19",
        "EngineVersion": "19.my_cev1",
        "DatabaseInstallationFilesS3BucketName": "us-east-1-123456789012-cev-customer-
installation-files",
        "DatabaseInstallationFilesS3Prefix": "123456789012/cev1",
        "CustomDBEngineVersionManifest": "{\n\"mediaImportTemplateVersion\":
 \"2020-08-14\",\n\"databaseInstallationFileNames\": [\n\"V982063-01.zip\"\n],
\n\"installationParameters\": {\n\"oracleBase\":\"/tmp\",\n\"oracleHome\":\"/
tmp/Oracle\"\n},\n\"opatchFileNames\": [\n\"p6880880_190000_Linux-x86-64.zip
\"\n],\n\"psuRuPatchFileNames\": [\n\"p32126828_190000_Linux-x86-64.zip
\"\n],\n\"otherPatchFileNames\": [\n\"p29213893_1910000DBRU_Generic.zip\",\n
\"p29782284_1910000DBRU_Generic.zip\",\n\"p28730253_190000_Linux-x86-64.zip\",\n
\"p29374604_1910000DBRU_Linux-x86-64.zip\",\n\"p28852325_190000_Linux-x86-64.zip\",
```

```
\n\"p29997937_190000_Linux-x86-64.zip\",\n\"p31335037_190000_Linux-x86-64.zip\",\n
\"p31335142_190000_Generic.zip\"\n]\n}\n",
        "DBParameterGroupFamily": "custom-oracle-ee-19",
        "DBEngineDescription": "Oracle Database server EE for RDS Custom",
        "DBEngineVersionArn": "arn:aws:rds:us-west-2:123456789012:cev:custom-oracle-
ee/19.my_cev1/0a123b45-6c78-901d-23e4-5678f901fg23",
        "DBEngineVersionDescription": "test",
        "KMSKeyId": "arn:aws:kms:us-east-1:123456789012:key/ab1c2de3-f4g5-6789-h012-
h3ijk4567l89",
        "CreateTime": "2022-11-18T09:17:07.693000+00:00",
        "ValidUpgradeTarget": [
        {
            "Engine": "custom-oracle-ee",
            "EngineVersion": "19.cev.2021-01.09",
            "Description": "test",
            "AutoUpgrade": false,
            "IsMajorVersionUpgrade": false
        }
]
```

Menghapus CEV

Anda dapat menghapus CEV menggunakan AWS Management Console atau AWS CLI. Biasanya, penghapusan membutuhkan waktu beberapa menit.

Untuk menghapusnya, CEV tidak dapat digunakan oleh salah satu dari berikut ini:

- Instans DB RDS Custom
- Snapshot instans DB RDS Custom
- Cadangan otomatis instans DB RDS Custom

Konsol

Cara menghapus CEV

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Versi mesin kustom.
3. Pilih CEV yang deskripsi atau statusnya ingin Anda hapus.
4. Untuk Tindakan, pilih Hapus.

Kotak dialog Hapus ***cev_name?*** muncul.

5. Masukkan **`delete me`**, lalu pilih Hapus.

   Di halaman Versi mesin kustom, banner menunjukkan bahwa CEV Anda sedang dihapus.

AWS CLI

Untuk menghapus CEV dengan menggunakanAWS CLI, jalankan perintah delete-custom-db-engine-version.

Opsi berikut diperlukan:

- `--engine custom-oracle-ee`
- `--engine-version` *`cev`*, dengan *`cev`* adalah nama versi mesin kustom yang akan dihapus

Contoh berikut menghapus CEV bernama `19.my_cev1`.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds delete-custom-db-engine-version \
    --engine custom-oracle-ee \
    --engine-version 19.my_cev1
```

Untuk Windows:

```
aws rds delete-custom-db-engine-version ^
    --engine custom-oracle-ee ^
    --engine-version 19.my_cev1
```

Mengonfigurasi instans DB untuk Amazon RDS Custom for Oracle

Anda dapat membuat instans DB RDS Custom, kemudian menghubungkannya menggunakan Secure Shell (SSH) atau AWS Systems Manager.

Topik

Pertimbangan arsitektur multi-penghuni

Jika Anda membuat instans DB Amazon RDS Custom for Oracle dengan arsitektur multi-penghuni (jenis mesin `custom-oracle-ee-cdb`), basis data Anda adalah basis data kontainer (CDB). Jika Anda tidak menentukan arsitektur multi-penghuni, basis data Anda adalah non-CDB tradisional yang menggunakan jenis mesin `custom-oracle-ee`. Non-CDB tidak bisa berisi basis data pluggable (PDB). Untuk informasi selengkapnya, lihat Arsitektur basis data untuk Amazon RDS Custom for Oracle.

Saat Anda membuat instans CDB RDS Custom for Oracle, pertimbangkan hal berikut:

- Anda dapat membuat basis data multi-penghuni hanya dari Oracle Database 19c CEV.
- Anda dapat membuat instans CDB hanya jika CEV menggunakan jenis mesin `custom-oracle-ee-cdb`.
- Secara default, CDB Anda diberi nama `RDSCDB`, yang juga merupakan nama Oracle System ID (Oracle SID). Anda dapat memilih nama yang berbeda.
- CDB Anda hanya berisi satu PDB awal. Nama default PDB adalah `ORCL`. Anda dapat memilih nama yang berbeda untuk PDB awal, tetapi Oracle SID dan nama PDB tidak boleh sama.
- RDS Custom for Oracle tidak menyediakan API untuk PDB. Untuk membuat PDB tambahan, gunakan perintah Oracle SQL `CREATE PLUGGABLE DATABASE`. RDS Custom for Oracle tidak

membatasi jumlah PDB yang dapat Anda buat. Secara umum, Anda bertanggung jawab untuk membuat dan mengelola PDB, seperti dalam deployment on-premise.

- Anda tidak dapat menggunakan API RDS untuk membuat, mengubah, dan menghapus PDB: Anda harus menggunakan pernyataan Oracle SQL. Saat Anda membuat PDB menggunakan Oracle SQL, kami sarankan Anda mengambil snapshot manual sesudahnya jika Anda perlu melakukan point-in-time pemulihan (PITR).
- Anda tidak dapat mengganti nama PDB yang ada menggunakan API Amazon RDS. Anda juga tidak dapat mengganti nama CDB menggunakan perintah `modify-db-instance`.
- Mode terbuka untuk root CDB adalah `READ WRITE` di basis data primer dan `MOUNTED` di basis data siaga terpasang. RDS Custom for Oracle mencoba membuka semua PDB ketika membuka CDB. Jika tidak dapat membuka semua PDB, RDS Custom for Oracle akan mengeluarkan peristiwa `tenant database shutdown`.

Membuat instans DB RDS Custom for Oracle

Buat instans DB Amazon RDS Custom for Oracle menggunakan AWS Management Console atau AWS CLI. Prosedurnya mirip dengan prosedur untuk membuat instans DB Amazon RDS. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.

Jika menyertakan parameter instalasi dalam manifes CEV, instans DB Anda menggunakan basis Oracle, beranda Oracle, dan ID serta nama pengguna dan grup UNIX/Linux yang Anda tentukan. File `oratab`, yang dibuat oleh Oracle Database selama instalasi, menunjuk ke lokasi instalasi sebenarnya, bukan ke tautan simbolis. Ketika menjalankan perintah, RDS Custom for Oracle berjalan sebagai pengguna OS yang dikonfigurasi, bukan pengguna default `rdsdb`. Untuk informasi selengkapnya, lihat Langkah 5: Siapkan manifes CEV.

Sebelum Anda mencoba membuat atau terhubung ke instans DB RDS Custom, selesaikan tugas di Menyiapkan lingkungan Anda untuk Amazon RDS Custom for Oracle.

Konsol

Cara membuat instans DB RDS Custom for Oracle

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data.
3. Pilih Buat basis data.

4. Di bagian Pilih metode pembuatan basis data, pilih Pembuatan Standar.
5. Di bagian Opsi mesin, lakukan hal berikut:

   a. Untuk Jenis mesin, pilih Oracle.
   b. Untuk Jenis manajemen basis data, pilih Amazon RDS Custom.
   c. Untuk Pengaturan arsitektur, lakukan hal berikut:

      - Pilih Arsitektur multipenyewa untuk membuat basis data kontainer (CDB). Saat pembuatan, CDB Anda berisi satu seed PDB dan satu PDB awal.

        > **Note**
        >
        > Pengaturan Arsitektur multipenyewa hanya didukung untuk Oracle Database 19c.

      - Hapus Arsitektur multipenyewa untuk membuat non-CDB. Non-CDB tidak boleh berisi PDB.

   d. Untuk Edisi, pilih Oracle Enterprise Edition.
   e. Untuk Versi mesin kustom, pilih versi mesin kustom (CEV) RDS Custom yang ada. CEV memiliki format seperti berikut: *major-engine-version*.*customized_string*. Contoh pengidentifikasi adalah`19.cdb_cev1`.

      Jika memilih Arsitektur multipenyewa pada langkah sebelumnya, Anda hanya dapat menentukan CEV yang menggunakan jenis mesin `custom-oracle-ee-cdb`. Konsol memfilter CEV yang dibuat dengan jenis mesin `custom-oracle-ee`.
6. Di bagian Templat, pilih Produksi.
7. Di bagian Pengaturan, lakukan hal berikut:

   a. Untuk Pengidentifikasi instans DB, masukkan nama unik untuk instans DB Anda.
   b. Untuk Nama pengguna master, masukkan nama pengguna. Anda dapat mengambil nilai ini dari konsol nanti.

      Ketika Anda terhubung ke non-CDB, pengguna master adalah pengguna untuk non-CDB. Ketika Anda terhubung ke CDB, pengguna master adalah pengguna untuk CDB. Untuk terhubung ke root CDB, masuk ke host, mulai klien SQL, dan buat pengguna administratif dengan perintah SQL.
   c. Hapus Buat kata sandi secara otomatis.
8. Pilih Kelas instans DB.

Untuk kelas yang didukung, lihat Dukungan kelas instans DB untuk RDS Custom for Oracle.

9. Di bagian Penyimpanan, lakukan hal berikut:

   a. Untuk Tipe penyimpanan, pilih tipe SSD: io1, gp2, atau gp3. Anda memiliki opsi tambahan berikut:

      - Untuk io1 atau gp3, pilih kecepatan untuk IOPS yang Tersedia. Default-nya adalah 1000 untuk io1 dan 12000 untuk gp3.
      - Untuk gp3, pilih kecepatan untuk Throughput penyimpanan. Defaultnya adalah 500 MiBps.

   b. Untuk Penyimpanan yang dialokasikan, pilih ukuran penyimpanan. Default-nya adalah 40 GiB.

10. Untuk Konektivitas, tentukan Cloud privat virtual (VPC), grup subnet DB, dan grup keamanan VPC (firewall).

11. Untuk Keamanan RDS Custom, lakukan hal berikut:

    a. Untuk Profil instans IAM, pilih profil instans untuk instans DB RDS Custom for Oracle Anda.

       Profil instans IAM harus dimulai dengan `AWSRDSCustom`, misalnya *`AWSRDSCustomInstanceProfileForRdsCustomInstance`*.

    b. Untuk Enkripsi, pilih Masukkan kunci ARN untuk mencantumkan kunci AWS KMS yang tersedia. Lalu pilih kunci Anda dari daftar.

       Kunci AWS KMS diperlukan untuk RDS Custom. Untuk informasi selengkapnya, lihat Langkah 1: Buat atau gunakan kembali kunci enkripsi simetris AWS KMS.

12. Untuk Opsi basis data, lakukan hal berikut:

    a. (Opsional) Untuk ID Sistem (SID), masukkan nilai untuk SID Oracle, yang juga merupakan nama CDB Anda. SID adalah nama instans basis data Oracle yang mengelola file basis data Anda. Dalam konteks ini, istilah "instans basis data Oracle" mengacu secara eksklusif pada area global sistem (SGA) dan proses latar belakang Oracle. Jika Anda tidak menentukan nilai, default-nya adalah **`RDSCDB`**.

    b. (Opsional) Untuk Nama basis data awal, masukkan nama. Nilai default-nya adalah **`ORCL`**. Dalam arsitektur mutipenyewa, nama basis data awal adalah nama PDB.

> **Note**
>
> Nama SID dan PDB harus berbeda.

c. Untuk Grup opsi, pilih grup opsi atau terima default.

> **Note**
>
> Satu-satunya opsi yang didukung untuk RDS Custom for Oracle adalah `Timezone`. Untuk informasi selengkapnya, lihat Zona waktu Oracle.

d. Untuk Periode retensi cadangan pilih nilai. Anda tidak dapat memilih 0 hari.

e. Untuk bagian yang tersisa, tentukan pengaturan instans DB RDS Custom pilihan Anda. Untuk informasi tentang setiap pengaturan, lihat Pengaturan untuk instans DB. Pengaturan berikut tidak muncul di konsol dan tidak didukung:

- Fitur prosesor
- Penskalaan otomatis penyimpanan
- Ketersediaan & daya tahan
- Opsi Autentikasi kata sandi dan Kerberos dalam Autentikasi basis data (hanya Autentikasi kata sandi yang didukung)
- Wawasan Kinerja
- Ekspor log
- Aktifkan peningkatan versi minor otomatis
- Perlindungan penghapusan

13. Pilih Buat basis data.

> **Important**
>
> Saat membuat instans DB RDS Custom for Oracle, Anda mungkin menerima kesalahan berikut: Peran terkait layanan sedang dalam proses pembuatan. Coba lagi nanti. Jika ya, tunggu beberapa menit dan coba buat instans DB lagi.

Tombol Lihat detail kredensial muncul di halaman Basis data.

Untuk melihat nama pengguna dan kata sandi master untuk instans DB RDS Custom, pilih Lihat detail kredensial.

Untuk terhubung ke instans DB sebagai pengguna master, gunakan nama pengguna dan kata sandi yang muncul.

> **Important**
>
> Anda tidak dapat melihat kata sandi pengguna master lagi di konsol. Jika tidak mencatatnya, Anda mungkin harus mengubahnya. Untuk mengubah kata sandi pengguna master setelah instans DB RDS Custom tersedia, masuk ke basis data dan jalankan perintah `ALTER USER`. Anda tidak dapat mengatur ulang kata sandi menggunakan opsi Ubah di konsol.

14. Pilih Basis data untuk melihat daftar instans DB RDS Custom.
15. Pilih instans DB RDS Custom yang baru saja Anda buat.

    Pada konsol RDS, detail untuk instans DB RDS Custom baru muncul:

    - Instans DB akan berstatus membuat hingga instans DB RDS Custom selesai dibuat dan siap digunakan. Saat statusnya berubah menjadi tersedia, Anda dapat terhubung ke instans DB. Bergantung pada kelas instans dan penyimpanan yang dialokasikan, perlu waktu beberapa menit agar instans DB baru tersedia.
    - Peran memiliki nilai Instans (RDS Custom).
    - Mode otomatisasi RDS Custom memiliki nilai Otomatisasi penuh. Pengaturan ini berarti bahwa instans DB menyediakan pemantauan otomatis dan pemulihan instans.

AWS CLI

Anda membuat instance RDS Custom DB dengan menggunakan create-db-instanceAWS CLIperintah.

Opsi berikut diperlukan:

- `--db-instance-identifier`
- `--db-instance-class` (untuk daftar kelas instans yang didukung, lihat Dukungan kelas instans DB untuk RDS Custom for Oracle)

- `--engine` *`engine-type`* (dengan *`engine-type`* adalah `custom-oracle-ee-cdb` untuk CDB dan `custom-oracle-ee` untuk non-CDB)
- `--engine-version` *`cev`* (dengan *`cev`* adalah nama versi mesin kustom yang Anda tentukan di Membuat CEV)
- `--kms-key-id` *`my-kms-key`*
- `--backup-retention-period` *`days`* (dengan nilai *`days`* lebih besar dari `0`)
- `--no-auto-minor-version-upgrade`
- `--custom-iam-instance-profile AWSRDSCustomInstanceRole-`*`us-east-1`* (dengan *`region`* merupakan Wilayah AWS tempat Anda membuat instans DB)

Contoh berikut membuat instans DB RDS Custom bernama `my-cfo-cdb-instance`. Basis data adalah CDB dengan nama nondefault *`MYCDB`*. Nama PDB nondefault adalah *`MYPDB`*. Periode retensi cadangan adalah tiga hari.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-instance \
    --engine custom-oracle-ee-cdb \
    --db-instance-identifier my-cfo-cdb-instance \
    --engine-version 19.cdb_cev1 \
    --db-name MYPDB \
    --db-system-id MYCDB \
    --allocated-storage 250 \
    --db-instance-class db.m5.xlarge \
    --db-subnet-group mydbsubnetgroup \
    --master-username myuser \
    --master-user-password mypassword \
    --backup-retention-period 3 \
    --port 8200 \
    --kms-key-id my-kms-key \
    --no-auto-minor-version-upgrade \
    --custom-iam-instance-profile AWSRDSCustomInstanceRole-us-east-1
```

Untuk Windows:

```
aws rds create-db-instance ^
    --engine custom-oracle-ee-cdb ^
```

```
    --db-instance-identifier my-cfo-cdb-instance ^
    --engine-version 19.cdb_cev1 ^
    --db-name MYPDB ^
    --db-system-id MYCDB ^
    --allocated-storage 250 ^
    --db-instance-class db.m5.xlarge ^
    --db-subnet-group mydbsubnetgroup ^
    --master-username myuser ^
    --master-user-password mypassword ^
    --backup-retention-period 3 ^
    --port 8200 ^
    --kms-key-id my-kms-key ^
    --no-auto-minor-version-upgrade ^
    --custom-iam-instance-profile AWSRDSCustomInstanceRole-us-east-1
```

> **Note**
>
> Tentukan kata sandi selain perintah yang ditampilkan di sini sebagai praktik keamanan terbaik.

Dapatkan detail tentang instans Anda menggunakan perintah `describe-db-instances`.

Example

```
aws rds describe-db-instances --db-instance-identifier my-cfo-cdb-instance
```

Output parsial berikut menunjukkan mesin, grup parameter, dan informasi lainnya.

```
        {
            "DBInstanceIdentifier": "my-cfo-cdb-instance",
            "DBInstanceClass": "db.m5.xlarge",
            "Engine": "custom-oracle-ee-cdb",
            "DBInstanceStatus": "available",
            "MasterUsername": "admin",
            "DBName": "MYPDB",
            "DBSystemID": "MYCDB",
            "Endpoint": {
                "Address": "my-cfo-cdb-instance.abcdefghijkl.us-
east-1.rds.amazonaws.com",
                "Port": 1521,
                "HostedZoneId": "A1B2CDEFGH34IJ"
```

```
            },
            "AllocatedStorage": 100,
            "InstanceCreateTime": "2023-04-12T18:52:16.353000+00:00",
            "PreferredBackupWindow": "08:46-09:16",
            "BackupRetentionPeriod": 7,
            "DBSecurityGroups": [],
            "VpcSecurityGroups": [
                {
                    "VpcSecurityGroupId": "sg-0a1bcd2e",
                    "Status": "active"
                }
            ],
            "DBParameterGroups": [
                {
                    "DBParameterGroupName": "default.custom-oracle-ee-cdb-19",
                    "ParameterApplyStatus": "in-sync"
                }
            ],
...
```

Peran tertaut layanan RDS Custom

Peran tertaut layanan memberi Amazon RDS Custom akses ke sumber daya di Akun AWS Anda. Hal ini membuat penggunaan RDS Custom lebih mudah karena Anda tidak perlu menambahkan izin yang diperlukan secara manual. RDS Custom menetapkan izin perannya yang tertaut layanan, dan kecuali ditetapkan lain, hanya RDS Custom yang dapat mengambil perannya. Izin yang ditentukan mencakup kebijakan kepercayaan dan kebijakan izin, dan kebijakan izin tersebut tidak dapat dilampirkan ke entitas IAM lainnya.

Saat Anda membuat instans DB RDS Custom, peran terkait layanan Amazon RDS dan RDS Custom dibuat (jika belum ada) dan digunakan. Untuk informasi selengkapnya, lihat Menggunakan peran terkait layanan untuk Amazon RDS.

Saat pertama kali membuat instans DB RDS Custom for Oracle, Anda mungkin menerima kesalahan berikut: Peran terkait layanan sedang dalam proses pembuatan. Coba lagi nanti. Jika ya, tunggu beberapa menit dan coba buat instans DB lagi.

Menghubungkan ke instans DB RDS Custom Anda menggunakan Session Manager

Setelah membuat instans DB RDS Custom, Anda dapat terhubung ke instans tersebut menggunakan AWS Systems Manager Session Manager. Ini adalah teknik yang lebih disukai ketika instans DB Anda tidak dapat diakses publik.

Session Manager memungkinkan Anda mengakses instans Amazon EC2 melalui shell berbasis browser atau melalui AWS CLI. Untuk informasi selengkapnya, lihat AWS Systems Manager Session Manager.

Konsol

Cara terhubung ke instans DB Anda menggunakan Session Manager

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data, kemudian pilih instans DB RDS Custom tempat Anda ingin terhubung.
3. Pilih Konfigurasi.
4. Catat ID Sumber Daya untuk instans DB Anda. Misalnya, ID sumber daya mungkin `db-ABCDEFGHIJKLMNOPQRS0123456`.
5. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
6. Di panel navigasi, pilih Instans.
7. Cari nama instans EC2 Anda, lalu klik ID instans yang terkait dengannya. Misalnya, ID instans mungkin `i-abcdefghijklm01234`.
8. Pilih Hubungkan.
9. Pilih Session Manager.
10. Pilih Hubungkan.

    Sebuah jendela terbuka untuk sesi Anda.

AWS CLI

Anda dapat terhubung ke instans DB RDS Custom menggunakan AWS CLI. Teknik ini membutuhkan plugin Session Manager untuk AWS CLI. Untuk mempelajari cara menginstal plugin, lihat Instal plugin Session Manager untuk AWS CLI.

Untuk menemukan ID sumber daya DB dari instans DB RDS Custom Anda, gunakan `aws rds describe-db-instances`.

```
aws rds describe-db-instances \
    --query 'DBInstances[*].[DBInstanceIdentifier,DbiResourceId]' \
    --output text
```

Output sampel berikut menunjukkan ID sumber daya untuk instans RDS Custom Anda. Prefiksnya adalah `db-`.

```
db-ABCDEFGHIJKLMNOPQRS0123456
```

Untuk menemukan ID instans EC2 dari instans DB Anda, gunakan `aws ec2 describe-instances`. Contoh berikut menggunakan `db-ABCDEFGHIJKLMNOPQRS0123456` untuk ID sumber daya.

```
aws ec2 describe-instances \
    --filters "Name=tag:Name,Values=db-ABCDEFGHIJKLMNOPQRS0123456" \
    --output text \
    --query 'Reservations[*].Instances[*].InstanceId'
```

Output sampel berikut menunjukkan ID instans EC2.

```
i-abcdefghijklm01234
```

Gunakan perintah `aws ssm start-session` yang menyediakan ID instans EC2 dalam parameter `--target`.

```
aws ssm start-session --target "i-abcdefghijklm01234"
```

Koneksi yang berhasil terlihat seperti berikut.

```
Starting session with SessionId: yourid-abcdefghijklm1234
[ssm-user@ip-123-45-67-89 bin]$
```

Menghubungkan ke instans DB RDS Custom Anda menggunakan SSH

Secure Shell Protocol (SSH) adalah protokol jaringan yang mendukung komunikasi terenkripsi melalui jaringan yang tidak aman. Setelah membuat instans DB RDS Custom, Anda dapat terhubung ke instans tersebut menggunakan klien ssh. Untuk informasi selengkapnya, lihat Terhubung ke instans Linux Anda menggunakan SSH.

Teknik koneksi SSH Anda bergantung pada apakah instans DB Anda bersifat pribadi, artinya tidak menerima koneksi dari internet publik. Dalam hal ini, Anda harus menggunakan tunneling SSH untuk menghubungkan utilitas ssh ke instans Anda. Teknik ini mengangkut data dengan aliran data khusus

(terowongan) di dalam sesi SSH yang ada. Anda dapat mengonfigurasi tunneling SSH menggunakan AWS Systems Manager.

> **Note**
>
> Berbagai strategi didukung untuk mengakses instans pribadi. Untuk mempelajari cara menghubungkan klien ssh ke instans pribadi menggunakan host bastion, lihat Host Bastion Linux Hosts di AWS. Untuk mempelajari cara mengonfigurasi penerusan port, lihat Penerusan Port Menggunakan AWS Systems Manager Session Manager.

Jika instans DB Anda berada di subnet publik dan memiliki pengaturan yang tersedia untuk umum, tidak diperlukan tunneling SSH. Anda dapat terhubung dengan SSH seperti halnya ke instans Amazon EC2 publik.

Untuk menghubungkan klien SSH ke instans DB Anda, selesaikan langkah-langkah berikut:

1. Langkah 1: Konfigurasi instans DB Anda untuk memungkinkan koneksi SSH
2. Langkah 2: Ambil kunci rahasia SSH dan ID instans EC2 Anda
3. Langkah 3: Terhubung ke instans EC2 Anda menggunakan utilitas SSH

Langkah 1: Konfigurasi instans DB Anda untuk memungkinkan koneksi SSH

Untuk memastikan instans DB Anda dapat menerima koneksi SSH, lakukan hal berikut:

- Pastikan grup keamanan instans DB Anda mengizinkan koneksi masuk pada port 22 untuk TCP.

  Untuk mempelajari cara mengonfigurasi grup keamanan bagi instans DB Anda, lihat Mengontrol akses dengan grup keamanan.
- Jika Anda tidak berencana untuk menggunakan tunneling SSH, pastikan instans DB Anda berada di subnet publik dan dapat diakses publik.

  Di konsol, bidang yang relevan dapat diakses publik di tab Konektivitas & keamanan pada halaman detail basis data. Untuk memeriksa pengaturan Anda di CLI, jalankan perintah berikut:

```
aws rds describe-db-instances \
--query 'DBInstances[*].
{DBInstanceIdentifier:DBInstanceIdentifier,PubliclyAccessible:PubliclyAccessible}' \
--output table
```

Untuk mengubah pengaturan aksesibilitas instans DB Anda, lihat Memodifikasi instans DB Amazon RDS.

Langkah 2: Ambil kunci rahasia SSH dan ID instans EC2 Anda

Untuk terhubung ke instans DB menggunakan SSH, Anda memerlukan pasangan kunci pair yang terkait dengan instans. RDS Custom membuat pasangan kunci SSH atas nama Anda, menamainya dengan prefiks `do-not-delete-rds-custom-ssh-privatekey-db-`. AWS Secrets Manager menyimpan kunci pribadi SSH Anda sebagai rahasia.

Ambil kunci rahasia SSH Anda menggunakan AWS Management Console atau AWS CLI. Jika instans Anda memiliki DNS publik dan Anda tidak ingin menggunakan tunneling SSH, ambil juga nama DNS. Anda menentukan nama DNS untuk koneksi publik.

Konsol

Cara mengambil kunci SSH rahasia

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data, kemudian pilih instans DB RDS Custom tempat Anda ingin terhubung.
3. Pilih Konfigurasi.
4. Perhatikan nilai ID Sumber Daya. Misalnya, ID sumber daya instans DB mungkin adalah `db-ABCDEFGHIJKLMNOPQRS0123456`.
5. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
6. Di panel navigasi, pilih Instans.
7. Temukan nama instans EC2 Anda dan pilih ID instans yang terkait dengannya. Misalnya, ID instans EC2 mungkin adalah `i-abcdefghijklm01234`.
8. Di bagian Detail, temukan Nama pasangan kunci. Nama pasangan menyertakan ID sumber daya instans DB. Misalnya, nama pasangan mungkin adalah `do-not-delete-rds-custom-ssh-privatekey-db-ABCDEFGHIJKLMNOPQRS0123456-0d726c`.
9. Jika instans EC2 Anda bersifat publik, perhatikan DNS IPv4 Publik. Misalnya, alamat Sistem Nama Domain (DNS) publik mungkin adalah `ec2-12-345-678-901.us-east-2.compute.amazonaws.com`.
10. Buka AWS Secrets Manager konsol di https://console.aws.amazon.com/secretsmanager/.

11. Pilih rahasia yang bernama sama dengan pasangan kunci Anda.
12. Pilih Ambil nilai rahasia.
13. Salin kunci privat SSH ke dalam file teks, kemudian simpan file dengan ekstensi `.pem`. Misalnya, simpan file sebagai `/tmp/do-not-delete-rds-custom-ssh-privatekey-db-ABCDEFGHIJKLMNOPQRS0123456-0d726c.pem`.

AWS CLI

Untuk mengambil kunci privat SSH dan menyimpannya dalam file .pem, Anda dapat menggunakan file AWS CLI.

1. Temukan ID sumber daya DB dari instans DB RDS Custom Anda menggunakan `aws rds describe-db-instances`.

```
aws rds describe-db-instances \
    --query 'DBInstances[*].[DBInstanceIdentifier,DbiResourceId]' \
    --output text
```

Output sampel berikut menunjukkan ID sumber daya untuk instans RDS Custom Anda. Prefiksnya adalah `db-`.

```
db-ABCDEFGHIJKLMNOPQRS0123456
```

2. Temukan ID instans EC2 dari instans DB Anda menggunakan `aws ec2 describe-instances`. Contoh berikut menggunakan `db-ABCDEFGHIJKLMNOPQRS0123456` untuk ID sumber daya.

```
aws ec2 describe-instances \
    --filters "Name=tag:Name,Values=db-ABCDEFGHIJKLMNOPQRS0123456" \
    --output text \
    --query 'Reservations[*].Instances[*].InstanceId'
```

Output sampel berikut menunjukkan ID instans EC2.

```
i-abcdefghijklm01234
```

3. Untuk menemukan nama kunci, tentukan ID instans EC2. Contoh berikut menjelaskan instans EC2 *i-0bdc4219e66944afa*.

```
aws ec2 describe-instances \
    --instance-ids i-0bdc4219e66944afa \
    --output text \
    --query 'Reservations[*].Instances[*].KeyName'
```

Output sampel berikut menunjukkan nama kunci yang menggunakan prefiks `do-not-delete-rds-custom-ssh-privatekey-`.

```
do-not-delete-rds-custom-ssh-privatekey-db-ABCDEFGHIJKLMNOPQRS0123456-0d726c
```

4. Simpan kunci privat dalam file .pem yang diberi nama menurut kunci menggunakan `aws secretsmanager`. Contoh berikut menyimpan file di direktori `/tmp` Anda.

```
aws secretsmanager get-secret-value \
    --secret-id do-not-delete-rds-custom-ssh-privatekey-db-
ABCDEFGHIJKLMNOPQRS0123456-0d726c \
    --query SecretString \
    --output text >/tmp/do-not-delete-rds-custom-ssh-privatekey-db-
ABCDEFGHIJKLMNOPQRS0123456-0d726c.pem
```

Langkah 3: Terhubung ke instans EC2 Anda menggunakan utilitas SSH

Teknik koneksi Anda bergantung pada apakah Anda terhubung ke instans DB pribadi atau terhubung ke instans publik. Koneksi pribadi mengharuskan Anda mengonfigurasi tunneling SSH melalui. AWS Systems Manager

Cara terhubung ke instans EC2 Anda menggunakan utilitas SSH

1. Untuk koneksi pribadi, ubah file konfigurasi SSH Anda menjadi perintah proksi AWS Systems Manager Session Manager. Untuk koneksi publik, lewati ke Langkah 2.

   Tambahkan baris berikut ke `~/.ssh/config`. Baris ini melakukan proksi perintah SSH untuk host yang namanya dimulai dengan `i-` atau `mi-`.

```
Host i-* mi-*
    ProxyCommand sh -c "aws ssm start-session --target %h --document-name AWS-
StartSSHSession --parameters 'portNumber=%p'"
```

2. Ubah ke direktori yang berisi file .pem Anda. Dengan menggunakan `chmod`, atur izin ke `400`.

```
cd /tmp
chmod 400 do-not-delete-rds-custom-ssh-privatekey-db-
ABCDEFGHIJKLMNOPQRS0123456-0d726c.pem
```

3. Jalankan utilitas ssh, tentukan file.pem dan nama DNS publik (untuk koneksi publik) atau ID instans EC2 (untuk koneksi pribadi). Masuk sebagai pengguna `ec2-user`.

   Contoh berikut menghubungkan ke instans publik menggunakan nama DNS *ec2-12-345-678-901.us-east-2.compute.amazonaws.com*.

   ```
   ssh -i \
       "do-not-delete-rds-custom-ssh-privatekey-db-
   ABCDEFGHIJKLMNOPQRS0123456-0d726c.pem" \
       ec2-user@ec2-12-345-678-901.us-east-2.compute.amazonaws.com
   ```

   Contoh berikut menghubungkan ke instans pribadi menggunakan ID instans EC2 *i-0bdc4219e66944afa*.

   ```
   ssh -i \
       "do-not-delete-rds-custom-ssh-privatekey-db-
   ABCDEFGHIJKLMNOPQRS0123456-0d726c.pem" \
       ec2-user@i-0bdc4219e66944afa
   ```

Masuk ke basis data RDS Custom for Oracle Anda sebagai SYS

Setelah membuat instans DB RDS Custom, Anda dapat masuk ke basis data Oracle sebagai pengguna SYS, yang memberi Anda hak istimewa SYSDBA. Anda memiliki opsi masuk berikut:

- Dapatkan kata sandi SYS dari Secrets Manager dan tentukan kata sandi ini di klien SQL Anda.
- Gunakan autentikasi OS untuk masuk ke basis data Anda. Dalam hal ini, Anda tidak memerlukan kata sandi.

Menemukan kata sandi SYS untuk basis data RDS Custom for Oracle

Anda dapat masuk ke basis data Oracle sebagai SYS atau SYSTEM, atau dengan menentukan nama pengguna master dalam panggilan API. Kata sandi untuk SYS dan SYSTEM disimpan di Secrets Manager. *Rahasianya menggunakan format penamaan do-not-delete-rds -custom-*

resource_id - uuid. Anda dapat menemukan kata sandi menggunakan AWS Management Console.

Konsol

Cara menemukan kata sandi SYS untuk basis data Anda di Secrets Manager

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di konsol RDS, selesaikan langkah-langkah berikut:
   a. Di panel navigasi, pilih Basis Data.
   b. Pilih nama instans DB RDS Custom for Oracle Anda.
   c. Pilih Konfigurasi.
   d. Salin nilai di bawah ID Sumber Daya. Misalnya, ID sumber daya mungkin adalah db-ABC12CDE3FGH4I5JKLMNO6PQR7.
3. Buka konsol Secrets Manager di https://console.aws.amazon.com/secretsmanager/.
4. Di konsol Secrets Manager, selesaikan langkah-langkah berikut:
   a. Pada panel navigasi kiri, pilih Rahasia.
   b. Filter rahasia berdasarkan ID sumber daya yang Anda salin di langkah 5.
   c. Pilih rahasia bernama do-not-delete-rds-custom- ***resource_id - uuid, di mana resource_id*** *adalah ID* sumber daya yang Anda salin di langkah 5. Misalnya, jika ID sumber daya Anda adalah DB-ABC12CDE3FGH4i5JKLmNo6PQR7, rahasia Anda akan diberi nama -Custom-DB-ABC12CDE3fGH4i5JKLmNo6PQR7. do-not-delete-rds
   d. Di bagian Nilai rahasia, pilih Ambil nilai rahasia.
   e. Di bagian Kunci/nilai, salin nilai untuk kata sandi.
5. Instal SQL*Plus pada instans DB Anda dan masuk ke basis data Anda sebagai SYS. Untuk informasi selengkapnya, lihat Langkah 3: Hubungkan klien SQL Anda ke instans DB Oracle.

Masuk ke basis data RDS Custom for Oracle Anda menggunakan autentikasi OS

Pengguna OS `rdsdb` memiliki biner basis data Oracle. Anda dapat beralih ke pengguna `rdsdb` dan masuk ke basis data RDS Custom for Oracle tanpa kata sandi.

1. Hubungkan ke instans DB Anda dengan AWS Systems Manager. Untuk informasi selengkapnya, lihat Menghubungkan ke instans DB RDS Custom Anda menggunakan Session Manager.
2. Di browser web, buka https://www.oracle.com/database/technologies/instant-client/linux-x86-64-downloads.html.
3. Untuk versi basis data terbaru yang muncul di halaman web, salin tautan .rpm (bukan tautan .zip) untuk Instant Client Basic Package dan SQL*Plus Package. Sebagai contoh, tautan berikut adalah untuk Oracle Database versi 21.9:

   - https://download.oracle.com/otn_software/linux/instantclient/219000/ oracle-instantclient-basic -21.9.0.0.0-1.el8.x86_64.rpm
   - https://download.oracle.com/otn_software/linux/instantclient/219000/ oracle-instantclient-sqlplus -21.9.0.0.0-1.el8.x86_64.rpm
4. Dalam sesi SSH Anda, jalankan perintah `wget` untuk mengunduh file .rpm dari tautan yang Anda peroleh pada langkah sebelumnya. Contoh berikut mengunduh file .rpm untuk Oracle Database versi 21.9:

```
wget https://download.oracle.com/otn_software/linux/instantclient/219000/oracle-instantclient-basic-21.9.0.0.0-1.el8.x86_64.rpm
wget https://download.oracle.com/otn_software/linux/instantclient/219000/oracle-instantclient-sqlplus-21.9.0.0.0-1.el8.x86_64.rpm
```

5. Instal paket dengan menjalankan perintah `yum` sebagai berikut:

```
sudo yum install oracle-instantclient-*.rpm
```

6. Beralih ke pengguna `rdsdb`.

```
sudo su - rdsdb
```

7. Masuk ke basis data Anda menggunakan autentikasi OS.

```
$ sqlplus / as sysdba

SQL*Plus: Release 21.0.0.0.0 - Production on Wed Apr 12 20:11:08 2023
Version 21.9.0.0.0

Copyright (c) 1982, 2020, Oracle.  All rights reserved.
```

```
Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.10.0.0.0
```

Menginstal komponen perangkat lunak tambahan pada instans DB RDS Custom for Oracle

Dalam instans DB yang baru dibuat, lingkungan basis data Anda mencakup biner Oracle, basis data, dan pendengar basis data. Anda mungkin ingin menginstal perangkat lunak tambahan pada sistem operasi host instans DB. Misalnya, Anda mungkin ingin menginstal Oracle Application Express (APEX), agen Oracle Enterprise Manager (OEM), atau agen Guardium S-TAP. Untuk panduan dan petunjuk tingkat tinggi, lihat postingan blog AWS terperinci Install additional software components on Amazon RDS Custom for Oracle.

Mengelola instans DB Amazon RDS Custom for Oracle

Amazon RDS Custom mendukung subset tugas manajemen biasa untuk instans DB Amazon RDS. Di bawah ini, Anda dapat menemukan petunjuk untuk tugas manajemen RDS Custom for Oracle yang didukung menggunakan AWS Management Console dan AWS CLI.

Topik

Menangani basis data kontainer (CDB) di RDS Custom for Oracle

Anda dapat membuat instans DB RDS Custom for Oracle dengan arsitektur Multipenyewa Oracle (jenis mesin `custom-oracle-ee-cdb`) atau dengan arsitektur non-CDB tradisional (jenis mesin `custom-oracle-ee`). Saat Anda membuat basis data kontainer (CDB), basis data tersebut berisi satu basis data pluggable (PDB) dan satu seed PDB. Anda dapat membuat PDB tambahan secara manual menggunakan Oracle SQL.

Nama PDB dan CDB

Saat membuat instans CDB RDS Custom for Oracle, Anda menentukan nama untuk PDB awal. Secara default, PDB awal Anda diberi nama `ORCL`. Anda dapat memilih nama yang berbeda.

Secara default, CDB Anda diberi nama `RDSCDB`. Anda dapat memilih nama yang berbeda. Nama CDB juga merupakan nama pengidentifikasi sistem (SID) Oracle Anda, yang secara unik mengidentifikasi memori dan proses yang mengelola CDB Anda. Untuk informasi lebih lanjut tentang SID Oracle, lihat Oracle System Identifier (SID) di Oracle Database Concepts.

Anda tidak dapat mengganti nama PDB yang ada menggunakan API Amazon RDS. Anda juga tidak dapat mengganti nama CDB menggunakan perintah `modify-db-instance`.

Manajemen PDB

Dalam model tanggung jawab bersama RDS Custom for Oracle, Anda bertanggung jawab untuk mengelola PDB dan membuat PDB tambahan. RDS Custom tidak membatasi jumlah PDB. Anda dapat membuat, memodifikasi, dan menghapus PDB secara manual dengan menghubungkan ke root CDB dan menjalankan pernyataan SQL. Buat PDB pada volume data Amazon EBS untuk mencegah instans DB keluar dari perimeter dukungan.

Untuk memodifikasi CDB atau PDB Anda, selesaikan langkah-langkah berikut:

1. Jeda otomatisasi untuk mencegah gangguan terhadap tindakan RDS Custom.
2. Ubah CDB atau PDB Anda.
3. Cadangkan semua PDB yang dimodifikasi.
4. Lanjutkan otomatisasi RDS Custom.

Pemulihan otomatis root CDB

RDS Custom membuat root CDB tetap terbuka dengan cara yang sama seperti membuat non-CDB tetap terbuka. Jika keadaan root CDB berubah, otomatisasi pemantauan dan pemulihan mencoba memulihkan root CDB ke keadaan yang diinginkan. Anda menerima pemberitahuan peristiwa RDS saat root CDB dimatikan (`RDS-EVENT-0004`) atau dimulai ulang (`RDS-EVENT-0006`), mirip dengan arsitektur non-CDB. RDS Custom mencoba untuk membuka semua PDB dalam mode `READ WRITE` saat memulai instans DB. Jika beberapa PDB tidak dapat dibuka, RDS Custom menerbitkan peristiwa berikut: `tenant database shutdown`.

Menggunakan fitur ketersediaan tinggi untuk RDS Custom for Oracle

Untuk mendukung replikasi antara instans RDS Custom for Oracle, Anda dapat mengonfigurasi ketersediaan tinggi (HA) dengan Oracle Data Guard. Instans DB primer secara otomatis menyinkronkan data ke instans siaga.

Anda dapat mengonfigurasi lingkungan ketersediaan tinggi dengan cara berikut:

- Konfigurasikan instans siaga di Zona Ketersediaan (AZ) yang berbeda agar tahan terhadap kegagalan AZ.
- Tempatkan basis data siaga Anda dalam mode terpasang atau hanya baca.
- Failover atau beralih dari basis data primer ke basis data siaga tanpa kehilangan data.

- Migrasikan data dengan mengonfigurasi ketersediaan tinggi untuk instans on-premise Anda, lalu lakukan failover atau beralih ke basis data siaga RDS Custom.

Untuk mempelajari cara mengonfigurasi ketersediaan tinggi, lihat laporan resmi Membangun ketersediaan tinggi untuk Amazon RDS Custom for Oracle menggunakan replika baca. Anda dapat melakukan tugas-tugas berikut:

- Gunakan terowongan jaringan privat virtual (VPN) guna mengenkripsi data bergerak untuk instans ketersediaan tinggi Anda. Enkripsi saat bergerak tidak dikonfigurasi secara otomatis oleh RDS Custom.
- Konfigurasikan Oracle Fast-Failover Observer (FSFO) untuk memantau instans ketersediaan tinggi Anda.
- Izinkan pengamat untuk melakukan failover otomatis ketika kondisi yang diperlukan terpenuhi.

Menyesuaikan lingkungan RDS Custom Anda

RDS Custom for Oracle menyertakan fitur bawaan yang memungkinkan Anda menyesuaikan lingkungan instans DB tanpa menghentikan otomatisasi. Misalnya, Anda dapat menggunakan API RDS untuk menyesuaikan lingkungan Anda dengan cara berikut:

- Buat dan pulihkan snapshot DB untuk membuat lingkungan klona.
- Buat replika baca.
- Ubah pengaturan penyimpanan.
- Ubah CEV untuk menerapkan pembaruan rilis

Untuk beberapa penyesuaian, seperti mengubah zona waktu atau set karakter, Anda tidak dapat menggunakan API RDS. Dalam kasus ini, Anda perlu mengubah lingkungan secara manual dengan mengakses instans Amazon EC2 sebagai pengguna root atau masuk ke basis data Oracle sebagai `SYSDBA`.

Untuk menyesuaikan instans secara manual, Anda harus menjeda dan melanjutkan otomatisasi RDS Custom. Jeda ini memastikan bahwa penyesuaian Anda tidak mengganggu otomatisasi RDS Custom. Dengan cara ini, Anda menghindari melanggar perimeter dukungan, yang menempatkan instans dalam status `unsupported-configuration` sampai Anda memperbaiki masalah yang mendasarinya. Menjeda dan melanjutkan adalah satu-satunya tugas otomatisasi yang didukung saat memodifikasi instans DB RDS Custom for Oracle.

Langkah-langkah umum untuk menyesuaikan lingkungan RDS Custom

Untuk menyesuaikan instans DB RDS Custom Anda, selesaikan langkah-langkah berikut:

1. Jeda otomatisasi RDS Custom selama periode tertentu menggunakan konsol atau CLI.
2. Identifikasi instans Amazon EC2 yang mendasari.
3. Terhubung ke instans Amazon EC2 yang mendasari menggunakan kunci SSH atau AWS Systems Manager.
4. Verifikasi pengaturan konfigurasi Anda saat ini di basis data atau lapisan sistem operasi.

   Anda dapat memvalidasi perubahan dengan membandingkan konfigurasi awal dengan konfigurasi yang diubah. Bergantung pada jenis penyesuaian, gunakan alat OS atau kueri basis data.
5. Sesuaikan instans DB RDS Custom for Oracle sesuai kebutuhan.
6. Boot ulang instans atau basis data Anda jika perlu.

   > **Note**
   >
   > Di CDB Oracle on-premise, Anda dapat mempertahankan mode terbuka yang ditentukan untuk PDB menggunakan perintah bawaan atau setelah pemicu startup. Mekanisme ini menentukan status PDB saat CDB dimulai ulang. Saat membuka CDB Anda, otomatisasi RDS Custom membuang status tersimpan yang ditentukan pengguna dan mencoba membuka semua PDB. Jika RDS Custom tidak dapat membuka semua PDB, peristiwa berikut akan muncul: `The following PDBs failed to open: list-of-PDBs`.
7. Verifikasi pengaturan konfigurasi baru Anda dengan membandingkannya dengan pengaturan sebelumnya.
8. Lanjutkan otomatisasi RDS Custom dengan salah satu cara berikut:
   - Lanjutkan otomatisasi secara manual.
   - Tunggu hingga periode jeda berakhir. Dalam hal ini, RDS Custom melanjutkan pemantauan dan pemulihan instans secara otomatis.
9. Verifikasi kerangka kerja otomatisasi RDS Custom

   Jika Anda mengikuti langkah-langkah sebelumnya dengan benar, RDS Custom memulai pencadangan otomatis. Status instans di konsol menunjukkan Tersedia.

Untuk praktik dan step-by-step petunjuk terbaik, lihat posting AWS blog Membuat perubahan konfigurasi pada instans Amazon RDS Custom for Oracle: Bagian 1 dan Buat Ulang Kustom Amazon RDS untuk database Oracle: Bagian 2.

Menjeda dan melanjutkan instans RDS Custom DB

Anda dapat menjeda dan melanjutkan otomatisasi untuk instans DB menggunakan konsol atau CLI.

Konsol

Cara menjeda atau melanjutkan otomatisasi RDS Custom

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data, lalu pilih instans DB RDS Custom yang ingin Anda ubah.
3. Pilih Ubah. Halaman Modifikasi instans DB muncul.
4. Untuk Mode otomatisasi RDS Custom, pilih salah satu opsi berikut:
   - Dijeda akan menjeda pemantauan dan pemulihan instans untuk instans DB RDS Custom. Masukkan durasi jeda yang Anda inginkan (dalam hitungan menit) untuk Durasi mode otomatisasi. Nilai minimum adalah 60 menit (default). Nilai maksimum adalah 1.440 menit.
   - Otomatisasi penuh akan melanjutkan otomatisasi.
5. Pilih Lanjutkan untuk memeriksa ringkasan perubahan.

   Sebuah pesan menunjukkan bahwa RDS Custom akan segera menerapkan perubahan.
6. Jika perubahan Anda benar, pilih Modifikasi instans DB. Anda juga dapat memilih Kembali untuk mengedit perubahan atau Batal untuk membatalkan perubahan.

   Pada konsol RDS, detail untuk modifikasi muncul. Jika Anda menjeda otomatisasi, Status instans DB RDS Custom Anda menunjukkan Otomatisasi dihentikan sementara.
7. (Opsional) Di panel navigasi, pilih Basis Data, lalu pilih instans DB RDS Custom.

   Di panel Ringkasan, Mode otomatisasi RDS Custom menunjukkan status otomatisasi. Jika otomatisasi dijeda, nilainya adalah Dihentikan sementara. Otomatisasi dilanjutkan dalam ***hitungan*** menit.

AWS CLI

Untuk menjeda atau melanjutkan otomatisasi RDS Custom, gunakan perintah `modify-db-instance` AWS CLI. Identifikasi instans DB menggunakan parameter `--db-instance-identifier` yang diperlukan. Kontrol mode otomatisasi dengan parameter berikut:

- `--automation-mode` menentukan status jeda instans DB. Nilai yang valid adalah `all-paused` yang menghentikan otomatisasi, dan `full` yang melanjutkannya.
- `--resume-full-automation-mode-minutes` menentukan durasi jeda. Nilai default adalah 60 menit.

> **Note**
>
> Terlepas dari apakah Anda menentukan `--no-apply-immediately` atau `--apply-immediately`, RDS Custom menerapkan modifikasi secara asinkron sesegera mungkin.

Dalam respons perintah, `ResumeFullAutomationModeTime` menunjukkan waktu untuk melanjutkan dalam timestamp UTC. Saat mode otomatisasi adalah `all-paused`, Anda dapat menggunakan `modify-db-instance` untuk melanjutkan mode otomatisasi atau memperpanjang periode jeda. Tidak ada opsi `modify-db-instance` lain yang didukung.

Contoh berikut menjeda otomatisasi `my-custom-instance` selama 90 menit.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --automation-mode all-paused \
    --resume-full-automation-mode-minutes 90
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --automation-mode all-paused ^
    --resume-full-automation-mode-minutes 90
```

Contoh berikut memperpanjang durasi jeda selama 30 menit. 30 menit ditambahkan ke waktu awal yang ditunjukkan dalam `ResumeFullAutomationModeTime`.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --automation-mode all-paused \
    --resume-full-automation-mode-minutes 30
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --automation-mode all-paused ^
    --resume-full-automation-mode-minutes 30
```

Contoh berikut melanjutkan otomatisasi penuh untuk `my-custom-instance`.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --automation-mode full \
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --automation-mode full
```

Dalam output sampel parsial berikut, nilai `AutomationMode` yang tertunda adalah `full`.

```
{
    "DBInstance": {
        "PubliclyAccessible": true,
        "MasterUsername": "admin",
        "MonitoringInterval": 0,
```

```
        "LicenseModel": "bring-your-own-license",
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "0123456789abcdefg"
            }
        ],
        "InstanceCreateTime": "2020-11-07T19:50:06.193Z",
        "CopyTagsToSnapshot": false,
        "OptionGroupMemberships": [
            {
                "Status": "in-sync",
                "OptionGroupName": "default:custom-oracle-ee-19"
            }
        ],
        "PendingModifiedValues": {
            "AutomationMode": "full"
        },
        "Engine": "custom-oracle-ee",
        "MultiAZ": false,
        "DBSecurityGroups": [],
        "DBParameterGroups": [
            {
                "DBParameterGroupName": "default.custom-oracle-ee-19",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        ...
        "ReadReplicaDBInstanceIdentifiers": [],
        "AllocatedStorage": 250,
        "DBInstanceArn": "arn:aws:rds:us-west-2:012345678912:db:my-custom-instance",
        "BackupRetentionPeriod": 3,
        "DBName": "ORCL",
        "PreferredMaintenanceWindow": "fri:10:56-fri:11:26",
        "Endpoint": {
            "HostedZoneId": "ABCDEFGHIJKLMNO",
            "Port": 8200,
            "Address": "my-custom-instance.abcdefghijk.us-west-2.rds.amazonaws.com"
        },
        "DBInstanceStatus": "automation-paused",
        "IAMDatabaseAuthenticationEnabled": false,
        "AutomationMode": "all-paused",
        "EngineVersion": "19.my_cev1",
        "DeletionProtection": false,
```

```
        "AvailabilityZone": "us-west-2a",
        "DomainMemberships": [],
        "StorageType": "gp2",
        "DbiResourceId": "db-ABCDEFGHIJKLMNOPQRSTUVW",
        "ResumeFullAutomationModeTime": "2020-11-07T20:56:50.565Z",
        "KmsKeyId": "arn:aws:kms:us-west-2:012345678912:key/
aa111a11-111a-11a1-1a11-1111a11a1a1a",
        "StorageEncrypted": false,
        "AssociatedRoles": [],
        "DBInstanceClass": "db.m5.xlarge",
        "DbInstancePort": 0,
        "DBInstanceIdentifier": "my-custom-instance",
        "TagList": []
    }
```

Memodifikasi instans DB RDS Custom for Oracle

Memodifikasi instans DB RDS Custom for Oracle sama dengan memodifikasi instans DB Amazon RDS. Anda dapat mengubah pengaturan seperti berikut ini:

- Kelas instans DB
- Alokasi dan jenis penyimpanan
- Periode retensi cadangan
- Perlindungan penghapusan
- Grup opsi
- CEV (lihat Memutakhirkan instans basis data RDS Custom for Oracle)
- Port

Topik

- Persyaratan dan batasan saat memodifikasi penyimpanan instans DB
- Persyaratan dan batasan saat memodifikasi kelas instans DB
- Cara RDS Custom membuat instans DB saat Anda memodifikasi kelas instans
- Memodifikasi instans DB RDS Custom for Oracle

Persyaratan dan batasan saat memodifikasi penyimpanan instans DB

Pertimbangkan persyaratan dan batasan berikut saat Anda memodifikasi penyimpanan untuk instans DB RDS Custom for Oracle:

- Penyimpanan minimum yang dialokasikan untuk RDS Custom for Oracle adalah 40 GiB, dan maksimum 64 TiB.
- Seperti Amazon RDS, Anda tidak dapat mengurangi penyimpanan yang dialokasikan. Ini adalah pembatasan volume Amazon EBS.
- Penskalaan otomatis penyimpanan tidak didukung untuk instans DB RDS Custom for Oracle.
- Setiap volume penyimpanan yang Anda lampirkan secara manual ke instans DB RDS Custom berada di luar perimeter dukungan.

  Untuk informasi selengkapnya, lihat Perimeter dukungan RDS Custom.
- Penyimpanan Amazon EBS magnetik (standar) tidak didukung untuk RDS Custom. Anda hanya dapat memilih jenis penyimpanan SSD io1, gp2, atau gp3.

Untuk informasi selengkapnya tentang penyimpanan Amazon EBS, lihat Penyimpanan instans DB Amazon RDS. Untuk informasi umum tentang modifikasi penyimpanan, lihat Menggunakan penyimpanan untuk instans DB Amazon RDS.

Persyaratan dan batasan saat memodifikasi kelas instans DB

Pertimbangkan persyaratan dan batasan berikut saat Anda memodifikasi kelas instans untuk instans DB RDS Custom for Oracle:

- Instans DB Anda harus berada dalam status `available`.
- Instans DB Anda harus memiliki ruang kosong minimal 100 MiB pada volume root, volume data, dan volume biner.
- Anda hanya dapat menetapkan satu IP elastis (EIP) ke instans DB RDS Custom for Oracle saat menggunakan antarmuka jaringan elastis (ENI) default. Jika Anda melampirkan beberapa ENI ke instans DB, operasi modifikasi akan gagal.
- Semua tag RDS Custom for Oracle harus ada.
- Jika Anda menggunakan replikasi RDS Custom for Oracle, perhatikan persyaratan dan batasan berikut:
  - Untuk instans DB dan replika baca primer, Anda dapat mengubah kelas instans hanya untuk satu instans DB pada satu waktu.

- Jika instans DB RDS Custom for Oracle Anda memiliki basis data primer atau replika on-premise, pastikan untuk memperbarui alamat IP pribadi secara manual pada instans DB on-premise setelah modifikasi selesai. Tindakan ini diperlukan untuk melestarikan DataGuard fungsionalitas Oracle. RDS Custom for Oracle menerbitkan peristiwa ketika modifikasi berhasil.
- Anda tidak dapat memodifikasi kelas instans DB RDS Custom for Oracle saat instans DB primer atau replika baca memiliki konfigurasi FSFO (Fast-Start Failover).

Cara RDS Custom membuat instans DB saat Anda memodifikasi kelas instans

Ketika Anda memodifikasi kelas instans, RDS Custom membuat instans DB seperti berikut:

- Membuat instans Amazon EC2.
- Membuat volume root dari snapshot DB terbaru. RDS Custom for Oracle tidak mempertahankan informasi yang ditambahkan ke volume root setelah snapshot DB terbaru.
- Membuat CloudWatch alarm Amazon.
- Membuat pasangan kunci SSH Amazon EC2 jika Anda telah menghapus pasangan kunci asli. Jika tidak, RDS Custom for Oracle mempertahankan pasangan kunci asli.
- Membuat sumber daya baru menggunakan tag yang dilampirkan ke instans DB Anda saat Anda memulai modifikasi. RDS Custom tidak mentransfer tag ke sumber daya baru ketika dilampirkan langsung ke sumber daya yang mendasarinya.
- Mentransfer volume biner dan data dengan modifikasi terbaru ke instans DB baru.
- Mentransfer alamat IP elastis (EIP). Jika instans DB dapat diakses publik, RDS Custom melampirkan alamat IP publik ke instans DB baru untuk sementara sebelum mentransfer EIP. Jika instans DB tidak dapat diakses publik, RDS Custom tidak membuat alamat IP publik.

Memodifikasi instans DB RDS Custom for Oracle

Anda dapat memodifikasi kelas atau penyimpanan instans DB menggunakan konsol, AWS CLI, atau API RDS.

Konsol

Cara memodifikasi instans DB RDS Custom for Oracle

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.

2. Di panel navigasi, pilih Basis data.
3. Pilih instans DB yang ingin Anda modifikasi.
4. Pilih Ubah.
5. (Opsional) Dalam Konfigurasi instans, pilih nilai untuk Kelas instans DB. Untuk kelas yang didukung, lihat Dukungan kelas instans DB untuk RDS Custom for Oracle.
6. (Opsional) Di bagian Penyimpanan, lakukan perubahan berikut sesuai kebutuhan:
   a. Masukkan nilai baru untuk Penyimpanan yang dialokasikan. Nilai ini harus lebih besar dari nilai saat ini, dan antara 40 GiB–64 TiB.
   b. Ubah nilai untuk Jenis penyimpanan menjadi SSD Tujuan Umum (gp2), SSD Tujuan Umum (gp3), atau IOPS yang Tersedia (io1).
   c. Jika menggunakan IOPS yang Tersedia (io1) atau SSD Tujuan Umum (gp3), Anda dapat mengubah nilai IOPS yang Tersedia.
7. (Opsional) Di bagian Konfigurasi tambahan, lakukan perubahan berikut sesuai kebutuhan:
   - Untuk Grup opsi, pilih grup opsi baru. Untuk informasi selengkapnya, lihat Menggunakan grup opsi di RDS Custom for Oracle.
8. Pilih Lanjutkan.
9. Pilih Terapkan segera atau Terapkan di jendela pemeliharaan terjadwal berikutnya.
10. Pilih Ubah instans DB.

AWS CLI

Untuk memodifikasi penyimpanan untuk RDS Custom untuk instans Oracle DB, gunakan perintah. modify-db-instanceAWS CLI Atur parameter berikut sesuai kebutuhan:

- `--db-instance-class` – Kelas instans baru. Untuk kelas yang didukung, lihat Dukungan kelas instans DB untuk RDS Custom for Oracle.
- `--allocated-storage` – Jumlah penyimpanan yang akan dialokasikan untuk instans DB, dalam gibibyte. Nilai ini harus lebih besar dari nilai saat ini, dan antara 40–65.536 GiB.
- `--storage-type` – Jenis penyimpanan: gp2, gp3, atau io1.
- `--iops` – IOPS yang tersedia untuk instans DB, jika menggunakan jenis penyimpanan io1 atau gp3.
- `--apply-immediately` – Gunakan `--apply-immediately` untuk langsung menerapkan perubahan penyimpanan.

Gunakan `--no-apply-immediately` (default) untuk menerapkan perubahan saat jendela pemeliharaan berikutnya.

Contoh berikut mengubah kelas instance DB my-cfo-instance untuk db.m5.16xlarge. Perintah ini juga mengubah ukuran penyimpanan menjadi 1 TiB, jenis penyimpanan menjadi io1, IOPS yang Tersedia menjadi 3000, dan grup opsi menjadi cfo-ee-19-mt.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-cfo-instance \
    --db-instance-class db.m5.16xlarge \
    --storage-type io1 \
    --iops 3000 \
    --allocated-storage 1024 \
    --option-group cfo-ee-19-mt \
    --apply-immediately
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-cfo-instance ^
    --db-instance-class db.m5.16xlarge ^
    --storage-type io1 ^
    --iops 3000 ^
    --allocated-storage 1024 ^
    --option-group cfo-ee-19-mt ^
    --apply-immediately
```

Mengubah set karakter instans DB RDS Custom for Oracle

Set karakter RDS Custom for Oracle default adalah US7ASCII. Anda mungkin ingin menentukan set karakter yang berbeda untuk memenuhi persyaratan bahasa atau karakter multibyte. Jika menggunakan RDS Custom for Oracle, Anda dapat menjeda otomatisasi dan kemudian mengubah set karakter basis data secara manual.

Mengubah set karakter instans DB RDS Custom for Oracle memiliki persyaratan berikut:

- Anda hanya dapat mengubah karakter pada instans RDS Custom yang baru tersedia yang memiliki basis data kosong atau starter tanpa data aplikasi. Untuk semua skenario lainnya, ubah set karakter menggunakan DMU (Database Migration Assistant for Unicode).
- Anda hanya dapat mengubah ke set karakter yang didukung oleh RDS for Oracle. Untuk informasi selengkapnya, lihat Set karakter DB yang didukung.

Cara mengubah set karakter instans DB RDS Custom for Oracle

1. Jeda otomatisasi RDS Custom. Untuk informasi selengkapnya, lihat Menjeda dan melanjutkan instans RDS Custom DB.
2. Masuk ke basis data Anda sebagai pengguna dengan hak istimewa SYSDBA.
3. Mulai ulang basis data dalam mode terbatas, ubah set karakter, dan kemudian mulai ulang basis data dalam mode normal.

   Jalankan skrip berikut di klien SQL Anda:

   ```
   SHUTDOWN IMMEDIATE;
   STARTUP RESTRICT;
   ALTER DATABASE CHARACTER SET INTERNAL_CONVERT AL32UTF8;
   SHUTDOWN IMMEDIATE;
   STARTUP;
   SELECT VALUE FROM NLS_DATABASE_PARAMETERS WHERE PARAMETER = 'NLS_CHARACTERSET';
   ```

   Verifikasi bahwa output menunjukkan set karakter yang benar:

   ```
   VALUE
   --------
   AL32UTF8
   ```
4. Lanjutkan otomatisasi RDS Custom. Untuk informasi selengkapnya, lihat Menjeda dan melanjutkan instans RDS Custom DB.

Mengatur nilai NLS_LANG di RDS Custom for Oracle

Lokal adalah serangkaian informasi yang membahas persyaratan bahasa dan budaya yang sesuai dengan bahasa dan negara tertentu. Guna menentukan perilaku lokal untuk perangkat lunak Oracle, atur variabel lingkungan NLS_LANG pada host klien Anda. Variabel ini mengatur bahasa, wilayah, dan set karakter yang digunakan oleh aplikasi klien dan server basis data.

Untuk RDS Custom for Oracle, Anda hanya dapat mengatur bahasa dalam variabel NLS_LANG: wilayah dan karakter menggunakan default. Bahasa ini digunakan untuk pesan, pemeriksaan, nama hari, dan nama bulan basis data Oracle. Setiap bahasa yang didukung memiliki nama yang unik, misalnya, Amerika, Prancis, atau Jerman. Jika bahasa tidak ditentukan, nilai default-nya adalah Amerika.

Setelah membuat basis data RDS Custom for Oracle, Anda dapat mengatur NLS_LANG di host klien Anda ke bahasa selain bahasa Inggris. Untuk melihat daftar bahasa yang didukung oleh Oracle Database, masuk ke basis data RDS Custom for Oracle dan jalankan kueri berikut:

```
SELECT VALUE FROM V$NLS_VALID_VALUES WHERE PARAMETER='LANGUAGE' ORDER BY VALUE;
```

Anda dapat mengatur NLS_LANG pada baris perintah host. Contoh berikut menetapkan bahasa ke bahasa Jerman untuk aplikasi klien Anda menggunakan shell Z di Linux.

```
export NLS_LANG=German
```

Aplikasi Anda membaca nilai NLS_LANG saat dimulai dan kemudian mengomunikasikannya ke basis data saat terhubung.

Untuk informasi selengkapnya, lihat Memilih Lokal dengan Variabel Lingkungan NLS_LANG di Oracle Database Globalization Support Guide.

Dukungan untuk Enkripsi Data Transparan

RDS Custom mendukung Enkripsi Data Transparan (TDE) untuk instans DB RDS Custom for Oracle.

Namun, Anda tidak dapat mengaktifkan TDE menggunakan opsi dalam grup opsi khusus seperti di RDS for Oracle. Anda mengaktifkan TDE secara manual. Untuk informasi tentang penggunaan Enkripsi Data Transparan Oracle, lihat Securing stored data using Transparent Data Encryption di dokumentasi Oracle.

Menandai sumber daya RDS Custom for Oracle

Anda dapat menandai sumber daya RDS Custom sama seperti sumber daya Amazon RDS, tetapi dengan beberapa perbedaan penting:

- Jangan membuat atau memodifikasi tag AWSRDSCustom yang diperlukan untuk otomatisasi RDS Custom. Jika melakukannya, Anda dapat merusak otomatisasi.

- Tag `Name` ditambahkan ke sumber daya RDS Custom dengan nilai prefiks `do-not-delete-rds-custom`. Setiap nilai yang diteruskan pelanggan untuk kunci akan ditimpa.
- Tag yang ditambahkan ke instans DB RDS Custom selama pembuatan disebarkan ke semua sumber daya RDS Custom terkait lainnya.
- Tag tidak disebarkan saat Anda menambahkannya ke sumber daya RDS Custom setelah pembuatan instans DB.

Untuk informasi umum tentang penandaan sumber daya, lihat Memberi tag pada sumber daya Amazon RDS.

Menghapus instans DB RDS Custom for Oracle

Untuk menghapus instans DB RDS Custom, lakukan hal berikut:

- Berikan nama instans DB.
- Hapus opsi untuk mengambil snapshot DB akhir dari instans DB.
- Pilih atau hapus opsi untuk mempertahankan cadangan otomatis.

Anda dapat menghapus instans DB RDS Custom menggunakan konsol atau CLI. Waktu yang diperlukan untuk menghapus instans DB dapat bervariasi bergantung pada periode retensi cadangan (yaitu, berapa banyak cadangan yang harus dihapus) dan berapa banyak data yang dihapus.

Konsol

Cara menghapus instans DB RDS Custom

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data, lalu pilih instans DB RDS Custom yang ingin dihapus. Instans DB RDS Custom menunjukkan peran Instans (RDS Custom).
3. Untuk Tindakan, pilih Hapus.
4. Untuk mempertahankan cadangan otomatis, pilih Pertahankan cadangan otomatis.
5. Masukkan **`delete me`** di kotak teks.
6. Pilih Hapus.

AWS CLI

Anda menghapus instans RDS Custom DB dengan menggunakan delete-db-instanceAWS CLIperintah. Identifikasi instans DB menggunakan parameter `--db-instance-identifier` yang diperlukan. Parameter yang tersisa sama dengan instans DB Amazon RDS, dengan pengecualian berikut:

- `--skip-final-snapshot` diperlukan.
- `--no-skip-final-snapshot` tidak didukung.
- `--final-db-snapshot-identifier` tidak didukung.

Contoh berikut menghapus instans DB RDS Custom bernama `my-custom-instance` dan mempertahankan cadangan otomatis.

Example

Untuk Linux, macOS, atau Unix:

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

Untuk Windows:

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

Menggunakan replika Oracle untuk RDS Custom for Oracle

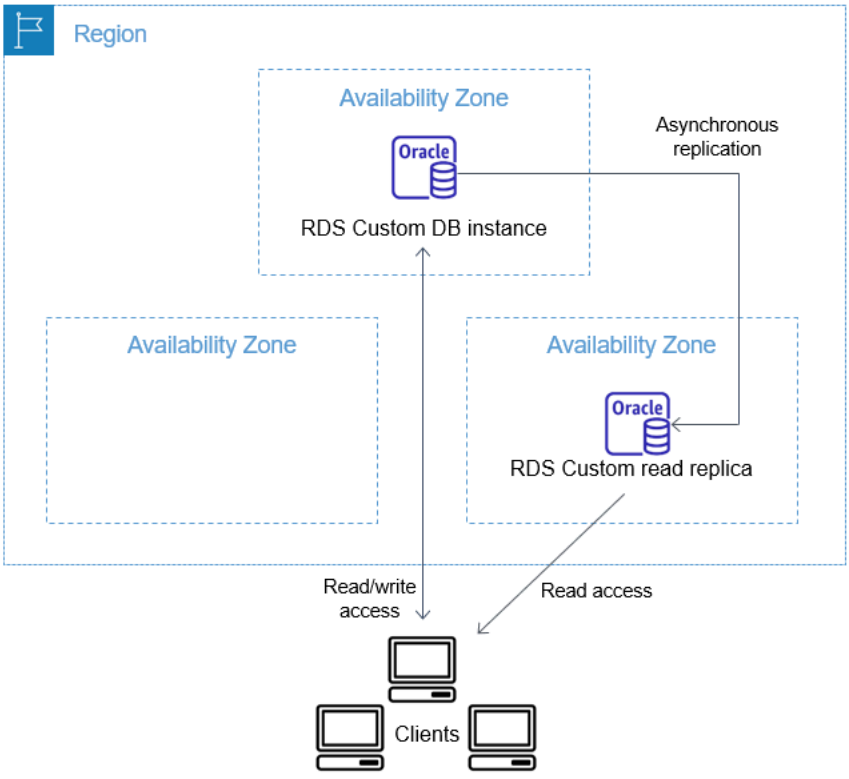
Anda dapat membuat replika Oracle untuk instans DB RDS Custom for Oracle. Basis data kontainer (CDB) dan non-CDB didukung. Membuat replika RDS Custom for Oracle mirip dengan membuat replika RDS for Oracle, tetapi dengan perbedaan penting. Untuk informasi umum tentang membuat dan mengelola replika Oracle, lihat Menggunakan replika baca instans DB dan Menggunakan replika baca untuk Amazon RDS for Oracle.

Topik

- Gambaran umum replikasi RDS Custom for Oracle
- Pedoman dan batasan untuk replikasi RDS Custom for Oracle
- Mempromosikan replika RDS Custom for Oracle ke instans DB mandiri

Gambaran umum replikasi RDS Custom for Oracle

Arsitektur replikasi RDS Custom for Oracle sebanding dengan replikasi RDS for Oracle. Instans DB primer direplikasi secara asinkron ke satu atau beberapa replika Oracle.



Jumlah maksimum replika

Seperti halnya RDS for Oracle, Anda dapat membuat hingga lima replika Oracle terkelola dari instans DB primer RDS Custom for Oracle Anda. Anda juga dapat membuat replika Oracle (eksternal) Anda sendiri yang dikonfigurasi secara manual. Replika eksternal tidak dihitung terhadap batas instans DB Anda. Replika ini juga terletak di luar perimeter dukungan RDS Custom. Untuk informasi selengkapnya tentang perimeter dukungan, lihat Perimeter dukungan RDS Custom.

Konvensi penamaan replika

Nama replika Oracle didasarkan pada nama unik basis data. Formatnya adalah *DB_UNIQUE_NAME_X*, dengan huruf ditambahkan secara berurutan. Misalnya, jika nama unik basis data Anda adalah `ORCL`, dua replika pertama akan diberi nama `ORCL_A` dan `ORCL_B`. Enam huruf pertama, A–F, dialokasikan untuk RDS Custom. RDS Custom menyalin parameter basis data dari instans DB primer Anda ke replika. Untuk informasi selengkapnya, lihat DB_UNIQUE_NAME dalam dokumentasi Oracle.

Retensi cadangan replika

Secara default, replika RDS Custom Oracle menggunakan periode retensi cadangan yang sama dengan instans DB primer Anda. Anda dapat memodifikasi periode retensi cadangan menjadi 1–35 hari. RDS Custom mendukung pencadangan, pemulihan, dan point-in-time pemulihan (PITR). Untuk informasi selengkapnya tentang mencadangkan dan memulihkan instans DB RDS Custom, lihat Mencadangkan dan memulihkan instans DB Amazon RDS Custom for Oracle.

> **Note**
>
> Saat membuat replika Oracle, RDS Custom menghentikan sementara pembersihan file log redo. Dengan cara ini, RDS Custom dipastikan dapat menerapkan log ini ke replika Oracle baru setelah tersedia.

Promosi replika

Anda dapat mempromosikan replika Oracle terkelola di RDS Custom for Oracle menggunakan konsol, perintah AWS CLI `promote-read-replica`, atau API `PromoteReadReplica`. Jika Anda menghapus instans DB primer Anda, dan semua replika berkondisi baik, RDS Custom for Oracle akan mempromosikan replika terkelola Anda ke instans mandiri secara otomatis. Jika replika telah dijeda otomatisasinya atau berada di luar perimeter dukungan, Anda harus memperbaiki

replika ini sebelum RDS Custom dapat mempromosikannya secara otomatis. Anda hanya dapat mempromosikan replika Oracle eksternal secara manual.

Pedoman dan batasan untuk replikasi RDS Custom for Oracle

Saat Anda membuat replika RDS Custom for Oracle, tidak semua opsi replika RDS Oracle didukung.

Topik

- Pedoman umum untuk replikasi RDS Custom for Oracle
- Batasan umum untuk replikasi RDS Custom for Oracle
- Persyaratan dan batasan jaringan untuk replikasi RDS Custom for Oracle
- Batasan replika eksternal untuk RDS Custom for Oracle
- Batasan promosi replika untuk RDS Custom for Oracle
- Pedoman promosi replika untuk RDS Custom for Oracle

Pedoman umum untuk replikasi RDS Custom for Oracle

Saat menggunakan RDS Custom for Oracle, ikuti pedoman ini:

- Jangan memodifikasi pengguna `RDS_DATAGUARD`. Pengguna ini dialokasikan untuk otomatisasi RDS Custom for Oracle. Memodifikasi pengguna ini dapat memberikan hasil yang tidak diinginkan, seperti ketidakmampuan untuk membuat replika Oracle untuk instans DB RDS Custom for Oracle Anda.
- Jangan mengubah kata sandi pengguna replikasi. kata sandi ini diperlukan untuk mengelola konfigurasi Oracle Data Guard pada host RDS Custom. Jika Anda mengubah kata sandi ini, RDS Custom for Oracle mungkin menempatkan replika Oracle Anda di luar perimeter dukungan. Untuk informasi selengkapnya, lihat Perimeter dukungan RDS Custom.

  Kata sandi disimpan di AWS Secrets Manager, yang diberi tag dengan ID sumber daya DB. Setiap replika Oracle memiliki rahasia tersendiri di Secrets Manager. Format rahasianya adalah sebagai berikut.

  ```
  do-not-delete-rds-custom-db-DB_resource_id-6-digit_UUID-dg
  ```

- Jangan mengubah `DB_UNIQUE_NAME` untuk instans DB primer. Mengubah nama ini akan menyebabkan operasi pemulihan menjadi macet.

- Jangan menentukan klausa `STANDBYS=NONE` dalam perintah `CREATE PLUGGABLE DATABASE` di CDB RDS Custom. Dengan demikian, jika failover terjadi, CDB siaga Anda akan berisi semua PDB.

Batasan umum untuk replikasi RDS Custom for Oracle

RDS Custom for Oracle memiliki batasan berikut:

- Anda tidak dapat membuat replika RDS Custom for Oracle dalam mode hanya baca. Namun, Anda dapat secara manual mengubah mode replika terpasang menjadi hanya-baca, dan dari hanya baca ke terpasang. Untuk informasi selengkapnya, lihat dokumentasi untuk perintah create-db-instance-read-replicaAWS CLI.
- Anda tidak dapat membuat replika RDS Custom for Oracle lintas Wilayah.
- Anda tidak dapat mengubah nilai parameter `CommunicationTimeout` Oracle Data Guard. Parameter ini diatur ke 15 detik untuk instans DB RDS Custom for Oracle.

Persyaratan dan batasan jaringan untuk replikasi RDS Custom for Oracle

Pastikan konfigurasi jaringan Anda mendukung replika RDS Custom for Oracle. Pertimbangkan hal berikut:

- Pastikan untuk mengaktifkan port 1140 untuk komunikasi masuk dan keluar dalam cloud privat virtual (VPC) Anda untuk instans DB primer dan semua replika. Hal ini diperlukan untuk komunikasi Oracle Data Guard antar-replika baca.
- RDS Custom for Oracle memvalidasi jaringan saat membuat replika Oracle. Jika instans DB primer dan replika baru tidak dapat terhubung melalui jaringan, RDS Custom for Oracle tidak membuat replika dan menetapkannya ke status `INCOMPATIBLE_NETWORK`.
- Untuk replika Oracle eksternal, seperti yang Anda buat di Amazon EC2 atau on-premise, gunakan port dan pendengar lain untuk replikasi Oracle Data Guard. Mencoba menggunakan port 1140 dapat menyebabkan konflik dengan otomatisasi RDS Custom.
- File `/rdsdbdata/config/tnsnames.ora` berisi nama layanan jaringan yang dipetakan ke alamat protokol pendengar. Perhatikan persyaratan dan rekomendasi berikut:
  - Entri di `tnsnames.ora` yang diawali dengan `rds_custom_` dialokasikan untuk RDS Custom saat menangani operasi replika Oracle.

    Saat membuat entri manual di `tnsnames.ora`, jangan gunakan awalan ini.

- Dalam beberapa kasus, Anda sebaiknya menjalankan peralihan atau failover secara manual, atau menggunakan teknologi failover seperti Fast-Start Failover (FSFO). Jika demikian, pastikan untuk menyinkronkan entri `tnsnames.ora` secara manual dari instans DB primer ke semua instans siaga. Rekomendasi ini berlaku untuk replika Oracle yang dikelola oleh RDS Custom dan replika Oracle eksternal.

  Otomatisasi kustom RDS memperbarui entri `tnsnames.ora` hanya pada instans DB primer. Pastikan juga untuk melakukan sinkronisasi saat Anda menambahkan atau menghapus replika Oracle.

  Jika Anda tidak menyinkronkan file `tnsnames.ora` dan menjalankan peralihan atau failover secara manual, Oracle Data Guard pada instans DB primer mungkin tidak dapat berkomunikasi dengan replika Oracle.

Batasan replika eksternal untuk RDS Custom for Oracle

Replika eksternal RDS Custom for Oracle, yang menyertakan replika on-premise, memiliki batasan berikut:

- RDS Custom for Oracle tidak mendeteksi perubahan peran instans pada failover manual, seperti FSFO, untuk replika Oracle eksternal.

  RDS Custom for Oracle mendeteksi perubahan untuk replika terkelola. Perubahan peran dicatat dalam log peristiwa. Anda juga dapat melihat status baru dengan menggunakan describe-db-instancesAWS CLIperintah.
- RDS Custom for Oracle tidak mendeteksi lag replikasi tinggi untuk replika Oracle eksternal.

  RDS Custom for Oracle mendeteksi lag untuk replika terkelola. Lag replikasi tinggi menghasilkan peristiwa `Replication has stopped`. Anda juga dapat melihat status replikasi dengan menggunakan describe-db-instancesAWS CLIperintah, tetapi mungkin ada penundaan untuk diperbarui.
- RDS Custom for Oracle tidak mempromosikan replika Oracle eksternal secara otomatis jika Anda menghapus instans DB primer Anda.

  Fitur promosi otomatis hanya tersedia untuk replika Oracle terkelola. Untuk informasi tentang mempromosikan replika Oracle secara manual, lihat laporan resmi Memungkinkan ketersediaan tinggi dengan Data Guard di Amazon RDS Custom for Oracle.

Batasan promosi replika untuk RDS Custom for Oracle

Mempromosikan replika Oracle yang dikelola RDS Custom for Oracle sama dengan mempromosikan replika yang dikelola RDS, dengan beberapa perbedaan. Perhatikan batasan berikut untuk replika RDS Custom for Oracle:

- Anda tidak dapat mempromosikan replika saat RDS Custom for Oracle mencadangkannya.
- Anda tidak dapat mengubah periode retensi cadangan menjadi `0` saat Anda mempromosikan replika Oracle Anda.
- Anda tidak dapat mempromosikan replika Anda ketika tidak dalam status kondisi baik.

  Jika Anda mengeluarkan `delete-db-instance` pada instans DB primer, RDS Custom for Oracle memvalidasi bahwa setiap replika Oracle terkelola berkondisi baik dan tersedia untuk promosi. Replika mungkin tidak memenuhi syarat untuk promosi karena otomatisasinya dijeda atau berada di luar perimeter dukungan. Dalam kasus seperti itu, RDS Custom for Oracle menerbitkan peristiwa yang menjelaskan masalahnya sehingga Anda dapat memperbaiki replika Oracle Anda secara manual.

Pedoman promosi replika untuk RDS Custom for Oracle

Saat mempromosikan replika, perhatikan pedoman berikut:

- Jangan memulai failover saat RDS Custom for Oracle mempromosikan replika Anda. Jika tidak, alur kerja promosi bisa macet.
- Jangan beralih ke instans DB primer Anda saat RDS Custom for Oracle mempromosikan replika Oracle Anda. Jika tidak, alur kerja promosi bisa macet.
- Jangan menonaktifkan instans DB primer Anda saat RDS Custom for Oracle mempromosikan replika Oracle Anda. Jika tidak, alur kerja promosi bisa macet.
- Jangan mencoba memulai ulang replikasi dengan instans DB Anda yang baru dipromosikan sebagai target. Setelah RDS Custom for Oracle mempromosikan replika Oracle Anda, replika ini menjadi instans DB mandiri dan tidak lagi memiliki peran replika.

Untuk informasi selengkapnya, lihat Memecahkan masalah promosi replika untuk RDS Custom for Oracle.

Mempromosikan replika RDS Custom for Oracle ke instans DB mandiri

Sama seperti RDS for Oracle, Anda dapat mempromosikan replika RDS Custom for Oracle ke instans DB mandiri. Saat Anda mempromosikan replika Oracle, RDS Custom for Oracle akan mem-boot ulang instans DB sebelum replika Oracle tersedia. Untuk informasi selengkapnya tentang mempromosikan replika Oracle, lihat Mempromosikan replika baca menjadi instans DB mandiri.

Langkah-langkah berikut ini menunjukkan proses umum untuk mempromosikan replika Oracle ke instans DB:

1. Hentikan transaksi apa pun agar tidak ditulis ke instans DB primer.
2. Tunggu RDS Custom for Oracle untuk menerapkan semua pembaruan ke replika Oracle Anda.
3. Promosikan replika Oracle Anda dengan memilih opsi Promosikan di konsol Amazon RDS, perintah AWS CLI `promote-read-replica`, atau operasi API Amazon RDS `PromoteReadReplica`.

Promosi replika Oracle membutuhkan waktu beberapa menit. Selama prosesnya, RDS Custom for Oracle menghentikan replikasi dan mem-boot ulang replika Anda. Saat boot ulang selesai, replika Oracle tersedia sebagai instans DB mandiri.

Konsol

Untuk mempromosikan replika RDS Custom for Oracle ke instans DB mandiri

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di konsol Amazon RDS, pilih Database.

   Panel Basis Data muncul. Setiap replika Oracle menampilkan Replika di kolom Peran.
3. Pilih replika RDS Custom for Oracle yang ingin Anda promosikan.
4. Untuk Tindakan, pilih Promosikan.
5. Di halaman Promosikan replika Oracle, masukkan periode retensi cadangan dan periode pencadangan untuk instans DB yang baru dipromosikan. Anda tidak dapat mengatur nilai ini ke 0.
6. Saat pengaturan sudah sesuai dengan keinginan Anda, pilih Promosikan replika Oracle.

AWS CLI

Untuk mempromosikan replika RDS Custom for Oracle Anda ke instans DB mandiri, gunakan perintah AWS CLI `promote-read-replica`.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds promote-read-replica \
--db-instance-identifier my-custom-read-replica \
--backup-retention-period 2 \
--preferred-backup-window 23:00-24:00
```

Untuk Windows:

```
aws rds promote-read-replica ^
--db-instance-identifier my-custom-read-replica ^
--backup-retention-period 2 ^
--preferred-backup-window 23:00-24:00
```

API RDS

Untuk mempromosikan replika RDS Custom for Oracle Anda menjadi instans DB mandiri, panggil operasi API Amazon RDS `PromoteReadReplica` dengan parameter wajib `DBInstanceIdentifier`.

Mencadangkan dan memulihkan instans DB Amazon RDS Custom for Oracle

Seperti Amazon RDS, RDS Custom membuat dan menyimpan cadangan otomatis instans DB RDS Custom Anda selama jendela pencadangan instans DB Anda. Anda juga dapat mencadangkan instans DB secara manual.

Prosedurnya sama dengan mengambil snapshot instans DB Amazon RDS. Snapshot pertama instans DB RDS Custom berisi data untuk instans DB penuh. Snapshot berikutnya bersifat inkremental.

Kembalikan snapshot DB menggunakan file AWS Management Console atau file. AWS CLI

Topik

Membuat snapshot RDS Custom for Oracle

RDS Custom for Oracle membuat snapshot volume penyimpanan instans DB Anda, mencadangkan seluruh instans DB dan bukan hanya basis data individual. Ketika instans DB Anda berisi basis data kontainer (CDB), snapshot instans menyertakan CDB root dan semua PDB.

Saat Anda membuat snapshot RDS Custom for Oracle, tentukan instans DB RDS Custom mana yang akan dicadangkan. Beri nama snapshot sehingga Anda dapat melakukan proses pemulihan dari snapshot tersebut nanti.

Saat Anda membuat snapshot, RDS Custom for Oracle membuat snapshot Amazon EBS untuk setiap volume yang dilampirkan ke instans DB. RDS Custom for Oracle menggunakan snapshot EBS volume root untuk mendaftarkan Amazon Machine Image (AMI) baru. Agar mudah dikaitkan dengan instans DB tertentu, snapshot ditandai dengan `DBSnapshotIdentifier`, `DbiResourceId`, dan `VolumeType`.

Membuat snapshot DB menghasilkan suspensi I/O singkat. Suspensi ini dapat bertahan beberapa detik hingga beberapa menit, bergantung pada ukuran dan kelas instans DB Anda. Waktu

pembuatan snapshot bervariasi sesuai dengan ukuran basis data Anda. Karena snapshot mencakup seluruh volume penyimpanan, ukuran file seperti file sementara juga memengaruhi waktu pembuatan snapshot. Untuk mempelajari selengkapnya tentang membuat snapshot, lihat Membuat snapshot DB untuk instans DB Single-AZ.

Buat snapshot RDS Custom for Oracle menggunakan konsol atau AWS CLI.

Konsol

Cara membuat snapshot RDS Custom

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data.
3. Dalam daftar instans DB RDS Custom, pilih instans yang ingin Anda ambil snapshot-nya.
4. Untuk Tindakan, pilih Ambil snapshot.

   Jendela Ambil snapshot DB akan muncul.
5. Untuk Nama snapshot, masukkan nama snapshot.
6. Pilih Ambil snapshot.

AWS CLI

Anda membuat snapshot dari instans RDS Custom DB dengan menggunakan perintah. create-db-snapshot AWS CLI

Tentukan opsi berikut:

- `--db-instance-identifier` — Mengidentifikasi instans DB RDS Custom mana yang akan Anda cadangkan
- `--db-snapshot-identifier` — Memberi nama snapshot RDS Custom sehingga Anda dapat melakukan proses pemulihan dari snapshot tersebut nanti

Dalam contoh ini, Anda membuat snapshot DB bernama *`my-custom-snapshot`* untuk instans DB RDS Custom bernama *`my-custom-instance`*.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-snapshot \
    --db-instance-identifier my-custom-instance \
    --db-snapshot-identifier my-custom-snapshot
```

Untuk Windows:

```
aws rds create-db-snapshot ^
    --db-instance-identifier my-custom-instance ^
    --db-snapshot-identifier my-custom-snapshot
```

Memulihkan dari snapshot DB RDS Custom for Oracle

Saat memulihkan instans DB RDS Custom for Oracle, Anda memberikan nama untuk snapshot DB dan instans baru. Anda tidak dapat memulihkan dari snapshot ke instans DB RDS Custom yang ada. Instans DB RDS Custom for Oracle baru dibuat saat Anda melakukan pemulihan.

Proses pemulihan berbeda dari pemulihan di Amazon RDS dalam hal berikut:

- Sebelum memulihkan snapshot, RDS Custom for Oracle mencadangkan file konfigurasi yang ada. File tersebut tersedia di instans yang dipulihkan di direktori `/rdsdbdata/config/backup`. RDS Custom for Oracle memulihkan snapshot DB dengan parameter default dan menimpa file konfigurasi basis data sebelumnya dengan yang sudah ada. Dengan demikian, instans yang dipulihkan tidak mempertahankan parameter dan perubahan kustom pada file konfigurasi basis data.
- Basis data yang dipulihkan memiliki nama yang sama seperti di snapshot. Anda tidak dapat menentukan nama yang berbeda. (Untuk RDS Custom for Oracle, default-nya adalah `ORCL`.)

Konsol

Cara memulihkan instans DB RDS Custom dari snapshot DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Snapshot.
3. Pilih snapshot DB yang ingin dipulihkan.
4. Untuk Tindakan, pilih Pulihkan snapshot.
5. Di halaman Pulihkan instans DB, untuk Pengidentifikasi instans DB, masukkan nama instans DB RDS Custom Anda yang dipulihkan.

6. Pilih Pulihkan instans DB.

AWS CLI

Anda mengembalikan snapshot RDS Custom DB dengan menggunakan perintah restore-db-instance-from AWS CLI -db-snapshot.

Jika snapshot yang Anda pulihkan adalah untuk instans DB privat, pastikan untuk menentukan `db-subnet-group-name` dan `no-publicly-accessible` yang benar. Jika tidak, default instans DB diatur agar dapat diakses publik. Opsi berikut diperlukan:

- `db-snapshot-identifier` — Mengidentifikasi snapshot yang akan dipulihkan
- `db-instance-identifier` — Menentukan nama instans DB RDS Custom yang akan dibuat dari snapshot DB
- `custom-iam-instance-profile` — Menentukan profil instans yang terkait dengan instans Amazon EC2 yang mendasari instans DB RDS Custom.

Kode berikut memulihkan snapshot bernama `my-custom-snapshot` untuk `my-custom-instance`.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds restore-db-instance-from-db-snapshot \
  --db-snapshot-identifier my-custom-snapshot \
  --db-instance-identifier my-custom-instance \
  --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance \
  --no-publicly-accessible
```

Untuk Windows:

```
aws rds restore-db-instance-from-db-snapshot ^
  --db-snapshot-identifier my-custom-snapshot ^
  --db-instance-identifier my-custom-instance ^
  --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance ^
  --no-publicly-accessible
```

Memulihkan instans RDS Custom for Oracle ke suatu titik waktu

Anda dapat memulihkan instans DB ke titik waktu tertentu (PITR) dan membuat instans DB baru. Untuk mendukung PITR, instans DB Anda harus mengatur retensi cadangan ke nilai bukan nol.

Waktu pemulihan terbaru untuk instans DB RDS Custom for Oracle bergantung pada beberapa faktor, tetapi biasanya dalam 5 menit dari waktu saat ini. Untuk melihat waktu restorable terbaru untuk instans DB, gunakan AWS CLI describe-db-instancesperintah dan lihat nilai yang dikembalikan di `LatestRestorableTime` bidang untuk instans DB. Untuk melihat waktu pemulihan terbaru setiap instans DB di konsol Amazon RDS, pilih Cadangan otomatis.

Anda dapat memulihkan ke titik waktu mana pun dalam periode retensi cadangan Anda. Untuk melihat waktu pemulihan terbaru setiap instans DB, pilih Cadangan otomatis di konsol Amazon RDS.

Untuk informasi umum tentang PITR, lihat Memulihkan instans DB dengan waktu yang ditentukan.

Topik

- Pertimbangan PITR untuk RDS Custom for Oracle

Pertimbangan PITR untuk RDS Custom for Oracle

PITR di RDS Custom for Oracle berbeda dari PITR di Amazon RDS dalam beberapa hal penting berikut:

- Basis data yang dipulihkan memiliki nama yang sama seperti pada instans DB sumber. Anda tidak dapat menentukan nama yang berbeda. Default-nya adalah `ORCL`.
- `AWSRDSCustomIamRolePolicy` membutuhkan izin baru. Untuk informasi selengkapnya, lihat Langkah 2: Tambahkan kebijakan akses ke AWSRDSCustomInstanceRoleForRdsCustomInstance.
- Semua instans DB RDS Custom for Oracle harus mengatur retensi cadangan ke nilai bukan nol.
- Jika Anda mengubah zona waktu sistem operasi atau instans DB, PITR mungkin tidak berfungsi. Untuk informasi selengkapnya tentang perubahan zona waktu, lihat Zona waktu Oracle.
- Jika Anda menyetel otomatisasi ke`ALL_PAUSED`, RDS Custom menjeda unggahan file log redo yang diarsipkan, termasuk log yang dibuat sebelum waktu restorable (LRT) terbaru. Sebaiknya Anda menjeda otomatisasi untuk jangka waktu yang singkat.

  Sebagai ilustrasi, asumsikan bahwa LRT Anda 10 menit yang lalu. Anda menjeda otomatisasi. Selama jeda, RDS Custom tidak mengunggah log pengulangan yang diarsipkan. Jika instans DB crash, Anda hanya dapat memulihkan ke waktu sebelum LRT yang ada saat Anda menjeda. Saat

Anda melanjutkan otomatisasi, RDS Custom melanjutkan pengunggahan log. LRT berlanjut. Aturan PITR normal berlaku.

- Di RDS Custom, Anda dapat menentukan secara manual jumlah jam arbitrer untuk mempertahankan log pengulangan yang diarsipkan sebelum RDS Custom menghapusnya setelah diunggah. Tentukan jumlah jam sebagai berikut:

  1. Buat file teks bernama `/opt/aws/rdscustomagent/config/redo_logs_custom_configuration.json`.
  2. Tambahkan objek JSON dalam format berikut: `{"archivedLogRetentionHours" : "num_of_hours"}`. Angka tersebut harus berupa bilangan bulat dalam kisaran 1-840.

- Asumsikan bahwa Anda menghubungkan non-CDB ke basis data kontainer (CDB) sebagai PDB dan kemudian mencoba PITR. Operasi hanya akan berhasil jika sebelumnya Anda mencadangkan PDB. Setelah membuat atau memodifikasi PDB, sebaiknya Anda selalu mencadangkan PDB.
- Sebaiknya Anda tidak menyesuaikan parameter inisialisasi basis data. Misalnya, memodifikasi parameter berikut memengaruhi PITR:
  - `CONTROL_FILE_RECORD_KEEP_TIME` memengaruhi aturan untuk mengunggah dan menghapus log.
  - `LOG_ARCHIVE_DEST_n` tidak mendukung banyak tujuan.
  - `ARCHIVE_LAG_TARGET`mempengaruhi waktu restorable terbaru. `ARCHIVE_LAG_TARGET`diatur ke `300` karena tujuan titik pemulihan (RPO) adalah 5 menit. Untuk menghormati tujuan ini, RDS mengganti log pengulangan online setiap 5 menit dan menyimpannya dalam ember Amazon S3. Jika frekuensi sakelar log menyebabkan masalah kinerja untuk database RDS Custom for Oracle, Anda dapat menskalakan instans dan penyimpanan DB Anda ke yang memiliki IOPS dan throughput yang lebih tinggi. Jika perlu untuk rencana pemulihan Anda, Anda dapat menyesuaikan pengaturan parameter `ARCHIVE_LAG_TARGET` inisialisasi ke nilai dari 60-7200.
- Jika Anda menyesuaikan parameter inisialisasi database, kami sangat menyarankan Anda menyesuaikan hanya yang berikut ini:
  - `COMPATIBLE`
  - `MAX_STRING_SIZE`
  - `DB_FILES`
  - `UNDO_TABLESPACE`
  - `ENABLE_PLUGGABLE_DATABASE`
  - `CONTROL_FILES`
  - `AUDIT_TRAIL`

- `AUDIT_TRAIL_DEST`

Untuk semua parameter inisialisasi lainnya, RDS Custom memulihkan nilai default. Jika Anda memodifikasi parameter yang tidak ada dalam daftar sebelumnya, mungkin ada efek buruk pada PITR dan menyebabkan hasil yang tidak terduga. Misalnya, `CONTROL_FILE_RECORD_KEEP_TIME` memengaruhi aturan untuk mengunggah dan menghapus log.

Anda dapat memulihkan instans RDS Custom DB ke titik waktu menggunakan AWS Management Console, API AWS CLI, atau RDS.

Konsol

Cara memulihkan instans DB RDS Custom ke waktu tertentu

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Pencadangan otomatis.
3. Pilih instans DB RDS Custom yang ingin Anda pulihkan.
4. Untuk Tindakan, pilih Pulihkan ke titik waktu.

   Jendela Pulihkan ke titik waktu akan muncul.
5. Pilih Waktu pemulihan terbaru untuk memulihkan ke waktu terbaru yang dimungkinkan atau pilih Kustom untuk memilih waktu.

   Jika Anda memilih Kustom, masukkan tanggal dan waktu untuk memulihkan instans.

   Waktu ditampilkan dalam zona waktu lokal Anda, yang ditunjukkan dengan offset dari Waktu Universal Terkoordinasi (UTC). Misalnya, UTC-5 adalah Waktu Standar Timur/Waktu Musim Panas Tengah.
6. Untuk Pengidentifikasi instans DB, masukkan nama target instans DB RDS Custom yang dipulihkan. Nama harus unik.
7. Pilih opsi lain sesuai kebutuhan, seperti kelas instans DB.
8. Pilih Pulihkan ke titik waktu.

AWS CLI

Anda mengembalikan instans DB ke waktu tertentu dengan menggunakan point-in-time AWS CLI perintah restore-db-instance-to- untuk membuat instance RDS Custom DB baru.

Gunakan salah satu opsi berikut untuk menentukan cadangan yang akan dipulihkan dari:

- `--source-db-instance-identifier` *`mysourcedbinstance`*
- `--source-dbi-resource-id` *`dbinstanceresourceID`*
- `--source-db-instance-automated-backups-arn` *`backupARN`*

Opsi `custom-iam-instance-profile` diperlukan.

Contoh berikut memulihkan `my-custom-db-instance` ke instans DB baru bernama `my-restored-custom-db-instance` pada waktu yang ditentukan.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds restore-db-instance-to-point-in-time \
    --source-db-instance-identifier my-custom-db-instance\
    --target-db-instance-identifier my-restored-custom-db-instance \
    --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance \
    --restore-time 2022-10-14T23:45:00.000Z
```

Untuk Windows:

```
aws rds restore-db-instance-to-point-in-time ^
    --source-db-instance-identifier my-custom-db-instance ^
    --target-db-instance-identifier my-restored-custom-db-instance ^
    --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance ^
    --restore-time 2022-10-14T23:45:00.000Z
```

Menghapus snapshot RDS Custom for Oracle

Anda dapat menghapus snapshot DB yang dikelola RDS Custom for Oracle saat tidak lagi membutuhkannya. Prosedur penghapusan sama untuk instans DB Amazon RDS dan RDS Custom.

Snapshot Amazon EBS untuk biner dan volume root tetap ada di akun Anda untuk waktu yang lebih lama karena mungkin ditautkan ke beberapa instans yang berjalan di akun Anda atau ke snapshot

RDS Custom for Oracle lainnya. Snapshot EBS ini dihapus secara otomatis setelah tidak lagi terkait dengan sumber daya RDS Custom for Oracle yang ada (instans DB atau cadangan).

Konsol

Cara menghapus snapshot instans DB RDS Custom

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Snapshot.
3. Pilih snapshot DB yang ingin Anda hapus.
4. Untuk Tindakan, pilih Hapus snapshot.
5. Pilih Hapus di halaman konfirmasi.

AWS CLI

Untuk menghapus snapshot RDS Custom, gunakan perintah. AWS CLI delete-db-snapshot

Opsi berikut diperlukan:

- `--db-snapshot-identifier` — Snapshot yang akan dihapus

Contoh berikut menghapus snapshot DB `my-custom-snapshot`.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds delete-db-snapshot \
  --db-snapshot-identifier my-custom-snapshot
```

Untuk Windows:

```
aws rds delete-db-snapshot ^
  --db-snapshot-identifier my-custom-snapshot
```

Menghapus cadangan otomatis RDS Custom for Oracle

Anda dapat menghapus cadangan otomatis yang disimpan untuk RDS Custom for Oracle saat tidak diperlukan lagi. Prosedurnya sama dengan prosedur untuk menghapus cadangan Amazon RDS.

Konsol

Untuk menghapus cadangan otomatis yang dipertahankan

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Pencadangan otomatis.
3. Pilih Dipertahankan.
4. Pilih cadangan otomatis yang dipertahankan yang ingin Anda hapus.
5. Untuk Tindakan, pilih Hapus.
6. Di halaman konfirmasi, masukkan **`delete me`** dan pilih Hapus.

AWS CLI

Anda dapat menghapus cadangan otomatis yang dipertahankan dengan menggunakan AWS CLI perintah delete-db-instance-automated-backup.

Opsi berikut digunakan untuk menghapus cadangan otomatis yang dipertahankan:

- `--dbi-resource-id` — Pengidentifikasi sumber daya untuk instans DB RDS Custom sumber.

  Anda dapat menemukan pengenal sumber daya untuk instance DB sumber dari cadangan otomatis yang dipertahankan dengan menggunakan AWS CLI perintah describe-db-instance-automated -backup.

Contoh berikut menghapus cadangan otomatis yang dipertahankan dengan pengidentifikasi sumber daya instans DB sumber `custom-db-123ABCEXAMPLE`.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds delete-db-instance-automated-backup \
```

```
    --dbi-resource-id custom-db-123ABCEXAMPLE
```

Untuk Windows:

```
aws rds delete-db-instance-automated-backup ^
    --dbi-resource-id custom-db-123ABCEXAMPLE
```

Menggunakan grup opsi di RDS Custom for Oracle

RDS Custom menggunakan grup opsi untuk mengaktifkan dan mengonfigurasi fitur tambahan. Grup opsi menentukan fitur yang disebut opsi, yang tersedia untuk instans DB RDS Custom for Oracle. Opsi dapat memiliki pengaturan yang menentukan cara kerja opsi. Saat Anda mengaitkan instans DB RDS Custom for Oracle dengan grup opsi, opsi yang ditentukan dan pengaturan opsi diaktifkan untuk instans ini. Untuk informasi umum tentang grup opsi di Amazon RDS, lihat Menggunakan grup opsi.

Topik

Gambaran umum grup opsi di RDS Custom for Oracle

Untuk mengaktifkan opsi ini untuk basis data Oracle, Anda dapat menambahkannya ke grup opsi, lalu mengaitkan grup opsi dengan instans DB Anda. Untuk informasi selengkapnya, lihat Menggunakan grup opsi.

Topik

Ringkasan opsi RDS Custom for Oracle

RDS Custom for Oracle mendukung opsi berikut untuk instans DB.

Opsi	ID Opsi	Deskripsi
Zona waktu Oracle	`Timezone`	Zona waktu yang digunakan oleh instans DB RDS Custom for Oracle Anda.

Langkah-langkah dasar untuk menambahkan opsi ke instans DB RDS Custom for Oracle

Prosedur umum untuk menambahkan opsi ke instans DB RDS Custom for Oracle Anda adalah sebagai berikut:

1. Buat grup opsi baru, atau salin atau ubah grup opsi yang ada.
2. Tambahkan opsi untuk grup opsi.
3. Kaitkan grup opsi dengan instans DB Anda saat membuat atau memodifikasinya.

Membuat grup opsi untuk di RDS Custom for Oracle

Anda dapat membuat grup opsi baru yang pengaturannya berasal dari grup opsi default. Anda kemudian perlu menambahkan satu atau beberapa opsi ke grup opsi baru. Sebagai alternatif, jika sudah memiliki grup opsi, Anda dapat menyalin grup opsi tersebut dengan semua opsinya ke grup opsi baru. Untuk mempelajari cara menyalin grup opsi, lihat Menyalin grup opsi.

Grup opsi default untuk RDS Custom for Oracle adalah `default:custom-oracle-ee` dan `default:custom-oracle-ee-cdb`. Ketika membuat grup opsi baru, pengaturannya diambil dari grup opsi default. Setelah menambahkan opsi `TIME_ZONE`, kemudian Anda dapat mengaitkan grup opsi dengan instans DB Anda.

Konsol

Salah satu cara untuk membuat grup opsi adalah dengan menggunakan AWS Management Console.

Untuk membuat grup opsi baru menggunakan konsol

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup opsi.
3. Pilih Buat grup.
4. Di jendela Buat grup opsi, lakukan hal berikut:
   a. Untuk Nama, ketikkan nama untuk grup opsi yang unik dalam akun AWS Anda. Nama tersebut hanya boleh berisi huruf, angka, dan tanda hubung.
   b. Untuk Deskripsi, ketikkan deskripsi singkat grup opsi. Deskripsi digunakan untuk tampilan.
   c. Untuk Mesin, pilih salah satu dari mesin DB RDS Custom for Oracle berikut:
      - custom-oracle-ee

- custom-oracle-ee-cdb

d. Untuk Versi mesin utama, pilih versi mesin utama yang didukung oleh RDS Custom for Oracle. Untuk informasi selengkapnya, lihat RDS Custom for Oracle.

5. Untuk melanjutkan, pilih Buat. Untuk membatalkan operasi, pilih Batal.

AWS CLI

Untuk membuat grup opsi, gunakan perintah AWS CLI `create-option-group` dengan parameter wajib berikut.

- `--option-group-name`
- `--engine-name`
- `--major-engine-version`
- `--option-group-description`

Example

Contoh berikut membuat grup opsi bernama `testoptiongroup`, yang terkait dengan mesin DB Oracle Enterprise Edition. Deskripsi diapit dalam tanda petik.

Untuk Linux, macOS, atau Unix:

```
aws rds create-option-group \
    --option-group-name testoptiongroup \
    --engine-name custom-oracle-ee-cdb \
    --major-engine-version 19 \
    --option-group-description "Test option group for a Custom Oracle CDB"
```

Untuk Windows:

```
aws rds create-option-group ^
    --option-group-name testoptiongroup ^
    --engine-name custom-oracle-ee-cdb ^
    --major-engine-version 19 ^
    --option-group-description "Test option group for a Custom Oracle CDB"
```

API RDS

Untuk membuat grup opsi, panggil operasi `CreateOptionGroup` API Amazon RDS.

Mengaitkan grup opsi dengan instans DB RDS Custom for Oracle

Anda dapat mengaitkan grup opsi Anda dengan instans DB baru atau yang sudah ada:

- Untuk instans DB baru, terapkan grup opsi saat Anda membuat instans. Untuk informasi selengkapnya, lihat Membuat instans DB RDS Custom for Oracle.
- Untuk instans DB yang sudah ada, terapkan grup opsi dengan memodifikasi instans dan menambahkan grup opsi baru. Untuk informasi selengkapnya, lihat Memodifikasi instans DB RDS Custom for Oracle.

Zona waktu Oracle

Untuk mengubah zona waktu sistem yang digunakan oleh instans DB RDS Custom for Oracle, gunakan opsi zona waktu. Misalnya, Anda dapat mengubah zona waktu instans DB agar kompatibel dengan lingkungan on-premise atau aplikasi lama. Opsi zona waktu mengubah zona waktu di tingkat host. Mengubah zona waktu memengaruhi semua kolom dan nilai tanggal, termasuk `SYSDATE` dan `SYSTIMESTAMP`.

Topik

- Pengaturan opsi zona waktu di RDS Custom for Oracle
- Zona waktu yang tersedia di RDS Custom for Oracle
- Pertimbangan untuk mengatur zona waktu di RDS Custom for Oracle
- Batasan untuk mengatur zona waktu di RDS Custom for Oracle
- Menambahkan opsi zona waktu ke grup opsi
- Menghapus opsi zona waktu

Pengaturan opsi zona waktu di RDS Custom for Oracle

Amazon RDS mendukung pengaturan berikut untuk opsi zona waktu.

Pengaturan opsi	Nilai valid	Deskripsi
`TIME_ZONE`	Salah satu zona waktu yang tersedia. Untuk daftar lengkapnya, lihat Zona waktu	Zona waktu baru untuk instans DB Anda.

Pengaturan opsi	Nilai valid	Deskripsi
	yang tersedia di RDS Custom for Oracle.	

Zona waktu yang tersedia di RDS Custom for Oracle

Anda dapat menggunakan nilai berikut untuk opsi zona waktu.

Zona	Zona waktu
Afrika	Afrika/Kairo, Afrika/Casablanca, Afrika/Harare, Afrika/Lagos, Afrika/Luanda, Afrika/Monrovia, Afrika/Nairobi, Afrika/Tripoli, Afrika/Windhoek
Amerika	Amerika/Araguaina, Amerika/Argentina/Buenos_Aires, Amerika/Asuncion, Amerika/Bogota, Amerika/Caracas, Amerika/Chicago, Amerika/Chihuahua, Amerika/Cuiaba, Amerika/Denver, Amerika/Detroit, Amerika/Fortaleza, Amerika/Godthab, Amerika/Guatemala, Amerika/Halifax, Amerika/Lima, Amerika/Los_Angeles, Amerika/Manaus, Amerika/Matamoros, Amerika/Mexico_City, Amerika/Monterrey, Amerika/Montevideo, Amerika/New_York, Amerika/Phoenix, Amerika/Santiago, Amerika/Sao_Paulo, Amerika/Tijuana, Amerika/Toronto
Asia	Asia/Amman, Asia/Ashgabat, Asia/Baghdad, Asia/Baku, Asia/Bangkok, Asia/Beirut, Asia/Calcutta, Asia/Damaskus, Asia/Dhaka, Asia/Hong_Kong, Asia/Irkutsk, Asia/Jakarta, Asia/Yerusalem, Asia/Kabul, Asia/Karachi, Asia/Kathmandu, Asia/Kolkata, Asia/Krasnoyarsk, Asia/Magadan, Asia/Manila, Asia/Muscat, Asia/Novosibirsk, Asia/Rangoon, Asia/Riyadh, Asia/Seoul, Asia/Shanghai, Asia/Singapura, Asia/Taipei, Asia/Teheran, Asia/Tokyo, Asia/Ulaanbaatar, Asia/Vladivostok, Asia/Yakutsk, Asia/Yerevan
Atlantik	Atlantik/Azores, Atlantik/Cape_Verde
Australia	Australia/Adelaide, Australia/Brisbane, Australia/Darwin, Australia/Eucla, Australia/Hobart, Australia/Lord_Howe, Australia/Perth, Australia/Sydney
Brazil	Brasil/, Brasil/Timur DeNoronha

Zona	Zona waktu
Canada	Kanada/Newfoundland, Kanada/Saskatchewan
Dll	Dll/GMT-3
Eropa	Eropa/Amsterdam, Eropa/Athena, Eropa/Berlin, Eropa/Dublin, Eropa/Helsinki, Eropa/Kaliningrad, Eropa/London, Eropa/Madrid, Eropa/Moskow, Eropa/Paris, Eropa/Praha, Eropa/Roma, Eropa/Sarajevo
Pasifik	Pasifik/Apia, Pasifik/Auckland, Pasifik/Chatham, Pasifik/Fiji, Pasifik/Guam, Pasifik/Honolulu, Pasifik/Kiritimati, Pasifik/Marquesas, Pasifik/Samoa, Pasifik/Tongatapu, Pasifik/Wake
AS	AS/Alaska, AS/Tengah, AS/East-Indiana, AS/Timur, AS/Pasifik
UTC	UTC

Pertimbangan untuk mengatur zona waktu di RDS Custom for Oracle

Jika Anda memilih untuk mengatur zona waktu untuk instans DB, pertimbangkan hal berikut:

- Saat Anda menambahkan opsi zona waktu, pemadaman singkat terjadi saat instans DB Anda dimulai ulang secara otomatis.
- Jika tidak sengaja mengatur zona waktu secara tidak benar, Anda harus memulihkan instans DB ke pengaturan zona waktu sebelumnya. Untuk alasan ini, kami sangat menyarankan agar Anda menggunakan salah satu strategi berikut sebelum menambahkan opsi zona waktu ke instans Anda:
  - Jika instans DB RDS Custom for Oracle Anda menggunakan grup opsi default, ambil snapshot instans DB. Untuk informasi selengkapnya, lihat Membuat snapshot RDS Custom for Oracle.
  - Jika instans DB Anda saat ini menggunakan grup opsi nondefault, ambil snapshot instans DB Anda, lalu buat grup opsi baru dengan opsi zona waktu.
- Kami sangat menyarankan Anda membuat cadangan instans DB secara manual setelah menerapkan opsi `Timezone`.
- Kami sangat menyarankan Anda untuk menguji opsi zona waktu pada instans DB uji sebelum menambahkannya ke instans DB produksi. Menambahkan opsi zona waktu dapat menyebabkan masalah dengan tabel yang menggunakan tanggal sistem untuk menambahkan tanggal atau

waktu. Sebaiknya Anda menganalisis data dan aplikasi untuk menilai dampak dari perubahan zona waktu.

Batasan untuk mengatur zona waktu di RDS Custom for Oracle

Perhatikan batasan berikut:

- Anda tidak dapat mengubah zona waktu secara langsung di host tanpa memindahkannya ke luar perimeter dukungan. Untuk mengubah zona waktu basis data, Anda harus membuat grup opsi.
- Karena opsi zona waktu adalah opsi persisten (tetapi bukan opsi permanen), Anda tidak dapat melakukan hal berikut:
  - Menghapus opsi dari grup opsi setelah Anda menambahkan opsi.
  - Mengubah pengaturan zona waktu opsi ke zona waktu yang berbeda.
- Anda tidak dapat mengaitkan beberapa grup opsi dengan instans DB RDS Custom for Oracle Anda.
- Anda tidak dapat mengatur zona waktu untuk masing-masing PDB dalam CDB.

Menambahkan opsi zona waktu ke grup opsi

Grup opsi default untuk RDS Custom for Oracle adalah `default:custom-oracle-ee` dan `default:custom-oracle-ee-cdb`. Ketika membuat grup opsi baru, pengaturannya diambil dari grup opsi default. Untuk informasi umum tentang grup opsi di Amazon RDS, lihat Menggunakan grup opsi.

Konsol

Cara menambahkan opsi zona waktu ke grup opsi

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup opsi.
3. Pilih grup opsi yang ingin Anda ubah, lalu pilih Tambahkan opsi.
4. Di jendela Tambahkan opsi, lakukan hal berikut:
   a. Pilih Zona Waktu.
   b. Di Pengaturan opsi, pilih zona waktu.

c. Untuk mengaktifkan opsi pada semua instans DB RDS Custom for Oracle segera setelah Anda menambahkannya, untuk Terapkan langsung, pilih Ya. Jika Anda memilih Tidak (default), opsi diaktifkan untuk setiap instans DB terkait selama jendela pemeliharaan berikutnya.

d.

> **Important**
>
> Jika Anda menambahkan opsi zona waktu ke grup opsi yang sudah ada yang sudah terpasang ke satu atau lebih instans DB, pemadaman singkat terjadi saat semua instans DB secara otomatis dimulai ulang.

5. Jika pengaturan sudah sesuai keinginan Anda, pilih Tambahkan opsi.
6. Cadangkan instans DB RDS Custom for Oracle yang zona waktunya diperbarui. Untuk informasi selengkapnya, lihat Membuat snapshot RDS Custom for Oracle.

AWS CLI

Contoh berikut menggunakan perintah AWS CLI add-option-to-option-group untuk menambahkan `Timezone` opsi dan pengaturan `TIME_ZONE` opsi ke grup opsi yang disebut`testoptiongroup`. Zona waktu ditetapkan ke `America/Los_Angeles`.

Untuk Linux, macOS, atau Unix:

```
aws rds add-option-to-option-group \
    --option-group-name "testoptiongroup" \
    --options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=America/
Los_Angeles}]" \
    --apply-immediately
```

Untuk Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name "testoptiongroup" ^
    --options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=America/
Los_Angeles}]" ^
    --apply-immediately
```

Menghapus opsi zona waktu

Opsi zona waktu adalah opsi persisten, tetapi bukan opsi permanen. Anda tidak dapat menghapus opsi dari grup opsi setelah menambahkannya. Untuk memisahkan grup opsi lama dari instans DB Anda:

1. Buat grup opsi baru dengan opsi `Timezone` yang diperbarui.
2. Kaitkan grup opsi dengan instans DB Anda saat membuat atau memodifikasi instans.

Memigrasikan basis data on-premise ke RDS Custom for Oracle

Sebelum memigrasikan basis data Oracle on-premise ke RDS Custom for Oracle, Anda perlu mempertimbangkan faktor-faktor berikut:

- Jumlah waktu henti yang mampu ditanggung aplikasi
- Ukuran basis data sumber
- Konektivitas jaringan
- Persyaratan untuk rencana fallback
- Sumber dan target versi basis data Oracle dan jenis OS instans DB
- Alat replikasi yang tersedia, seperti AWS Database Migration Service, Oracle GoldenGate, atau alat replikasi pihak ketiga

Berdasarkan faktor-faktor tersebut, Anda dapat memilih migrasi fisik, migrasi logis, atau kombinasi. Jika memilih migrasi fisik, Anda dapat menggunakan teknik berikut:

Duplikasi RMAN

Duplikasi basis data aktif tidak memerlukan cadangan basis data sumber Anda. Tindakan ini menduplikasi basis data sumber langsung ke host tujuan dengan menyalin file basis data melalui jaringan ke instans tambahan. Perintah `DUPLICATE` RMAN menyalin file yang diperlukan sebagai salinan gambar atau set cadangan. Untuk mempelajari teknik ini, lihat postingan blog AWS Physical migration of Oracle databases to Amazon RDS Custom using RMAN duplication.

Oracle Data Guard

Dalam teknik ini, Anda mencadangkan basis data on-premise primer dan menyalin cadangan ke bucket Amazon S3. Kemudian Anda menyalin cadangan ke instans DB siaga RDS Custom for Oracle. Setelah melakukan konfigurasi yang diperlukan, Anda beralih secara manual dari basis data utama ke basis data siaga RDS Custom for Oracle. Untuk mempelajari teknik ini, lihat postingan blog AWS Physical migration of Oracle databases to Amazon RDS Custom using Data Guard.

Untuk informasi umum tentang impor data secara logis ke RDS for Oracle, lihat Mengimpor data ke Oracle di Amazon RDS.

Memutakhirkan instans basis data untuk Amazon RDS Custom for Oracle

Anda dapat memutakhirkan instans basis data Amazon RDS Custom dengan mengubahnya agar menggunakan versi mesin kustom (CEV) baru. Lihat informasi umum tentang pemutakhiran di Meng-upgrade versi mesin instans DB.

Topik

Ikhtisar pemutakhiran di RDS Custom for Oracle

Dengan RDS Custom for Oracle, Anda dapat menambal basis data Oracle atau sistem operasi (OS) instans basis data Anda dengan membuat CEV baru dan lalu mengubah instans Anda agar menggunakan CEV baru itu.

Topik

Opsi-opsi pemutakhiran CEV

Saat membuat CEV untuk pemutakhiran, Anda memiliki opsi-opsi yang saling eksklusif berikut:

Basis data saja

Gunakan ulang Amazon Machine Image (AMI) yang saat ini digunakan oleh instans basis data Anda, tetapi tentukan file biner basis data yang berbeda. RDS Custom mengalokasikan volume

biner baru, lalu melampirkannya pada instans Amazon EC2 yang ada. RDS Custom mengganti seluruh volume basis data dengan volume baru yang menggunakan versi basis data target Anda.

OS saja

Gunakan ulang file biner basis data yang saat ini digunakan oleh instans basis data Anda, tetapi tentukan AMI yang berbeda. RDS Custom mengalokasikan instans Amazon EC2 baru, lalu melampirkan volume biner yang ada pada instans yang baru. Volume basis data yang ada dipertahankan.

Jika ingin memutakhirkan OS dan basis data, Anda harus memutakhirkan CEV dua kali. Anda dapat memutakhirkan OS dan kemudian basis data, atau memutakhirkan basis data dan kemudian OS.

> **Warning**
>
> Ketika Anda menambal OS, Anda kehilangan data volume root dan segala kustomisasi OS yang ada. Maka, kami sangat menganjurkan agar Anda tidak menggunakan volume root untuk instalasi atau untuk menyimpan data atau file permanen. Kami juga menganjurkan supaya Anda membuat cadangan data sebelum melakukan pemutakhiran.

Menambal tanpa CEV

Kami sangat menganjurkan agar Anda memutakhirkan instans basis data RDS Custom for Oracle dengan menggunakan CEV. Automasi RDS Custom for Oracle menyinkronkan metadata tambalan dengan file biner basis data pada instans basis data Anda.

Dalam keadaan khusus, RDS Custom mendukung penerapan tambalan basis data "satu kali" secara langsung ke instans Amazon EC2 yang mendasari dengan menggunakan utilitas OPatch. Kasus penggunaan yang valid mungkin adalah tambalan basis data yang ingin Anda terapkan seketika, tetapi tim RDS Custom sedang memutakhirkan fitur CEV, sehingga ada ketertundaan. Untuk menerapkan tambalan basis data secara manual, lakukan langkah-langkah berikut:

1. Jeda automasi RDS Custom.
2. Terapkan tambalan Anda ke file biner basis data di instans Amazon EC2.
3. Lanjutkan automasi RDS Custom.

Kerugian teknik di atas adalah Anda harus menerapkan tambalan basis data secara manual ke setiap instans yang ingin Anda mutakhirkan. Sebaliknya, ketika membuat CEV baru, Anda dapat membuat atau memutakhirkan beberapa instans basis data dengan menggunakan CEV yang sama.

Langkah-langkah umum untuk menambal instans basis data Anda dengan CEV

Apakah Anda menambal OS atau basis data, lakukan langkah-langkah dasar berikut:

1. Buat CEV yang berisi salah satu elemen berikut, sesuai dengan apakah Anda menambal basis data atau OS:
   - Oracle Database RU yang ingin diterapkan untuk instans basis data Anda
   - AMI yang berbeda—entah yang terbaru yang tersedia atau yang Anda tentukan—dan CEV yang ada untuk digunakan sebagai sumber

   Ikuti langkah-langkah di Membuat CEV.
2. (Opsional untuk penambalan basis data) Periksa pemutakhiran versi mesin yang tersedia dengan menjalankan `describe-db-engine-versions`.
3. Mulai proses penambalan dengan menjalankan `modify-db-instance`.

   Status instans yang ditambal berbeda sebagai berikut:
   - Selagi RDS menambal basis data, status instans basis data berubah ke Memutakhirkan.
   - Selagi RDS menambal OS, status instans basis data berubah ke Memodifikasi.

   Ketika instans basis data memiliki status Tersedia, penambalan selesai.
4. Pastikan bahwa instans basis data Anda menggunakan CEV baru dengan menjalankan `describe-db-instances`.

Persyaratan untuk pemutakhiran RDS Custom for Oracle

Saat memutakhirkan instans basis data RDS Custom for Oracle ke CEV target, pastikan untuk memenuhi persyaratan berikut:

- CEV target yang menjadi sasaran pemutakhiran Anda harus ada.
- Anda harus memutakhirkan entah OS atau basis data dalam satu operasi. Memutakhirkan OS dan basis data sekaligus dalam satu panggilan API tidak didukung.

- CEV target harus menggunakan setelan parameter instalasi yang ada dalam manifes CEV saat ini. Misalnya, Anda tidak dapat memutakhirkan basis data yang menggunakan Oracle home default ke CEV yang menggunakan Oracle home non-default.
- Untuk pemutakhiran basis data, CEV target harus menggunakan versi basis data kecil baru, bukan versi utama baru. Misalnya, Anda tidak dapat memutakhirkan dari CEV Oracle Database 12c ke CEV Oracle Database 19c. Namun, Anda dapat memutakhirkan dari versi-versi 21.0.0.0.ru-2023-04.rur-2023-04.r1 ke versi 21.0.0.0.ru-2023-07.rur-2023-07.r1.
- Untuk pemutakhiran OS, CEV target harus menggunakan AMI yang berbeda, tetapi memiliki versi utama yang sama.

Pertimbangan-pertimbangan untuk pemutakhiran basis data RDS Custom for Oracle

Jika Anda merencanakan untuk memutakhirkan basis data, pertimbangkan hal-hal berikut:

- Saat Anda memutakhirkan file biner basis data di instans basis data utama Anda, RDS Custom for Oracle memutakhirkan replika baca Anda secara otomatis. Saat memutakhirkan OS, Anda harus memutakhirkan replika baca secara manual.
- Ketika Anda memutakhirkan basis data kontainer (CDB) ke versi basis data baru, RDS Custom for Oracle memeriksa bahwa semua PDB terbuka atau dapat dibuka. Jika kondisi ini tidak terpenuhi, RDS Custom menghentikan pemeriksaan dan mengembalikan basis data ke keadaan semula tanpa mencoba pemutakhiran. Jika kondisi terpenuhi, RDS Custom menambal dahulu root CDB, lalu menambal semua PDB lain (yang meliputi `PDB$SEED`) secara paralel.

  Setelah penambalan selesai, RDS Custom mencoba membuka semua PDB. Jika ada PDB yang gagal dibuka, Anda menerima peristiwa berikut: `The following PDBs failed to open: list-of-PDBs`. Jika RDS Custom gagal menambal root CDB atau salah satu PDB, instans ditempatkan ke dalam keadaan `PATCH_DB_FAILED`.
- Anda mungkin ingin melakukan pemutakhiran versi basis data utama dan konversi non-CDB ke CDB dengan serentak. Dalam hal ini, sebaiknya lanjutkan sebagai berikut:
  1. Buat RDS Custom baru untuk instans basis data Oracle yang menggunakan arsitektur multi-penghuni Oracle.
  2. Lekatkan non-CDB ke root CDB Anda, menjadikannya PDB. Pastikan bahwa non-CDB adalah versi utama yang sama dengan CDB Anda.
  3. Lakukan konversi PDB Anda dengan menjalankan skrip Oracle SQL `noncdb_to_pdb.sql`.
  4. Lakukan validasi instans CDB Anda.

5. Mutakhirkan instans CDB Anda.

Pertimbangan-pertimbangan untuk pemutakhiran OS RDS Custom for Oracle

Ketika Anda merencanakan pemutakhiran OS, pertimbangkan hal-hal berikut:

- Anda tidak dapat menyediakan AMI Anda sendiri untuk digunakan dalam CEV RDS Custom for Oracle. Anda dapat menentukan AMI default atau AMI yang pernah digunakan oleh CEV RDS Custom for Oracle.

  > **Note**
  >
  > RDS Custom for Oracle merilis AMI default baru ketika ditemukan kerentanan dan eksposur umum. Tidak ada jadwal tetap yang tersedia atau dijamin. RDS Custom for Oracle cenderung menerbitkan AMI default baru setiap 30 hari.

- Saat memutakhirkan OS di instans basis data utama, Anda harus memutakhirkan replika bacanya yang terkait secara manual.
- Cadangkan kapasitas komputasi Amazon EC2 yang cukup untuk jenis instans Anda di AZ sebelum mulai menambal OS.

  Saat membuat Reservasi Kapasitas, Anda menentukan AZ, jumlah instans, dan atribut-atribut instans (yang meliputi jenis instans). Misalnya, jika instans basis data Anda menggunakan instans EC2 tipe r5.large yang mendasari, sebaiknya cadangkan kapasitas EC2 untuk r5.large di AZ Anda. Selama penambalan OS, RDS Custom membuat satu host baru bertipe db.r5.large, yang dapat mogok jika AZ tidak memiliki kapasitas EC2 untuk jenis instans ini. Jika mencadangkan kapasitas EC2, Anda menurunkan risiko penambalan diblokir akibat batasan kapasitas. Lihat informasi yang lebih lengkap di Reservasi Kapasitas Atas Permintaan dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.
- Cadangkan instans basis data Anda sebelum memutakhirkan OS-nya. Pemutakhiran menghapus data volume root Anda dan segala kustomisasi OS yang ada.
- Dalam model tanggung jawab bersama, Anda bertanggung jawab untuk menjaga OS tetap mutakhir. RDS Custom for Oracle tidak mengamanatkan tambalan yang Anda terapkan untuk OS Anda. Jika RDS Custom for Oracle Anda berfungsi, Anda dapat menggunakan AMI yang terkait dengan CEV ini tanpa batas waktu.

Melihat target pemutakhiran CEV yang valid untuk instans basis data RDS Custom for Oracle

Anda dapat melihat CEV yang ada di halaman Versi mesin kustom di AWS Management Console.

Anda juga dapat menggunakan describe-db-engine-versions AWS CLI perintah untuk menemukan CEV yang valid untuk digunakan ketika Anda meng-upgrade instance DB Anda, seperti yang ditunjukkan pada contoh berikut. Contoh ini beranggapan bahwa Anda membuat instans basis data dengan menggunakan versi mesin `19.my_cev1`, dan bahwa ada versi-versi pemutakhiran `19.my_cev2` dan `19.my_cev`.

```
aws rds describe-db-engine-versions --engine custom-oracle-ee --engine-version
 19.my_cev1
```

Output-nya menyerupai yang berikut. Bidang `ImageId` menunjukkan ID AMI.

```
{
    "DBEngineVersions": [
        {
            "Engine": "custom-oracle-ee",
            "EngineVersion": "19.my_cev1",
            ...
            "Image": {
                "ImageId": "ami-2345",
                "Status": "active"
            },
            "DBEngineVersionArn": "arn:aws:rds:us-west-2:123456789012:cev:custom-
oracle-ee/19.my_cev1/12a34b5c-67d8-90e1-2f34-gh56ijk78lm9"
            "ValidUpgradeTarget": [
                {
                    "Engine": "custom-oracle-ee",
                    "EngineVersion": "19.my_cev2",
                    "Description": "19.my_cev2 description",
                    "AutoUpgrade": false,
                    "IsMajorVersionUpgrade": false
                },
                {
                    "Engine": "custom-oracle-ee",
                    "EngineVersion": "19.my_cev3",
                    "Description": "19.my_cev3 description",
                    "AutoUpgrade": false,
                    "IsMajorVersionUpgrade": false
```

```
        }
    ]
    ...
```

Memutakhirkan instans basis data RDS Custom for Oracle

Untuk memutakhirkan instans basis data RDS Custom for Oracle, ubahlah instans agar menggunakan CEV baru. CEV ini boleh berisi entah file biner basis data baru atau AMI baru. Jika ingin memutakhirkan basis data dan OS, Anda harus melakukan dua pemutakhiran terpisah.

> **Note**
>
> Jika Anda memutakhirkan basis data, RDS Custom memutakhirkan secara otomatis replika baca setelah memutakhirkan instans basis data utama. Jika Anda memutakhirkan OS, Anda harus memutakhirkan replika secara manual.

Sebelum Anda mulai, tinjau Persyaratan untuk pemutakhiran RDS Custom for Oracle dan Pertimbangan-pertimbangan untuk pemutakhiran basis data RDS Custom for Oracle.

Konsol

Untuk memutakhirkan instans basis data RDS Custom for Oracle

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data, lalu pilih instans basis data RDS Custom for Oracle yang ingin Anda mutakhirkan.
3. Pilih Ubah. Halaman Ubah instans basis data muncul.
4. Untuk Versi mesin basis data, pilih CEV baru. Lakukan hal-hal berikut:
   - Jika Anda menambal basis data, pastikan bahwa CEV menentukan file biner basis data yang berbeda dengan yang digunakan oleh instans basis data, dan tidak menentukan AMI yang berbeda dengan AMI yang saat ini digunakan oleh instans basis data.
   - Jika Anda menambal OS, pastikan bahwa CEV menentukan AMI yang berbeda dengan AMI yang saat ini digunakan oleh instans basis data, dan tidak menentukan file biner basis data yang berbeda.

> **Warning**
>
> Ketika Anda menambal OS, Anda kehilangan data volume root dan segala kustomisasi OS yang ada.

5. Pilih Lanjutkan untuk memeriksa ringkasan perubahan.

   Untuk menerapkan perubahan dengan serta-merta, pilih Terapkan seketika.

6. Jika perubahan Anda benar, pilih Ubah instans basis data. Atau pilih Kembali untuk mengedit perubahan atau Batalkan untuk membatalkan perubahan.

AWS CLI

Contoh-contoh berikut menunjukkan beberapa skenario pemutakhiran yang mungkin. Contoh-contoh beranggapan bahwa Anda membuat instans basis data RDS Custom for Oracle dengan karakteristik-karakteristik sebagai berikut:

- Instans basis data bernama `my-custom-instance`
- CEV bernama `19.my_cev1`
- Oracle Database 19c yang menggunakan arsitektur non-CDB
- Oracle Linux 7.9 yang menggunakan AMI `ami-1234`

AMI terbaru yang disediakan layanan adalah `ami-2345`. Anda dapat menemukan AMI dengan menjalankan perintah CLI `describe-db-engine-versions`.

Topik

- Memutakhirkan OS
- Memutakhirkan basis data

Memutakhirkan OS

Dalam contoh ini, Anda ingin memutakhirkan `ami-1234` ke `ami-2345`, yang merupakan AMI terbaru yang disediakan layanan. Karena ini pemutakhiran OS, file biner basis data untuk `ami-1234` dan `ami-2345` harus sama. Anda membuat CEV baru bernama `19.my_cev2` berdasarkan `19.my_cev1`.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds create-custom-db-engine-version \
    --engine custom-oracle-ee \
    --engine-version 19.my_cev2 \
    --description "Non-CDB CEV based on ami-2345" \
    --kms-key-id key-name \
    --source-custom-db-engine-version-identifer arn:aws:rds:us-
west-2:123456789012:cev:custom-oracle-ee/19.my_cev1/12345678-ab12-1234-cde1-
abcde123456789 \
    --image-id ami-2345
```

Untuk Windows:

```
aws rds create-custom-db-engine-version ^
    --engine custom-oracle-ee ^
    --engine-version 19.my_cev2 ^
    --description "Non-CDB CEV based on ami-2345" ^
    --kms-key-id key-name ^
    --source-custom-db-engine-version-identifer arn:aws:rds:us-
west-2:123456789012:cev:custom-oracle-ee/19.my_cev1/12345678-ab12-1234-cde1-
abcde123456789 ^
    --image-id ami-2345
```

Untuk memutakhirkan instans RDS Custom DB, gunakan modify-db-instance AWS CLI perintah dengan parameter berikut:

- `--db-instance-identifier` – Tentukan instans basis data RDS Custom for Oracle yang akan dimutakhirkan.
- `--engine-version` – Tentukan CEV yang memiliki AMI baru.
- `--no-apply-immediately` | `--apply-immediately` – Tentukan apakah akan melakukan pemutakhiran dengan serta-merta atau menunggu sampai jendela pemeliharaan terjadwal.

Contoh berikut memutakhirkan `my-custom-instance` ke versi `19.my_cev2`. Hanya OS yang dimutakhirkan.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --engine-version 19.my_cev2 \
    --apply-immediately
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --engine-version 19.my_cev2 ^
    --apply-immediately
```

Memutakhirkan basis data

Dalam contoh ini, Anda ingin menerapkan tambalan Oracle p35042068 ke instans basis data RDS for Oracle Anda. Karena Anda memutakhirkan OS di Memutakhirkan OS, instans basis data Anda saat ini menggunakan `19.my_cev2`, yang berdasarkan `ami-2345`. Anda membuat CEV baru bernama `19.my_cev3` yang juga menggunakan `ami-2345`, tetapi Anda menentukan manifes JSON baru dalam variabel lingkungan `$MANIFEST`. Maka, hanya file biner basis data yang berbeda di CEV baru Anda dan di CEV yang saat ini digunakan instans Anda.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds create-custom-db-engine-version \
    --engine custom-oracle-ee \
    --engine-version 19.my_cev3 \
    --description "Non-CDB CEV with p35042068 based on ami-2345" \
    --kms-key-id key-name \
    --image-id ami-2345 \
    --manifest $MANIFEST
```

Untuk Windows:

```
aws rds create-custom-db-engine-version ^
    --engine custom-oracle-ee ^
    --engine-version 19.my_cev3 ^
    --description "Non-CDB CEV with p35042068 based on ami-2345" ^
    --kms-key-id key-name ^
```

```
    --image-id ami-2345 ^
    --manifest $MANIFEST
```

Contoh berikut memutakhirkan `my-custom-instance` ke versi mesin `19.my_cev3`. Hanya basis data yang dimutakhirkan.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --engine-version 19.my_cev3 \
    --apply-immediately
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --engine-version 19.my_cev3 ^
    --apply-immediately
```

Melihat pemutakhiran basis data yang tertunda untuk instans basis data RDS Custom

Anda dapat melihat upgrade database tertunda untuk instans Amazon RDS Custom DB Anda dengan menggunakan perintah or. describe-db-instancesdescribe-pending-maintenance-actions AWS CLI

Namun, pendekatan ini tidak berfungsi jika Anda menggunakan opsi `--apply-immediately` atau jika pemutakhiran sedang berlangsung.

Perintah `describe-db-instances` berikut menunjukkan pemutakhiran basis data tertunda untuk `my-custom-instance`.

```
aws rds describe-db-instances --db-instance-identifier my-custom-instance
```

Output-nya menyerupai yang berikut.

```
{
    "DBInstances": [
        {
```

```
            "DBInstanceIdentifier": "my-custom-instance",
             "EngineVersion": "19.my_cev1",
             ...
             "PendingModifiedValues": {
                 "EngineVersion": "19.my_cev3"
             ...
             }
        }
    ]
}
```

Memecahkan masalah kegagalan pemutakhiran untuk instans basis data RDS Custom for Oracle

Jika pemutakhiran instans basis data RDS Custom gagal, peristiwa RDS dihasilkan dan status instans basis data menjadi `upgrade-failed`.

Anda dapat melihat status ini dengan menggunakan describe-db-instances AWS CLI perintah, seperti yang ditunjukkan pada contoh berikut.

```
aws rds describe-db-instances --db-instance-identifier my-custom-instance
```

Output-nya menyerupai yang berikut.

```
{
    "DBInstances": [
        {
           "DBInstanceIdentifier": "my-custom-instance",
            "EngineVersion": "19.my_cev1",
            ...
            "PendingModifiedValues": {
                "EngineVersion": "19.my_cev3"
            ...
            }
            "DBInstanceStatus": "upgrade-failed"
        }
    ]
}
```

Setelah suatu kegagalan pemutakhiran, semua tindakan basis data diblokir kecuali untuk mengubah instans basis data guna melakukan tugas-tugas berikut:

- Mencoba lagi pemutakhiran yang sama
- Menjeda dan melanjutkan automasi RDS Custom
- oint-in-time Pemulihan P (PITR)
- Menghapus instans basis data

> **Note**
>
> Jika automasi telah dijeda untuk instans basis data RDS Custom, Anda tidak dapat mencoba lagi pemutakhiran sampai melanjutkan automasi.
> Tindakan yang sama berlaku untuk kegagalan pemutakhiran bagi replika baca yang dikelola RDS sebagaimana untuk yang utama.

Untuk informasi selengkapnya, lihat Memecahkan masalah pemutakhiran untuk RDS Custom for Oracle.

Memecahkan masalah basis data untuk Amazon RDS Custom for Oracle

Model tanggung jawab bersama RDS Custom menyediakan akses tingkat shell OS dan akses administrator basis data. RDS Custom menjalankan sumber daya di akun Anda, tidak seperti Amazon RDS, yang menjalankan sumber daya di akun sistem. Bersama akses yang lebih besar datang tanggung jawab yang lebih besar. Pada bagian-bagian berikut, Anda dapat mempelajari cara memecahkan masalah pada instans basis data Amazon RDS Custom.

> **Note**
>
> Bagian ini menjelaskan cara memecahkan masalah RDS Custom for Oracle. Lihat pemecahan masalah RDS Custom for SQL Server di Memecahkan masalah basis data untuk Amazon RDS Custom for SQL Server.

Topik

Melihat peristiwa RDS Custom

Prosedur untuk melihat peristiwa adalah sama untuk instans basis data RDS Custom dan Amazon RDS. Untuk informasi selengkapnya, lihat Melihat peristiwa Amazon RDS.

Untuk melihat pemberitahuan acara khusus RDS menggunakan AWS CLI, gunakan `describe-events` perintah. RDS Custom memperkenalkan beberapa peristiwa baru. Kategori-kategori peristiwa sama dengan untuk Amazon RDS. Lihat daftar peristiwa di Kategori peristiwa dan pesan peristiwa Amazon RDS.

Contoh berikut mengambil perincian peristiwa-peristiwa yang telah terjadi untuk instans basis data RDS Custom yang ditentukan.

```
aws rds describe-events \
    --source-identifier my-custom-instance \
    --source-type db-instance
```

Berlangganan acara RDS Custom

Prosedur untuk berlangganan peristiwa sama untuk instans basis data RDS Custom dan Amazon RDS. Untuk informasi selengkapnya, lihat Berlangganan pemberitahuan peristiwa Amazon RDS.

Untuk berlangganan notifikasi peristiwa RDS Custom dengan menggunakan CLI, gunakan perintah `create-event-subscription`. Sertakan parameter-parameter yang diperlukan berikut:

- `--subscription-name`
- `--sns-topic-arn`

Contoh berikut membuat pelangganan untuk peristiwa-peristiwa pencadangan dan pemulihan untuk sebuah instans basis data RDS Custom di akun AWS saat ini. Notifikasi dikirim ke topik Amazon Simple Notification Service (Amazon SNS), yang ditentukan oleh `--sns-topic-arn`.

```
aws rds create-event-subscription \
    --subscription-name my-instance-events \
    --source-type db-instance \
    --event-categories '["backup","recovery"]' \
    --sns-topic-arn arn:aws:sns:us-east-1:123456789012:interesting-events
```

Memecahkan masalah pembuatan versi mesin kustom untuk RDS Custom for Oracle

Jika pembuatan CEV gagal, RDS Custom menerbitkan `RDS-EVENT-0198` dengan pesan `Creation failed for custom engine version` *`major-engine-version.cev_name`* dan menyertakan detail kegagalan itu. Misalnya, peristiwa mencetak file yang hilang.

Pembuatan CEV mungkin gagal karena masalah berikut:

- Bucket Amazon S3 yang berisi file instalasi Anda tidak berada di AWS Wilayah yang sama dengan CEV Anda.
- Saat Anda meminta pembuatan CEV Wilayah AWS untuk pertama kalinya, RDS Custom membuat bucket S3 untuk menyimpan sumber daya Kustom RDS (seperti artefak CEV, AWS CloudTrail log, dan log transaksi).

Pembuatan CEV gagal jika RDS Custom tidak dapat membuat bucket S3. Entah pemanggil tidak memiliki izin-izin S3 seperti dijelaskan di Langkah 5: Berikan izin yang diperlukan ke pengguna atau peran IAM Anda, atau jumlah bucket S3 telah mencapai batas.

- Pemanggil tidak memiliki izin-izin untuk mendapatkan file dari bucket S3 yang berisi file-file media instalasi. Izin-izin ini dijelaskan di Langkah 7: Tambahkan izin IAM yang diperlukan.
- Kebijakan IAM Anda memiliki syarat `aws:SourceIp`. Pastikan untuk mengikuti rekomendasi di AWS Menolak akses ke AWS berdasarkan IP sumber dalam Panduan Pengguna AWS Identity and Access Management . Pastikan juga bahwa pemanggil memiliki izin S3 yang dijelaskan di Langkah 5: Berikan izin yang diperlukan ke pengguna atau peran IAM Anda.
- File-file media instalasi yang tercantum dalam manifes CEV tidak ada di bucket S3 Anda.
- Checksum SHA-256 file-file instalasi tidak dikenal bagi RDS Custom.

  Pastikan bahwa checksum SHA-256 file-file yang disediakan cocok dengan checksum SHA-256 di situs web Oracle. Jika checksum cocok, hubungi Dukungan AWS dan berikan nama CEV, nama file, dan checksum yang gagal.
- Versi OPatch tidak kompatibel dengan file-file tambalan Anda. Anda mungkin mendapatkan pesan berikut: `OPatch is lower than minimum required version. Check that the version meets the requirements for all patches, and try again.` Untuk menerapkan tambalan Oracle, Anda harus menggunakan versi utilitas OPatch yang kompatibel. Anda dapat menemukan versi utilitas OPatch yang disyaratkan di file readme tambalan. Unduh utilitas OPatch terbaru dari My Oracle Support, dan coba buat CEV lagi.
- Tambalan yang ditentukan dalam manifes CEV berurutan salah.

Anda dapat melihat peristiwa RDS baik di konsol RDS (di panel navigasi, pilih Acara) atau dengan menggunakan perintah. `describe-events` AWS CLI Durasi bawaan adalah 60 menit. Jika tidak ada peristiwa yang ditampilkan, tentukan durasi yang lebih lama, seperti ditunjukkan pada contoh berikut.

```
aws rds describe-events --duration 360
```

Saat ini, MediaImport layanan yang mengimpor file dari Amazon S3 untuk membuat CEV tidak terintegrasi dengannya. AWS CloudTrail Oleh karena itu, jika Anda mengaktifkan pencatatan data untuk Amazon RDS CloudTrail, panggilan ke MediaImport layanan seperti `CreateCustomDbEngineVersion` peristiwa tidak dicatat.

Namun, Anda mungkin melihat panggilan dari gateway API yang mengakses bucket Amazon S3 Anda. Panggilan ini berasal dari MediaImport layanan untuk `CreateCustomDbEngineVersion` acara tersebut.

Memperbaiki konfigurasi yang tidak didukung di RDS Custom for Oracle

Dalam model tanggung jawab bersama, Anda bertanggung jawab untuk memperbaiki masalah konfigurasi yang menempatkan instans basis data RDS Custom for Oracle Anda ke dalam keadaan `unsupported-configuration`. Jika masalahnya ada pada AWS infrastruktur, Anda dapat menggunakan konsol atau AWS CLI untuk memperbaikinya. Jika masalahnya ada pada sistem operasi atau konfigurasi basis data, Anda dapat masuk ke host untuk memperbaikinya.

> **Note**
>
> Bagian ini menjelaskan cara memperbaiki konfigurasi yang tidak didukung di RDS Custom for Oracle. Lihat informasi tentang RDS Custom for SQL Server di Memperbaiki konfigurasi yang tidak didukung di RDS Custom for SQL Server.

Pada tabel berikut, Anda dapat menemukan deskripsi notifikasi dan peristiwa yang dikirim oleh perimeter dukungan dan cara memperbaikinya. Semua notifikasi ini dan perimeter dukungan dapat berubah sewaktu-waktu. Lihat latar belakang perimeter dukungan di Perimeter dukungan RDS Custom. Lihat deskripsi peristiwa di Kategori peristiwa dan pesan peristiwa Amazon RDS.

ID peristi	Konfigurasi	Pesan peristiwa RDS	Tindakan
SP-00000	Konfigurasi manual yang tidak didukung	*Status instans RDS Custom DB disetel ke [Konfigurasi tidak didukung] karena: alasan.*	Untuk mengatasi masalah ini, buat AWS Support kasus.
AWS sumber daya (infrastruktur)			

ID peristi	Konfigura si	Pesan peristiwa RDS	Tindakan
SP-O1001	Volume Amazon Elastic Block Store (Amazon EBS)	*Volume EBS berikut ditambahkan ke instance EC2 ec2_id: volume_id.* Untuk mengatasi masalah ini, lepaskan volume yang ditentukan dari instance.	RDS Custom membuat dua jenis volume EBS, selain volume root yang dibuat dari Amazon Machine Image (AMI), dan mengaitkannya dengan instans EC2: • Volume biner tempat biner perangkat lunak database berada • Volume data tempat file database berada Saat Anda membuat instans DB, konfigurasi penyimpanan yang Anda tentukan mengonfigurasi volume data. Perimeter dukungan memantau hal-hal berikut: • Volume EBS awal yang dibuat dengan instans DB masih terkait dengan instance. • Volume EBS awal masih memiliki konfigurasi yang sama dengan yang awalnya ditetapkan: jenis penyimpanan, ukuran, IOPS yang Tersedia, dan throughput penyimpanan. • Tidak ada volume EBS tambahan yang dilampirkan pada instans basis data. Gunakan perintah CLI berikut untuk membandingkan jenis volume detail volume EBS dan detail instans RDS Custom for Oracle DB: `aws rds describe-db-instances \ --db-instance-identifier db-instance-name | grep StorageType`

ID peristi	Konfigura si	Pesan peristiwa RDS	Tindakan
SP-O1002	Volume Amazon Elastic Block Store (Amazon EBS)	*Volume EBS volume_id telah terlepas dari instance EC2 [ec2_id].* Anda tidak dapat melepaska n volume asli dari instance ini. Untuk mengatasi masalah ini, pasang kembali *volume_id ke ec2_id.*	RDS Custom membuat dua jenis volume EBS, selain volume root yang dibuat dari Amazon Machine Image (AMI), dan mengaitkannya dengan instans EC2: • Volume biner tempat biner perangkat lunak database berada • Volume data tempat file database berada Saat Anda membuat instans DB, konfigurasi penyimpanan yang Anda tentukan mengonfigurasi volume data. Perimeter dukungan memantau hal-hal berikut: • Volume EBS awal yang dibuat dengan instans DB masih terkait dengan instance. • Volume EBS awal masih memiliki konfigurasi yang sama dengan yang awalnya ditetapkan: jenis penyimpanan, ukuran, IOPS yang Tersedia, dan throughput penyimpanan. • Tidak ada volume EBS tambahan yang dilampirkan pada instans basis data. Gunakan perintah CLI berikut untuk membandingkan jenis volume detail volume EBS dan detail instans RDS Custom for Oracle DB: `aws rds describe-db-instances \ --db-instance-identifier db-instance-name | grep StorageType`

ID peristi	Konfigura si	Pesan peristiwa RDS	Tindakan
SP-O1003	Volume Amazon Elastic Block Store (Amazon EBS)	*Volume volume EBS asli yang dilampirkan ke EC2 instance ec2_id telah dimodifikasi sebagai berikut: size [X] to [Y], type [N] to [M], atau IOPS [J] to [K].* Untuk mengatasi masalah, kembalikan modifikasi.	RDS Custom membuat dua jenis volume EBS, selain volume root yang dibuat dari Amazon Machine Image (AMI), dan mengaitkannya dengan instans EC2: • Volume biner tempat biner perangkat lunak database berada • Volume data tempat file database berada Saat Anda membuat instans DB, konfigurasi penyimpanan yang Anda tentukan mengonfigurasi volume data. Perimeter dukungan memantau hal-hal berikut: • Volume EBS awal yang dibuat dengan instans DB masih terkait dengan instance. • Volume EBS awal masih memiliki konfigurasi yang sama dengan yang awalnya ditetapkan: jenis penyimpanan, ukuran, IOPS yang Tersedia, dan throughput penyimpanan. • Tidak ada volume EBS tambahan yang dilampirkan pada instans basis data. Gunakan perintah CLI berikut untuk membandingkan jenis volume detail volume EBS dan detail instans RDS Custom for Oracle DB: `aws rds describe-db-instances \ --db-instance-identifier db-instance-name | grep StorageType`

ID peristi	Konfigura si	Pesan peristiwa RDS	Tindakan
SP-O1004	Keadaan instans Amazon EC2	Pemulihan otomatis meninggalkan instans EC2 [*ec2_id*] dalam keadaan terganggu . Untuk mengatasi masalah ini, lihat Memecahkan masalah kegagalan pemulihan instans.	Untuk memeriksa status instans DB, gunakan konsol atau jalankan AWS CLI perintah berikut: `aws rds describe-db-instances \ --db-instance-identifier db-instance-name |grep DBInstanceStatus`
SP-O1005	Atribut-atribut instans Amazon EC2	*Instance EC2 [ec2_id] dimodifikasi sebagai berikut: atribut [att1] berubah dari [val-old] menjadi [val-new], atribut [att2] diubah dari [val-old] menjadi [val-new].* Untuk mengatasi masalah, kembalikan ke nilai aslinya.	

ID peristi	Konfigurasi	Pesan peristiwa RDS	Tindakan
SP-O1006	Keadaan instans Amazon EC2	Instans EC2 [*ec2_id*] dihentikan atau tidak dapat ditemukan. Untuk mengatasi masalah ini, hapus instans RDS Custom DB.	Perimeter dukungan memantau notifikasi perubahan keadaan instans EC2. Instans EC2 harus selalu berjalan. Untuk menghapus instans DB Anda 1. Untuk memeriksa status instans DB, gunakan konsol atau jalankan AWS CLI perintah berikut: `aws rds describe-db-instances \ --db-instance-identifier db-instance-name |grep DBInstanceStatus` 2. Hapus RDS Custom Anda untuk instans Oracle DB.
SP-O1007	Keadaan instans Amazon EC2	Instans EC2 [*ec2_id*] dihentikan. Untuk mengatasi masalah ini, mulailah instance.	Perimeter dukungan memantau notifikasi perubahan keadaan instans EC2. Instans EC2 harus selalu berjalan. Untuk memulai ulang instans DB Anda 1. Untuk memeriksa status instans DB, gunakan konsol atau jalankan AWS CLI perintah berikut: `aws rds describe-db-instances \ --db-instance-identifier db-instance-name |grep DBInstanceStatus` 2. Mulai instans DB Anda. 3. Tumpangkan ulang volume-volume biner dan data.
Sistem operasi			

ID peristi	Konfigura si	Pesan peristiwa RDS	Tindakan

SP-O2001	Status agen RDS Custom	*Agen Kustom RDS tidak berjalan pada instans EC2 [ec2_id].* Pastikan agen berjalan di [*ec2_id*].	Pada RDS Custom for Oracle, instans basis data berada di luar perimeter dukungan jika agen RDS Custom berhenti. Agen menerbitkan `IamAlive` metrik ke Amazon CloudWatch setiap 30 detik. Alarm terpicu jika metrik tidak diterbitkan selama 30 detik. Perimeter dukungan juga memantau status proses agen RDS Custom pada host setiap 30 menit.
			
			Untuk memulai ulang agen Kustom RDS
			
			1. Masuk ke host Anda dan pastikan bahwa agen RDS Custom berjalan.
			2. Jalankan perintah berikut untuk menemukan status agen.
			
			`service rdscustomagent status`
			
			3. Gunakan perintah berikut untuk memulai agen.
			
			`service rdscustomagent start`
			
			Saat agen Kustom RDS berjalan lagi, `IamAlive` metrik dipublikasikan ke Amazon CloudWatch, dan alarm beralih ke `OK` status. Peralihan ini memberi tahu perimeter dukungan bahwa agen sedang berjalan.

ID peristi	Konfigura si	Pesan peristiwa RDS	Tindakan
SP-02002	AWS Systems Manager status agen (agen SSM)	Agen Systems Manager pada instans EC2 [*ec2_id*] tidak dapat dijangkau . Pastikan Anda telah mengonfig urasi izin jaringan, agen, dan IAM dengan benar.	Agen SSM harus selalu berjalan. Agen RDS Custom bertanggung jawab untuk memastikan bahwa agen Systems Manager berjalan. Jika Agen SSM dihentika n dan kemudian dimulai ulang, agen Kustom RDS menerbitkan metrik ke. CloudWatch Agen RDS Custom mengatur alarm set metrik agar terpicu ketika telah ada pemulaian ulang dalam setiap menit dari dari tiga menit ke belakang. Perimeter dukungan juga memantau status proses Agen SSM di host setiap 30 menit. Lihat informasi yang lebih lengkap di Memecahkan masalah Agen SSM.
SP-02003	AWS Systems Manager status agen (agen SSM)	Agen Systems Manager pada instans EC2 [*ec2_id*] crash beberapa kali. Untuk informasi selengkapnya, lihat dokumenta si pemecahan masalah Agen SSM.	Lihat informasi yang lebih lengkap di Memecahkan masalah Agen SSM.

ID peristi	Konfigura si	Pesan peristiwa RDS	Tindakan
SP-O2004	Zona waktu OS	Zona waktu pada instans EC2 [*ec2_id*] diubah. Untuk mengatasi masalah ini, kembalikan zona waktu ke pengatura n sebelumnya [] *previous-time-zone* . Kemudian gunakan grup opsi RDS untuk mengubah zona waktu.	Automasi RDS mendeteksi bahwa zona waktu pada host diubah tanpa menggunakan grup opsi. Perubahan tingkat host ini dapat menyebabkan kegagalan automasi RDS, sehingga instans EC2 ditempatkan dalam keadaan `unsupported-configuration` . Untuk memperbaiki pengaturan zona waktu 1. Masuk ke host EC2 Anda dan periksa zona waktu OS sebagai berikut: `timedatectl` 2. Jeda otomatisasi RDS Custom. Untuk informasi selengkapnya, lihat Menjeda dan melanjutkan instans RDS Custom DB. 3. Hentikan instans basis data. 4. Kembalikan perubahan zona waktu pada sistem operasi. 5. Mulai instans basis data. 6. Lanjutkan automasi RDS Custom. Instans basis data Anda menjadi tersedia dalam 30 menit. Untuk mencegah keluar dari perimeter di masa depan, ubah zona waktu Anda melalui grup opsi. Untuk informasi selengkapnya, lihat Zona waktu Oracle.

ID peristi	Konfigura si	Pesan peristiwa RDS	Tindakan
SP-O200	Konfigura si-konfig urasi sudo	Konfigurasi sudo pada instans EC2 [*ec2_id*] tidak memiliki izin yang diperlukan. Untuk mengatasi masalah ini, kembalikan perubahan terbaru ke konfigurasi sudo.	Perimeter dukungan memantau bahwa pengguna OS tertentu diizinkan untuk menjalankan perintah tertentu pada kotak. Perimeter memantau konfigurasi-konfig urasi sudo terhadap keadaan yang didukung. Ketika konfigurasi-konfigurasi sudo tidak didukung, RDS Custom mencoba menindas agar kembali ke keadaan terdukung sebelumnya. Jika berhasil, notifikasi berikut dikirim: RDS Custom berhasil menindas konfigurasi Anda. Untuk menyelidiki perubahan pada konfigurasi sudo 1. Masuk ke host Anda. 2. Jalankan perintah berikut. `visudo -c -f /etc/sudoers.d/ individual_sudo_files` 3. Ubah sudo konfigurasi seperlunya. Setelah perimeter dukungan menentukan bahwa sudo konfigurasi didukung, instans RDS Custom for Oracle DB Anda akan tersedia dalam waktu 30 menit.

ID peristi	Konfigura si	Pesan peristiwa RDS	Tindakan
SP-O2006	Aksesibil itas bucket S3	*Otomatisasi khusus RDS tidak dapat mengunduh file dari bucket S3 pada instans EC2 [ec2_id].* Periksa konfigurasi jaringan Anda dan pastikan instance memungkinkan koneksi ke dan dari S3.	
Basis data			

ID peristi	Konfigura si	Pesan peristiwa RDS	Tindakan
SP-O3001	Target kelambata n arsip basis data	*Parameter ARCHIVE_L AG_TARGET pada instance EC2 [ec2_id] berada di luar rentang value_range yang direkomen dasikan.* Untuk mengatasi masalah, atur parameter ke nilai dalam value_range.	Perimeter dukungan memantau parameter `ARCHIVE_LAG_TARGET` database untuk memverifi kasi bahwa waktu restorable terbaru dari instans DB berada dalam batas yang wajar. Untuk mengubah target lag untuk log pengulangan yang diarsipkan 1. Masuk ke host EC2 Anda 2. Connect ke instans RDS Custom for Oracle DB Anda 3. Ubah `ARCHIVE_LAG_TARGET` parameter menjadi nilai dari 60-7200. Misalnya, gunakan pernyataan SQL berikut. `ALTER SYSTEM SET ARCHIVE_LAG_TARGET=300 SCOPE=BOTH;` Instans basis data Anda menjadi tersedia dalam 30 menit.

ID peristi	Konfigura si	Pesan peristiwa RDS	Tindakan

SP-O3002	Peran Oracle Data Guard	*Peran database [role_name] tidak didukung untuk Oracle Data Guard pada instans EC2 [ec2_id].* Untuk mengatasi masalah, setel parameter DATABASE_ROLE ke PRIMARY atau PHYSICAL STANDBY.	Perimeter dukungan memantau peran database saat ini setiap 15 detik dan mengirimkan CloudWatch pemberitahuan jika peran database telah berubah. Parameter `DATABASE_ROLE` Oracle Data Guard harus `PRIMARY` atau `PHYSICAL STANDBY`.
			
			Untuk mengembalikan peran database Oracle Data Guard Anda ke nilai yang didukung
			
			1. Periksa peran Oracle Data Guard dengan menjalankan pernyataan berikut:
			
			`SELECT DATABASE_ROLE FROM V$DATABASE;`
			
			2. Jika instans DB Anda berdiri sendiri, gunakan salah satu pernyataan berikut untuk mengubahnya kembali ke `PRIMARY` peran:
			
			`ALTER DATABASE COMMIT TO SWITCHOVER PRIMARY;`
			`ALTER DATABASE ACTIVATE STANDBY DATABASE;`
			
			Jika instans DB Anda adalah replika, gunakan pernyataan berikut untuk mengubahnya kembali ke `PHYSICAL STANDBY` peran:
			
			`ALTER DATABASE CONVERT TO PHYSICAL STANDBY;`
			
			Setelah perimeter dukungan menentukan bahwa peran basis data didukung, instans basis data RDS Custom for Oracle Anda akan tersedia dalam waktu 15 detik.

ID peristi	Konfigura si	Pesan peristiwa RDS	Tindakan
SP-O3003	Kesehatan basis data	Proses SMON dari database Oracle berada dalam keadaan zombie. Untuk mengatasi masalah ini, pulihkan database secara manual pada instans EC2 [*ec2_id*], buka database, lalu segera buat cadangan. Untuk bantuan lebih lanjut, hubungi AWS Support.	Perimeter dukungan memantau keadaan instans basis data. Perimeter juga memantau jumlah pemulaian ulang yang terjadi selama jam dan hari sebelumnya. Anda diberi tahu saat instans berada dalam keadaan dengan instans masih ada, tetapi Anda tidak dapat berinteraksi dengannya. Untuk membuat perimeter dukungan mengevaluasi status instans Anda 1. Masuk ke host Anda dan tentukan status database. `ps -eo pid,state,command | grep smon` 2. Jika perlu, restart instans DB Anda. Jika restart gagal, lanjutkan ke langkah berikutnya. 3. Jika perlu, restart host EC2 Anda. Setelah instans DB Anda restart, agen RDS Custom mendeteksi bahwa instans DB Anda tidak lagi dalam keadaan tidak responsif. Agen itu lalu memberi tahu perimeter dukungan untuk mengevaluasi ulang keadaan instans basis data Anda.

ID peristi	Konfigura si	Pesan peristiwa RDS	Tindakan

SP-O3004	Mode log basis data	*Mode log database pada instans EC2 [ec2_id] diubah menjadi [value_b].* Untuk mengatasi masalah ini, atur mode log ke [*value_a*].	Untuk mengubah mode log instans DB Anda menjadi **ARCHIVELOG**
			1. Masuk ke host EC2 Anda.
			2. Connect ke database Anda dan jalankan pernyataa n berikut:
			`SELECT LOG_MODE FROM V$DATABASE;`
			Atau Anda dapat menjalankan perintah ikuti di SQL* Plus:
			`ARCHIVE LOG LIST`
			3. Jalankan perintah SQL* Plus berikut untuk memulai shutdown yang konsisten.
			`SHUTDOWN IMMEDIATE`
			Agen Kustom RDS secara otomatis me-restart instans DB Anda dan menyetel mode log ke. `ARCHIVELOG` Instans basis data Anda menjadi tersedia dalam 30 menit.

ID peristi	Konfigura si	Pesan peristiwa RDS	Tindakan
SP-O3005	Jalur rumah Oracle	*Beranda Oracle pada instans EC2 [ec2_id] diubah menjadi new_path.* Untuk mengatasi masalah, kembalikan pengaturan ke *old_path*.	

SP-O3006	Nama unik database	*Nama unik database pada instans EC2 [ec2_id] diubah menjadi new_value .* Untuk mengatasi masalah ini, kembalikan nama ke *old_value* .	Untuk mengubah nama unik database untuk instans DB Anda
			1. Masuk ke host EC2 Anda.
			2. Connect ke database dan jalankan pernyataan berikut:
			`SELECT DB_UNIQUE_NAME FROM V$DATABASE;`
			3. Tentukan nama unik database asli menggunak an perintah`ALTER SYSTEM SET DB_UNIQUE _NAME` .
			4. Jalankan pernyataan SQL berikut untuk memulai shutdown yang konsisten.
			`SHUTDOWN IMMEDIATE;`
			Agen Kustom RDS secara otomatis me-restart instans DB Anda dan menyetel mode log ke. `ARCHIVELOG` Instans basis data Anda menjadi tersedia dalam 30 menit.

Memecahkan masalah pemutakhiran untuk RDS Custom for Oracle

Pemutakhiran instans RDS Custom for Oracle Anda mungkin gagal. Di bawah, Anda dapat menemukan teknik-teknik yang dapat Anda gunakan selama pemutakhiran basis data RDS Custom untuk instans basis data Oracle:

- Periksa file log output pemutakhiran di direktori `/tmp` pada instans basis data Anda. Nama-nama log bergantung pada versi mesin basis data Anda. Misalnya, Anda mungkin melihat log yang berisi string `catupgrd` atau`catup`.
- Periksa file `alert.log` yang terletak di direktori `/rdsdbdata/log/trace`.
- Jalankan perintah `grep` berikut di direktori `root` untuk melacak proses pemutakhiran OS. Perintah ini menunjukkan letak file log sedang ditulis dan menentukan keadaan proses pemutakhiran.

  ```
  ps -aux | grep upg
  ```

  Berikut menampilkan contoh output.

  ```
  root     18884  0.0  0.0 235428  8172 ?        S<   17:03   0:00 /usr/bin/
  sudo -u rdsdb /rdsdbbin/scripts/oracle-control ORCL op_apply_upgrade_sh RDS-
  UPGRADE/2.upgrade.sh
  rdsdb    18886  0.0  0.0 153968 12164 ?        S<   17:03   0:00 /usr/bin/perl -T -
  w /rdsdbbin/scripts/oracle-control ORCL op_apply_upgrade_sh RDS-UPGRADE/2.upgrade.sh
  rdsdb    18887  0.0  0.0 113196  3032 ?        S<   17:03   0:00 /bin/sh /rdsdbbin/
  oracle/rdbms/admin/RDS-UPGRADE/2.upgrade.sh
  rdsdb    18900  0.0  0.0 113196  1812 ?        S<   17:03   0:00 /bin/sh /rdsdbbin/
  oracle/rdbms/admin/RDS-UPGRADE/2.upgrade.sh
  rdsdb    18901  0.1  0.0 167652 20620 ?        S<   17:03   0:07 /rdsdbbin/oracle/
  perl/bin/perl catctl.pl -n 4 -d /rdsdbbin/oracle/rdbms/admin -l /tmp catupgrd.sql
  root     29944  0.0  0.0 112724  2316 pts/0    S+   18:43   0:00 grep --color=auto
   upg
  ```

- Jalankan kueri SQL berikut untuk memeriksa keadaan komponen-komponen saat ini untuk menemukan versi basis data dan opsi-opsi yang diinstal pada instans basis data.

  ```
  SET LINESIZE 180
  COLUMN COMP_ID FORMAT A15
  COLUMN COMP_NAME FORMAT A40 TRUNC
  COLUMN STATUS FORMAT A15 TRUNC
  SELECT COMP_ID, COMP_NAME, VERSION, STATUS FROM DBA_REGISTRY ORDER BY 1;
  ```

Outputnya menyerupai berikut.

```
COMP_NAME                                STATUS               PROCEDURE
---------------------------------------- --------------------
 ------------------------------------------------
Oracle Database Catalog Views            VALID
 DBMS_REGISTRY_SYS.VALIDATE_CATALOG
Oracle Database Packages and Types       VALID
 DBMS_REGISTRY_SYS.VALIDATE_CATPROC
Oracle Text                              VALID                VALIDATE_CONTEXT
Oracle XML Database                      VALID                DBMS_REGXDB.VALIDATEXDB

4 rows selected.
```

- Jalankan kueri SQL berikut untuk memeriksa objek-objek yang tidak valid yang mungkin mengganggu proses pemutakhiran.

```
SET PAGES 1000 LINES 2000
COL OBJECT FOR A40
SELECT SUBSTR(OWNER,1,12) OWNER,
       SUBSTR(OBJECT_NAME,1,30) OBJECT,
       SUBSTR(OBJECT_TYPE,1,30) TYPE, STATUS,
       CREATED
FROM   DBA_OBJECTS
WHERE  STATUS <>'VALID'
AND    OWNER IN ('SYS','SYSTEM','RDSADMIN','XDB');
```

Memecahkan masalah promosi replika untuk RDS Custom for Oracle

Anda dapat mempromosikan replika Oracle terkelola di RDS Custom for Oracle menggunakan konsol, `promote-read-replica` AWS CLI perintah, atau API. `PromoteReadReplica` Jika Anda menghapus instans DB primer Anda, dan semua replika berkondisi baik, RDS Custom for Oracle akan mempromosikan replika terkelola Anda ke instans mandiri secara otomatis. Jika replika telah menjeda automasi atau berada di luar perimeter dukungan, Anda harus memperbaiki replika sebelum RDS Custom dapat mempromosikannya secara otomatis. Untuk informasi selengkapnya, lihat Batasan promosi replika untuk RDS Custom for Oracle.

Alur kerja promosi replika mungkin macet dalam situasi berikut:

- Instans basis data utama berada dalam keadaan `STORAGE_FULL`.

- Basis data utama tidak dapat mengarsipkan semua log redo daring/online.
- Ada celah antara file log redo yang diarsipkan di replika Oracle Anda dan basis data utama.

Untuk merespons alur kerja yang macet, selesaikan langkah-langkah berikut:

1. Sinkronkan celah log redo pada instans basis data replika Oracle Anda.
2. Paksa promosi replika baca ke log redo terbaru yang diterapkan. Jalankan perintah SQL*Plus berikut:

```
ALTER DATABASE ACTIVATE STANDBY DATABASE;
SHUTDOWN IMMEDIATE
STARTUP
```

3. Hubungi AWS Support dan minta untuk memindahkan instans DB Anda ke status yang tersedia.

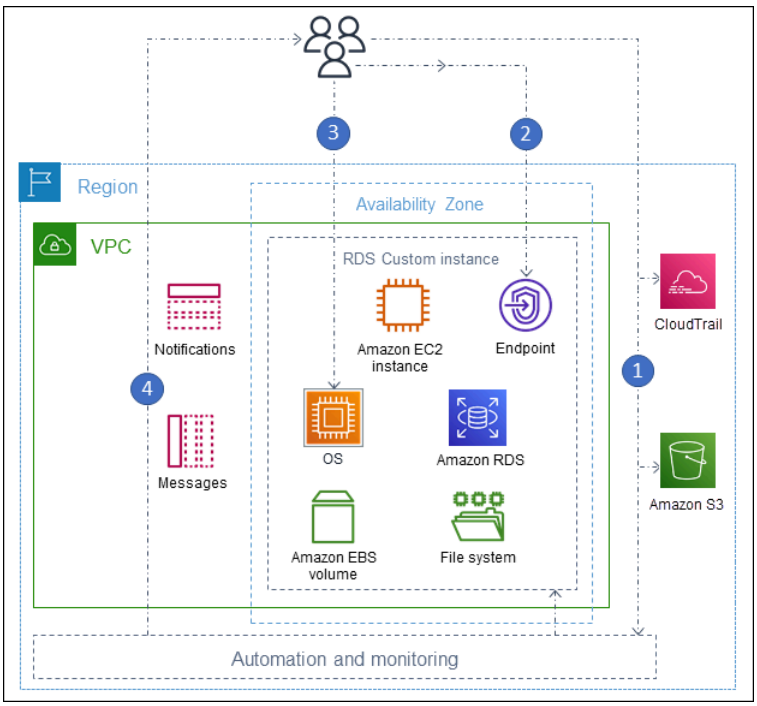
Menggunakan RDS Custom for SQL Server

Di bagian berikut ini, Anda dapat menemukan petunjuk untuk membuat, mengelola, dan memelihara instans DB RDS Custom for SQL Server.

Topik

Alur kerja RDS Custom for SQL Server

Diagram berikut menunjukkan alur kerja yang lazim untuk RDS Custom for SQL Server.



Langkah konversinya adalah sebagai berikut:

1. Buat instans basis data RDS Custom for SQL Server dari versi mesin yang ditawarkan oleh RDS Custom.

   Lihat informasi yang lebih lengkap di Membuat instans DB untuk RDS Custom for SQL Server.
2. Hubungkan aplikasi Anda dengan titik akhir instans basis data RDS Custom.

   Lihat informasi yang lebih lengkap di Menghubungkan ke instans RDS Custom DB Anda menggunakan AWS Systems Manager dan Menghubungkan ke instans RDS Custom DB Anda menggunakan RDP.
3. (Opsional) Akses host untuk menyesuaikan perangkat lunak Anda.
4. Pantau notifikasi dan pesan yang dihasilkan oleh automasi RDS Custom.

Membuat instans basis data untuk RDS Custom

Anda membuat instans basis data RDS Custom dengan menggunakan perintah `create-db-instance`. Prosedurnya mirip dengan membuat instans Amazon RDS. Namun, beberapa parameternya berbeda. Lihat informasi yang lebih lengkap di Membuat dan menghubungkan ke instans DB untuk Amazon RDS Custom for SQL Server.

Koneksi basis data

Seperti instans basis data Amazon RDS, instans basis data RDS Custom for SQL Server berada di VPC. Aplikasi Anda menghubungi instans RDS Custom dengan menggunakan klien seperti SQL Server Management Suite (SSMS), persis seperti di RDS for SQL Server.

Kustomisasi RDS Custom

Anda dapat mengakses host RDS Custom untuk menginstal atau menyesuaikan perangkat lunak. Untuk menghindari konflik antara perubahan Anda dan automasi RDS Custom, Anda dapat menjeda automasi selama jangka waktu tertentu. Selama periode ini, RDS Custom tidak melakukan pemantauan atau pemulihan instans. Pada akhir periode, RDS Custom melanjutkan automasi penuh. Lihat informasi yang lebih lengkap di Menjeda dan melanjutkan otomatisasi RDS Custom.

Persyaratan dan batasan untuk Amazon RDS Custom for SQL Server

Berikut ini, Anda dapat menemukan ringkasan persyaratan dan batasan Amazon RDS Custom for SQL Server untuk referensi cepat. Persyaratan dan batasan juga muncul di bagian yang relevan.

Topik

Ketersediaan wilayah dan versi

Ketersediaan dan dukungan fitur bervariasi di seluruh versi spesifik dari setiap mesin basis data, dan di seluruh Wilayah AWS. Untuk informasi selengkapnya tentang ketersediaan versi dan Wilayah Amazon RDS dengan Amazon RDS Custom for SQL Server, lihat RDS Custom for SQL Server.

Persyaratan umum untuk RDS Custom for SQL Server

Pastikan mengikuti persyaratan berikut untuk Amazon RDS Custom for SQL Server:

- Gunakan kelas instans yang ditunjukkan di Dukungan kelas instans DB untuk RDS Custom for SQL Server. Satu-satunya jenis penyimpanan yang didukung adalah solid state drive (SSD) tipe gp2, gp3, io1, dan io2 Block Express. Batas penyimpanan maksimum adalah 16 TiB.
- Pastikan Anda memiliki AWS KMS kunci enkripsi simetris untuk membuat instance RDS Custom DB. Untuk informasi selengkapnya, lihat Pastikan Anda memiliki kunci enkripsi simetris AWS KMS.
- Pastikan Anda membuat peran AWS Identity and Access Management (IAM) dan profil instance. Untuk informasi selengkapnya, lihat Membuat profil instans dan peran IAM Anda secara manual.
- Pastikan untuk menyediakan konfigurasi jaringan yang dapat digunakan RDS Custom untuk mengakses lainnya Layanan AWS. Untuk persyaratan khusus, lihat Konfigurasikan jaringan, profil instans, dan enkripsi.

- Jumlah gabungan instans DB RDS Custom dan Amazon RDS tidak boleh melebihi batas kuota Anda. Misalnya, jika kuota adalah 40 instans DB, Anda dapat memiliki 20 instans DB RDS Custom for SQL Server dan 20 instans DB Amazon RDS.

Dukungan kelas instans DB untuk RDS Custom for SQL Server

Periksa apakah kelas instans DB didukung di Wilayah Anda dengan menggunakan perintah describe-orderable-db-instance-options.

RDS Kustom untuk SQL Server mendukung kelas instans DB yang ditunjukkan pada tabel berikut:

Edisi SQL Server	Dukungan RDS Custom
Enterprise Edition	db.r5.xlarge–db.r5.24xlarge db.r5b.xlarge—db.r5b.24xlarge db.m5.xlarge–db.m5.24xlarge db.r6i.xlarge–db.r6i.32xlarge db.m6i.xbesar—db.m6i.32xlarge db.x2iedn.xlarge–db.x2iedn.32xlarge
Standard Edition	db.r5.large–db.r5.24xlarge db.r5b.besar—db.r5b.8xlarge db.m5.large–db.m5.24xlarge db.r6i.large–db.r6i.8xlarge db.m6i.large–db.m6i.8xlarge db.x2iedn.xlarge—db.x2iedn.8xlarge
Developer Edition	db.r5.xlarge–db.r5.24xlarge

Edisi SQL Server	Dukungan RDS Custom
	db.r5b.xlarge—db.r5b.24xlarge db.m5.xlarge–db.m5.24xlarge db.r6i.xlarge–db.r6i.32xlarge db.m6i.xbesar—db.m6i.32xlarge db.x2iedn.xlarge–db.x2iedn.32xlarge
Web Edition	db.r5.large–db.r5.4xlarge db.m5.large–db.m5.4xlarge db.r6i.large–db.r6i.4xlarge db.m6i.besar—db.m6i.4xlarge db.r5b.large–db.r5b.4xlarge

Rekomendasi berikut berlaku untuk tipe kelas db.x2iedn:

- Saat pembuatan, penyimpanan lokal adalah perangkat mentah dan tidak terisi. Sebelum menggunakan instance DB dengan kelas instance ini, Anda harus memasang dan memformat penyimpanan lokal. Setelah itu, `tempdb` konfigurasikan untuk memastikan kinerja yang optimal. Untuk informasi selengkapnya, lihat Mengoptimalkan performa tempdb di Amazon RDS Custom for SQL Server menggunakan penyimpanan instans lokal.
- Penyimpanan lokal kembali ke status mentah dan tidak dialokasikan saat Anda menjalankan operasi instans DB seperti komputasi skala, penggantian instance, pemulihan snapshot, atau point-in-time pemulihan (PITR). Dalam situasi ini, Anda harus memasang kembali, memformat ulang, dan mengkonfigurasi ulang drive dan `tempdb` mengembalikan fungsionalitas.
- Untuk instans multi-AZ, sebaiknya Anda melakukan konfigurasi pada instans DB siaga. Dengan cara ini, jika failover terjadi, sistem terus beroperasi tanpa masalah karena konfigurasi sudah ada pada instance siaga.

Batasan untuk RDS Custom for SQL Server

Batasan berikut berlaku untuk RDS Custom for SQL Server:

- Anda tidak dapat membuat replika baca di Amazon RDS untuk instans DB RDS Custom for SQL Server. Namun, Anda dapat mengonfigurasi ketersediaan tinggi secara otomatis dengan deployment Multi-AZ. Untuk informasi selengkapnya, lihat Mengelola deployment Multi-AZ untuk RDS Custom for SQL Server.
- Untuk instans DB RDS Custom for SQL Server yang tidak dibuat dengan versi mesin kustom (CEV), perubahan pada sistem operasi Microsoft Windows atau drive C: tidak dijamin akan tetap ada. Misalnya, Anda akan kehilangan perubahan tersebut saat menskalakan komputasi atau memulai operasi pemulihan snapshot. Jika instans DB RDS Custom for SQL Server dibuat dengan CEV, perubahan tersebut tetap ada.
- Tidak semua opsi didukung. Misalnya, saat membuat instans DB RDS Custom for SQL Server, Anda tidak dapat melakukan hal berikut:
  - Mengubah jumlah inti dan thread per inti CPU pada kelas instans DB.
  - Mengaktifkan penskalaan otomatis penyimpanan.
  - Konfigurasikan autentikasi Kerberos menggunakan AWS Management Console. Namun, Anda dapat mengonfigurasi Autentikasi Windows secara manual dan menggunakan Kerberos.
  - Tentukan grup parameter DB, grup opsi, atau set karakter Anda sendiri.
  - Aktifkan Wawasan Performa.
  - Aktifkan peningkatan versi minor otomatis.
- Penyimpanan instans DB maksimum adalah 16 TiB.

Dukungan kolasi dan karakter untuk instans DB RDS Custom for SQL Server

RDS Custom for SQL Server mendukung berbagai kolasi server, baik dalam enkode tradisional maupun UTF-8, untuk lokal SQL_Latin, Jepang, Jerman, dan Arab. Kolasi server default adalah `SQL_Latin1_General_CP1_CI_AS`, tetapi Anda dapat memilih menggunakan kolasi lain yang didukung. Anda dapat memilih kolasi menggunakan prosedur yang sama dengan RDS for SQL Server. Untuk informasi selengkapnya, lihat Kolasi dan set karakter untuk Microsoft SQL Server.

Persyaratan dan batasan berikut berlaku saat menggunakan kolasi server di RDS Custom for SQL Server:

- Anda dapat mengatur kolasi server saat membuat instans DB RDS Custom for SQL Server. Anda tidak dapat mengubah kolasi tingkat server setelah instans DB dibuat.
- Anda tidak dapat mengubah kolasi tingkat server saat melakukan pemulihan dari snapshot DB atau selama pemulihan titik waktu (PITR).
- Saat Anda membuat instans DB dari RDS Custom for SQL Server CEV, instans DB tidak mewarisi kolasi server dari CEV. Sebaliknya, kolasi server default `SQL_Latin1_General_CP1_CI_AS` digunakan. Jika Anda telah mengonfigurasi kolasi server non-default pada RDS Custom for SQL Server CEV dan ingin menggunakan kolasi server yang sama pada instans DB baru, pastikan untuk memilih kolasi yang sama saat membuat instans DB dari CEV.

  > **Note**
  >
  > Jika kolasi yang Anda pilih saat membuat instans DB berbeda dari kolasi CEV, basis data sistem Microsoft SQL Server pada instans DB RDS Custom for SQL Server baru akan dibuat kembali untuk menggunakan kolasi yang diperbarui. Proses pembuatan kembali hanya dilakukan pada instans DB RDS Custom for SQL Server baru dan tidak berdampak pada CEV itu sendiri. Setiap modifikasi sebelumnya yang Anda buat untuk basis data sistem pada CEV tidak akan dipertahankan pada instans DB RDS Custom for SQL Server baru setelah basis data sistem dibuat kembali. Contoh beberapa modifikasi termasuk objek yang ditentukan pengguna dalam basis data `master`, pekerjaan terjadwal dalam basis data `msdb`, atau perubahan pengaturan basis data default dalam basis data `model` pada CEV Anda. Anda dapat membuat ulang modifikasi secara manual setelah instans DB RDS Custom for SQL Server baru dibuat.

- Saat Anda membuat instans DB dari versi mesin kustom (CEV) RDS Custom for SQL Server dan memilih kolasi yang berbeda dari CEV, pastikan citra emas (AMI) Anda yang digunakan untuk pembuatan CEV memenuhi persyaratan berikut sehingga basis data sistem Microsoft SQL Server pada instans DB baru dapat dibuat kembali:
  - Untuk SQL Server 2022, pastikan `setup.exe` file berada di jalur berikut: `C:\Program Files \Microsoft SQL Server\160\Setup Bootstrap\SQL2022\setup.exe`
  - Untuk SQL Server 2019, pastikan file `setup.exe` berada di jalur berikut: `C:\Program Files \Microsoft SQL Server\150\Setup Bootstrap\SQL2019\setup.exe`
  - Salinan data dan templat log untuk basis data `master`, `model`, dan `msdb` harus ada di lokasi default. Untuk informasi selengkapnya, lihat Rebuild system databases di dokumentasi publik Microsoft.

  - Pastikan SQL Server Database Engine Anda menggunakan `NT Service\MSSQLSERVER` atau `NT AUTHORITY\NETWORK SERVICE` sebagai akun layanan. Akun lain tidak akan memiliki izin yang diperlukan pada drive `C:\` saat mengonfigurasi kolasi server non-default untuk instans DB.
- Jika kolasi server yang dipilih untuk instans DB baru sama dengan yang dikonfigurasi pada CEV, basis data sistem Microsoft SQL Server pada instans DB RDS Custom for SQL Server baru tidak menjalani proses pembuatan ulang. Setiap modifikasi sebelumnya yang Anda buat pada basis data sistem di CEV secara otomatis akan tetap ada di instans DB RDS Custom for SQL Server yang baru.

Anda dapat mengatur kolasi Anda ke salah satu nilai yang tercantum dalam tabel berikut ini.

Server Collation	Description
Arabic_100_BIN	Arabic-100, binary sort
Arabic_100_BIN2	Arabic-100, binary code point comparison sort
Arabic_100_CI_AI	Arabic-100, case-insensitive, accent-insensitive, kanatype
Arabic_100_CI_AI_KS	Arabic-100, case-insensitive, accent-insensitive, kanatype
Arabic_100_CI_AI_KS_SC	Arabic-100, case-insensitive, accent-insensitive, kanatype
Arabic_100_CI_AI_KS_SC_UTF8	Arabic-100, case-insensitive, accent-insensitive, kanatype
Arabic_100_CI_AI_KS_WS	Arabic-100, case-insensitive, accent-insensitive, kanatype
Arabic_100_CI_AI_KS_WS_SC	Arabic-100, case-insensitive, accent-insensitive, kanatype
Arabic_100_CI_AI_KS_WS_SC_UTF8	Arabic-100, case-insensitive, accent-insensitive, kanatype
Arabic_100_CI_AI_SC	Arabic-100, case-insensitive, accent-insensitive, kanatype
Arabic_100_CI_AI_SC_UTF8	Arabic-100, case-insensitive, accent-insensitive, kanatype
Arabic_100_CI_AI_WS	Arabic-100, case-insensitive, accent-insensitive, kanatype
Arabic_100_CI_AI_WS_SC	Arabic-100, case-insensitive, accent-insensitive, kanatype
Arabic_100_CI_AI_WS_SC_UTF8	Arabic-100, case-insensitive, accent-insensitive, kanatype

Arabic_100_CI_AS	Arabic-100, case-insensitive, accent-sensitive, kanatype-
Arabic_100_CI_AS_KS	Arabic-100, case-insensitive, accent-sensitive, kanatype-
Arabic_100_CI_AS_KS_SC	Arabic-100, case-insensitive, accent-sensitive, kanatype-
Arabic_100_CI_AS_KS_SC_UTF8	Arabic-100, case-insensitive, accent-sensitive, kanatype-
Arabic_100_CI_AS_KS_WS	Arabic-100, case-insensitive, accent-sensitive, kanatype-
Arabic_100_CI_AS_KS_WS_SC	Arabic-100, case-insensitive, accent-sensitive, kanatype-
Arabic_100_CI_AS_KS_WS_SC_UTF8	Arabic-100, case-insensitive, accent-sensitive, kanatype-
Arabic_100_CI_AS_SC	Arabic-100, case-insensitive, accent-sensitive, kanatype-
Arabic_100_CI_AS_SC_UTF8	Arabic-100, case-insensitive, accent-sensitive, kanatype-
Arabic_100_CI_AS_WS	Arabic-100, case-insensitive, accent-sensitive, kanatype-
Arabic_100_CI_AS_WS_SC	Arabic-100, case-insensitive, accent-sensitive, kanatype-
Arabic_100_CI_AS_WS_SC_UTF8	Arabic-100, case-insensitive, accent-sensitive, kanatype-
Arabic_100_CS_AI	Arabic-100, case-sensitive, accent-insensitive, kanatype-i
Arabic_100_CS_AI_KS	Arabic-100, case-sensitive, accent-insensitive, kanatype-s
Arabic_100_CS_AI_KS_SC	Arabic-100, case-sensitive, accent-insensitive, kanatype-s
Arabic_100_CS_AI_KS_SC_UTF8	Arabic-100, case-sensitive, accent-insensitive, kanatype-s
Arabic_100_CS_AI_KS_WS	Arabic-100, case-sensitive, accent-insensitive, kanatype-s
Arabic_100_CS_AI_KS_WS_SC	Arabic-100, case-sensitive, accent-insensitive, kanatype-s
Arabic_100_CS_AI_KS_WS_SC_UTF8	Arabic-100, case-sensitive, accent-insensitive, kanatype-s
Arabic_100_CS_AI_SC	Arabic-100, case-sensitive, accent-insensitive, kanatype-i
Arabic_100_CS_AI_SC_UTF8	Arabic-100, case-sensitive, accent-insensitive, kanatype-i
Arabic_100_CS_AI_WS	Arabic-100, case-sensitive, accent-insensitive, kanatype-i

Arabic_100_CS_AI_WS_SC	Arabic-100, case-sensitive, accent-insensitive, kanatype-i
Arabic_100_CS_AI_WS_SC_UTF8	Arabic-100, case-sensitive, accent-insensitive, kanatype-i
Arabic_100_CS_AS	Arabic-100, case-sensitive, accent-sensitive, kanatype-in
Arabic_100_CS_AS_KS	Arabic-100, case-sensitive, accent-sensitive, kanatype-se
Arabic_100_CS_AS_KS_SC	Arabic-100, case-sensitive, accent-sensitive, kanatype-se
Arabic_100_CS_AS_KS_SC_UTF8	Arabic-100, case-sensitive, accent-sensitive, kanatype-se
Arabic_100_CS_AS_KS_WS	Arabic-100, case-sensitive, accent-sensitive, kanatype-se
Arabic_100_CS_AS_KS_WS_SC	Arabic-100, case-sensitive, accent-sensitive, kanatype-se
Arabic_100_CS_AS_KS_WS_SC_UTF8	Arabic-100, case-sensitive, accent-sensitive, kanatype-se
Arabic_100_CS_AS_SC	Arabic-100, case-sensitive, accent-sensitive, kanatype-in
Arabic_100_CS_AS_SC_UTF8	Arabic-100, case-sensitive, accent-sensitive, kanatype-in
Arabic_100_CS_AS_WS	Arabic-100, case-sensitive, accent-sensitive, kanatype-in
Arabic_100_CS_AS_WS_SC	Arabic-100, case-sensitive, accent-sensitive, kanatype-in
Arabic_100_CS_AS_WS_SC_UTF8	Arabic-100, case-sensitive, accent-sensitive, kanatype-in
Arabic_BIN	Arabic, binary sort
Arabic_BIN2	Arabic, binary code point comparison sort
Arabic_CI_AI	Arabic, case-insensitive, accent-insensitive, kanatype-ins
Arabic_CI_AI_KS	Arabic, case-insensitive, accent-insensitive, kanatype-ser
Arabic_CI_AI_KS_WS	Arabic, case-insensitive, accent-insensitive, kanatype-ser
Arabic_CI_AI_WS	Arabic, case-insensitive, accent-insensitive, kanatype-ins
Arabic_CI_AS	Arabic, case-insensitive, accent-sensitive, kanatype-inse
Arabic_CI_AS_KS	Arabic, case-insensitive, accent-sensitive, kanatype-sens

Arabic_CI_AS_KS_WS	Arabic, case-insensitive, accent-sensitive, kanatype-sens
Arabic_CI_AS_WS	Arabic, case-insensitive, accent-sensitive, kanatype-inse
Arabic_CS_AI	Arabic, case-sensitive, accent-insensitive, kanatype-inser
Arabic_CS_AI_KS	Arabic, case-sensitive, accent-insensitive, kanatype-sens
Arabic_CS_AI_KS_WS	Arabic, case-sensitive, accent-insensitive, kanatype-sens
Arabic_CS_AI_WS	Arabic, case-sensitive, accent-insensitive, kanatype-inser
Arabic_CS_AS	Arabic, case-sensitive, accent-sensitive, kanatype-insens
Arabic_CS_AS_KS	Arabic, case-sensitive, accent-sensitive, kanatype-sensit
Arabic_CS_AS_KS_WS	Arabic, case-sensitive, accent-sensitive, kanatype-sensit
Arabic_CS_AS_WS	Arabic, case-sensitive, accent-sensitive, kanatype-insens
Chinese_PRC_BIN2	Chinese-PRC, binary code point comparison sort
Chinese_PRC_CI_AS	Chinese-PRC, case-insensitive, accent-sensitive, kanaty
Chinese_Taiwan_Stroke_CI_AS	Chinese-Taiwan-Stroke, case-insensitive, accent-sensitiv
Danish_Norwegian_CI_AS	Danish-Norwegian, case-insensitive, accent-sensitive, ka
Finnish_Swedish_CI_AS	Finnish-Swedish, case-insensitive, accent-sensitive, kana
French_CI_AS	French, case-insensitive, accent-sensitive, kanatype-inse
German_PhoneBook_100_BIN	German-PhoneBook-100, binary sort
German_PhoneBook_100_BIN2	German-PhoneBook-100, binary code point comparison s
German_PhoneBook_100_CI_AI	German-PhoneBook-100, case-insensitive, accent-insen
German_PhoneBook_100_CI_AI_KS	German-PhoneBook-100, case-insensitive, accent-insen
German_PhoneBook_100_CI_AI_KS_SC	German-PhoneBook-100, case-insensitive, accent-insen

German_PhoneBook_100_CI_AI_KS_SC_UTF8	German-PhoneBook-100, case-insensitive, accent-insen
German_PhoneBook_100_CI_AI_KS_WS	German-PhoneBook-100, case-insensitive, accent-insen
German_PhoneBook_100_CI_AI_KS_WS_SC	German-PhoneBook-100, case-insensitive, accent-insen
German_PhoneBook_100_CI_AI_KS_WS_SC_UTF8	German-PhoneBook-100, case-insensitive, accent-insen
German_PhoneBook_100_CI_AI_SC	German-PhoneBook-100, case-insensitive, accent-insen
German_PhoneBook_100_CI_AI_SC_UTF8	German-PhoneBook-100, case-insensitive, accent-insen
German_PhoneBook_100_CI_AI_WS	German-PhoneBook-100, case-insensitive, accent-insen
German_PhoneBook_100_CI_AI_WS_SC	German-PhoneBook-100, case-insensitive, accent-insen
German_PhoneBook_100_CI_AI_WS_SC_UTF8	German-PhoneBook-100, case-insensitive, accent-insen
German_PhoneBook_100_CI_AS	German-PhoneBook-100, case-insensitive, accent-sensit
German_PhoneBook_100_CI_AS_KS	German-PhoneBook-100, case-insensitive, accent-sensit
German_PhoneBook_100_CI_AS_KS_SC	German-PhoneBook-100, case-insensitive, accent-sensit
German_PhoneBook_100_CI_AS_KS_SC_UTF8	German-PhoneBook-100, case-insensitive, accent-sensit
German_PhoneBook_100_CI_AS_KS_WS	German-PhoneBook-100, case-insensitive, accent-sensit
German_PhoneBook_100_CI_AS_KS_WS_SC	German-PhoneBook-100, case-insensitive, accent-sensit
German_PhoneBook_100_CI_AS_KS_WS_SC_UTF8	German-PhoneBook-100, case-insensitive, accent-sensit
German_PhoneBook_100_CI_AS_SC	German-PhoneBook-100, case-insensitive, accent-sensit
German_PhoneBook_100_CI_AS_SC_UTF8	German-PhoneBook-100, case-insensitive, accent-sensit
German_PhoneBook_100_CI_AS_WS	German-PhoneBook-100, case-insensitive, accent-sensit

German_PhoneBook_100_CI_AS_WS_SC	German-PhoneBook-100, case-insensitive, accent-sensit
German_PhoneBook_100_CI_AS_WS_SC_UTF8	German-PhoneBook-100, case-insensitive, accent-sensit
German_PhoneBook_100_CS_AI	German-PhoneBook-100, case-sensitive, accent-insensit
German_PhoneBook_100_CS_AI_KS	German-PhoneBook-100, case-sensitive, accent-insensit
German_PhoneBook_100_CS_AI_KS_SC	German-PhoneBook-100, case-sensitive, accent-insensit
German_PhoneBook_100_CS_AI_KS_SC_UTF8	German-PhoneBook-100, case-sensitive, accent-insensit
German_PhoneBook_100_CS_AI_KS_WS	German-PhoneBook-100, case-sensitive, accent-insensit
German_PhoneBook_100_CS_AI_KS_WS_SC	German-PhoneBook-100, case-sensitive, accent-insensit
German_PhoneBook_100_CS_AI_KS_WS_SC_UTF8	German-PhoneBook-100, case-sensitive, accent-insensit
German_PhoneBook_100_CS_AI_SC	German-PhoneBook-100, case-sensitive, accent-insensit
German_PhoneBook_100_CS_AI_SC_UTF8	German-PhoneBook-100, case-sensitive, accent-insensit
German_PhoneBook_100_CS_AI_WS	German-PhoneBook-100, case-sensitive, accent-insensit
German_PhoneBook_100_CS_AI_WS_SC	German-PhoneBook-100, case-sensitive, accent-insensit
German_PhoneBook_100_CS_AI_WS_SC_UTF8	German-PhoneBook-100, case-sensitive, accent-insensit
German_PhoneBook_100_CS_AS	German-PhoneBook-100, case-sensitive, accent-sensitiv
German_PhoneBook_100_CS_AS_KS	German-PhoneBook-100, case-sensitive, accent-sensitiv
German_PhoneBook_100_CS_AS_KS_SC	German-PhoneBook-100, case-sensitive, accent-sensitiv
German_PhoneBook_100_CS_AS_KS_SC_UTF8	German-PhoneBook-100, case-sensitive, accent-sensitiv
German_PhoneBook_100_CS_AS_KS_WS	German-PhoneBook-100, case-sensitive, accent-sensitiv

German_PhoneBook_100_CS_AS_KS_WS_SC	German-PhoneBook-100, case-sensitive, accent-sensitiv
German_PhoneBook_100_CS_AS_KS_WS_SC_UTF8	German-PhoneBook-100, case-sensitive, accent-sensitiv
German_PhoneBook_100_CS_AS_SC	German-PhoneBook-100, case-sensitive, accent-sensitiv
German_PhoneBook_100_CS_AS_SC_UTF8	German-PhoneBook-100, case-sensitive, accent-sensitiv
German_PhoneBook_100_CS_AS_WS	German-PhoneBook-100, case-sensitive, accent-sensitiv
German_PhoneBook_100_CS_AS_WS_SC	German-PhoneBook-100, case-sensitive, accent-sensitiv
German_PhoneBook_100_CS_AS_WS_SC_UTF8	German-PhoneBook-100, case-sensitive, accent-sensitiv
German_PhoneBook_BIN	German-PhoneBook, binary sort
German_PhoneBook_BIN2	German-PhoneBook, binary code point comparison sort
German_PhoneBook_CI_AI	German-PhoneBook, case-insensitive, accent-insensitive
German_PhoneBook_CI_AI_KS	German-PhoneBook, case-insensitive, accent-insensitive
German_PhoneBook_CI_AI_KS_WS	German-PhoneBook, case-insensitive, accent-insensitive
German_PhoneBook_CI_AI_WS	German-PhoneBook, case-insensitive, accent-insensitive
German_PhoneBook_CI_AS	German-PhoneBook, case-insensitive, accent-sensitive,
German_PhoneBook_CI_AS_KS	German-PhoneBook, case-insensitive, accent-sensitive,
German_PhoneBook_CI_AS_KS_WS	German-PhoneBook, case-insensitive, accent-sensitive,
German_PhoneBook_CI_AS_WS	German-PhoneBook, case-insensitive, accent-sensitive,
German_PhoneBook_CS_AI	German-PhoneBook, case-sensitive, accent-insensitive,
German_PhoneBook_CS_AI_KS	German-PhoneBook, case-sensitive, accent-insensitive,
German_PhoneBook_CS_AI_KS_WS	German-PhoneBook, case-sensitive, accent-insensitive,

German_PhoneBook_CS_AI_WS	German-PhoneBook, case-sensitive, accent-insensitive,
German_PhoneBook_CS_AS	German-PhoneBook, case-sensitive, accent-sensitive, ka
German_PhoneBook_CS_AS_KS	German-PhoneBook, case-sensitive, accent-sensitive, ka
German_PhoneBook_CS_AS_KS_WS	German-PhoneBook, case-sensitive, accent-sensitive, ka
German_PhoneBook_CS_AS_WS	German-PhoneBook, case-sensitive, accent-sensitive, ka
Hebrew_BIN	Hebrew, binary sort
Hebrew_CI_AS	Hebrew, case-insensitive, accent-sensitive, kanatype-ins
Japanese_90_BIN	Japanese-90, binary sort
Japanese_90_BIN2	Japanese-90, binary code point comparison sort
Japanese_90_CI_AI	Japanese-90, case-insensitive, accent-insensitive, kanaty
Japanese_90_CI_AI_KS	Japanese-90, case-insensitive, accent-insensitive, kanaty
Japanese_90_CI_AI_KS_SC	Japanese-90, case-insensitive, accent-insensitive, kanaty
Japanese_90_CI_AI_KS_SC_UTF8	Japanese-90, case-insensitive, accent-insensitive, kanaty
Japanese_90_CI_AI_KS_WS	Japanese-90, case-insensitive, accent-insensitive, kanaty
Japanese_90_CI_AI_KS_WS_SC	Japanese-90, case-insensitive, accent-insensitive, kanaty
Japanese_90_CI_AI_KS_WS_SC_UTF8	Japanese-90, case-insensitive, accent-insensitive, kanaty
Japanese_90_CI_AI_SC	Japanese-90, case-insensitive, accent-insensitive, kanaty
Japanese_90_CI_AI_SC_UTF8	Japanese-90, case-insensitive, accent-insensitive, kanaty
Japanese_90_CI_AI_WS	Japanese-90, case-insensitive, accent-insensitive, kanaty
Japanese_90_CI_AI_WS_SC	Japanese-90, case-insensitive, accent-insensitive, kanaty
Japanese_90_CI_AI_WS_SC_UTF8	Japanese-90, case-insensitive, accent-insensitive, kanaty
Japanese_90_CI_AS	Japanese-90, case-insensitive, accent-sensitive, kanatyp

Japanese_90_CI_AS_KS	Japanese-90, case-insensitive, accent-sensitive, kanatyp
Japanese_90_CI_AS_KS_SC	Japanese-90, case-insensitive, accent-sensitive, kanatyp
Japanese_90_CI_AS_KS_SC_UTF8	Japanese-90, case-insensitive, accent-sensitive, kanatyp
Japanese_90_CI_AS_KS_WS	Japanese-90, case-insensitive, accent-sensitive, kanatyp
Japanese_90_CI_AS_KS_WS_SC	Japanese-90, case-insensitive, accent-sensitive, kanatyp
Japanese_90_CI_AS_KS_WS_SC_UTF8	Japanese-90, case-insensitive, accent-sensitive, kanatyp
Japanese_90_CI_AS_SC	Japanese-90, case-insensitive, accent-sensitive, kanatyp
Japanese_90_CI_AS_SC_UTF8	Japanese-90, case-insensitive, accent-sensitive, kanatyp
Japanese_90_CI_AS_WS	Japanese-90, case-insensitive, accent-sensitive, kanatyp
Japanese_90_CI_AS_WS_SC	Japanese-90, case-insensitive, accent-sensitive, kanatyp
Japanese_90_CI_AS_WS_SC_UTF8	Japanese-90, case-insensitive, accent-sensitive, kanatyp
Japanese_90_CS_AI	Japanese-90, case-sensitive, accent-insensitive, kanatype
Japanese_90_CS_AI_KS	Japanese-90, case-sensitive, accent-insensitive, kanatype
Japanese_90_CS_AI_KS_SC	Japanese-90, case-sensitive, accent-insensitive, kanatype
Japanese_90_CS_AI_KS_SC_UTF8	Japanese-90, case-sensitive, accent-insensitive, kanatype
Japanese_90_CS_AI_KS_WS	Japanese-90, case-sensitive, accent-insensitive, kanatype
Japanese_90_CS_AI_KS_WS_SC	Japanese-90, case-sensitive, accent-insensitive, kanatype
Japanese_90_CS_AI_KS_WS_SC_UTF8	Japanese-90, case-sensitive, accent-insensitive, kanatype
Japanese_90_CS_AI_SC	Japanese-90, case-sensitive, accent-insensitive, kanatype
Japanese_90_CS_AI_SC_UTF8	Japanese-90, case-sensitive, accent-insensitive, kanatype
Japanese_90_CS_AI_WS	Japanese-90, case-sensitive, accent-insensitive, kanatype
Japanese_90_CS_AI_WS_SC	Japanese-90, case-sensitive, accent-insensitive, kanatype

Japanese_90_CS_AI_WS_SC_UTF8	Japanese-90, case-sensitive, accent-insensitive, kanatyp
Japanese_90_CS_AS	Japanese-90, case-sensitive, accent-sensitive, kanatype
Japanese_90_CS_AS_KS	Japanese-90, case-sensitive, accent-sensitive, kanatype
Japanese_90_CS_AS_KS_SC	Japanese-90, case-sensitive, accent-sensitive, kanatype
Japanese_90_CS_AS_KS_SC_UTF8	Japanese-90, case-sensitive, accent-sensitive, kanatype
Japanese_90_CS_AS_KS_WS	Japanese-90, case-sensitive, accent-sensitive, kanatype
Japanese_90_CS_AS_KS_WS_SC	Japanese-90, case-sensitive, accent-sensitive, kanatype
Japanese_90_CS_AS_KS_WS_SC_UTF8	Japanese-90, case-sensitive, accent-sensitive, kanatype
Japanese_90_CS_AS_SC	Japanese-90, case-sensitive, accent-sensitive, kanatype
Japanese_90_CS_AS_SC_UTF8	Japanese-90, case-sensitive, accent-sensitive, kanatype
Japanese_90_CS_AS_WS	Japanese-90, case-sensitive, accent-sensitive, kanatype
Japanese_90_CS_AS_WS_SC	Japanese-90, case-sensitive, accent-sensitive, kanatype
Japanese_90_CS_AS_WS_SC_UTF8	Japanese-90, case-sensitive, accent-sensitive, kanatype
Japanese_BIN	Japanese, binary sort
Japanese_BIN2	Japanese, binary code point comparison sort
Japanese_Bushu_Kakusu_100_BIN	Japanese-Bushu-Kakusu-100, binary sort
Japanese_Bushu_Kakusu_100_BIN2	Japanese-Bushu-Kakusu-100, binary code point compari
Japanese_Bushu_Kakusu_100_CI_AI	Japanese-Bushu-Kakusu-100, case-insensitive, accent-i
Japanese_Bushu_Kakusu_100_CI_AI_KS	Japanese-Bushu-Kakusu-100, case-insensitive, accent-i
Japanese_Bushu_Kakusu_100_CI_AI_KS_SC	Japanese-Bushu-Kakusu-100, case-insensitive, accent-i
Japanese_Bushu_Kakusu_100_CI_AI_KS_SC_UTF8	Japanese-Bushu-Kakusu-100, case-insensitive, accent-i

Japanese_Bushu_Kakusu_100_CI_AI_KS_WS	Japanese-Bushu-Kakusu-100, case-insensitive, accent-i
Japanese_Bushu_Kakusu_100_CI_AI_KS_WS_SC	Japanese-Bushu-Kakusu-100, case-insensitive, accent-i
Japanese_Bushu_Kakusu_100_CI_AI_KS_WS_SC_UTF8	Japanese-Bushu-Kakusu-100, case-insensitive, accent-i
Japanese_Bushu_Kakusu_100_CI_AI_SC	Japanese-Bushu-Kakusu-100, case-insensitive, accent-i
Japanese_Bushu_Kakusu_100_CI_AI_SC_UTF8	Japanese-Bushu-Kakusu-100, case-insensitive, accent-i
Japanese_Bushu_Kakusu_100_CI_AI_WS	Japanese-Bushu-Kakusu-100, case-insensitive, accent-i
Japanese_Bushu_Kakusu_100_CI_AI_WS_SC	Japanese-Bushu-Kakusu-100, case-insensitive, accent-i
Japanese_Bushu_Kakusu_100_CI_AI_WS_SC_UTF8	Japanese-Bushu-Kakusu-100, case-insensitive, accent-i
Japanese_Bushu_Kakusu_100_CI_AS	Japanese-Bushu-Kakusu-100, case-insensitive, accent-s
Japanese_Bushu_Kakusu_100_CI_AS_KS	Japanese-Bushu-Kakusu-100, case-insensitive, accent-s
Japanese_Bushu_Kakusu_100_CI_AS_KS_SC	Japanese-Bushu-Kakusu-100, case-insensitive, accent-s
Japanese_Bushu_Kakusu_100_CI_AS_KS_SC_UTF8	Japanese-Bushu-Kakusu-100, case-insensitive, accent-s
Japanese_Bushu_Kakusu_100_CI_AS_KS_WS	Japanese-Bushu-Kakusu-100, case-insensitive, accent-s
Japanese_Bushu_Kakusu_100_CI_AS_KS_WS_SC	Japanese-Bushu-Kakusu-100, case-insensitive, accent-s
Japanese_Bushu_Kakusu_100_CI_AS_KS_WS_SC_UTF8	Japanese-Bushu-Kakusu-100, case-insensitive, accent-s
Japanese_Bushu_Kakusu_100_CI_AS_SC	Japanese-Bushu-Kakusu-100, case-insensitive, accent-s

Japanese_Bushu_Kakusu_100_CI_AS_SC_UTF8	Japanese-Bushu-Kakusu-100, case-insensitive, accent-s
Japanese_Bushu_Kakusu_100_CI_AS_WS	Japanese-Bushu-Kakusu-100, case-insensitive, accent-s
Japanese_Bushu_Kakusu_100_CI_AS_WS_SC	Japanese-Bushu-Kakusu-100, case-insensitive, accent-s
Japanese_Bushu_Kakusu_100_CI_AS_WS_SC_UTF8	Japanese-Bushu-Kakusu-100, case-insensitive, accent-s
Japanese_Bushu_Kakusu_100_CS_AI	Japanese-Bushu-Kakusu-100, case-sensitive, accent-ins
Japanese_Bushu_Kakusu_100_CS_AI_KS	Japanese-Bushu-Kakusu-100, case-sensitive, accent-ins
Japanese_Bushu_Kakusu_100_CS_AI_KS_SC	Japanese-Bushu-Kakusu-100, case-sensitive, accent-ins
Japanese_Bushu_Kakusu_100_CS_AI_KS_SC_UTF8	Japanese-Bushu-Kakusu-100, case-sensitive, accent-ins
Japanese_Bushu_Kakusu_100_CS_AI_KS_WS	Japanese-Bushu-Kakusu-100, case-sensitive, accent-ins
Japanese_Bushu_Kakusu_100_CS_AI_KS_WS_SC	Japanese-Bushu-Kakusu-100, case-sensitive, accent-ins
Japanese_Bushu_Kakusu_100_CS_AI_KS_WS_SC_UTF8	Japanese-Bushu-Kakusu-100, case-sensitive, accent-ins
Japanese_Bushu_Kakusu_100_CS_AI_SC	Japanese-Bushu-Kakusu-100, case-sensitive, accent-ins
Japanese_Bushu_Kakusu_100_CS_AI_SC_UTF8	Japanese-Bushu-Kakusu-100, case-sensitive, accent-ins
Japanese_Bushu_Kakusu_100_CS_AI_WS	Japanese-Bushu-Kakusu-100, case-sensitive, accent-ins
Japanese_Bushu_Kakusu_100_CS_AI_WS_SC	Japanese-Bushu-Kakusu-100, case-sensitive, accent-ins
Japanese_Bushu_Kakusu_100_CS_AI_WS_SC_UTF8	Japanese-Bushu-Kakusu-100, case-sensitive, accent-ins

Japanese_Bushu_Kakusu_100_CS_AS	Japanese-Bushu-Kakusu-100, case-sensitive, accent-se
Japanese_Bushu_Kakusu_100_CS_AS_KS	Japanese-Bushu-Kakusu-100, case-sensitive, accent-se
Japanese_Bushu_Kakusu_100_CS_AS_KS_SC	Japanese-Bushu-Kakusu-100, case-sensitive, accent-se
Japanese_Bushu_Kakusu_100_CS_AS_KS_SC_UTF8	Japanese-Bushu-Kakusu-100, case-sensitive, accent-se
Japanese_Bushu_Kakusu_100_CS_AS_KS_WS	Japanese-Bushu-Kakusu-100, case-sensitive, accent-se
Japanese_Bushu_Kakusu_100_CS_AS_KS_WS_SC	Japanese-Bushu-Kakusu-100, case-sensitive, accent-se
Japanese_Bushu_Kakusu_100_CS_AS_KS_WS_SC_UTF8	Japanese-Bushu-Kakusu-100, case-sensitive, accent-se
Japanese_Bushu_Kakusu_100_CS_AS_SC	Japanese-Bushu-Kakusu-100, case-sensitive, accent-se
Japanese_Bushu_Kakusu_100_CS_AS_SC_UTF8	Japanese-Bushu-Kakusu-100, case-sensitive, accent-se
Japanese_Bushu_Kakusu_100_CS_AS_WS	Japanese-Bushu-Kakusu-100, case-sensitive, accent-se
Japanese_Bushu_Kakusu_100_CS_AS_WS_SC	Japanese-Bushu-Kakusu-100, case-sensitive, accent-se
Japanese_Bushu_Kakusu_100_CS_AS_WS_SC_UTF8	Japanese-Bushu-Kakusu-100, case-sensitive, accent-se
Japanese_Bushu_Kakusu_140_BIN	Japanese-Bushu-Kakusu-140, binary sort
Japanese_Bushu_Kakusu_140_BIN2	Japanese-Bushu-Kakusu-140, binary code point compari
Japanese_Bushu_Kakusu_140_CI_AI	Japanese-Bushu-Kakusu-140, case-insensitive, accent-i insensitive
Japanese_Bushu_Kakusu_140_CI_AI_KS	Japanese-Bushu-Kakusu-140, case-insensitive, accent-i insensitive

Japanese_Bushu_Kakusu_140_CI_AI_KS_UTF8	Japanese-Bushu-Kakusu-140, case-insensitive, accent-i insensitive, UTF8
Japanese_Bushu_Kakusu_140_CI_AI_KS_VSS	Japanese-Bushu-Kakusu-140, case-insensitive, accent-i sensitive
Japanese_Bushu_Kakusu_140_CI_AI_KS_VSS_UTF8	Japanese-Bushu-Kakusu-140, case-insensitive, accent-i sensitive, UTF8
Japanese_Bushu_Kakusu_140_CI_AI_KS_WS	Japanese-Bushu-Kakusu-140, case-insensitive, accent-i insensitive
Japanese_Bushu_Kakusu_140_CI_AI_KS_WS_UTF8	Japanese-Bushu-Kakusu-140, case-insensitive, accent-i insensitive, UTF8
Japanese_Bushu_Kakusu_140_CI_AI_KS_WS_VSS	Japanese-Bushu-Kakusu-140, case-insensitive, accent-i sensitive
Japanese_Bushu_Kakusu_140_CI_AI_KS_WS_VSS_UTF8	Japanese-Bushu-Kakusu-140, case-insensitive, accent-i sensitive, UTF8
Japanese_Bushu_Kakusu_140_CI_AI_UTF8	Japanese-Bushu-Kakusu-140, case-insensitive, accent-i insensitive, UTF8
Japanese_Bushu_Kakusu_140_CI_AI_VSS	Japanese-Bushu-Kakusu-140, case-insensitive, accent-i sensitive
Japanese_Bushu_Kakusu_140_CI_AI_VSS_UTF8	Japanese-Bushu-Kakusu-140, case-insensitive, accent-i sensitive, UTF8
Japanese_Bushu_Kakusu_140_CI_AI_WS	Japanese-Bushu-Kakusu-140, case-insensitive, accent-i insensitive
Japanese_Bushu_Kakusu_140_CI_AI_WS_UTF8	Japanese-Bushu-Kakusu-140, case-insensitive, accent-i insensitive, UTF8
Japanese_Bushu_Kakusu_140_CI_AI_WS_VSS	Japanese-Bushu-Kakusu-140, case-insensitive, accent-i sensitive

Japanese_Bushu_Kakusu_140_CI_AI_WS_VSS_UTF8	Japanese-Bushu-Kakusu-140, case-insensitive, accent-i sensitive, UTF8
Japanese_Bushu_Kakusu_140_CI_AS	Japanese-Bushu-Kakusu-140, case-insensitive, accent-s insensitive
Japanese_Bushu_Kakusu_140_CI_AS_KS	Japanese-Bushu-Kakusu-140, case-insensitive, accent-s insensitive
Japanese_Bushu_Kakusu_140_CI_AS_KS_UTF8	Japanese-Bushu-Kakusu-140, case-insensitive, accent-s insensitive, UTF8
Japanese_Bushu_Kakusu_140_CI_AS_KS_VSS	Japanese-Bushu-Kakusu-140, case-insensitive, accent-s sensitive
Japanese_Bushu_Kakusu_140_CI_AS_KS_VSS_UTF8	Japanese-Bushu-Kakusu-140, case-insensitive, accent-s sensitive, UTF8
Japanese_Bushu_Kakusu_140_CI_AS_KS_WS	Japanese-Bushu-Kakusu-140, case-insensitive, accent-s insensitive
Japanese_Bushu_Kakusu_140_CI_AS_KS_WS_UTF8	Japanese-Bushu-Kakusu-140, case-insensitive, accent-s insensitive, UTF8
Japanese_Bushu_Kakusu_140_CI_AS_KS_WS_VSS	Japanese-Bushu-Kakusu-140, case-insensitive, accent-s sensitive
Japanese_Bushu_Kakusu_140_CI_AS_KS_WS_VSS_UTF8	Japanese-Bushu-Kakusu-140, case-insensitive, accent-s sensitive, UTF8
Japanese_Bushu_Kakusu_140_CI_AS_UTF8	Japanese-Bushu-Kakusu-140, case-insensitive, accent-s insensitive, UTF8
Japanese_Bushu_Kakusu_140_CI_AS_VSS	Japanese-Bushu-Kakusu-140, case-insensitive, accent-s sensitive
Japanese_Bushu_Kakusu_140_CI_AS_VSS_UTF8	Japanese-Bushu-Kakusu-140, case-insensitive, accent-s sensitive, UTF8

Japanese_Bushu_Kakusu_140_CI_AS_WS	Japanese-Bushu-Kakusu-140, case-insensitive, accent-s insensitive
Japanese_Bushu_Kakusu_140_CI_AS_WS_UTF8	Japanese-Bushu-Kakusu-140, case-insensitive, accent-s insensitive, UTF8
Japanese_Bushu_Kakusu_140_CI_AS_WS_VSS	Japanese-Bushu-Kakusu-140, case-insensitive, accent-s sensitive
Japanese_Bushu_Kakusu_140_CI_AS_WS_VSS_UTF8	Japanese-Bushu-Kakusu-140, case-insensitive, accent-s sensitive, UTF8
Japanese_Bushu_Kakusu_140_CS_AI	Japanese-Bushu-Kakusu-140, case-sensitive, accent-ins insensitive
Japanese_Bushu_Kakusu_140_CS_AI_KS	Japanese-Bushu-Kakusu-140, case-sensitive, accent-ins insensitive
Japanese_Bushu_Kakusu_140_CS_AI_KS_UTF8	Japanese-Bushu-Kakusu-140, case-sensitive, accent-ins insensitive, UTF8
Japanese_Bushu_Kakusu_140_CS_AI_KS_VSS	Japanese-Bushu-Kakusu-140, case-sensitive, accent-ins sensitive
Japanese_Bushu_Kakusu_140_CS_AI_KS_VSS_UTF8	Japanese-Bushu-Kakusu-140, case-sensitive, accent-ins sensitive, UTF8
Japanese_Bushu_Kakusu_140_CS_AI_KS_WS	Japanese-Bushu-Kakusu-140, case-sensitive, accent-ins insensitive
Japanese_Bushu_Kakusu_140_CS_AI_KS_WS_UTF8	Japanese-Bushu-Kakusu-140, case-sensitive, accent-ins insensitive, UTF8
Japanese_Bushu_Kakusu_140_CS_AI_KS_WS_VSS	Japanese-Bushu-Kakusu-140, case-sensitive, accent-ins sensitive
Japanese_Bushu_Kakusu_140_CS_AI_KS_WS_VSS_UTF8	Japanese-Bushu-Kakusu-140, case-sensitive, accent-ins sensitive, UTF8

Japanese_Bushu_Kakusu_140_CS_AI_UTF8	Japanese-Bushu-Kakusu-140, case-sensitive, accent-ins insensitive, UTF8
Japanese_Bushu_Kakusu_140_CS_AI_VSS	Japanese-Bushu-Kakusu-140, case-sensitive, accent-ins sensitive
Japanese_Bushu_Kakusu_140_CS_AI_VSS_UTF8	Japanese-Bushu-Kakusu-140, case-sensitive, accent-ins sensitive, UTF8
Japanese_Bushu_Kakusu_140_CS_AI_WS	Japanese-Bushu-Kakusu-140, case-sensitive, accent-ins insensitive
Japanese_Bushu_Kakusu_140_CS_AI_WS_UTF8	Japanese-Bushu-Kakusu-140, case-sensitive, accent-ins insensitive, UTF8
Japanese_Bushu_Kakusu_140_CS_AI_WS_VSS	Japanese-Bushu-Kakusu-140, case-sensitive, accent-ins sensitive
Japanese_Bushu_Kakusu_140_CS_AI_WS_VSS_UTF8	Japanese-Bushu-Kakusu-140, case-sensitive, accent-ins sensitive, UTF8
Japanese_Bushu_Kakusu_140_CS_AS	Japanese-Bushu-Kakusu-140, case-sensitive, accent-se insensitive
Japanese_Bushu_Kakusu_140_CS_AS_KS	Japanese-Bushu-Kakusu-140, case-sensitive, accent-se insensitive
Japanese_Bushu_Kakusu_140_CS_AS_KS_UTF8	Japanese-Bushu-Kakusu-140, case-sensitive, accent-se insensitive, UTF8
Japanese_Bushu_Kakusu_140_CS_AS_KS_VSS	Japanese-Bushu-Kakusu-140, case-sensitive, accent-se sensitive
Japanese_Bushu_Kakusu_140_CS_AS_KS_VSS_UTF8	Japanese-Bushu-Kakusu-140, case-sensitive, accent-se sensitive, UTF8
Japanese_Bushu_Kakusu_140_CS_AS_KS_WS	Japanese-Bushu-Kakusu-140, case-sensitive, accent-se insensitive

Japanese_Bushu_Kakusu_140_CS_AS_KS_WS_UTF8	Japanese-Bushu-Kakusu-140, case-sensitive, accent-sen insensitive, UTF8
Japanese_Bushu_Kakusu_140_CS_AS_KS_WS_VSS	Japanese-Bushu-Kakusu-140, case-sensitive, accent-sen sensitive
Japanese_Bushu_Kakusu_140_CS_AS_KS_WS_VSS_UTF8	Japanese-Bushu-Kakusu-140, case-sensitive, accent-sen sensitive, UTF8
Japanese_Bushu_Kakusu_140_CS_AS_UTF8	Japanese-Bushu-Kakusu-140, case-sensitive, accent-sen insensitive, UTF8
Japanese_Bushu_Kakusu_140_CS_AS_VSS	Japanese-Bushu-Kakusu-140, case-sensitive, accent-sen sensitive
Japanese_Bushu_Kakusu_140_CS_AS_VSS_UTF8	Japanese-Bushu-Kakusu-140, case-sensitive, accent-sen sensitive, UTF8
Japanese_Bushu_Kakusu_140_CS_AS_WS	Japanese-Bushu-Kakusu-140, case-sensitive, accent-sen insensitive
Japanese_Bushu_Kakusu_140_CS_AS_WS_UTF8	Japanese-Bushu-Kakusu-140, case-sensitive, accent-sen insensitive, UTF8
Japanese_Bushu_Kakusu_140_CS_AS_WS_VSS	Japanese-Bushu-Kakusu-140, case-sensitive, accent-sen sensitive
Japanese_Bushu_Kakusu_140_CS_AS_WS_VSS_UTF8	Japanese-Bushu-Kakusu-140, case-sensitive, accent-sen sensitive, UTF8
Japanese_CI_AI	Japanese, case-insensitive, accent-insensitive, kanatype-
Japanese_CI_AI_KS	Japanese, case-insensitive, accent-insensitive, kanatype-
Japanese_CI_AI_KS_WS	Japanese, case-insensitive, accent-insensitive, kanatype-
Japanese_CI_AI_WS	Japanese, case-insensitive, accent-insensitive, kanatype-
Japanese_CI_AS	Japanese, case-insensitive, accent-sensitive, kanatype-i
Japanese_CI_AS_KS	Japanese, case-insensitive, accent-sensitive, kanatype-s

Japanese_CI_AS_KS_WS	Japanese, case-insensitive, accent-sensitive, kanatype-s
Japanese_CI_AS_WS	Japanese, case-insensitive, accent-sensitive, kanatype-i
Japanese_CS_AI	Japanese, case-sensitive, accent-insensitive, kanatype-in
Japanese_CS_AI_KS	Japanese, case-sensitive, accent-insensitive, kanatype-s
Japanese_CS_AI_KS_WS	Japanese, case-sensitive, accent-insensitive, kanatype-s
Japanese_CS_AI_WS	Japanese, case-sensitive, accent-insensitive, kanatype-in
Japanese_CS_AS	Japanese, case-sensitive, accent-sensitive, kanatype-ins
Japanese_CS_AS_KS	Japanese, case-sensitive, accent-sensitive, kanatype-se
Japanese_CS_AS_KS_WS	Japanese, case-sensitive, accent-sensitive, kanatype-se
Japanese_CS_AS_WS	Japanese, case-sensitive, accent-sensitive, kanatype-ins
Japanese_Unicode_BIN	Japanese-Unicode, binary sort
Japanese_Unicode_BIN2	Japanese-Unicode, binary code point comparison sort
Japanese_Unicode_CI_AI	Japanese-Unicode, case-insensitive, accent-insensitive,
Japanese_Unicode_CI_AI_KS	Japanese-Unicode, case-insensitive, accent-insensitive,
Japanese_Unicode_CI_AI_KS_WS	Japanese-Unicode, case-insensitive, accent-insensitive,
Japanese_Unicode_CI_AI_WS	Japanese-Unicode, case-insensitive, accent-insensitive,
Japanese_Unicode_CI_AS	Japanese-Unicode, case-insensitive, accent-sensitive, ka
Japanese_Unicode_CI_AS_KS	Japanese-Unicode, case-insensitive, accent-sensitive, ka
Japanese_Unicode_CI_AS_KS_WS	Japanese-Unicode, case-insensitive, accent-sensitive, ka
Japanese_Unicode_CI_AS_WS	Japanese-Unicode, case-insensitive, accent-sensitive, ka
Japanese_Unicode_CS_AI	Japanese-Unicode, case-sensitive, accent-insensitive, ka
Japanese_Unicode_CS_AI_KS	Japanese-Unicode, case-sensitive, accent-insensitive, ka

Japanese_Unicode_CS_AI_KS_WS	Japanese-Unicode, case-sensitive, accent-insensitive, ka
Japanese_Unicode_CS_AI_WS	Japanese-Unicode, case-sensitive, accent-insensitive, ka
Japanese_Unicode_CS_AS	Japanese-Unicode, case-sensitive, accent-sensitive, kan
Japanese_Unicode_CS_AS_KS	Japanese-Unicode, case-sensitive, accent-sensitive, kan
Japanese_Unicode_CS_AS_KS_WS	Japanese-Unicode, case-sensitive, accent-sensitive, kan
Japanese_Unicode_CS_AS_WS	Japanese-Unicode, case-sensitive, accent-sensitive, kan
Japanese_XJIS_100_BIN	Japanese-XJIS-100, binary sort
Japanese_XJIS_100_BIN2	Japanese-XJIS-100, binary code point comparison sort
Japanese_XJIS_100_CI_AI	Japanese-XJIS-100, case-insensitive, accent-insensitive
Japanese_XJIS_100_CI_AI_KS	Japanese-XJIS-100, case-insensitive, accent-insensitive
Japanese_XJIS_100_CI_AI_KS_SC	Japanese-XJIS-100, case-insensitive, accent-insensitive
Japanese_XJIS_100_CI_AI_KS_SC_UTF8	Japanese-XJIS-100, case-insensitive, accent-insensitive
Japanese_XJIS_100_CI_AI_KS_WS	Japanese-XJIS-100, case-insensitive, accent-insensitive
Japanese_XJIS_100_CI_AI_KS_WS_SC	Japanese-XJIS-100, case-insensitive, accent-insensitive
Japanese_XJIS_100_CI_AI_KS_WS_SC_UTF8	Japanese-XJIS-100, case-insensitive, accent-insensitive
Japanese_XJIS_100_CI_AI_SC	Japanese-XJIS-100, case-insensitive, accent-insensitive
Japanese_XJIS_100_CI_AI_SC_UTF8	Japanese-XJIS-100, case-insensitive, accent-insensitive
Japanese_XJIS_100_CI_AI_WS	Japanese-XJIS-100, case-insensitive, accent-insensitive
Japanese_XJIS_100_CI_AI_WS_SC	Japanese-XJIS-100, case-insensitive, accent-insensitive
Japanese_XJIS_100_CI_AI_WS_SC_UTF8	Japanese-XJIS-100, case-insensitive, accent-insensitive
Japanese_XJIS_100_CI_AS	Japanese-XJIS-100, case-insensitive, accent-sensitive, k

Japanese_XJIS_100_CI_AS_KS	Japanese-XJIS-100, case-insensitive, accent-sensitive, k
Japanese_XJIS_100_CI_AS_KS_SC	Japanese-XJIS-100, case-insensitive, accent-sensitive, k
Japanese_XJIS_100_CI_AS_KS_SC_UTF8	Japanese-XJIS-100, case-insensitive, accent-sensitive, k
Japanese_XJIS_100_CI_AS_KS_WS	Japanese-XJIS-100, case-insensitive, accent-sensitive, k
Japanese_XJIS_100_CI_AS_KS_WS_SC	Japanese-XJIS-100, case-insensitive, accent-sensitive, k
Japanese_XJIS_100_CI_AS_KS_WS_SC_UTF8	Japanese-XJIS-100, case-insensitive, accent-sensitive, k
Japanese_XJIS_100_CI_AS_SC	Japanese-XJIS-100, case-insensitive, accent-sensitive, k
Japanese_XJIS_100_CI_AS_SC_UTF8	Japanese-XJIS-100, case-insensitive, accent-sensitive, k
Japanese_XJIS_100_CI_AS_WS	Japanese-XJIS-100, case-insensitive, accent-sensitive, k
Japanese_XJIS_100_CI_AS_WS_SC	Japanese-XJIS-100, case-insensitive, accent-sensitive, k
Japanese_XJIS_100_CI_AS_WS_SC_UTF8	Japanese-XJIS-100, case-insensitive, accent-sensitive, k
Japanese_XJIS_100_CS_AI	Japanese-XJIS-100, case-sensitive, accent-insensitive, k
Japanese_XJIS_100_CS_AI_KS	Japanese-XJIS-100, case-sensitive, accent-insensitive, k
Japanese_XJIS_100_CS_AI_KS_SC	Japanese-XJIS-100, case-sensitive, accent-insensitive, k
Japanese_XJIS_100_CS_AI_KS_SC_UTF8	Japanese-XJIS-100, case-sensitive, accent-insensitive, k
Japanese_XJIS_100_CS_AI_KS_WS	Japanese-XJIS-100, case-sensitive, accent-insensitive, k
Japanese_XJIS_100_CS_AI_KS_WS_SC	Japanese-XJIS-100, case-sensitive, accent-insensitive, k
Japanese_XJIS_100_CS_AI_KS_WS_SC_UTF8	Japanese-XJIS-100, case-sensitive, accent-insensitive, k
Japanese_XJIS_100_CS_AI_SC	Japanese-XJIS-100, case-sensitive, accent-insensitive, k
Japanese_XJIS_100_CS_AI_SC_UTF8	Japanese-XJIS-100, case-sensitive, accent-insensitive, k

Japanese_XJIS_100_CS_AI_WS	Japanese-XJIS-100, case-sensitive, accent-insensitive, k
Japanese_XJIS_100_CS_AI_WS_SC	Japanese-XJIS-100, case-sensitive, accent-insensitive, k
Japanese_XJIS_100_CS_AI_WS_SC_UTF8	Japanese-XJIS-100, case-sensitive, accent-insensitive, k
Japanese_XJIS_100_CS_AS	Japanese-XJIS-100, case-sensitive, accent-sensitive, ka
Japanese_XJIS_100_CS_AS_KS	Japanese-XJIS-100, case-sensitive, accent-sensitive, ka
Japanese_XJIS_100_CS_AS_KS_SC	Japanese-XJIS-100, case-sensitive, accent-sensitive, ka
Japanese_XJIS_100_CS_AS_KS_SC_UTF8	Japanese-XJIS-100, case-sensitive, accent-sensitive, ka
Japanese_XJIS_100_CS_AS_KS_WS	Japanese-XJIS-100, case-sensitive, accent-sensitive, ka
Japanese_XJIS_100_CS_AS_KS_WS_SC	Japanese-XJIS-100, case-sensitive, accent-sensitive, ka
Japanese_XJIS_100_CS_AS_KS_ WS_SC_UTF8	Japanese-XJIS-100, case-sensitive, accent-sensitive, ka
Japanese_XJIS_100_CS_AS_SC	Japanese-XJIS-100, case-sensitive, accent-sensitive, ka
Japanese_XJIS_100_CS_AS_SC_UTF8	Japanese-XJIS-100, case-sensitive, accent-sensitive, ka
Japanese_XJIS_100_CS_AS_WS	Japanese-XJIS-100, case-sensitive, accent-sensitive, ka
Japanese_XJIS_100_CS_AS_WS_SC	Japanese-XJIS-100, case-sensitive, accent-sensitive, ka
Japanese_XJIS_100_CS_AS_WS_SC_UTF8	Japanese-XJIS-100, case-sensitive, accent-sensitive, ka
Japanese_XJIS_140_BIN	Japanese-XJIS-140, binary sort
Japanese_XJIS_140_BIN2	Japanese-XJIS-140, binary code point comparison sort
Japanese_XJIS_140_CI_AI	Japanese-XJIS-140, case-insensitive, accent-insensitive ve
Japanese_XJIS_140_CI_AI_KS	Japanese-XJIS-140, case-insensitive, accent-insensitive
Japanese_XJIS_140_CI_AI_KS_UTF8	Japanese-XJIS-140, case-insensitive, accent-insensitive ve, UTF8

Japanese_XJIS_140_CI_AI_KS_VSS	Japanese-XJIS-140, case-insensitive, accent-insensitive
Japanese_XJIS_140_CI_AI_KS_VSS_UTF8	Japanese-XJIS-140, case-insensitive, accent-insensitive UTF8
Japanese_XJIS_140_CI_AI_KS_WS	Japanese-XJIS-140, case-insensitive, accent-insensitive
Japanese_XJIS_140_CI_AI_KS_WS_UTF8	Japanese-XJIS-140, case-insensitive, accent-insensitive UTF8
Japanese_XJIS_140_CI_AI_KS_WS_VSS	Japanese-XJIS-140, case-insensitive, accent-insensitive
Japanese_XJIS_140_CI_AI_KS_WS_VSS_UTF8	Japanese-XJIS-140, case-insensitive, accent-insensitive UTF8
Japanese_XJIS_140_CI_AI_UTF8	Japanese-XJIS-140, case-insensitive, accent-insensitive, UTF8
Japanese_XJIS_140_CI_AI_VSS	Japanese-XJIS-140, case-insensitive, accent-insensitive
Japanese_XJIS_140_CI_AI_VSS_UTF8	Japanese-XJIS-140, case-insensitive, accent-insensitive, UTF8
Japanese_XJIS_140_CI_AI_WS	Japanese-XJIS-140, case-insensitive, accent-insensitive
Japanese_XJIS_140_CI_AI_WS_UTF8	Japanese-XJIS-140, case-insensitive, accent-insensitive, UTF8
Japanese_XJIS_140_CI_AI_WS_VSS	Japanese-XJIS-140, case-insensitive, accent-insensitive
Japanese_XJIS_140_CI_AI_WS_VSS_UTF8	Japanese-XJIS-140, case-insensitive, accent-insensitive UTF8
Japanese_XJIS_140_CI_AS	Japanese-XJIS-140, case-insensitive, accent-sensitive, k
Japanese_XJIS_140_CI_AS_KS	Japanese-XJIS-140, case-insensitive, accent-sensitive, k
Japanese_XJIS_140_CI_AS_KS_UTF8	Japanese-XJIS-140, case-insensitive, accent-sensitive, k UTF8
Japanese_XJIS_140_CI_AS_KS_VSS	Japanese-XJIS-140, case-insensitive, accent-sensitive, k

Japanese_XJIS_140_CI_AS_KS_VSS_UTF8	Japanese-XJIS-140, case-insensitive, accent-sensitive, k UTF8
Japanese_XJIS_140_CI_AS_KS_WS	Japanese-XJIS-140, case-insensitive, accent-sensitive, k
Japanese_XJIS_140_CI_AS_KS_WS_UTF8	Japanese-XJIS-140, case-insensitive, accent-sensitive, k UTF8
Japanese_XJIS_140_CI_AS_KS_WS_VSS	Japanese-XJIS-140, case-insensitive, accent-sensitive, k
Japanese_XJIS_140_CI_AS_KS_WS_VSS_UTF8	Japanese-XJIS-140, case-insensitive, accent-sensitive, k UTF8
Japanese_XJIS_140_CI_AS_UTF8	Japanese-XJIS-140, case-insensitive, accent-sensitive, k ve, UTF8
Japanese_XJIS_140_CI_AS_VSS	Japanese-XJIS-140, case-insensitive, accent-sensitive, k
Japanese_XJIS_140_CI_AS_VSS_UTF8	Japanese-XJIS-140, case-insensitive, accent-sensitive, k UTF8
Japanese_XJIS_140_CI_AS_WS	Japanese-XJIS-140, case-insensitive, accent-sensitive, k
Japanese_XJIS_140_CI_AS_WS_UTF8	Japanese-XJIS-140, case-insensitive, accent-sensitive, k UTF8
Japanese_XJIS_140_CI_AS_WS_VSS	Japanese-XJIS-140, case-insensitive, accent-sensitive, k
Japanese_XJIS_140_CI_AS_WS_VSS_UTF8	Japanese-XJIS-140, case-insensitive, accent-sensitive, k UTF8
Japanese_XJIS_140_CS_AI	Japanese-XJIS-140, case-sensitive, accent-insensitive, k
Japanese_XJIS_140_CS_AI_KS	Japanese-XJIS-140, case-sensitive, accent-insensitive, k
Japanese_XJIS_140_CS_AI_KS_UTF8	Japanese-XJIS-140, case-sensitive, accent-insensitive, k UTF8
Japanese_XJIS_140_CS_AI_KS_VSS	Japanese-XJIS-140, case-sensitive, accent-insensitive, k

Japanese_XJIS_140_CS_AI_KS_VSS_UTF8	Japanese-XJIS-140, case-sensitive, accent-insensitive, k UTF8
Japanese_XJIS_140_CS_AI_KS_WS	Japanese-XJIS-140, case-sensitive, accent-insensitive, k
Japanese_XJIS_140_CS_AI_KS_WS_UTF8	Japanese-XJIS-140, case-sensitive, accent-insensitive, k UTF8
Japanese_XJIS_140_CS_AI_KS_WS_VSS	Japanese-XJIS-140, case-sensitive, accent-insensitive, k
Japanese_XJIS_140_CS_AI_KS_WS_VSS_UTF8	Japanese-XJIS-140, case-sensitive, accent-insensitive, k UTF8
Japanese_XJIS_140_CS_AI_UTF8	Japanese-XJIS-140, case-sensitive, accent-insensitive, k ve, UTF8
Japanese_XJIS_140_CS_AI_VSS	Japanese-XJIS-140, case-sensitive, accent-insensitive, k
Japanese_XJIS_140_CS_AI_VSS_UTF8	Japanese-XJIS-140, case-sensitive, accent-insensitive, k UTF8
Japanese_XJIS_140_CS_AI_WS	Japanese-XJIS-140, case-sensitive, accent-insensitive, k
Japanese_XJIS_140_CS_AI_WS_UTF8	Japanese-XJIS-140, case-sensitive, accent-insensitive, k UTF8
Japanese_XJIS_140_CS_AI_WS_VSS	Japanese-XJIS-140, case-sensitive, accent-insensitive, k
Japanese_XJIS_140_CS_AI_WS_VSS_UTF8	Japanese-XJIS-140, case-sensitive, accent-insensitive, k UTF8
Japanese_XJIS_140_CS_AS	Japanese-XJIS-140, case-sensitive, accent-sensitive, ka
Japanese_XJIS_140_CS_AS_KS	Japanese-XJIS-140, case-sensitive, accent-sensitive, ka
Japanese_XJIS_140_CS_AS_KS_UTF8	Japanese-XJIS-140, case-sensitive, accent-sensitive, ka UTF8
Japanese_XJIS_140_CS_AS_KS_VSS	Japanese-XJIS-140, case-sensitive, accent-sensitive, ka

Japanese_XJIS_140_CS_AS_KS_VSS_UTF8	Japanese-XJIS-140, case-sensitive, accent-sensitive, ka UTF8
Japanese_XJIS_140_CS_AS_KS_WS	Japanese-XJIS-140, case-sensitive, accent-sensitive, ka
Japanese_XJIS_140_CS_AS_KS_WS_UTF8	Japanese-XJIS-140, case-sensitive, accent-sensitive, ka UTF8
Japanese_XJIS_140_CS_AS_KS_WS_VSS	Japanese-XJIS-140, case-sensitive, accent-sensitive, ka
Japanese_XJIS_140_CS_AS_KS_ WS_VSS_UTF8	Japanese-XJIS-140, case-sensitive, accent-sensitive, ka
Japanese_XJIS_140_CS_AS_UTF8	Japanese-XJIS-140, case-sensitive, accent-sensitive, ka UTF8
Japanese_XJIS_140_CS_AS_VSS	Japanese-XJIS-140, case-sensitive, accent-sensitive, ka
Japanese_XJIS_140_CS_AS_VSS_UTF8	Japanese-XJIS-140, case-sensitive, accent-sensitive, ka UTF8
Japanese_XJIS_140_CS_AS_WS	Japanese-XJIS-140, case-sensitive, accent-sensitive, ka
Japanese_XJIS_140_CS_AS_WS_UTF8	Japanese-XJIS-140, case-sensitive, accent-sensitive, ka UTF8
Japanese_XJIS_140_CS_AS_WS_VSS	Japanese-XJIS-140, case-sensitive, accent-sensitive, ka
Japanese_XJIS_140_CS_AS_WS_VSS_UTF8	Japanese-XJIS-140, case-sensitive, accent-sensitive, ka UTF8
Korean_Wansung_CI_AS	Korean-Wansung, case-insensitive, accent-sensitive, kar
Latin1_General_100_BIN	Latin1-General-100, binary sort
Latin1_General_100_BIN2	Latin1-General-100, binary code point comparison sort
Latin1_General_100_BIN2_UTF8	Latin1-General-100, binary code point comparison sort, U
Latin1_General_100_CI_AS	Latin1-General-100, case-insensitive, accent-sensitive, k

Latin1_General_100_CI_AS_SC_UTF8	Latin1-General-100, case-insensitive, accent-sensitive, k
Latin1_General_BIN	Latin1-General, binary sort
Latin1_General_BIN2	Latin1-General, binary code point comparison sort
Latin1_General_CI_AI	Latin1-General, case-insensitive, accent-insensitive, kan
Latin1_General_CI_AS	Latin1-General, case-insensitive, accent-sensitive, kanat
Latin1_General_CI_AS_KS	Latin1-General, case-insensitive, accent-sensitive, kanat
Latin1_General_CS_AS	Latin1-General, case-sensitive, accent-sensitive, kanatyp
Modern_Spanish_CI_AS	Modern-Spanish, case-insensitive, accent-sensitive, kan
SQL_1xCompat_CP850_CI_AS	Latin1-General, case-insensitive, accent-sensitive, kanat 850 for non-Unicode Data
SQL_Latin1_General_CP1_CI_AI	Latin1-General, case-insensitive, accent-insensitive, kan 1252 for non-Unicode Data
SQL_Latin1_General_CP1_CI_AS	Latin1-General, case-insensitive, accent-sensitive, kanat 1252 for non-Unicode Data
SQL_Latin1_General_CP1_CS_AS	Latin1-General, case-sensitive, accent-sensitive, kanatyp 1252 for non-Unicode Data
SQL_Latin1_General_CP1250_CI_AS	Latin1-General, case-insensitive, accent-sensitive, kanat 1250 for non-Unicode Data
SQL_Latin1_General_CP1250_CS_AS	Latin1-General, case-sensitive, accent-sensitive, kanatyp 1250 for non-Unicode Data
SQL_Latin1_General_CP1251_CI_AS	Latin1-General, case-insensitive, accent-sensitive, kanat 1251 for non-Unicode Data
SQL_Latin1_General_CP1251_CS_AS	Latin1-General, case-sensitive, accent-sensitive, kanatyp 1251 for non-Unicode Data

SQL_Latin1_General_CP1253_CI_AI	Latin1-General, case-insensitive, accent-insensitive, kana Page 1253 for non-Unicode Data
SQL_Latin1_General_CP1253_CI_AS	Latin1-General, case-insensitive, accent-sensitive, kanat 1253 for non-Unicode Data
SQL_Latin1_General_CP1253_CS_AS	Latin1-General, case-sensitive, accent-sensitive, kanatyp 1253 for non-Unicode Data
SQL_Latin1_General_CP1254_CI_AS	Turkish, case-insensitive, accent-sensitive, kanatype-inse for non-Unicode Data
SQL_Latin1_General_CP1254_CS_AS	Turkish, case-sensitive, accent-sensitive, kanatype-insen non-Unicode Data
SQL_Latin1_General_CP1255_CI_AS	Latin1-General, case-insensitive, accent-sensitive, kanat 1255 for non-Unicode Data
SQL_Latin1_General_CP1255_CS_AS	Latin1-General, case-sensitive, accent-sensitive, kanatyp 1255 for non-Unicode Data
SQL_Latin1_General_CP1256_CI_AS	Latin1-General, case-insensitive, accent-sensitive, kanat 1256 for non-Unicode Data
SQL_Latin1_General_CP1256_CS_AS	Latin1-General, case-sensitive, accent-sensitive, kanatyp 1256 for non-Unicode Data
SQL_Latin1_General_CP1257_CI_AS	Latin1-General, case-insensitive, accent-sensitive, kanat 1257 for non-Unicode Data
SQL_Latin1_General_CP1257_CS_AS	Latin1-General, case-sensitive, accent-sensitive, kanatyp 1257 for non-Unicode Data
SQL_Latin1_General_CP437_BIN	Latin1-General, binary sort for Unicode Data, SQL Serve
SQL_Latin1_General_CP437_BIN2	Latin1-General, binary code point comparison sort for Un
SQL_Latin1_General_CP437_CI_AI	Latin1-General, case-insensitive, accent-insensitive, kana 437 for non-Unicode Data

SQL_Latin1_General_CP437_CI_AS	Latin1-General, case-insensitive, accent-sensitive, kanat 437 for non-Unicode Data
SQL_Latin1_General_CP437_CS_AS	Latin1-General, case-sensitive, accent-sensitive, kanatyp 437 for non-Unicode Data
SQL_Latin1_General_CP850_BIN	Latin1-General, binary sort for Unicode Data, SQL Serve
SQL_Latin1_General_CP850_BIN2	Latin1-General, binary code point comparison sort for Un
SQL_Latin1_General_CP850_CI_AI	Latin1-General, case-insensitive, accent-insensitive, kan 850 for non-Unicode Data
SQL_Latin1_General_CP850_CI_AS	Latin1-General, case-insensitive, accent-sensitive, kanat 850 for non-Unicode Data
SQL_Latin1_General_CP850_CS_AS	Latin1-General, case-sensitive, accent-sensitive, kanatyp 850 for non-Unicode Data
SQL_Latin1_General_Pref_CP1_CI_AS	Latin1-General, case-insensitive, accent-sensitive, kanat 1252 for non-Unicode Data
SQL_Latin1_General_Pref_CP437_CI_AS	Latin1-General, case-insensitive, accent-sensitive, kanat 437 for non-Unicode Data
SQL_Latin1_General_Pref_CP850_CI_AS	Latin1-General, case-insensitive, accent-sensitive, kanat 850 for non-Unicode Data
Thai_CI_AS	Thai, case-insensitive, accent-sensitive, kanatype-insens

Zona waktu lokal untuk instans DB RDS Custom for SQL Server

Zona waktu instans DB RDS Custom for SQL Server diatur secara default. Nilai default saat ini adalah Waktu Universal Terkoordinasi (UTC). Anda dapat mengatur zona waktu instans DB ke zona waktu lokal agar sesuai dengan zona waktu aplikasi Anda.

Anda akan menetapkan zona waktu saat pertama kali membuat instans DB. Anda dapat membuat instans DB dengan menggunakan AWS Management Console, tindakan CreatedBInstance Amazon RDS API, atau perintah. AWS CLI create-db-instance

Jika instans DB Anda adalah bagian dari deployment Multi-AZ, ketika Anda mengalami failover, zona waktu lokal yang Anda tetapkan tidak akan berubah.

Saat Anda meminta point-in-time pemulihan, Anda menentukan waktu untuk memulihkannya. Waktu ditampilkan dalam zona waktu lokal Anda. Untuk informasi selengkapnya, lihat Memulihkan instans DB dengan waktu yang ditentukan.

Berikut ini adalah batasan untuk menetapkan zona waktu lokal pada instans DB:

- Anda dapat mengonfigurasi zona waktu untuk instans DB selama pembuatan instans, tetapi Anda tidak dapat mengubah zona waktu instans DB RDS Custom for SQL Server yang ada.
- Jika zona waktu instans DB RDS Custom for SQL Server yang ada dimodifikasi, RDS Custom mengubah status instans DB menjadi `unsupported-configuration` dan mengirimkan pemberitahuan peristiwa.
- Anda tidak dapat memulihkan snapshot dari instans DB dalam satu zona waktu ke instans DB dalam zona waktu yang berbeda.
- Kami sangat menyarankan agar Anda tidak memulihkan file cadangan dari satu zona waktu ke zona waktu yang berbeda. Jika memulihkan file cadangan dari satu zona waktu ke zona waktu yang berbeda, Anda harus mengaudit kueri dan aplikasi Anda untuk mengetahui efek dari perubahan zona waktu. Untuk informasi selengkapnya, lihat Mengimpor dan mengekspor basis data SQL Server menggunakan pencadangan dan pemulihan native.

Zona waktu yang didukung

Anda dapat mengatur zona waktu lokal Anda ke salah satu nilai yang tercantum dalam tabel berikut ini.

Zona waktu yang didukung untuk RDS Custom for SQL Server

Zona waktu	Offset waktu standar	Deskripsi	Catatan
Waktu Standar Afghanistan	(UTC+04.30)	Kabul	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Alaska	(UTC–09.00)	Alaska	

Zona waktu	Offset waktu standar	Deskripsi	Catatan
Waktu Standar Aleutian	(UTC–10.00)	Kepulauan Aleutian	
Waktu Standar Altai	(UTC+07.00)	Barnaul, Gorno-Altaysk	
Waktu Standar Arab	(UTC+03.00)	Kuwait, Riyadh	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Arab	(UTC+04.00)	Abu Dhabi, Muscat	
Waktu Standar Arab	(UTC+03.00)	Bagdad	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Argentina	(UTC–03.00)	Kota Buenos Aires	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Astrakhan	(UTC+04.00)	Astrakhan, Ulyanovsk	
Waktu Standar Atlantik	(UTC–04.00)	Waktu Atlantik (Kanada)	
Waktu Standar Tengah AUS	(UTC+09.30)	Darwin	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Barat Tengah Aus	(UTC+08.45)	Eucla	

Zona waktu	Offset waktu standar	Deskripsi	Catatan
Waktu Standar Timur AUS	(UTC+10.00)	Canberra, Melbourne, Sydney	
Waktu Standar Azerbaijan	(UTC+04.00)	Baku	
Waktu Standar Azores	(UTC–01.00)	Azores	
Waktu Standar Bahia	(UTC–03.00)	Salvador	
Waktu Standar Bangladesh	(UTC+06.00)	Dhaka	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Belarusia	(UTC+03.00)	Minsk	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Bougainville	(UTC+11.00)	Pulau Bougainville	
Waktu Standar Kanada Pusat	(UTC–06.00)	Saskatchewan	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Cape Verde	(UTC–01.00)	Cabo Verde Is.	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Kaukasus	(UTC+04.00)	Yerevan	
Cen. Waktu Standar Australia	(UTC+09.30)	Adelaide	

Zona waktu	Offset waktu standar	Deskripsi	Catatan
Waktu Standar Amerika Tengah	(UTC–06.00)	Amerika Tengah	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Asia Tengah	(UTC+06.00)	Astana	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Brasil Tengah	(UTC–04.00)	Cuiaba	
Waktu Standar Eropa Tengah	(UTC+01.00)	Belgrade, Bratislava, Budapest, Ljubljana, Praha	
Waktu Standar Eropa Tengah	(UTC+01.00)	Sarajevo, Skopje, Warsawa, Zagreb	
Waktu Standar Pasifik Tengah	(UTC+11.00)	Kepulauan Solomon, Kaledonia Baru	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Tengah	(UTC–06.00)	Waktu Tengah (AS dan Kanada)	
Waktu Standar Tengah (Meksiko)	(UTC–06.00)	Guadalajara, Mexico City, Monterrey	
Waktu Standar Kepulauan Chatham	(UTC+12.45)	Kepulauan Chatham	

Zona waktu	Offset waktu standar	Deskripsi	Catatan
Waktu Standar Tiongkok	(UTC+08.00)	Beijing, Chongqing, Hong Kong, Urumqi	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Kuba	(UTC–05.00)	Havana	
Waktu Standar Garis Batas Tanggal	(UTC–12.00)	Garis Batas Tanggal Internasional Bagian Barat	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Afrika Timur	(UTC+03.00)	Nairobi	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Australia Timur	(UTC+10.00)	Brisbane	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Eropa Timur	(UTC+02.00)	Chisinau	
Waktu Standar Amerika Selatan Timur	(UTC–03.00)	Brasilia	
Waktu Standar Pulau Paskah	(UTC–06.00)	Pulau Paskah	
Waktu Standar Timur	(UTC–05.00)	Waktu Timur (AS dan Kanada)	
Waktu Standar Timur (Meksiko)	(UTC–05.00)	Chetumal	
Waktu Standar Mesir	(UTC+02.00)	Kairo	

Zona waktu	Offset waktu standar	Deskripsi	Catatan
Waktu Standar Ekaterinburg	(UTC+05.00)	Ekaterinburg	
Waktu Standar Fiji	(UTC+12.00)	Fiji	
Waktu Standar FLE	(UTC+02.00)	Helsinki, Kyiv, Riga, Sofia, Tallinn, Vilnius	
Waktu Standar Georgia	(UTC+04.00)	Tbilisi	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar GMT	(UTC)	Dublin, Edinburgh, Lisbon, London	Zona waktu ini tidak sama dengan Greenwich Mean Time. Zona waktu ini menggunakan waktu musim panas.
Waktu Standar Greenland	(UTC–03.00)	Greenland	
Waktu Standar Greenwich	(UTC)	Monrovia, Reykjavik	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar GTB	(UTC+02.00)	Athena, Bukares	
Waktu Standar Haiti	(UTC–05.00)	Haiti	
Waktu Standar Hawaii	(UTC–10.00)	Hawaii	

Zona waktu	Offset waktu standar	Deskripsi	Catatan
Waktu Standar India	(UTC+05.30)	Chennai, Kolkata, Mumbai, New Delhi	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Iran	(UTC+03.30)	Teheran	
Waktu Standar Israel	(UTC+02.00)	Yerusalem	
Waktu Standar Yordania	(UTC+02.00)	Amman	
Waktu Standar Kaliningrad	(UTC+02.00)	Kaliningrad	
Waktu Standar Kamchatka	(UTC+12.00)	Petropavlovsk-Kamchatsky – Lama	
Waktu Standar Korea	(UTC+09.00)	Seoul	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Libya	(UTC+02.00)	Tripoli	
Waktu Standar Kepulauan Line	(UTC+14.00)	Pulau Kiritimati	
Waktu Standar Lord Howe	(UTC+10.30)	Pulau Lord Howe	
Waktu Standar Magadan	(UTC+11.00)	Magadan	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Magallanes	(UTC–03.00)	Punta Arenas	
Waktu Standar Marquesas	(UTC–09.30)	Kepulauan Marquesas	

Zona waktu	Offset waktu standar	Deskripsi	Catatan
Waktu Standar Mauritius	(UTC+04.00)	Port Louis	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Timur Tengah	(UTC+02.00)	Beirut	
Waktu Standar Montevideo	(UTC–03.00)	Montevideo	
Waktu Standar Maroko	(UTC+01.00)	Casablanca	
Waktu Standar Pegunungan	(UTC–07.00)	Waktu Pegunungan (AS dan Kanada)	
Waktu Standar Pegunungan (Meksiko)	(UTC–07.00)	Chihuahua, La Paz, Mazatlan	
Waktu Standar Myanmar	(UTC+06.30)	Yangon (Rangoon)	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Asia Tengah Utara	(UTC+07.00)	Novosibirsk	
Waktu Standar Namibia	(UTC+02.00)	Windhoek	
Waktu Standar Nepal	(UTC+05.45)	Kathmandu	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Selandia Baru	(UTC+12.00)	Auckland, Wellington	
Waktu Standar Newfoundland	(UTC–03.30)	Newfoundland	

Zona waktu	Offset waktu standar	Deskripsi	Catatan
Waktu Standar Norfolk	(UTC+11.00)	Pulau Norfolk	
Waktu Standar Timur Asia Utara	(UTC+08.00)	Irkutsk	
Waktu Standar Asia Utara	(UTC+07.00)	Krasnoyarsk	
Waktu Standar Korea Utara	(UTC+09.00)	Pyongyang	
Waktu Standar Omsk	(UTC+06.00)	Omsk	
Waktu Standar SA Pasifik	(UTC–03.00)	Santiago	
Waktu Standar Pasifik	(UTC–08.00)	Waktu Pasifik (AS dan Kanada)	
Waktu Standar Pasifik (Meksiko)	(UTC–08.00)	Baja California	
Waktu Standar Pakistan	(UTC+05.00)	Islamabad, Karachi	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Paraguay	(UTC–04.00)	Asuncion	
Waktu Standar Romance	(UTC+01.00)	Brussels, Copenhagen, Madrid, Paris	
Zona Waktu Rusia 10	(UTC+11.00)	Chokurdakh	
Zona Waktu Rusia 11	(UTC+12.00)	Anadyr, Petropavlovsk-Kamchatsky	
Zona Waktu Rusia 3	(UTC+04.00)	Izhevsk, Samara	

Zona waktu	Offset waktu standar	Deskripsi	Catatan
Waktu Standar Rusia	(UTC+03.00)	Moskow, St. Petersburg, Volgograd	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Timur SA	(UTC–03.00)	Cayenne, Fortaleza	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Pasifik SA	(UTC–05.00)	Bogota, Lima, Quito, Rio Branco	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Barat SA	(UTC–04.00)	Georgetown, La Paz, Manaus, San Juan	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Saint Pierre	(UTC–03.00)	Saint Pierre dan Miquelon	
Waktu Standar Sakhalin	(UTC+11.00)	Sakhalin	
Waktu Standar Samoa	(UTC+13.00)	Samoa	
Waktu Standar Sao Tome	(UTC+01.00)	Sao Tome	
Waktu Standar Saratov	(UTC+04.00)	Saratov	
Waktu Standar Asia Tenggara	(UTC+07.00)	Bangkok, Hanoi, Jakarta	Zona waktu ini tidak menggunakan waktu musim panas.

Zona waktu	Offset waktu standar	Deskripsi	Catatan
Waktu Standar Singapura	(UTC+08.00)	Kuala Lumpur, Singapura	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Afrika Selatan	(UTC+02.00)	Harare, Pretoria	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Sri Lanka	(UTC+05.30)	Sri Jayawardenepura	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Sudan	(UTC+02.00)	Khartoum	
Waktu Standar Syria	(UTC+02.00)	Damascus	
Waktu Standar Taipei	(UTC+08.00)	Taipei	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Tasmania	(UTC+10.00)	Hobart	
Waktu Standar Tocantins	(UTC–03.00)	Araguaina	
Waktu Standar Tokyo	(UTC+09.00)	Osaka, Sapporo, Tokyo	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Tomsk	(UTC+07.00)	Tomsk	

Zona waktu	Offset waktu standar	Deskripsi	Catatan
Waktu Standar Tonga	(UTC+13.00)	Nuku'alofa	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Transbaikal	(UTC+09.00)	Chita	
Waktu Standar Turki	(UTC+03.00)	Istanbul	
Waktu Standar Turks dan Caicos	(UTC–05.00)	Turks dan Caicos	
Waktu Standar Ulaanbaatar	(UTC+08.00)	Ulaanbaatar	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Timur AS	(UTC–05.00)	Indiana (Timur)	
Waktu Standar Pegunungan AS	(UTC–07.00)	Arizona	Zona waktu ini tidak menggunakan waktu musim panas.
UTC	UTC	Waktu Universal Terkoordinasi	Zona waktu ini tidak menggunakan waktu musim panas.
UTC–02	(UTC–02.00)	Waktu Universal Terkoordinasi–02	Zona waktu ini tidak menggunakan waktu musim panas.
UTC–08	(UTC–08.00)	Waktu Universal Terkoordinasi–08	

Zona waktu	Offset waktu standar	Deskripsi	Catatan
UTC–09	(UTC–09.00)	Waktu Universal Terkoordinasi–09	
UTC–11	(UTC–11.00)	Waktu Universal Terkoordinasi–11	Zona waktu ini tidak menggunakan waktu musim panas.
UTC+12	(UTC+12.00)	Waktu Universal Terkoordinasi+12	Zona waktu ini tidak menggunakan waktu musim panas.
UTC+13	(UTC+13.00)	Waktu Universal Terkoordinasi+13	
Waktu Standar Venezuela	(UTC–04.00)	Caracas	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Vladivostok	(UTC+10.00)	Vladivostok	
Waktu Standar Volgograd	(UTC+04.00)	Volgograd	
Waktu Standar Australia Barat	(UTC+08.00)	Perth	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Afrika Tengah Barat	(UTC+01.00)	Afrika Tengah Barat	Zona waktu ini tidak menggunakan waktu musim panas.

Zona waktu	Offset waktu standar	Deskripsi	Catatan
Waktu Standar Eropa Barat	(UTC+01.00)	Amsterdam, Berlin, Bern, Roma, Stockholm, Wina	
Waktu Standar Mongolia Barat	(UTC+07.00)	Hovd	
Waktu Standar Asia Barat	(UTC+05.00)	Ashgabat, Tashkent	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Tepi Barat	(UTC+02.00)	Gaza, Hebron	
Waktu Standar Pasifik Barat	(UTC+10.00)	Guam, Port Moresby	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Yakutsk	(UTC+09.00)	Yakutsk	

Menggunakan Kunci Master Layanan dengan RDS Custom for SQL Server

RDS Custom for SQL Server mendukung penggunaan Kunci Master Layanan (SMK). RDS Custom mempertahankan SMK yang sama selama masa pakai instans DB RDS Custom for SQL Server Anda. Dengan mempertahankan SMK yang sama, instans DB Anda dapat menggunakan objek yang dienkripsi dengan SMK, seperti kata sandi dan kredensial server yang ditautkan. Jika Anda menggunakan deployment Multi-AZ, RDS Custom juga menyinkronkan dan mengelola SMK antara instans DB primer dan sekunder.

Topik

- Ketersediaan wilayah dan versi
- Fitur yang didukung
- Menggunakan TDE

- Mengonfigurasi fitur
- Persyaratan dan pembatasan

Ketersediaan wilayah dan versi

Penggunaan SMK didukung di semua Wilayah tempat RDS Custom for SQL Server tersedia, untuk semua versi SQL Server yang tersedia di RDS Custom. Untuk informasi selengkapnya tentang ketersediaan versi dan Wilayah Amazon RDS dengan RDS Custom for SQL Server, lihat RDS Custom for SQL Server.

Fitur yang didukung

Saat menggunakan SMK dengan RDS Custom for SQL Server, fitur berikut didukung:

- Enkripsi Data Transparan (TDE)
- Enkripsi tingkat kolom
- Database Mail
- Server Tertaut
- SQL Server Integration Services (SSIS)

Menggunakan TDE

SMK memungkinkan kemampuan untuk mengonfigurasi Enkripsi Data Transparan (TDE), yang mengenkripsi data sebelum ditulis ke penyimpanan, dan mendekripsi data secara otomatis saat data dibaca dari penyimpanan. Tidak seperti RDS for SQL Server, mengonfigurasi TDE pada instans DB RDS Custom for SQL Server tidak memerlukan penggunaan grup opsi. Sebagai gantinya, setelah membuat sertifikat dan kunci enkripsi basis data, Anda dapat menjalankan perintah berikut untuk mengaktifkan TDE di tingkat basis data:

```
ALTER DATABASE [myDatabase] SET ENCRYPTION ON;
```

Untuk informasi selengkapnya tentang penggunaan TDE dengan RDS for SQL Server, lihat Dukungan untuk Enkripsi Data Transparan di SQL Server.

Untuk informasi detail tentang TDE di Microsoft SQL Server, lihat Transparent data encryption di dokumentasi Microsoft.

Mengonfigurasi fitur

Untuk langkah-langkah mendetail tentang mengonfigurasi fitur yang menggunakan SMK dengan RDS Custom for SQL Server, Anda dapat menggunakan postingan di blog basis data Amazon RDS berikut:

- Server tertaut: Configuring linked servers on RDS Custom for SQL Server.
- SSIS: Migrate SSIS packages to RDS Custom for SQL Server.
- TDE: Secure your data using TDE on RDS Custom for SQL Server.

Persyaratan dan pembatasan

Saat menggunakan SMK dengan instans DB RDS Custom for SQL Server, ingat persyaratan dan batasan berikut:

- Jika membuat ulang SMK pada instans DB Anda, Anda harus segera melakukan snapshot DB manual. Sebaiknya hindari pembuatan ulang SMK jika memungkinkan.
- Anda harus menjaga cadangan sertifikat server dan kata sandi kunci master basis data. Jika tidak mempertahankan cadangan tersebut, Anda dapat kehilangan data.
- Jika mengonfigurasi SSIS, Anda harus menggunakan dokumen SSM untuk menggabungkan instans DB RDS Custom for SQL Server ke domain jika terjadi komputasi skala atau penggantian host.
- Ketika TDE atau enkripsi kolom diaktifkan, cadangan basis data dienkripsi secara otomatis. Saat Anda melakukan pemulihan snapshot atau pemulihan titik waktu, SMK dari instans DB sumber akan dipulihkan guna mendekripsi data untuk pemulihan, dan SMK baru akan dihasilkan untuk mengenkripsi ulang data pada instans yang dipulihkan.

Untuk informasi selengkapnya tentang Kunci Master Layanan di Microsoft SQL Server, lihat SQL Server and Database Encryption Keys dalam dokumentasi Microsoft.

Menyiapkan lingkungan Anda untuk Amazon RDS Custom for SQL Server

Sebelum Anda membuat dan mengelola instans DB untuk instans DB Amazon RDS Custom for SQL Server, pastikan untuk melakukan tugas-tugas berikut.

Daftar Isi

Prasyarat untuk menyiapkan RDS Custom for SQL Server

Sebelum membuat instans DB RDS Custom for SQL Server, pastikan lingkungan Anda memenuhi persyaratan yang dijelaskan dalam topik ini. Sebagai bagian dari proses penyiapan ini, pastikan untuk mengonfigurasi prasyarat berikut:

- Konfigurasikan pengguna dan peran AWS Identity and Access Management (IAM) yang ditentukan.

  Kedua hal ini akan digunakan untuk membuat instans DB RDS Custom atau diteruskan sebagai parameter dalam permintaan pembuatan.
- Konfirmasikan bahwa tidak ada kebijakan kontrol layanan (SCP) yang membatasi izin tingkat akun.

  Jika akun yang Anda gunakan adalah bagian dari Organisasi AWS, akun tersebut mungkin memiliki kebijakan kontrol layanan (SCP) yang membatasi izin tingkat akun. Pastikan SCP tidak membatasi izin pada pengguna dan peran yang Anda buat menggunakan prosedur berikut.

  Untuk informasi selengkapnya tentang SCP, lihat Kebijakan kontrol layanan (SCP) dalam Panduan Pengguna AWS Organizations. Gunakan perintah AWS CLIdescribe-organization untuk memeriksa apakah akun Anda adalah bagian dari Organisasi AWS.

  Untuk informasi selengkapnya tentang AWS Organizations, lihat Apa itu AWS Organizations dalam Panduan Pengguna AWS Organizations.

> **Note**
>
> Untuk step-by-step tutorial tentang cara mengatur prasyarat dan meluncurkan Amazon RDS Custom untuk SQL Server, lihat posting blog Memulai Amazon RDS Custom for SQL Server menggunakan template (Pengaturan jaringan) CloudFormation

Untuk setiap tugas, Anda dapat menemukan deskripsi berikut terkait persyaratan dan batasan khusus untuk tugas tersebut. Misalnya, saat Anda membuat instans DB RDS Custom for SQL Server, gunakan salah satu instans SQL Server yang tercantum di dalam Dukungan kelas instans DB untuk RDS Custom for SQL Server.

Untuk persyaratan umum yang berlaku untuk RDS Custom for SQL Server, lihat Persyaratan umum untuk RDS Custom for SQL Server.

Unduh dan instal AWS CLI.

AWS memberi Anda antarmuka baris perintah untuk menggunakan fitur RDS Custom. Anda dapat menggunakan AWS CLI versi 1 atau versi 2.

Untuk informasi tentang mengunduh dan menginstal AWS CLI, lihat Menginstal atau memperbarui versi terbaru AWS CLI.

Lewati langkah ini jika salah satu hal berikut ini berlaku:

- Anda berencana untuk mengakses RDS Custom hanya dari AWS Management Console.
- Anda telah mengunduh AWS CLI untuk Amazon RDS atau mesin DB RDS Custom yang berbeda.

Berikan izin yang diperlukan ke prinsipal IAM Anda

Anda menggunakan peran IAM atau pengguna IAM (disebut sebagai prinsipal IAM) untuk membuat instans DB RDS Custom for SQL Server menggunakan konsol atau CLI. Prinsipal IAM ini harus memiliki salah satu dari kebijakan berikut untuk pembuatan instans DB yang berhasil:

- Kebijakan `AdministratorAccess`
- Kebijakan `AmazonRDSFullAccess` dengan izin tambahan berikut:

```
iam:SimulatePrincipalPolicy
cloudtrail:CreateTrail
cloudtrail:StartLogging
s3:CreateBucket
s3:PutBucketPolicy
s3:PutBucketObjectLockConfiguration
s3:PutBucketVersioning
kms:CreateGrant
kms:DescribeKey
```

Untuk informasi selengkapnya tentang izin `kms:CreateGrant`, lihat Manajemen AWS KMS key.

Contoh kebijakan JSON berikut memberikan izin yang diperlukan.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ValidateIamRole",
            "Effect": "Allow",
            "Action": "iam:SimulatePrincipalPolicy",
            "Resource": "*"
        },
```

```
        {
            "Sid": "CreateCloudTrail",
            "Effect": "Allow",
            "Action": [
                "cloudtrail:CreateTrail",
                "cloudtrail:StartLogging"
            ],
            "Resource": "arn:aws:cloudtrail:*:*:trail/do-not-delete-rds-custom-*"
        },
        {
            "Sid": "CreateS3Bucket",
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:PutBucketPolicy",
                "s3:PutBucketObjectLockConfiguration",
                "s3:PutBucketVersioning"
            ],
            "Resource": "arn:aws:s3:::do-not-delete-rds-custom-*"
        },
        {
            "Sid": "CreateKmsGrant",
            "Effect": "Allow",
            "Action": [
                "kms:CreateGrant",
                "kms:DescribeKey"
            ],
            "Resource": "*"
        }
    ]
}
```

Selain itu, prinsipal IAM memerlukan izin `iam:PassRole` pada peran IAM. Izin ini harus dilampirkan ke profil instans yang diteruskan dalam parameter `custom-iam-instance-profile` dalam permintaan untuk membuat instans DB RDS Custom. Profil instans dan peran terlampirnya dibuat nanti dalam Konfigurasikan jaringan, profil instans, dan enkripsi.

Pastikan izin yang tercantum sebelumnya tidak dibatasi oleh kebijakan kontrol layanan (SCP), batasan izin, atau kebijakan sesi yang terkait dengan prinsipal IAM.

Konfigurasikan jaringan, profil instans, dan enkripsi

Anda dapat mengonfigurasi peran profil instans IAM, cloud privat virtual (VPC), dan kunci enkripsi simetris AWS KMS dengan menggunakan salah satu dari proses berikut:

- Mengonfigurasi dengan AWS CloudFormation (direkomendasikan)
- Mengonfigurasi secara manual

Jika akun Anda merupakan bagian dari AWS Organisasi, pastikan izin yang diperlukan oleh peran profil instans tidak dibatasi oleh kebijakan kontrol layanan (SCP).

Konfigurasi jaringan berikut dirancang untuk beroperasi paling baik dengan instans DB yang tidak dapat diakses secara publik. Artinya, Anda tidak dapat terhubung langsung ke instans DB dari luar VPC.

Mengonfigurasi dengan AWS CloudFormation

Untuk menyederhanakan pengaturan, Anda dapat menggunakan file AWS CloudFormation template untuk membuat CloudFormation tumpukan. Untuk mempelajari cara membuat tumpukan, lihat Membuat tumpukan di konsol AWS CloudFormation dalam Panduan Pengguna AWS CloudFormation.

Untuk tutorial tentang cara meluncurkan Amazon RDS Custom for SQL Server menggunakan templat AWS CloudFormation, lihat Memulai Amazon RDS Custom for SQL Server menggunakan templat AWS CloudFormation dalam Blog Basis Data AWS.

Topik

- Sumber daya yang dibuat oleh CloudFormation
- Mengunduh file templat
- Mengkonfigurasi sumber daya menggunakan CloudFormation

Sumber daya yang dibuat oleh CloudFormation

Berhasil membuat CloudFormation tumpukan membuat sumber daya berikut diAkun AWS:

- Kunci KMS enkripsi simetris untuk enkripsi data yang dikelola oleh RDS Custom.
- Profil instans dan peran IAM terkait untuk dilampirkan ke instans RDS Custom.

- VPC dengan rentang CIDR ditentukan sebagai parameter. CloudFormation Nilai default-nya adalah `10.0.0.0/16`.
- Dua subnet privat dengan rentang CIDR yang ditentukan dalam parameter, dan dua Zona Ketersediaan yang berbeda di Wilayah AWS. Nilai default untuk CIDR subnet adalah `10.0.128.0/20` dan `10.0.144.0/20`.
- Opsi DHCP yang diatur untuk VPC dengan resolusi nama domain ke server Sistem Nama Domain (DNS) Amazon.
- Tabel rute untuk dikaitkan dengan dua subnet privat dan tanpa akses ke internet.
- Daftar Kontrol Akses (ACL) Jaringan untuk dikaitkan dengan dua subnet privat dan akses yang dibatasi pada HTTPS.
- Grup keamanan VPC untuk dikaitkan dengan instans RDS Custom. Akses dibatasi untuk HTTPS keluar ke titik akhir Layanan AWS yang diperlukan oleh RDS Custom.
- Grup keamanan VPC untuk dikaitkan dengan titik akhir VPC yang dibuat untuk titik akhir Layanan AWS yang diperlukan oleh RDS Custom.
- Grup subnet DB tempat instans RDS Custom dibuat.
- Titik akhir VPC untuk setiap titik akhir Layanan AWS yang diperlukan oleh RDS Custom.

Gunakan prosedur berikut untuk membuat CloudFormation tumpukan untuk RDS Custom for SQL Server.

Mengunduh file templat

Untuk mengunduh file templat

1. Buka menu konteks (klik kanan) untuk custom-sqlserver-onboardtautan.zip dan pilih Simpan Tautan Sebagai.
2. Simpan dan ekstrak file ke komputer Anda.

Mengkonfigurasi sumber daya menggunakan CloudFormation

Untuk mengkonfigurasi sumber daya menggunakan CloudFormation

1. Buka CloudFormation konsol di https://console.aws.amazon.com/cloudformation.
2. Untuk memulai wizard Buat Tumpukan, pilih Buat Tumpukan.

   Halaman Buat tumpukan muncul.

3. Untuk Prasyarat - Siapkan templat, pilih Template sudah siap.
4. Untuk Tentukan templat, lakukan hal berikut:
   a. Untuk Sumber templat, pilih Unggah file templat.
   b. Untuk Pilih file, navigasikan ke dan pilih file yang benar.
5. Pilih Berikutnya.

   Halaman Tentukan detail tumpukan muncul.
6. Untuk Nama tumpukan, masukkan **`rds-custom-sqlserver`**.
7. Untuk Parameter, lakukan hal berikut:
   a. Untuk mempertahankan opsi default, pilih Berikutnya.
   b. Untuk mengubah opsi, pilih rentang blok CIDR yang sesuai untuk VPC dan dua subnetnya, lalu pilih Berikutnya.

      Baca deskripsi setiap parameter dengan cermat sebelum mengubah parameter.
8. Pada halaman Konfigurasikan opsi tumpukan, pilih Berikutnya.
9. Pada halaman Tinjauan rds-custom-sqlserver, lakukan hal berikut:
   a. Untuk Kemampuan, pilih kotak centang Saya memahami bahwa AWS CloudFormation dapat membuat sumber daya IAM dengan nama kustom.
   b. Pilih Buat tumpukan.
10. (Opsional): Anda dapat memperbarui izin SQS dalam peran profil instans.

    Jika Anda hanya ingin menerapkan instans DB AZ tunggal, Anda dapat mengedit file CloudFormation template untuk menghapus izin SQS. Izin SQS hanya diperlukan untuk deployment Multi-AZ dan memungkinkan RDS Custom for SQL Server memanggil Amazon SQS untuk melakukan tindakan tertentu. Karena tidak diperlukan untuk deployment AZ Tunggal, Anda dapat memilih untuk menghapus izin ini untuk mengikuti prinsip hak akses paling rendah.

    Jika Anda ingin mengonfigurasi deployment Multi-AZ, Anda tidak perlu menghapus izin SQS.

> **Note**
>
> Jika Anda menghapus izin SQS lalu memilih untuk memodifikasi ke deployment Multi-AZ, pembuatan Multi-AZ akan gagal. Anda perlu menambahkan kembali izin SQS sebelum memodifikasi ke deployment Multi-AZ.

Untuk membuat perubahan opsional ini ke CloudFormation template, buka CloudFormation konsol di https://console.aws.amazon.com/cloudformation, dan edit file template dengan menghapus baris berikut:

```
                {
        "Sid": "SendMessageToSQSQueue",
        "Effect": "Allow",
        "Action": [
          "SQS:SendMessage",
          "SQS:ReceiveMessage",
          "SQS:DeleteMessage",
          "SQS:GetQueueUrl"

        ],
        "Resource": [
          {
            "Fn::Sub": "arn:${AWS::Partition}:sqs:${AWS::Region}:
${AWS::AccountId}:do-not-delete-rds-custom-*"
          }
        ],
        "Condition": {
          "StringLike": {
            "aws:ResourceTag/AWSRDSCustom": "custom-sqlserver"
          }
        }
      }
```

CloudFormation menciptakan sumber daya yang dibutuhkan RDS Custom untuk SQL Server. Jika pembuatan tumpukan gagal, baca tab Peristiwa untuk melihat pembuatan sumber daya mana yang gagal dan alasan statusnya.

Tab Output untuk CloudFormation tumpukan ini di konsol harus memiliki informasi tentang semua sumber daya yang akan diteruskan sebagai parameter untuk membuat instance RDS Custom untuk SQL Server DB. Pastikan untuk menggunakan grup keamanan VPC dan grup subnet DB yang dibuat oleh CloudFormation untuk instans RDS Custom DB. Secara default, RDS mencoba melampirkan grup keamanan VPC default, yang mungkin tidak memiliki akses yang Anda butuhkan.

> **Note**
>
> Saat Anda menghapus CloudFormation tumpukan, semua sumber daya yang dibuat oleh tumpukan akan dihapus kecuali kunci KMS. Kunci KMS beralih ke status `pending-deletion` dan dihapus setelah 30 hari. Untuk menjaga kunci KMS, lakukan CancelKeyDeletionoperasi selama masa tenggang 30 hari.

Jika Anda biasa CloudFormation membuat sumber daya, Anda dapat melewatiMengonfigurasi secara manual.

Mengonfigurasi secara manual

Jika Anda memilih untuk mengonfigurasi sumber daya secara manual, lakukan tugas berikut.

> **Note**
>
> Untuk menyederhanakan pengaturan, Anda dapat menggunakan file AWS CloudFormation template untuk membuat CloudFormation tumpukan daripada konfigurasi manual. Untuk informasi selengkapnya, lihat Mengonfigurasi dengan AWS CloudFormation.

Topik

- Pastikan Anda memiliki kunci enkripsi simetris AWS KMS
- Membuat profil instans dan peran IAM Anda secara manual
- Mengonfigurasi VPC Anda secara manual

Pastikan Anda memiliki kunci enkripsi simetris AWS KMS

Enkripsi simetris AWS KMS key diperlukan untuk RDS Custom. Saat Anda membuat instans DB RDS Custom for SQL Server, pastikan untuk menyediakan pengidentifikasi kunci KMS. Untuk informasi

selengkapnya, lihat Membuat dan menghubungkan ke instans DB untuk Amazon RDS Custom for SQL Server.

Anda memiliki opsi berikut:

- Jika Anda memiliki kunci KMS yang dikelola pelanggan yang sudah ada di Akun AWS Anda, Anda dapat menggunakannya dengan RDS Custom. Tidak ada tindakan lebih lanjut yang diperlukan.
- Jika Anda telah membuat kunci KMS enkripsi simetris yang dikelola pelanggan untuk mesin RDS Custom yang berbeda, Anda dapat menggunakan kembali kunci KMS yang sama. Tidak ada tindakan lebih lanjut yang diperlukan.
- Jika Anda tidak memiliki kunci KMS enkripsi simetris yang dikelola pelanggan yang sudah ada di akun Anda, buat kunci KMS dengan mengikuti petunjuk dalam Membuat kunci dalam Panduan Developer AWS Key Management Service.
- Jika Anda membuat CEV atau instans DB RDS Custom, dan kunci KMS Anda berada di Akun AWS yang berbeda, pastikan untuk menggunakan AWS CLI. Anda tidak dapat menggunakan konsol AWS dengan kunci KMS lintas akun.

> **Important**
>
> RDS Custom tidak mendukung kunci KMS yang dikelola AWS.

Pastikan kunci enkripsi simetris Anda memberikan akses ke operasi `kms:Decrypt` dan `kms:GenerateDataKey` ke peran AWS Identity and Access Management (IAM) di profil instans IAM Anda. Jika Anda memiliki kunci enkripsi simetris baru di akun Anda, perubahan tidak diperlukan. Jika tidak, pastikan kebijakan kunci enkripsi simetris Anda memberikan akses ke operasi ini.

Untuk informasi selengkapnya, lihat Langkah 4: Konfigurasikan IAM untuk RDS Custom for Oracle.

Membuat profil instans dan peran IAM Anda secara manual

Untuk menggunakan RDS Custom for SQL Server, buat profil instans IAM dan peran IAM seperti yang dijelaskan berikut.

Untuk membuat profil instans IAM dan peran IAM untuk RDS Custom for SQL Server

1. Buat peran IAM bernama `AWSRDSCustomSQLServerInstanceRole` dengan kebijakan kepercayaan yang memungkinkan Amazon EC2 mengambil peran ini.

2. Tambahkan kebijakan akses ke `AWSRDSCustomSQLServerInstanceRole`.
3. Buat profil instans IAM untuk RDS Custom for SQL Server yang bernama `AWSRDSCustomSQLServerInstanceProfile`.
4. Tambahkan peran `AWSRDSCustomSQLServerInstanceRole` ke profil instans.

Buat peran AWSRDSCustomSQLServerInstanceRole IAM

Contoh berikut membuat peran `AWSRDSCustomSQLServerInstanceRole`. Kebijakan kepercayaan memungkinkan Amazon EC2 mengambil peran tersebut.

```
aws iam create-role \
    --role-name AWSRDSCustomSQLServerInstanceRole \
    --assume-role-policy-document '{
        "Version": "2012-10-17",
          "Statement": [
            {
              "Action": "sts:AssumeRole",
              "Effect": "Allow",
              "Principal": {
                  "Service": "ec2.amazonaws.com"
              }
            }
          ]
        }'
```

Menambahkan kebijakan akses ke AWSRDSCustomSQLServerInstanceRole

Saat Anda menyematkan kebijakan inline dalam sebuah peran, kebijakan inline ini digunakan sebagai bagian dari kebijakan akses (izin) peran. Anda membuat kebijakan `AWSRDSCustomSQLServerIamRolePolicy`, yang memungkinkan Amazon EC2 mendapatkan dan menerima pesan serta melakukan berbagai tindakan.

Pastikan izin dalam kebijakan akses tidak dibatasi oleh SCP atau batasan izin yang terkait dengan peran profil instans.

Contoh berikut membuat kebijakan akses bernama `AWSRDSCustomSQLServerIamRolePolicy`, dan menambahkannya ke peran `AWSRDSCustomSQLServerInstanceRole`. Contoh ini mengasumsikan bahwa variabel `'$REGION'`, `$ACCOUNT_ID`, dan `'$CUSTOMER_KMS_KEY_ID'` telah ditetapkan. `'$CUSTOMER_KMS_KEY_ID'` adalah ID, bukan Amazon Resource Name (ARN), milik kunci KMS yang Anda tentukan di Pastikan Anda memiliki kunci enkripsi simetris AWS KMS.

```
                aws iam put-role-policy \
--role-name AWSRDSCustomSQLServerInstanceRole \
--policy-name AWSRDSCustomSQLServerIamRolePolicy \
--policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ssmAgent1",
            "Effect": "Allow",
            "Action": [
                "ssm:GetDeployablePatchSnapshotForInstance",
                "ssm:ListAssociations",
                "ssm:PutInventory",
                "ssm:PutConfigurePackageResult",
                "ssm:UpdateInstanceInformation",
                "ssm:GetManifest"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ssmAgent2",
            "Effect": "Allow",
            "Action": [
                "ssm:ListInstanceAssociations",
                "ssm:PutComplianceItems",
                "ssm:UpdateAssociationStatus",
                "ssm:DescribeAssociation",
                "ssm:UpdateInstanceAssociationStatus"
            ],
            "Resource": "arn:aws:ec2:'$REGION':'$ACCOUNT_ID':instance/*",
            "Condition": {
                "StringLike": {
                    "aws:ResourceTag/AWSRDSCustom": "custom-sqlserver"
                }
            }
        },
        {
            "Sid": "ssmAgent3",
            "Effect": "Allow",
            "Action": [
                "ssm:UpdateAssociationStatus",
                "ssm:DescribeAssociation",
```

```
                "ssm:GetDocument",
                "ssm:DescribeDocument"
            ],
            "Resource": "arn:aws:ssm:*:*:document/*"
        },
        {
            "Sid": "ssmAgent4",
            "Effect": "Allow",
            "Action": [
                "ssmmessages:CreateControlChannel",
                "ssmmessages:CreateDataChannel",
                "ssmmessages:OpenControlChannel",
                "ssmmessages:OpenDataChannel"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ssmAgent5",
            "Effect": "Allow",
            "Action": [
                "ec2messages:AcknowledgeMessage",
                "ec2messages:DeleteMessage",
                "ec2messages:FailMessage",
                "ec2messages:GetEndpoint",
                "ec2messages:GetMessages",
                "ec2messages:SendReply"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ssmAgent6",
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameters",
                "ssm:GetParameter"
            ],
            "Resource": "arn:aws:ssm:*:*:parameter/*"
        },
        {
            "Sid": "ssmAgent7",
            "Effect": "Allow",
            "Action": [
                "ssm:UpdateInstanceAssociationStatus",
                "ssm:DescribeAssociation"
```

```
            ],
            "Resource": "arn:aws:ssm:*:*:association/*"
        },
        {
            "Sid": "eccSnapshot1",
            "Effect": "Allow",
            "Action": "ec2:CreateSnapshot",
            "Resource": [
                "arn:aws:ec2:'$REGION':'$ACCOUNT_ID':volume/*"
            ],
            "Condition": {
                "StringLike": {
                    "aws:ResourceTag/AWSRDSCustom": "custom-sqlserver"
                }
            }
        },
        {
            "Sid": "eccSnapshot2",
            "Effect": "Allow",
            "Action": "ec2:CreateSnapshot",
            "Resource": [
                "arn:aws:ec2:'$REGION'::snapshot/*"
            ],
            "Condition": {
                "StringLike": {
                    "aws:RequestTag/AWSRDSCustom": "custom-sqlserver"
                }
            }
        },
        {
            "Sid": "eccCreateTag",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aws:RequestTag/AWSRDSCustom": "custom-sqlserver",
                    "ec2:CreateAction": [
                        "CreateSnapshot"
                    ]
                }
            }
        },
        {
```

```
                "Sid": "s3BucketAccess",
                "Effect": "Allow",
                "Action": [
                    "s3:putObject",
                    "s3:getObject",
                    "s3:getObjectVersion",
                    "s3:AbortMultipartUpload"
                ],
                "Resource": [
                    "arn:aws:s3:::do-not-delete-rds-custom-*/*"
                ]
            },
            {
                "Sid": "customerKMSEncryption",
                "Effect": "Allow",
                "Action": [
                    "kms:Decrypt",
                    "kms:GenerateDataKey*"
                ],
                "Resource": [
                    "arn:aws:kms:'$REGION':'$ACCOUNT_ID':key/'$CUSTOMER_KMS_KEY_ID'"
                ]
            },
            {
                "Sid": "readSecretsFromCP",
                "Effect": "Allow",
                "Action": [
                    "secretsmanager:GetSecretValue",
                    "secretsmanager:DescribeSecret"
                ],
                "Resource": [
                    "arn:aws:secretsmanager:'$REGION':'$ACCOUNT_ID':secret:do-not-
delete-rds-custom-*"
                ],
                "Condition": {
                    "StringLike": {
                        "aws:ResourceTag/AWSRDSCustom": "custom-sqlserver"
                    }
                }
            },
            {
                "Sid": "publishCWMetrics",
                "Effect": "Allow",
                "Action": "cloudwatch:PutMetricData",
```

```
                "Resource": "*",
                "Condition": {
                    "StringEquals": {
                        "cloudwatch:namespace": "rdscustom/rds-custom-sqlserver-agent"
                    }
                }
            },
            {
                "Sid": "putEventsToEventBus",
                "Effect": "Allow",
                "Action": "events:PutEvents",
                "Resource": "arn:aws:events:'$REGION':'$ACCOUNT_ID':event-bus/default"
            },
            {
                "Sid": "cwlOperations1",
                "Effect": "Allow",
                "Action": [
                    "logs:PutRetentionPolicy",
                    "logs:PutLogEvents",
                    "logs:DescribeLogStreams",
                    "logs:CreateLogStream",
                    "logs:CreateLogGroup"
                ],
                "Resource": "arn:aws:logs:'$REGION':'$ACCOUNT_ID':log-group:rds-custom-
instance-*"
            },
            {
            "Condition": {
                "StringLike": {
                    "aws:ResourceTag/AWSRDSCustom": "custom-sqlserver"
                }
            },
            "Action": [
                "SQS:SendMessage",
                "SQS:ReceiveMessage",
                "SQS:DeleteMessage",
                "SQS:GetQueueUrl"
            ],
            "Resource": [
                "arn:aws:sqs:'$REGION':'$ACCOUNT_ID':do-not-delete-rds-custom-*"
            ],
            "Effect": "Allow",
            "Sid": "SendMessageToSQSQueue"
        }
```

```
        ]
    }'
```

Buat profil instans RDS Custom for SQL Server

Buat profil instans Anda sebagai berikut, dengan memberinya nama `AWSRDSCustomSQLServerInstanceProfile`.

```
aws iam create-instance-profile \
    --instance-profile-name AWSRDSCustomSQLServerInstanceProfile
```

Tambahkan AWSRDSCustomSQLServerInstanceRole ke profil instans RDS Custom for SQL Server

Tambahkan peran `AWSRDSCustomInstanceRoleForRdsCustomInstance` ke profil `AWSRDSCustomSQLServerInstanceProfile`.

```
aws iam add-role-to-instance-profile \
    --instance-profile-name AWSRDSCustomSQLServerInstanceProfile \
    --role-name AWSRDSCustomSQLServerInstanceRole
```

Mengonfigurasi VPC Anda secara manual

Instans DB RDS Custom Anda berada di cloud privat virtual (VPC) yang didasarkan pada layanan Amazon VPC, seperti instans Amazon EC2 atau instans Amazon RDS. Anda menyediakan dan mengonfigurasi VPC Anda sendiri. Dengan demikian, Anda memiliki kontrol penuh atas pengaturan jaringan instans Anda.

RDS Custom mengirimkan komunikasi dari instans DB Anda ke Layanan AWS lain. Pastikan layanan berikut dapat diakses dari subnet tempat Anda membuat instans DB RDS Custom for Oracle:

- Amazon CloudWatch
- CloudWatch Log Amazon
- CloudWatch Acara Amazon
- Amazon EC2
- Amazon EventBridge
- Amazon S3
- AWS Secrets Manager
- AWS Systems Manager

Jika akses ke Layanan AWS di atas saat ini tidak ada, konfigurasikan titik akhir VPC berikut:

- `com.amazonaws.region.ec2messages`
- `com.amazonaws.region.events`
- `com.amazonaws.region.logs`
- `com.amazonaws.region.monitoring`
- `com.amazonaws.region.s3`
- `com.amazonaws.region.secretsmanager`
- `com.amazonaws.region.ssmmessages`

Jika RDS Custom tidak dapat berkomunikasi dengan layanan yang diperlukan, RDS Custom akan menerbitkan peristiwa berikut:

```
Database instance in incompatible-network. SSM Agent connection not available. Amazon
 RDS can't connect to the dependent AWS services.
```

Untuk menghindari kesalahan `incompatible-network`, pastikan komponen VPC yang diperlukan dalam komunikasi antara instans DB RDS Custom Anda dan Layanan AWS memenuhi persyaratan berikut:

- Instans DB dapat membuat koneksi keluar pada port 443 ke Layanan AWS lainnya.
- VPC mengizinkan respons masuk untuk permintaan yang berasal dari instans DB RDS Custom Anda.
- RDS Custom dapat secara tepat me-resolve nama domain titik akhir untuk masing-masing Layanan AWS.

RDS Custom mengandalkan konektivitas AWS Systems Manager untuk otomatisasi. Untuk informasi tentang cara mengonfigurasi titik akhir VPC, lihat Membuat titik akhir VPC untuk Systems Manager. Untuk daftar titik akhir di setiap Wilayah, lihat Titik akhir dan kuota AWS Systems Manager dalam Referensi Umum Amazon Web Services.

Jika Anda sudah mengonfigurasi VPC untuk mesin DB RDS Custom yang berbeda, Anda dapat menggunakan kembali VPC tersebut dan melewati proses ini.

Topik

- Konfigurasikan grup keamanan VPC

- Konfigurasikan titik akhir untuk Layanan AWS dependen
- Konfigurasikan layanan metadata instans

Konfigurasikan grup keamanan VPC

Grup keamanan bertindak sebagai firewall virtual untuk instans VPC, yang mengontrol lalu lintas masuk dan keluar. Instans DB RDS Custom memiliki grup keamanan default yang melindungi instans ini. Pastikan grup keamanan Anda mengizinkan lalu lintas antara RDS Custom dan Layanan AWS lainnya.

Untuk mengonfigurasi grup keamanan Anda untuk RDS Custom

1. Masuk ke AWS Management Console dan buka konsol Amazon VPC di https://console.aws.amazon.com/vpc.
2. Izinkan RDS Custom untuk menggunakan grup keamanan default, atau buat grup keamanan Anda sendiri.

   Untuk petunjuk mendetail, lihat Memberikan akses ke instans DB di VPC Anda dengan membuat grup keamanan.
3. Pastikan grup keamanan Anda mengizinkan koneksi keluar pada port 443. RDS Custom membutuhkan port ini untuk berkomunikasi dengan Layanan AWS dependen.
4. Jika Anda memiliki VPC privat dan menggunakan titik akhir VPC, pastikan grup keamanan yang terkait dengan instans DB mengizinkan koneksi keluar pada port 443 ke titik akhir VPC. Pastikan juga bahwa grup keamanan yang terkait dengan VPC mengizinkan koneksi masuk pada port 443 dari instans DB.

   Jika koneksi masuk tidak diizinkan, instans RDS Custom tidak dapat terhubung ke AWS Systems Manager dan titik akhir Amazon EC2. Untuk informasi selengkapnya, lihat Membuat titik akhir Cloud Privat Virtual dalam Panduan Pengguna AWS Systems Manager.

Untuk informasi selengkapnya tentang grup keamanan, lihat Grup keamanan untuk VPC Anda dalam Panduan Developer Amazon VPC.

Konfigurasikan titik akhir untuk Layanan AWS dependen

Pastikan VPC Anda mengizinkan lalu lintas keluar ke Layanan AWS berikut yang berkomunikasi dengan instans DB:

- Amazon CloudWatch
- CloudWatch Log Amazon
- CloudWatch Acara Amazon
- Amazon EC2
- Amazon EventBridge
- Amazon S3
- AWS Secrets Manager
- AWS Systems Manager

Kami menyarankan Anda menambahkan titik akhir untuk setiap layanan ke VPC Anda menggunakan petunjuk berikut. Namun, Anda dapat menggunakan solusi apa pun yang memungkinkan VPC Anda berkomunikasi dengan titik akhir layanan AWS. Misalnya, Anda dapat menggunakan Network Address Translation (NAT) atau AWS Direct Connect.

Untuk mengonfigurasi titik akhir untuk Layanan AWS yang beroperasi dengan RDS Custom

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Pada bilah navigasi, gunakan pemilih Wilayah untuk memilih Wilayah AWS.
3. Di panel navigasi, pilih Titik akhir. Pada panel utama, pilih Buat Titik Akhir.
4. Untuk Kategori layanan, pilih Layanan AWS.
5. Untuk Nama Layanan, pilih titik akhir yang ditunjukkan dalam tabel.
6. Untuk VPC, pilih VPC Anda.
7. Untuk Subnet, pilih subnet dari setiap Zona Ketersediaan yang akan disertakan.

   Titik akhir VPC dapat mencakup beberapa Zona Ketersediaan. AWS membuat sebuah antarmuka jaringan elastis untuk titik akhir VPC di setiap subnet yang Anda pilih. Setiap antarmuka jaringan memiliki nama host Sistem Nama Domain (DNS) dan alamat IP privat.
8. Untuk Grup keamanan, pilih atau buat grup keamanan.

   Anda dapat menggunakan grup keamanan untuk mengontrol akses ke titik akhir Anda, seperti Anda menggunakan firewall. Untuk informasi selengkapnya, lihat Grup Keamanan untuk VPC Anda dalam Panduan Pengguna Amazon VPC.
9. Secara opsional, Anda dapat melampirkan kebijakan ke titik akhir VPC. Kebijakan titik akhir dapat mengontrol akses ke Layanan AWS yang Anda hubungkan. Kebijakan default

mengizinkan semua permintaan melewati titik akhir. Jika Anda menggunakan kebijakan kustom, pastikan permintaan dari instans DB diizinkan dalam kebijakan ini.

10. Pilih Buat titik akhir.

Tabel berikut menjelaskan cara menemukan daftar titik akhir yang dibutuhkan VPC Anda untuk komunikasi keluar.

Layanan	Format titik akhir	Catatan dan tautan
AWS Systems Manager	Gunakan format titik akhir berikut: • `ssm.region.amazonaws.com` • `ssmmessages. region.amazonaws.com`	Untuk daftar titik akhir di setiap Wilayah, lihat Titik akhir dan kuota AWS Systems Manager dalam Referensi Umum Amazon Web Services.
AWS Secrets Manager	Gunakan format titik akhir `secretsmanager. region.amazonaws.com .`	Untuk daftar titik akhir di setiap Wilayah, lihat Titik akhir dan kuota AWS Secrets Manager dalam Referensi Umum Amazon Web Services.
Amazon CloudWatch	Gunakan format titik akhir berikut: • Untuk CloudWatch metrik, gunakan `monitoring. region.amazonaws.com` • Untuk CloudWatch Acara, gunakan `events.region.amazonaws.com` • Untuk CloudWatch Log, gunakan `logs.region.amazonaws.com`	Untuk daftar titik akhir di setiap Wilayah, lihat: • CloudWatch Titik akhir dan kuota Amazon di Referensi Umum Amazon Web Services • Amazon CloudWatch Logs titik akhir dan kuota di Referensi Umum Amazon Web Services • Titik akhir dan kuota CloudWatch Acara Amazon di Referensi Umum Amazon Web Services
Amazon EC2	Gunakan format titik akhir berikut: • `ec2.region.amazonaws.com`	Untuk daftar titik akhir di setiap Wilayah, lihat Titik akhir dan kuota Amazon Elastic Compute Cloud

Layanan	Format titik akhir	Catatan dan tautan
	• `ec2messages. region .amazonaws.com`	dalam Referensi Umum Amazon Web Services.
Amazon S3	Gunakan format titik akhir `s3. region .amazonaws.com` .	Untuk daftar titik akhir di setiap Wilayah, lihat Titik akhir dan kuota Amazon Simple Storage Service dalam Referensi Umum Amazon Web Services. Untuk mempelajari selengkap nya tentang titik akhir gateway untuk Amazon S3, lihat Titik Akhir untuk Amazon S3 dalam Panduan Developer Amazon VPC. Untuk mempelajari cara membuat titik akses, lihat Membuat titik akses dalam Panduan Developer Amazon VPC. Untuk mempelajari cara membuat titik akhir gateway untuk Amazon S3, lihat Titik akhir VPC Gateway.

Konfigurasikan layanan metadata instans

Pastikan instans Anda dapat melakukan hal berikut:

- Mengakses layanan metadata instans menggunakan Instance Metadata Service Version 2 (IMDSv2).
- Memungkinkan komunikasi keluar melalui port 80 (HTTP) ke alamat IP tautan IMDS.
- Meminta metadata instans dari `http://169.254.169.254`, tautan IMDSv2.

Untuk informasi selengkapnya, lihat Gunakan IMDSv2 dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Bawa Media Sendiri dengan RDS Custom for SQL Server

RDS Custom for SQL Server mendukung dua model lisensi: Lisensi Tercakup (LI, License Included) dan Bawa Media Sendiri (BYOM, Bring Your Own Media).

Dengan BYOM, Anda dapat melakukan hal-hal berikut:

1. Menyediakan dan menginstal file biner Microsoft SQL Server Anda sendiri dengan pembaruan kumulatif (CU) yang didukung pada AWS EC2 Windows AMI.
2. Menyimpan AMI sebagai citra emas, yakni templat yang dapat Anda gunakan untuk membuat versi mesin kustom (CEV).
3. Membuat CEV dari citra emas Anda.
4. Membuat instans basis data RDS Custom for SQL Server baru dengan menggunakan CEV Anda.

Amazon RDS kemudian mengelola instans basis data Anda untuk Anda.

> **Note**
>
> Jika memiliki juga instans basis data RDS Custom for SQL Server dengan Lisensi Tercakup (LI), Anda tidak dapat menggunakan perangkat lunak SQL Server dari instans basis data dengan BYOM ini. Anda harus membawa file biner SQL Server Anda sendiri ke BYOM.

Persyaratan untuk BYOM bagi RDS Custom for SQL Server

Persyaratan umum yang sama untuk versi mesin kustom dengan RDS Custom for SQL Server juga berlaku untuk BYOM. Untuk informasi selengkapnya, lihat Persyaratan untuk CEV RDS Custom for SQL Server.

Saat BYOM digunakan, pastikan bahwa Anda memenuhi persyaratan tambahan berikut:

- Gunakan salah satu edisi yang didukung berikut: SQL Server 2022 atau 2019 Enterprise, Standard, atau edisi Developer.
- Berikan privilese peran server sysadmin (SA) SQL Server kepada `NT AUTHORITY\SYSTEM`.
- Jaga agar Windows Server OS dikonfigurasikan dengan waktu `UTC`.

Instans Amazon EC2 Windows diatur secara bawaan ke zona waktu UTC. Lihat informasi yang lebih lengkap tentang cara melihat dan mengubah waktu untuk instans Windows di Mengatur waktu untuk instans Windows.

- Buka TCP porta 1433 dan UDP porta 1434 untuk memungkinkan koneksi SSM.

Keterbatasan BYOM untuk RDS Custom for SQL Server

Keterbatasan umum yang sama untuk RDS Custom for SQL Server juga berlaku untuk BYOM. Untuk informasi selengkapnya, lihat Persyaratan dan batasan untuk Amazon RDS Custom for SQL Server.

Dengan BYOM, keterbatasan tambahan berikut berlaku:

- Hanya instans SQL Server bawaan (MSSQLSERVER) yang didukung. Instans SQL Server bernama tidak didukung. RDS Custom for SQL Server mendeteksi dan memantau hanya instans SQL Server bawaan.
- Hanya instalasi tunggal SQL Server yang didukung pada setiap AMI. Lebih dari satu instalasi berbagai versi SQL Server tidak didukung.
- Edisi SQL Server Web tidak didukung dengan BYOM.
- Versi evaluasi edisi SQL Server tidak didukung dengan BYOM. Saat Anda menginstal SQL Server, jangan centang kotak untuk menggunakan versi evaluasi.
- Ketersediaan dan dukungan fitur bervariasi di antara versi-versi spesifik setiap mesin basis data, dan di antara Wilayah AWS. Lihat informasi yang lebih lengkap di Ketersediaan wilayah untuk CEV RDS Custom for SQL Server dan Dukungan versi untuk CEV RDS Custom for SQL Server.

Membuat instans basis data RDS Custom for SQL Server dengan BYOM

Lihat cara menyiapkan dan membuat instans basis data RDS Custom for SQL Server dengan BYOM di Mempersiapkan CEV menggunakan Bawa Media Anda Sendiri (BYOM).

Menggunakan versi mesin kustom untuk RDS Custom for SQL Server

Versi mesin kustom (CEV) untuk RDS Custom for SQL Server adalah Amazon Machine Image (AMI) yang mencakup Microsoft SQL Server.

Langkah-langkah dasar alur kerja CEV adalah sebagai berikut:

1. Pilih AMI Windows AWS EC2 untuk digunakan sebagai citra dasar bagi CEV. Anda memiliki opsi untuk menggunakan Microsoft SQL Server yang sudah diinstal sebelumnya atau membawa media Anda sendiri untuk menginstal SQL Server sendiri.
2. Instal perangkat lunak lain pada sistem operasi (OS) serta sesuaikan konfigurasi OS dan SQL Server untuk memenuhi kebutuhan perusahaan Anda.
3. Simpan AMI sebagai citra emas
4. Buat versi mesin kustom (CEV) dari citra emas Anda.
5. Buat instans DB RDS Custom for SQL Server baru dengan menggunakan CEV Anda.

Amazon RDS kemudian mengelola instans DB tersebut untuk Anda.

CEV memungkinkan Anda mempertahankan konfigurasi acuan OS dan basis data pilihan Anda. Menggunakan CEV memastikan bahwa konfigurasi host, seperti instalasi agen pihak ketiga atau kustomisasi OS lainnya, dipertahankan pada instans DB RDS Custom for SQL Server. Dengan CEV, Anda dapat dengan cepat menyebarkan armada instans DB RDS Custom for SQL Server dengan konfigurasi yang sama.

Topik

Persiapan membuat CEV untuk RDS Custom for SQL Server

Anda dapat membuat CEV menggunakan Amazon Machine Image (AMI) yang berisi Microsoft SQL Server Termasuk Lisensi (LI) yang sudah diinstal sebelumnya atau dengan AMI tempat Anda menginstal media instalasi SQL Server sendiri (BYOM).

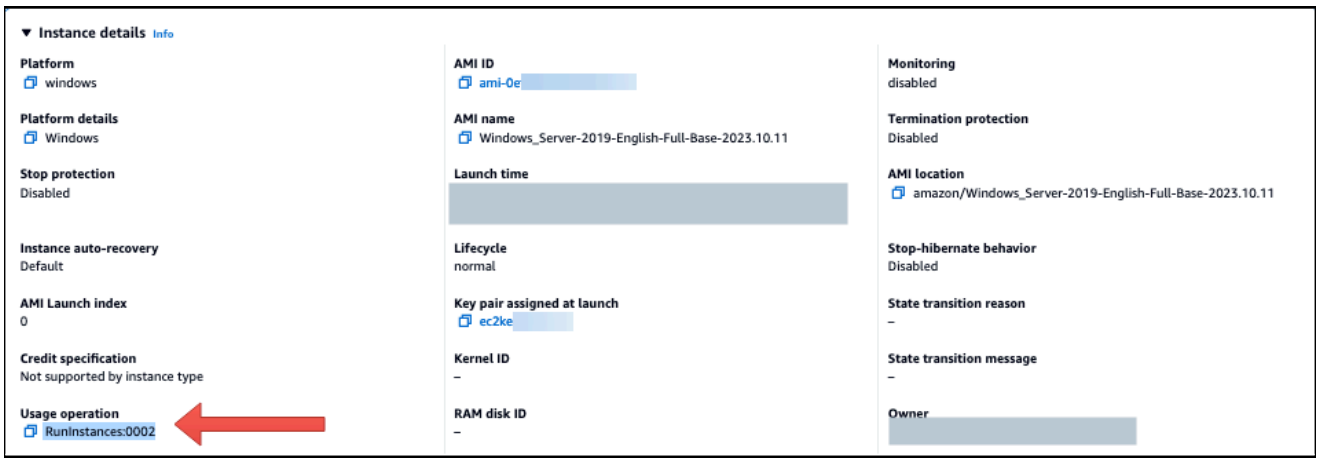
Daftar Isi

Mempersiapkan CEV menggunakan Bawa Media Anda Sendiri (BYOM)

Langkah-langkah berikut menggunakan AMI dengan Windows Server 2019 Base sebagai contoh.

Cara membuat CEV menggunakan BYOM

1. Pada konsol Amazon EC2, pilih Luncurkan Instans.
2. Untuk Nama, masukkan nama instans.
3. Di bagian Mulai Cepat, pilih Windows.
4. Pilih Microsoft Windows Server 2019 Base.
5. Pilih jenis instans yang sesuai, pasangan kunci, pengaturan jaringan dan penyimpanan, lalu luncurkan instans.
6. Setelah meluncurkan atau membuat instans EC2, pastikan AMI Windows yang benar dipilih dari Langkah 4:
    a. Pilih instans EC2 di konsol Amazon EC2.
    b. Di bagian Detail, periksa operasi Penggunaan dan pastikan bahwa itu diatur ke: 0002RunInstances.



7. Masuk ke instans EC2 dan salin media instalasi SQL Server Anda ke instans.

> **Note**
>
> Jika Anda membangun CEV menggunakan edisi SQL Server Developer, Anda mungkin perlu mendapatkan media instalasi menggunakan langganan Microsoft Visual Studio Anda.

8. Instal SQL Server. Pastikan Anda melakukan hal berikut:

   a. Tinjau Persyaratan untuk BYOM bagi RDS Custom for SQL Server dan Dukungan versi untuk CEV RDS Custom for SQL Server.

   b. Atur direktori root instans ke default `C:\Program Files\Microsoft SQL Server\`. Jangan ubah direktori ini.

   c. Atur SQL Server Database Engine Account Name ke `NT Service\MSSQLSERVER` atau `NT AUTHORITY\NETWORK SERVICE`.

   d. Atur mode Startup SQL Server ke Manual.

   e. Pilih mode Autentikasi SQL Server sebagai Mixed.

   f. Biarkan pengaturan saat ini untuk direktori Data default dan lokasi TempDB.

9. Berikan hak istimewa peran server sysadmin (SA) SQL Server untuk `NT AUTHORITY\SYSTEM`:

```
USE [master]
GO
EXEC master..sp_addsrvrolemember @loginame = N'NT AUTHORITY\SYSTEM' , @rolename =
 N'sysadmin'
```

```
GO
```

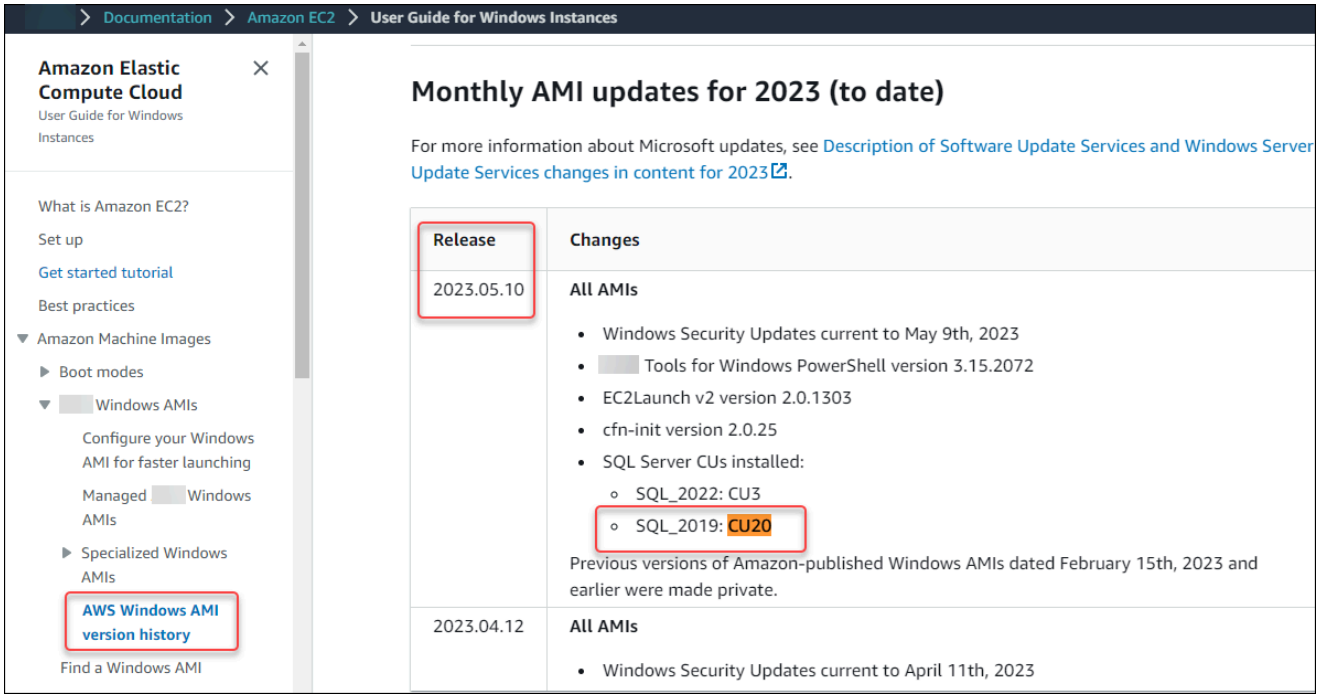
10. Instal perangkat lunak tambahan atau sesuaikan konfigurasi OS dan basis data untuk memenuhi kebutuhan Anda.
11. Jalankan Sysprep pada instans EC2. Untuk informasi selengkapnya, lihat Membuat Amazon Machine Image (AMI) terstandardisasi menggunakan Sysprep.
12. Simpan AMI berisi versi SQL Server yang diinstal, perangkat lunak lain, dan penyesuaian. Ini akan menjadi citra emas Anda.
13. Buat CEV baru dengan memberikan ID AMI dari gambar yang Anda buat. Untuk langkah mendetail, lihat Membuat CEV untuk RDS Custom for SQL Server.
14. Buat RDS Custom baru untuk instans DB SQL Server menggunakan CEV. Untuk langkah mendetail, lihat Membuat instans DB RDS Custom for SQL Server dari CEV.

Mempersiapkan CEV menggunakan SQL Server (LI) yang sudah diinstal sebelumnya

Langkah-langkah berikut untuk membuat CEV menggunakan Microsoft SQL Server (LI) yang sudah diinstal sebelumnya menggunakan AMI dengan SQL Server CU20 nomor Rilis `2023.05.10` sebagai contoh. Ketika Anda membuat CEV, pilih AMI dengan nomor rilis terbaru. Hal ini memastikan bahwa Anda menggunakan versi Windows Server dan SQL Server yang didukung dengan Pembaruan Kumulatif (CU) terbaru.

Cara membuat CEV menggunakan Microsoft SQL Server (LI) yang sudah diinstal sebelumnya

1. Pilih AWS EC2 Windows Amazon Machine Image (AMI) terbaru yang tersedia dengan Microsoft Windows Server Termasuk Lisensi (LI) dan SQL Server.
   a. Cari CU20 dalam riwayat versi AMI Windows.
   b. Perhatikan nomor Rilis. Untuk SQL Server 2019 CU20, nomor rilisnya adalah `2023.05.10`.



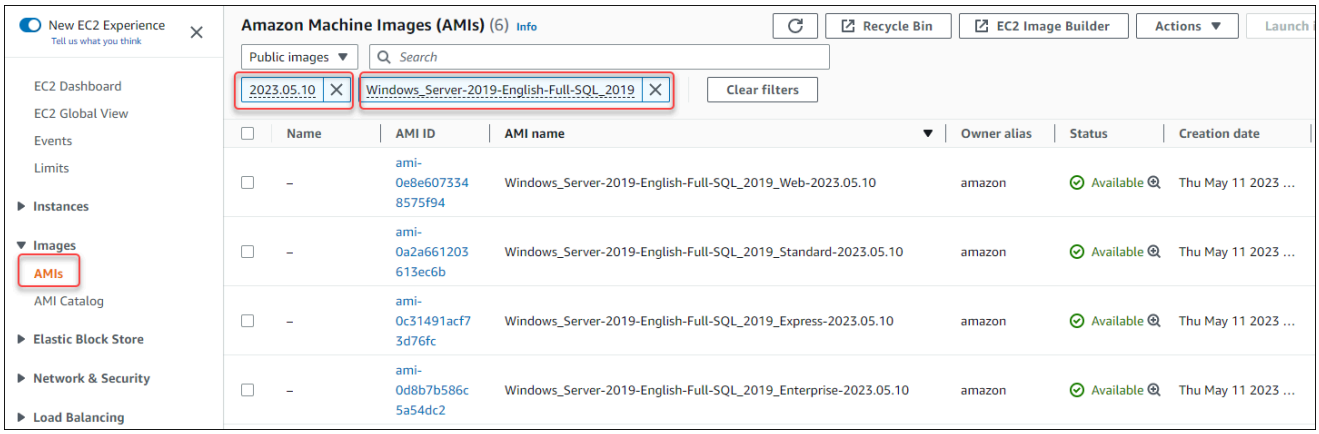
c. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.

d. Di panel navigasi kiri konsol Amazon EC2, pilih Gambar, lalu AMI.

e. Pilih Gambar publik.

f. Masukkan `2023.05.10` ke kotak pencarian. Daftar AMI akan muncul.

g. Masukkan `Windows_Server-2019-English-Full-SQL_2019` ke kotak pencarian untuk memfilter hasilnya. Hasil berikut akan muncul.



h. Pilih AMI dengan edisi SQL Server yang ingin Anda gunakan.

2. Buat atau luncurkan instans EC2 dari AMI pilihan Anda.

3. Masuk ke instans EC2 dan instal perangkat lunak tambahan atau sesuaikan konfigurasi OS dan basis data untuk memenuhi kebutuhan Anda.

4. Jalankan Sysprep pada instans EC2. Untuk informasi selengkapnya tentang menyiapkan AMI menggunakan Sysprep, lihat Membuat Amazon Machine Image (AMI) terstandardisasi menggunakan Sysprep.
5. Simpan AMI berisi versi SQL Server yang diinstal, perangkat lunak lain, dan penyesuaian. Ini akan menjadi citra emas Anda.
6. Buat CEV baru dengan memberikan ID AMI dari gambar yang Anda buat. Untuk langkah-langkah mendetail tentang membuat CEV, lihat Membuat CEV untuk RDS Custom for SQL Server.
7. Buat RDS Custom baru untuk instans DB SQL Server menggunakan CEV. Untuk langkah mendetail, lihat Membuat instans DB RDS Custom for SQL Server dari CEV.

Ketersediaan wilayah untuk CEV RDS Custom for SQL Server

Dukungan versi mesin kustom (CEV) untuk RDS Custom for SQL Server tersedia di Wilayah AWS berikut:

- AS Timur (Ohio)
- AS Timur (Virginia Utara)
- US West (Oregon)
- Asia Pasifik (Mumbai)
- Asia Pasifik (Seoul)
- Asia Pasifik (Singapura)
- Asia Pasifik (Sydney)
- Asia Pasifik (Tokyo)
- Kanada (Pusat)
- Eropa (Frankfurt)
- Eropa (Irlandia)
- Europe (London)
- Europe (Stockholm)
- Amerika Selatan (Sao Paulo)

Dukungan versi untuk CEV RDS Custom for SQL Server

Pembuatan CEV untuk RDS Custom for SQL Server didukung untuk AMI Windows AWS EC2 berikut:

- Untuk CEV yang menggunakan media pra-instal, AMI Windows AWS EC2 dengan Lisensi Termasuk (LI) Microsoft Windows Server 2019 (OS) dan SQL Server 2022 atau 2019
- Untuk CEV yang menggunakan bring your own media (BYOM), AWS EC2 Windows AMI dengan Microsoft Windows Server 2019 (OS)

Pembuatan CEV untuk RDS Custom for SQL Server didukung untuk edisi sistem operasi (OS) dan basis data berikut:

- Untuk CEV yang menggunakan media pra-instal:
  - SQL Server 2022 dengan CU9, untuk edisi Enterprise, Standard, dan Web
  - SQL Server 2019 dengan CU17, CU18, CU20, dan CU22, untuk edisi Enterprise, Standard, dan Web
- Untuk CEV yang menggunakan bring your own media (BYOM):
  - SQL Server 2022 dengan CU9, untuk edisi Enterprise, Standard, dan Developer
  - SQL Server 2019 dengan CU17, CU18, CU20, dan CU22, untuk edisi Enterprise, Standard, dan Developer
- Untuk CEV yang menggunakan media yang sudah diinstal sebelumnya atau bawa media Anda sendiri (BYOM), Windows Server 2019 adalah satu-satunya OS yang didukung

Persyaratan untuk CEV RDS Custom for SQL Server

Persyaratan berikut berlaku saat membuat CEV untuk RDS Custom for SQL Server:

- AMI yang digunakan untuk membuat CEV harus didasarkan pada konfigurasi OS dan basis data yang didukung oleh RDS Custom for SQL Server. Untuk informasi selengkapnya tentang konfigurasi yang didukung, lihat Persyaratan dan batasan untuk Amazon RDS Custom for SQL Server.
- CEV harus memiliki nama yang unik. Anda tidak dapat membuat CEV dengan nama yang sama seperti CEV yang sudah ada.
- Anda harus memberi nama CEV menggunakan pola penamaan SQL Server yaitu versi utama + versi minor + string yang disesuaikan. Versi utama + versi minor harus sesuai dengan versi SQL Server yang disediakan dengan AMI. Misalnya, Anda dapat memberi nama AMI dengan SQL Server 2019 CU17 sebagai 15.00.4249.2.my_cevtest.

- Anda harus menyiapkan AMI menggunakan Sysprep. Untuk informasi selengkapnya tentang menyiapkan AMI menggunakan Sysprep, lihat Membuat Amazon Machine Image (AMI) terstandardisasi menggunakan Sysprep.
- Anda bertanggung jawab untuk menjaga siklus hidup AMI. Instans DB RDS Custom for SQL Server yang dibuat dari CEV tidak menyimpan salinan AMI. Instans ini mempertahankan pointer ke AMI yang Anda gunakan untuk membuat CEV. AMI harus ada untuk instans DB RDS Custom for SQL Server agar tetap dapat dioperasikan.

Batasan untuk CEV RDS Custom for SQL Server

Batasan berikut berlaku untuk versi mesin kustom dengan RDS Custom for SQL Server:

- Anda tidak dapat menghapus CEV jika ada sumber daya, seperti instans DB atau snapshot DB, yang terkait dengannya.
- Untuk membuat instans DB RDS Custom for SQL Server, CEV harus memiliki status `pending-validation`, `available`, `failed`, atau `validating`. Anda tidak dapat membuat instans DB RDS Custom for SQL Server menggunakan CEV jika status CEV `incompatible-image-configuration`.
- Untuk memodifikasi instans DB RDS Custom for SQL Server agar dapat menggunakan CEV baru, CEV harus berstatus `available`.
- Anda tidak dapat membuat AMI atau CEV dari instans DB RDS Custom for SQL Server yang ada.
- Anda tidak dapat memodifikasi CEV yang ada untuk menggunakan AMI yang berbeda. Namun, Anda dapat memodifikasi instans DB RDS Custom for SQL Server agar menggunakan CEV yang berbeda. Untuk informasi selengkapnya, lihat Memodifikasi instans DB RDS Custom for SQL Server.
- Salinan CEV lintas wilayah tidak didukung.
- Salinan CEV lintas akun tidak didukung.
- Anda tidak dapat mengembalikan atau memulihkan CEV setelah menghapusnya. Namun, Anda dapat membuat CEV baru dari AMI yang sama.
- Instans DB RDS Custom for SQL Server menyimpan file basis data SQL Server Anda dalam drive D:\. AMI yang terkait dengan CEV harus menyimpan file basis data sistem Microsoft SQL Server di drive C:\.
- Instans DB RDS Custom for SQL Server mempertahankan perubahan konfigurasi yang dibuat ke SQL Server. Semua perubahan konfigurasi untuk OS pada instans DB RDS Custom for SQL Server yang sedang berjalan yang dibuat dari CEV tidak dipertahankan. Jika Anda perlu membuat

perubahan konfigurasi permanen ke OS dan mempertahankannya sebagai konfigurasi dasar baru, buat CEV baru dan modifikasi instans DB untuk menggunakan CEV baru.

> **Important**
>
> Memodifikasi instans DB RDS Custom for SQL Server untuk menggunakan CEV baru adalah operasi offline. Anda dapat langsung melakukan modifikasi atau menjadwalkannya agar dilakukan saat jendela pemeliharaan mingguan.

- Saat Anda memodifikasi CEV, Amazon RDS tidak mendorong modifikasi tersebut ke instans DB RDS Custom for SQL Server terkait. Anda harus memodifikasi setiap instans DB RDS Custom for SQL Server untuk menggunakan CEV baru atau yang diperbarui. Untuk informasi selengkapnya, lihat Memodifikasi instans DB RDS Custom for SQL Server.
-

> **Important**
>
> Jika AMI yang digunakan oleh CEV dihapus, modifikasi apa pun yang mungkin memerlukan penggantian host, misalnya komputasi skala, akan gagal. Instans DB RDS Custom for SQL Server kemudian akan ditempatkan di luar perimeter dukungan RDS. Sebaiknya Anda menghindari menghapus AMI apa pun yang terkait dengan CEV.

Membuat CEV untuk RDS Custom for SQL Server

Anda dapat membuat versi mesin kustom (CEV) menggunakan AWS Management Console atau AWS CLI. Kemudian, Anda dapat menggunakan CEV tersebut untuk membuat instans DB RDS Custom for SQL Server.

Pastikan bahwa Amazon Machine Image (AMI) berada di akun dan Wilayah AWS yang sama dengan CEV Anda. Jika tidak, proses untuk membuat CEV akan gagal.

Untuk informasi selengkapnya, lihat Membuat dan menghubungkan ke instans DB untuk Amazon RDS Custom for SQL Server.

> **Important**
>
> Langkah-langkah untuk membuat CEV sama untuk AMI yang dibuat dengan SQL Server yang sudah diinstal sebelumnya dan yang dibuat menggunakan bawa media Anda sendiri (BYOM).

Konsol

Cara membuat CEV

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Versi mesin kustom.

   Halaman Versi mesin kustom menampilkan semua CEV yang ada saat ini. Jika Anda belum membuat CEV, tabel akan kosong.
3. Pilih Buat versi mesin kustom.
4. Untuk Jenis mesin, pilih Microsoft SQL Server.
5. Untuk Edisi, pilih edisi mesin DB yang ingin Anda gunakan.
6. Untuk Versi utama, pilih versi mesin utama yang diinstal pada AMI Anda.
7. Di Detail versi, masukkan nama yang valid dalam Nama versi mesin kustom.

   Format namanya adalah *major-engine-version*.*minor-engine-version*.*customized_string*. Anda dapat menggunakan 1-50 karakter alfanumerik, garis bawah, tanda hubung, dan titik. Misalnya, Anda dapat memasukkan nama **15.00.4249.2.my_cevtest**.

   Jika ingin, masukkan Deskripsi untuk CEV Anda.
8. Untuk Media Instalasi, telusuri atau masukkan ID AMI yang ingin Anda buat menjadi CEV.
9. Di bagian Tag, tambahkan tag apa pun untuk mengidentifikasi CEV.
10. Pilih Buat versi mesin kustom.

Halaman Versi mesin kustom muncul. CEV Anda ditampilkan dengan status pending-validation

AWS CLI

Untuk membuat CEV dengan menggunakanAWS CLI, jalankan perintah create-custom-db-engine-version.

Opsi berikut diperlukan:

- `--engine`
- `--engine-version`

- `--image-id`

Anda juga dapat menentukan parameter berikut:

- `--description`
- `--region`
- `--tags`

Contoh berikut membuat CEV bernama `15.00.4249.2.my_cevtest`. Pastikan bahwa nama CEV Anda dimulai dengan nomor versi mesin utama.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds create-custom-db-engine-version \
    --engine custom-sqlserver-ee \
    --engine-version 15.00.4249.2.my_cevtest \
    --image-id ami-0r93cx31t5r596482 \
    --description "Custom SQL Server EE 15.00.4249.2 cev test"
```

Output parsial berikut menunjukkan mesin, grup parameter, dan informasi lainnya.

```
"DBEngineVersions": [
    {
    "Engine": "custom-sqlserver-ee",
    "MajorEngineVersion": "15.00",
    "EngineVersion": "15.00.4249.2.my_cevtest",
    "DBEngineDescription": "Microsoft SQL Server Enterprise Edition for RDS Custom for
 SQL Server",
    "DBEngineVersionArn": "arn:aws:rds:us-east-1:<my-account-id>:cev:custom-sqlserver-
ee/15.00.4249.2.my_cevtest/a1234a1-123c-12rd-bre1-1234567890",
    "DBEngineVersionDescription": "Custom SQL Server EE 15.00.4249.2 cev test",

    "Image": [
        "ImageId": "ami-0r93cx31t5r596482",
        "Status": "pending-validation"
     ],
    "CreateTime": "2022-11-20T19:30:01.831000+00:00",
```

```
    "SupportsLogExportsToCloudwatchLogs": false,
    "SupportsReadReplica": false,
    "Status": "pending-validation",
    "SupportsParallelQuery": false,
    "SupportsGlobalDatabases": false,
    "TagList": []
    }
]
```

Jika proses untuk membuat CEV gagal, RDS Custom for SQL Server mengeluarkan `RDS-EVENT-0198` dengan pesan `Creation failed for custom engine version` *`major-engine-version.cev_name`*. Pesan tersebut mencakup detail tentang kegagalan, misalnya, peristiwa mencetak file yang tidak ada. Untuk menemukan ide pemecahan masalah dalam pembuatan CEV, lihat Memecahkan masalah kesalahan CEV untuk RDS Custom for SQL Server.

Membuat instans DB RDS Custom for SQL Server dari CEV

Setelah Anda berhasil membuat CEV, status CEV akan menunjukkan `pending-validation`. Kini Anda dapat membuat RDS Custom baru untuk instans DB SQL Server menggunakan CEV. Untuk membuat instans DB RDS Custom for SQL Server baru dari CEV, lihat Membuat instans DB untuk RDS Custom for SQL Server.

Siklus hidup CEV

Siklus hidup CEV mencakup status-status berikut.

Status CEV	Deskripsi	Saran pemecahan masalah
`pending-validation`	CEV telah dibuat dan sedang menunggu validasi AMI terkait. CEV akan tetap dalam status `pending-validation` sampai instans DB RDS Custom	Jika tidak ada tugas, buat instans DB RDS Custom for SQL Server baru dari CEV. Saat membuat instans DB RDS Custom for SQL Server, sistem mencoba memvalidasi AMI terkait untuk CEV.

Status CEV	Deskripsi	Saran pemecahan masalah	
	for SQL Server dibuat darinya.		
`validating`	Tugas pembuatan untuk instans DB RDS Custom for SQL Server berdasarkan CEV baru sedang berlangsung. Saat membuat instans RDS Custom for SQL Server DB, sistem mencoba memvalidasi AMI CEV terkait.	Tunggu tugas pembuatan instans DB RDS Custom for SQL Server yang ada selesai. Anda dapat menggunakan konsol RDS EVENTS untuk meninjau pesan peristiwa terperinci untuk pemecahan masalah.	
`available`	CEV berhasil divalidasi. CEV akan memasuki status `available` setelah instans DB RDS Custom for SQL Server telah berhasil dibuat darinya.	CEV tidak memerlukan validasi tambahan. CEV dapat digunakan untuk membuat instans DB RDS Custom for SQL Server tambahan atau memodifikasi yang sudah ada.	

Status CEV	Deskripsi	Saran pemecahan masalah	
`inactive`	Status CEV telah diubah menjadi tidak aktif.	Anda tidak dapat membuat atau meningkatkan instans DB RDS Custom dengan CEV ini. Anda juga tidak dapat memulihkan snapshot DB untuk membuat instans DB RDS Custom baru dengan CEV ini. Untuk informasi tentang cara mengubah status menjadi `ACTIVE`, lihat Memodifikasi CEV untuk RDS Custom for SQL Server.	
`failed`	Langkah membuat instans DB gagal untuk CEV ini sebelum dapat memvalidasi AMI. Selain itu, status AMI yang mendasari yang digunakan oleh CEV tidak tersedia.	Lakukan pemecahan masalah akar penyebab sistem tidak dapat membuat instans DB. Lihat detail pesan kesalahan dan coba buat instans DB baru lagi. Pastikan bahwa status AMI yang mendasari yang digunakan oleh CEV tersedia.	

Status CEV	Deskripsi	Saran pemecahan masalah	
`incompatible-image-configuration`	Terjadi kesalahan saat memvalidasi AMI.	Lihat detail teknis kesalahan. Anda tidak dapat mencoba memvalidasi AMI dengan CEV ini lagi. Tinjau rekomendasi berikut: • Pastikan CEV Anda diberi nama menggunakan pola penamaan SQL Server yang diperlukan yaitu versi utama + versi minor + string yang disesuaikan. • Pastikan versi SQL Server dalam nama CEV cocok dengan versi yang disediakan dengan AMI. • Pastikan versi build OS memenuhi versi build minimum yang diperlukan. • Pastikan versi utama OS memenuhi versi utama minimum yang diperlukan. Buat CEV baru menggunakan informasi yang benar. Jika diperlukan, buat instans EC2 baru menggunakan AMI yang didukung dan jalankan proses Sysprep pada instans.	

Memodifikasi CEV untuk RDS Custom for SQL Server

Anda dapat mengubah CEV menggunakan AWS Management Console atau AWS CLI. Anda dapat mengubah deskripsi CEV atau status ketersediaannya. CEV Anda memiliki salah satu dari nilai status berikut:

- `available` – Anda dapat menggunakan CEV ini untuk membuat instans DB RDS Custom baru atau meningkatkan instans DB. Ini adalah status default untuk CEV yang baru dibuat.
- `inactive` – Anda tidak dapat membuat atau meningkatkan instans DB RDS Custom dengan CEV ini. Anda tidak dapat memulihkan snapshot DB untuk membuat instans DB RDS Custom baru dengan CEV ini.

Anda dapat mengubah status CEV dari `available` ke `inactive` atau dari `inactive` ke `available`. Anda dapat mengubah status ke `INACTIVE` untuk mencegah penggunaan CEV yang tidak disengaja atau membuat CEV yang dihentikan memenuhi syarat untuk digunakan lagi.

Konsol

Cara mengubah CEV

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Versi mesin kustom.
3. Pilih CEV yang deskripsi atau statusnya ingin Anda ubah.
4. Untuk Tindakan, pilih Ubah.
5. Lakukan salah satu dari perubahan berikut:
   - Untuk Pengaturan status CEV, pilih status ketersediaan baru.
   - Untuk Deskripsi versi, masukkan deskripsi baru.
6. Pilih Ubah CEV.

   Jika CEV sedang digunakan, konsol akan menampilkan Anda tidak dapat mengubah status CEV. Perbaiki masalah, lalu coba lagi.

Halaman Versi mesin kustom muncul.

AWS CLI

Untuk memodifikasi CEV dengan menggunakanAWS CLI, jalankan perintah modify-custom-db-engine-version. Anda dapat menemukan CEV untuk dimodifikasi dengan menjalankan perintah. describe-db-engine-versions

Opsi berikut diperlukan:

- `--engine`
- `--engine-version` *`cev`*, dengan *`cev`* adalah nama versi mesin kustom yang ingin Anda modifikasi
- `--status` *`status`*, dengan *`status`* adalah status ketersediaan yang ingin Anda tetapkan ke CEV

Contoh berikut mengubah CEV bernama `15.00.4249.2.my_cevtest` dari statusnya saat ini menjadi`inactive`.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds modify-custom-db-engine-version \
    --engine custom-sqlserver-ee \
    --engine-version 15.00.4249.2.my_cevtest \
    --status inactive
```

Untuk Windows:

```
aws rds modify-custom-db-engine-version ^
    --engine custom-sqlserver-ee ^
    --engine-version 15.00.4249.2.my_cevtest ^
    --status inactive
```

Memodifikasi instans DB RDS Custom for SQL Server untuk menggunakan CEV baru

Anda dapat memodifikasi instans DB RDS Custom for SQL Server yang ada agar menggunakan CEV yang berbeda. Perubahan yang dapat Anda lakukan meliputi:

- Mengubah CEV
- Mengubah kelas instans DB
- Mengubah periode retensi cadangan dan waktu pencadangan
- Mengubah waktu pemeliharaan

Konsol

Cara memodifikasi instans DB RDS Custom for SQL Server

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data.
3. Pilih instans DB yang ingin Anda modifikasi.
4. Pilih Ubah.

5. Lakukan perubahan berikut sesuai kebutuhan:

    a. Untuk Versi mesin DB, pilih CEV yang berbeda.

    b. Ubah nilai untuk Kelas instans DB. Untuk kelas yang didukung, lihat Dukungan kelas instans DB untuk RDS Custom for SQL Server.

    c. Ubah nilai untuk Periode retensi cadangan.

    d. Untuk Jendela cadangan, tetapkan nilai untuk Waktu mulai dan Durasi.

    e. Untuk Jendela pemeliharaan instans DB, tetapkan nilai untuk Hari mulai, Waktu mulai, dan Durasi.

6. Pilih Lanjutkan.
7. Pilih Terapkan segera atau Terapkan di jendela pemeliharaan terjadwal berikutnya.
8. Pilih Ubah instans DB.

> **Note**
>
> Saat memodifikasi instans DB dari satu CEV ke CEV lain, misalnya, saat meningkatkan versi minor, basis data sistem SQL Server, termasuk data dan konfigurasinya, dipertahankan dari instans DB RDS Custom for SQL Server saat ini.

AWS CLI

Untuk memodifikasi instance DB untuk menggunakan CEV yang berbeda dengan menggunakanAWS CLI, jalankan modify-db-instanceperintah.

Opsi berikut diperlukan:

- `--db-instance-identifier`
- `--engine-version` *`cev`*, dengan *`cev`* adalah nama versi mesin kustom yang ingin Anda gunakan untuk instans DB.

Contoh berikut memodifikasi instans DB bernama `my-cev-db-instance` untuk menggunakan CEV bernama `15.00.4249.2.my_cevtest_new` dan menerapkan perubahan segera.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-cev-db-instance \
    --engine-version 15.00.4249.2.my_cevtest_new \
    --apply-immediately
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-cev-db-instance ^
    --engine-version 15.00.4249.2.my_cevtest_new ^
    --apply-immediately
```

Melihat detail CEV untuk Amazon RDS Custom for SQL Server

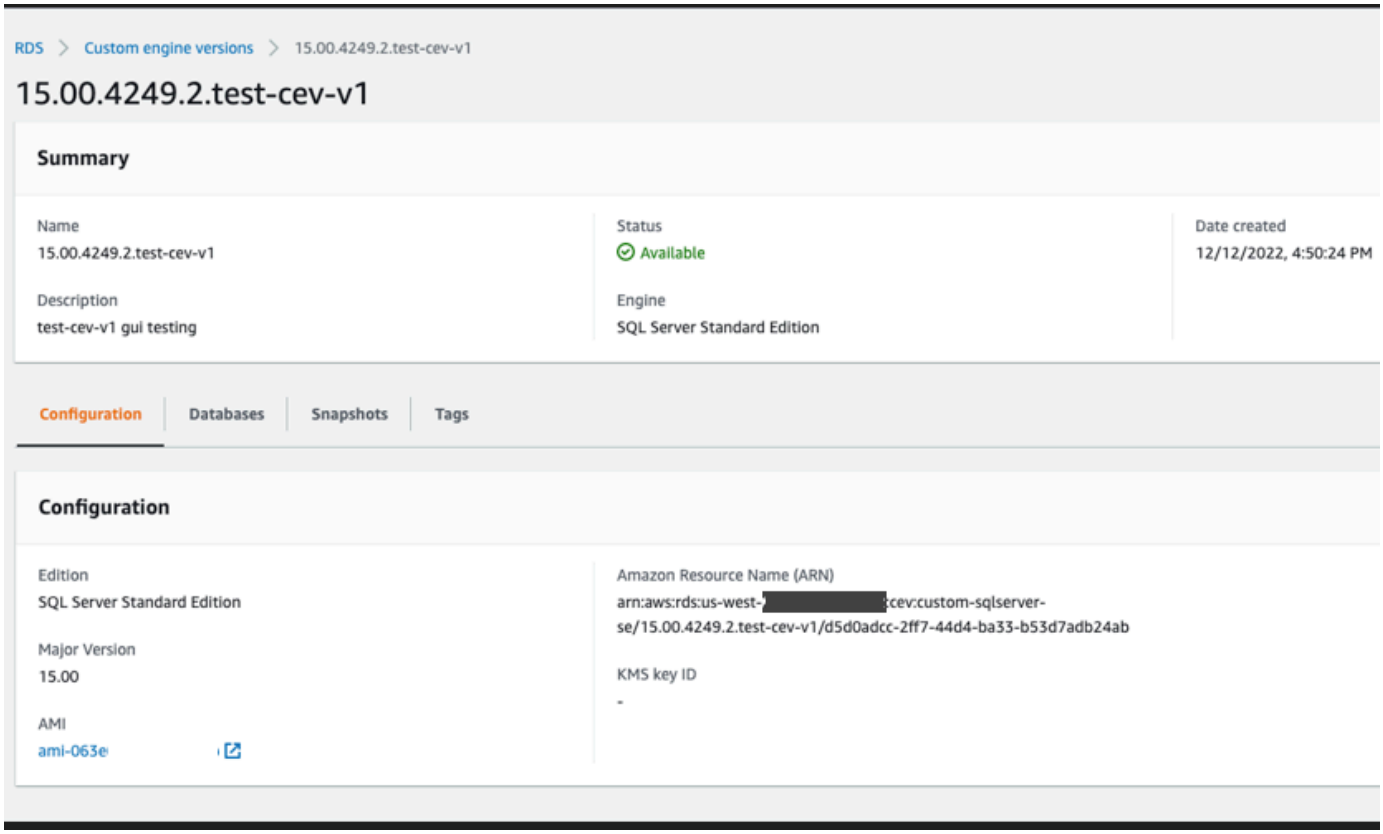
Anda dapat melihat detail CEV dengan menggunakan AWS Management Console atau AWS CLI.

Konsol

Untuk melihat detail CEV

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Versi mesin kustom.

   Halaman Versi mesin kustom menampilkan semua CEV yang saat ini ada. Jika Anda belum membuat CEV sama sekali, halaman akan kosong.
3. Pilih nama CEV yang ingin Anda lihat.
4. Pilih Konfigurasi untuk melihat detailnya.



AWS CLI

Untuk melihat detail CEV dengan menggunakan AWS CLI, jalankan perintah describe-db-engine-versions.

Anda juga dapat menentukan opsi-opsi berikut:

- `--include-all`, untuk melihat semua CEV dengan keadaan siklus hidup apa pun. Tanpa opsi `--include-all`, hanya CEV dalam keadaan siklus hidup `available` yang akan dihasilkan.

```
aws rds describe-db-engine-versions --engine custom-sqlserver-ee --engine-version
 15.00.4249.2.my_cevtest --include-all
{
    "DBEngineVersions": [
        {
            "Engine": "custom-sqlserver-ee",
            "MajorEngineVersion": "15.00",
            "EngineVersion": "15.00.4249.2.my_cevtest",
            "DBParameterGroupFamily": "custom-sqlserver-ee-15.0",
```

```
            "DBEngineDescription": "Microsoft SQL Server Enterprise Edition for custom
 RDS",
            "DBEngineVersionArn": "arn:aws:rds:us-east-1:{my-account-id}:cev:custom-
sqlserver-ee/15.00.4249.2.my_cevtest/a1234a1-123c-12rd-bre1-1234567890",
            "DBEngineVersionDescription": "Custom SQL Server EE 15.00.4249.2 cev test",
            "Image": {
                "ImageId": "ami-0r93cx31t5r596482",
                "Status": "pending-validation"
            },
            "DBEngineMediaType": "AWS Provided",
            "CreateTime": "2022-11-20T19:30:01.831000+00:00",
            "ValidUpgradeTarget": [],
            "SupportsLogExportsToCloudwatchLogs": false,
            "SupportsReadReplica": false,
            "SupportedFeatureNames": [],
            "Status": "pending-validation",
            "SupportsParallelQuery": false,
            "SupportsGlobalDatabases": false,
            "TagList": [],
            "SupportsBabelfish": false
        }
    ]
}
```

Anda dapat menggunakan filter untuk melihat CEV dengan status siklus hidup tertentu. Misalnya, untuk melihat CEV yang memiliki status siklus hidup `pending-validation`, `available`, atau`failed`:

```
aws rds describe-db-engine-versions engine custom-sqlserver-ee
                region us-west-2 include-all query 'DBEngineVersions[?Status ==
 pending-validation ||
                Status == available || Status == failed]'
```

Menghapus CEV untuk RDS Custom for SQL Server

Anda dapat menghapus CEV menggunakan AWS Management Console atau AWS CLI. Biasanya, tugas ini membutuhkan waktu beberapa menit.

Sebelum menghapus CEV, pastikan CEV tidak digunakan oleh salah satu dari berikut ini:

- Instans DB RDS Custom
- Snapshot instans DB RDS Custom

- Cadangan otomatis instans DB RDS Custom

Konsol

Cara menghapus CEV

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Versi mesin kustom.
3. Pilih CEV yang deskripsi atau statusnya ingin Anda hapus.
4. Untuk Tindakan, pilih Hapus.

   Kotak dialog Hapus ***cev_name?*** muncul.
5. Masukkan **`delete me`**, lalu pilih Hapus.

   Di halaman Versi mesin kustom, banner menunjukkan bahwa CEV Anda sedang dihapus.

AWS CLI

Untuk menghapus CEV dengan menggunakanAWS CLI, jalankan perintah delete-custom-db-engine-version.

Opsi berikut diperlukan:

- `--engine custom-sqlserver-ee`
- `--engine-version` *`cev`*, dengan *`cev`* adalah nama versi mesin kustom yang akan dihapus

Contoh berikut menghapus CEV bernama `15.00.4249.2.my_cevtest`.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds delete-custom-db-engine-version \
    --engine custom-sqlserver-ee \
    --engine-version 15.00.4249.2.my_cevtest
```

Untuk Windows:

```
aws rds delete-custom-db-engine-version ^
    --engine custom-sqlserver-ee ^
    --engine-version 15.00.4249.2.my_cevtest
```

Membuat dan menghubungkan ke instans DB untuk Amazon RDS Custom for SQL Server

Anda dapat membuat instans DB RDS Custom, kemudian menghubungkannya menggunakan AWS Systems Manager atau Remote Desktop Protocol (RDP).

> **Important**
>
> Sebelum dapat membuat atau terhubung ke instans DB RDS Custom for SQL Server, pastikan untuk menyelesaikan tugas dalam Menyiapkan lingkungan Anda untuk Amazon RDS Custom for SQL Server.
> Anda dapat menandai instans DB RDS Custom saat membuatnya, tetapi jangan membuat atau memodifikasi tag `AWSRDSCustom` yang diperlukan untuk otomatisasi RDS Custom. Untuk informasi selengkapnya, lihat Menandai sumber daya RDS Custom for SQL Server.
> Saat pertama kali membuat instans DB RDS Custom for SQL Server, Anda mungkin menerima kesalahan berikut: Peran terkait layanan sedang dalam proses pembuatan. Coba lagi nanti. Jika ya, tunggu beberapa menit dan coba buat instans DB lagi.

Topik

- Membuat instans DB untuk RDS Custom for SQL Server
- Peran tertaut layanan RDS Custom
- Menghubungkan ke instans RDS Custom DB Anda menggunakan AWS Systems Manager
- Menghubungkan ke instans RDS Custom DB Anda menggunakan RDP

Membuat instans DB untuk RDS Custom for SQL Server

Buat instans DB Amazon RDS Custom for SQL Server menggunakan AWS Management Console atau AWS CLI. Prosedurnya mirip dengan prosedur untuk membuat instans DB Amazon RDS.

Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.

Konsol

Cara membuat instans DB RDS Custom for SQL Server

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.

2. Di panel navigasi, pilih Basis Data.
3. Pilih Buat basis data.
4. Pilih Pembuatan Standar untuk metode pembuatan basis data.
5. Untuk Opsi mesin, pilih Microsoft SQL Server untuk jenis mesin.
6. Untuk Jenis manajemen basis data, pilih Amazon RDS Custom.
7. Di bagian Edisi, pilih edisi mesin DB yang ingin Anda gunakan.
8. (Opsional) Jika Anda bermaksud membuat instans DB dari CEV, centang kotak centang Gunakan versi mesin kustom (CEV). Pilih CEV Anda di daftar drop-down.
9. Untuk versi Database, pertahankan versi nilai default.
10. Untuk Templat, pilih Produksi.
11. Di bagian Pengaturan, masukkan nama unik untuk Pengidentifikasi instans DB.
12. Untuk memasukkan kata sandi master, lakukan hal berikut:

    a. Di bagian Pengaturan, buka Pengaturan Kredensial.

    b. Hapus kotak centang Buat kata sandi secara otomatis.

    c. Ubah nilai Nama pengguna master dan masukkan kata sandi yang sama di Kata sandi master dan Konfirmasi kata sandi.

    Secara default, instans DB RDS Custom baru menggunakan kata sandi yang dihasilkan secara otomatis untuk pengguna master.

13. Di bagian Ukuran instans DB, pilih nilai untuk Kelas instans DB.

    Untuk kelas yang didukung, lihat Dukungan kelas instans DB untuk RDS Custom for SQL Server.

14. Pilih pengaturan Penyimpanan.
15. Untuk Keamanan RDS Custom, lakukan hal berikut:

    a. Untuk Profil instans IAM, pilih profil instans untuk instans DB RDS Custom for SQL Server Anda.

    Profil instans IAM harus dimulai dengan `AWSRDSCustom`, misalnya *`AWSRDSCustomInstanceProfileForRdsCustomInstance`*.

    b. Untuk Enkripsi, pilih Masukkan kunci ARN untuk mencantumkan kunci AWS KMS yang tersedia. Lalu pilih kunci Anda dari daftar.

Kunci AWS KMS diperlukan untuk RDS Custom. Untuk informasi selengkapnya, lihat Pastikan Anda memiliki kunci enkripsi simetris AWS KMS.

16. Untuk bagian yang tersisa, tentukan pengaturan instans DB RDS Custom pilihan Anda. Untuk informasi tentang setiap pengaturan, lihat Pengaturan untuk instans DB. Pengaturan berikut tidak muncul di konsol dan tidak didukung:

    - Fitur prosesor
    - Penskalaan otomatis penyimpanan
    - Ketersediaan & daya tahan
    - Opsi Autentikasi kata sandi dan Kerberos dalam Autentikasi basis data (hanya Autentikasi kata sandi yang didukung)
    - Grup Opsi basis data dalam Konfigurasi tambahan
    - Wawasan Kinerja
    - Ekspor log
    - Aktifkan peningkatan versi minor otomatis
    - Perlindungan penghapusan

    Periode retensi cadangan didukung, tetapi Anda tidak dapat memilih 0 hari.

17. Pilih Buat basis data.

    Tombol Lihat detail kredensial muncul di halaman Basis data.

    Untuk melihat nama pengguna dan kata sandi master untuk instans DB RDS Custom, pilih Lihat detail kredensial.

    Untuk terhubung ke instans DB sebagai pengguna master, gunakan nama pengguna dan kata sandi yang muncul.

    > **Important**
    >
    > Anda tidak dapat melihat kata sandi pengguna master lagi. Jika tidak mencatatnya, Anda mungkin harus mengubahnya. Untuk mengubah kata sandi pengguna master setelah instans DB RDS Custom tersedia, ubah instans DB. Untuk informasi selengkapnya

tentang cara mengubah instans DB, lihat Mengelola instans DB Amazon RDS Custom for SQL Server.

18. Pilih Basis data untuk melihat daftar instans DB RDS Custom.
19. Pilih instans DB RDS Custom yang baru saja Anda buat.

    Pada konsol RDS, detail untuk instans DB RDS Custom baru muncul:

    - Instans DB akan berstatus membuat hingga instans DB RDS Custom selesai dibuat dan siap digunakan. Saat statusnya berubah menjadi tersedia, Anda dapat terhubung ke instans DB. Bergantung pada kelas instans dan penyimpanan yang dialokasikan, perlu waktu beberapa menit agar instans DB baru tersedia.
    - Peran memiliki nilai Instans (RDS Custom).
    - Mode otomatisasi RDS Custom memiliki nilai Otomatisasi penuh. Pengaturan ini berarti bahwa instans DB menyediakan pemantauan otomatis dan pemulihan instans.

AWS CLI

Anda membuat instance RDS Custom DB dengan menggunakan create-db-instanceAWS CLIperintah.

Opsi berikut diperlukan:

- `--db-instance-identifier`
- `--db-instance-class` (untuk daftar kelas instans yang didukung, lihat Dukungan kelas instans DB untuk RDS Custom for SQL Server)
- `--engine` (`custom-sqlserver-ee`, `custom-sqlserver-se`, atau `custom-sqlserver-web`)
- `--kms-key-id`
- `--custom-iam-instance-profile`

Contoh berikut membuat instans DB RDS Custom for SQL Server bernama `my-custom-instance`. Periode retensi cadangan adalah 3 hari.

> **Note**
>
> Untuk membuat instans DB dari versi mesin kustom (CEV), berikan nama CEV yang ada ke parameter `--engine-version`. Misalnya, `--engine-version 15.00.4249.2.my_cevtest`

Example

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-instance \
    --engine custom-sqlserver-ee \
    --engine-version 15.00.4073.23.v1 \
    --db-instance-identifier my-custom-instance \
    --db-instance-class db.m5.xlarge \
    --allocated-storage 20 \
    --db-subnet-group mydbsubnetgroup \
    --master-username myuser \
    --master-user-password mypassword \
    --backup-retention-period 3 \
    --no-multi-az \
    --port 8200 \
    --kms-key-id mykmskey \
    --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance
```

Untuk Windows:

```
aws rds create-db-instance ^
    --engine custom-sqlserver-ee ^
    --engine-version 15.00.4073.23.v1 ^
    --db-instance-identifier my-custom-instance ^
    --db-instance-class db.m5.xlarge ^
    --allocated-storage 20 ^
    --db-subnet-group mydbsubnetgroup ^
    --master-username myuser ^
    --master-user-password mypassword ^
    --backup-retention-period 3 ^
    --no-multi-az ^
    --port 8200 ^
    --kms-key-id mykmskey ^
```

```
    --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance
```

> **Note**
>
> Tentukan kata sandi selain perintah yang ditampilkan di sini sebagai praktik keamanan terbaik.

Dapatkan detail tentang instans Anda menggunakan perintah `describe-db-instances`.

```
aws rds describe-db-instances --db-instance-identifier my-custom-instance
```

Output parsial berikut menunjukkan mesin, grup parameter, dan informasi lainnya.

```
{
    "DBInstances": [
        {
            "PendingModifiedValues": {},
            "Engine": "custom-sqlserver-ee",
            "MultiAZ": false,
            "DBSecurityGroups": [],
            "DBParameterGroups": [
                {
                    "DBParameterGroupName": "default.custom-sqlserver-ee-15",
                    "ParameterApplyStatus": "in-sync"
                }
            ],
            "AutomationMode": "full",
            "DBInstanceIdentifier": "my-custom-instance",
            "TagList": []
        }
    ]
}
```

Peran tertaut layanan RDS Custom

Peran tertaut layanan memberi Amazon RDS Custom akses ke sumber daya di Akun AWS Anda. Hal ini membuat penggunaan RDS Custom lebih mudah karena Anda tidak perlu menambahkan izin yang diperlukan secara manual. RDS Custom menetapkan izin perannya yang tertaut layanan, dan kecuali ditetapkan lain, hanya RDS Custom yang dapat mengambil perannya. Izin yang ditentukan

mencakup kebijakan kepercayaan dan kebijakan izin, dan kebijakan izin tersebut tidak dapat dilampirkan ke entitas IAM lainnya.

Saat Anda membuat instans DB RDS Custom, peran terkait layanan Amazon RDS dan RDS Custom dibuat (jika belum ada) dan digunakan. Untuk informasi selengkapnya, lihat Menggunakan peran terkait layanan untuk Amazon RDS.

Saat pertama kali membuat instans DB RDS Custom for SQL Server, Anda mungkin menerima kesalahan berikut: Peran terkait layanan sedang dalam proses pembuatan. Coba lagi nanti. Jika ya, tunggu beberapa menit dan coba buat instans DB lagi.

Menghubungkan ke instans RDS Custom DB Anda menggunakan AWS Systems Manager

Setelah membuat instans DB RDS Custom, Anda dapat terhubung ke instans tersebut menggunakan Session Manager AWS Systems Manager. Session Manager adalah kapabilitas Systems Manager terkelola penuh yang memungkinkan Anda mengelola instans Amazon EC2 melalui shell berbasis browser atau melalui AWS CLI. Untuk informasi selengkapnya, lihat AWS Systems Manager Session Manager.

Konsol

Cara terhubung ke instans DB Anda menggunakan Session Manager

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data, kemudian pilih instans DB RDS Custom tempat Anda ingin terhubung.
3. Pilih Konfigurasi.
4. Catat nilai ID Sumber Daya untuk instans DB Anda. Misalnya, ID sumber daya mungkin `db-ABCDEFGHIJKLMNOPQRS0123456`.
5. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
6. Di panel navigasi, pilih Instans.
7. Cari nama instans EC2 Anda, lalu pilih ID instans yang terkait dengannya. Misalnya, ID instans mungkin `i-abcdefghijklm01234`.
8. Pilih Hubungkan.

9. Pilih Session Manager.
10. Pilih Hubungkan.

    Sebuah jendela terbuka untuk sesi Anda.

AWS CLI

Anda dapat terhubung ke instans DB RDS Custom menggunakan AWS CLI. Teknik ini membutuhkan plugin Session Manager untuk AWS CLI. Untuk mempelajari cara menginstal plugin, lihat Instal plugin Session Manager untuk AWS CLI.

Untuk menemukan ID sumber daya DB dari instans DB RDS Custom Anda, gunakan `describe-db-instances`.

```
aws rds describe-db-instances \
    --query 'DBInstances[*].[DBInstanceIdentifier,DbiResourceId]' \
    --output text
```

Output sampel berikut menunjukkan ID sumber daya untuk instans RDS Custom Anda. Prefiksnya adalah `db-`.

```
db-ABCDEFGHIJKLMNOPQRS0123456
```

Untuk menemukan ID instans EC2 dari instans DB Anda, gunakan `aws ec2 describe-instances`. Contoh berikut menggunakan `db-ABCDEFGHIJKLMNOPQRS0123456` untuk ID sumber daya.

```
aws ec2 describe-instances \
    --filters "Name=tag:Name,Values=db-ABCDEFGHIJKLMNOPQRS0123456" \
    --output text \
    --query 'Reservations[*].Instances[*].InstanceId'
```

Output sampel berikut menunjukkan ID instans EC2.

```
i-abcdefghijklm01234
```

Gunakan perintah `aws ssm start-session` yang menyediakan ID instans EC2 dalam parameter `--target`.

```
aws ssm start-session --target "i-abcdefghijklm01234"
```

Koneksi yang berhasil terlihat seperti berikut.

```
Starting session with SessionId: yourid-abcdefghijklm1234
[ssm-user@ip-123-45-67-89 bin]$
```

Menghubungkan ke instans RDS Custom DB Anda menggunakan RDP

Setelah membuat instans DB RDS Custom, Anda dapat terhubung ke instans tersebut menggunakan klien RDP. Prosedurnya sama dengan menghubungkan ke instans Amazon EC2. Untuk informasi selengkapnya, lihat Terhubung ke instans Windows Anda.

Untuk terhubung ke instans DB, Anda memerlukan pasangan kunci yang terkait dengan instans. RDS Custom membuat pasangan kunci untuk Anda. Nama pasangan menggunakan prefiks`do-not-delete-rds-custom-`*`DBInstanceIdentifier`*. AWS Secrets Manager menyimpan kunci privat Anda sebagai rahasia.

Selesaikan tugas dalam langkah-langkah berikut:

1. Konfigurasi instans DB Anda untuk memungkinkan koneksi RDP.
2. Ambil kunci rahasia Anda.
3. Terhubung ke instans EC2 Anda menggunakan utilitas RDP.

Konfigurasi instans DB Anda untuk memungkinkan koneksi RDP

Untuk mengizinkan koneksi RDP, konfigurasikan grup keamanan VPC Anda dan tetapkan aturan firewall pada host.

Konfigurasi grup keamanan VPC

Pastikan grup keamanan VPC yang terkait dengan instans DB Anda mengizinkan koneksi masuk pada port 3389 untuk Transmission Control Protocol (TCP). Untuk mempelajari cara mengonfigurasi grup keamanan VPC, lihat Konfigurasikan grup keamanan VPC.

Tetapkan aturan firewall pada host

Untuk mengizinkan koneksi masuk pada port 3389 untuk TCP, tetapkan aturan firewall pada host. Contoh berikut menunjukkan cara melakukan hal ini.

Sebaiknya Anda menggunakan nilai `-Profile` spesifik: `Public`, `Private`, atau `Domain`. Menggunakan `Any` mengacu pada ketiga nilai. Anda juga dapat menentukan kombinasi nilai yang dipisahkan dengan koma. Untuk informasi selengkapnya tentang menyetel aturan firewall, lihat Mengatur- NetFirewallRule dalam dokumentasi Microsoft.

Cara menggunakan Systems Manager Session Manager untuk menetapkan aturan firewall

1. Hubungkan ke Session Manager seperti yang ditunjukkan di Menghubungkan ke instans RDS Custom DB Anda menggunakan AWS Systems Manager.
2. Jalankan perintah berikut.

```
Set-NetFirewallRule -DisplayName "Remote Desktop - User Mode (TCP-In)" -Direction
 Inbound -LocalAddress Any -Profile Any
```

Cara menggunakan perintah CLI Systems Manager untuk menetapkan aturan firewall

1. Gunakan perintah berikut untuk membuka RDP pada host.

```
OPEN_RDP_COMMAND_ID=$(aws ssm send-command --region $AWS_REGION \
    --instance-ids $RDS_CUSTOM_INSTANCE_EC2_ID \
    --document-name "AWS-RunPowerShellScript" \
    --parameters '{"commands":["Set-NetFirewallRule -DisplayName \"Remote Desktop -
 User Mode (TCP-In)\" -Direction Inbound -LocalAddress Any -Profile Any"]}' \
    --comment "Open RDP port" | jq -r ".Command.CommandId")
```

2. Gunakan ID perintah yang ditampilkan dalam output untuk mendapatkan status perintah sebelumnya. Untuk menggunakan kueri berikut guna menampilkan ID perintah, pastikan bahwa Anda telah menginstal plug-in jq.

```
aws ssm list-commands \
    --region $AWS_REGION \
    --command-id $OPEN_RDP_COMMAND_ID
```

Ambil kunci rahasia Anda

Ambil kunci rahasia Anda menggunakan AWS Management Console atau AWS CLI.

Konsol

Cara mengambil kunci rahasia

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data, kemudian pilih instans DB RDS Custom tempat Anda ingin terhubung.
3. Pilih tab Konfigurasi.
4. Perhatikan ID instans DB untuk instans DB Anda, misalnya, *my-custom-instance*.
5. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
6. Di panel navigasi, pilih Instans.
7. Cari nama instans EC2 Anda, lalu pilih ID instans yang terkait dengannya.

   Dalam contoh ini, ID instans adalah `i-abcdefghijklm01234`.
8. Di bagian Detail, temukan Nama pasangan kunci. Nama pasangan mencakup pengidentifikasi DB. Dalam contoh ini, nama pasangan adalah `do-not-delete-rds-custom-`*`my-custom-instance`*`-`*`0d726c`*.
9. Dalam ringkasan instans, temukan DNS IPv4 Publik. Misalnya, DNS publik mungkin adalah `ec2-12-345-678-901.us-east-2.compute.amazonaws.com`.
10. Buka AWS Secrets Manager konsol di https://console.aws.amazon.com/secretsmanager/.
11. Pilih rahasia yang bernama sama dengan pasangan kunci Anda.
12. Pilih Ambil nilai rahasia.

AWS CLI

Cara mengambil kunci privat

1. Dapatkan daftar instans DB RDS Custom Anda dengan memanggil perintah `aws rds describe-db-instances`.

```
aws rds describe-db-instances \
    --query 'DBInstances[*].[DBInstanceIdentifier,DbiResourceId]' \
    --output text
```

2. Pilih pengidentifikasi instans DB dari output sampel, misalnya `do-not-delete-rds-custom-`*`my-custom-instance`*.
3. Temukan ID instans EC2 dari instans DB Anda dengan memanggil perintah `aws ec2 describe-instances`. Contoh berikut menggunakan nama instans EC2 untuk mendeskripsikan instans DB.

```
aws ec2 describe-instances \
    --filters "Name=tag:Name,Values=do-not-delete-rds-custom-my-custom-instance" \
    --output text \
    --query 'Reservations[*].Instances[*].InstanceId'
```

Output sampel berikut menunjukkan ID instans EC2.

```
i-abcdefghijklm01234
```

4. Temukan nama kunci dengan menentukan ID instans EC2, seperti yang ditunjukkan pada contoh berikut.

```
aws ec2 describe-instances \
    --instance-ids i-abcdefghijklm01234 \
    --output text \
    --query 'Reservations[*].Instances[*].KeyName'
```

Output sampel berikut menunjukkan nama kunci yang menggunakan prefiks `do-not-delete-rds-custom-`*`DBInstanceIdentifier`*.

```
do-not-delete-rds-custom-my-custom-instance-0d726c
```

Terhubung ke instans EC2 Anda menggunakan utilitas RDP

Ikuti prosedur di Hubungkan ke instans Windows Anda menggunakan RDP di Panduan Pengguna Amazon EC2 untuk Instans Windows. Prosedur ini mengasumsikan bahwa Anda membuat file .pem yang berisi kunci privat Anda.

Mengelola instans DB Amazon RDS Custom for SQL Server

Amazon RDS Custom for SQL Server mendukung subset tugas manajemen biasa untuk instans DB Amazon RDS. Di bawah ini, Anda dapat menemukan petunjuk untuk tugas manajemen RDS Custom for SQL Server yang didukung menggunakan AWS Management Console dan AWS CLI.

Topik

Menjeda dan melanjutkan otomatisasi RDS Custom

Secara otomatis, RDS Custom menyediakan pemantauan dan pemulihan instans untuk instans DB RDS Custom for SQL Server. Jika Anda perlu menyesuaikan instans, lakukan hal berikut:

1. Jeda otomatisasi RDS Custom untuk periode tertentu. Jeda memastikan bahwa penyesuaian Anda tidak mengganggu otomatisasi RDS Custom.
2. Sesuaikan instans DB RDS Custom for SQL Server sesuai kebutuhan.
3. Lakukan salah satu dari langkah berikut:
   - Lanjutkan otomatisasi secara manual.
   - Tunggu hingga periode jeda berakhir. Dalam hal ini, RDS Custom melanjutkan pemantauan dan pemulihan instans secara otomatis.

> **Important**
>
> Menjeda dan melanjutkan otomatisasi adalah satu-satunya tugas otomatisasi yang didukung saat memodifikasi instans DB RDS Custom for SQL Server.

Konsol

Cara menjeda atau melanjutkan otomatisasi RDS Custom

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data, lalu pilih instans DB RDS Custom yang ingin Anda ubah.
3. Pilih Ubah. Halaman Modifikasi instans DB muncul.
4. Untuk Mode otomatisasi RDS Custom, pilih salah satu opsi berikut:
   - Dijeda akan menjeda pemantauan dan pemulihan instans untuk instans DB RDS Custom. Masukkan durasi jeda yang Anda inginkan (dalam hitungan menit) untuk Durasi mode otomatisasi. Nilai minimum adalah 60 menit (default). Nilai maksimum adalah 1.440 menit.
   - Otomatisasi penuh akan melanjutkan otomatisasi.
5. Pilih Lanjutkan untuk memeriksa ringkasan perubahan.

   Sebuah pesan menunjukkan bahwa RDS Custom akan segera menerapkan perubahan.
6. Jika perubahan Anda benar, pilih Modifikasi instans DB. Anda juga dapat memilih Kembali untuk mengedit perubahan atau Batal untuk membatalkan perubahan.

   Pada konsol RDS, detail untuk modifikasi muncul. Jika Anda menjeda otomatisasi, Status instans DB RDS Custom Anda menunjukkan Otomatisasi dihentikan sementara.
7. (Opsional) Di panel navigasi, pilih Basis Data, lalu pilih instans DB RDS Custom.

   Di panel Ringkasan, Mode otomatisasi RDS Custom menunjukkan status otomatisasi. Jika otomatisasi dijeda, nilainya adalah Dihentikan sementara. Otomatisasi dilanjutkan dalam ***hitungan*** menit.

AWS CLI

Untuk menjeda atau melanjutkan otomatisasi kustom RDS, gunakan perintah. `modify-db-instance` AWS CLI Identifikasi instans DB menggunakan parameter `--db-instance-identifier` yang diperlukan. Kontrol mode otomatisasi dengan parameter berikut:

- `--automation-mode` menentukan status jeda instans DB. Nilai yang valid adalah `all-paused` yang menghentikan otomatisasi, dan `full` yang melanjutkannya.

- `--resume-full-automation-mode-minutes` menentukan durasi jeda. Nilai default adalah 60 menit.

> **Note**
>
> Terlepas dari apakah Anda menentukan `--no-apply-immediately` atau `--apply-immediately`, RDS Custom menerapkan modifikasi secara asinkron sesegera mungkin.

Dalam respons perintah, `ResumeFullAutomationModeTime` menunjukkan waktu untuk melanjutkan dalam timestamp UTC. Saat mode otomatisasi adalah `all-paused`, Anda dapat menggunakan `modify-db-instance` untuk melanjutkan mode otomatisasi atau memperpanjang periode jeda. Tidak ada opsi `modify-db-instance` lain yang didukung.

Contoh berikut menjeda otomatisasi `my-custom-instance` selama 90 menit.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --automation-mode all-paused \
    --resume-full-automation-mode-minutes 90
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --automation-mode all-paused ^
    --resume-full-automation-mode-minutes 90
```

Contoh berikut memperpanjang durasi jeda selama 30 menit. 30 menit ditambahkan ke waktu awal yang ditunjukkan dalam `ResumeFullAutomationModeTime`.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
```

```
    --automation-mode all-paused \
    --resume-full-automation-mode-minutes 30
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --automation-mode all-paused ^
    --resume-full-automation-mode-minutes 30
```

Contoh berikut melanjutkan otomatisasi penuh untuk `my-custom-instance`.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --automation-mode full \
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --automation-mode full
```

Dalam output sampel parsial berikut, nilai `AutomationMode` yang tertunda adalah `full`.

```
{
    "DBInstance": {
        "PubliclyAccessible": true,
        "MasterUsername": "admin",
        "MonitoringInterval": 0,
        "LicenseModel": "bring-your-own-license",
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "0123456789abcdefg"
            }
        ],
        "InstanceCreateTime": "2020-11-07T19:50:06.193Z",
        "CopyTagsToSnapshot": false,
```

```
        "OptionGroupMemberships": [
            {
                "Status": "in-sync",
                "OptionGroupName": "default:custom-oracle-ee-19"
            }
        ],
        "PendingModifiedValues": {
            "AutomationMode": "full"
        },
        "Engine": "custom-oracle-ee",
        "MultiAZ": false,
        "DBSecurityGroups": [],
        "DBParameterGroups": [
            {
                "DBParameterGroupName": "default.custom-oracle-ee-19",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        ...
        "ReadReplicaDBInstanceIdentifiers": [],
        "AllocatedStorage": 250,
        "DBInstanceArn": "arn:aws:rds:us-west-2:012345678912:db:my-custom-instance",
        "BackupRetentionPeriod": 3,
        "DBName": "ORCL",
        "PreferredMaintenanceWindow": "fri:10:56-fri:11:26",
        "Endpoint": {
            "HostedZoneId": "ABCDEFGHIJKLMNO",
            "Port": 8200,
            "Address": "my-custom-instance.abcdefghijk.us-west-2.rds.amazonaws.com"
        },
        "DBInstanceStatus": "automation-paused",
        "IAMDatabaseAuthenticationEnabled": false,
        "AutomationMode": "all-paused",
        "EngineVersion": "19.my_cev1",
        "DeletionProtection": false,
        "AvailabilityZone": "us-west-2a",
        "DomainMemberships": [],
        "StorageType": "gp2",
        "DbiResourceId": "db-ABCDEFGHIJKLMNOPQRSTUVW",
        "ResumeFullAutomationModeTime": "2020-11-07T20:56:50.565Z",
        "KmsKeyId": "arn:aws:kms:us-west-2:012345678912:key/
aa111a11-111a-11a1-1a11-1111a11a1a1a",
        "StorageEncrypted": false,
        "AssociatedRoles": [],
```

```
        "DBInstanceClass": "db.m5.xlarge",
        "DbInstancePort": 0,
        "DBInstanceIdentifier": "my-custom-instance",
        "TagList": []
    }
```

Memodifikasi instans DB RDS Custom for SQL Server

Cara memodifikasi instans DB RDS Custom for SQL Server mirip dengan Amazon RDS, tetapi perubahan yang dapat Anda lakukan terbatas pada hal berikut:

- Mengubah kelas instans DB
- Mengubah periode retensi cadangan dan waktu pencadangan
- Mengubah waktu pemeliharaan
- Meningkatkan versi mesin DB saat versi baru tersedia
- Mengubah penyimpanan yang dialokasikan, IOPS yang tersedia, dan jenis penyimpanan
- Mengubah port basis data
- Mengubah pengidentifikasi instans DB
- Mengubah kredensial master
- Mengizinkan dan menghapus deployment Multi-AZ
- Mengizinkan akses publik
- Mengubah grup keamanan
- Mengubah grup subnet

Batasan berikut berlaku untuk modifikasi instans DB RDS Custom for SQL Server:

- Opsi DB kustom dan grup parameter tidak didukung.
- Setiap volume penyimpanan yang Anda lampirkan secara manual ke instans DB RDS Custom berada di luar perimeter dukungan.

  Untuk informasi selengkapnya, lihat Perimeter dukungan RDS Custom.

Konsol

Cara memodifikasi instans DB RDS Custom for SQL Server

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data.
3. Pilih instans DB yang ingin Anda ubah.
4. Pilih Ubah.
5. Lakukan perubahan berikut sesuai kebutuhan:
    a. Untuk Versi mesin DB, pilih versi baru.
    b. Ubah nilai untuk Kelas instans DB. Untuk kelas yang didukung, lihat Dukungan kelas instans DB untuk RDS Custom for SQL Server
    c. Ubah nilai untuk Periode retensi cadangan.
    d. Untuk Jendela cadangan, tetapkan nilai untuk Waktu mulai dan Durasi.
    e. Untuk Jendela pemeliharaan instans DB, tetapkan nilai untuk Hari mulai, Waktu mulai, dan Durasi.
6. Pilih Lanjutkan.
7. Pilih Terapkan segera atau Terapkan di jendela pemeliharaan terjadwal berikutnya.
8. Pilih Ubah instans DB.

AWS CLI

Untuk memodifikasi contoh RDS Custom untuk SQL Server DB, gunakan perintah. modify-db-instance AWS CLI Atur parameter berikut sesuai kebutuhan:

- `--db-instance-class` – Untuk kelas yang didukung, lihat Dukungan kelas instans DB untuk RDS Custom for SQL Server
- `--engine-version` – Nomor versi mesin basis data yang akan ditingkatkan.
- `--backup-retention-period` – Lama waktu mempertahankan cadangan otomatis, dari 0–35 hari.
- `--preferred-backup-window` – Rentang waktu harian selama cadangan otomatis dibuat.
- `--preferred-maintenance-window` – Rentang waktu mingguan (dalam UTC) pemeliharaan sistem dapat dilakukan.

- `--apply-immediately` – Gunakan `--apply-immediately` untuk langsung menerapkan perubahan penyimpanan.

  Gunakan `--no-apply-immediately` (default) untuk menerapkan perubahan saat jendela pemeliharaan berikutnya.

Memodifikasi penyimpanan untuk instans DB RDS Custom for SQL Server

Memodifikasi penyimpanan untuk instans DB RDS Custom for SQL Server mirip dengan memodifikasi penyimpanan untuk instans DB Amazon RDS, tetapi Anda hanya dapat melakukan hal berikut:

- Meningkatkan ukuran penyimpanan yang dialokasikan.
- Mengubah jenis penyimpanan. Anda dapat menggunakan jenis penyimpanan yang tersedia seperti Tujuan Umum atau IOPS yang Tersedia. IOPS yang disediakan didukung untuk jenis penyimpanan Block Express gp3, io1, dan io2.
- Ubah IOPS yang disediakan, jika Anda menggunakan tipe volume yang mendukung IOPS Terketentuan.

Batasan berikut berlaku saat memodifikasi penyimpanan untuk instans DB RDS Custom for SQL Server:

- Ukuran penyimpanan minimum yang dialokasikan untuk RDS Custom for SQL Server adalah 20 GiB dan ukuran penyimpanan maksimum yang didukung adalah 16 TiB.
- Seperti Amazon RDS, Anda tidak dapat mengurangi penyimpanan yang dialokasikan. Ini adalah batasan volume Amazon Elastic Block Store (Amazon EBS). Lihat informasi yang lebih lengkap di Menggunakan penyimpanan untuk instans DB Amazon RDS
- Penskalaan otomatis penyimpanan tidak didukung untuk instans DB RDS Custom for SQL Server.
- Volume penyimpanan apa pun yang Anda lampirkan secara manual ke instans DB RDS Custom Anda tidak dipertimbangkan untuk penskalaan penyimpanan. Hanya volume data default yang disediakan RDS, yaitu drive D, yang dipertimbangkan untuk penskalaan penyimpanan.

  Untuk informasi selengkapnya, lihat Perimeter dukungan RDS Custom.
- Menskalakan penyimpanan biasanya tidak menyebabkan pemadaman atau penurunan performa instans DB. Setelah Anda mengubah ukuran penyimpanan untuk instans DB, status instans DB adalah storage-optimization.

- Optimalisasi penyimpanan dapat membutuhkan waktu beberapa jam. Anda tidak dapat melakukan modifikasi penyimpanan lebih lanjut selama enam (6) jam atau hingga optimalisasi penyimpanan pada instans selesai, mana pun yang lebih lama. Lihat informasi yang lebih lengkap di Menggunakan penyimpanan untuk instans DB Amazon RDS

Untuk informasi selengkapnya tentang penyimpanan, lihat Penyimpanan instans DB Amazon RDS.

Untuk informasi umum tentang modifikasi penyimpanan, lihat Menggunakan penyimpanan untuk instans DB Amazon RDS.

Konsol

Cara mengubah penyimpanan untuk instans DB RDS Custom for SQL Server

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data.
3. Pilih instans DB yang ingin Anda ubah.
4. Pilih Ubah.
5. Lakukan perubahan berikut sesuai kebutuhan:
    a. Masukkan nilai baru untuk Penyimpanan yang dialokasikan. Nilai ini harus lebih besar dari nilai saat ini, dan antara 20 GiB–16 TiB.
    b. Ubah nilai untuk Jenis penyimpanan. Anda dapat memilih dari jenis penyimpanan IOPS Tujuan Umum atau Provisioned yang tersedia. IOPS yang disediakan didukung untuk jenis penyimpanan Block Express gp3, io1, dan io2.
    c. Jika Anda menentukan jenis penyimpanan yang mendukung IOPS Terketentuan, Anda dapat menentukan nilai IOPS yang Disediakan.
6. Pilih Lanjutkan.
7. Pilih Terapkan segera atau Terapkan di jendela pemeliharaan terjadwal berikutnya.
8. Pilih Ubah instans DB.

AWS CLI

Untuk memodifikasi penyimpanan untuk instans RDS Custom for SQL Server DB, gunakan perintah. modify-db-instance AWS CLI Atur parameter berikut sesuai kebutuhan:

- `--allocated-storage` – Jumlah penyimpanan yang akan dialokasikan untuk instans DB, dalam gibibyte. Nilai ini harus lebih besar dari nilai saat ini, dan antara 20–16.384 GiB.
- `--storage-type`— Jenis penyimpanan, misalnya, gp2, gp3, io1, atau io2.
- `--iops` – IOPS yang Tersedia untuk instans DB. Anda dapat menentukan ini hanya untuk jenis penyimpanan yang mendukung IOPS Terketentuan (gp3, io1, dan io2).
- `--apply-immediately` – Gunakan `--apply-immediately` untuk langsung menerapkan perubahan penyimpanan.

  Gunakan `--no-apply-immediately` (default) untuk menerapkan perubahan saat jendela pemeliharaan berikutnya.

Contoh berikut mengubah ukuran penyimpanan menjadi 200 GiB, tipe penyimpanan my-custom-instance menjadi io1, dan IOPS yang Disediakan menjadi 3000.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --storage-type io1 \
    --iops 3000 \
    --allocated-storage 200 \
    --apply-immediately
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --storage-type io1 ^
    --iops 3000 ^
    --allocated-storage 200 ^
    --apply-immediately
```

Menandai sumber daya RDS Custom for SQL Server

Anda dapat menandai sumber daya RDS Custom sama seperti sumber daya Amazon RDS, tetapi dengan beberapa perbedaan penting:

- Jangan membuat atau memodifikasi tag `AWSRDSCustom` yang diperlukan untuk otomatisasi RDS Custom. Jika melakukannya, Anda dapat merusak otomatisasi.
- Tag `Name` ditambahkan ke sumber daya RDS Custom dengan nilai prefiks `do-not-delete-rds-custom`. Setiap nilai yang diteruskan pelanggan untuk kunci akan ditimpa.
- Tag yang ditambahkan ke instans DB RDS Custom selama pembuatan disebarkan ke semua sumber daya RDS Custom terkait lainnya.
- Tag tidak disebarkan saat Anda menambahkannya ke sumber daya RDS Custom setelah pembuatan instans DB.

Untuk informasi umum tentang penandaan sumber daya, lihat Memberi tag pada sumber daya Amazon RDS.

Menghapus instans DB RDS Custom for SQL Server

Untuk menghapus instans DB RDS Custom for SQL Server, lakukan hal berikut:

- Berikan nama instans DB.
- Pilih atau hapus opsi untuk mengambil snapshot DB akhir dari instans DB.
- Pilih atau hapus opsi untuk mempertahankan cadangan otomatis.

Anda dapat menghapus instans DB RDS Custom for SQL Server menggunakan konsol atau CLI. Waktu yang diperlukan untuk menghapus instans DB dapat bervariasi bergantung pada periode retensi cadangan (yaitu, berapa banyak cadangan yang harus dihapus), berapa banyak data yang dihapus, dan apakah snapshot akhir diambil.

> **Warning**
>
> Menghapus instans DB RDS Custom for SQL Server akan menghapus instans EC2 dan volume Amazon EBS terkait secara permanen. Anda tidak boleh menghentikan atau menghapus sumber daya ini kapan saja karena dapat menyebabkan kegagalan penghapusan dan pembuatan snapshot akhir.

> **Note**
>
> Anda tidak dapat membuat snapshot DB akhir dari instans DB Anda jika memiliki status `creating`, `failed`, `incompatible-create`, `incompatible-restore`, atau

`incompatible-network`. Untuk informasi selengkapnya, lihat Melihat status instans DB Amazon RDS.

Important

Ketika memilih untuk mengambil snapshot akhir, sebaiknya Anda menghindari penulisan data ke instans DB saat penghapusan instans DB sedang berlangsung. Setelah penghapusan instans DB dimulai, perubahan data tidak dijamin akan ditangkap oleh snapshot akhir.

Konsol

Cara menghapus instans DB RDS Custom

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data, lalu pilih instans DB RDS Custom for SQL Server yang ingin Anda hapus. Instans DB RDS Custom for SQL Server menunjukkan peran Instans (RDS Custom for SQL Server).
3. Untuk Tindakan, pilih Hapus.
4. Untuk mengambil snapshot akhir, pilih Buat snapshot akhir dan berikan nama untuk Nama snapshot akhir.
5. Untuk mempertahankan cadangan otomatis, pilih Pertahankan cadangan otomatis.
6. Masukkan **`delete me`** dalam kotak.
7. Pilih Hapus.

AWS CLI

Anda menghapus instans RDS Custom untuk SQL Server DB dengan menggunakan perintah. delete-db-instance AWS CLI Identifikasi instans DB menggunakan parameter `--db-instance-identifier` yang diperlukan. Parameter yang tersisa sama dengan instans DB Amazon RDS.

Contoh berikut menghapus instans DB RDS Custom for SQL Server bernamamy`-custom-instance`, mengambil snapshot akhir, dan mempertahankan cadangan otomatis.

Example

Untuk Linux, macOS, atau Unix:

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

Untuk Windows:

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

Untuk mengambil snapshot akhir, opsi `--final-db-snapshot-identifier` diperlukan dan harus ditentukan.

Untuk melewati snapshot akhir, tentukan opsi `--skip-final-snapshot`, bukan opsi `--no-skip-final-snapshot` dan `--final-db-snapshot-identifier` dalam perintah.

Untuk menghapus cadangan otomatis, tentukan opsi `--delete-automated-backups`, bukan opsi `--no-delete-automated-backups` dalam perintah.

Memulai dan menghentikan instans DB RDS Custom for SQL Server

Anda dapat memulai dan menghentikan instans DB RDS Custom for SQL Server. Persyaratan dan batasan umum yang sama untuk instans DB RDS for SQL Server berlaku untuk menghentikan dan memulai instans RDS Custom for SQL Server DB Anda. Untuk informasi selengkapnya, lihat Menghentikan sementara instans DB Amazon RDS.

Pertimbangan berikut juga berlaku untuk memulai dan menghentikan instans DB RDS Custom for SQL Server Anda:

- Memodifikasi atribut instans EC2 dari instans DB RDS Custom for SQL Server saat instans DB `STOPPED` tidak didukung.

- Anda dapat menghentikan dan memulai instans DB RDS Custom for SQL Server hanya jika dikonfigurasi untuk Zona Ketersediaan tunggal. Anda tidak dapat menghentikan instans DB RDS Custom for SQL Server dalam konfigurasi Multi-AZ.
- Snapshot `SYSTEM` akan dibuat ketika Anda menghentikan instans DB RDS Custom for SQL Server. Snapshot akan dihapus secara otomatis ketika Anda memulai instans DB RDS Custom for SQL Server lagi.
- Jika Anda menghapus instans EC2 saat instans DB RDS Custom for SQL Server dihentikan, drive `C:` akan diganti ketika Anda memulai instans DB RDS Custom for SQL Server lagi.
- Drive `C:\`, nama host, dan konfigurasi kustom Anda tetap ada saat Anda menghentikan instans DB RDS Custom for SQL Server, selama Anda tidak memodifikasi jenis instans.
- Tindakan berikut akan mengakibatkan RDS Custom menempatkan instans DB di luar perimeter dukungan, dan Anda masih dikenakan biaya untuk jam instans DB:
  - Memulai instans EC2 yang mendasarinya saat Amazon RDS dihentikan. Untuk mengatasinya, Anda dapat memanggil API Amazon RDS `start-db-instance` atau menghentikan EC2 sehingga instans RDS Custom kembali `STOPPED`.
  - Menghentikan instans EC2 yang mendasari saat instans RDS Custom for SQL Server DB `ACTIVE`.

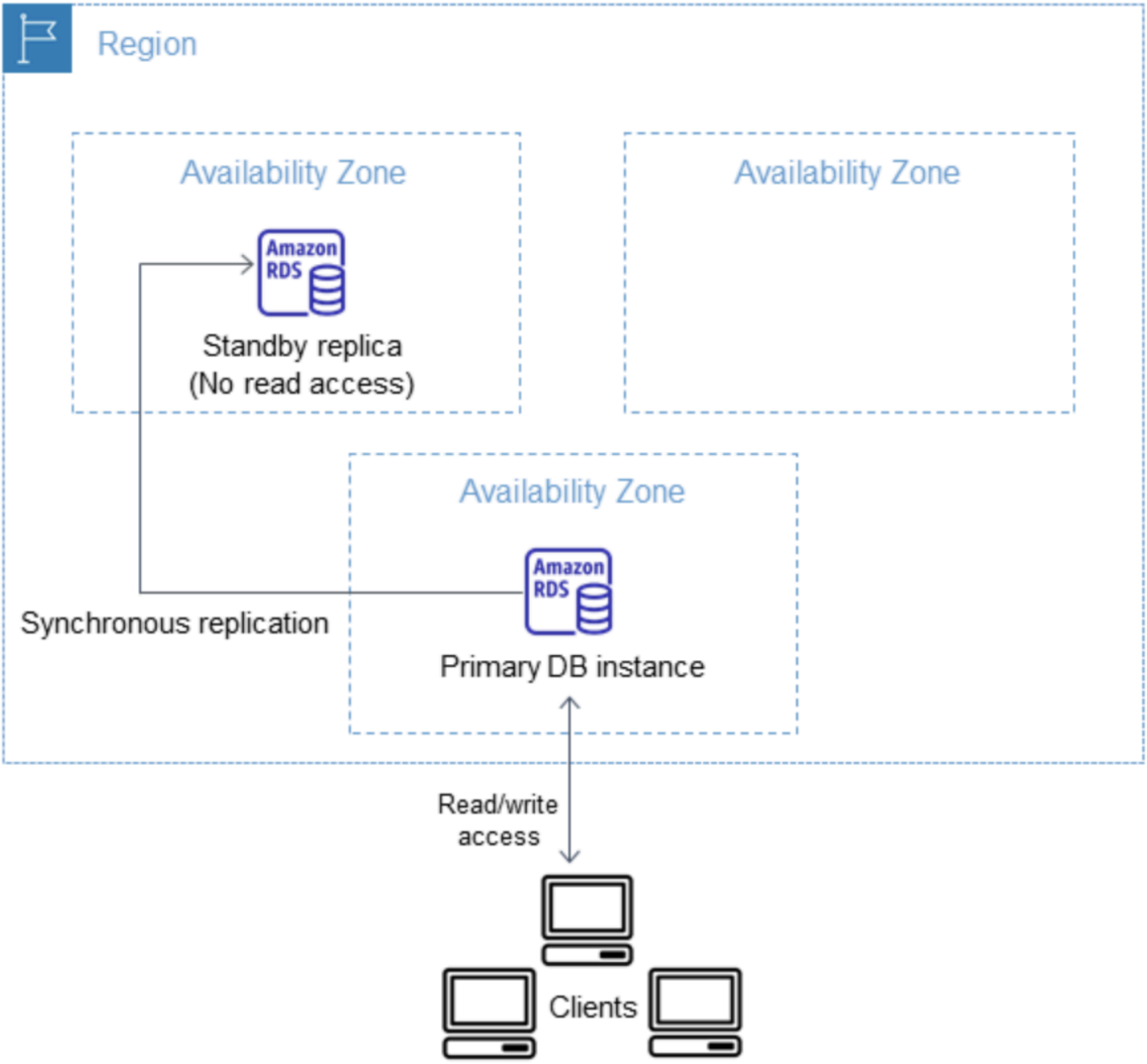
Untuk detail selengkapnya tentang menghentikan dan memulai instans DB, lihat Menghentikan sementara instans DB Amazon RDS dan Memulai instans DB Amazon RDS yang sebelumnya dihentikan.

Mengelola deployment Multi-AZ untuk RDS Custom for SQL Server

Dalam deployment instans DB Multi-AZ untuk RDS Custom for SQL Server, Amazon RDS secara otomatis menyediakan dan memelihara replika siaga sinkron di Zona Ketersediaan (AZ) yang berbeda. Instans DB primer direplikasi secara sinkron ke seluruh Zona Ketersediaan ke replika siaga untuk memberikan redundansi data.

> **Important**
>
> Deployment Multi-AZ untuk RDS Custom for SQL Server berbeda dari Multi-AZ untuk RDS for SQL Server. Tidak seperti Multi-AZ untuk RDS untuk SQL Server, Anda harus menyiapkan prasyarat untuk RDS Custom for SQL Server sebelum membuat instans DB Multi-AZ Anda karena RDS Custom berjalan di dalam akun Anda sendiri, yang memerlukan izin.
>
> Jika Anda tidak menyelesaikan prasyarat, instans DB Multi-AZ Anda mungkin gagal dijalankan, atau secara otomatis kembali ke instans DB AZ Tunggal. Untuk informasi selengkapnya tentang prasyarat, lihat Prasyarat untuk deployment Multi-AZ dengan RDS Custom for SQL Server.

Menjalankan instans DB dengan ketersediaan tinggi dapat meningkatkan ketersediaan selama pemeliharaan sistem terencana. Jika terjadi pemeliharaan basis data yang direncanakan atau gangguan layanan yang tidak direncanakan, Amazon RDS secara otomatis gagal ke instans DB up-to-date sekunder. Fungsi ini memungkinkan operasi basis data berlanjut dengan cepat tanpa gangguan manual. Instans primer dan siaga menggunakan titik akhir yang sama, yang alamat jaringan fisiknya beralih ke replika sekunder sebagai bagian dari proses failover. Anda tidak perlu mengonfigurasi ulang aplikasi Anda saat terjadi failover.



Anda dapat membuat deployment Multi-AZ RDS Custom for SQL Server dengan menentukan Multi-AZ saat membuat instans DB RDS Custom. Anda dapat menggunakan konsol untuk mengonversi instans DB RDS Custom for SQL Server ke deployment Multi-AZ dengan memodifikasi instans DB dan menentukan opsi Multi-AZ. Anda juga dapat menentukan deployment instans DB Multi-AZ dengan CLI AWS atau API Amazon RDS.

Konsol RDS menunjukkan Zona Ketersediaan dari replika siaga (AZ sekunder). Anda juga dapat menggunakan perintah `describe-db-instances` CLI atau operasi `DescribeDBInstances` API untuk menemukan AZ sekunder.

Instans DB RDS Custom for SQL Server dengan deployment Multi-AZ dapat meningkatkan latensi penulisan dan commit dibandingkan dengan deployment AZ Tunggal. Peningkatan ini dapat terjadi karena replikasi data sinkron di antara instans DB. Anda mungkin mengalami perubahan latensi jika deployment ke replika siaga gagal, meskipun AWS direkayasa dengan konektivitas jaringan latensi rendah di antara Zona Ketersediaan.

> **Note**
>
> Untuk beban kerja produksi, kami menyarankan agar Anda menggunakan kelas instans DB dengan IOPS yang Tersedia (operasi input/output per detik) untuk performa yang cepat dan konsisten. Untuk informasi selengkapnya tentang kelas instans DB, lihat Persyaratan dan batasan untuk Amazon RDS Custom for SQL Server.

Topik

Ketersediaan wilayah dan versi

Deployment Multi-AZ untuk RDS Custom for SQL Server didukung untuk edisi SQL Server berikut:

- SQL Server 2022 dan 2019: Edisi Perusahaan, Standar, Web, dan Pengembang

> **Note**
>
> Deployment Multi-AZ untuk RDS Custom for SQL Server tidak didukung di SQL Server 2019 CU8 (15.00.4073.23) atau versi yang lebih rendah.

Deployment Multi-AZ untuk RDS Custom for SQL Server tersedia di semua Wilayah tempat RDS Custom for SQL Server tersedia. Untuk informasi selengkapnya tentang ketersediaan Wilayah untuk deployment Multi-AZ untuk RDS Custom for SQL Server, lihat RDS Custom for SQL Server.

Batasan untuk deployment Multi-AZ dengan RDS Custom for SQL Server

Deployment Multi-AZ dengan RDS Custom for SQL Server memiliki batasan berikut:

- Deployment Multi-AZ lintas Wilayah tidak didukung.
- Anda tidak dapat mengonfigurasi instans DB sekunder untuk menerima aktivitas baca basis data.
- Saat Anda menggunakan Versi Mesin Kustom (CEV) dengan deployment Multi-AZ, instans DB sekunder Anda juga akan menggunakan CEV yang sama. Instans DB sekunder tidak dapat menggunakan CEV yang berbeda.

Prasyarat untuk deployment Multi-AZ dengan RDS Custom for SQL Server

Jika Anda memiliki deployment RDS Custom for SQL Server AZ Tunggal, prasyarat tambahan berikut diperlukan sebelum memodifikasinya ke deployment Multi-AZ. Anda dapat memilih untuk menyelesaikan prasyarat secara manual atau dengan templat yang disediakan. CloudFormation CloudFormation Template terbaru berisi prasyarat untuk penerapan Single-AZ dan Multi-AZ.

> **Important**
>
> Untuk menyederhanakan penyiapan, kami sarankan Anda menggunakan file templat AWS CloudFormation terbaru yang disediakan dalam petunjuk penyiapan jaringan untuk membuat prasyarat. Untuk informasi selengkapnya, lihat Mengonfigurasi dengan AWS CloudFormation.

> **Note**
>
> Saat Anda memodifikasi deployment AZ Tunggal untuk RDS Custom for SQL Server yang ada ke deployment Multi-AZ, Anda harus menyelesaikan prasyarat ini. Jika Anda tidak

menyelesaikan prasyarat, penyiapan Multi-AZ akan gagal. Untuk menyelesaikan prasyarat, ikuti langkah-langkah dalam Mengubah deployment AZ Tunggal RDS for SQL Server ke deployment Multi-AZ.

- Perbarui aturan masuk dan keluar grup keamanan RDS untuk mengizinkan port 1120.
- Tambahkan sebuah aturan dalam Daftar Kontrol Akses (ACL) jaringan privat Anda yang mengizinkan port TCP `0-65535` untuk VPC instans DB.
- Buat titik akhir VPC Amazon SQS baru yang mengizinkan instans DB RDS Custom for SQL Server berkomunikasi dengan SQS.
- Perbarui izin SQS dalam peran profil instans.

Membuat deployment Multi-AZ RDS Custom for SQL Server

Untuk membuat deployment Multi-AZ RDS Custom for SQL Server, ikuti langkah-langkah dalam Membuat dan menghubungkan ke instans DB untuk Amazon RDS Custom for SQL Server.

Important

Untuk menyederhanakan penyiapan, kami sarankan Anda menggunakan file templat AWS CloudFormation terbaru yang disediakan dalam petunjuk penyiapan jaringan. Untuk informasi selengkapnya, lihat Mengonfigurasi dengan AWS CloudFormation.

Membuat deployment Multi-AZ membutuhkan waktu beberapa menit.

Mengubah deployment AZ Tunggal RDS for SQL Server ke deployment Multi-AZ

Anda dapat memodifikasi instans DB RDS Custom for SQL Server yang ada dari deployment AZ Tunggal ke deployment Multi-AZ. Jika Anda mengubah instans DB, Amazon RDS akan melakukan beberapa tindakan:

- Mengambil snapshot instans DB primer.
- Membuat volume baru untuk replika siaga dari snapshot. Volume tersebut diinisialisasi di latar belakang, dan performa volume maksimum akan tercapai setelah data sepenuhnya diinisialisasi.
- Mengaktifkan replikasi tingkat blok yang sinkron antara instans DB primer dan sekunder.

> **Important**
>
> Sebaiknya hindari memodifikasi instans DB RDS Custom for SQL Server dari deployment AZ Tunggal ke Multi-AZ pada instans DB produksi selama periode aktivitas puncak.

AWS menggunakan snapshot untuk membuat instans siaga untuk menghindari waktu henti saat Anda mengonversi dari AZ Tunggal ke Multi-AZ, tetapi performa mungkin akan terpengaruh selama dan setelah mengonversi ke Multi-AZ. Hal ini dapat memberikan dampak yang signifikan terhadap beban kerja yang sensitif terhadap latensi tulis. Meskipun kemampuan ini memungkinkan volume besar dipulihkan dengan cepat dari snapshot, hal ini dapat menyebabkan peningkatan latensi operasi I/O karena replikasi sinkron. Latensi ini dapat memengaruhi performa basis data Anda.

Topik

- Mengkonfigurasi prasyarat untuk memodifikasi single-AZ ke penerapan Multi-AZ menggunakan CloudFormation
- Mengonfigurasi prasyarat untuk memodifikasi Deployment AZ Tunggal ke Multi-AZ
- Modifikasi menggunakan konsol RDS, AWS CLI, atau API RDS.

Mengkonfigurasi prasyarat untuk memodifikasi single-AZ ke penerapan Multi-AZ menggunakan CloudFormation

Untuk menggunakan penerapan Multi-AZ, Anda harus memastikan bahwa Anda telah menerapkan CloudFormation template terbaru dengan prasyarat, atau mengonfigurasi prasyarat terbaru secara manual. Jika Anda telah menerapkan template CloudFormation prasyarat terbaru, Anda dapat melewati langkah-langkah ini.

Untuk mengonfigurasi prasyarat penerapan RDS Custom for SQL Server Multi-AZ menggunakan CloudFormation

1. Buka CloudFormation konsol di https://console.aws.amazon.com/cloudformation.
2. Untuk memulai wizard Buat Tumpukan, pilih tumpukan yang ada yang Anda gunakan untuk membuat deployment AZ Tunggal lalu pilih Perbarui.

   Halaman Perbarui tumpukan muncul.
3. Untuk Prasyarat - Siapkan templat, pilih Ganti templat saat ini.
4. Untuk Tentukan templat, lakukan hal berikut:

a. Unduh file templat AWS CloudFormation terbaru. Buka menu konteks (klik kanan) untuk custom-sqlserver-onboardtautan.zip dan pilih Simpan Tautan Sebagai.

b. Simpan dan ekstrak file `custom-sqlserver-onboard.json` ke komputer Anda.

c. Untuk Sumber templat, pilih Unggah file templat.

d. Untuk Pilih file, navigasikan ke dan pilih `custom-sqlserver-onboard.json`.

5. Pilih Berikutnya.

   Halaman Tentukan detail tumpukan muncul.

6. Untuk mempertahankan opsi default, pilih Berikutnya.

   Halaman Opsi Lanjutan muncul.

7. Untuk mempertahankan opsi default, pilih Berikutnya.

8. Untuk mempertahankan opsi default, pilih Berikutnya.

9. Pada halaman Tinjauan Perubahan, lakukan hal berikut:

   a. Untuk Kemampuan, pilih kotak centang Saya memahami bahwa AWS CloudFormation dapat membuat sumber daya IAM dengan nama kustom.

   b. Pilih Kirim.

10. Periksa apakah pembaruan berhasil. Status operasi yang sukses menunjukkan `UPDATE_COMPLETE`.

Jika pembaruan gagal, konfigurasi baru apa pun yang ditentukan dalam proses pembaruan akan dibatalkan. Sumber daya yang ada masih akan dapat digunakan. Misalnya, jika Anda menambahkan aturan ACL jaringan bernomor 18 dan 19, tetapi ada aturan yang ada dengan nomor yang sama, pembaruan akan menampilkan kesalahan berikut: `Resource handler returned message: "The network acl entry identified by 18 already exists.` Dalam skenario ini, Anda dapat mengubah aturan ACL yang ada untuk menggunakan nomor yang lebih rendah dari 18, lalu coba lagi pembaruan tersebut.

Mengonfigurasi prasyarat untuk memodifikasi Deployment AZ Tunggal ke Multi-AZ

> **Important**
>
> Untuk menyederhanakan penyiapan, kami sarankan Anda menggunakan file templat AWS CloudFormation terbaru yang disediakan dalam petunjuk penyiapan jaringan. Untuk informasi

selengkapnya, lihat Mengkonfigurasi prasyarat untuk memodifikasi single-AZ ke penerapan Multi-AZ menggunakan CloudFormation.

Jika Anda memilih untuk mengonfigurasi prasyarat secara manual, lakukan tugas berikut.

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Pilih Titik akhir. Halaman Buat titik akhir muncul.
3. Untuk Kategori Layanan, pilih Layanan AWS.
4. Di Layanan, cari *SQS*
5. Di VPC, pilih VPC tempat instans DB RDS Custom for SQL Server digunakan.
6. Di Subnet, pilih subnet tempat instans DB RDS Custom for SQL Server digunakan.
7. Di Grup Keamanan, pilih vpc-endpoint-sg grup -.
8. Untuk Kebijakan, pilih Kustom
9. Dalam kebijakan kustom Anda, ganti *Partisi AWS*, *Wilayah*, *accountId*, dan *IAM-Instance-role* dengan nilai Anda sendiri.

```
                {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Condition": {
                    "StringLike": {
                        "aws:ResourceTag/AWSRDSCustom": "custom-sqlserver"
                    }
                },
                "Action": [
                    "SQS:SendMessage",
                    "SQS:ReceiveMessage",
                    "SQS:DeleteMessage",
                    "SQS:GetQueueUrl"
                ],
                "Resource": "arn:${AWS::Partition}:sqs:${AWS::Region}:
${AWS::AccountId}:do-not-delete-rds-custom-*",
                "Effect": "Allow",
                "Principal": {
                    "AWS": "arn:${AWS::Partition}:iam::${AWS::AccountId}:role/{IAM-
Instance-role}"
```

```
                }
            }
        ]
    }
```

10. Perbarui Profil instans dengan izin untuk mengakses Amazon SQS. Ganti *Partisi AWS*, *Wilayah*, dan *accountId* dengan nilai Anda sendiri.

```
                        {
    "Sid": "SendMessageToSQSQueue",
    "Effect": "Allow",
    "Action": [
      "SQS:SendMessage",
      "SQS:ReceiveMessage",
      "SQS:DeleteMessage",
      "SQS:GetQueueUrl"

    ],
    "Resource": [
      {
        "Fn::Sub": "arn:${AWS::Partition}:sqs:${AWS::Region}:${AWS::AccountId}:do-
not-delete-rds-custom-*"
      }
    ],
    "Condition": {
      "StringLike": {
        "aws:ResourceTag/AWSRDSCustom": "custom-sqlserver"
      }
    }
  }
                        >
```

11. Perbarui aturan masuk dan keluar grup keamanan Amazon RDS untuk mengizinkan port 1120.

    a. Di Grup Keamanan, pilih rds-custom-instance-sg grup -.

    b. *Untuk **Aturan Masuk, buat aturan TCP Kustom** untuk mengizinkan port 1120 dari grup sumber. rds-custom-instance-sg*

    c. *Untuk **Aturan Keluar**, buat aturan **TCP Kustom** untuk mengizinkan port 1120 ke grup tujuan. rds-custom-instance-sg*

12. Tambahkan sebuah aturan dalam Daftar Kontrol Akses (ACL) jaringan privat Anda yang mengizinkan port TCP `0-65535` untuk subnet sumber instans DB.

    > **Note**
    >
    > Saat membuat Aturan Masuk dan Aturan Keluar, catat Nomor aturan tertinggi yang ada. Aturan baru yang Anda buat harus memiliki Nomor aturan yang lebih rendah dari 100 dan tidak sama dengan Nomor aturan yang ada.

    a. Di ACL Jaringan, pilih private-network-acl grup `-`.
    b. Untuk Aturan Masuk, buat aturan Semua TCP untuk mengizinkan port TCP `0-65535` dengan sumber *`privatesubnet1`* dan *`privatesubnet2`*.
    c. Untuk Aturan Keluar, buat aturan Semua TCP untuk mengizinkan port TCP `0-65535` ke *`privatesubnet1`* dan *`privatesubnet2`* tujuan.

Modifikasi menggunakan konsol RDS, AWS CLI, atau API RDS.

Setelah menyelesaikan prasyarat, Anda dapat memodifikasi instans DB RDS Custom for SQL Server dari deployment AZ Tunggal ke Multi-AZ menggunakan konsol RDS, AWS CLI, atau API RDS.

Konsol

Untuk memodifikasi deployment AZ Tunggal RDS Custom for SQL Server yang ada ke Multi-AZ

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di konsol Amazon RDS, pilih Database.

   Panel Basis Data muncul.
3. Pilih instans DB RDS Custom for SQL Server yang ingin Anda modifikasi.
4. Untuk Tindakan, pilih Konversikan ke deployment Multi-AZ.
5. Pada halaman Konfirmasi, pilih Terapkan segera untuk segera menerapkan perubahan. Memilih opsi ini tidak akan menyebabkan waktu henti, tetapi ada kemungkinan dampak performa. Atau, Anda dapat memilih untuk menerapkan pembaruan pada periode pemeliharaan berikutnya. Untuk informasi selengkapnya, lihat Menggunakan pengaturan Terapkan Segera.
6. Pada halaman Konfirmasi, pilih Konversi ke Multi-AZ.

AWS CLI

Untuk mengonversi ke penyebaran instans DB multi-AZ dengan menggunakanAWS CLI, panggil modify-db-instanceperintah dan atur opsi. `--multi-az` Tentukan pengidentifikasi instans DB dan nilai untuk opsi lain yang ingin Anda modifikasi. Untuk informasi tentang setiap opsi, lihat Pengaturan untuk instans DB.

Example

Kode berikut memodifikasi `mycustomdbinstance` dengan menyertakan opsi `--multi-az`. Perubahan diterapkan selama jendela pemeliharaan berikutnya dengan menggunakan `--no-apply-immediately`. Gunakan `--apply-immediately` untuk menerapkan perubahan dengan serta-merta. Untuk informasi selengkapnya, lihat Menggunakan pengaturan Terapkan Segera.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mycustomdbinstance \
    --multi-az \
    --no-apply-immediately
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mycustomdbinstance ^
    --multi-az  \ ^
    --no-apply-immediately
```

API RDS

Untuk mengonversi ke deployment instans DB Multi-AZ dengan API RDS, panggil operasi ModifyDBInstance dan atur parameter `MultiAZ` ke true.

Memodifikasi deployment Multi-AZ RDS Custom for SQL Server ke deployment AZ Tunggal

Anda dapat memodifikasi instans DB RDS Custom for SQL Server yang ada dari deployment Multi-AZ ke AZ Tunggal.

Konsol

Untuk memodifikasi instans DB RDS Custom for SQL Server dari deployment Multi-AZ ke AZ Tunggal.

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di konsol Amazon RDS, pilih Database.

   Panel Basis Data muncul.
3. Pilih instans DB RDS Custom for SQL Server yang ingin Anda modifikasi.
4. Untuk Deployment Multi-AZ, pilih Tidak.
5. Pada halaman Konfirmasi, pilih Terapkan segera untuk segera menerapkan perubahan. Memilih opsi ini tidak akan menyebabkan waktu henti, tetapi ada kemungkinan dampak performa. Atau, Anda dapat memilih untuk menerapkan pembaruan pada periode pemeliharaan berikutnya. Untuk informasi selengkapnya, lihat Menggunakan pengaturan Terapkan Segera.
6. Di halaman Konfirmasi, pilih Modifikasi Instans DB.

AWS CLI

Untuk memodifikasi penyebaran Multi-AZ ke penyebaran Single-AZ dengan menggunakanAWS CLI, panggil modify-db-instanceperintah dan sertakan opsi. `--no-multi-az` Tentukan pengidentifikasi instans DB dan nilai untuk opsi lain yang ingin Anda modifikasi. Untuk informasi tentang setiap opsi, lihat Pengaturan untuk instans DB.

Example

Kode berikut memodifikasi `mycustomdbinstance` dengan menyertakan opsi `--no-multi-az`. Perubahan diterapkan selama jendela pemeliharaan berikutnya dengan menggunakan `--no-apply-immediately`. Gunakan `--apply-immediately` untuk menerapkan perubahan dengan serta-merta. Untuk informasi selengkapnya, lihat Menggunakan pengaturan Terapkan Segera.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mycustomdbinstance \
    --no-multi-az  \
    --no-apply-immediately
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mycustomdbinstance ^
    --no-multi-az \ ^
    --no-apply-immediately
```

API RDS

Untuk memodifikasi deployment Multi-AZ ke deployment AZ Tunggal dengan menggunakan API RDS, panggil operasi ModifyDBInstance dan atur parameter `MultiAZ` ke `false`.

Proses failover untuk deployment Multi-AZ RDS for SQL Server

Jika penghentian instans DB terencana atau tidak terencana terjadi karena cacat infrastruktur, Amazon RDS akan otomatis beralih ke replika siaga di Zona Ketersediaan lain jika Anda telah mengaktifkan Multi-AZ. Durasi penyelesaian failover bergantung pada aktivitas basis data dan kondisi lain pada saat instans DB primer tidak tersedia. Durasi failover biasanya 60–120 detik. Namun, transaksi besar atau proses pemulihan yang panjang dapat meningkatkan durasi failover. Setelah failover selesai, perlu waktu tambahan agar konsol RDS dapat menampilkan Zona Ketersediaan baru.

> **Note**
>
> Anda dapat memaksa failover secara manual saat mem-boot ulang instans DB dengan failover. Untuk informasi selengkapnya tentang mem-boot ulang instans DB, lihat Mem-boot ulang instans DB

Amazon RDS menangani failover secara otomatis sehingga Anda dapat melanjutkan operasi basis data secepat mungkin tanpa intervensi administratif. Instans DB primer otomatis beralih ke replika siaga jika salah satu dari kondisi yang dijelaskan dalam tabel berikut terjadi. Anda dapat melihat alasan failover ini di log peristiwa RDS.

Alasan failover	Deskripsi
`The operating system for the RDS Custom for SQL Server`	Failover dipicu selama periode pemeliharaan untuk patch OS atau pembaruan keamanan. Untuk informasi selengkapnya, lihat Memelihara instans DB.

Alasan failover	Deskripsi
`Multi-AZ DB instance is being patched in an offline operation`	
`The primary host of the RDS Custom for SQL Server Multi-AZ DB instance is unhealthy.`	Deployment instans DB Multi-AZ mendeteksi instans DB primer yang terganggu dan melakukan failover.
`The primary host of the RDS Custom for SQL Server Multi-AZ DB instance is unreachable due to loss of network connectivity.`	Pemantauan RDS mendeteksi kegagalan keterjangkauan jaringan ke instans DB primer dan telah memicu failover.
`The RDS Custom for SQL Server Multi-AZ DB instance was modified by the customer.`	Modifikasi instans DB memicu failover. Untuk informasi selengkap nya, lihat Memodifikasi instans DB RDS Custom for SQL Server.
`The storage volume of the primary host of the RDS Custom for SQL Server Multi-AZ DB instance experienced a failure.`	Deployment instans DB Multi-AZ mendeteksi masalah penyimpanan pada instans DB primer dan melakukan failover.

Alasan failover	Deskripsi
`The user requested a failover of the RDS Custom for SQL Server Multi-AZ DB instance.`	Instans DB Multi-AZ RDS Custom for SQL Server di-boot ulang dengan failover. Untuk informasi selengkapnya, lihat Mem-boot ulang instans DB.
`The RDS Custom for SQL Server Multi-AZ primary DB instance is busy or unresponsive.`	Instans DB primer tidak responsif. Kami menyarankan Anda mencoba langkah-langkah berikut: • Periksa log peristiwa dan CloudWatch log untuk penggunaan CPU, memori, atau ruang swap yang berlebihan. Untuk informasi selengkapnya, lihat Menggunakan pemberitahuan peristiwa Amazon RDS. • Buat aturan yang dipicu berdasarkan peristiwa Amazon RDS. Untuk informasi selengkapnya, lihat Membuat aturan yang memicu peristiwa Amazon RDS. • Evaluasi beban kerja Anda untuk menentukan apakah Anda menggunakan kelas instans DB yang sesuai. Untuk informasi selengkapnya, lihat Kelas instans DB .

Untuk mengetahui apakah instans DB Multi-AZ mengalami failover, Anda dapat melakukan tindakan berikut:

- Siapkan langganan peristiwa DB untuk memberi tahu Anda melalui email atau SMS bahwa failover telah dimulai. Untuk informasi selengkapnya tentang peristiwa, lihat Menggunakan pemberitahuan peristiwa Amazon RDS.
- Lihat peristiwa DB Anda dengan menggunakan konsol RDS atau operasi API.
- Lihat status deployment instans DB Multi-AZ RDS Custom for SQL Server Anda saat ini dengan menggunakan konsol RDS, CLI, atau operasi API.

Pengaturan time to live (TTL) dengan aplikasi yang menggunakan deployment Multi-AZ RDS Custom for SQL Server

Mekanisme failover secara otomatis mengubah catatan Sistem Nama Domain (DNS) milik instans DB untuk mengarah ke instans DB siaga. Oleh karena itu, Anda perlu membuat kembali koneksi yang ada ke instans DB Anda. Pastikan bahwa setiap nilai konfigurasi cache DNS time-to-live (TTL) rendah, dan validasi bahwa aplikasi Anda tidak akan cache DNS untuk waktu yang lama. Nilai TTL yang tinggi dapat mencegah aplikasi Anda terhubung kembali dengan cepat ke instans DB setelah failover.

Mencadangkan dan memulihkan instans DB Amazon RDS Custom for SQL Server

Seperti Amazon RDS, RDS Custom membuat dan menyimpan cadangan otomatis instans DB RDS Custom for SQL Server Anda saat retensi cadangan diaktifkan. Anda juga dapat mencadangkan instans DB secara manual. Cadangan otomatis terdiri dari cadangan snapshot dan cadangan log transaksi. Cadangan snapshot diambil untuk seluruh volume penyimpanan instans DB selama jendela cadangan yang Anda tentukan. Cadangan log transaksi diambil untuk basis data yang memenuhi syarat PITR pada periode interval reguler. RDS Custom menyimpan cadangan otomatis instans DB Anda sesuai dengan periode retensi cadangan yang Anda tentukan. Anda dapat menggunakan cadangan otomatis untuk memulihkan instans DB Anda ke titik waktu dalam periode retensi cadangan.

Anda juga dapat mengambil cadangan snapshot secara manual. Anda dapat membuat instans DB baru dari cadangan snapshot ini kapan saja. Untuk informasi selengkapnya tentang cara membuat snapshot DB secara manual, lihat Membuat snapshot RDS Custom for SQL Server.

Meskipun cadangan snapshot berfungsi secara operasional sebagai cadangan penuh, Anda hanya ditagih untuk penggunaan penyimpanan tambahan. Snapshot pertama instans DB RDS Custom berisi data untuk instans DB penuh. Snapshot berikutnya dari basis data yang sama bersifat inkremental, artinya hanya data yang berubah setelah snapshot terbaru Anda yang disimpan.

Topik

- Membuat snapshot RDS Custom for SQL Server
- Memulihkan dari snapshot DB RDS Custom for SQL Server
- Memulihkan instans RDS Custom for SQL Server ke suatu titik waktu
- Menghapus snapshot RDS Custom for SQL Server
- Menghapus cadangan otomatis RDS Custom for SQL Server

Membuat snapshot RDS Custom for SQL Server

RDS Custom for SQL Server membuat snapshot volume penyimpanan instans DB Anda, mencadangkan seluruh instans DB dan bukan hanya basis data individual. Saat Anda membuat snapshot, tentukan instans DB RDS Custom for SQL Server mana yang akan dicadangkan. Beri nama snapshot sehingga Anda dapat melakukan proses pemulihan dari snapshot tersebut nanti.

Saat Anda membuat snapshot, RDS Custom for SQL Server membuat snapshot Amazon EBS untuk volume `(D:)`, yang merupakan volume basis data yang dilampirkan ke instans DB. Agar mudah dikaitkan dengan instans DB tertentu, snapshot ditandai dengan `DBSnapshotIdentifier`, `DbiResourceId`, dan `VolumeType`.

Membuat snapshot DB menghasilkan suspensi I/O singkat. Suspensi ini dapat bertahan beberapa detik hingga beberapa menit, bergantung pada ukuran dan kelas instans DB Anda. Waktu pembuatan snapshot bervariasi menurut jumlah total dan ukuran basis data Anda. Untuk mempelajari selengkapnya tentang jumlah basis data yang memenuhi syarat untuk operasi pemulihan titik waktu (PITR), lihat Jumlah basis data yang memenuhi syarat untuk PITR per jenis kelas instans.

Karena snapshot mencakup seluruh volume penyimpanan, ukuran file seperti file sementara juga memengaruhi waktu pembuatan snapshot. Untuk mempelajari selengkapnya tentang membuat snapshot, lihat Membuat snapshot DB untuk instans DB Single-AZ.

Buat snapshot RDS Custom for SQL Server menggunakan konsol atau AWS CLI.

Konsol

Cara membuat snapshot RDS Custom

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data.
3. Dalam daftar instans DB RDS Custom, pilih instans yang ingin Anda ambil snapshot-nya.
4. Untuk Tindakan, pilih Ambil snapshot.

   Jendela Ambil snapshot DB akan muncul.
5. Untuk Nama snapshot, masukkan nama snapshot.
6. Pilih Ambil snapshot.

AWS CLI

Anda membuat snapshot dari instans RDS Custom DB dengan menggunakan perintah. create-db-snapshotAWS CLI

Tentukan opsi berikut:

- `--db-instance-identifier` — Mengidentifikasi instans DB RDS Custom mana yang akan Anda cadangkan
- `--db-snapshot-identifier` — Memberi nama snapshot RDS Custom sehingga Anda dapat melakukan proses pemulihan dari snapshot tersebut nanti

Dalam contoh ini, Anda membuat snapshot DB bernama *my-custom-snapshot* untuk instans DB RDS Custom bernama *my-custom-instance*.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-snapshot \
    --db-instance-identifier my-custom-instance \
    --db-snapshot-identifier my-custom-snapshot
```

Untuk Windows:

```
aws rds create-db-snapshot ^
    --db-instance-identifier my-custom-instance ^
    --db-snapshot-identifier my-custom-snapshot
```

Memulihkan dari snapshot DB RDS Custom for SQL Server

Saat memulihkan instans DB RDS Custom for SQL Server, Anda memberikan nama untuk snapshot DB dan instans baru. Anda tidak dapat memulihkan dari snapshot ke instans DB RDS Custom yang ada. Instans DB RDS Custom for SQL Server baru dibuat saat Anda melakukan pemulihan.

Memulihkan dari snapshot akan mengembalikan volume penyimpanan ke titik waktu ketika snapshot diambil. Hal ini akan mencakup semua basis data dan file lain yang ada pada volume (`D:`).

Konsol

Cara memulihkan instans DB RDS Custom dari snapshot DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Snapshot.
3. Pilih snapshot DB yang ingin Anda pulihkan.

4. Untuk Tindakan, pilih Pulihkan snapshot.
5. Di halaman Pulihkan instans DB, untuk Pengidentifikasi instans DB, masukkan nama instans DB RDS Custom Anda yang dipulihkan.
6. Pilih Pulihkan instans DB.

AWS CLI

Anda mengembalikan snapshot RDS Custom DB dengan menggunakan perintah restore-db-instance-fromAWS CLI-db-snapshot.

Jika snapshot yang Anda pulihkan adalah untuk instans DB privat, pastikan untuk menentukan `db-subnet-group-name` dan `no-publicly-accessible` yang benar. Jika tidak, default instans DB diatur agar dapat diakses publik. Opsi berikut diperlukan:

- `db-snapshot-identifier` — Mengidentifikasi snapshot yang akan dipulihkan
- `db-instance-identifier` — Menentukan nama instans DB RDS Custom yang akan dibuat dari snapshot DB
- `custom-iam-instance-profile` — Menentukan profil instans yang terkait dengan instans Amazon EC2 yang mendasari instans DB RDS Custom.

Kode berikut memulihkan snapshot bernama `my-custom-snapshot` untuk `my-custom-instance`.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds restore-db-instance-from-db-snapshot \
  --db-snapshot-identifier my-custom-snapshot \
  --db-instance-identifier my-custom-instance \
  --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance \
  --no-publicly-accessible
```

Untuk Windows:

```
aws rds restore-db-instance-from-db-snapshot ^
  --db-snapshot-identifier my-custom-snapshot ^
  --db-instance-identifier my-custom-instance ^
  --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance ^
  --no-publicly-accessible
```

Memulihkan instans RDS Custom for SQL Server ke suatu titik waktu

Anda dapat memulihkan instans DB ke titik waktu tertentu (PITR) dan membuat instans DB baru. Untuk mendukung PITR, instans DB Anda harus mengaktifkan retensi cadangan.

Waktu pemulihan terbaru untuk instans DB RDS Custom for SQL Server bergantung pada beberapa faktor, tetapi biasanya dalam 5 menit dari waktu saat ini. Untuk melihat waktu restorable terbaru untuk instans DB, gunakan AWS CLI describe-db-instancesperintah dan lihat nilai yang dikembalikan di `LatestRestorableTime` bidang untuk instans DB. Untuk melihat waktu pemulihan terbaru setiap instans DB di konsol Amazon RDS, pilih Cadangan otomatis.

Anda dapat memulihkan ke titik waktu mana pun dalam periode retensi cadangan Anda. Untuk melihat waktu pemulihan terbaru setiap instans DB, pilih Cadangan otomatis di konsol Amazon RDS.

Untuk informasi umum tentang PITR, lihat Memulihkan instans DB dengan waktu yang ditentukan.

Topik

- Pertimbangan PITR untuk RDS Custom for SQL Server
- Jumlah basis data yang memenuhi syarat untuk PITR per jenis kelas instans
- Membuat basis data tidak memenuhi syarat untuk PITR
- Log transaksi di Amazon S3
- Pemulihan PITR menggunakan AWS Management Console, AWS CLI, atau API RDS.

Pertimbangan PITR untuk RDS Custom for SQL Server

PITR di RDS Custom for SQL Server berbeda dari PITR di Amazon RDS dalam beberapa hal penting berikut:

- PITR hanya memulihkan basis data dalam instans DB. PITR tidak memulihkan sistem operasi atau file pada drive C:.
- Untuk instans DB RDS Custom for SQL Server, basis data dicadangkan secara otomatis dan memenuhi syarat untuk PITR hanya dalam kondisi berikut:
  - Basis data online.
  - Model pemulihannya diatur ke `FULL`.
  - Dapat ditulis.
  - Memiliki file fisik di drive D:.

  - Tidak tercantum dalam tabel `rds_pitr_blocked_databases`. Untuk informasi selengkapnya, lihat Membuat basis data tidak memenuhi syarat untuk PITR.
- Basis data yang memenuhi syarat untuk PITR ditentukan oleh urutan ID basis data. RDS Custom for SQL Server memungkinkan hingga 5.000 basis data per instans DB. Namun, jumlah maksimum basis data yang dipulihkan oleh operasi PITR untuk instans DB RDS Custom for SQL Server bergantung pada jenis kelas instans. Untuk informasi selengkapnya, lihat Jumlah basis data yang memenuhi syarat untuk PITR per jenis kelas instans.

  Basis data lain yang bukan bagian dari PITR dapat dipulihkan dari snapshot DB, termasuk cadangan snapshot otomatis yang digunakan untuk PITR.
- Menambahkan basis data baru, mengganti nama basis data, atau memulihkan basis data yang memenuhi syarat untuk PITR memulai snapshot instans DB.
- Jumlah maksimum basis data yang memenuhi syarat untuk PITR berubah ketika instans basis data melewati operasi komputasi skala, bergantung pada jenis kelas instans target. Jika skala instans dinaikkan dan memungkinkan lebih banyak basis data pada instans memenuhi syarat untuk PITR, snapshot baru akan diambil.
- Basis data yang dipulihkan memiliki nama yang sama seperti pada instans DB sumber. Anda tidak dapat menentukan nama yang berbeda.
- `AWSRDSCustomSQLServerIamRolePolicy` membutuhkan akses ke layanan AWS lain. Untuk informasi selengkapnya, lihat Menambahkan kebijakan akses ke AWSRDSCustomSQLServerInstanceRole.
- Perubahan zona waktu tidak didukung untuk RDS Custom for SQL Server. Jika Anda mengubah zona waktu sistem operasi atau instans DB, PITR (dan otomatisasi lainnya) tidak berfungsi.

Jumlah basis data yang memenuhi syarat untuk PITR per jenis kelas instans

Tabel berikut menunjukkan jumlah maksimum basis data yang memenuhi syarat untuk PITR berdasarkan jenis kelas instans.

Jenis kelas instans	Jumlah maksimum basis data yang memenuhi syarat untuk PITR
db.*.large	100
db.*.xlarge hingga db.*.2xlarge	150
db.*.4xlarge hingga db.*.8xlarge	300
db.*.12xlarge hingga db.*.16xlarge	600
db.*.24xlarge, db.*32xlarge	1000

* Menunjukkan jenis kelas instans yang berbeda.

Jumlah maksimum basis data yang memenuhi syarat untuk PITR pada instans DB bergantung pada jenis kelas instans. Jumlahnya berkisar dari 100 pada jenis kelas instans terkecil hingga 1000 pada jenis kelas instans terbesar yang didukung oleh RDS Custom for SQL Server. Basis data sistem SQL server (`master, model, msdb, tempdb`), tidak termasuk dalam batas ini. Ketika skala instans DB dinaikkan atau diturunkan, bergantung pada jenis kelas instans target, RDS Custom akan otomatis memperbarui jumlah basis data yang memenuhi syarat untuk PITR. RDS Custom for SQL Server akan mengirim `RDS-EVENT-0352` ketika jumlah maksimum basis data yang memenuhi syarat untuk PITR berubah pada instans DB. Untuk informasi selengkapnya, lihat Peristiwa versi mesin kustom.

> **Note**
>
> Dukungan PITR untuk lebih dari 100 basis data hanya tersedia pada instans DB yang dibuat setelah 26 Agustus 2023. Untuk instans yang dibuat sebelum 26 Agustus 2023, jumlah maksimum basis data yang memenuhi syarat untuk PITR adalah 100, terlepas dari kelas instansnya. Guna mengaktifkan dukungan PITR untuk lebih dari 100 basis data pada instans DB yang dibuat sebelum 26 Agustus 2023, Anda dapat melakukan tindakan berikut:
>
> - Tingkatkan versi mesin DB ke 15.00.4322.2.v1 atau lebih tinggi

Selama operasi PITR, RDS Custom akan memulihkan semua basis data yang merupakan bagian dari PITR pada instans DB sumber pada waktu pemulihan. Setelah instans DB target menyelesaikan operasi pemulihan, jika retensi cadangan diaktifkan, instans DB akan mulai mencadangkan berdasarkan jumlah maksimum basis data yang memenuhi syarat untuk PITR pada instans DB target.

Misalnya, jika instans DB Anda berjalan pada `db.*.xlarge` yang memiliki 200 basis data:

1. RDS Custom for SQL Server akan memilih 150 basis data pertama yang diurutkan berdasarkan ID basis data untuk cadangan PITR.
2. Anda memodifikasi instans untuk menaikkan skala hingga db.*.4xlarge.
3. Setelah operasi komputasi skala selesai, RDS Custom for SQL Server akan memilih 300 basis data pertama, diurutkan berdasarkan ID basis data, untuk cadangan PITR. Masing-masing dari 200 basis data yang memenuhi kondisi persyaratan PITR sekarang akan memenuhi syarat untuk PITR.
4. Sekarang Anda memodifikasi instans untuk menurunkan skala kembali ke db.*.xlarge.
5. Setelah operasi komputasi skala selesai, RDS Custom for SQL Server akan kembali memilih 150 basis data pertama, diurutkan berdasarkan ID basis data, untuk cadangan PITR.

Membuat basis data tidak memenuhi syarat untuk PITR

Anda dapat memilih untuk mengecualikan basis data individual dari PITR. Untuk melakukan ini, masukkan nilai `database_id` ke dalam tabel `rds_pitr_blocked_databases`. Gunakan skrip SQL berikut untuk membuat tabel.

Cara membuat tabel rds_pitr_blocked_databases

- Jalankan skrip SQL berikut.

```
create table msdb..rds_pitr_blocked_databases
(
database_id INT NOT NULL,
database_name SYSNAME NOT NULL,
db_entry_updated_date datetime NOT NULL DEFAULT GETDATE(),
db_entry_updated_by SYSNAME NOT NULL DEFAULT CURRENT_USER,
PRIMARY KEY (database_id)
);
```

Untuk daftar basis data yang memenuhi syarat dan tidak memenuhi syarat, lihat file `RI.End` pada direktori `RDSCustomForSQLServer/Instances/`*`DB_instance_resource_ID`*`/TransactionLogMetadata` di bucket Amazon S3 `do-not-delete-rds-custom-`*`$ACCOUNT_ID`*`-`*`$REGION`*`-`*`unique_identifier`*. Untuk informasi selengkapnya tentang file `RI.End`, lihat Log transaksi di Amazon S3.

Anda juga dapat menentukan daftar basis data yang memenuhi syarat untuk PITR menggunakan skrip SQL berikut. Tetapkan variabel `@limit` ke jumlah maksimum basis data yang memenuhi syarat untuk PITR untuk kelas instans. Untuk informasi selengkapnya, lihat Jumlah basis data yang memenuhi syarat untuk PITR per jenis kelas instans.

Cara menentukan daftar basis data yang memenuhi syarat untuk PITR pada kelas instans DB

- Jalankan skrip SQL berikut.

```
DECLARE @Limit INT;
SET @Limit = (insert-database-instance-limit-here);

USE msdb;
IF (EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_SCHEMA = 'dbo' AND
 TABLE_NAME = 'rds_pitr_blocked_databases'))
    WITH TABLE0 AS (
        SELECT hdrs.database_id as DatabaseId, sdb.name as DatabaseName,
 'ALWAYS_ON_NOT_WRITABLE_REPLICA' as Reason, NULL as DatabaseNameOnPitrTable
        FROM sys.dm_hadr_database_replica_states hdrs
        INNER JOIN sys.databases sdb ON sdb.database_id = hdrs.database_id
        WHERE (hdrs.is_local = 1 AND hdrs.is_primary_replica = 0)
```

```
       OR (sys.fn_hadr_is_primary_replica (sdb.name) = 1 AND DATABASEPROPERTYEX
(sdb.name, 'Updateability') = 'READ_ONLY')
   ),
   TABLE1 as (
           SELECT dbs.database_id as DatabaseId, sysdbs.name as DatabaseName,
'OPTOUT' as Reason,
           CASE WHEN dbs.database_name = sysdbs.name THEN NULL ELSE
dbs.database_name END AS DatabaseNameOnPitrTable
           FROM msdb.dbo.rds_pitr_blocked_databases dbs
           INNER JOIN sys.databases sysdbs ON dbs.database_id = sysdbs.database_id
           WHERE sysdbs.database_id > 4
           ),
   TABLE2 as (
           SELECT
           db.name AS DatabaseName,
           db.create_date AS CreateDate,
           db.state_desc AS DatabaseState,
           db.database_id AS DatabaseId,
           rs.database_guid AS DatabaseGuid,
           rs.last_log_backup_lsn AS LastLogBackupLSN,
           rs.recovery_fork_guid RecoveryForkGuid,
           rs.first_recovery_fork_guid AS FirstRecoveryForkGuid,
           db.recovery_model_desc AS RecoveryModel,
           db.is_auto_close_on AS IsAutoClose,
           db.is_read_only as IsReadOnly,
           NEWID() as FileName,
           CASE WHEN(db.state_desc = 'ONLINE'
                   AND db.recovery_model_desc != 'SIMPLE'
                   AND((db.is_auto_close_on = 0 and db.collation_name IS NOT NULL)
OR db.is_auto_close_on = 1))
                   AND db.is_read_only != 1
                   AND db.user_access = 0
                   AND db.source_database_id IS NULL
                   AND db.is_in_standby != 1
                   THEN 1 ELSE 0 END AS IsPartOfSnapshot,
           CASE WHEN db.source_database_id IS NULL THEN 0 ELSE 1 END AS
IsDatabaseSnapshot
           FROM sys.databases db
           INNER JOIN sys.database_recovery_status rs
           ON db.database_id = rs.database_id
           WHERE DB_NAME(db.database_id) NOT IN('tempdb') AND
           db.database_id NOT IN (SELECT DISTINCT DatabaseId FROM TABLE1) AND
           db.database_id NOT IN (SELECT DISTINCT DatabaseId FROM TABLE0)
       ),
```

```
        TABLE3 as(
            Select @Limit+count(DatabaseName) as TotalNumberOfDatabases from TABLE2
 where TABLE2.IsPartOfSnapshot=1 and DatabaseName in ('master','model','msdb')
        )
        SELECT TOP(SELECT TotalNumberOfDatabases from TABLE3)
 DatabaseName,CreateDate,DatabaseState,DatabaseId from TABLE2 where
 TABLE2.IsPartOfSnapshot=1
        ORDER BY TABLE2.DatabaseID ASC
ELSE
    WITH TABLE0 AS (
        SELECT hdrs.database_id as DatabaseId, sdb.name as DatabaseName,
 'ALWAYS_ON_NOT_WRITABLE_REPLICA' as Reason, NULL as DatabaseNameOnPitrTable
        FROM sys.dm_hadr_database_replica_states hdrs
        INNER JOIN sys.databases sdb ON sdb.database_id = hdrs.database_id
        WHERE (hdrs.is_local = 1 AND hdrs.is_primary_replica = 0)
        OR (sys.fn_hadr_is_primary_replica (sdb.name) = 1 AND DATABASEPROPERTYEX
 (sdb.name, 'Updateability') = 'READ_ONLY')
    ),
    TABLE1 as (
            SELECT
            db.name AS DatabaseName,
            db.create_date AS CreateDate,
            db.state_desc AS DatabaseState,
            db.database_id AS DatabaseId,
            rs.database_guid AS DatabaseGuid,
            rs.last_log_backup_lsn AS LastLogBackupLSN,
            rs.recovery_fork_guid RecoveryForkGuid,
            rs.first_recovery_fork_guid AS FirstRecoveryForkGuid,
            db.recovery_model_desc AS RecoveryModel,
            db.is_auto_close_on AS IsAutoClose,
            db.is_read_only as IsReadOnly,
            NEWID() as FileName,
            CASE WHEN(db.state_desc = 'ONLINE'
                    AND db.recovery_model_desc != 'SIMPLE'
                    AND((db.is_auto_close_on = 0 and db.collation_name IS NOT NULL)
 OR db.is_auto_close_on = 1))
                    AND db.is_read_only != 1
                    AND db.user_access = 0
                    AND db.source_database_id IS NULL
                    AND db.is_in_standby != 1
                    THEN 1 ELSE 0 END AS IsPartOfSnapshot,
            CASE WHEN db.source_database_id IS NULL THEN 0 ELSE 1 END AS
 IsDatabaseSnapshot
            FROM sys.databases db
```

```
            INNER JOIN sys.database_recovery_status rs
            ON db.database_id = rs.database_id
            WHERE DB_NAME(db.database_id) NOT IN('tempdb') AND
            db.database_id NOT IN (SELECT DISTINCT DatabaseId FROM TABLE0)
        ),
        TABLE2 as(
            SELECT @Limit+count(DatabaseName) as TotalNumberOfDatabases from TABLE1
where TABLE1.IsPartOfSnapshot=1 and DatabaseName in ('master','model','msdb')
        )
        select top(select TotalNumberOfDatabases from TABLE2)
DatabaseName,CreateDate,DatabaseState,DatabaseId from TABLE1 where
TABLE1.IsPartOfSnapshot=1
        ORDER BY TABLE1.DatabaseID ASC
```

> **Note**
>
> Basis data yang hanya merupakan tautan simbolis juga dikecualikan dari basis data yang memenuhi syarat untuk operasi PITR. Kueri di atas tidak memfilter berdasarkan kriteria ini.

Log transaksi di Amazon S3

Periode retensi cadangan menentukan apakah log transaksi untuk instans DB RDS Custom for SQL Server secara otomatis diekstraksi dan diunggah ke Amazon S3. Nilai bukan nol berarti cadangan otomatis dibuat dan agen RDS Custom mengunggah log transaksi ke S3 setiap 5 menit.

File log transaksi pada S3 dienkripsi saat diam menggunakan AWS KMS key yang Anda berikan saat membuat instans DB. Untuk informasi selengkapnya, lihat Melindungi data menggunakan enkripsi sisi server di Panduan Pengguna Amazon Simple Storage Service.

Log transaksi untuk setiap basis data diunggah ke bucket S3 bernama `do-not-delete-rds-custom-`*`$ACCOUNT_ID`*`-`*`$REGION`*`-`*`unique_identifier`*. Direktori `RDSCustomForSQLServer/Instances/`*`DB_instance_resource_ID`* di bucket S3 berisi dua subdirektori:

- `TransactionLogs` — Berisi log transaksi untuk setiap basis data dan metadata masing-masing.

  Nama file log transaksi mengikuti pola *`yyyyMMddHHmm`*`.`*`database_id`*`.`*`timestamp`*, misalnya:

  ```
  202110202230.11.1634769287
  ```

Nama file yang sama dengan akhiran `_metadata` berisi informasi tentang log transaksi seperti nomor urut log, nama basis data, dan `RdsChunkCount`. `RdsChunkCount` menentukan berapa banyak file fisik yang mewakili satu file log transaksi. Anda mungkin melihat file dengan sufiks `_0001`, `_0002`, dan sebagainya, yang berarti potongan fisik dari file log transaksi. Jika Anda ingin menggunakan potongan file log transaksi, pastikan untuk menggabungkan potongan setelah mengunduhnya.

Pertimbangkan skenario ketika Anda memiliki file berikut:

- `202110202230.11.1634769287`
- `202110202230.11.1634769287_0001`
- `202110202230.11.1634769287_0002`
- `202110202230.11.1634769287_metadata`

`RdsChunkCount` adalah `3`. Urutan untuk menggabungkan file adalah sebagai berikut: `202110202230.11.1634769287`, `202110202230.11.1634769287_0001`, `202110202230.11.1634769287_0002`.

- `TransactionLogMetadata` — Berisi informasi metadata tentang setiap iterasi ekstraksi log transaksi.

  File `RI.End` berisi informasi untuk semua basis data yang log transaksinya diekstraksi dan semua basis data yang ada tetapi tidak log transaksinya tidak diekstraksi. Nama file `RI.End` mengikuti pola *`yyyyMMddHHmm`*`.RI.End.`*`timestamp`*, misalnya:

  ```
  202110202230.RI.End.1634769281
  ```

Pemulihan PITR menggunakan AWS Management Console, AWS CLI, atau API RDS.

Anda dapat memulihkan instans DB RDS Custom for SQL Server ke suatu titik waktu menggunakan AWS Management Console, AWS CLI, atau API RDS.

Konsol

Cara memulihkan instans DB RDS Custom ke waktu tertentu

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.

2. Di panel navigasi, pilih Pencadangan otomatis.
3. Pilih instans DB RDS Custom yang ingin Anda pulihkan.
4. Untuk Tindakan, pilih Pulihkan ke titik waktu.

   Jendela Pulihkan ke titik waktu akan muncul.
5. Pilih Waktu pemulihan terbaru untuk memulihkan ke waktu terbaru yang dimungkinkan atau pilih Kustom untuk memilih waktu.

   Jika Anda memilih Kustom, masukkan tanggal dan waktu untuk memulihkan instans.

   Waktu ditampilkan dalam zona waktu lokal Anda, yang ditunjukkan dengan offset dari Waktu Universal Terkoordinasi (UTC). Misalnya, UTC-5 adalah Waktu Standar Timur/Waktu Musim Panas Tengah.
6. Untuk Pengidentifikasi instans DB, masukkan nama target instans DB RDS Custom yang dipulihkan. Nama harus unik.
7. Pilih opsi lain sesuai kebutuhan, seperti kelas instans DB.
8. Pilih Pulihkan ke titik waktu.

AWS CLI

Anda mengembalikan instans DB ke waktu tertentu dengan menggunakan point-in-time AWS CLI perintah restore-db-instance-to- untuk membuat instance RDS Custom DB baru.

Gunakan salah satu opsi berikut untuk menentukan cadangan yang akan dipulihkan dari:

- `--source-db-instance-identifier` *`mysourcedbinstance`*
- `--source-dbi-resource-id` *`dbinstanceresourceID`*
- `--source-db-instance-automated-backups-arn` *`backupARN`*

Opsi `custom-iam-instance-profile` diperlukan.

Contoh berikut memulihkan `my-custom-db-instance` ke instans DB baru bernama `my-restored-custom-db-instance` pada waktu yang ditentukan.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds restore-db-instance-to-point-in-time \
    --source-db-instance-identifier my-custom-db-instance\
    --target-db-instance-identifier my-restored-custom-db-instance \
    --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance \
    --restore-time 2022-10-14T23:45:00.000Z
```

Untuk Windows:

```
aws rds restore-db-instance-to-point-in-time ^
    --source-db-instance-identifier my-custom-db-instance ^
    --target-db-instance-identifier my-restored-custom-db-instance ^
    --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance ^
    --restore-time 2022-10-14T23:45:00.000Z
```

Menghapus snapshot RDS Custom for SQL Server

Anda dapat menghapus snapshot DB yang dikelola RDS Custom for SQL Server saat tidak lagi membutuhkannya. Prosedur penghapusan sama untuk instans DB Amazon RDS dan RDS Custom.

Snapshot Amazon EBS untuk biner dan volume root tetap ada di akun Anda untuk waktu yang lebih lama karena mungkin ditautkan ke beberapa instans yang berjalan di akun Anda atau ke snapshot RDS Custom for SQL Server lainnya. Snapshot EBS ini dihapus secara otomatis setelah tidak lagi terkait dengan sumber daya RDS Custom for SQL Server yang ada (instans DB atau cadangan).

Konsol

Cara menghapus snapshot instans DB RDS Custom

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Snapshot.
3. Pilih snapshot DB yang ingin Anda hapus.
4. Untuk Tindakan, pilih Hapus snapshot.
5. Pilih Hapus di halaman konfirmasi.

AWS CLI

Untuk menghapus snapshot RDS Custom, gunakan perintah. AWS CLI delete-db-snapshot

Opsi berikut diperlukan:

- `--db-snapshot-identifier` — Snapshot yang akan dihapus

Contoh berikut menghapus snapshot DB `my-custom-snapshot`.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds delete-db-snapshot \
  --db-snapshot-identifier my-custom-snapshot
```

Untuk Windows:

```
aws rds delete-db-snapshot ^
  --db-snapshot-identifier my-custom-snapshot
```

Menghapus cadangan otomatis RDS Custom for SQL Server

Anda dapat menghapus cadangan otomatis yang disimpan untuk RDS Custom for SQL Server saat tidak diperlukan lagi. Prosedurnya sama dengan prosedur untuk menghapus cadangan Amazon RDS.

Konsol

Untuk menghapus cadangan otomatis yang dipertahankan

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Pencadangan otomatis.
3. Pilih Dipertahankan.
4. Pilih cadangan otomatis yang dipertahankan yang ingin Anda hapus.
5. Untuk Tindakan, pilih Hapus.
6. Di halaman konfirmasi, masukkan **`delete me`** dan pilih Hapus.

AWS CLI

Anda dapat menghapus cadangan otomatis yang dipertahankan dengan menggunakan AWS CLI perintah delete-db-instance-automated-backup.

Opsi berikut digunakan untuk menghapus cadangan otomatis yang dipertahankan:

- `--dbi-resource-id` — Pengidentifikasi sumber daya untuk instans DB RDS Custom sumber.

  Anda dapat menemukan pengenal sumber daya untuk instance DB sumber dari cadangan otomatis yang dipertahankan dengan menggunakan AWS CLI perintah describe-db-instance-automated -backup.

Contoh berikut menghapus cadangan otomatis yang dipertahankan dengan pengidentifikasi sumber daya instans DB sumber `custom-db-123ABCEXAMPLE`.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds delete-db-instance-automated-backup \
    --dbi-resource-id custom-db-123ABCEXAMPLE
```

Untuk Windows:

```
aws rds delete-db-instance-automated-backup ^
    --dbi-resource-id custom-db-123ABCEXAMPLE
```

Memigrasikan basis data on-premise ke Amazon RDS Custom for SQL Server

Anda dapat menggunakan proses berikut untuk memigrasikan basis data Microsoft SQL Server on-premise ke Amazon RDS Custom for SQL Server menggunakan pencadangan dan pemulihan native:

1. Lakukan pencadangan lengkap basis data pada instans DB on-premise.
2. Unggah file cadangan ke Amazon S3.
3. Unggah file cadangan dari S3 ke instans DB RDS Custom for SQL Server Anda.
4. Kembalikan basis data menggunakan file cadangan yang diunduh pada instans DB RDS Custom for SQL Server.

Proses ini menjelaskan migrasi basis data dari on-premise ke RDS Custom for SQL Server menggunakan pencadangan dan pemulihan penuh native. Untuk mengurangi waktu cutover selama proses migrasi, Anda juga dapat mempertimbangkan untuk menggunakan cadangan diferensial atau log.

Untuk informasi umum tentang pencadangan dan pemulihan native untuk RDS for SQL Server, lihat Mengimpor dan mengekspor basis data SQL Server menggunakan pencadangan dan pemulihan native.

Topik

Prasyarat

Lakukan tugas berikut sebelum memigrasikan basis data:

1. Konfigurasi Remote Desktop Connection (RDP) untuk instans DB RDS Custom for SQL Server Anda. Untuk informasi selengkapnya, lihat Menghubungkan ke instans RDS Custom DB Anda menggunakan RDP.

2. Konfigurasikan akses ke Amazon S3 sehingga Anda dapat mengunggah dan mengunduh file cadangan basis data. Untuk informasi selengkapnya, lihat Mengintegrasikan instans DB Amazon RDS for SQL Server dengan Amazon S3.

Mencadangkan basis data on-premise

Anda menggunakan cadangan native SQL Server untuk mengambil cadangan penuh basis data pada instans DB on-premise.

Contoh berikut menunjukkan cadangan basis data yang disebut `mydatabase`, dengan opsi `COMPRESSION` yang ditentukan untuk mengurangi ukuran file cadangan.

Cara membuat cadangan basis data on-premise

1. Dengan menggunakan SQL Server Management Studio (SSMS), hubungkan ke instans SQL Server on-premise.
2. Jalankan perintah T-SQL berikut.

```
backup database mydatabase to
disk ='C:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL\Backup\mydb-full-compressed.bak'
with compression;
```

Mengunggah file cadangan ke Amazon S3

Anda menggunakan file AWS Management Console untuk mengunggah file cadangan `mydb-full-compressed.bak` ke Amazon S3.

Cara mengunggah file cadangan ke Amazon S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di https://console.aws.amazon.com/s3/.
2. Untuk Bucket, pilih nama bucket tempat Anda ingin mengunggah file cadangan.
3. Pilih Unggah.
4. Di jendela Unggah, lakukan salah satu hal berikut:
   - Seret dan lepaskan `mydb-full-compressed.bak` ke jendela Unggah.
   - Pilih Tambah file, pilih `mydb-full-compressed.bak`, lalu pilih Buka.

Amazon S3 mengunggah file cadangan Anda sebagai objek S3. Setelah unggahan selesai, Anda melihat pesan sukses di halaman Unggah: status.

Mengunduh file cadangan dari Amazon S3

Anda menggunakan konsol untuk mengunduh file cadangan dari S3 ke instans DB RDS Custom for SQL Server.

Cara mengunduh file cadangan dari S3

1. Gunakan RDP untuk menghubungkan ke instans DB RDS for SQL Server.
2. Masuk ke AWS Management Console dan buka konsol Amazon S3 di https://console.aws.amazon.com/s3/.
3. Dari daftar Bucket, pilih nama bucket yang berisi file cadangan Anda.
4. Pilih file cadangan `mydb-full-compressed.bak`.
5. Untuk Tindakan, pilih Unduh sebagai.
6. Buka menu konteks (klik kanan) untuk tautan yang diberikan, lalu pilih Simpan Sebagai.
7. Simpan `mydb-full-compressed.bak` ke direktori `D:\rdsdbdata\BACKUP`.

Memulihkan file cadangan ke instans DB RDS Custom for SQL Server

Anda menggunakan pemulihan SQL Server native untuk memulihkan file cadangan ke instans DB RDS Custom for SQL Server.

Dalam contoh ini, opsi `MOVE` ditentukan karena direktori data dan file log berbeda dari instans DB on-premise.

Cara mengembalikan file cadangan

1. Gunakan SSMS untuk menghubungkan ke instans DB RDS for SQL Server.
2. Jalankan perintah T-SQL berikut.

```
restore database mydatabase from disk='D:\rdsdbdata\BACKUP\mydb-full-
compressed.bak'
with move 'mydatabase' to 'D:\rdsdbdata\DATA\mydatabase.mdf',
move 'mydatabase_log' to 'D:\rdsdbdata\DATA\mydatabase_log.ldf';
```

Memutakhirkan instans basis data untuk Amazon RDS Custom for SQL Server

Anda dapat memutakhirkan instans basis data Amazon RDS Custom for SQL Server dengan mengubahnya agar menggunakan versi mesin basis data baru, sebagaimana Anda lakukan untuk Amazon RDS.

Keterbatasan yang sama untuk memutakhirkan instans basis data RDS Custom for SQL Server berlaku sebagaimana mengubah instans basis data RDS Custom for SQL Server secara umum. Untuk informasi selengkapnya, lihat Memodifikasi instans DB RDS Custom for SQL Server.

Lihat informasi umum tentang memutakhirkan instans basis data di Meng-upgrade versi mesin instans DB.

Peningkatan versi utama

Amazon RDS Custom for SQL Server saat ini mendukung peningkatan versi utama berikut.

Versi saat ini	Versi upgrade yang didukung
SQL Server 2019	SQL Server 2022

Anda dapat menggunakan kueri AWS CLI, seperti contoh berikut, untuk menemukan upgrade yang tersedia untuk versi mesin basis data tertentu.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds describe-db-engine-versions \
    --engine sqlserver-se \
    --engine-version 15.00.4322.2.v1 \
    --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" \
    --output table
```

Untuk Windows:

```
aws rds describe-db-engine-versions ^
    --engine sqlserver-se ^
```

```
    --engine-version 15.00.4322.2.v1 ^
    --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" ^
    --output table
```

Tingkat kompatibilitas basis data

Anda dapat menggunakan basis data Microsoft SQL Server untuk menyesuaikan beberapa perilaku basis data guna meniru versi SQL Server yang lebih lama. Untuk informasi selengkapnya, lihat Compatibility level dalam dokumentasi Microsoft.

Saat Anda meng-upgrade instans DB, semua basis data yang ada tetap memiliki tingkat kompatibilitas aslinya. Misalnya, jika Anda meningkatkan dari SQL Server 2019 ke SQL Server 2022, semua database yang ada memiliki tingkat kompatibilitas 150. Setiap database baru yang dibuat setelah upgrade memiliki tingkat kompatibilitas 160.

Anda dapat mengubah tingkat kompatibilitas basis data dengan menggunakan perintah ALTER DATABASE. Misalnya, untuk mengubah database bernama `customeracct` agar kompatibel dengan SQL Server 2022, keluarkan perintah berikut:

```
ALTER DATABASE customeracct SET COMPATIBILITY_LEVEL = 160
```

Memecahkan masalah basis data untuk Amazon RDS Custom for SQL Server

Model tanggung jawab bersama RDS Custom menyediakan akses tingkat shell OS dan akses administrator basis data. RDS Custom menjalankan sumber daya di akun Anda, tidak seperti Amazon RDS, yang menjalankan sumber daya di akun sistem. Bersama akses yang lebih besar datang tanggung jawab yang lebih besar. Pada bagian-bagian berikut, Anda dapat mempelajari cara memecahkan masalah pada instans basis data Amazon RDS Custom for SQL Server.

> **Note**
>
> Bagian ini menjelaskan cara memecahkan masalah RDS Custom for SQL Server. Untuk pemecahan masalah RDS Custom for Oracle, lihat Memecahkan masalah basis data untuk Amazon RDS Custom for Oracle.

Topik

Menampilkan peristiwa RDS Custom

Prosedur untuk melihat peristiwa adalah sama untuk instans basis data RDS Custom dan Amazon RDS. Untuk informasi selengkapnya, lihat Melihat peristiwa Amazon RDS.

Untuk melihat pemberitahuan acara khusus RDS menggunakan AWS CLI, gunakan `describe-events` perintah. RDS Custom memperkenalkan beberapa peristiwa baru. Kategori-kategori peristiwa sama dengan untuk Amazon RDS. Lihat daftar peristiwa di Kategori peristiwa dan pesan peristiwa Amazon RDS.

Contoh berikut mengambil perincian peristiwa-peristiwa yang telah terjadi untuk instans basis data RDS Custom yang ditentukan.

```
aws rds describe-events \
```

```
    --source-identifier my-custom-instance \
    --source-type db-instance
```

Berlangganan acara RDS Custom

Prosedur untuk berlangganan peristiwa sama untuk instans basis data RDS Custom dan Amazon RDS. Untuk informasi selengkapnya, lihat Berlangganan pemberitahuan peristiwa Amazon RDS.

Untuk berlangganan notifikasi peristiwa RDS Custom dengan menggunakan CLI, gunakan perintah `create-event-subscription`. Sertakan parameter-parameter yang diperlukan berikut:

- `--subscription-name`
- `--sns-topic-arn`

Contoh berikut membuat pelangganan untuk peristiwa-peristiwa pencadangan dan pemulihan untuk sebuah instans basis data RDS Custom di akun AWS saat ini. Notifikasi dikirim ke topik Amazon Simple Notification Service (Amazon SNS), yang ditentukan oleh `--sns-topic-arn`.

```
aws rds create-event-subscription \
    --subscription-name my-instance-events \
    --source-type db-instance \
    --event-categories '["backup","recovery"]' \
    --sns-topic-arn arn:aws:sns:us-east-1:123456789012:interesting-events
```

Memecahkan masalah kesalahan CEV untuk RDS Custom for SQL Server

Ketika Anda mencoba untuk membuat CEV, itu mungkin gagal. Dalam hal ini, RDS Custom menerbitkan pesan peristiwa `RDS-EVENT-0198`. Lihat informasi yang lebih lengkap tentang penampilan peristiwa RDS di Kategori peristiwa dan pesan peristiwa Amazon RDS.

Gunakan informasi berikut untuk membantu Anda mengatasi penyebab yang mungkin.

Pesan	Saran pemecahan masalah		
`Custom Engine Version creation expected a Sysprep'd AMI. Retry creation using a Sysprep'd AMI.`	Jalankan Sysprep pada instans EC2 yang Anda buat dari AMI. Lihat informasi yang lebih lengkap tentang penyiapan AMI dengan menggunakan Sysprep di Membuat		

Pesan	Saran pemecahan masalah		
	Amazon Machine Image (AMI) terstandarkan dengan menggunakan Sysprep.		
`EC2 Image permissions for image (AMI_ID) weren't found for customer (Customer_ID). Verify customer (Customer_ID) has valid permissions on the EC2 Image.`	Periksa bahwa akun dan profil Anda yang digunakan untuk pembuatan memiliki izin-izin yang diperlukan pada `create EC2 Instance` dan `Describe Images` untuk AMI yang dipilih.		
`Failed to rebuild databases with server collation (collation name) due to missing setup.exe file for SQL Server.`	Periksa bahwa file `setup` tersedia di `C:\Program Files\Microsoft SQL Server\nnn\Setup Bootstrap\SQLnnnn\setup.exe` .		
`Image (AMI_ID) doesn't exist in your account (ACCOUNT_ID). Verify (ACCOUNT_ID) is the owner of the EC2 image.`	Pastikan bahwa AMI ada di akun pelanggan yang sama.		
`Image id (AMI_ID) isn't valid. Specify a valid image id, and try again.`	Nama AMI tidak benar. Pastikan bahwa ID AMI yang benar disediakan.		

Pesan	Saran pemecahan masalah
`Image (AMI_ID) operating system platform isn't supported. Specify a valid image, and try again.`	Pilih AMI yang didukung yang memiliki Windows Server with SQL Server edisi Enterprise, Standard, atau Web. Pilih AMI dengan salah satu kode operasi penggunaan berikut dari EC2 Marketplace: • RunInstances:0102 - Windows dengan SQL Server Enterprise • RunInstances:0006 - Windows dengan SQL Server Standar • RunInstances0202 - Windows dengan SQL Server Web
`SQL Server Web Edition isn't supported for creating a Custom Engine Version using Bring Your Own Media. Specify a valid image, and try again.`	Gunakan AMI yang berisi edisi SQL Server yang didukung. Untuk informasi selengkapnya, lihat Dukungan versi untuk CEV RDS Custom for SQL Server.
`The custom engine version can't be the same as the OEV engine version. Specify a valid CEV, and try again.`	RDS Custom klasik untuk versi-versi mesin SQL Server tidak didukung. Misalnya, versi 15.00.4073.23.v1. Gunakan nomor versi yang didukung.
`The custom engine version isn't in an active state. Specify a valid CEV, and try again.`	CEV harus dalam keadaan `AVAILABLE` untuk menyelesaikan operasi. Ubah CEV dari `INACTIVE` ke `AVAILABLE` .

Pesan	Saran pemecahan masalah
`The custom engine version isn't valid for an upgrade. Specify a valid CEV with an engine version greater or equal to (X), and try again.`	CEV target tidak valid. Periksa persyaratan untuk jalur peningkatan yang valid.
`The custom engine version isn't valid. Names can include only lowercase letters (a-z), dashes (-), underscores (_), and periods (.). Specify a valid CEV, and try again.`	Ikuti konvensi penamaan CEV yang diharuskan. Untuk informasi selengkapnya, lihat Persyaratan untuk CEV RDS Custom for SQL Server.
`The custom engine version isn't valid. Specify valid database engine version, and try again. Example: 15.00.4073.23-cev123.`	Tersedia versi mesin basis data yang tidak didukung. Gunakan versi mesin basis data yang didukung.
`The expected architecture is (X) for image (AMI_ID), but architecture (Y) was found.`	Gunakan AMI yang dibangun di atas arsitektur x86_64.
`The expected owner of image (AMI_ID) is customer account ID (ACCOUNT_ID), but owner (ACCOUNT_ID) was found.`	Buat instans EC2 dari AMI yang izinnya Anda miliki. Jalankan Sysprep pada instans EC2 untuk membuat dan menyimpan citra dasar.
`The expected platform is (X) for image (AMI_ID), but platform (Y) was found.`	Gunakan AMI yang dibangun dengan platform Windows.

Pesan	Saran pemecahan masalah
`The expected root device type is (X) for image %s, but root device type (Y) was found.`	Buat AMI dengan tipe perangkat EBS.
`The expected SQL Server edition is (X), but (Y) was found.`	Pilih AMI yang didukung yang memiliki Windows Server with SQL Server edisi Enterprise, Standard, atau Web. Pilih AMI dengan salah satu kode operasi penggunaan berikut dari EC2 Marketplace: • RunInstances:0102 - Windows dengan SQL Server Enterprise • RunInstances:0006 - Windows dengan SQL Server Standar • RunInstances0202 - Windows dengan SQL Server Web
`The expected state is (X) for image (AMI_ID), but the following state was found: (Y).`	Pastikan bahwa AMI dalam keadaan `AVAILABLE` .
`The provided Windows OS name (X) isn’t valid. Make sure the OS is one of the following: (Y).`	Gunakan Windows OS yang didukung.
`Unable to find bootstrap log file in path.`	Periksa bahwa file log tersedia di `C:\Program Files\Microsoft SQL Server\nnn\Setup Bootstrap\Log\Summary.txt` .

Pesan	Saran pemecahan masalah		
`RDS expected a Windows build version greater than or equal to (X), but found version (Y)..`	Gunakan AMI dengan versi bangun OS minimum 14393.		
`RDS expected a Windows major version greater than or equal to (X), but found version (Y)..`	Gunakan AMI dengan versi utama OS minimum 10.0 atau lebih tinggi.		

Memperbaiki konfigurasi yang tidak didukung di RDS Custom for SQL Server

Akibat model tanggung jawab bersama, Anda bertanggung jawab untuk memperbaiki masalah konfigurasi yang menempatkan instans basis data RDS Custom for SQL Server Anda ke dalam keadaan `unsupported-configuration`. Jika masalahnya ada pada AWS infrastruktur, Anda dapat menggunakan konsol atau AWS CLI untuk memperbaikinya. Jika masalahnya ada pada sistem operasi atau konfigurasi basis data, Anda dapat masuk ke host untuk memperbaikinya.

> **Note**
>
> Bagian ini menjelaskan cara memperbaiki konfigurasi yang tidak didukung di RDS Custom for SQL Server. Lihat informasi yang lebih lengkap tentang RDS Custom for Oracle di Memperbaiki konfigurasi yang tidak didukung di RDS Custom for Oracle.

Pada tabel berikut, Anda dapat menemukan deskripsi notifikasi dan peristiwa yang dikirim oleh perimeter dukungan dan cara memperbaikinya. Semua notifikasi ini dan perimeter dukungan dapat berubah sewaktu-waktu. Lihat latar belakang perimeter dukungan di Perimeter dukungan RDS Custom. Lihat deskripsi peristiwa di Kategori peristiwa dan pesan peristiwa Amazon RDS.

Kode Acara	Area konfigurasi	Pesan peristiwa RDS	Proses validasi
`SP-S0000`	Manual Konfigurasi Tidak Didukung	Status instans RDS Custom	Untuk mengatasi masalah ini, buat kasus dukungan.

Kode Acara	Area konfigurasi	Pesan peristiwa RDS	Proses validasi
		DB disetel ke [Konfigurasi tidak didukung] karena:. X	
AWS Sumber daya (infrastruktur)			

Kode Acara	Area konfigurasi	Pesan peristiwa RDS	Proses validasi
SP-S1001	Status Instans EC2	Status instans RDS Custom DB disetel ke [Konfigurasi tidak didukung] karena: Instance EC2 yang mendasari %s telah dihentikan tanpa menghentikan instans RDS. Anda dapat menyelesaikan ini dengan memulai instans EC2 yang mendasarinya dan memastikan bahwa biner dan volume data dilampirkan. Jika niat Anda adalah untuk menghentikan instance RDS, pastikan instans EC2 yang mendasarinya berada dalam status AVAILABLE terlebih dahulu dan kemudian gunakan konsol RDS atau CLI untuk menghentikan instance RDS.	Untuk memeriksa status instans DB, gunakan konsol atau jalankan AWS CLI perintah berikut: `aws rds describe-db-instances \ --db-instance-identifier db-instance-name |grep DBInstanceStatus`

Kode Acara	Area konfigurasi	Pesan peristiwa RDS	Proses validasi

SP-S1002	Status Instans EC2	Status instans DB Kustom RDS disetel ke [Konfigurasi tidak didukung] karena: Status instans RDS DB disetel ke STOPPED tetapi instans EC2 yang mendasari %s telah dimulai. Anda dapat mengatasi ini dengan menghentikan instans EC2 yang mendasarinya. Jika niat Anda adalah memulai instance RDS, gunakan konsol atau CLI.	Gunakan AWS CLI perintah berikut untuk memeriksa status instans DB:
			`aws rds describe-db-instances \ --db-instance-id entifier db-instance-name |grep DBInstanceStatus`
			Anda juga dapat memeriksa status instans EC2 menggunakan konsol EC2.
			Untuk memulai instance DB, gunakan konsol atau jalankan AWS CLI perintah berikut:
			`aws rds start-db-instance \ --db-instance-id entifier db-instance-name`

Kode Acara	Area konfigurasi	Pesan peristiwa RDS	Proses validasi
SP-S1003	Kelas Instans EC2	Status instans RDS Custom DB disetel ke [Konfigurasi tidak didukung] karena: Ada ketidakco cokan antara kelas instans DB yang diharapkan dan dikonfigurasi dari host EC2. Anda dapat menyelesa ikan ini dengan memodifikasi kelas instans DB ke jenis kelas aslinya.	Gunakan perintah CLI berikut untuk memeriksa kelas instans DB yang diharapkan: `aws rds describe-db-instances \ --db-instance-id entifier db-instance-name |grep DBInstanc eClass`
SP-S1004	Volume Penyimpanan EBS Tidak Dapat Diakses	Status instans RDS Custom DB disetel ke [Konfigurasi tidak didukung] karena: Volume penyimpanan EBS asli %s yang dikaitkan dengan instans EC2 saat ini tidak dapat diakses.	

Kode Acara	Area konfigurasi	Pesan peristiwa RDS	Proses validasi
SP-S1005	Volume Penyimpanan EBS Terpisah	Status instans RDS Custom DB disetel ke [Konfigurasi tidak didukung] karena: Volume penyimpanan EBS asli “volume-id” tidak dilampirkan. Anda dapat mengatasinya dengan melampirkan volume EBS yang terkait dengan instans EC2.	Setelah memasang kembali volume EBS, gunakan perintah CLI berikut untuk memeriksa apakah volume EBS 'volume-id' terpasang dengan benar ke instance RDS: `aws ec2 describe-volumes \ --volume-ids volume-id | grep InstanceId`

Kode Acara	Area konfigurasi	Pesan peristiwa RDS	Proses validasi

SP-S1006	Ukuran Volume Penyimpanan EBS	Status instans RDS Custom DB disetel ke [Konfigurasi tidak didukung] karena: Ada ketidakcocokan antara pengaturan volume penyimpanan EBS “volume-id” yang diharapkan dan yang dikonfigurasi. Ukuran volume telah diubah secara manual pada tingkat EC2 dari nilai aslinya dari [%s]. Untuk mengatasi masalah ini, buat kasus dukungan.	Gunakan perintah CLI berikut untuk membandingkan ukuran volume detail volume 'volume-id' EBS dan detail instance RDS:
			`aws rds describe-db-instances \ --db-instance-id entifier db-instance-name |grep Allocated Storage`
			Gunakan perintah CLI berikut untuk melihat ukuran volume yang dialokasikan sebenarnya:
			`aws ec2 describe-volumes \ --volume-ids |grep Size`

Kode Acara	Area konfigurasi	Pesan peristiwa RDS	Proses validasi

SP-S1007	Konfigurasi Volume Penyimpanan EBS	Status instans RDS Custom DB disetel ke [Konfigurasi tidak didukung] karena: Ada ketidakcocokan antara pengaturan volume penyimpanan EBS “volume-id” yang diharapkan dan yang dikonfigurasi. Anda dapat menyelesaikan ini dengan memodifikasi konfigurasi volume penyimpanan EBS [IOPS, Throughput, Volume type] ke nilai aslinya dari [IOPS: %s, Throughput: %s, Volume type: %s] pada level EC2. Untuk modifikasi penyimpanan masa depan, gunakan konsol RDS atau CLI. Ukuran volume juga telah diubah	Gunakan perintah CLI berikut untuk membandingkan jenis volume detail volume 'volume-id' EBS dan detail instance RDS. Pastikan nilai pada level EBS cocok dengan nilai di level RDS:
			`aws rds describe-db-instances \ --db-instance-id entifier db-instance-name |grep StorageType`
			Untuk mendapatkan nilai yang diharapkan untuk Storage Throughput di level RDS:
			`aws rds describe-db-instances \ --db-instance-id entifier db-instance-name |grep StorageThroughput`
			Untuk mendapatkan nilai yang diharapkan untuk Volume IOPS di level RDS:
			`aws rds describe-db-instances \ --db-instance-id entifier db-instance-name |grep Iops`

Kode Acara	Area konfigurasi	Pesan peristiwa RDS	Proses validasi

		secara manual pada tingkat EC2 dari nilai aslinya [%s]. Untuk mengatasi masalah ini, buat kasus dukungan.	Untuk mendapatkan Jenis Penyimpanan saat ini di Tingkat EC2:
			`aws ec2 describe-volumes \ --volume-ids |grep VolumeType`
			Untuk mendapatkan nilai saat ini untuk Storage Throughput pada Level EC2:
			`aws ec2 describe-volumes \ --volume-ids |grep Throughput`
			Untuk mendapatkan nilai saat ini untuk Volume IOPS di Level EC2:
			`aws ec2 describe-volumes \ --volume-ids |grep Iops`

Kode Acara	Area konfigurasi	Pesan peristiwa RDS	Proses validasi

SP-S1008	Ukuran dan Konfigurasi Volume Penyimpanan EBS	Status instans RDS Custom DB disetel ke [Konfigurasi tidak didukung] karena: Ada ketidakcocokan antara pengaturan volume penyimpanan EBS “volume-id” yang diharapkan dan yang dikonfigurasi. Anda dapat menyelesaikan ini dengan memodifikasi konfigurasi volume penyimpanan EBS [IOPS, Throughput, Volume type] ke nilai aslinya dari [IOPS: %s, Throughput: %s, Volume type: %s] pada level EC2. Untuk modifikasi penyimpanan masa depan, gunakan konsol RDS atau CLI. Ukuran volume juga telah diubah	Gunakan perintah CLI berikut untuk membandingkan jenis volume detail volume 'volume-id' EBS dan detail instance RDS. Pastikan nilai pada level EBS cocok dengan nilai di level RDS:
			`aws rds describe-db-instances \ --db-instance-id entifier db-instance-name |grep StorageType`
			Untuk mendapatkan nilai yang diharapkan untuk Storage Throughput di level RDS:
			`aws rds describe-db-instances \ --db-instance-id entifier db-instance-name |grep StorageThroughput`
			Untuk mendapatkan nilai yang diharapkan untuk Volume IOPS di level RDS:
			`aws rds describe-db-instances \ --db-instance-id entifier db-instance-name |grep Iops`

Kode Acara	Area konfigurasi	Pesan peristiwa RDS	Proses validasi

		secara manual pada tingkat EC2 dari nilai aslinya [%s]. Untuk mengatasi masalah ini, buat kasus dukungan.	Untuk mendapatkan Jenis Penyimpanan saat ini di Tingkat EC2:
			`aws ec2 describe-volumes \ --volume-ids |grep VolumeType`
			Untuk mendapatkan nilai saat ini untuk Storage Throughput pada Level EC2:
			`aws ec2 describe-volumes \ --volume-ids |grep Throughput`
			Untuk mendapatkan nilai saat ini untuk Volume IOPS di Level EC2:
			`aws ec2 describe-volumes \ --volume-ids |grep Iops`
			Untuk mendapatkan Ukuran Volume yang Dialokasikan yang diharapkan:
			`aws rds describe-db-instances \ --db-instance-id entifier db-instance-name |grep Allocated Storage`

Kode Acara	Area konfigurasi	Pesan peristiwa RDS	Proses validasi

			Untuk mendapatkan Ukuran Volume yang Dialokasikan yang sebenarnya:
			`aws ec2 describe-volumes \`
			`--volume-ids |grep Size`

SP-S1009	Izin SQS	Status instans DB Kustom RDS disetel ke [Konfigurasi tidak didukung] karena izin: Amazon Simple Queue Service (SQS) Amazon Simple Queue Service (SQS) tidak ada untuk profil instans IAM. Anda dapat mengatasi nya dengan memastikan profil IAM yang terkait dengan host memiliki izin berikut: ["SQS: “, "SQS: “, "SQS: SendMessage “, "SQS: ReceiveMessage “, "SQS: DeleteMessage “]. GetQueueUrl	

Kode Acara	Area konfigurasi	Pesan peristiwa RDS	Proses validasi
SP-S1010	Titik Akhir VPC SQS	Status instans DB Kustom RDS disetel ke [Konfigurasi tidak didukung] karena: Kebijakan titik akhir VPC memblokir operasi Amazon Simple Queue Service (SQS). Anda dapat mengatasi nya dengan memodifikasi kebijakan titik akhir VPC Anda untuk mengizink an tindakan SQS yang diperlukan.	
Sistem Operasi			

Kode Acara	Area konfigurasi	Pesan peristiwa RDS	Proses validasi
SP-S2001	Status Layanan SQL	Status instans RDS Custom DB disetel ke [Konfigurasi tidak didukung] karena: Layanan SQL Server tidak dimulai. Anda dapat mengatasi ini dengan memulai kembali layanan SQL Server pada host. Jika instans DB ini adalah instans DB Multi-AZ dan restart gagal, maka hentikan dan mulai host untuk memulai failover.	

Kode Acara	Area konfigurasi	Pesan peristiwa RDS	Proses validasi

SP-S2002	Status Agen Kustom RDS	Status instans RDS Custom DB disetel ke [Konfigurasi tidak didukung] karena: Layanan Agen Kustom RDS tidak diinstal atau tidak dapat dimulai. Anda dapat mengatasi ini dengan meninjau Windows Event Log untuk menentukan mengapa layanan tidak akan dimulai, dan mengambil langkah-langkah yang tepat untuk memperbai ki masalah. Untuk bantuan tambahan, buat kasus dukungan.	Masuk ke host dan pastikan bahwa agen RDS Custom berjalan.
			
			Anda dapat menggunakan perintah berikut untuk melihat status agen.
			
			`$name = "RDSCustomAgent"`
			`$service = Get-Service $name`
			`Write-Host $service.Status`
			
			Jika statusnya tidak `Running`, Anda dapat memulai layanan dengan perintah berikut:
			
			`Start-Service $name`
			
			Jika agen tidak dapat memulai, periksa Acara Windows untuk melihat mengapa tidak dapat dimulai. Agen membutuhk an pengguna Windows untuk memulai layanan. Pastikan pengguna Windows ada dan memiliki hak istimewa untuk menjalankan layanan.

Kode Acara	Area konfigurasi	Pesan peristiwa RDS	Proses validasi
`SP-S2003`	Status Agen SSM	Status instans DB Kustom RDS disetel ke [Konfigurasi tidak didukung] karena: Layanan Agen SSM Amazon tidak dapat dijangkau. Anda dapat memecahkan masalah ini dengan memeriksa status layanan dengan `Get-Service AmazonSSMAgent` PowerShell perintah, atau memulai layanan dengan. `Start-Service AmazonSSMAgent` Pastikan bahwa lalu lintas keluar HTTPS (port 443) ke titik akhir regional ssm, ssmmessages, dan ec2messages diizinkan.	Lihat informasi yang lebih lengkap di Memecahkan masalah Agen SSM. Untuk memecahkan masalah titik akhir SSM, lihat Tidak dapat terhubung ke titik akhir SSM dan Menggunakan ssm-cli untuk memecahkan masalah ketersediaan node terkelola.

Kode Acara	Area konfigurasi	Pesan peristiwa RDS	Proses validasi
SP-S2004	Login Agen Kustom RDS	SP-S2004Status instans RDS Custom DB disetel ke [Konfigurasi tidak didukung] karena: Masalah tak terduga terjadi dengan login SQL. "$HOSTNAME/RDSAgent" Untuk mengatasi masalah ini, buat kasus dukungan.	

Kode Acara	Area konfigurasi	Pesan peristiwa RDS	Proses validasi
SP-S2005	Zona waktu	Status instans DB Kustom RDS disetel ke [Konfigurasi tidak didukung] karena: Zona waktu pada Instans Amazon EC2 [%s] diubah. Anda dapat mengatasi nya dengan memodifikasi zona waktu kembali ke pengaturan yang ditentukan selama pembuatan instance. Jika Anda ingin membuat instance dengan zona waktu tertentu, lihat dokumentasi Kustom RDS.	Jalankan `Get-Timezone` PowerShell perintah untuk mengonfirmasi zona waktu. Untuk informasi selengkapnya, lihat Zona waktu lokal untuk instans DB RDS Custom for SQL Server.

Kode Acara	Area konfigurasi	Pesan peristiwa RDS	Proses validasi
SP-S2006	Versi Solusi Perangkat Lunak Ketersediaan Tinggi	Status instans RDS Custom DB disetel ke [Konfigurasi tidak didukung] karena: Solusi perangkat lunak ketersedi aan tinggi dari instance saat ini berbeda dari versi yang diharapkan. Untuk mengatasi masalah ini, buat kasus dukungan.	

Kode Acara	Area konfigurasi	Pesan peristiwa RDS	Proses validasi
SP-S2007	Konfigurasi Solusi Perangkat Lunak Ketersediaan Tinggi	Status instans RDS Custom DB disetel ke [Konfigurasi tidak didukung] karena: Pengaturan konfigurasi solusi perangkat lunak ketersediaan tinggi telah dimodifikasi ke nilai tak terduga pada instance %s. Untuk memperbaiki masalah ini, reboot instance EC2. Saat Anda me-reboot instans EC2, secara otomatis memperbarui pengaturan ke konfigurasi yang diperlukan untuk solusi perangkat lunak ketersediaan tinggi.	
Basis data			

Kode Acara	Area konfigurasi	Pesan peristiwa RDS	Proses validasi
SP-S3001	Protokol Memori Bersama SQL Server	Status instans RDS Custom DB diatur ke [Konfigurasi tidak didukung] karena: Protokol memori bersama SQL Server dinonaktifkan. Anda dapat mengatasi ini dengan mengaktifkan protokol memori bersama di SQL Server Configuration Manager.	Anda dapat memvalidasi ini dengan memeriksa: SQL Server Configuration Manager > SQL Server Network Configuration > Protocols for MSSQLSERVER> Shared Memory as Enabled. Setelah Anda mengaktifkan protokol, restart proses SQL Server.
SP-S3002	Layanan Master Key	Status instans RDS Custom DB diatur ke [Konfigurasi tidak didukung] karena: RDS Automation tidak dapat mengambil cadangan Service Master Key (SMK) sebagai bagian dari generasi SMK baru. Untuk mengatasi masalah ini, buat kasus dukungan.	

Kode Acara	Area konfigurasi	Pesan peristiwa RDS	Proses validasi
SP-S3003	Layanan Master Key	Status instans RDS Custom DB disetel ke [Konfigurasi tidak didukung] karena: Metadata yang terkait dengan Service Master Key (SMK) hilang atau tidak lengkap. Untuk mengatasi masalah ini, buat kasus dukungan.	

Kode Acara	Area konfigurasi	Pesan peristiwa RDS	Proses validasi

SP-S3004	Versi dan Edisi Mesin DB	Status instans RDS Custom DB disetel ke [Konfigurasi tidak didukung] karena: Edisi dan versi SQL Server saat ini tidak cocok dengan edisi SQL Server yang diharapkan: [%s] dan versi: [%s]. Anda dapat menyelesaikan ini dengan memastikan bahwa versi dan edisi SQL Server saat ini cocok dengan apa yang Anda tentukan selama pembuatan atau modifikasi terakhir dari instans DB.	Jalankan kueri berikut untuk mendapatkan versi SQL:
			`select @@version`
			Jalankan AWS CLI perintah berikut untuk mendapatkan versi mesin RDS SQL:
			`aws rds describe-db-instances \ --db-instance-id entifier db-instance-name |grep Engine`

Kode Acara	Area konfigurasi	Pesan peristiwa RDS	Proses validasi

SP-S3005	Edisi Mesin DB	Status instans RDS Custom DB disetel ke [Konfigurasi tidak didukung] karena: Edisi SQL Server saat ini tidak cocok dengan edisi SQL Server yang diharapkan: [%s]. Anda dapat mengatasinya dengan memastikan bahwa SQL Server Edition saat ini cocok dengan apa yang Anda tentukan selama pembuatan instance.	Jalankan kueri berikut untuk mendapatkan edisi SQL:
			Example
			`select @@version`
			Jalankan AWS CLI perintah berikut untuk mendapatkan edisi mesin RDS SQL:
			`aws rds describe-db-instances \ --db-instance-id entifier db-instance-name |grep Engine`

Kode Acara	Area konfigurasi	Pesan peristiwa RDS	Proses validasi

SP-S3006	Versi Mesin DB	Status instans RDS Custom DB disetel ke [Konfigurasi tidak didukung] karena: Versi SQL Server saat ini tidak cocok dengan versi SQL Server yang diharapkan: %s. Anda dapat menyelesaikan ini dengan memastikan bahwa versi SQL Server saat ini cocok dengan apa yang Anda tentukan selama pembuatan atau modifikasi terakhir dari instans DB. Untuk bantuan lebih lanjut, buat kasus dukungan.	Jalankan kueri berikut untuk mendapatkan versi SQL:
			
			Example
			
			`select @@version`
			
			Jalankan AWS CLI perintah berikut untuk mendapatkan versi mesin RDS SQL:
			
			`aws rds describe-db-instances \ --db-instance-identifier db-instance-name |grep EngineVersion`

Kode Acara	Area konfigurasi	Pesan peristiwa RDS	Proses validasi

SP-S3007	Lokasi file database	Status instans RDS Custom DB diatur ke [Konfigurasi tidak didukung] karena: File database dikonfigurasi di luar drive D:\. Anda dapat menyelesaikan ini dengan memastikan bahwa semua file database, termasuk ROW, LOG, FILESTREAM, dll... disimpan di drive D:\.	Jalankan kueri berikut untuk mencantumkan lokasi file database yang tidak berada di jalur default:
			`USE master;`
			`SELECT physical_name as files_not_in_default_path`
			`FROM sys.master_files`
			`WHERE SUBSTRING(physical_name,1,3)!='D:\';`

Pemecahan masalah **`Storage-Full`** di RDS Kustom untuk SQL Server

RDS Custom memantau ruang yang tersedia pada volume root (C:) dan data (D:) dari RDS Custom for SQL Server DB instance. RDS Custom memindahkan status instans ke `Storage-Full` status ketika salah satu volume memiliki ruang disk kurang dari 500 MiB yang tersedia. Untuk menskalakan penyimpanan instance, lihatMemodifikasi penyimpanan untuk instans DB RDS Custom for SQL Server.

> **Note**
>
> Instans dalam `Storage-Full` dapat memakan waktu hingga 30 menit untuk diselesaikan setelah penskalaan penyimpanan.

Menggunakan Amazon RDS on AWS Outposts

Amazon RDS on AWS Outposts memperluas basis data RDS for SQL Server, RDS for MySQL, dan RDS for PostgreSQL ke lingkungan AWS Outposts. AWS Outposts menggunakan perangkat keras yang sama dengan yang ada di Wilayah AWS publik untuk menghadirkan layanan AWS, infrastruktur, dan model operasi on-premise. Dengan RDS on Outposts, Anda dapat menyediakan instans DB terkelola di dekat aplikasi bisnis yang harus dijalankan secara on-premise. Untuk informasi selengkapnya tentang AWS Outposts, lihat AWS Outposts.

Untuk menyediakan dan mengelola instans DB RDS on Outposts on-premise, Anda menggunakan AWS Management Console, AWS CLI, dan RDS API yang sama seperti instans DB RDS yang berjalan di AWS Cloud. RDS on Outposts mengotomatiskan tugas, seperti penyediaan basis data, sistem operasi dan patch basis data, pencadangan, dan pengarsipan jangka panjang di Amazon S3.

RDS on Outposts mendukung pencadangan otomatis instans DB. Perlu adanya konektivitas jaringan antara Outpost Anda dan Wilayah AWS untuk mencadangkan dan memulihkan instans DB. Semua snapshot DB dan log transaksi dari Outpost disimpan di Wilayah AWS Anda. Dari Wilayah AWS Anda, Anda dapat memulihkan instans DB dari snapshot DB ke Outpost yang berbeda. Untuk informasi selengkapnya, lihat Pengantar cadangan.

RDS on Outposts mendukung pemeliharaan otomatis dan peningkatan instans DB. Untuk informasi selengkapnya, lihat Memelihara instans DB.

RDS on Outposts menggunakan enkripsi diam untuk instans DB dan snapshot DB yang menggunakan AWS KMS key Anda. Untuk informasi selengkapnya tentang enkripsi diam, lihat Mengenkripsi sumber daya Amazon RDS.

Secara default, instans EC2 di subnet Outpost dapat menggunakan Layanan DNS Amazon Route 53 untuk menyelesaikan nama domain ke alamat IP. Waktu resolusi dengan Route 53 mungkin akan lebih lama, tergantung latensi jalur antara Outpost Anda dan Wilayah AWS. Dalam kasus tersebut, Anda dapat menggunakan server DNS yang diinstal secara lokal di lingkungan on-premise Anda. Untuk informasi selengkapnya, lihat DNS dalam Panduan Pengguna AWS Outposts.

Saat konektivitas jaringan ke Wilayah AWS tidak tersedia, instans DB Anda akan terus berjalan secara lokal. Anda dapat terus mengakses instans DB menggunakan resolusi nama DNS dengan mengonfigurasi server DNS lokal sebagai server sekunder. Namun, Anda tidak dapat membuat instans DB baru atau mengambil tindakan baru pada instans DB yang sudah ada. Pencadangan otomatis tidak dapat dilakukan saat tidak ada konektivitas. Jika terjadi kegagalan instans DB, instans

DB tidak secara otomatis diganti hingga konektivitas pulih. Kami sarankan untuk memulihkan konektivitas jaringan secepatnya.

Topik

Prasyarat untuk Amazon RDS on AWS Outposts

Berikut adalah prasyarat penggunaan Amazon RDS on AWS Outposts:

- Instal AWS Outposts di pusat data on-premise Anda. Untuk informasi selengkapnya tentang AWS Outposts, lihat AWS Outposts.
- Pastikan ada setidaknya satu subnet yang tersedia untuk RDS on Outposts. Anda dapat menggunakan subnet yang sama untuk beban kerja lainnya.
- Pastikan koneksi jaringan antara Outpost Anda dan Wilayah AWS kuat.

Dukungan Amazon RDS on AWS Outposts untuk fitur Amazon RDS

Tabel berikut menjelaskan fitur Amazon RDS yang didukung oleh Amazon RDS on AWS Outposts.

Fitur	Didukung	Catatan	Informasi lain

Penyediaan instans DB	Ya	Anda hanya dapat membuat instans DB untuk mesin DB RDS for SQL Server, RDS for MySQL, dan RDS for PostgreSQL. Versi yang didukung sebagai berikut:
		• Microsoft SQL Server:
		  • 15.00.4043.16.v1 dan versi 2019 yang lebih tinggi
		  • 14.00.3294.2.v1 dan versi 2017 yang lebih tinggi
		  • 13.00.5820.21.v1 dan versi 2016 yang lebih tinggi
		• MySQL versi 8.0.28 dan MySQL versi 8.0 yang lebih tinggi
		• Semua PostgreSQL versi 16, 15, 14, dan 13, PostgreSQL versi 12.5, dan PostgreSQL versi 12 yang lebih tinggi
			Membuat instans DB untuk Amazon RDS on AWS Outposts

Terhubung ke instans	Ya	Beberapa versi TLS dan cipher enkripsi	Menghubungkan ke instans DB yang menjalank

Fitur	Didukung	Catatan	Informasi lain
DB Microsoft SQL Server dengan Microsoft SQL Server Management Studio		mungkin tidak aman. Untuk menonaktifkannya, ikuti petunjuk di Mengonfigurasi protokol keamanan dan cipher.	an mesin basis data Microsoft SQL Server
Mengubah kata sandi pengguna master	Ya	—	Memodifikasi instans DB Amazon RDS
Mengganti nama instans DB	Ya	—	Memodifikasi instans DB Amazon RDS
Me-reboot instans DB	Ya	—	Mem-boot ulang instans DB
Menghentikan instans DB	Ya	—	Menghentikan sementara instans DB Amazon RDS
Memulai instans DB	Ya	—	Memulai instans DB Amazon RDS yang sebelumnya dihentikan
Deployment multi-AZ	Ya	Deployment multi-AZ didukung pada instans DB MySQL dan PostgreSQL. Deployment multi-AZ tidak mendukung Perutean VPC Langsung (DVR).	Membuat instans DB untuk Amazon RDS on AWS Outposts Mengonfigurasi dan mengelola deployment Multi-AZ
Grup parameter DB	Ya	—	Bekerja dengan grup parameter

Fitur	Didukung	Catatan	Informasi lain
Replika baca	Ya	Replika baca didukung untuk instans DB MySQL dan PostgreSQL. Replika baca tidak mendukung Perutean VPC Langsung (DVR).	Membuat replika baca untuk Amazon RDS di AWS Outposts
Enkripsi diam	Ya	RDS on Outposts tidak mendukung instans DB yang tidak dienkripsi.	Mengenkripsi sumber daya Amazon RDS
Autentikasi basis data AWS Identity and Access Management (IAM)	Tidak	—	Autentikasi basis data IAM untuk MariaDB, MySQL, dan PostgreSQL
Mengaitkan peran IAM dengan instans DB	Tidak	—	add-role-to-dbperintah - instance AWS CLI AddRoleToOperasi DBInstance RDS API
Autentikasi Kerberos	Tidak	—	Autentikasi Kerberos
Memberi tag pada sumber daya Amazon RDS	Ya	—	Memberi tag pada sumber daya Amazon RDS
Grup opsi	Ya	—	Menggunakan grup opsi
Mengubah masa pemeliharaan	Ya	—	Memelihara instans DB

Fitur	Didukung	Catatan	Informasi lain
Peningkatan versi minor otomatis	Ya	—	Meng-upgrade versi mesin minor secara otomatis
Mengubah masa pencadangan	Ya	—	Pengantar cadangan Memodifikasi instans DB Amazon RDS
Mengubah kelas instans DB	Ya	—	Memodifikasi instans DB Amazon RDS
Mengubah alokasi penyimpanan	Ya	—	Memodifikasi instans DB Amazon RDS
Penskalaan otomatis penyimpanan	Ya	—	Mengelola kapasitas secara otomatis dengan penskalaan otomatis penyimpanan Amazon RDS

Fitur	Didukung	Catatan	Informasi lain
Snapshot instans DB manual dan otomatis	Ya	Anda dapat menyimpan cadangan otomatis dan snapshot manual di Wilayah AWS Anda. Atau Anda dapat menyimpannya secara lokal di Outpost. Pencadangan lokal didukung di instans DB MySQL dan PostgreSQL. Untuk menyimpan cadangan di Outpost, pastikan Anda telah mengonfigurasi Amazon S3 on Outposts. Pencadangan lokal tidak didukung untuk deploymen t instans Multi-AZ.	Membuat instans DB untuk Amazon RDS on AWS Outposts Amazon S3 on Outposts Membuat snapshot DB untuk instans DB Single-AZ
Memulihkan dari snapshot DB	Ya	Anda dapat menyimpan cadangan otomatis dan snapshot manual untuk instans DB yang dipulihkan di Wilayah AWS induk atau secara lokal di Outpost Anda.	Pertimbangan untuk memulihkan instans DB di Amazon RDS di AWS Outposts Memulihkan dari snapshot DB
Memulihkan instans DB dari Amazon S3	Tidak	—	Memulihkan cadangan ke instans DB MySQL
Mengekspor data snapshot ke Amazon S3	Tidak	—	Mengekspor data snapshot DB ke Amazon S3

Fitur	Didukung	Catatan	Informasi lain
oint-in-time Pemulihan P	Ya	Anda dapat menyimpan cadangan otomatis dan snapshot manual untuk instans DB yang dipulihka n di Wilayah AWS induk atau secara lokal di Outpost Anda, dengan satu pengecualian.	Pertimbangan untuk memulihkan instans DB di Amazon RDS di AWS Outposts Memulihkan instans DB dengan waktu yang ditentukan
Pemantauan yang ditingkat kan	Tidak	—	Memantau metrik OS dengan Pemantauan yang Disempurnakan
CloudWatch Pemantauan Amazon	Ya	Anda dapat melihat kumpulan metrik yang sama yang tersedia untuk basis data Anda di Wilayah AWS.	Memantau metrik Amazon RDS dengan Amazon CloudWatch
Menerbitkan log mesin database ke CloudWatch Log	Ya	—	Menerbitkan log basis data ke Log Amazon CloudWatc h
Pemberitahuan peristiwa	Ya	—	Menggunakan pemberita huan peristiwa Amazon RDS
Wawasan Performa Amazon RDS	Tidak	—	Memantau muatan DB dengan Wawasan Performa di Amazon RDS

Fitur	Didukung	Catatan	Informasi lain
Melihat atau mengunduh log basis data	Tidak	RDS on Outposts tidak mendukung melihat log basis data menggunak an konsol atau mendeskri psikan log basis data menggunakan AWS CLI atau RDS API. RDS on Outposts tidak mendukung pengunduhan log basis data menggunak an konsol, AWS CLI, atau RDS API.	Memantau file log Amazon RDS
Proksi Amazon RDS	Tidak	—	Menggunakan Proksi Amazon RDS
Prosedur tersimpan untuk Amazon RDS for MySQL	Ya	—	RDS for MySQL
Replikasi dengan basis data eksternal untuk RDS for MySQL	Tidak	—	Mengonfigurasi replikasi posisi file log biner dengan instans sumber eksternal
Cadangan asli dan pemulihan Amazon RDS for Microsoft SQL Server	Ya	—	Mengimpor dan mengekspor basis data SQL Server menggunak an pencadangan dan pemulihan native

Kelas instans DB yang didukung untuk Amazon RDS on AWS Outposts

Amazon RDS on AWS Outposts mendukung kelas instans DB berikut:

- Kelas instans DB tujuan umum
  - db.m5.24xlarge
  - db.m5.12xlarge
  - db.m5.4xlarge
  - db.m5.2xlarge
  - db.m5.xlarge
  - db.m5.large
- Kelas instans DB yang dioptimalkan memori
  - db.r5.24xlarge
  - db.r5.12xlarge
  - db.r5.4xlarge
  - db.r5.2xlarge
  - db.r5.xlarge
  - db.r5.large

Bergantung pada konfigurasi Outpost Anda, beberapa kelas ini mungkin tidak tersedia. Sebagai contoh, jika Anda belum membeli kelas db.r5 untuk Outpost Anda, maka Anda tidak dapat menggunakannya dengan RDS on Outposts.

Hanya penyimpanan SSD tujuan umum yang didukung untuk instans DB RDS on Outposts. Untuk informasi selengkapnya tentang kelas instans DB, lihat Kelas instans DB .

Amazon RDS mengelola pemeliharaan dan pemulihan untuk instans DB Anda dan membutuhkan kapasitas aktif pada Outpost untuk melakukannya. Kami menyarankan Anda mengonfigurasi instans EC2 N+1 untuk setiap kelas instans DB di lingkungan produksi Anda. RDS on Outposts dapat menggunakan kapasitas ekstra dari instans EC2 ini untuk operasi pemeliharaan dan perbaikan. Misalnya, jika lingkungan produksi Anda memiliki 3 kelas instans DB db.m5.large dan 5 db.r5.xlarge, kami sarankan untuk memiliki setidaknya 4 instans EC2 m5.large dan 6 instans EC2 r5.xlarge.

Untuk informasi selengkapnya, lihat Ketahanan di AWS Outposts dalam Panduan Pengguna AWS Outposts.

Alamat IP milik pelanggan untuk Amazon RDS on AWS Outposts

Amazon RDS on AWS Outposts menggunakan informasi tentang jaringan on-premise yang Anda berikan untuk membuat kolam alamat. Kolam ini dikenal sebagai kolam alamat IP milik pelanggan (kolom CoIP). Alamat IP milik pelanggan (CoIP) menyediakan konektivitas lokal atau eksternal untuk sumber daya di subnet Outpost Anda melalui jaringan on-premise. Untuk informasi selengkapnya tentang CoIP, lihat Alamat IP milik pelanggan dalam Panduan Pengguna AWS Outposts.

Setiap instans DB RDS on Outposts memiliki alamat IP privat untuk lalu lintas di dalam cloud privat virtual (VPC). Alamat IP privat ini tidak dapat diakses publik. Anda dapat menggunakan opsi Publik untuk menentukan apakah selain alamat IP privat, instans DB juga memiliki alamat IP publik. Penggunaan alamat IP publik untuk koneksi akan merutekannya melalui internet dan dapat mengakibatkan latensi tinggi dalam beberapa kasus.

Alih-alih menggunakan alamat IP privat dan publik ini, RDS on Outposts mendukung penggunaan CoIP untuk instans DB melalui subnet mereka. Saat menggunakan CoIP untuk instans DB RDS on Outposts, Anda terhubung ke instans DB dengan titik akhir instans DB. RDS on Outposts kemudian secara otomatis menggunakan CoIP untuk semua koneksi dari dalam dan luar VPC.

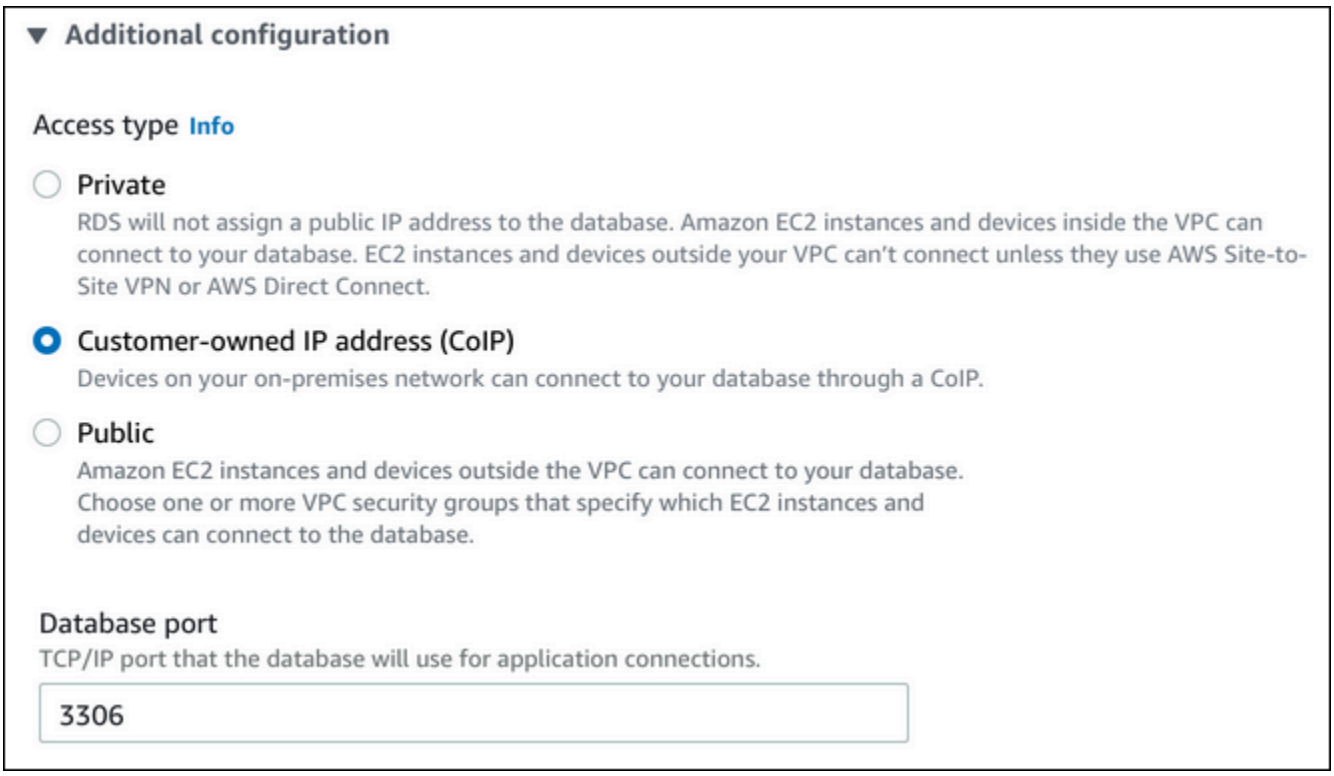
Berikut manfaat CoIP untuk instans DB RDS on Outposts:

- Latensi koneksi lebih rendah
- Keamanan yang ditingkatkan

Menggunakan CoIP

Anda dapat mengaktifkan atau menonaktifkan CoIP untuk instans DB RDS on Outposts menggunakan AWS Management Console, AWS CLI, atau RDS API:

- Dengan AWS Management Console, pilih pengaturan Alamat IP milik pelanggan (CoIP) di Jenis akses untuk menggunakan CoIP. Pilih salah satu pengaturan lain untuk menonaktifkannya.



- Dengan AWS CLI, gunakan opsi `--enable-customer-owned-ip | --no-enable-customer-owned-ip`.
- Dengan RDS API, gunakan parameter `EnableCustomerOwnedIp`.

Anda dapat mengaktifkan atau menonaktifkan CoIP saat melakukan salah satu tindakan berikut:

- Membuat instans DB

  Untuk informasi selengkapnya, lihat Membuat instans DB untuk Amazon RDS on AWS Outposts.
- Memodifikasi instans DB

  Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.
- Membuat replika baca

  Untuk informasi selengkapnya, lihat Membuat replika baca untuk Amazon RDS di AWS Outposts.
- Memulihkan instans DB dari snapshot DB

  Untuk informasi selengkapnya, lihat Memulihkan dari snapshot DB.
- Untuk memulihkan instans DB ke waktu tertentu

Untuk informasi selengkapnya, lihat Memulihkan instans DB dengan waktu yang ditentukan.

> **Note**
>
> Dalam beberapa kasus, Anda mungkin telah mengaktifkan CoIP untuk instans DB tetapi Amazon RDS tidak dapat mengalokasikan CoIP untuk instans DB tersebut. Dalam kasus seperti itu, status instans DB diubah menjadi jaringan tidak kompatibel. Untuk informasi selengkapnya tentang status instans DB, lihat Melihat status instans DB Amazon RDS.

Batasan

Batasan berikut berlaku untuk dukungan CoIP pada instans DB RDS on Outposts:

- Saat menggunakan CoIP untuk instans DB, pastikan aksesibilitas publik untuk instans DB tersebut telah dinonaktifkan.
- Pastikan bahwa aturan masuk untuk grup keamanan VPC Anda menyertakan rentang alamat CoIP (blok CIDR). Untuk informasi selengkapnya tentang pengaturan grup keamanan, lihat Memberikan akses ke instans DB di VPC Anda dengan membuat grup keamanan.
- Anda tidak dapat menetapkan CoIP dari kolam CoIP ke instans DB. Saat Anda menggunakan CoIP untuk instans DB, Amazon RDS secara otomatis menetapkan CoIP dari kolam CoIP ke instans DB.
- Anda harus menggunakan Akun AWS yang memiliki sumber daya Outpost (pemilik) atau berbagi sumber daya berikut dengan Akun AWS (konsumen) lain dalam organisasi yang sama:
  - Outpost
  - Tabel rute gateway lokal (LGW) untuk VPC instans DB
  - Kolam CoIP atau kolam untuk tabel rute LGW

  Untuk informasi selengkapnya, lihat Menggunakan sumber daya bersama AWS Outposts dalam Panduan Pengguna AWS Outposts.

Mengelola deployment Multi-AZ untuk Amazon RDS di AWS Outposts

Untuk deployment multi-AZ, Amazon RDS membuat instans DB primer di satu AWS Outpost. RDS secara sinkron mereplikasi data ke instans DB siaga di Outpost yang berbeda.

Deployment multi-AZ pada AWS Outposts beroperasi seperti deployment Multi-AZ di Wilayah AWS, tetapi dengan perbedaan sebagai berikut:

- Memerlukan koneksi lokal antara dua Outpost atau lebih.
- Memerlukan kolam IP milik pelanggan (CoIP). Untuk informasi selengkapnya, lihat Alamat IP milik pelanggan untuk Amazon RDS on AWS Outposts.
- Replikasi berjalan di jaringan lokal Anda.

Multi-AZ di AWS Outposts tersedia untuk semua versi MySQL dan PostgreSQL yang didukung di RDS di Outpost. Pencadangan lokal tidak didukung untuk deployment multi-AZ. Untuk informasi selengkapnya, lihat Membuat instans DB untuk Amazon RDS on AWS Outposts.

Mengelola model tanggung jawab bersama

Meski AWS menggunakan upaya komersial sewajarnya untuk menyediakan instans DB yang dikonfigurasi untuk ketersediaan tinggi, ketersediaan menggunakan model tanggung jawab bersama. Kemampuan RDS di Outpost untuk melakukan failover dan memperbaiki instans DB mengharuskan setiap Outpost Anda terhubung ke Wilayah AWS.

RDS di Outpost juga memerlukan konektivitas antara Outpost yang menjadi host instans DB primer dan Outpost yang menjadi host instans DB siaga untuk sinkronisasi replikasi. Segala dampak pada koneksi ini dapat menghambat failover RDS di Outpost.

Sebagai dampak dari replikasi data sinkron, latensi untuk deployment instans DB standar mungkin akan naik. Bandwidth dan latensi koneksi antara Outpost yang menjadi host instans DB primer dan Outpost yang menjadi host instans DB siaga secara langsung memengaruhi latensi. Untuk informasi selengkapnya, lihat Prasyarat.

Meningkatkan ketersediaan

Guna meningkatkan ketersediaan, kami sarankan untuk melakukan tindakan berikut:

- Alokasikan kapasitas tambahan yang cukup untuk aplikasi misi penting Anda agar dapat melakukan pemulihan dan failover jika terjadi masalah pada host yang mendasarinya. Ini berlaku untuk semua Outpost yang berisi subnet di grup subnet DB Anda. Untuk informasi selengkapnya, lihat Ketahanan di AWS Outposts.
- Sediakan konektivitas jaringan tambahan untuk Outpost Anda.
- Gunakan lebih dari dua Outpost. Memiliki lebih dari dua Outpost memungkinkan Amazon RDS memulihkan instans DB. RDS melakukan pemulihan ini dengan memindahkan instans DB ke Outpost lain jika Outpost saat ini mengalami kegagalan.
- Sediakan sumber daya ganda dan konektivitas jaringan tambahan untuk Outpost Anda.

Kami merekomendasikan hal berikut untuk jaringan lokal Anda:

- Latensi waktu pulang pergi (RTT) antara Outpost yang menjadi host instans DB primer Anda dan Outpost yang menjadi host instans DB siaga Anda secara langsung memengaruhi latensi penulisan. Pertahankan latensi RTT antara AWS Outposts dalam milidetik satu digit kecil. Sebaiknya kurang dari 5 milidetik, tetapi persyaratan Anda mungkin berbeda.

  Anda dapat menemukan dampak bersih terhadap latensi jaringan di metrik Amazon CloudWatch untuk `WriteLatency`. Untuk informasi selengkapnya, lihat CloudWatch Metrik Amazon untuk Amazon RDS.
- Ketersediaan koneksi di antara Outpost memengaruhi ketersediaan keseluruhan instans DB Anda. Sediakan konektivitas jaringan tambahan di antara Outpost.

Prasyarat

Berikut prasyarat deployment multi-AZ pada RDS di Outpost:

- Memiliki setidaknya dua Outpost yang terhubung melalui koneksi lokal dan dilampirkan ke Zona Ketersediaan yang berbeda di Wilayah AWS.
- Pastikan hal berikut ada di grup subnet DB Anda:
  - Setidaknya dua subnet dalam setidaknya dua Zona Ketersediaan dalam Wilayah AWS yang ditentukan.
  - Subnet hanya di Outpost.
  - Setidaknya dua subnet dalam setidaknya dua Outpost di dalam cloud privat virtual (VPC) yang sama.

- Kaitkan VPC instans DB Anda dengan semua tabel rute gateway lokal Anda. Pengaitan ini perlu dilakukan karena replikasi berjalan di jaringan lokal Anda menggunakan gateway lokal Outpost Anda.

  Sebagai contoh, misalnya VPC Anda berisi subnet-A di Outpost-A dan subnet-B di Outpost-B. Outpost-A menggunakan LocalGateway-A (LGW-A), dan Outpost-B menggunakan LocalGateway-B (LGW-B). LGW-A memiliki RouteTable-A, dan LGW-B memiliki RouteTable-B. Anda ingin menggunakan RouteTable-A dan RouteTable-B untuk lalu lintas replikasi. Untuk melakukannya, kaitkan VPC Anda dengan RouteTable-A dan RouteTable-B.

  Untuk informasi selengkapnya tentang cara pembuatan pengaitan, lihat perintah AWS CLI create-local-gateway-route-table-vpc-association Amazon EC2.
- Pastikan Outpost Anda menggunakan perutean IP milik pelanggan (CoIP). Setiap tabel rute juga harus memiliki setidaknya satu kolam alamat. Amazon RDS mengalokasikan alamat IP tambahan untuk setiap instans DB primer dan siaga untuk sinkronisasi data.
- Pastikan Akun AWS yang memiliki instans DB RDS memiliki tabel rute gateway lokal dan kolam CoIP. Atau pastikan akun tersebut menjadi bagian dari Resource Access Manager dengan akses ke tabel rute gateway lokal dan kolam CoIP.
- Pastikan alamat IP di kolam CoIP Anda dapat dirutekan dari satu gateway lokal Outpost ke gateway lokal lainnya.
- Pastikan blok CIDR VPC (misalnya, 10.0.0.0/4) dan blok CIDR kolam CoIP Anda tidak berisi alamat IP dari Kelas E (240.0.0.0/4). RDS menggunakan alamat IP ini secara internal.
- Pastikan penyiapan lalu lintas keluar dan lalu lintas masuk terkait sudah benar.

  RDS di Outpost membuat koneksi jaringan privat virtual (VPN) antara instans DB primer dan siaga. Agar koneksi ini berfungsi dengan baik, jaringan lokal Anda harus mengizinkan lalu lintas keluar dan lalu lintas masuk terkait untuk Internet Security Association and Key Management Protocol (ISAKMP). Hal ini dilakukan menggunakan User Datagram Protocol (UDP) port 500 dan IP Security (IPsec) Network Address Translation Traversal (NAT-T) menggunakan UDP port 4500.

Untuk informasi selengkapnya tentang CoIP, lihat Alamat IP milik pelanggan untuk Amazon RDS on AWS Outposts dalam panduan ini, dan Alamat IP milik pelanggan dalam Panduan Pengguna AWS Outposts.

Mengelola operasi API untuk izin Amazon EC2

Baik Anda menggunakan CoIP untuk instans DB di AWS Outposts atau tidak, RDS perlu mengakses sumber daya kolam CoIP Anda. RDS dapat memanggil operasi API izin EC2 berikut ini untuk CoIP atas nama Anda untuk deployment Multi-AZ:

- `CreateCoipPoolPermission` – Saat Anda membuat instans DB Multi-AZ di RDS di Outpost
- `DeleteCoipPoolPermission` – Saat Anda menghapus instans DB Multi-AZ di RDS di Outpost

Operasi API ini memberikan izin kepada atau menghapusnya dari akun RDS internal untuk mengalokasikan alamat IP elastis dari kolam CoIP yang ditetapkan oleh izin tersebut. Anda dapat melihat alamat IP ini menggunakan operasi API `DescribeCoipPoolUsage`. Untuk informasi selengkapnya tentang CoIP, lihat Alamat IP milik pelanggan untuk Amazon RDS on AWS Outposts dan Alamat IP milik pelanggan dalam Panduan Pengguna AWS Outposts.

RDS juga dapat memanggil operasi API izin EC2 berikut ini untuk tabel rute gateway lokal atas nama Anda untuk deployment Multi-AZ:

- `CreateLocalGatewayRouteTablePermission` – Saat Anda membuat instans DB Multi-AZ di RDS di Outpost
- `DeleteLocalGatewayRouteTablePermission` – Saat Anda menghapus instans DB Multi-AZ di RDS di Outpost

Operasi API ini memberikan izin kepada, atau menghapusnya dari, akun RDS internal untuk mengaitkan VPC RDS internal dengan tabel rute gateway lokal Anda. Anda dapat melihat kaitan tabel rute-VPC ini menggunakan operasi API `DescribeLocalGatewayRouteTableVpcAssociations`.

Membuat instans DB untuk Amazon RDS on AWS Outposts

Pembuatan instans DB Amazon RDS on AWS Outposts mirip dengan pembuatan instans DB Amazon RDS di AWS Cloud. Namun, grup subnet DB yang berkaitan dengan Outpost harus ditentukan.

Cloud privat virtual (VPC) berdasarkan layanan Amazon VPC dapat menjangkau semua Zona Ketersediaan dalam sebuah Wilayah AWS. Anda dapat memperluas VPC di Wilayah AWS ke Outpost Anda dengan menambahkan subnet Outpost. Untuk menambahkan subnet Outpost ke VPC, tentukan Amazon Resource Name (ARN) Outpost saat membuat subnet.

Sebelum membuat instans DB RDS on Outposts, Anda dapat membuat grup subnet DB yang menyertakan satu subnet yang dikaitkan dengan Outpost Anda. Tentukan grup subnet DB ini saat Anda membuat instans DB RDS on Outposts. Anda juga dapat membuat grup subnet DB baru saat membuat instans DB.

Untuk informasi selengkapnya tentang konfigurasi AWS Outposts, lihat Panduan Pengguna AWS Outposts.

Konsol

Membuat grup subnet DB

Buat grup subnet DB dengan satu subnet yang ditautkan dengan Outpost Anda.

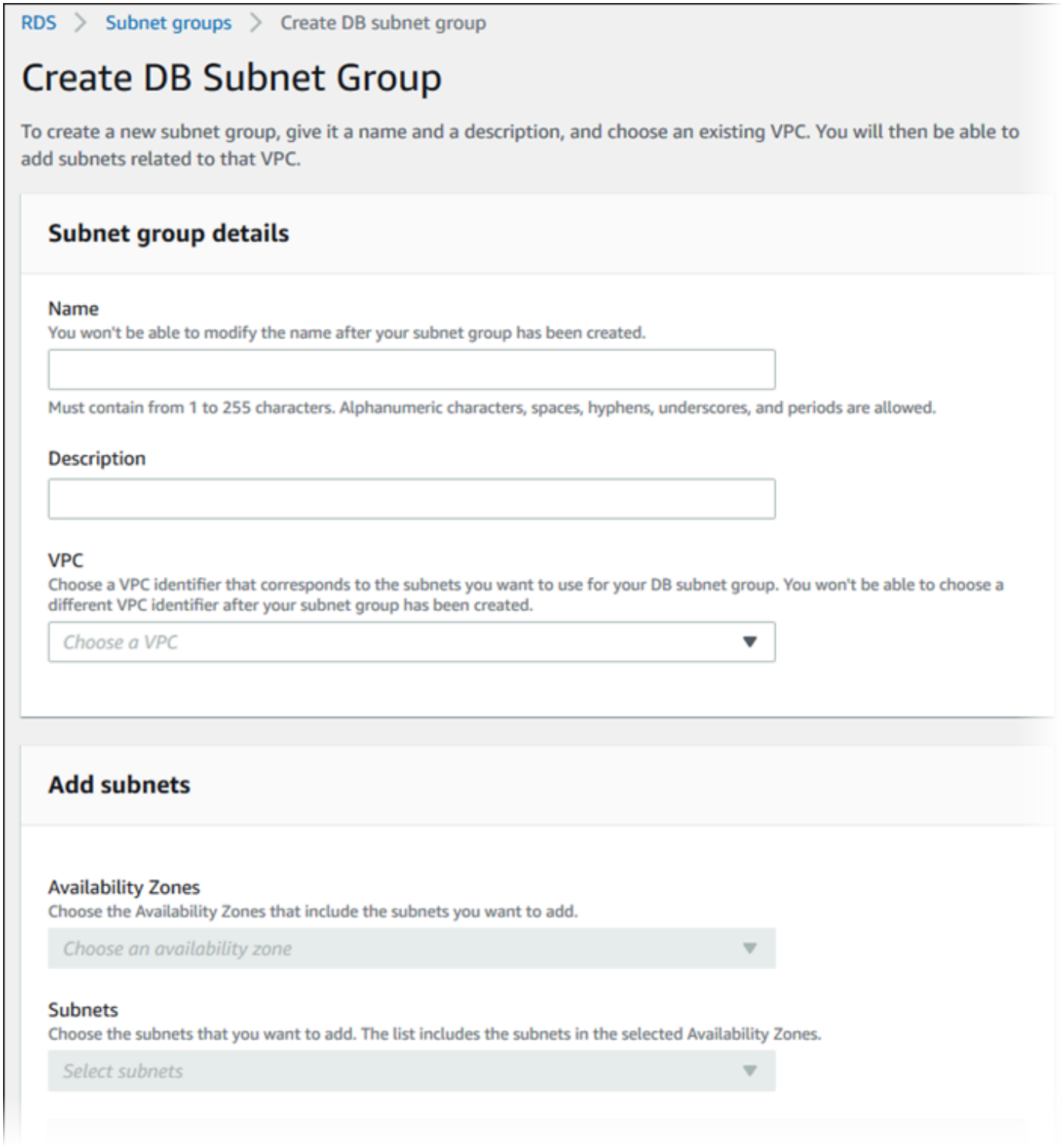
Anda juga dapat membuat grup subnet DB baru Outpost saat membuat instans DB. Jika Anda ingin melakukannya, lewati prosedur ini.

> **Note**
>
> Untuk membuat grup subnet DB untuk AWS Cloud, tentukan setidaknya dua subnet.

Untuk membuat grup subnet DB untuk Outpost Anda

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di sudut kanan atas konsol Amazon RDS, pilih Wilayah AWS tempat Anda ingin membuat grup subnet DB.
3. Pilih Grup subnet, lalu pilih Buat Grup Subnet DB.

Halaman Buat grup subnet DB akan muncul.



4. Untuk Nama, pilih nama grup subnet DB.
5. Untuk Deskripsi, pilih deskripsi untuk grup subnet DB.
6. Untuk VPC, pilih VPC yang akan dibuatkan grup subnet DB.

7. Untuk Zona Ketersediaan, pilih Zona Ketersediaan untuk Pos Anda.
8. Untuk Subnet, pilih subnet yang akan digunakan oleh RDS on Outposts.
9. Pilih Buat untuk membuat grup subnet DB.

Membuat instans DB RDS on Outposts

Buat instans DB, dan pilih Outpost untuk instans DB Anda.

Untuk membuat instans DB RDS on Outposts menggunakan konsol

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di sudut kanan atas konsol Amazon RDS, pilih Wilayah AWS untuk melampirkan Outpost yang akan dibuatkan instans DB.
3. Di panel navigasi, pilih Basis data.
4. Pilih Buat basis data.

   AWS Management Console mendeteksi Outpost yang tersedia yang telah Anda konfigurasi dan memberikan opsi On-premise di bagian Lokasi basis data.

   > **Note**
   >
   > Jika Anda belum mengonfigurasi Outpost, bagian Lokasi basis data tidak akan muncul atau opsi RDS on Outposts tidak tersedia di bagian Pilih metode pembuatan on-premise.

5. Untuk Lokasi basis data, pilih On-premise.
6. Untuk Metode pembuatan on-premise, pilih RDS on Outposts.
7. Tentukan pengaturan untuk Konektivitas Outpost. Pengaturan ini untuk Outpost yang menggunakan VPC yang memiliki grup subnet DB untuk instans DB Anda. VPC Anda harus berdasarkan layanan Amazon VPC.

   a. Untuk Cloud Privat Virtual (VPC), pilih VPC yang berisi grup subnet DB untuk instans DB Anda.
   b. Untuk Grup keamanan VPC, pilih grup keamanan Amazon VPC untuk instans DB Anda.
   c. Untuk Grup subnet DB, pilih grup subnet DB untuk instans DB Anda.

Anda dapat memilih grup subnet DB yang sudah ada yang dikaitkan dengan Outpost—misalnya, jika Anda melakukan prosedur di Membuat grup subnet DB.

Anda juga dapat membuat grup subnet DB baru untuk Outpost.

8. Untuk Deployment Multi-AZ, pilih Buat instans siaga (disarankan untuk penggunaan produksi) untuk membuat instans DB siaga di Outpost lain.

> **Note**
>
> Opsi ini tidak tersedia untuk Microsoft SQL Server.
> Jika Anda memilih untuk membuat deployment Multi-AZ, Anda tidak dapat menyimpan cadangan di Outpost Anda.

9. Di bagian Cadangan, lakukan hal berikut:

   a. Untuk Target pencadangan, pilih salah satu opsi berikut:

   - AWS Cloud untuk menyimpan cadangan otomatis dan snapshot manual di Wilayah AWS induk.
   - Outpost (on-premise) untuk membuat cadangan lokal.

   > **Note**
   >
   > Untuk menyimpan cadangan di Outpost, Outpost Anda harus memiliki kemampuan Amazon S3. Untuk informasi selengkapnya, lihat Amazon S3 on Outposts.
   > Pencadangan lokal tidak didukung untuk deployment multi-AZ atau replika baca.

   b. Pilih Aktifkan pencadangan otomatis untuk membuat point-in-time snapshot dari instans DB Anda.

   Saat pencadangan otomatis diaktifkan, Anda dapat memilih nilai untuk Periode penyimpanan cadangan dan Jendela pencadangan, atau menggunakan nilai default.

10. Tentukan pengaturan instans DB lainnya sesuai kebutuhan.

    Untuk informasi tentang pengaturan saat membuat instans DB, lihat Pengaturan untuk instans DB.

11. Pilih Buat basis data.

Halaman Basis data muncul. Banner memberitahukan bahwa instans DB Anda sedang dibuat, dan menampilkan tombol Lihat detail kredensial.

Melihat detail instans DB

Setelah membuat instans DB, Anda dapat melihat kredensial dan detail lainnya.

Untuk melihat detail instans DB

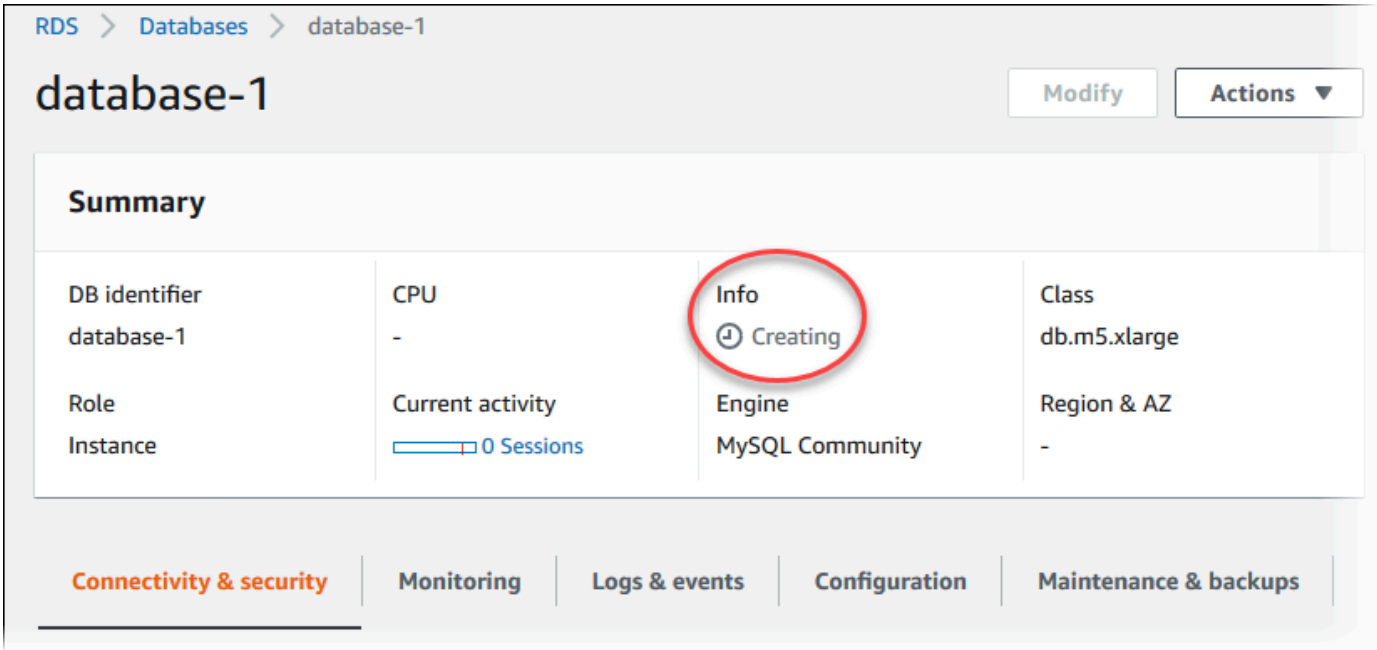
1. Untuk melihat nama pengguna utama dan kata sandi instans DB, pilih Lihat detail kredensial di halaman Basis data.

   Anda dapat terhubung ke instans DB sebagai pengguna utama menggunakan kredensial tersebut.

   > **Important**
   >
   > Anda tidak dapat melihat kata sandi pengguna utama lagi. Jika tidak direkam, Anda mungkin perlu mengubahnya. Untuk mengubah kata sandi pengguna utama setelah instans DB tersedia, modifikasi instans DB. Untuk informasi selengkapnya tentang modifikasi instans DB, lihat Memodifikasi instans DB Amazon RDS.

2. Pilih nama instans DB baru di halaman Basis data.

   Detail instans DB baru akan muncul di konsol RDS. Instans DB akan berstatus Sedang dibuat hingga proses pembuatannya selesai dan siap untuk digunakan. Setelah statusnya berubah menjadi Tersedia, Anda dapat terhubung ke instans DB ini. Perubahan status instans DB menjadi tersedia mungkin perlu beberapa menit, tergantung alokasi penyimpanan dan kelas instans DB.



Setelah instans DB tersedia, Anda dapat mengelolanya dengan cara yang sama seperti Anda mengelola instans DB RDS di AWS Cloud.

AWS CLI

Sebelum membuat instans DB baru di Outpost dengan AWS CLI, pertama-tama buat grup subnet DB untuk digunakan oleh RDS on Outposts.

Untuk membuat grup subnet DB untuk Outpost Anda

- Gunakan perintah create-db-subnet-group. Untuk `--subnet-ids`, tentukan grup subnet di Outpost yang akan digunakan oleh RDS on Outposts.

  Untuk Linux, macOS, atau Unix:

```
aws rds create-db-subnet-group \
    --db-subnet-group-name myoutpostdbsubnetgr \
    --db-subnet-group-description "DB subnet group for RDS on Outposts" \
    --subnet-ids subnet-abc123
```

  Untuk Windows:

```
aws rds create-db-subnet-group ^
    --db-subnet-group-name myoutpostdbsubnetgr ^
```

```
    --db-subnet-group-description "DB subnet group for RDS on Outposts" ^
    --subnet-ids subnet-abc123
```

Untuk membuat instans DB RDS on Outposts menggunakan AWS CLI

- Gunakan perintah create-db-instance. Tentukan Availability Zone untuk Outpost, grup keamanan Amazon VPC yang ditautkan dengan Outpost, dan grup subnet DB yang Anda buat untuk Outpost. Anda dapat menyertakan opsi berikut:
  - `--db-instance-identifier`
  - `--db-instance-class`
  - `--engine` – Mesin basis data. Gunakan salah satu nilai berikut:
    - MySQL - Pilih `mysql`.
    - PostgreSQL – Pilih `postgres`.
    - Microsoft SQL Server – Pilih `sqlserver-ee`, `sqlserver-se`, atau `sqlserver-web`.
  - `--availability-zone`
  - `--vpc-security-group-ids`
  - `--db-subnet-group-name`
  - `--allocated-storage`
  - `--max-allocated-storage`
  - `--master-username`
  - `--master-user-password`
  - `--multi-az | --no-multi-az` – (Opsional) Apakah akan membuat instans DB siaga di Zona Ketersediaan yang berbeda. Default-nya adalah `--no-multi-az`.

    Opsi `--multi-az` tidak tersedia untuk SQL Server.
  - `--backup-retention-period`
  - `--backup-target` – (Opsional) Tempat penyimpanan cadangan otomatis dan snapshot manual. Gunakan salah satu nilai berikut:
    - `outposts` – Simpan secara lokal di Outpost.
    - `region` – Simpan di Wilayah AWS induk. Ini adalah nilai default-nya.

Jika Anda menggunakan opsi `--multi-az`, Anda tidak dapat menggunakan `outposts` untuk `--backup-target`. Selain itu, instans DB tidak dapat memiliki replika baca jika Anda menggunakan `outposts` untuk `--backup-target`.

- `--storage-encrypted`
- `--kms-key-id`

Example

Contoh berikut membuat instans DB MySQL bernama `myoutpostdbinstance` dan cadangan disimpan di Outpost Anda.

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-instance \
    --db-instance-identifier myoutpostdbinstance \
    --engine-version 8.0.17 \
    --db-instance-class db.m5.large \
    --engine mysql \
    --availability-zone us-east-1d \
    --vpc-security-group-ids outpost-sg \
    --db-subnet-group-name myoutpostdbsubnetgr \
    --allocated-storage 100 \
    --max-allocated-storage 1000 \
    --master-username masterawsuser \
    --manage-master-user-password \
    --backup-retention-period 3 \
    --backup-target outposts \
    --storage-encrypted \
    --kms-key-id mykey
```

Untuk Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier myoutpostdbinstance ^
    --engine-version 8.0.17 ^
    --db-instance-class db.m5.large ^
    --engine mysql ^
    --availability-zone us-east-1d ^
    --vpc-security-group-ids outpost-sg ^
    --db-subnet-group-name myoutpostdbsubnetgr ^
```

```
    --allocated-storage 100 ^
    --max-allocated-storage 1000 ^
    --master-username masterawsuser ^
    --manage-master-user-password ^
    --backup-retention-period 3 ^
    --backup-target outposts ^
    --storage-encrypted ^
    --kms-key-id mykey
```

Untuk informasi tentang pengaturan saat membuat instans DB, lihat Pengaturan untuk instans DB.

API RDS

Untuk membuat instans DB baru di Outpost dengan RDS API, pertama-tama buat grup subnet DB untuk digunakan oleh RDS di Outposts dengan memanggil operasi createDB. SubnetGroup Untuk `SubnetIds`, tentukan grup subnet di Outpost yang akan digunakan oleh RDS on Outposts.

Selanjutnya, panggil operasi CreateDBInstance dengan parameter berikut. Tentukan Zona Ketersediaan untuk Outpost, grup keamanan Amazon VPC yang terkait dengan Outpost, dan grup subnet DB yang Anda buat untuk Outpost.

- `AllocatedStorage`
- `AvailabilityZone`
- `BackupRetentionPeriod`
- `BackupTarget`

  Jika Anda membuat deployment instans DB Multi-AZ, Anda tidak dapat menggunakan `outposts` untuk `BackupTarget`. Selain itu, instans DB tidak dapat memiliki replika baca jika Anda menggunakan `outposts` untuk `BackupTarget`.
- `DBInstanceClass`
- `DBInstanceIdentifier`
- `VpcSecurityGroupIds`
- `DBSubnetGroupName`
- `Engine`
- `EngineVersion`
- `MasterUsername`
- `MasterUserPassword`

- `MaxAllocatedStorage` (opsional)
- `MultiAZ` (opsional)
- `StorageEncrypted`
- `KmsKeyID`

Untuk informasi tentang pengaturan saat membuat instans DB, lihat Pengaturan untuk instans DB.

Membuat replika baca untuk Amazon RDS di AWS Outposts

Amazon RDS on AWS Outposts menggunakan fungsionalitas replikasi bawaan mesin MySQL dan PostgreSQL DB untuk membuat replika baca dari instans DB sumber. Instans DB sumber menjadi instans DB primer. Pembaruan pada instans DB primer disalin secara asinkron ke replika baca. Anda dapat mengurangi beban instans DB primer dengan merutekan kueri baca dari aplikasi Anda ke replika baca. Dengan replika baca, Anda dapat dengan mudah menskalakan di luar batasan kapasitas instans DB tunggal untuk beban kerja basis data dengan pembacaan intensif.

Saat Anda membuat replika baca dari instans DB RDS on Outposts, replika baca ini menggunakan alamat IP milik pelanggan (CoIP). Untuk informasi selengkapnya, lihat Alamat IP milik pelanggan untuk Amazon RDS on AWS Outposts.

Replika baca di RDS on Outposts memiliki batasan sebagai berikut:

- Anda tidak dapat membuat replika baca untuk RDS for SQL Server di instans DB RDS on Outposts.
- Replika baca lintas Wilayah tidak didukung di RDS on Outposts.
- Replika baca bertingkat tidak didukung di RDS on Outposts.
- Instans DB RDS on Outposts tidak dapat memiliki cadangan lokal. Target cadangan untuk instans DB sumber harus berupa Wilayah AWS Anda.
- Replika baca memerlukan kolam IP milik pelanggan (CoIP). Untuk informasi selengkapnya, lihat Alamat IP milik pelanggan untuk Amazon RDS on AWS Outposts.
- Replika baca di RDS di Outposts hanya dapat dibuat di virtual private cloud (VPC) yang sama dengan instans DB sumber.
- Replika baca di RDS di Outposts dapat ditemukan di Outpost yang sama atau Outpost lain di VPC yang sama dengan instans DB sumber.

Anda dapat membuat replika baca dari RDS pada instans DB Outposts menggunakan AWS Management Console, AWS CLI, atau RDS API. Untuk informasi selengkapnya tentang replika baca, lihat Menggunakan replika baca instans DB.

Konsol

Untuk membuat replika baca dari instans DB sumber

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data.
3. Pilih instans DB yang ingin Anda gunakan sebagai sumber untuk replika baca.
4. Untuk Tindakan, pilih Buat replika baca.
5. Untuk Pengidentifikasi instans DB, masukkan nama replika baca.
6. Tentukan pengaturan untuk Konektivitas Outpost. Pengaturan ini ditujukan untuk Outpost yang menggunakan cloud privat virtual (VPC) yang memiliki grup subnet DB untuk instans DB Anda. VPC Anda harus berdasarkan layanan Amazon VPC.
7. Pilih kelas instans DB Anda. Sebaiknya gunakan kelas dan jenis penyimpanan instans DB yang sama seperti atau lebih besar dari instans DB sumber untuk replika baca.
8. Untuk Deployment multi-AZ, pilih Buat instans siaga (direkomendasikan untuk penggunaan produksi) untuk membuat instans DB siaga di Zona Ketersediaan yang berbeda.

   Pembuatan replika baca sebagai instans Multi-AZ DB tidak bergantung pada apakah basis data sumber merupakan instans DB Multi-AZ.
9. (Opsional) Di bagian bawah Konektivitas, tentukan nilai untuk Grup Subnet dan Zona Ketersediaan.

   Jika Anda menentukan nilai baik untuk Grup Subnet maupun Zona Ketersediaan, replika baca akan dibuat di Outpost yang terkait dengan Zona Ketersediaan di grup subnet DB.

   Jika Anda menentukan nilai untuk Grup Subnet dan Tidak ada preferensi untuk Zona Ketersediaan, replika baca akan dibuat di Outpost acak di grup subnet DB.
10. Untuk AWS KMS key, pilih AWS KMS key pengenal kunci KMS.

    Replika baca harus dienkripsi.
11. Pilih opsi lain sesuai kebutuhan.
12. Pilih Buat replika baca.

Setelah replika baca dibuat, Anda dapat melihatnya di halaman Basis data di konsol RDS. Halaman tersebut menunjukkan Replika di kolom Peran.

AWS CLI

Untuk membuat replika baca dari sumber MySQL atau PostgreSQL DB instance, gunakan perintah - replica. AWS CLI create-db-instance-read

Anda dapat mengontrol tempat pembuatan replika baca dengan menentukan opsi `--db-subnet-group-name` dan `--availability-zone`:

- Jika Anda menentukan opsi `--db-subnet-group-name` dan `--availability-zone`, replika baca akan dibuat di Outpost yang terkait dengan Zona Ketersediaan di grup subnet DB.
- Jika Anda hanya menentukan opsi `--db-subnet-group-name` tetapi tidak dengan opsi `--availability-zone`, replika baca akan dibuat di Outpost acak di grup subnet DB.
- Jika Anda tidak menentukan keduanya, replika baca akan dibuat di Outpost yang sama dengan instans DB sumber RDS on Outposts.

Contoh berikut membuat replika dan menentukan lokasi replika baca dengan menyertakan opsi `--db-subnet-group-name` dan `--availability-zone`.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-instance-read-replica \
    --db-instance-identifier myreadreplica \
    --source-db-instance-identifier mydbinstance \
    --db-subnet-group-name myoutpostdbsubnetgr \
    --availability-zone us-west-2a
```

Untuk Windows:

```
aws rds create-db-instance-read-replica ^
    --db-instance-identifier myreadreplica ^
    --source-db-instance-identifier mydbinstance ^
    --db-subnet-group-name myoutpostdbsubnetgr ^
    --availability-zone us-west-2a
```

API RDS

Untuk membuat replika baca dari instans MySQL atau PostgreSQL DB sumber, panggil operasi Amazon RDS API CreateDB dengan parameter wajib berikut: InstanceReadReplica

- `DBInstanceIdentifier`
- `SourceDBInstanceIdentifier`

Anda dapat mengontrol tempat pembuatan replika baca dengan menentukan parameter `DBSubnetGroupName` dan `AvailabilityZone`:

- Jika Anda menentukan parameter `DBSubnetGroupName` dan `AvailabilityZone`, replika baca akan dibuat di Outpost yang terkait dengan Zona Ketersediaan di grup subnet DB.
- Jika Anda hanya menentukan parameter `DBSubnetGroupName` tetapi tidak dengan parameter `AvailabilityZone`, replika baca akan dibuat di Outpost acak di grup subnet DB.
- Jika Anda tidak menentukan keduanya, replika baca akan dibuat di Outpost yang sama dengan instans DB sumber RDS on Outposts.

Pertimbangan untuk memulihkan instans DB di Amazon RDS di AWS Outposts

Saat memulihkan instans DB di Amazon RDS di AWS Outposts, umumnya Anda dapat memilih lokasi penyimpanan untuk pencadangan otomatis dan snapshot manual instans DB yang dipulihkan.

- Saat memulihkan dari snapshot DB manual, Anda dapat menyimpan cadangan di Wilayah AWS induk atau secara lokal di Outpost Anda.
- Saat memulihkan dari cadangan otomatis (pemulihan titik waktu), opsi Anda lebih terbatas:
  - Jika memulihkan dari Wilayah AWS induk, Anda dapat menyimpan cadangan baik di Wilayah AWS atau di Outpost Anda.
  - Jika memulihkan dari Outpost Anda, Anda hanya menyimpan cadangan di Outpost.

Menggunakan Proksi Amazon RDS

Dengan menggunakan Proksi Amazon RDS, Anda dapat mengizinkan berbagai aplikasi untuk berkumpul dan berbagi koneksi basis data untuk meningkatkan kemampuan penskalaannya. Proksi RDS membuat aplikasi lebih tangguh terhadap kegagalan basis data dengan secara otomatis menghubungkan ke sebuah instans DB siaga sekaligus menjaga koneksi aplikasi. Dengan menggunakan RDS Proxy, Anda juga dapat menerapkan autentikasi AWS Identity and Access Management (IAM) untuk database, dan menyimpan kredensil dengan aman. AWS Secrets Manager

Dengan Proksi RDS, Anda dapat menangani lonjakan yang tidak dapat diprediksi dalam lalu lintas basis data. Jika tidak, lonjakan ini dapat menyebabkan masalah karena permintaan koneksi berlebihan atau koneksi baru yang dibuat sangatlah tinggi. Proksi RDS membangun kumpulan koneksi basis data dan menggunakan ulang koneksi dalam kumpulan ini. Pendekatan ini menghindari overhead memori dan CPU dari membuka koneksi basis data baru. Untuk melindungi basis data dari permintaan berlebihan, Anda dapat mengontrol jumlah koneksi basis data yang dibuat.

Proksi RDS mengantrekan atau membatasi koneksi aplikasi yang tidak dapat dilayani segera dari kumpulan koneksi. Meskipun latensi dapat meningkat, aplikasi Anda dapat terus diskalakan tanpa kegagalan mendadak atau membanjiri basis data. Jika permintaan koneksi melebihi batas yang Anda tentukan, Proksi RDS akan menolak koneksi aplikasi (yakni melepaskan beban). Selain itu, Proksi RDS akan mempertahankan performa yang dapat diprediksi untuk beban yang dapat dilayani RDS dengan kapasitas yang tersedia.

Anda dapat mengurangi overhead untuk memproses kredensial dan membangun koneksi yang aman untuk setiap koneksi baru. Proksi RDS dapat menangani beberapa dari pekerjaan itu untuk mewakili basis data.

Proksi RDS sepenuhnya kompatibel dengan versi mesin yang didukungnya. Anda dapat mengaktifkan Proksi RDS pada sebagian besar aplikasi tanpa perubahan kode.

Topik

- Memulai dengan Proksi RDS
- Mengelola Proksi RDS
- Bekerja dengan titik akhir Proksi Amazon RDS
- Memantau metrik Proxy RDS dengan Amazon CloudWatch
- Bekerja dengan peristiwa Proksi RDS
- Contoh baris perintah Proksi RDS
- Pemecahan masalah untuk Proksi RDS
- Penggunaan Proksi RDS dengan AWS CloudFormation

Ketersediaan wilayah dan versi

Ketersediaan dan dukungan fitur bervariasi di berbagai versi khusus dari setiap mesin basis data, dan di seluruh Wilayah AWS. Untuk informasi selengkapnya tentang versi dan ketersediaan Wilayah Amazon RDS dengan Proksi RDS, lihat Proksi Amazon RDS.

Kuota dan Pembatasan untuk Proksi RDS

Kuota dan batasan berikut berlaku untuk Proksi RDS:

- Anda dapat memiliki hingga 20 proxy untuk setiap ID AWS akun. Jika aplikasi Anda memerlukan lebih banyak proxy, Anda dapat meminta proxy tambahan dengan membuka tiket dengan organisasi Support. AWS
- Setiap proksi dapat memiliki hingga 200 rahasia Secrets Manager terkait. Sehingga, setiap proksi dapat terhubung ke hingga 200 akun pengguna yang berbeda pada waktu tertentu.
- Setiap proksi memiliki titik akhir default. Anda juga dapat menambahkan hingga 20 titik akhir proksi untuk setiap proksi. Anda dapat membuat, melihat, mengubah, dan menghapus titik akhir ini.
- Untuk instans DB RDS dalam konfigurasi replikasi, Anda dapat mengaitkan proksi dengan instans DB penulis saja, bukan dengan replika baca.
- Proksi RDS Anda harus berada dalam cloud privat virtual (VPC) yang sama seperti basis data. Proksi tersebut tidak dapat diakses publik, meskipun basis datanya dapat diakses publik. Misalnya, jika membuat prototipe basis data di host lokal, Anda tidak dapat terhubung ke proksi kecuali Anda menyiapkan persyaratan jaringan yang diperlukan untuk mengizinkan koneksi ke proksi. Ini karena host lokal Anda berada di luar VPC proksi.

- Anda tidak dapat menggunakan Proksi RDS dengan VPC yang penghuniannya diatur ke `dedicated`.
- Jika Anda menggunakan Proksi RDS dengan instans DB RDS yang autentikasi IAM-nya diaktifkan, periksa autentikasi pengguna. Pengguna yang terhubung melalui proksi harus melakukan autentikasi melalui kredensial masuk. Untuk detail tentang Secrets Manager dan dukungan IAM di Proksi RDS, lihat Menyiapkan kredensi database di AWS Secrets Manager dan Menyiapkan AWS Identity and Access Management kebijakan (IAM).
- Anda tidak dapat menggunakan Proksi RDS dengan DNS kustom saat menggunakan validasi nama host SSL.
- Setiap proksi dapat dikaitkan dengan satu klaster DB target. Namun, Anda dapat mengaitkan beberapa proksi dengan klaster DB yang sama.
- Pernyataan apa pun dengan teks berukuran lebih dari 16 KB menyebabkan proksi menyematkan sesi ke koneksi saat ini.
- Wilayah tertentu memiliki batasan Zona Ketersediaan (AZ) untuk dipertimbangkan saat membuat proksi. Wilayah AS Timur (Virginia Utara) tidak mendukung Proksi RDS di Zona Ketersediaan `use1-az3`. Wilayah AS Barat (California Utara) tidak mendukung Proksi RDS di Zona Ketersediaan `usw1-az2`. Saat memilih subnet sekaligus membuat proksi, pastikan Anda tidak memilih subnet di Zona Ketersediaan yang disebutkan di atas.

Untuk batasan tambahan untuk setiap mesin DB, lihat bagian berikut:

- Batasan tambahan untuk RDS for MariaDB
- Batasan tambahan untuk RDS for Microsoft SQL Server
- Batasan tambahan untuk RDS for MySQL
- Batasan tambahan untuk RDS for PostgreSQL

Batasan tambahan untuk RDS for MariaDB

Batasan tambahan berikut berlaku untuk Proksi RDS dengan basis data RDS for MariaDB:

- Saat ini, semua proksi mendengarkan di port 3306 untuk MariaDB. Proksi ini masih terhubung ke basis data Anda menggunakan port yang sudah ditentukan dalam pengaturan basis data.
- Anda tidak dapat menggunakan Proksi RDS dengan basis data MariaDB yang dikelola sendiri dalam instans Amazon EC2.

- Anda tidak dapat menggunakan Proksi RDS dengan instans DB RDS for MariaDB dengan parameter `read_only` dalam grup parameter DB-nya diatur ke `1`.
- Proksi RDS tidak mendukung mode terkompresi MariaDB. Misalnya, Proksi RDS tidak mendukung kompresi yang digunakan oleh opsi `--compress` atau `-C` perintah `mysql`.
- Beberapa pernyataan dan fungsi SQL dapat mengubah status koneksi tanpa menyebabkan penyematan. Untuk perilaku penyematan terbaru, lihat Menghindari penyematan.
- Proksi RDS tidak mendukung plugin `auth_ed25519` MariaDB.
- Proksi RDS tidak mendukung Keamanan Lapisan Pengangkutan (TLS) versi 1.3 untuk basis data MariaDB.
- Koneksi basis data yang memproses perintah `GET DIAGNOSTIC` mungkin menampilkan informasi yang tidak akurat saat Proksi RDS menggunakan kembali koneksi basis data yang sama untuk menjalankan kueri lain. Hal ini bisa terjadi ketika Proksi RDS me-multipleks koneksi basis data. Untuk informasi selengkapnya, lihat Ikhtisar konsep Proksi RDS.

> **Important**
>
> Untuk proksi yang terkait dengan basis data MariaDB, jangan atur parameter konfigurasi `sql_auto_is_null` ke `true` atau nilai bukan nol dalam kueri inisialisasi. Tindakan ini bisa menyebabkan perilaku aplikasi yang salah.

Batasan tambahan untuk RDS for Microsoft SQL Server

Batasan tambahan berikut berlaku untuk Proksi RDS dengan basis data RDS for Microsoft SQL Server:

- Jumlah rahasia Secrets Manager yang perlu dibuat untuk proksi bergantung pada kolasi yang digunakan instans DB Anda. Misalnya, misalkan instans DB Anda menggunakan kolasi peka huruf besar/kecil. Jika aplikasi Anda menerima "Admin" dan "admin," berarti proksi Anda memerlukan dua rahasia terpisah. Untuk informasi selengkapnya tentang pemeriksaan di SQL Server, lihat dokumentasi Microsoft SQL Server.
- Proksi RDS tidak mendukung koneksi yang menggunakan Active Directory.
- Anda tidak dapat menggunakan autentikasi IAM dengan klien yang tidak mendukung properti token. Untuk informasi selengkapnya, lihat Pertimbangan untuk terhubung ke proksi dengan Microsoft SQL Server.

- Hasil dari `@@IDENTITY`, `@@ROWCOUNT`, dan `SCOPE_IDENTITY` tidak selalu akurat. Sebagai solusi, ambil nilainya dalam pernyataan sesi yang sama untuk memastikan bahwa hasilnya menampilkan informasi yang benar.
- Jika koneksi menggunakan beberapa kumpulan hasil aktif (MARS), Proksi RDS tidak akan menjalankan kueri inisialisasi. Untuk informasi tentang MARS, lihat dokumentasi Microsoft SQL Server.
- RDS Proxy tidak mendukung RDS untuk instans SQL Server DB yang berjalan pada versi utama SQL Server 2014.

Batasan tambahan untuk RDS for MySQL

Batasan tambahan berikut berlaku untuk Proksi RDS dengan basis data RDS for MySQL:

- Proksi RDS tidak mendukung plugin autentikasi `sha256_password` dan `caching_sha2_password` MySQL. Plugin ini menerapkan hashing SHA-256 untuk kata sandi akun pengguna.
- Saat ini, semua proksi mendengarkan di port 3306 untuk MySQL. Proksi ini masih terhubung ke basis data Anda menggunakan port yang sudah ditentukan dalam pengaturan basis data.
- Anda tidak dapat menggunakan Proksi RDS dengan basis data MySQL yang dikelola sendiri dalam instans EC2.
- Anda tidak dapat menggunakan Proksi RDS dengan instans DB RDS for MySQL dengan parameter `read_only` dalam grup parameter DB-nya diatur ke 1.
- Proksi RDS tidak mendukung mode terkompresi MySQL. Misalnya, Proksi RDS tidak mendukung kompresi yang digunakan oleh opsi `--compress` atau `-C` perintah `mysql`.
- Koneksi basis data yang memproses perintah `GET DIAGNOSTIC` mungkin menampilkan informasi yang tidak akurat saat Proksi RDS menggunakan kembali koneksi basis data yang sama untuk menjalankan kueri lain. Hal ini bisa terjadi ketika Proksi RDS me-multipleks koneksi basis data.
- Beberapa pernyataan dan fungsi SQL seperti `SET LOCAL` dapat mengubah status koneksi tanpa menyebabkan penyematan. Untuk perilaku penyematan terbaru, lihat Menghindari penyematan.

> **Important**
>
> Untuk proksi yang terkait dengan basis data MySQL, jangan atur parameter konfigurasi `sql_auto_is_null` ke `true` atau nilai bukan nol dalam kueri inisialisasi. Tindakan ini bisa menyebabkan perilaku aplikasi yang salah.

Batasan tambahan untuk RDS for PostgreSQL

Batasan tambahan berikut berlaku untuk Proksi RDS dengan basis data RDS for PostgreSQL:

- Proksi RDS tidak mendukung filter penyematan sesi untuk PostgreSQL.
- Saat ini, semua proksi mendengarkan di port 5432 untuk PostgreSQL.
- Untuk PostgreSQL, Proksi RDS saat ini tidak mendukung pembatalan kueri dari klien dengan mengeluarkan `CancelRequest`. Misalnya, hal ini bisa terjadi ketika Anda membatalkan kueri yang berjalan lama dalam sesi psql interaktif dengan menggunakan Ctrl+C.
- Hasil dari fungsi lastval PostgreSQL tidak selalu akurat. Sebagai solusi, gunakan pernyataan INSERT dengan klausul `RETURNING`.
- Proksi RDS saat ini tidak mendukung mode replikasi streaming.
- Dengan RDS for PostgreSQL 16, modifikasi nilai `scram_iterations` secara khusus memengaruhi proses autentikasi antara proksi dan basis data. Khususnya, jika Anda `ClientPasswordAuthType` mengonfigurasinya`scram-sha-256`, penyesuaian apa pun yang dilakukan pada `scram_iterations` nilai tidak memengaruhi otentikasi client-to-proxy kata sandi. Sebaliknya, nilai iterasi untuk otentikasi client-to-proxy kata sandi ditetapkan pada 4096.

> **Important**
>
> Untuk proksi yang ada dengan basis data PostgreSQL, jika Anda mengubah autentikasi basis data untuk menggunakan `SCRAM` saja, proksi akan menjadi tidak tersedia selama maksimal 60 detik. Untuk menghindari masalah ini, lakukan salah satu hal berikut:
>
> - Pastikan basis data memungkinkan autentikasi `SCRAM` dan `MD5`.
> - Untuk hanya menggunakan autentikasi `SCRAM`, buat proksi baru, migrasi lalu lintas aplikasi Anda ke proksi baru, lalu hapus proksi yang sebelumnya terkait dengan basis data.

Merencanakan lokasi penggunaan Proksi RDS

Anda dapat menentukan instans, klaster, dan aplikasi DB mana yang mungkin paling banyak mendapatkan manfaat dari penggunaan Proksi RDS. Untuk melakukannya, pertimbangkan faktor-faktor berikut:

- Instans DB apa pun yang mengalami kesalahan "terlalu banyak koneksi" adalah kandidat yang baik untuk dikaitkan dengan proksi. Ini sering ditandai dengan nilai `ConnectionAttempts` CloudWatch metrik yang tinggi. Proksi tersebut memungkinkan aplikasi untuk membuka banyak koneksi klien sekaligus mengelola koneksi jangka panjang dalam jumlah lebih kecil ke klaster DB.
- Untuk instans DB yang menggunakan kelas AWS instans yang lebih kecil, seperti T2 atau T3, menggunakan proxy dapat membantu menghindari kondisi. out-of-memory Tindakan ini juga dapat membantu mengurangi overhead CPU untuk membangun koneksi. Kondisi ini dapat terjadi saat berurusan dengan koneksi dalam jumlah besar.
- Anda dapat memantau CloudWatch metrik Amazon tertentu untuk menentukan apakah instans DB mendekati jenis batas tertentu. Batasan ini ditujukan untuk jumlah koneksi dan memori terkait dengan pengelolaan koneksi. Anda juga dapat memantau CloudWatch metrik tertentu untuk menentukan apakah instans DB menangani banyak koneksi berumur pendek. Pembukaan dan penutupan koneksi tersebut dapat membebani overhead performa pada basis data Anda. Untuk informasi tentang metrik yang akan dipantau, lihat Memantau metrik Proxy RDS dengan Amazon CloudWatch.
- Fungsi AWS Lambda juga bisa menjadi kandidat yang tepat untuk penggunaan proksi. Fungsi ini sering membuat koneksi basis data pendek yang mendapatkan manfaat dari kumpulan koneksi yang ditawarkan oleh Proksi RDS. Anda dapat memanfaatkan autentikasi IAM apa pun yang Anda miliki untuk fungsi Lambda, alih-alih mengelola kredensial basis data dalam kode aplikasi Lambda.
- Aplikasi yang biasanya membuka dan menutup koneksi basis data dalam jumlah besar dan tidak memiliki mekanisme pengumpulan koneksi bawaan adalah kandidat yang tepat untuk menggunakan proksi.
- Aplikasi yang dapat mempertahankan koneksi terbuka dalam jumlah besar selama jangka waktu yang lama biasanya merupakan kandidat yang tepat untuk penggunaan proksi. Aplikasi dalam industri seperti Perangkat Lunak sebagai Layanan (SaaS) atau e-niaga sering kali meminimalkan latensi untuk permintaan basis data dengan membiarkan koneksi terbuka. Dengan Proksi RDS, aplikasi dapat mempertahankan lebih banyak koneksi tetap terbuka daripada saat terhubung langsung ke klaster DB.
- Mungkin Anda belum mengadopsi autentikasi IAM dan Secrets Manager karena kompleksitas penyiapan autentikasi tersebut untuk semua klaster DB. Jika demikian, Anda dapat terus

menerapkan metode autentikasi yang sudah ada dan mendelegasikan autentikasi ke proksi. Proksi dapat menerapkan kebijakan autentikasi koneksi klien untuk aplikasi tertentu. Anda dapat memanfaatkan autentikasi IAM apa pun yang Anda miliki untuk fungsi Lambda, alih-alih mengelola kredensial basis data dalam kode aplikasi Lambda.

- Proksi RDS dapat membantu membuat aplikasi lebih tangguh dan transparan terhadap kegagalan basis data. Proksi RDS melewati cache Sistem Nama Domain (DNS) untuk mengurangi waktu failover hingga 66% untuk instans DB Amazon RDS Multi-AZ. Proksi RDS juga secara otomatis merutekan lalu lintas ke instans basis data baru sekaligus mempertahankan koneksi aplikasi. Hal ini membuat failover lebih transparan untuk aplikasi.

Konsep dan terminologi Proksi RDS

Anda dapat menyederhanakan manajemen koneksi untuk instans DB Amazon RDS dengan menggunakan Proksi RDS.

Proksi RDS menangani lalu lintas jaringan antara aplikasi klien dan basis data. Tindakan ini dilakukan secara aktif terlebih dahulu dengan memahami protokol basis data. Lalu perilakunya disesuaikan berdasarkan operasi SQL dari aplikasi Anda dan serangkaian hasil dari basis data.

Proksi RDS mengurangi overhead memori dan CPU untuk manajemen koneksi pada basis data Anda. basis data membutuhkan lebih sedikit sumber daya memori dan CPU saat aplikasi membuka banyak koneksi secara bersamaan. Logika juga tidak dibutuhkan dalam aplikasi Anda untuk menutup dan membuka kembali koneksi yang idle dalam waktu yang lama. Demikian pula, logika aplikasi yang dibutuhkan untuk membangun kembali koneksi juga lebih sedikit jika terjadi masalah pada basis data.

Infrastruktur untuk Proksi RDS memiliki ketersediaan tinggi dan di-deploy pada berbagai Zona Ketersediaan (AZ). Komputasi, memori, dan penyimpanan untuk Proksi RDS tidak bergantung pada instans DB RDS. Independensi ini membantu menurunkan overhead pada server basis data Anda, sehingga dapat mencurahkan sumber dayanya untuk melayani beban kerja basis data. Sumber daya komputasi Proksi RDS bersifat nirserver, yang diskalakan secara otomatis berdasarkan beban kerja basis data Anda.

Topik

- Transaksi

Ikhtisar konsep Proksi RDS

Proksi RDS menangani infrastruktur untuk melakukan pengumpulan koneksi dan fitur lain yang dijelaskan di bagian berikutnya. Anda melihat proksi ditampilkan dalam konsol RDS pada halaman Proksi.

Setiap proksi menangani koneksi ke instans DB RDS tunggal. Proksi tersebut secara otomatis menentukan instans penulis saat ini untuk instans atau klaster DB RDS Multi-AZ.

Koneksi yang dibiarkan terbuka dan disediakan oleh proksi untuk digunakan oleh aplikasi basis data Anda akan membentuk kumpulan koneksi.

Secara default, Proksi RDS dapat menggunakan ulang koneksi setelah setiap transaksi dalam sesi Anda. Penggunaan kembali tingkat transaksi ini disebut multiplexing. Jika Proksi RDS secara temporer menghapus satu koneksi dari kumpulan koneksi untuk menggunakannya kembali, operasi tersebut disebut borrowing koneksi. Jika aman dilakukan, Proksi RDS akan mengembalikan koneksi tersebut ke kumpulan koneksi.

Dalam beberapa kasus, Proksi RDS tidak dapat memastikan keamanan penggunaan ulang sebuah koneksi basis data di luar sesi saat ini. Untuk kasus seperti ini, Proksi RDS akan tetap mempertahankan sesi pada koneksi yang sama hingga sesi berakhir. Perilaku fallback ini disebut pinning.

Proksi memiliki titik akhir default. Anda terhubung ke titik akhir ini saat menggunakan instans DB Amazon RDS. Anda melakukannya alih-alih terhubung ke titik akhir baca/tulis yang terhubung langsung ke klaster . Untuk klaster DB RDS, Anda juga dapat membuat titik akhir baca/tulis dan hanya-baca tambahan. Untuk informasi selengkapnya, lihat Ikhtisar titik akhir proksi.

Misalnya, Anda masih dapat terhubung ke titik akhir klaster untuk koneksi baca/tulis tanpa pengumpulan koneksi. Anda masih dapat terhubung ke titik akhir pembaca untuk koneksi keseimbangan beban hanya-baca. Anda masih dapat terhubung ke titik akhir instans untuk melakukan diagnosis dan memecahkan masalah instans DB tertentu dengan klaster. Jika Anda menggunakan layanan AWS lain seperti AWS Lambda untuk terhubung ke basis data RDS, ubah pengaturan koneksinya untuk menggunakan titik akhir proksi. Misalnya, tentukan titik akhir proksi untuk mengizinkan fungsi Lambda mengakses basis data Anda sekaligus memanfaatkan fungsionalitas Proksi RDS.

Setiap proksi berisi sebuah grup target. Grup target ini berisi instans DB RDS yang menjadi tujuan koneksi proksi. Instans DB RDS yang terkait dengan proksi disebut sebagai target proksi tersebut. Untuk kemudahan, saat Anda membuat proksi melalui konsol, Proksi RDS juga membuat grup target yang sesuai dan mendaftarkan target terkait ini secara otomatis.

Keluarga mesin adalah serangkaian mesin basis data terkait yang menggunakan protokol DB yang sama. Anda dapat memilih keluarga mesin untuk setiap proksi yang Anda buat.

Pengumpulan koneksi

Setiap proksi melakukan pengumpulan koneksi untuk instans penulis dari basis data RDS terkaitnya. Pengumpulan koneksi adalah pengoptimalan yang menurunkan overhead yang terkait dengan pembukaan dan penutupan koneksi dan dengan menjaga banyak koneksi terbuka secara bersamaan. Overhead ini mencakup memori yang diperlukan untuk menangani setiap koneksi baru. Ini juga melibatkan overhead CPU untuk menutup setiap koneksi dan membuka koneksi yang baru. Contohnya meliputi jabat tangan Keamanan Lapisan Pengangkutan/Lapisan Soket Aman (TLS/SSL), autentikasi, kemampuan negosiasi, dan sebagainya. Pengumpulan koneksi menyederhanakan logika aplikasi Anda. Anda tidak perlu menulis kode aplikasi untuk meminimalkan jumlah koneksi terbuka secara bersamaan.

Setiap proksi juga melakukan multipleks koneksi, yang juga dikenal sebagai penggunaan ulang koneksi. Dengan multiplexing, Proksi RDS melakukan semua operasi untuk transaksi menggunakan satu koneksi basis data acuan. RDS kemudian dapat menggunakan koneksi yang berbeda untuk transaksi berikutnya. Anda dapat membuka banyak koneksi ke proksi secara bersamaan, dan proksi akan mempertahankan koneksi terbuka dalam jumlah yang lebih kecil ke instans atau klaster DB. Tindakan ini akan lebih meminimalkan overhead memori untuk koneksi di server basis data. Teknik ini juga mengurangi kemungkinan kesalahan "terlalu banyak koneksi".

Keamanan Proksi RDS

Proksi RDS menggunakan mekanisme keamanan RDS yang sudah ada seperti TLS/SSL dan AWS Identity and Access Management (IAM). Untuk informasi umum tentang fitur keamanan tersebut, lihat Keamanan dalam Amazon RDS. Selain itu, pastikan Anda benar-benar mengetahui bagaimana cara kerja RDS dengan autentikasi, otorisasi, dan bidang keamanan lainnya.

Proksi RDS dapat bertindak sebagai lapisan keamanan tambahan antara aplikasi klien dan basis data acuan. Misalnya, Anda dapat terhubung ke proksi menggunakan TLS 1.2, meskipun instans DB acuan mendukung TLS versi yang lebih lama. Anda dapat terhubung ke proksi menggunakan

peran IAM. Hal ini terjadi bahkan jika proksi terhubung ke basis data menggunakan pengguna native dan metode autentikasi kata sandi. Dengan menggunakan teknik ini, Anda dapat menerapkan persyaratan autentikasi yang kuat untuk aplikasi basis data tanpa sebuah upaya migrasi yang mahal untuk instans DB itu sendiri.

Anda menyimpan kredensial basis data yang digunakan oleh Proksi RDS dalam AWS Secrets Manager. Setiap pengguna basis data untuk sebuah instans DB RDS yang diakses oleh proksi harus memiliki rahasia yang sesuai di Secrets Manager. Anda juga dapat menyiapkan autentikasi IAM untuk pengguna Proksi RDS. Dengan melakukannya, Anda dapat menerapkan autentikasi IAM untuk akses basis data meskipun basis data menggunakan autentikasi kata sandi native. Sebaiknya gunakan fitur keamanan ini, alih-alih menyematkan kredensial basis data dalam kode aplikasi Anda.

Penggunaan TLS/SSL dengan Proksi RDS

Anda dapat terhubung ke Proksi RDS menggunakan protokol TLS/SSL.

> **Note**
>
> Proksi RDS menggunakan sertifikat dari AWS Certificate Manager (ACM). Jika Anda menggunakan Proksi RDS, Anda tidak perlu mengunduh sertifikat Amazon RDS atau memperbarui aplikasi yang menggunakan koneksi Proksi RDS.

Untuk menerapkan TLS untuk semua koneksi antara proksi dan basis data, Anda dapat menentukan pengaturan Wajibkan Keamanan Lapisan Pengangkutan saat membuat atau mengubah sebuah AWS Management Console.

Proksi RDS juga dapat memastikan agar sesi Anda menggunakan TLS/SSL antara klien Anda dan titik akhir Proksi RDS. Untuk membuat Proksi RDS melakukannya, tentukan persyaratan pada sisi klien. Variabel sesi SSL tidak diatur untuk koneksi SSL ke basis data menggunakan Proksi RDS.

- Untuk RDS for MySQL, tentukan persyaratan pada sisi klien dengan parameter `--ssl-mode` saat Anda menjalankan perintah `mysql`.
- Untuk Amazon RDS PostgreSQL, tentukan `sslmode=require` sebagai bagian dari string `conninfo` saat Anda menjalankan perintah `psql`.

Proksi RDS mendukung protokol TLS versi 1.0, 1.1, dan 1.2. Anda dapat terhubung ke proksi menggunakan versi TLS yang lebih tinggi daripada yang digunakan dalam basis data acuan.

Secara default, program klien membangun koneksi terenkripsi dengan Proksi RDS, dengan ketersediaan kontrol lebih lanjut melalui opsi `--ssl-mode`. Dari sisi klien, Proksi RDS mendukung semua mode SSL.

Untuk klien, mode SSL adalah sebagai berikut:

DIUTAMAKAN

SSL adalah pilihan pertama, tetapi tidak diharuskan.

NONAKTIFKAN

Tidak ada SSL yang diperbolehkan.

DIPERLUKAN

Menerapkan SSL.

VERIFY_CA

Menerapkan SSL dan memverifikasi otoritas sertifikat (CA).

VERIFY_IDENTITY

Menerapkan SSL dan memverifikasi CA dan nama host CA.

Saat menggunakan sebuah klien dengan `--ssl-mode` `VERIFY_CA` atau `VERIFY_IDENTITY`, tentukan opsi `--ssl-ca` yang menunjuk ke CA dalam format `.pem`. Untuk file `.pem` yang akan digunakan, unduh semua CA PEM root dari Layanan Kepercayaan Amazon dan tempatkan ke dalam satu file `.pem`.

Proksi RDS menggunakan sertifikat wildcard, yang berlaku untuk domain dan subdomainnya. Jika Anda menggunakan klien `mysql` untuk terhubung dengan mode SSL `VERIFY_IDENTITY`, Anda kini harus menggunakan perintah `mysql` yang kompatibel dengan MySQL 8.0.

Failover

Failover adalah fitur ketersediaan tinggi yang menggantikan instans basis data dengan yang lain saat instans asli tidak tersedia. Failover dapat terjadi karena sebuah masalah pada instans basis data. Bisa juga bagian dari prosedur pemeliharaan normal, seperti saat upgrade basis data. Failover berlaku untuk instans DB RDS dengan konfigurasi Multi-AZ.

Terhubung melalui proksi membuat aplikasi Anda lebih tangguh menghadapi failover basis data. Saat instans DB asli tidak tersedia, Proksi RDS akan terhubung ke basis data siaga tanpa kehilangan

koneksi aplikasi yang idle. Hal ini dapat membantu mempercepat dan menyederhanakan proses failover. Dampaknya terhadap aplikasi juga lebih kecil dibandingkan masalah reboot atau basis data pada umumnya.

Tanpa Proksi RDS, failover dapat menyebabkan pemadaman singkat. Selama pemadaman, Anda tidak dapat melakukan operasi tulis pada basis data dengan failover. Koneksi semua basis data yang ada akan terganggu, dan aplikasi Anda harus membuka ulang koneksi tersebut. basis data akan tersedia untuk koneksi dan operasi tulis baru saat instans DB hanya-baca dipromosikan menggantikan yang tidak tersedia.

Selama failover DB, Proksi RDS terus menerima koneksi di alamat IP yang sama dan secara otomatis mengarahkan koneksi ke instans DB primer baru. Klien yang terhubung melalui Proksi RDS tidak rentan terhadap hal berikut:

- Penundaan propagasi Sistem Nama Domain (DNS) saat failover.
- Caching DNS lokal.
- Waktu koneksi habis.
- Ketidakpastian tentang instans DB mana yang merupakan instans penulis saat ini.
- Menunggu respons kueri dari penulis sebelumnya yang menjadi tidak tersedia tanpa menutup koneksi.

Untuk aplikasi yang mempertahankan kumpulan koneksinya sendiri, melewati Proksi RDS berarti sebagian besar koneksi tetap aktif selama failover atau gangguan lainnya. Hanya koneksi yang berada di tengah transaksi atau pernyataan SQL yang dibatalkan. Proksi RDS segera menerima koneksi baru. Saat penulis basis data tidak tersedia, Proksi RDS akan mengantrekan permintaan masuk.

Untuk aplikasi yang tidak mempertahankan kumpulan koneksinya sendiri, Proksi RDS menawarkan tingkat koneksi yang lebih cepat dan lebih banyak koneksi terbuka. Hal ini menyebabkan overhead yang mahal dikarenakan seringnya koneksi ulang dari basis data. Hal ini dilakukan dengan menggunakan kembali koneksi basis data yang dipertahankan dalam kumpulan koneksi Proksi RDS. Pendekatan ini sangatlah penting untuk koneksi TLS, yang memerlukan biaya penyiapan yang sangat besar.

Transaksi

Semua pernyataan dalam satu transaksi selalu menggunakan koneksi basis data acuan yang sama. Koneksi akan dapat digunakan oleh sesi yang berbeda saat transaksi berakhir. Penggunaan transaksi sebagai unit granularitas memiliki konsekuensi sebagai berikut:

- Penggunaan ulang koneksi bisa terjadi setelah setiap pernyataan jika pengaturan `autocommit` RDS for MySQL diaktifkan.
- Sebaliknya, jika pengaturan `autocommit` dinonaktifkan, pernyataan pertama yang Anda terbitkan dalam sesi akan memulai transaksi baru. Misalnya, Anda memasukkan urutan pernyataan `SELECT`, `INSERT`, `UPDATE`, dan bahasa manipulasi data (DML) lainnya. Dalam hal ini, penggunaan kembali koneksi tidak terjadi hingga Anda mengeluarkan `COMMIT`, `ROLLBACK`, atau mengakhiri transaksi.
- Memasukkan pernyataan bahasa definisi data (DDL) akan menyebabkan transaksi berakhir setelah pernyataan tersebut selesai.

Proksi RDS mendeteksi waktu saat transaksi berakhir melalui protokol jaringan yang digunakan oleh aplikasi klien basis data. Deteksi transaksi tidak bergantung pada kata kunci seperti `COMMIT` atau `ROLLBACK` yang muncul dalam teks pernyataan SQL.

Dalam beberapa kasus, Proksi RDS dapat mendeteksi permintaan basis data yang membuatnya tidak praktis untuk memindahkan sesi Anda ke koneksi yang berbeda. Dalam kasus ini, multiplexing dinonaktifkan untuk koneksi tersebut hingga sesi Anda berakhir. Aturan serupa berlaku jika Proksi RDS tidak dapat memastikan apakah multiplexing praktis untuk sesi ini. Operasi ini disebut pinning. Untuk mendeteksi dan meminimalkan pinning, lihat Menghindari penyematan.

Memulai dengan Proksi RDS

Di bagian berikut ini, Anda dapat menemukan cara menyiapkan dan mengelola Proksi RDS. Anda juga dapat menemukan cara mengatur opsi keamanan terkait. Opsi ini mengontrol siapa saja yang dapat mengakses setiap proksi dan cara setiap proksi terhubung ke instans DB.

Topik

- Menyiapkan prasyarat jaringan
- Menyiapkan kredensi database di AWS Secrets Manager
- Menyiapkan AWS Identity and Access Management kebijakan (IAM)
- Membuat Proksi RDS

- Melihat Proksi RDS
- Terhubung ke basis data melalui Proksi RDS

Menyiapkan prasyarat jaringan

Menggunakan Proksi RDS mengharuskan Anda memiliki cloud privat virtual (VPC) umum antara instans DB RDS dan Proksi RDS. VPC ini harus memiliki minimal dua subnet yang berada di Zona Ketersediaan yang berbeda. Akun Anda dapat memiliki subnet ini atau membagikannya dengan akun lain. Untuk informasi tentang berbagi VPC, lihat Bekerja dengan VPC bersama.

Sumber daya aplikasi klien seperti Amazon EC2, Lambda, atau Amazon ECS bisa berada di VPC yang sama dengan proksi. Atau sumber daya ini bisa berada di VPC terpisah dari proksi. Jika Anda berhasil terhubung ke instans DB RDS apa pun, berarti Anda sudah memiliki sumber daya jaringan yang diperlukan.

Topik

- Mendapatkan informasi tentang subnet Anda
- Perencanaan untuk kapasitas alamat IP

Mendapatkan informasi tentang subnet Anda

Untuk membuat proxy, Anda harus menyediakan subnet dan VPC tempat proxy beroperasi di dalamnya. Contoh Linux berikut menunjukkan AWS CLI perintah yang memeriksa VPC dan subnet yang dimiliki oleh Anda. Akun AWS Khususnya, Anda meneruskan ID subnet sebagai parameter ketika Anda membuat proksi menggunakan CLI.

```
aws ec2 describe-vpcs
aws ec2 describe-internet-gateways
aws ec2 describe-subnets --query '*[].[VpcId,SubnetId]' --output text | sort
```

Contoh Linux berikut menunjukkan AWS CLI perintah untuk menentukan ID subnet yang sesuai dengan instance RDS DB cluster DB tertentu. Temukan ID VPC untuk instans DB. Periksa VPC untuk menemukan subnetnya. Contoh Linux berikut menunjukkan caranya.

```
$ #From the DB instance, trace through the DBSubnetGroup and Subnets to find the subnet
 IDs.
```

```
$ aws rds describe-db-instances --db-instance-identifier my_instance_id --query '*[].
[DBSubnetGroup]|[0]|[0]|[Subnets]|[0]|[*].SubnetIdentifier' --output text
```

```
subnet_id_1
subnet_id_2
subnet_id_3
...
```

```
$ #From the DB instance, find the VPC.
$ aws rds describe-db-instances --db-instance-identifier my_instance_id --query '*[].
[DBSubnetGroup]|[0]|[0].VpcId' --output text
```

```
my_vpc_id
```

```
$ aws ec2 describe-subnets --filters Name=vpc-id,Values=my_vpc_id --query '*[].
[SubnetId]' --output text
```

```
subnet_id_1
subnet_id_2
subnet_id_3
subnet_id_4
subnet_id_5
subnet_id_6
```

Perencanaan untuk kapasitas alamat IP

Proksi RDS secara otomatis menyesuaikan kapasitasnya sesuai kebutuhan berdasarkan ukuran dan jumlah instans DB yang didaftarkan di proksi. Operasi tertentu mungkin juga memerlukan kapasitas proksi tambahan seperti menambah ukuran basis data yang terdaftar atau operasi pemeliharaan Proksi RDS internal. Selama operasi ini, proksi Anda mungkin memerlukan lebih banyak alamat IP untuk menyediakan kapasitas tambahan. Alamat tambahan ini memungkinkan proksi Anda diskalakan tanpa memengaruhi beban kerja Anda. Kurangnya alamat IP kosong di subnet Anda mencegah proksi dari menaikkan skala. Hal ini dapat menyebabkan latensi kueri yang lebih tinggi atau kegagalan koneksi klien. RDS memberi tahu Anda melalui peristiwa `RDS-EVENT-0243` ketika tidak ada cukup alamat IP kosong di subnet Anda. Untuk informasi tentang peristiwa ini, lihat Bekerja dengan peristiwa Proksi RDS.

Berikut adalah jumlah minimum alamat IP yang disarankan untuk dibiarkan kosong di subnet Anda untuk proksi Anda berdasarkan ukuran kelas instans DB.

Kelas instans DB	Alamat IP kosong minimum
db.*.xlarge atau lebih kecil	10
db.*.2xlarge	15
db.*.4xlarge	25
db.*.8xlarge	45
db.*.12xlarge	60
db.*.16xlarge	75
db.*.24xlarge	110

Jumlah alamat IP yang direkomendasikan ini adalah perkiraan untuk proksi dengan titik akhir default saja. Proksi dengan titik akhir tambahan atau replika baca mungkin memerlukan lebih banyak alamat IP kosong. Untuk setiap titik akhir tambahan, sebaiknya Anda mencadangkan tiga alamat IP lagi. Untuk setiap replika baca, sebaiknya Anda mencadangkan alamat IP tambahan seperti yang ditentukan dalam tabel berdasarkan ukuran replika baca tersebut.

> **Note**
>
> Proksi RDS tidak mendukung lebih dari 215 alamat IP di satu VPC.

Menyiapkan kredensi database di AWS Secrets Manager

Untuk setiap proksi yang Anda buat, pertama-tama Anda harus menggunakan layanan Secrets Manager untuk menyimpan kumpulan kredensial nama pengguna dan kata sandi. Anda dapat membuat rahasia Secrets Manager terpisah untuk setiap akun pengguna basis data yang terhubung ke instans DB RDS.

Dalam konsol Secrets Manager, Anda dapat membuat rahasia ini dengan nilai untuk kolom `username` dan `password`. Dengan melakukannya, proksi dapat terhubung ke pengguna basis data

yang sesuai di instans DB RDS yang Anda kaitkan dengan proksi. Untuk melakukannya, Anda dapat menggunakan pengaturan Kredensial untuk basis data lain, Kredensial untuk basis data RDS, atau Jenis rahasia lainnya. Masukkan nilai yang sesuai di kolom Nama pengguna dan Kata sandi, dan nilai untuk kolom lain yang wajib diisi. Proksi akan mengabaikan kolom lain seperti Host dan Port jika ada dalam rahasia tersebut. Detail tersebut secara otomatis diisi oleh proksi.

Anda juga dapat memilih Jenis rahasia lainnya. Dalam kasus ini, Anda membuat rahasia dengan kunci bernama `username` dan `password`.

Karena rahasia yang digunakan oleh proksi Anda tidak terikat dengan server basis data tertentu, Anda dapat menggunakan kembali rahasia di beberapa proksi. Untuk melakukannya, gunakan kredensial yang sama di beberapa server basis data. Misalnya, Anda dapat menggunakan kredensial yang sama di seluruh server pengembangan dan pengujian.

Untuk terhubung melalui proksi sebagai pengguna basis data tertentu, pastikan kata sandi yang terkait dengan rahasia cocok dengan kata sandi basis data untuk pengguna tersebut. Jika ada ketidakcocokan, Anda dapat memperbarui rahasia terkait dalam Secrets Manager. Dalam kasus ini, Anda masih dapat terhubung ke akun lain yang memiliki kredensial rahasia dan kata sandi basis data yang cocok.

> **Note**
>
> Untuk RDS untuk SQL Server, RDS Proxy memerlukan rahasia di Secrets Manager yang peka huruf besar/kecil terhadap kode aplikasi terlepas dari pengaturan pemeriksaan instans DB. Misalnya, jika aplikasi Anda dapat menggunakan nama pengguna “Admin” atau “admin”, maka konfigurasikan proxy dengan rahasia untuk “Admin” dan “admin”. RDS Proxy tidak mengakomodasi username case-insensitivity dalam proses otentikasi antara klien dan proxy. Untuk informasi selengkapnya tentang pemeriksaan di SQL Server, lihat dokumentasi Microsoft SQL Server.

Saat Anda membuat proxy melalui AWS CLI atau RDS API, Anda menentukan Amazon Resource Names (ARN) dari rahasia yang sesuai. Anda melakukannya untuk semua akun pengguna DB yang dapat diakses proksi. Dalam AWS Management Console, Anda memilih rahasia dengan nama deskriptif mereka.

Untuk petunjuk tentang cara membuat rahasia di Secrets Manager, lihat halaman Membuat rahasia dalam dokumentasi Secrets Manager. Gunakan salah satu teknik berikut:

- Gunakan Secrets Manager di konsol.
- Untuk menggunakan CLI untuk membuat rahasia Secrets Manager untuk digunakan oleh Proksi RDS, gunakan perintah seperti berikut.

```
aws secretsmanager create-secret
  --name "secret_name"
  --description "secret_description"
  --region region_name
  --secret-string '{"username":"db_user","password":"db_user_password"}'
```

Misalnya, perintah berikut ini membuat rahasia Secrets Manager untuk dua pengguna basis data, yang satu bernama `admin` dan yang lain bernama `app-user`.

```
aws secretsmanager create-secret \
  --name admin_secret_name --description "db admin user" \
  --secret-string '{"username":"admin","password":"choose_your_own_password"}'

aws secretsmanager create-secret \
  --name proxy_secret_name --description "application user" \
  --secret-string '{"username":"app-user","password":"choose_your_own_password"}'
```

Untuk melihat rahasia yang dimiliki oleh AWS akun Anda, gunakan perintah seperti berikut ini.

```
aws secretsmanager list-secrets
```

Saat Anda membuat proksi menggunakan CLI, Anda meneruskan Amazon Resource Name (ARN) dari satu atau beberapa rahasia ke parameter `--auth`. Contoh Linux berikut menunjukkan cara menyiapkan laporan hanya dengan nama dan ARN dari setiap rahasia yang dimiliki oleh akun Anda AWS . Contoh ini menggunakan parameter `--output table` yang tersedia di AWS CLI versi 2. Jika Anda menggunakan AWS CLI versi 1, gunakan `--output text` sebagai gantinya.

```
aws secretsmanager list-secrets --query '*[].[Name,ARN]' --output table
```

Untuk memverifikasi bahwa Anda telah menyimpan kredensial yang benar dan dalam format yang benar dalam rahasia, gunakan perintah seperti berikut. Gantikan nama pendek atau ARN rahasia untuk *`your_secret_name`*.

```
aws secretsmanager get-secret-value --secret-id your_secret_name
```

Output harus mencakup baris yang menampilkan nilai yang dienkode JSON seperti berikut.

```
"SecretString": "{\"username\":\"your_username\",\"password\":\"your_password\"}",
```

Menyiapkan AWS Identity and Access Management kebijakan (IAM)

Setelah membuat rahasia di Secrets Manager, Anda dapat membuat kebijakan IAM yang dapat mengakses rahasia tersebut. Untuk informasi umum tentang cara menggunakan IAM, lihat Manajemen identitas dan akses untuk Amazon RDS.

> **Tip**
>
> Prosedur berikut berlaku jika Anda menggunakan konsol IAM. Jika Anda menggunakan AWS Management Console for RDS, RDS dapat membuat kebijakan IAM untuk Anda secara otomatis. Dalam kasus ini, Anda dapat melewati prosedur berikut.

Untuk membuat kebijakan IAM yang mengakses rahasia Secrets Manager untuk digunakan oleh proksi Anda

1. Masuk ke konsol IAM. Ikuti proses Buat peran, seperti yang dijelaskan dalam Membuat peran IAM, memilih Membuat peran untuk mendelegasikan izin ke layanan. AWS

   Pilih Layanan AWS untuk Jenis entitas tepercaya. Di bagian Kasus penggunaan, pilih RDS dari menu drop-down Kasus penggunaan untuk layanan AWS . Pilih RDS – Tambahkan Peran ke basis data.

2. Untuk peran baru, lakukan langkah Tambahkan kebijakan sebaris. Gunakan prosedur umum yang sama seperti dalam Mengedit kebijakan IAM. Tempelkan JSON berikut ke dalam kotak teks JSON. Ganti ID akun Anda sendiri. Gantikan AWS Wilayah Anda untuk`us-east-2`. Ganti Amazon Resource Name (ARN) dengan rahasia yang Anda buat, lihat Menentukan kunci KMS dalam pernyataan kebijakan IAM. Untuk `kms:Decrypt` tindakan, gantikan ARN dari default AWS KMS key atau kunci KMS Anda sendiri. Mana yang Anda gunakan bergantung pada mana yang Anda gunakan untuk mengenkripsi rahasia Secrets Manager.

   ```
   {
       "Version": "2012-10-17",
       "Statement": [
           {
   ```

```
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "secretsmanager:GetSecretValue",
            "Resource": [
                "arn:aws:secretsmanager:us-east-2:account_id:secret:secret_name_1",
                "arn:aws:secretsmanager:us-east-2:account_id:secret:secret_name_2"
            ]
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": "kms:Decrypt",
            "Resource": "arn:aws:kms:us-east-2:account_id:key/key_id",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "secretsmanager.us-east-2.amazonaws.com"
                }
            }
        }
    ]
}
```

3. Edit kebijakan kepercayaan untuk peran IAM ini. Tempelkan JSON berikut ke dalam kotak teks JSON.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "rds.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Perintah berikut akan melakukan operasi yang sama melalui AWS CLI.

```
PREFIX=my_identifier
```

```
USER_ARN=$(aws sts get-caller-identity --query "Arn" --output text)

aws iam create-role --role-name my_role_name \
  --assume-role-policy-document '{"Version":"2012-10-17","Statement":
[{"Effect":"Allow","Principal":{"Service":
["rds.amazonaws.com"]},"Action":"sts:AssumeRole"}]}'

ROLE_ARN=arn:aws:iam::account_id:role/my_role_name

aws iam put-role-policy --role-name my_role_name \
  --policy-name $PREFIX-secret-reader-policy --policy-document
 '{"Version":"2012-10-17","Statement":
[{"Sid":"getsecretvalue","Effect":"Allow","Action":
["secretsmanager:GetSecretValue","kms:Decrypt"],"Resource":"*"}]}'

aws kms create-key --description "$PREFIX-test-key" --policy '{
  "Id":"$PREFIX-kms-policy",
  "Version":"2012-10-17",
  "Statement":
    [
      {
        "Sid":"Enable IAM User Permissions",
        "Effect":"Allow",
        "Principal":{"AWS":"arn:aws::iam:account_id:root"},
        "Action":"kms:*","Resource":"*"
      },
      {
        "Sid":"Allow access for Key Administrators",
        "Effect":"Allow",
        "Principal":
          {
            "AWS":
              ["$USER_ARN","arn:aws::iam:account_id:role/Admin"]
          },
        "Action":
          [
            "kms:Create*",
            "kms:Describe*",
            "kms:Enable*",
            "kms:List*",
            "kms:Put*",
            "kms:Update*",
            "kms:Revoke*",
            "kms:Disable*",
```

```
            "kms:Get*",
            "kms:Delete*",
            "kms:TagResource",
            "kms:UntagResource",
            "kms:ScheduleKeyDeletion",
            "kms:CancelKeyDeletion"
          ],
        "Resource":"*"
      },
      {
        "Sid":"Allow use of the key",
        "Effect":"Allow",
        "Principal":{"AWS":"$ROLE_ARN"},
        "Action":["kms:Decrypt","kms:DescribeKey"],
        "Resource":"*"
      }
    ]
}'
```

Membuat Proksi RDS

Untuk mengelola koneksi untuk kumpulan instans DB tertentu, Anda dapat membuat proksi. Anda dapat mengaitkan proksi dengan instans DB RDS for MariaDB, RDS for Microsoft SQL Server, RDS for MySQL, atau RDS for PostgreSQL.

AWS Management Console

Untuk membuat proksi

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Proksi.
3. Pilih Buat proksi.
4. Pilih semua pengaturan untuk proksi Anda.

   Untuk Konfigurasi proksi, masukkan informasi untuk opsi berikut:

   - Keluarga mesin. Pengaturan ini menentukan protokol jaringan basis data mana yang dikenali proksi ketika menafsirkan lalu lintas jaringan ke dan dari basis data. Untuk RDS for MariaDB atau RDS for MySQL, pilih MariaDB dan MySQL. Untuk RDS for PostgreSQL, pilih PostgreSQL. Untuk RDS for SQL Server, pilih SQL Server.

- ID proksi. Tentukan nama yang unik dalam ID AWS akun Anda dan AWS Wilayah saat ini.
- Batas waktu koneksi klien idle. Pilih periode waktu saat koneksi klien menjadi idle sebelum proksi menutupnya. Nilai default-nya adalah 1.800 detik (30 menit). Koneksi klien dianggap idle jika aplikasi tidak mengirimkan permintaan baru dalam waktu yang ditentukan setelah permintaan sebelumnya selesai. Koneksi basis data yang mendasarinya akan tetap terbuka dan dikembalikan ke kumpulan koneksi. Oleh karena itu, koneksi dapat digunakan kembali untuk koneksi klien baru.

  Jika Anda ingin proksi secara proaktif menghapus koneksi yang sudah tidak terpakai, kurangi batas waktu koneksi klien idle. Saat beban kerja meningkat, untuk menghemat biaya pembangunan koneksi, tambah batas waktu koneksi klien idle."

Untuk Konfigurasi grup target, masukkan informasi untuk opsi berikut:

- basis data. Pilih satu instans DB RDS yang akan diakses melalui proksi ini. Daftar ini hanya mencakup instans dan klaster DB dengan mesin basis data, versi mesin, dan pengaturan lainnya yang kompatibel. Jika daftar kosong, buat instans atau klaster DB baru yang kompatibel dengan Proksi RDS. Untuk melakukannya, ikuti prosedur dalam Membuat instans DB Amazon RDS. Lalu, coba buat proksi lagi.
- Koneksi maksimum kumpulan koneksi. Tentukan sebuah nilai dari 1 hingga 100. Pengaturan ini merepresentasikan persentase dari nilai `max_connections` yang dapat digunakan oleh Proksi RDS untuk koneksinya. Jika hanya ingin menggunakan satu proksi dengan instans atau klaster DB ini, Anda dapat mengatur nilai ini ke 100. Untuk detail tentang cara Proksi RDS menggunakan pengaturan ini, lihat MaxConnectionsPercent.
- Filter penyematan sesi. (Opsional) Opsi ini memungkinkan Anda untuk memaksa Proksi RDS untuk tidak memberi pin untuk jenis status sesi tertentu yang terdeteksi. Tindakan ini menghindari langkah-langkah keamanan default untuk me-multipleks koneksi basis data di seluruh koneksi klien. Saat ini, pengaturan tidak didukung untuk PostgreSQL. Satu-satunya pilihan adalah `EXCLUDE_VARIABLE_SETS`.

  Mengaktifkan pengaturan ini dapat menyebabkan variabel sesi dari satu koneksi memengaruhi koneksi lain. Hal ini dapat menyebabkan kesalahan atau masalah ketepatan jika kueri Anda bergantung pada nilai variabel sesi yang ditetapkan di luar transaksi saat ini. Pertimbangkan untuk menggunakan opsi ini setelah memastikan bahwa aplikasi Anda sudah bisa berbagi koneksi basis data dengan aman di seluruh koneksi klien.

  Pola berikut bisa dianggap aman:

  - Pernyataan SET di mana tidak ada perubahan pada nilai variabel sesi efektif, yaitu tidak ada perubahan pada variabel sesi.
  - Anda mengubah nilai variabel sesi dan mengeksekusi pernyataan dalam transaksi yang sama.

  Untuk informasi selengkapnya, lihat Menghindari penyematan.

- Batas waktu peminjaman koneksi. Dalam beberapa kasus, mungkin Anda berharap proksi tersebut sekali-sekali menggunakan semua koneksi basis data yang tersedia. Dalam kasus seperti itu, Anda dapat menentukan durasi proksi harus menunggu koneksi basis data menjadi tersedia sebelum menampilkan kesalahan batas waktu. Anda dapat menentukan periode hingga maksimum lima menit. Pengaturan ini hanya berlaku jika proksi memiliki koneksi terbuka maksimum dan semua koneksi sudah digunakan.
- Kueri inisialisasi. (Opsional) Anda dapat menentukan satu atau beberapa pernyataan SQL agar proksi berjalan saat membuka setiap koneksi basis data baru. Pengaturan ini biasanya digunakan dengan pernyataan SET untuk memastikan bahwa setiap koneksi memiliki pengaturan yang identik seperti zona waktu dan kumpulan karakter. Untuk beberapa pernyataan, gunakan titik koma sebagai pemisah. Anda juga dapat menyertakan beberapa variabel dalam satu pernyataan SET, seperti `SET x=1, y=2`.

Untuk Autentikasi, masukkan informasi untuk opsi berikut:

- Peran IAM. Pilih peran IAM yang memiliki izin untuk mengakses rahasia Secrets Manager yang Anda pilih sebelumnya. Atau, Anda dapat membuat peran IAM baru dari AWS Management Console.
- Rahasia Secrets Manager. Pilih setidaknya satu rahasia Secrets Manager terpisah yang berisi kredensial pengguna basis data yang memungkinkan proksi mengakses instans DB RDS.
- Jenis autentikasi klien. Pilih jenis autentikasi yang digunakan proksi untuk koneksi dari klien. Pilihan Anda berlaku untuk semua rahasia Secrets Manager yang Anda kaitkan dengan proksi ini. Jika Anda perlu menentukan jenis otentikasi klien yang berbeda untuk setiap rahasia, maka buat proxy Anda dengan menggunakan AWS CLI atau API sebagai gantinya.
- Autentikasi IAM. Pilih apakah akan mewajibkan , mengizinkan, atau melarang autentikasi IAM untuk koneksi ke proksi Anda. Opsi izinkan hanya berlaku untuk proksi untuk RDS for SQL Server. Pilihan Anda berlaku untuk semua rahasia Secrets Manager yang Anda kaitkan dengan proksi ini. Jika Anda perlu menentukan otentikasi IAM yang berbeda untuk setiap rahasia, buat proxy Anda dengan menggunakan AWS CLI atau API sebagai gantinya.

Untuk Konektivitas, masukkan informasi untuk opsi berikut:

- Wajibkan Keamanan Lapisan Pengangkutan. Pilih pengaturan ini jika Anda ingin proksi menerapkan TLS/SSL untuk semua koneksi klien. Untuk koneksi terenkripsi atau tidak terenkripsi ke sebuah proksi, proksi menggunakan pengaturan enkripsi yang sama saat membuat koneksi ke basis data acuan.
- Subnet. Bidang ini telah diisi sebelumnya dengan semua subnet yang terkait dengan VPC Anda. Anda dapat menghapus subnet apa pun yang tidak diperlukan oleh proksi ini. Anda harus membiarkan setidaknya dua subnet.

Masukkan konfigurasi konektivitas tambahan:

- Grup keamanan VPC. Pilih grup keamanan VPC yang sudah ada. Atau, Anda dapat membuat grup keamanan baru dari AWS Management Console. Anda harus mengonfigurasi Aturan masuk untuk mengizinkan aplikasi Anda mengakses proksi. Anda juga harus mengonfigurasi Aturan keluar untuk mengizinkan lalu lintas dari target DB Anda.

  > **Note**
  >
  > Grup keamanan ini harus mengizinkan koneksi dari proksi ke basis data. Grup keamanan yang sama digunakan sebagai jalur masuk dari aplikasi ke proksi, dan jalur keluar dari proksi ke basis data. Misalnya, anggap saja Anda menggunakan grup keamanan yang sama untuk basis data dan proksi Anda. Dalam kasus ini, pastikan sumber daya dalam grup keamanan tersebut dapat berkomunikasi dengan sumber daya lain dalam grup keamanan yang sama.
  > Saat menggunakan VPC bersama, Anda tidak dapat menggunakan grup keamanan default untuk VPC, atau grup keamanan milik akun lain. Pilih grup keamanan milik akun Anda. Jika belum ada, buat satu. Untuk informasi selengkapnya tentang batasan ini, lihat Bekerja dengan VPC bersama.

  RDS mendeploy proksi di beberapa Zona Ketersediaan untuk memastikan ketersediaan yang tinggi. Untuk mengaktifkan komunikasi lintas-AZ untuk proksi semacam ini, daftar kontrol akses (ACL) untuk subnet proksi Anda harus mengizinkan jalan keluar khusus port mesin dan semua port untuk masuk. Untuk informasi selengkapnya tentang ACL jaringan, lihat Mengontrol lalu lintas ke subnet menggunakan ACL jaringan. Jika ACL jaringan untuk proksi

dan target Anda identik, Anda harus menambahkan aturan masuknya protokol TCP tempat Sumber diatur ke CIDR VPC. Anda juga harus menambahkan aturan keluar protokol TCP khusus port mesin tempat Sumber diatur ke CIDR VPC.

(Opsional) Masukkan konfigurasi lanjutan:

- Aktifkan pengelogan yang disempurnakan. Anda dapat mengaktifkan pengaturan ini untuk memecahkan masalah kompatibilitas proksi atau masalah performa.

  Jika pengaturan ini diaktifkan, Proksi RDS akan menyertakan informasi mendetail tentang pernyataan SQL dalam log-nya. Informasi ini membantu Anda untuk men-debug masalah yang melibatkan perilaku SQL atau performa serta skalabilitas koneksi proksi. Informasi debug mencakup teks dari pernyataan SQL yang Anda kirimkan melalui proksi. Oleh karena, hanya aktifkan pengaturan ini untuk debugging atau jika Anda memiliki prosedur keamanan untuk melindungi informasi sensitif apa pun yang muncul dalam log.

  Untuk meminimalkan overhead yang terkait dengan proksi Anda, Proksi RDS secara otomatis menonaktifkan pengaturan ini 24 jam setelah Anda mengaktifkannya. Aktifkan sementara untuk memecahkan masalah tertentu.

5. Pilih Buat Proksi.

AWS CLI

Untuk membuat proxy dengan menggunakan AWS CLI, panggil create-db-proxyperintah dengan parameter yang diperlukan berikut:

- `--db-proxy-name`
- `--engine-family`
- `--role-arn`
- `--auth`
- `--vpc-subnet-ids`

Nilai `--engine-family` ini bersifat peka huruf besar-kecil.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-proxy \
    --db-proxy-name proxy_name \
    --engine-family { MYSQL | POSTGRESQL | SQLSERVER } \
    --auth ProxyAuthenticationConfig_JSON_string \
    --role-arn iam_role \
    --vpc-subnet-ids space_separated_list \
    [--vpc-security-group-ids space_separated_list] \
    [--require-tls | --no-require-tls] \
    [--idle-client-timeout value] \
    [--debug-logging | --no-debug-logging] \
    [--tags comma_separated_list]
```

Untuk Windows:

```
aws rds create-db-proxy ^
    --db-proxy-name proxy_name ^
    --engine-family { MYSQL | POSTGRESQL | SQLSERVER } ^
    --auth ProxyAuthenticationConfig_JSON_string ^
    --role-arn iam_role ^
    --vpc-subnet-ids space_separated_list ^
    [--vpc-security-group-ids space_separated_list] ^
    [--require-tls | --no-require-tls] ^
    [--idle-client-timeout value] ^
    [--debug-logging | --no-debug-logging] ^
    [--tags comma_separated_list]
```

Berikut ini adalah contoh nilai JSON untuk opsi `--auth`. Contoh ini menerapkan jenis autentikasi klien yang berbeda untuk setiap rahasia.

```
[
  {
    "Description": "proxy description 1",
    "AuthScheme": "SECRETS",
    "SecretArn": "arn:aws:secretsmanager:us-
west-2:123456789123:secret/1234abcd-12ab-34cd-56ef-1234567890ab",
    "IAMAuth": "DISABLED",
    "ClientPasswordAuthType": "POSTGRES_SCRAM_SHA_256"
  },

  {
    "Description": "proxy description 2",
    "AuthScheme": "SECRETS",
```

```
    "SecretArn": "arn:aws:secretsmanager:us-
west-2:111122223333:seret/1234abcd-12ab-34cd-56ef-1234567890cd",
    "IAMAuth": "DISABLED",
    "ClientPasswordAuthType": "POSTGRES_MD5"

  },

  {
    "Description": "proxy description 3",
    "AuthScheme": "SECRETS",
    "SecretArn": "arn:aws:secretsmanager:us-
west-2:111122221111:secret/1234abcd-12ab-34cd-56ef-1234567890ef",
    "IAMAuth": "REQUIRED"
  }

]
```

Tip

Jika Anda belum tahu ID subnet yang akan digunakan untuk parameter `--vpc-subnet-ids`, lihat Menyiapkan prasyarat jaringan untuk contoh tentang cara menemukannya.

Note

Grup keamanan ini harus mengizinkan akses ke basis data yang terhubung ke proksi. Grup keamanan yang sama digunakan sebagai jalur masuk dari aplikasi ke proksi, dan jalur keluar dari proksi ke basis data. Misalnya, anggap saja Anda menggunakan grup keamanan yang sama untuk basis data dan proksi Anda. Dalam kasus ini, pastikan sumber daya dalam grup keamanan tersebut dapat berkomunikasi dengan sumber daya lain dalam grup keamanan yang sama.

Saat menggunakan VPC bersama, Anda tidak dapat menggunakan grup keamanan default untuk VPC, atau grup keamanan milik akun lain. Pilih grup keamanan milik akun Anda. Jika belum ada, buat satu. Untuk informasi selengkapnya tentang batasan ini, lihat Bekerja dengan VPC bersama.

Untuk membuat asosiasi yang tepat untuk proxy, Anda juga menggunakan register-db-proxy-targetsperintah. Tentukan nama grup target `default`. Proksi RDS secara otomatis membuat grup target dengan nama ini saat Anda membuat setiap proksi.

```
aws rds register-db-proxy-targets
    --db-proxy-name value
    [--target-group-name target_group_name]
    [--db-instance-identifiers space_separated_list]  # rds db instances, or
    [--db-cluster-identifiers cluster_id]        # rds db cluster (all instances)
```

API RDS

Untuk membuat proksi RDS, panggil operasi API Amazon RDS CreateDBProxy. Anda melewati parameter dengan struktur AuthConfigdata.

Proksi RDS secara otomatis membuat grup target bernama `default` saat Anda membuat setiap proksi. Anda mengaitkan instance RDS DB cluster dengan grup target dengan memanggil fungsi registerDB. ProxyTargets

Melihat Proksi RDS

Setelah membuat satu atau beberapa proksi RDS, Anda dapat melihat semuanya. Dengan begitu, Anda dapat memeriksa detail konfigurasinya dan memilih mana yang akan dimodifikasi, dihapus, dan sebagainya.

Agar aplikasi basis data dapat menggunakan proksi, Anda harus menyediakan titik akhir proksi dalam string koneksi.

AWS Management Console

Untuk melihat proksi

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di sudut kanan atas AWS Management Console, pilih AWS Wilayah tempat Anda membuat Proxy RDS.
3. Di panel navigasi, pilih Proksi.
4. Pilih nama proksi RDS untuk menampilkan detailnya.
5. Pada halaman detail, bagian Grup target menunjukkan bagaimana proksi dikaitkan dengan instans DB RDS. Anda dapat mengikuti tautan ke halaman grup target default untuk melihat

detail selengkapnya tentang pengaitan antara proksi dan basis data. Halaman ini adalah tempat Anda melihat pengaturan yang Anda tentukan saat membuat proksi. Ini termasuk persentase koneksi maksimum, batas waktu peminjaman koneksi, keluarga mesin, dan filter penyematan sesi.

CLI

Untuk melihat proxy Anda menggunakan CLI, gunakan perintah. describe-db-proxies Secara default, ini menampilkan semua proxy yang dimiliki oleh akun Anda AWS . Untuk melihat detail dari satu proksi, masukkan namanya dengan parameter `--db-proxy-name`.

```
aws rds describe-db-proxies [--db-proxy-name proxy_name]
```

Untuk melihat informasi lain yang terkait dengan proksi, gunakan perintah berikut.

```
aws rds describe-db-proxy-target-groups  --db-proxy-name proxy_name

aws rds describe-db-proxy-targets --db-proxy-name proxy_name
```

Gunakan urutan perintah berikut untuk melihat detail selengkapnya tentang hal-hal yang terkait dengan proksi:

1. Untuk mendapatkan daftar proxy, jalankan. describe-db-proxies
2. Untuk menampilkan parameter koneksi seperti persentase maksimum koneksi yang dapat digunakan proxy, jalankan describe-db-proxy-target-groups`--db-proxy-name`. Gunakan nama proksi sebagai nilai parameter.
3. Untuk melihat detail cluster yang terkait dengan grup target yang dikembalikan, jalankan. describe-db-proxy-targets

API RDS

Untuk melihat proksi Anda menggunakan API RDS, gunakan operasi DescribeDBProxies. Operasi ini akan menampilkan nilai dari jenis data DBProxy.

Untuk melihat detail pengaturan koneksi untuk proxy, gunakan pengidentifikasi proxy dari nilai pengembalian ini dengan operasi ProxyTargetGroupsDescribeDB. Ia mengembalikan nilai-nilai dari tipe ProxyTargetGroup data DB.

Untuk melihat instance RDS atau cluster Aurora DB yang terkait dengan proxy, gunakan operasi DescribeDB. ProxyTargets Ia mengembalikan nilai-nilai dari tipe ProxyTarget data DB.

Terhubung ke basis data melalui Proksi RDS

Cara terhubung ke instans DB RDS melalui proksi atau dengan menghubungkan ke basis data umumnya sama. Untuk informasi selengkapnya, lihat Ikhtisar titik akhir proksi.

Topik

- Terhubung ke sebuah proksi menggunakan autentikasi native
- Terhubung ke sebuah proksi menggunakan autentikasi IAM
- Pertimbangan untuk terhubung ke proksi dengan Microsoft SQL Server
- Pertimbangan untuk terhubung ke sebuah proksi dengan PostgreSQL

Terhubung ke sebuah proksi menggunakan autentikasi native

Gunakan langkah berikut untuk terhubung ke proksi menggunakan autentikasi native:

1. Temukan titik akhir proksi. Di AWS Management Console, Anda dapat menemukan titik akhir pada halaman detail untuk proxy yang sesuai. Dengan itu AWS CLI, Anda dapat menggunakan describe-db-proxiesperintah. Contoh berikut menunjukkan caranya.

```
# Add --output text to get output as a simple tab-separated list.
$ aws rds describe-db-proxies --query '*[*].
{DBProxyName:DBProxyName,Endpoint:Endpoint}'
[
    [
        {
            "Endpoint": "the-proxy.proxy-demo.us-east-1.rds.amazonaws.com",
            "DBProxyName": "the-proxy"
        },
        {
            "Endpoint": "the-proxy-other-secret.proxy-demo.us-
east-1.rds.amazonaws.com",
            "DBProxyName": "the-proxy-other-secret"
        },
        {
            "Endpoint": "the-proxy-rds-secret.proxy-demo.us-
east-1.rds.amazonaws.com",
```

```
            "DBProxyName": "the-proxy-rds-secret"
        },
        {
            "Endpoint": "the-proxy-t3.proxy-demo.us-east-1.rds.amazonaws.com",
            "DBProxyName": "the-proxy-t3"
        }
    ]
]
```

2. Tentukan titik akhir sebagai parameter host dalam string koneksi untuk aplikasi klien Anda. Misalnya, tentukan titik akhir proksi sebagai nilai untuk opsi `mysql -h` atau opsi `psql -h`.
3. Masukkan nama dan kata sandi pengguna basis data yang sama seperti biasanya.

Terhubung ke sebuah proksi menggunakan autentikasi IAM

Saat Anda menggunakan autentikasi IAM dengan Proksi RDS, siapkan pengguna basis data Anda untuk melakukan autentikasi dengan nama dan kata sandi pengguna reguler. Autentikasi IAM berlaku untuk Proksi RDS yang mengambil kredensial nama dan kata sandi pengguna dari Secrets Manager. Koneksi dari Proksi RDS ke basis data acuan tidak melewati IAM.

Untuk terhubung ke Proksi RDS menggunakan autentikasi IAM, gunakan prosedur koneksi umum yang sama seperti autentikasi IAM dengan instans DB RDS. Untuk informasi umum tentang cara menggunakan IAM, lihat Keamanan dalam Amazon RDS.

Perbedaan utama dalam penggunaan IAM untuk Proksi RDS meliputi:

- Anda tidak dapat mengonfigurasi setiap pengguna basis data dengan plugin otorisasi. Pengguna basis data masih memiliki nama dan kata sandi pengguna reguler dalam basis data. Anda dapat menyiapkan rahasia Secrets Manager yang berisi nama dan kata sandi pengguna ini, dan mengotorisasi Proksi RDS untuk mengambil kredensial dari Secrets Manager.

  Autentikasi IAM berlaku untuk koneksi antara program klien Anda dan proksi. Proksi kemudian melakukan autentikasi ke basis data menggunakan kredensial nama dan kata sandi pengguna yang diambil dari Secrets Manager.
- Anda menentukan titik akhir proksi, bukan instans, klaster, atau titik akhir pembaca. Untuk detail tentang titik akhir proksi, lihat Menghubungkan ke instans DB menggunakan autentikasi IAM.
- Dalam kasus autentikasi IAM basis data langsung, Anda secara selektif memilih pengguna basis data dan mengonfigurasinya untuk diidentifikasi dengan plugin autentikasi khusus. Anda kemudian dapat terhubung ke pengguna tersebut menggunakan autentikasi IAM.

Dalam kasus penggunaan proksi, Anda memberi proksi Rahasia yang berisi nama pengguna dan kata sandi pengguna tertentu (autentikasi native). Anda kemudian terhubung ke proksi menggunakan autentikasi IAM. Di sini, Anda melakukannya dengan membuat token autentikasi dengan titik akhir proksi, bukan titik akhir basis data. Anda juga menggunakan nama pengguna yang cocok dengan salah satu nama pengguna untuk rahasia yang Anda berikan.

- Pastikan Anda menggunakan Keamanan Lapisan Pengangkutan (TLS)/Lapisan Soket Aman (SSL) saat terhubung ke sebuah proksi menggunakan autentikasi IAM.

Anda dapat memberi pengguna tertentu akses ke proksi dengan mengubah kebijakan IAM. Berikut contohnya.

```
"Resource": "arn:aws:rds-db:us-east-2:1234567890:dbuser:prx-ABCDEFGHIJKL01234/db_user"
```

Pertimbangan untuk terhubung ke proksi dengan Microsoft SQL Server

Untuk terhubung ke proksi menggunakan autentikasi IAM, Anda tidak menggunakan kolom kata sandi. Sebagai gantinya, Anda memasukkan properti token yang sesuai untuk setiap jenis driver basis data di kolom token. Misalnya, gunakan properti `accessToken` untuk JDBC, atau properti `sql_copt_ss_access_token` untuk ODBC. Atau gunakan `AccessToken` properti untuk SqlClient driver.NET. Anda tidak dapat menggunakan autentikasi IAM dengan klien yang tidak mendukung properti token.

Dalam beberapa kondisi, proksi tidak dapat berbagi koneksi basis data dan sebagai gantinya menyematkan koneksi dari aplikasi klien Anda ke proksi ke koneksi basis data khusus. Untuk informasi selengkapnya tentang cara kondisi ini, lihat Menghindari penyematan.

Pertimbangan untuk terhubung ke sebuah proksi dengan PostgreSQL

Untuk PostgreSQL, saat klien memulai koneksi ke basis data PostgreSQL, pesan startup akan dikirimkan. Pesan ini berisi pasangan nama parameter dan string nilai. Untuk detailnya, lihat `StartupMessage` dalam Format pesan PostgreSQL dalam dokumentasi PostgreSQL.

Saat terhubung melalui proksi RDS, pesan startup bisa menyertakan parameter yang dikenal saat ini sebagai berikut:

- `user`
- `database`

- `replication`

Pesan startup juga bisa menyertakan parameter runtime tambahan berikut:

- `application_name`
- `client_encoding`
- `DateStyle`
- `TimeZone`
- `extra_float_digits`

Untuk informasi selengkapnya tentang pesan PostgreSQL, lihat Protokol Frontend/Backend dalam dokumentasi PostgreSQL.

Untuk PostgreSQL, jika Anda menggunakan JDBC, sebaiknya lakukan tindakan berikut untuk menghindari penyematan:

- Atur parameter koneksi JDBC `assumeMinServerVersion` ke setidaknya `9.0` untuk menghindari penyematan. Tindakan ini dapat mencegah driver JDBC melakukan perjalanan roundtrip ekstra selama startup koneksi saat menjalankan `SET extra_float_digits = 3`.
- Atur parameter koneksi JDBC `ApplicationName` ke *`any/your-application-name`* untuk menghindari penyematan. Tindakan ini dapat mencegah driver JDBC melakukan roundtrip ekstra selama startup koneksi saat menjalankan `SET application_name = "PostgreSQL JDBC Driver"`. Perhatikan bahwa parameter JDBC adalah `ApplicationName`, tetapi parameter PostgreSQL `StartupMessage` adalah `application_name`.

Untuk informasi selengkapnya, lihat Menghindari penyematan. Untuk informasi selengkapnya tentang cara terhubung menggunakan JDBC, lihat Terhubung ke basis data dalam dokumentasi PostgreSQL.

Mengelola Proksi RDS

Bagian ini berisi informasi tentang cara mengelola operasi dan konfigurasi Proksi RDS. Prosedur ini membantu aplikasi Anda memaksimalkan koneksi basis data dan mencapai penggunaan ulang koneksi maksimum. Semakin banyak yang dapat Anda manfaatkan dari penggunaan ulang koneksi, semakin banyak overhead CPU dan memori yang bisa dihemat. Pada akhirnya, tindakan ini dapat mengurangi latensi untuk aplikasi Anda dan memungkinkan basis data untuk mendedikasikan lebih banyak sumber dayanya untuk memproses permintaan aplikasi.

Topik

Mengubah Proksi RDS

Anda dapat mengubah pengaturan spesifik yang terkait dengan proksi setelah Anda membuat proksi. Caranya adalah dengan mengubah proksi itu sendiri, grup target terkaitnya, atau keduanya. Setiap proksi memiliki satu grup target terkait.

AWS Management Console

> **Important**
>
> Nilai dalam kolom Jenis autentikasi klien dan Autentikasi IAM berlaku untuk semua rahasia Secrets Manager yang terkait dengan proksi ini. Untuk menentukan nilai yang berbeda untuk setiap rahasia, ubah proxy Anda dengan menggunakan AWS CLI atau API sebagai gantinya.

Untuk mengubah pengaturan proksi

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Proksi.
3. Dalam daftar proksi, pilih proksi yang pengaturannya ingin diubah atau kunjungi halaman detailnya.
4. Untuk Tindakan, pilih Ubah.
5. Masukkan atau pilih properti yang akan diubah. Anda dapat mengubah opsi berikut:
    - ID proksi – Mengganti nama proksi dengan memasukkan ID baru.

- Batas waktu koneksi klien idle – Masukkan periode waktu untuk batas waktu koneksi klien idle.
- Peran IAM – Mengubah peran IAM yang digunakan untuk mengambil rahasia dari Secrets Manager.
- Rahasia Secrets Manager – Menambahkan atau membuang rahasia Secrets Manager. Rahasia ini sesuai dengan nama dan kata sandi pengguna basis data.
- Jenis autentikasi klien – (PostgreSQL saja) Mengubah jenis autentikasi untuk koneksi klien ke proksi.
- Autentikasi IAM – Mewajibkan atau melarang autentikasi IAM untuk koneksi ke proksi.
- Wajibkan Keamanan Lapisan Pengangkutan – Mengaktifkan atau menonaktifkan persyaratan untuk Keamanan Lapisan Pengangkutan (TLS).
- Grup keamanan VPC – Menambahkan atau menghapus grup keamanan VPC yang akan digunakan proksi.
- Aktifkan pencatatan log yang disempurnakan – Mengaktifkan atau menonaktifkan pencatatan log yang disempurnakan.

6. Pilih Ubah.

Jika pengaturan yang ingin diubah tidak tercantum, gunakan prosedur berikut untuk memperbarui grup target untuk proksi. Grup target yang terkait dengan proksi mengontrol pengaturan yang terkait dengan koneksi basis data fisik. Setiap proksi memiliki satu grup target terkait bernama `default`, yang dibuat secara otomatis dengan proksi.

Anda hanya dapat mengubah grup target dari halaman detail proksi, bukan dari daftar pada halaman Proksi.

Untuk mengubah pengaturan grup target proksi

1. Pada halaman Proksi, buka halaman detail proksi.
2. Untuk Grup target, pilih tautan `default`. Saat ini, semua proksi memiliki satu grup target bernama `default`.
3. Pada halaman detail grup target default, pilih Ubah.
4. Pilih pengaturan baru untuk properti yang dapat diubah:

   - Basis data — Pilih instans atau klaster DB RDS.
   - Koneksi maksimum kumpulan koneksi – Sesuaikan persentase maksimum koneksi yang tersedia yang dapat digunakan proksi.

- Filter penyematan sesi – (Opsional) Pilih filter penyematan sesi. Tindakan ini menghindari langkah-langkah keamanan default untuk me-multipleks koneksi basis data di seluruh koneksi klien. Saat ini, pengaturan tidak didukung untuk PostgreSQL. Satu-satunya pilihan adalah `EXCLUDE_VARIABLE_SETS`.

  Mengaktifkan pengaturan ini dapat menyebabkan variabel sesi dari satu koneksi memengaruhi koneksi lain. Hal ini dapat menyebabkan kesalahan atau masalah ketepatan jika kueri Anda bergantung pada nilai variabel sesi yang ditetapkan di luar transaksi saat ini. Pertimbangkan untuk menggunakan opsi ini setelah memastikan bahwa aplikasi Anda sudah bisa berbagi koneksi basis data dengan aman di seluruh koneksi klien.

  Pola berikut bisa dianggap aman:

  - Pernyataan `SET` di mana tidak ada perubahan pada nilai variabel sesi efektif, yaitu tidak ada perubahan pada variabel sesi.
  - Anda mengubah nilai variabel sesi dan mengeksekusi pernyataan dalam transaksi yang sama.

  Untuk informasi selengkapnya, lihat Menghindari penyematan.

- Batas waktu peminjaman koneksi – Sesuaikan interval batas waktu peminjaman koneksi. Pengaturan ini berlaku saat jumlah maksimum koneksi sudah digunakan semua untuk proksi. Pengaturan ini menentukan seberapa lama proksi harus menunggu koneksi menjadi tersedia sebelum menampilkan sebuah kesalahan batas waktu.
- Kueri inisialisasi – (Opsional) Tambahkan kueri inisialisasi, atau ubah kueri inisialisasi ini. Anda dapat menentukan satu atau beberapa pernyataan SQL untuk proksi yang akan dijalankan saat membuka setiap koneksi basis data baru. Pengaturan ini biasanya digunakan dengan pernyataan `SET` untuk memastikan bahwa setiap koneksi memiliki pengaturan yang identik seperti zona waktu dan kumpulan karakter. Untuk beberapa pernyataan, gunakan titik koma sebagai pemisah. Anda juga dapat menyertakan beberapa variabel dalam satu pernyataan `SET`, seperti `SET x=1, y=2`.

Anda tidak dapat mengubah properti tertentu, seperti ID grup target dan mesin basis data.

5. Pilih Ubah grup target.

AWS CLI

Untuk memodifikasi proxy menggunakan AWS CLI, gunakan perintah modify-db-proxy, modify-db-proxy-target-group deregister-db-proxy-targets, dan register-db-proxy-targets.

Dengan perintah `modify-db-proxy`, Anda dapat mengubah properti seperti berikut:

- Kumpulan rahasia Secrets Manager yang digunakan proksi.
- Apakah TLS diperlukan.
- Batas waktu klien idle.
- Apakah harus mencatat informasi tambahan dari pernyataan SQL untuk debugging.
- Peran IAM yang digunakan untuk mengambil rahasia Secrets Manager.
- Grup keamanan yang digunakan proksi.

Contoh berikut menunjukkan cara mengganti nama proksi yang sudah ada.

```
aws rds modify-db-proxy --db-proxy-name the-proxy --new-db-proxy-name the_new_name
```

Untuk mengubah pengaturan terkait koneksi atau mengganti nama grup target, gunakan perintah `modify-db-proxy-target-group`. Saat ini, semua proksi memiliki satu grup target bernama `default`. Saat bekerja dengan grup target ini, Anda menentukan nama proksi dan `default` untuk nama grup target.

Contoh berikut ini menunjukkan cara memeriksa pengaturan `MaxIdleConnectionsPercent` untuk proksi terlebih dahulu dan kemudian mengubahnya menggunakan grup target.

```
aws rds describe-db-proxy-target-groups --db-proxy-name the-proxy

{
    "TargetGroups": [
        {
            "Status": "available",
            "UpdatedDate": "2019-11-30T16:49:30.342Z",
            "ConnectionPoolConfig": {
                "MaxIdleConnectionsPercent": 50,
                "ConnectionBorrowTimeout": 120,
                "MaxConnectionsPercent": 100,
                "SessionPinningFilters": []
            },
            "TargetGroupName": "default",
```

```
            "CreatedDate": "2019-11-30T16:49:27.940Z",
            "DBProxyName": "the-proxy",
            "IsDefault": true
        }
    ]
}

aws rds modify-db-proxy-target-group --db-proxy-name the-proxy --target-group-name
 default --connection-pool-config '
{ "MaxIdleConnectionsPercent": 75 }'

{
    "DBProxyTargetGroup": {
        "Status": "available",
        "UpdatedDate": "2019-12-02T04:09:50.420Z",
        "ConnectionPoolConfig": {
            "MaxIdleConnectionsPercent": 75,
            "ConnectionBorrowTimeout": 120,
            "MaxConnectionsPercent": 100,
            "SessionPinningFilters": []
        },
        "TargetGroupName": "default",
        "CreatedDate": "2019-11-30T16:49:27.940Z",
        "DBProxyName": "the-proxy",
        "IsDefault": true
    }
}
```

Dengan perintah `deregister-db-proxy-targets` dan `register-db-proxy-targets`, Anda dapat mengubah instans DB RDS mana yang dikaitkan dengan proksi melalui grup targetnya. Saat ini, setiap proksi dapat terhubung ke satu instans DB RDS. Grup target melacak detail koneksi untuk semua instans DB RDS dalam sebuah konfigurasi Multi-AZ.

Contoh berikut dimulai dengan proksi yang dikaitkan dengan klaster Aurora MySQL bernama `cluster-56-2020-02-25-1399`. Contoh ini menunjukkan cara mengubah proksi sehingga dapat terhubung ke klaster lainnya yang bernama `provisioned-cluster`.

Saat menggunakan instans DB RDS, Anda dapat menentukan opsi `--db-instance-identifier`.

Contoh berikut mengubah proksi Aurora MySQL. Proksi Aurora PostgreSQL memiliki port 5432.

```
aws rds describe-db-proxy-targets --db-proxy-name the-proxy
```

```
{
    "Targets": [
        {
            "Endpoint": "instance-9814.demo.us-east-1.rds.amazonaws.com",
            "Type": "RDS_INSTANCE",
            "Port": 3306,
            "RdsResourceId": "instance-9814"
        },
        {
            "Endpoint": "instance-8898.demo.us-east-1.rds.amazonaws.com",
            "Type": "RDS_INSTANCE",
            "Port": 3306,
            "RdsResourceId": "instance-8898"
        },
        {
            "Endpoint": "instance-1018.demo.us-east-1.rds.amazonaws.com",
            "Type": "RDS_INSTANCE",
            "Port": 3306,
            "RdsResourceId": "instance-1018"
        },
        {
            "Type": "TRACKED_CLUSTER",
            "Port": 0,
            "RdsResourceId": "cluster-56-2020-02-25-1399"
        },
        {
            "Endpoint": "instance-4330.demo.us-east-1.rds.amazonaws.com",
            "Type": "RDS_INSTANCE",
            "Port": 3306,
            "RdsResourceId": "instance-4330"
        }
    ]
}

aws rds deregister-db-proxy-targets --db-proxy-name the-proxy --db-cluster-identifier
 cluster-56-2020-02-25-1399

aws rds describe-db-proxy-targets --db-proxy-name the-proxy

{
    "Targets": []
}
```

```
aws rds register-db-proxy-targets --db-proxy-name the-proxy --db-cluster-identifier
 provisioned-cluster

{
    "DBProxyTargets": [
        {
            "Type": "TRACKED_CLUSTER",
            "Port": 0,
            "RdsResourceId": "provisioned-cluster"
        },
        {
            "Endpoint": "gkldje.demo.us-east-1.rds.amazonaws.com",
            "Type": "RDS_INSTANCE",
            "Port": 3306,
            "RdsResourceId": "gkldje"
        },
        {
            "Endpoint": "provisioned-1.demo.us-east-1.rds.amazonaws.com",
            "Type": "RDS_INSTANCE",
            "Port": 3306,
            "RdsResourceId": "provisioned-1"
        }
    ]
}
```

API RDS

Untuk memodifikasi proxy menggunakan RDS API, Anda menggunakan operasi ModifyDBProxy, ModifyDB, DeregisterDB, dan operasi RegisterDB. ProxyTargetGroup ProxyTargets ProxyTargets

Dengan `ModifyDBProxy`, Anda dapat mengubah properti seperti berikut:

- Kumpulan rahasia Secrets Manager yang digunakan proksi.
- Apakah TLS diperlukan.
- Batas waktu klien idle.
- Apakah harus mencatat informasi tambahan dari pernyataan SQL untuk debugging.
- Peran IAM yang digunakan untuk mengambil rahasia Secrets Manager.
- Grup keamanan yang digunakan proksi.

Dengan `ModifyDBProxyTargetGroup`, Anda dapat mengubah pengaturan terkait koneksi atau mengganti nama grup target. Saat ini, semua proksi memiliki satu grup target bernama `default`. Saat bekerja dengan grup target ini, Anda menentukan nama proksi dan `default` untuk nama grup target.

Dengan `DeregisterDBProxyTargets` dan `RegisterDBProxyTargets`, Anda dapat mengubah instans DB RDS mana yang dikaitkan dengan proksi melalui grup targetnya. Saat ini, setiap proksi dapat terhubung ke satu instans DB RDS . Grup target melacak detail koneksi untuk instans DB RDS dalam konfigurasi Multi-AZ .

Menambahkan pengguna basis data baru

Dalam beberapa kasus, Anda dapat menambahkan pengguna basis data baru ke instans atau klaster DB RDS yang terkait dengan proksi. Jika demikian, tambahkan atau ganti tujuan sebuah rahasia Secrets Manager untuk menyimpan kredensial dari pengguna tersebut. Untuk melakukan ini, pilih salah satu opsi berikut:

1. Buat rahasia Secrets Manager yang baru, dengan menggunakan prosedur yang dijelaskan dalam Menyiapkan kredensi database di AWS Secrets Manager.
2. Perbarui peran IAM untuk memberi Proksi RDS akses ke rahasia Secrets Manager baru. Untuk melakukannya, perbarui bagian sumber daya dari kebijakan peran IAM.
3. Ubah Proksi RDS untuk menambahkan rahasia Secrets Manager baru di bagian Rahasia Secrets Manager.
4. Jika pengguna baru menggantikan yang sudah ada, perbarui kredensial yang tersimpan dalam rahasia Secrets Manager proksi untuk pengguna yang sudah ada.

Menambahkan pengguna basis data baru ke basis data PostgreSQL

Saat menambahkan pengguna baru ke database PostgreSQL Anda, jika Anda telah menjalankan perintah berikut:

```
REVOKE CONNECT ON DATABASE postgres FROM PUBLIC;
```

Berikan hak istimewa `CONNECT` kepada pengguna `rdsproxyadmin` sehingga pengguna dapat memantau koneksi pada basis data target.

```
GRANT CONNECT ON DATABASE postgres TO rdsproxyadmin;
```

Anda juga dapat mengizinkan pengguna basis data target lainnya untuk melakukan pemeriksaan kondisi dengan mengubah `rdsproxyadmin` ke pengguna basis data dalam perintah di atas.

Mengubah kata sandi untuk pengguna basis data

Dalam beberapa kasus, Anda dapat mengubah kata sandi untuk pengguna basis data dalam klaster Aurora yang terkait dengan proksi. Jika demikian, perbarui rahasia Secrets Manager yang sesuai dengan kata sandi baru.

Koneksi klien dan basis data

Koneksi dari aplikasi Anda ke Proksi RDS dikenal sebagai koneksi klien. Koneksi dari proxy ke basis data adalah koneksi basis data. Saat menggunakan Proksi RDS, koneksi klien berakhir di proksi sementara koneksi basis data dikelola dalam Proksi RDS.

Pengumpulan koneksi sisi aplikasi dapat memberikan manfaat untuk mengurangi pembuatan koneksi berulang antara aplikasi Anda dan Proksi RDS.

Pertimbangkan aspek konfigurasi berikut sebelum menerapkan kumpulan koneksi sisi aplikasi:

- Masa pakai maks koneksi klien: Proksi RDS menerapkan masa pakai maksimum koneksi klien selama 24 jam. Nilai ini tidak dapat dikonfigurasi. Konfigurasikan kumpulan Anda dengan masa pakai koneksi maksimum kurang dari 24 jam guna menghindari penurunan koneksi klien yang tidak terduga.
- Batas waktu idle koneksi klien: Proksi RDS menerapkan waktu idle maksimum untuk koneksi klien. Konfigurasikan kumpulan Anda dengan batas waktu koneksi idle dengan nilai yang lebih rendah dari pengaturan batas waktu idle koneksi klien untuk Proksi RDS guna menghindari penurunan koneksi yang tidak terduga.

Jumlah maksimum koneksi klien yang dikonfigurasi dalam kumpulan koneksi sisi aplikasi Anda tidak harus dibatasi pada pengaturan max_connections untuk Proksi RDS.

Pengumpulan koneksi klien menghasilkan masa pakai koneksi klien yang lebih lama. Jika koneksi Anda mengalami penyematan, pengumpulan koneksi klien dapat mengurangi efisiensi multiplexing. Koneksi klien yang disematkan tetapi idle dalam kumpulan koneksi sisi aplikasi terus berpegang pada koneksi basis data dan mencegah koneksi basis data digunakan kembali oleh koneksi klien lainnya. Tinjau log proksi untuk memeriksa apakah koneksi Anda mengalami penyematan.

Mengonfigurasi pengaturan koneksi

Untuk menyesuaikan pengumpulan koneksi Proksi RDS, Anda dapat mengubah pengaturan berikut:

- IdleClientTimeout
- MaxConnectionsPercent
- MaxIdleConnectionsPercent
- ConnectionBorrowTimeout

IdleClientTimeout

Anda dapat menentukan berapa lama koneksi klien bisa berada dalam status idle sebelum proksi menutupnya. Nilai default-nya adalah 1.800 detik (30 menit).

Koneksi klien dianggap idle jika aplikasi tidak mengirimkan permintaan baru dalam waktu yang ditentukan setelah permintaan sebelumnya selesai. Koneksi basis data yang mendasarinya akan tetap terbuka dan dikembalikan ke kumpulan koneksi. Oleh karena itu, koneksi dapat digunakan kembali untuk koneksi klien baru. Jika Anda ingin proksi secara proaktif menghapus koneksi yang sudah tidak terpakai, turunkan batas waktu koneksi klien yang idle. Jika beban kerja Anda sering terhubung dengan proksi, maka naikkan batas waktu koneksi klien yang idle untuk menghemat biaya pembangunan koneksi.

Pengaturan ini diwakili oleh bidang batas waktu koneksi klien Idle di konsol RDS dan `IdleClientTimeout` pengaturan di AWS CLI dan API. Untuk mempelajari cara mengubah nilai kolom Batas waktu koneksi klien idle di konsol RDS, lihat AWS Management Console. Untuk mempelajari cara mengubah nilai `IdleClientTimeout` setelan, lihat perintah CLI modify-db-proxyatau operasi API ModifyDBProxy.

MaxConnectionsPercent

Anda dapat membatasi jumlah koneksi yang dapat dibuat oleh Proksi RDS dengan basis data target. Anda menentukan batas dalam bentuk persentase koneksi maksimum yang tersedia untuk basis data Anda. Pengaturan ini diwakili oleh bidang koneksi maksimum Connection pool di konsol RDS dan `MaxConnectionsPercent` pengaturan di AWS CLI dan API.

Nilai `MaxConnectionsPercent` dinyatakan sebagai persentase dari pengaturan `max_connections` untuk instans DB RDS yang digunakan oleh grup target. Proksi tidak membuat

semua koneksi ini di depan. Pengaturan ini memungkinkan proksi membuat koneksi ini karena beban kerja membutuhkannya.

Misalnya, untuk target basis data terdaftar dengan `max_connections` diatur ke 1000, dan `MaxConnectionsPercent` diatur ke 95, Proksi RDS menetapkan 950 koneksi sebagai batas atas koneksi bersamaan ke target basis data tersebut.

Efek samping umum beban kerja yang mencapai jumlah maksimum koneksi basis data yang diizinkan adalah peningkatan latensi kueri secara keseluruhan, disertai peningkatan metrik `DatabaseConnectionsBorrowLatency`. Anda dapat memantau koneksi basis data yang saat ini digunakan dan total koneksi basis data yang diizinkan dengan membandingkan metrik `DatabaseConnections` dan `MaxDatabaseConnectionsAllowed`.

Saat mengatur parameter ini, perhatikan praktik terbaik berikut:

- Izinkan headroom koneksi yang cukup untuk perubahan pola beban kerja. Sebaiknya atur parameter ini setidaknya 30% di atas penggunaan maksimum yang dipantau baru-baru ini. Karena Proksi RDS mendistribusikan ulang kuota koneksi basis data di beberapa simpul, perubahan kapasitas internal mungkin memerlukan setidaknya 30% headroom untuk koneksi tambahan guna menghindari peningkatan latensi pinjaman.
- Proksi RDS mencadangkan jumlah koneksi tertentu untuk pemantauan aktif guna mendukung failover cepat, perutean lalu lintas, dan operasi internal. Metrik `MaxDatabaseConnectionsAllowed` tidak mencakup koneksi yang dicadangkan ini. Metrik ini mewakili jumlah koneksi yang tersedia untuk melayani beban kerja, dan bisa lebih rendah dari nilai yang berasal dari pengaturan `MaxConnectionsPercent`.

  Nilai `MaxConnectionsPercent` minimum yang direkomendasikan

  - db.t3.small: 30
  - db.t3.medium atau lebih: 20

Untuk mempelajari cara mengubah nilai kolom Batas waktu maksimum kumpulan koneksi di konsol RDS, lihat AWS Management Console. Untuk mempelajari cara mengubah nilai `MaxConnectionsPercent` setelan, lihat perintah CLI modify-db-proxy-target-group atau operasi API ModifyDB. ProxyTargetGroup

Untuk informasi tentang batas koneksi basis data, lihat Jumlah maksimum koneksi basis data.

MaxIdleConnectionsPercent

Anda dapat mengontrol jumlah koneksi basis data idle yang dapat disimpan oleh Proksi RDS di kumpulan koneksi. Secara default, Proksi RDS menganggap koneksi basis data di kumpulannya menjadi idle jika tidak ada aktivitas pada koneksi selama lima menit.

Anda menentukan batas dalam bentuk persentase koneksi maksimum yang tersedia untuk basis data Anda. Nilai default-nya adalah 50 persen dari `MaxConnectionsPercent`, dan batas atasnya adalah nilai `MaxConnectionsPercent`. Dengan nilai tinggi, proksi membiarkan koneksi basis data idle dengan persentase yang tinggi tetap terbuka. Dengan nilai rendah, proksi menutup koneksi basis data idle dengan persentase yang tinggi. Jika beban kerja Anda tidak dapat diprediksi, pertimbangkan untuk mengatur nilai tinggi untuk `MaxIdleConnectionsPercent`. Jika Anda melakukannya, Proksi RDS dapat mengakomodasi lonjakan aktivitas tanpa membuka banyak koneksi basis data baru.

Pengaturan ini diwakili oleh `MaxIdleConnectionsPercent` pengaturan `DBProxyTargetGroup` di AWS CLI dan API. Untuk mempelajari cara mengubah nilai `MaxIdleConnectionsPercent` setelan, lihat perintah CLI modify-db-proxy-target-group atau operasi API ModifyDB. ProxyTargetGroup

Untuk informasi tentang batas koneksi basis data, lihat Jumlah maksimum koneksi basis data.

ConnectionBorrowTimeout

Anda dapat memilih berapa lama Proksi RDS menunggu koneksi basis data dalam kumpulan koneksi menjadi tersedia untuk digunakan sebelum menampilkan eror batas waktu. Periode default-nya adalah 120 detik. Pengaturan ini berlaku saat jumlah koneksi mencapai titik maksimum, sehingga tidak ada koneksi yang tersedia dalam kumpulan koneksi. Pengaturan ini juga berlaku ketika tidak ada instans basis data yang tersedia untuk menangani permintaan, seperti saat operasi failover sedang berlangsung. Dengan pengaturan ini, Anda dapat mengatur periode tunggu terbaik untuk aplikasi Anda tanpa mengubah batas waktu kueri dalam kode aplikasi Anda.

Pengaturan ini diwakili oleh bidang batas waktu pinjam koneksi di konsol RDS atau `ConnectionBorrowTimeout` pengaturan `DBProxyTargetGroup` di API atau. AWS CLI Untuk mempelajari cara mengubah nilai kolom Batas waktu peminjaman koneksi di konsol RDS, lihat AWS Management Console. Untuk mempelajari cara mengubah nilai `ConnectionBorrowTimeout` setelan, lihat perintah CLI modify-db-proxy-target-group atau operasi API ModifyDB. ProxyTargetGroup

Menghindari penyematan

Multiplexing akan lebih efisien saat permintaan basis data tidak bergantung pada informasi status dari permintaan sebelumnya. Dalam kasus ini, Proksi RDS dapat menggunakan kembali koneksi saat setiap transaksi selesai. Contoh informasi status tersebut mencakup sebagian besar variabel dan parameter konfigurasi yang dapat diubah melalui pernyataan `SET` atau `SELECT`. Secara default, transaksi SQL pada koneksi klien dapat bermultipleks antar-koneksi basis data acuan.

Koneksi ke proksi dapat memasukkan status yang disebut sebagai penyematan. Saat koneksi disematkan, setiap transaksi berikutnya akan menggunakan koneksi basis data acuan yang sama hingga sesi berakhir. Koneksi klien lainnya juga tidak dapat menggunakan kembali koneksi basis data tersebut hingga sesi berakhir. Sesi berakhir saat koneksi klien terputus.

Proksi RDS secara otomatis menyematkan koneksi klien ke koneksi DB tertentu saat mendeteksi perubahan sebuah status sesi yang tidak sesuai untuk sesi lainnya. Penyematan mengurangi efektivitas penggunaan kembali koneksi. Jika semua atau hampir semua koneksi Anda mengalami penyematan, pertimbangkan untuk mengubah kode aplikasi atau beban kerja untuk mengurangi kondisi yang menyebabkan penyematan.

Misalnya, aplikasi Anda mengubah variabel sesi atau parameter konfigurasi. Dalam hal ini, pernyataan selanjutnya dapat mengandalkan variabel atau parameter baru yang berlaku. Oleh karena itu, saat Proksi RDS memproses permintaan untuk mengubah variabel sesi atau pengaturan konfigurasi, sesi ini akan disematkan ke koneksi DB. Dengan demikian, tahap sesi tetap berfungsi untuk semua transaksi berikutnya dalam sesi yang sama.

Untuk beberapa mesin basis data, aturan ini tidak berlaku untuk semua parameter yang dapat Anda atur. Proksi RDS melacak pernyataan dan variabel tertentu. Oleh karena itu, Proksi RDS tidak menyematkan sesi saat Anda mengubahnya. Dalam kasus ini, Proksi RDS hanya menggunakan kembali koneksi untuk sesi lain yang memiliki nilai yang sama untuk pengaturan tersebut. Untuk detail tentang apa yang dilacak Proksi RDS untuk mesin basis data, lihat berikut ini:

- Apa yang Dilacak Proksi RDS untuk basis data RDS for SQL Server
- Apa yang dilacak Proksi RDS untuk basis data RDS for MariaDB dan RDS for MySQL

Apa yang Dilacak Proksi RDS untuk basis data RDS for SQL Server

Berikut ini adalah pernyataan SQL Server yang dilacak Proksi RDS:

- `USE`

- SET ANSI_NULLS
- SET ANSI_PADDING
- SET ANSI_WARNINGS
- SET ARITHABORT
- SET CONCAT_NULL_YIELDS_NULL
- SET CURSOR_CLOSE_ON_COMMIT
- SET DATEFIRST
- SET DATEFORMAT
- SET LANGUAGE
- SET LOCK_TIMEOUT
- SET NUMERIC_ROUNDABORT
- SET QUOTED_IDENTIFIER
- SET TEXTSIZE
- SET TRANSACTION ISOLATION LEVEL

Apa yang dilacak Proksi RDS untuk basis data RDS for MariaDB dan RDS for MySQL

Berikut ini adalah pernyataan MariaDB dan MySQL yang dilacak Proksi RDS:

- DROP DATABASE
- DROP SCHEMA
- USE

Berikut ini adalah variabel MySQL dan MariaDB yang dilacak Proksi RDS:

- AUTOCOMMIT
- AUTO_INCREMENT_INCREMENT
- CHARACTER SET (or CHAR SET)
- CHARACTER_SET_CLIENT
- CHARACTER_SET_DATABASE
- CHARACTER_SET_FILESYSTEM

- CHARACTER_SET_CONNECTION
- CHARACTER_SET_RESULTS
- CHARACTER_SET_SERVER
- COLLATION_CONNECTION
- COLLATION_DATABASE
- COLLATION_SERVER
- INTERACTIVE_TIMEOUT
- NAMES
- NET_WRITE_TIMEOUT
- QUERY_CACHE_TYPE
- SESSION_TRACK_SCHEMA
- SQL_MODE
- TIME_ZONE
- TRANSACTION_ISOLATION (or TX_ISOLATION)
- TRANSACTION_READ_ONLY (or TX_READ_ONLY)
- WAIT_TIMEOUT

Meminimalkan penyematan

Penyetelan performa untuk Proksi RDS meliputi upaya memaksimalkan penggunaan kembali koneksi tingkat transaksi (multiplexing) dengan meminimalkan penyematan.

Anda dapat meminimalkan penyematan dengan melakukan hal berikut:

- Hindari permintaan basis data yang tidak perlu yang dapat menyebabkan penyematan.
- Atur variabel dan pengaturan konfigurasi secara konsisten di semua koneksi. Dengan demikian, sesi berikutnya cenderung menggunakan kembali koneksi yang memiliki pengaturan tertentu tersebut.

  Akan tetapi, untuk pengaturan PostgreSQL, sebuah variabel bisa menimbulkan penyematan sesi.
- Untuk basis data keluarga MySQL, terapkan sebuah filter penyematan sesi ke proksi. Anda dapat mengecualikan jenis operasi tertentu dari penyematan sesi jika Anda mengetahui bahwa tindakan ini tidak memengaruhi operasi yang benar aplikasi Anda.

- Lihat seberapa sering penyematan terjadi dengan memantau CloudWatch metrik `DatabaseConnectionsCurrentlySessionPinned` Amazon. Untuk informasi tentang ini dan CloudWatch metrik lainnya, lihatMemantau metrik Proxy RDS dengan Amazon CloudWatch.
- Jika menggunakan pernyataan `SET` untuk melakukan inisialisasi yang identik untuk setiap koneksi klien, Anda dapat melakukannya sekaligus mempertahankan multiplexing tingkat transaksi. Dalam kasus ini, Anda memindahkan pernyataan yang menyiapkan status sesi awal ke dalam kueri inisialisasi yang digunakan oleh proksi. Properti ini adalah string yang berisi satu atau beberapa pernyataan SQL, yang dipisahkan oleh titik koma.

  Misalnya, Anda dapat menentukan kueri inisialisasi untuk proksi yang menetapkan parameter konfigurasi tertentu. Kemudian, Proksi RDS menerapkan pengaturan tersebut setiap kali koneksi baru untuk proksi itu disiapkan. Anda dapat menghapus pernyataan `SET` yang sesuai dari kode aplikasi, sehingga tidak mengganggu multiplexing tingkat transaksi.

  Untuk metrik tentang seberapa sering penyematan terjadi pada proksi, lihat Memantau metrik Proxy RDS dengan Amazon CloudWatch.

Kondisi yang menyebabkan penyematan untuk semua keluarga mesin

Proksi menyematkan sesi ke koneksi saat ini dalam situasi berikut ketika multiplexing dapat menyebabkan perilaku yang tidak terduga:

- Pernyataan apa pun dengan ukuran teks lebih dari 16 KB bisa menyebabkan proksi menyematkan sesi.

Kondisi yang menyebabkan penyematan untuk RDS for Microsoft SQL Server

Untuk RDS for SQL Server, interaksi berikut dapat menyebabkan penyematan:

- Menggunakan beberapa kumpulan hasil yang aktif (MARS). Untuk informasi tentang MARS, lihat dokumentasi SQL Server.
- Menggunakan komunikasi koordinator transaksi terdistribusi (DTC).
- Membuat tabel sementara, transaksi, kursor, atau pernyataan yang disiapkan.
- Menggunakan pernyataan `SET` berikut:
  - `SET ANSI_DEFAULTS`
  - `SET ANSI_NULL_DFLT`

- `SET ARITHIGNORE`
- `SET DEADLOCK_PRIORITY`
- `SET FIPS_FLAGGER`
- `SET FMTONLY`
- `SET FORCEPLAN`
- `SET IDENTITY_INSERT`
- `SET NOCOUNT`
- `SET NOEXEC`
- `SET OFFSETS`
- `SET PARSEONLY`
- `SET QUERY_GOVERNOR_COST_LIMIT`
- `SET REMOTE_PROC_TRANSACTIONS`
- `SET ROWCOUNT`
- `SET SHOWPLAN_ALL`, `SHOWPLAN_TEXT`, dan `SHOWPLAN_XML`
- `SET STATISTICS`
- `SET XACT_ABORT`

Kondisi yang menyebabkan penyematan untuk RDS for MariaDB dan RDS for MySQL

Untuk MariaDB dan MySQL, interaksi berikut juga dapat menyebabkan penyematan:

- Pernyataan kunci tabel eksplisit `LOCK TABLE`, `LOCK TABLES`, atau `FLUSH TABLES WITH READ LOCK` menyebabkan proksi menyematkan sesi.
- Membuat kunci bernama dengan menggunakan `GET_LOCK` menyebabkan proksi menyematkan sesi.
- Menetapkan variabel pengguna atau variabel sistem (dengan beberapa pengecualian) menyebabkan proksi menyematkan sesi. Jika situasi ini terlalu banyak mengurangi frekuensi penggunaan kembali koneksi Anda, pilih operasi `SET` agar tidak menyebabkan penyematan. Untuk informasi tentang cara melakukannya dengan mengatur properti filter penyematan sesi, lihat Membuat Proksi RDS dan Mengubah Proksi RDS.
- Membuat tabel sementara menyebabkan proksi menyematkan sesi. Dengan begitu, konten tabel sementara dipertahankan sepanjang sesi, terlepas dari batasan transaksi.

- Memanggil fungsi `ROW_COUNT`, `FOUND_ROWS`, dan `LAST_INSERT_ID` terkadang menyebabkan penyematan.
- Pernyataan yang disiapkan menyebabkan proksi menyematkan sesi. Aturan ini berlaku terlepas dari apakah pernyataan yang disiapkan menggunakan teks SQL maupun protokol biner.
- Proksi RDS tidak menyematkan koneksi saat Anda menggunakan SET LOCAL.
- Memanggil prosedur tersimpan dan fungsi tersimpan tidak menyebabkan pinning. Proksi RDS tidak mendeteksi perubahan status sesi apa pun yang terjadi akibat perintah tersebut. Pastikan aplikasi Anda tidak mengubah status sesi di dalam rutinitas tersimpan jika Anda mengandalkan status sesi tersebut untuk bertahan di seluruh transaksi. Misalnya, Proxy RDS saat ini tidak kompatibel dengan prosedur tersimpan yang membuat tabel sementara yang bertahan di semua transaksi.

Jika memiliki pengetahuan mendalam tentang perilaku aplikasi, Anda dapat menangani perilaku penyematan untuk pernyataan aplikasi tertentu. Untuk melakukannya, pilih opsi Filter penyematan sesi saat membuat proksi. Saat ini, Anda dapat memilih untuk tidak menggunakan penyematan sesi untuk pengaturan variabel sesi dan pengaturan konfigurasi.

Kondisi yang menyebabkan penyematan untuk RDS for PostgreSQL

Untuk PostgreSQL, interaksi berikut juga menyebabkan penyematan:

- Menggunakan `SET` perintah.
- Menggunakan`PREPARE`,`DISCARD`,`DEALLOCATE`, atau `EXECUTE` perintah untuk mengelola pernyataan yang disiapkan.
- Membuat urutan sementara, tabel, atau tampilan.
- Mendeklarasikan kursor.
- Membuang status sesi.
- Mendengarkan di saluran notifikasi.
- Memuat modul perpustakaan seperti`auto_explain`.
- Memanipulasi urutan menggunakan fungsi seperti `nextval` dan. `setval`
- Berinteraksi dengan kunci menggunakan fungsi seperti `pg_advisory_lock` dan`pg_try_advisory_lock`.
- Mengatur parameter, atau mengatur ulang parameter ke defaultnya. Secara khusus, menggunakan `SET` dan `set_config` perintah untuk menetapkan nilai default ke variabel sesi.
- Memanggil prosedur tersimpan dan fungsi tersimpan tidak menyebabkan pinning. Proksi RDS tidak mendeteksi perubahan status sesi apa pun yang terjadi akibat perintah tersebut. Pastikan aplikasi

Anda tidak mengubah status sesi di dalam rutinitas tersimpan jika Anda mengandalkan status sesi tersebut untuk bertahan di seluruh transaksi. Misalnya, Proxy RDS saat ini tidak kompatibel dengan prosedur tersimpan yang membuat tabel sementara yang bertahan di semua transaksi.

Menghapus Proksi RDS

Anda dapat menghapus proksi saat tidak lagi membutuhkannya. Atau Anda dapat menghapus proksi jika Anda merasa instans atau klaster DB yang terkait dengannya sedang dalam perbaikan.

AWS Management Console

Untuk menghapus proksi

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Proksi.
3. Pilih proksi yang akan dihapus dari daftar.
4. Pilih Hapus Proksi.

AWS CLI

Untuk menghapus proxy DB, gunakan AWS CLI perintah delete-db-proxy. Untuk menghapus asosiasi terkait, gunakan juga deregister-db-proxy-targetsperintah.

```
aws rds delete-db-proxy --name proxy_name
```

```
aws rds deregister-db-proxy-targets
    --db-proxy-name proxy_name
    [--target-group-name target_group_name]
    [--target-ids comma_separated_list]       # or
    [--db-instance-identifiers instance_id]       # or
    [--db-cluster-identifiers cluster_id]
```

API RDS

Untuk menghapus proksi DB, panggil fungsi Amazon RDS API DeleteDBProxy. Untuk menghapus item dan asosiasi terkait, Anda juga memanggil fungsi DeleteDB ProxyTargetGroup dan DeregisterDB. ProxyTargets

Bekerja dengan titik akhir Proksi Amazon RDS

Pelajari titik akhir untuk Proksi RDS dan cara menggunakannya. Dengan titik akhir proksi, Anda dapat memanfaatkan kemampuan berikut:

- Anda dapat menggunakan beberapa titik akhir dengan proksi untuk memantau dan memecahkan masalah koneksi dari aplikasi yang berbeda secara independen.
- Anda dapat menggunakan titik akhir lintas-VPC untuk mengizinkan akses ke basis data dalam satu VPC dari sumber daya seperti instans Amazon EC2 di VPC yang berbeda.

Topik

Ikhtisar titik akhir proksi

Bekerja dengan titik akhir Proksi RDS melibatkan jenis prosedur yang sama seperti dengan titik akhir instans RDS. Jika Anda belum terbiasa dengan titik akhir RDS, temukan informasi selengkapnya di Menghubungkan ke instans DB yang menjalankan mesin basis data MySQL dan Menghubungkan ke instans DB yang menjalankan mesin basis data PostgreSQL.

Untuk titik akhir proksi yang Anda buat, Anda juga dapat mengaitkan titik akhir tersebut dengan cloud privat virtual (VPC) yang berbeda dengan yang digunakan oleh proksi itu sendiri. Dengan demikian, Anda dapat terhubung ke proksi dari VPC yang berbeda, misalnya VPC yang digunakan oleh aplikasi lain dalam organisasi Anda.

Untuk informasi tentang batas yang terkait dengan titik akhir proksi, lihat Batasan untuk titik akhir proksi.

Dalam log Proksi RDS, setiap entri diawali dengan nama titik akhir proksi terkait. Nama ini bisa berupa nama yang Anda tentukan untuk titik akhir yang ditentukan pengguna. Atau, bisa menjadi nama khusus `default` untuk titik akhir default proksi yang melakukan permintaan baca/tulis.

Setiap titik akhir proxy memiliki kumpulan CloudWatch metriknya sendiri. Anda dapat memantau metrik untuk semua titik akhir proksi. Anda juga dapat memantau metrik untuk titik akhir tertentu, atau untuk semua titik akhir baca/tulis atau hanya-baca dari proksi. Untuk informasi selengkapnya, lihat Memantau metrik Proxy RDS dengan Amazon CloudWatch.

Titik akhir proksi menggunakan mekanisme autentikasi yang sama seperti proksi yang terkait. Proksi RDS secara otomatis menyiapkan izin dan otorisasi untuk titik akhir yang ditentukan pengguna, yang konsisten dengan properti proksi terkait.

Titik akhir proksi untuk klaster DB Multi-AZ

Secara default, titik akhir yang Anda hubungkan saat Anda menggunakan Proksi RDS dengan klaster DB Multi-AZ memiliki kapabilitas baca/tulis. Akibatnya, titik akhir ini mengirimkan semua permintaan ke instans penulis klaster. Semua koneksi tersebut dihitung terhadap nilai `max_connections` untuk instans penulis. Jika proksi Anda dikaitkan dengan klaster DB Multi-AZ, Anda dapat membuat titik akhir baca/tulis atau hanya-baca tambahan untuk proksi itu.

Anda dapat menggunakan titik akhir hanya baca dengan proksi Anda untuk kueri hanya baca. Anda dapat melakukannya dengan cara yang sama seperti Anda menggunakan titik akhir pembaca untuk klaster DB multi-AZ. Tindakan ini membantu Anda memanfaatkan skalabilitas baca klaster Multi-AZ dengan satu atau beberapa instans DB pembaca. Anda dapat menjalankan lebih banyak kueri simultan dan membuat lebih banyak koneksi simultan dengan menggunakan titik akhir hanya-baca dan menambahkan lebih banyak instans DB pembaca untuk klaster DB Multi-AZ Anda sesuai kebutuhan. Titik akhir pembaca ini membantu meningkatkan skalabilitas baca aplikasi padat kueri. Titik akhir pembaca juga membantu meningkatkan ketersediaan koneksi Anda jika instans DB pembaca di klaster Anda menjadi tidak tersedia.

Titik akhir pembaca untuk klaster DB Multi-AZ

Dengan Proksi RDS, Anda dapat membuat dan menggunakan titik akhir pembaca. Namun, titik akhir ini hanya berfungsi untuk proksi yang terkait dengan klaster DB Multi-AZ. Jika menggunakan RDS CLI atau API, Anda dapat melihat atribut `TargetRole` dengan nilai `READ_ONLY`. Anda dapat memanfaatkan proksi tersebut dengan mengubah target proksi dari instans DB RDS untuk klaster DB Multi-AZ.

Anda dapat membuat dan terhubung ke titik akhir hanya-baca yang disebut titik akhir pembaca jika Anda menggunakan Proksi RDS dengan klaster DB Multi-AZ.

Cara titik akhir pembaca membantu ketersediaan aplikasi

Dalam beberapa kasus, instans pembaca di klaster Anda mungkin menjadi tidak tersedia. Jika itu terjadi, koneksi yang menggunakan titik akhir pembaca proksi DB dapat pulih lebih cepat daripada koneksi yang menggunakan titik akhir pembaca klaster Multi-AZ. Proksi RDS merutekan koneksi hanya ke instans pembaca yang tersedia dalam klaster. Tidak ada penundaan dikarenakan caching DNS saat instans menjadi tidak tersedia.

Jika koneksi di-multipleks, Proksi RDS mengarahkan kueri berikutnya ke instans pembaca yang berbeda tanpa mengganggu aplikasi. Jika instans pembaca dalam status tidak tersedia, semua koneksi klien ke titik akhir instans tersebut ditutup.

Jika koneksi disematkan, kueri berikutnya pada koneksi akan menampilkan kesalahan. Namun, aplikasi Anda dapat langsung terhubung kembali ke titik akhir proksi yang sama. Proksi RDS merutekan koneksi ke instans DB pembaca yang berbeda yang ada dalam status `available`. Ketika menghubungkan kembali secara manual, Proksi RDS tidak memeriksa keterlambatan replikasi antara instans pembaca lama dan baru.

Jika klaster DB Multi-AZ Anda tidak memiliki instans pembaca yang tersedia, Proksi RDS akan mencoba terhubung ke titik akhir pembaca jika tersedia. Jika tidak ada instans pembaca yang tersedia dalam periode batas waktu peminjaman koneksi, upaya koneksi akan gagal. Jika instans pembaca menjadi tersedia, upaya koneksi berhasil.

Cara titik akhir pembaca membantu skalabilitas kueri

Titik akhir pembaca untuk proksi membantu skalabilitas kueri klaster DB Multi-AZ dengan cara berikut:

- Jika praktis, Proksi RDS menggunakan instans DB pembaca yang sama untuk semua masalah kueri yang menggunakan koneksi titik akhir pembaca tertentu. Dengan demikian, satu kumpulan kueri terkait pada tabel yang sama dapat memanfaatkan caching, optimasi rencana, dan sebagainya, pada instans DB tertentu.
- Jika instans DB pembaca tidak tersedia, pengaruh pada aplikasi Anda bergantung pada apakah sesi di-multipleks atau disematkan. Jika sesi di-multipleks, Proksi RDS merutekan setiap kueri berikutnya untuk instans DB pembaca yang berbeda tanpa campur tangan Anda. Jika sesi disematkan, aplikasi Anda mengalami kesalahan dan harus dihubungkan kembali. Anda dapat

segera terhubung kembali ke titik akhir pembaca dan Proksi RDS merutekan koneksi ke instans DB pembaca yang tersedia. Untuk informasi selengkapnya tentang cara memultipleks dan menyematkan untuk sesi proksi, lihat Ikhtisar konsep Proksi RDS.

Mengakses basis data RDS di seluruh VPC

Secara default, semua komponen tumpukan teknologi RDS Anda berada dalam Amazon VPC yang sama. Misalnya, anggaplah bahwa aplikasi yang berjalan di instans Amazon EC2 terhubung ke instans DB Amazon RDS. Dalam hal ini, server aplikasi dan basis data keduanya harus berada dalam VPC yang sama.

Dengan Proksi RDS, Anda dapat menyiapkan akses ke instans DB Amazon RDS di satu VPC dari sumber daya di VPC lain, seperti instans EC2. Misalnya, organisasi Anda mungkin memiliki beberapa aplikasi yang mengakses sumber daya basis data yang sama. Setiap aplikasi mungkin berada dalam VPC-nya sendiri.

Untuk mengaktifkan akses lintas-VPC, Anda membuat titik akhir baru untuk proksi tersebut. Proksi itu sendiri berada di VPC yang sama dengan instans DB Amazon RDS. Namun, titik akhir lintas-VPC berada di VPC lain, bersama dengan sumber daya lain seperti instans EC2. Titik akhir lintas-VPC dikaitkan dengan subnet dan grup keamanan dari VPC yang sama sebagai EC2 dan sumber daya lainnya. Pengaitan ini memungkinkan Anda terhubung ke titik akhir dari aplikasi yang jika sebaliknya tidak dapat mengakses basis data dikarenakan pembatasan VPC.

Langkah-langkah berikut menjelaskan cara membuat dan mengakses titik akhir lintas-VPC melalui Proksi RDS:

1. Buat dua VPC, atau pilih dua VPC yang sudah Anda gunakan untuk pekerjaan RDS. Setiap VPC harus memiliki sumber daya jaringan yang terkaitnya sendiri seperti gateway internet, tabel rute, subnet, dan grup keamanan. Jika Anda hanya memiliki satu VPC, Anda dapat melihat Mulai menggunakan Amazon RDS untuk mengetahui langkah-langkah penyiapan VPC lain agar berhasil menggunakan RDS. Anda juga dapat memeriksa VPC yang ada di konsol Amazon EC2 untuk melihat jenis sumber daya yang dapat dihubungkan bersama.
2. Buat proksi DB yang terkait dengan instans Amazon RDS yang ingin Anda hubungkan. Ikuti prosedur di Membuat Proksi RDS.
3. Pada halaman Detail untuk proksi Anda di konsol RDS, di bagian Titik akhir proksi, pilih Buat titik akhir. Ikuti prosedur di Membuat titik akhir proksi.
4. Pilih apakah lintas-titik akhir VPC akan bersifat baca/tulis atau hanya-baca.

5. Alih-alih menerima pengaturan default VPC yang sama dengan instans DB Amazon RDS, pilih VPC yang berbeda. VPC ini harus berada di Wilayah AWS yang sama dengan VPC tempat proksi berada.
6. Sekarang, alih-alih menerima pengaturan default untuk subnet dan grup keamanan dari VPC yang sama dengan instans DB Amazon RDS, buatlah pilihan baru. Buat pilihan ini berdasarkan subnet dan grup keamanan dari VPC yang Anda pilih.
7. Anda tidak perlu mengubah pengaturan apa pun untuk rahasia Secrets Manager. Kredensial yang sama dapat digunakan untuk semua titik akhir proksi Anda, terlepas dari VPC tempat setiap titik akhir berada.
8. Tunggu sampai titik akhir baru untuk mencapai status Tersedia.
9. Buat catatan nama titik akhir lengkap. Ini adalah nilai yang berakhiran *`Region_name`*`.rds.amazonaws.com` yang Anda berikan sebagai bagian dari string koneksi untuk aplikasi basis data Anda.
10. Akses titik akhir baru dari sumber daya di VPC yang sama dengan titik akhir. Cara mudah untuk menguji proses ini adalah membuat instans EC2 baru di VPC ini. Kemudian, masuk ke instans EC2 dan jalankan perintah `mysql` atau `psql` untuk terhubung dengan menggunakan nilai titik akhir dalam string koneksi Anda.

Membuat titik akhir proksi

Konsol

Untuk membuat titik akhir proksi

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Proksi.
3. Klik nama proksi yang titik akhir barunya ingin dibuat.

   Halaman detail untuk proksi tersebut muncul.
4. Di bagian Titik akhir proksi, pilih Buat titik akhir proksi.

   Jendela Buat titik akhir proksi akan muncul.
5. Untuk Nama titik akhir proksi, masukkan nama deskriptif pilihan Anda.

6. Untuk Peran target, pilih apakah akan membuat titik akhir lintas-VPC bersifat baca/tulis atau hanya-baca.

   Koneksi yang menggunakan titik akhir baca/tulis dapat melakukan segala jenis operasi, seperti pernyataan bahasa definisi data (DDL), pernyataan bahasa manipulasi data (DML), dan kueri. Titik akhir ini terhubung ke instans primer klaster DB RDS. Anda dapat menggunakan titik akhir baca/tulis untuk operasi basis data umum jika Anda hanya menggunakan titik akhir tunggal dalam aplikasi Anda. Anda juga dapat menggunakan endpoint baca/tulis untuk operasi administratif, aplikasi pemrosesan transaksi online (OLTP), dan extract-transform-load (ETL) pekerjaan.

   Koneksi yang menggunakan titik akhir hanya-baca hanya dapat melakukan kueri. Proksi RDS dapat menggunakan salah satu instans pembaca untuk setiap koneksi ke titik akhir. Dengan begitu, aplikasi intensif kueri dapat memanfaatkan kemampuan pengklasteran klaster DB Multi-AZ. Koneksi hanya-baca ini tidak membebankan overhead apa pun pada instans primer klaster. Dengan begitu, kueri pelaporan dan analisis Anda tidak akan memperlambat operasi menulis aplikasi OLTP Anda.
7. Untuk Cloud Privat Virtual (VPC), pilih default untuk mengakses titik akhir dari instans EC2 yang sama atau sumber daya lain yang biasanya digunakan untuk mengakses proksi atau basis data terkaitnya. Untuk menyiapkan akses lintas-VPC untuk proksi ini, pilih VPC selain default. Untuk informasi selengkapnya tentang akses lintas-VPC, lihat Mengakses basis data RDS di seluruh VPC.
8. Untuk Subnet, Proksi RDS mengisi subnet yang sama dengan proksi terkait secara default. Untuk membatasi akses ke titik akhir agar hanya sebagian dari rentang alamat VPC dapat terhubung, hapus satu subnet atau lebih.
9. Untuk Grup keamanan VPC, Anda dapat memilih grup keamanan yang sudah ada atau membuat grup keamanan baru. Proksi RDS mengisi grup keamanan atau beberapa grup keamanan yang sama sebagai proksi terkait secara default. Jika aturan masuk dan keluar proksi sesuai untuk titik akhir ini, pertahankan pilihan default.

   Jika Anda memilih untuk membuat grup keamanan baru, tentukan nama grup keamanan tersebut di halaman ini. Kemudian, edit pengaturan grup keamanan dari konsol EC2 nanti.
10. Pilih Buat titik akhir proksi.

AWS CLI

Untuk membuat titik akhir proxy, gunakan AWS CLI create-db-proxy-endpointperintah.

Sertakan parameter-parameter yang diperlukan berikut:

- `--db-proxy-name` *`value`*
- `--db-proxy-endpoint-name` *`value`*
- `--vpc-subnet-ids` *`list_of_ids`*. Pisahkan ID subnet dengan spasi. Anda tidak menentukan ID dari VPC itu sendiri.

Anda juga dapat menentukan parameter opsional berikut:

- `--target-role { READ_WRITE | READ_ONLY }`. Pengaturan default parameter ini adalah `READ_WRITE`. Ketika proksi dikaitkan dengan klaster DB Multi-AZ yang hanya berisi instans DB penulis, Anda tidak dapat menentukan `READ_ONLY`. Untuk informasi selengkapnya tentang tujuan penggunaan titik akhir hanya-baca dengan klaster DB Multi-AZ, lihat Titik akhir pembaca untuk klaster DB Multi-AZ.
- `--vpc-security-group-ids` *`value`*. Pisahkan ID grup keamanan dengan spasi. Jika Anda menghilangkan parameter ini, Proksi RDS menggunakan grup keamanan default untuk VPC. Proksi RDS menentukan VPC berdasarkan ID subnet yang Anda tentukan untuk parameter `--vpc-subnet-ids`.

Example

Contoh berikut membuat titik akhir proksi bernama `my-endpoint`.

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-proxy-endpoint \
  --db-proxy-name my-proxy \
  --db-proxy-endpoint-name my-endpoint \
  --vpc-subnet-ids subnet_id subnet_id subnet_id ... \
  --target-role READ_ONLY \
  --vpc-security-group-ids security_group_id ]
```

Untuk Windows:

```
aws rds create-db-proxy-endpoint ^
  --db-proxy-name my-proxy ^
  --db-proxy-endpoint-name my-endpoint ^
  --vpc-subnet-ids subnet_id_1 subnet_id_2 subnet_id_3 ... ^
```

```
  --target-role READ_ONLY ^
  --vpc-security-group-ids security_group_id
```

API RDS

Untuk membuat titik akhir proxy, gunakan tindakan ProxyEndpointCreateDB API RDS.

Melihat titik akhir proksi

Konsol

Untuk melihat detail titik akhir proksi

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Proksi.
3. Dalam daftar, pilih proksi yang titik akhirnya ingin dilihat. Klik nama proksi untuk melihat halaman detailnya.
4. Di bagian Titik akhir proksi, pilih titik akhir yang ingin dilihat. Klik namanya untuk melihat halaman detailnya.
5. Periksa parameter yang nilainya Anda minati. Anda dapat mengubah properti seperti berikut ini:
   - Apakah titik akhir ini baca/tulis atau hanya-baca.
   - Alamat titik akhir yang Anda gunakan dalam string koneksi basis data.
   - VPC, subnet, dan grup keamanan yang terkait dengan titik akhir.

AWS CLI

Untuk melihat satu atau beberapa titik akhir proxy, gunakan AWS CLI describe-db-proxy-endpointsperintah.

Anda dapat menyertakan parameter opsional berikut:

- `--db-proxy-endpoint-name`
- `--db-proxy-name`

Contoh berikut menjelaskan titik akhir proksi `my-endpoint`.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds describe-db-proxy-endpoints \
  --db-proxy-endpoint-name my-endpoint
```

Untuk Windows:

```
aws rds describe-db-proxy-endpoints ^
  --db-proxy-endpoint-name my-endpoint
```

API RDS

Untuk mendeskripsikan satu atau beberapa titik akhir proxy, gunakan operasi RDS API ProxyEndpointsDescribeDB.

Mengubah titik akhir proksi

Konsol

Untuk mengubah satu atau beberapa titik akhir proksi

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Proksi.
3. Dalam daftar, pilih proksi yang titik akhirnya ingin diubah. Klik nama proksi untuk melihatnya
4. Di bagian Titik akhir proksi, pilih titik akhir yang ingin diubah. Anda dapat memilihnya dalam daftar, atau mengeklik namanya untuk melihat halaman detail.
5. Di halaman detail proksi, di bagian Titik akhir proksi, pilih Edit. Atau, di halaman detail titik akhir proksi, untuk Tindakan, pilih Edit.
6. Ubah nilai parameter yang ingin dimodifikasi.
7. Pilih Simpan perubahan.

AWS CLI

Untuk memodifikasi titik akhir proxy, gunakan AWS CLI modify-db-proxy-endpointperintah dengan parameter yang diperlukan berikut:

- `--db-proxy-endpoint-name`

Tentukan perubahan ke properti titik akhir dengan menggunakan satu atau beberapa parameter berikut:

- `--new-db-proxy-endpoint-name`
- `--vpc-security-group-ids`. Pisahkan ID grup keamanan dengan spasi.

Contoh berikut mengganti nama titik akhir proksi `my-endpoint` menjadi `new-endpoint-name`.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-proxy-endpoint \
  --db-proxy-endpoint-name my-endpoint \
  --new-db-proxy-endpoint-name new-endpoint-name
```

Untuk Windows:

```
aws rds modify-db-proxy-endpoint ^
  --db-proxy-endpoint-name my-endpoint ^
  --new-db-proxy-endpoint-name new-endpoint-name
```

API RDS

Untuk memodifikasi titik akhir proxy, gunakan operasi RDS API ProxyEndpointModifyDB.

Menghapus titik akhir proksi

Anda dapat menghapus titik akhir proksi menggunakan konsol seperti yang dijelaskan berikut ini.

> **Note**
>
> Anda tidak dapat menghapus titik akhir proksi default yang dibuat secara otomatis oleh Proksi RDS untuk setiap proksi.
> Saat Anda menghapus proksi, Proksi RDS secara otomatis menghapus semua titik akhir terkait.

Konsol

Untuk menghapus titik akhir proksi menggunakan AWS Management Console

1. Di panel navigasi, pilih Proksi.
2. Dalam daftar, pilih proksi yang titik akhirnya ingin dihapus. Klik nama proksi untuk melihat halaman detailnya.
3. Di bagian Titik akhir proksi, pilih titik akhir yang ingin dihapus. Anda dapat memilih satu atau beberapa titik akhir dalam daftar, atau mengeklik nama titik akhir tunggal untuk melihat halaman detail.
4. Di halaman detail proksi, di bagian Titik akhir proksi, pilih Hapus. Atau, di halaman detail titik akhir proksi, untuk Tindakan, pilih Hapus.

AWS CLI

Untuk menghapus titik akhir proxy, jalankan delete-db-proxy-endpointperintah dengan parameter yang diperlukan berikut:

- `--db-proxy-endpoint-name`

Perintah berikut akan menghapus titik akhir proksi bernama `my-endpoint`.

Untuk Linux, macOS, atau Unix:

```
aws rds delete-db-proxy-endpoint \
  --db-proxy-endpoint-name my-endpoint
```

Untuk Windows:

```
aws rds delete-db-proxy-endpoint ^
  --db-proxy-endpoint-name my-endpoint
```

API RDS

Untuk menghapus titik akhir proxy dengan RDS API, jalankan operasi ProxyEndpointDeleteDB. Tentukan nama titik akhir proksi untuk parameter `DBProxyEndpointName`.

Batasan untuk titik akhir proksi

Titik akhir Proksi RDS memiliki batasan berikut:

- Setiap proksi memiliki titik akhir default yang dapat dimodifikasi, tetapi tidak dapat dibuat atau dihapus.
- Jumlah maksimum titik akhir yang ditentukan pengguna untuk proksi adalah 20. Dengan demikian, proksi dapat memiliki hingga 21 titik akhir: titik akhir default, ditambah 20 titik akhir yang Anda buat.
- Jika Anda mengaitkan titik akhir tambahan dengan proksi, Proksi RDS secara otomatis menentukan instans DB dalam klaster Anda yang digunakan untuk setiap titik akhir.

Memantau metrik Proxy RDS dengan Amazon CloudWatch

Anda dapat memantau Proxy RDS dengan menggunakan Amazon CloudWatch. CloudWatch mengumpulkan dan memproses data mentah dari proxy menjadi metrik yang dapat dibaca. near-real-time Untuk menemukan metrik ini di CloudWatch konsol, pilih Metrik, lalu pilih RDS, dan pilih Metrik Per-Proxy. Untuk informasi selengkapnya, lihat Menggunakan CloudWatch metrik Amazon di Panduan CloudWatch Pengguna Amazon.

> **Note**
>
> RDS menerbitkan metrik ini untuk setiap instans Amazon EC2 yang mendasarinya yang terkait dengan proksi. Satu proksi dapat dilayani oleh lebih dari satu instans EC2. Gunakan CloudWatch statistik untuk menggabungkan nilai untuk proxy di semua instance terkait. Beberapa metrik ini mungkin tidak terlihat hingga koneksi berhasil disambungkan terlebih dahulu oleh proksi.

Dalam log Proksi RDS, setiap entri diawali dengan nama titik akhir proksi terkait. Nama ini dapat berupa nama yang Anda tentukan untuk titik akhir yang ditetapkan pengguna, atau nama khusus `default` untuk titik akhir default proksi yang melakukan permintaan baca/tulis.

Semua metrik Proksi RDS berada di dalam grup `proxy`.

Setiap titik akhir proxy memiliki CloudWatch metriknya sendiri. Anda dapat memantau penggunaan setiap titik akhir proksi secara independen. Untuk informasi selengkapnya tentang titik akhir proksi, lihat Bekerja dengan titik akhir Proksi Amazon RDS.

Anda dapat menggabungkan beberapa nilai untuk setiap metrik menggunakan salah satu kumpulan dimensi berikut. Misalnya, dengan menggunakan kumpulan dimensi `ProxyName`, Anda dapat menganalisis semua lalu lintas untuk proksi tertentu. Dengan menggunakan kumpulan dimensi lainnya, Anda dapat membagi beberapa metrik dengan cara yang berbeda. Anda dapat membagi metrik berdasarkan titik akhir atau basis data target yang berbeda dari setiap proksi, atau lalu lintas baca/tulis dan hanya baca ke setiap basis data.

- Kumpulan dimensi 1: `ProxyName`
- Kumpulan dimensi 2: `ProxyName`, `EndpointName`
- Kumpulan dimensi 3: `ProxyName`, `TargetGroup`, `Target`
- Kumpulan dimensi 4: `ProxyName`, `TargetGroup`, `TargetRole`

Metrik	Deskripsi	Periode yang valid	CloudWatch set dimensi
`AvailabilityPercentage`	Persentase waktu saat grup target tersedia dalam peran yang diindikasikan oleh dimensi. Metrik ini dilaporkan setiap menit. Statistik yang paling berguna untuk metrik ini adalah `Average`.	1 menit	Dimension set 4
`ClientConnections`	Jumlah koneksi klien saat ini. Metrik ini dilaporkan setiap menit. Statistik yang paling berguna untuk metrik ini adalah `Sum`.	1 menit	Dimension set 1, Dimension set 2
`ClientConnectionsClosed`	Jumlah koneksi klien yang ditutup. Statistik yang paling	1 menit ke atas	Dimension set 1, Dimension set 2

Metrik	Deskripsi	Periode yang valid	CloudWatch set dimensi
	berguna untuk metrik ini adalah Sum.		
ClientConnectionsNoTLS	Jumlah koneksi klien saat ini tanpa Keamanan Lapisan Pengangkutan (TLS). Metrik ini dilaporkan setiap menit. Statistik yang paling berguna untuk metrik ini adalah Sum.	1 menit ke atas	Dimension set 1, Dimension set 2
ClientConnectionsReceived	Jumlah permintaan koneksi klien yang diterima. Statistik yang paling berguna untuk metrik ini adalah Sum.	1 menit ke atas	Dimension set 1, Dimension set 2
ClientConnectionsSetupFailedAuth	Jumlah upaya koneksi klien yang gagal karena konfigurasi autentikasi atau TLS yang salah. Statistik yang paling berguna untuk metrik ini adalah Sum.	1 menit ke atas	Dimension set 1, Dimension set 2

Metrik	Deskripsi	Periode yang valid	CloudWatch set dimensi
`ClientConnectionsSetupSucceeded`	Jumlah koneksi klien yang berhasil dibangun dengan mekanisme autentikasi apa pun dengan atau tanpa TLS. Statistik yang paling berguna untuk metrik ini adalah `Sum`.	1 menit ke atas	Dimension set 1, Dimension set 2
`ClientConnectionsTLS`	Jumlah koneksi klien saat ini dengan TLS. Metrik ini dilaporkan setiap menit. Statistik yang paling berguna untuk metrik ini adalah `Sum`.	1 menit ke atas	Dimension set 1, Dimension set 2
`DatabaseConnectionRequests`	Jumlah permintaan untuk membuat koneksi basis data. Statistik yang paling berguna untuk metrik ini adalah `Sum`.	1 menit ke atas	Dimension set 1, Dimension set 3, Dimension set 4
`DatabaseConnectionRequestsWithTLS`	Jumlah permintaan untuk membuat koneksi basis data dengan TLS. Statistik yang paling berguna untuk metrik ini adalah `Sum`.	1 menit ke atas	Dimension set 1, Dimension set 3, Dimension set 4

Metrik	Deskripsi	Periode yang valid	CloudWatch set dimensi
`DatabaseConnections`	Jumlah koneksi basis data saat ini. Metrik ini dilaporkan setiap menit. Statistik yang paling berguna untuk metrik ini adalah `Sum`.	1 menit	Dimension set 1, Dimension set 3, Dimension set 4
`DatabaseConnectionsBorrowLatency`	Waktu dalam mikrodetik yang dibutuhkan bagi proksi yang dipantau untuk mendapatkan koneksi basis data. Statistik yang paling berguna untuk metrik ini adalah `Average`.	1 menit ke atas	Dimension set 1, Dimension set 2
`DatabaseConnectionsCurrentlyBorrowed`	Jumlah koneksi basis data saat ini yang berada dalam status meminjam. Metrik ini dilaporkan setiap menit. Statistik yang paling berguna untuk metrik ini adalah `Sum`.	1 menit	Dimension set 1, Dimension set 3, Dimension set 4
`DatabaseConnectionsCurrentlyInTransaction`	Jumlah koneksi basis data saat ini dalam transaksi. Metrik ini dilaporkan setiap menit. Statistik yang paling berguna untuk metrik ini adalah `Sum`.	1 menit	Dimension set 1, Dimension set 3, Dimension set 4

Metrik	Deskripsi	Periode yang valid	CloudWatch set dimensi
`DatabaseConnectionsCurrentlySessionPinned`	Jumlah koneksi basis data saat ini yang saat ini disematkan karena adanya operasi dalam permintaan klien yang mengubah status sesi. Metrik ini dilaporkan setiap menit. Statistik yang paling berguna untuk metrik ini adalah `Sum`.	1 menit	Dimension set 1, Dimension set 3, Dimension set 4
`DatabaseConnectionsSetupFailed`	Jumlah permintaan koneksi basis data yang gagal. Statistik yang paling berguna untuk metrik ini adalah `Sum`.	1 menit ke atas	Dimension set 1, Dimension set 3, Dimension set 4
`DatabaseConnectionsSetupSucceeded`	Jumlah koneksi basis data yang berhasil dibangun dengan atau tanpa TLS. Statistik yang paling berguna untuk metrik ini adalah `Sum`.	1 menit ke atas	Dimension set 1, Dimension set 3, Dimension set 4

Metrik	Deskripsi	Periode yang valid	CloudWatch set dimensi
`DatabaseConnectionsWithTLS`	Jumlah koneksi basis data saat ini dengan TLS. Metrik ini dilaporkan setiap menit. Statistik yang paling berguna untuk metrik ini adalah `Sum`.	1 menit	Dimension set 1, Dimension set 3, Dimension set 4
`MaxDatabaseConnectionsAllowed`	Jumlah maksimum koneksi basis data yang diperbolehkan. Metrik ini dilaporkan setiap menit. Statistik yang paling berguna untuk metrik ini adalah `Sum`.	1 menit	Dimension set 1, Dimension set 3, Dimension set 4
`QueryDatabaseResponseLatency`	Waktu dalam mikrodetik yang dibutuhkan basis data untuk merespons kueri. Statistik yang paling berguna untuk metrik ini adalah `Average`.	1 menit ke atas	Dimension set 1, Dimension set 2, Dimension set 3, Dimension set 4
`QueryRequests`	Jumlah kueri yang diterima. Kueri yang mencakup beberapa pernyataan yang dihitung sebagai satu kueri. Statistik yang paling berguna untuk metrik ini adalah `Sum`.	1 menit ke atas	Dimension set 1, Dimension set 2

Metrik	Deskripsi	Periode yang valid	CloudWatch set dimensi
`QueryRequestsNoTLS`	Jumlah kueri yang diterima dari koneksi non-TLS. Kueri yang mencakup beberapa pernyataan yang dihitung sebagai satu kueri. Statistik yang paling berguna untuk metrik ini adalah `Sum`.	1 menit ke atas	Dimension set 1, Dimension set 2
`QueryRequestsTLS`	Jumlah kueri yang diterima dari koneksi TLS. Kueri yang mencakup beberapa pernyataan yang dihitung sebagai satu kueri. Statistik yang paling berguna untuk metrik ini adalah `Sum`.	1 menit ke atas	Dimension set 1, Dimension set 2
`QueryResponseLatency`	Waktu dalam mikrodetik antara saat mendapatkan permintaan kueri dan saat proksi meresponsnya. Statistik yang paling berguna untuk metrik ini adalah `Average`.	1 menit ke atas	Dimension set 1, Dimension set 2

Anda dapat menemukan log aktivitas Proxy RDS CloudWatch di bawah. AWS Management Console Setiap proksi memiliki entri di halaman Grup log.

> **Important**
>
> Log ini ditujukan untuk konsumsi manusia untuk tujuan pemecahan masalah dan bukan untuk akses terprogram. Format dan konten log dapat berubah sewaktu-waktu.
> Khususnya, log lama tidak berisi awalan yang menunjukkan titik akhir untuk setiap permintaan. Dalam log yang lebih baru, setiap entri diawali dengan nama titik akhir proksi terkait. Nama ini dapat berupa nama yang Anda tentukan untuk titik akhir yang ditentukan pengguna, atau nama khusus `default` untuk permintaan yang menggunakan titik akhir default proksi.

Bekerja dengan peristiwa Proksi RDS

Peristiwa menunjukkan perubahan dalam lingkungan seperti lingkungan AWS atau layanan atau aplikasi dari mitra perangkat lunak sebagai layanan (SaaS). Atau bisa berupa salah satu aplikasi atau layanan kustom Anda sendiri. Misalnya, Amazon RDS menghasilkan peristiwa saat Anda membuat atau memodifikasi Proksi RDS. Amazon RDS Aurora mengirimkan acara ke CloudWatch Acara dan EventBridge Amazon dalam waktu hampir nyata. Temukan daftar peristiwa Proksi RDS yang dapat dijadikan langganan dan contoh peristiwa Proksi RDS di bawah.

Berikut informasi selengkapnya tentang cara bekerja dengan peristiwa:

- Untuk petunjuk tentang cara melihat peristiwa menggunakan AWS Management Console, AWS CLI, atau API RDS, lihat Melihat peristiwa Amazon RDS.
- Untuk mempelajari cara mengonfigurasi Amazon RDS Aurora untuk mengirim acara EventBridge, lihat. Membuat aturan yang memicu peristiwa Amazon RDS

Peristiwa Proksi RDS

Tabel berikut menunjukkan kategori peristiwa dan daftar peristiwa saat Proksi RDS berupa jenis sumber.

Kategori	ID peristiwa RDS	Pesan	Catatan
perubahan konfigurasi	RDS-EVENT-0204	RDS memodifikasi proksi DB *name*.	

Kategori	ID peristiwa RDS	Pesan	Catatan
perubahan konfigurasi	RDS-EVENT-0207	RDS memodifikasi titik akhir proksi DB *name*.	
perubahan konfigurasi	RDS-EVENT-0213	RDS mendeteksi penambahan instans DB dan secara otomatis menambahkannya ke grup target proksi DB *name*.	
perubahan konfigurasi	RDS-EVENT-0213	RDS mendeteksi pembuatan instans DB *name* dan secara otomatis menambahkannya ke grup target *name* proksi DB *name*.	
perubahan konfigurasi	RDS-EVENT-0214	RDS mendeteksi penghapusan instans DB *name* dan secara otomatis menghapusnya dari grup target *name* proksi DB *name*.	
perubahan konfigurasi	RDS-EVENT-0215	RDS mendeteksi penghapusan klaster DB *name* dan secara otomatis menghapusnya dari grup target *name* proksi DB *name*.	
pembuatan	RDS-EVENT-0203	RDS membuat proksi DB *name*.	
pembuatan	RDS-EVENT-0206	RDS membuat titik akhir *name* untuk proksi DB *name*.	

Kategori	ID peristiwa RDS	Pesan	Catatan
penghapusan	RDS-EVENT-0205	RDS menghapus proksi DB *name*.	
penghapusan	RDS-EVENT-0208	RDS menghapus titik akhir *name* untuk proksi DB *name*.	
kegagalan	RDS-EVENT-0243	RDS gagal menyediak an kapasitas untuk proksi *name* karena tidak ada alamat IP yang cukup yang tersedia di subnet Anda: *name*. Untuk memperbai ki masalah ini, pastikan subnet Anda memiliki jumlah minimum alamat IP yang tidak digunakan seperti yang direkomen dasikan dalam dokumenta si Proksi RDS.	Untuk menentukan jumlah yang direkomendasikan untuk kelas instans Anda, lihat Perencanaan untuk kapasitas alamat IP.
kegagalan	RDS-EVENT-0275	*RDS membatasi beberapa koneksi ke nama proxy DB.* Jumlah permintaan koneksi simultan dari klien ke proxy telah melampaui batas.	

Berikut adalah contoh peristiwa Proksi RDS dalam format JSON. Peristiwa ini menunjukkan bahwa RDS memodifikasi titik akhir bernama `my-endpoint` dari Proksi RDS bernama `my-rds-proxy`. ID peristiwa ini adalah RDS-EVENT-0207.

```
{
  "version": "0",
  "id": "68f6e973-1a0c-d37b-f2f2-94a7f62ffd4e",
```

```
  "detail-type": "RDS DB Proxy Event",
  "source": "aws.rds",
  "account": "123456789012",
  "time": "2018-09-27T22:36:43Z",
  "region": "us-east-1",
  "resources": [
     "arn:aws:rds:us-east-1:123456789012:db-proxy:my-rds-proxy"
  ],
  "detail": {
    "EventCategories": [
      "configuration change"
    ],
    "SourceType": "DB_PROXY",
    "SourceArn": "arn:aws:rds:us-east-1:123456789012:db-proxy:my-rds-proxy",
    "Date": "2018-09-27T22:36:43.292Z",
    "Message": "RDS modified endpoint my-endpoint of DB Proxy my-rds-proxy.",
    "SourceIdentifier": "my-endpoint",
    "EventID": "RDS-EVENT-0207"
  }
}
```

Contoh baris perintah Proksi RDS

Untuk melihat cara kombinasi perintah koneksi dan pernyataan SQL berinteraksi dengan Proksi RDS, perhatikan contoh berikut.

Contoh

- Preserving Connections to a MySQL Database Across a Failover
- Adjusting the max_connections Setting for an Aurora DB Cluster

Example Mencadangkan koneksi ke basis data MySQL di seluruh failover

Contoh MySQL ini menunjukkan bagaimana koneksi terbuka terus berfungsi selama failover. Contohnya adalah ketika Anda me-reboot basis data atau basis data tidak tersedia karena terjadi masalah. Contoh ini menggunakan proksi bernama `the-proxy` dan klaster DB Aurora dengan instans DB `instance-8898` dan `instance-9814`. Saat Anda menjalankan perintah `failover-db-cluster` dari baris perintah Linux, instans penulis yang terhubung ke proksi berubah ke instans DB yang berbeda. Anda dapat melihat bahwa instans DB yang terkait dengan proksi ini berubah saat koneksi masih terbuka.

```
$ mysql -h the-proxy.proxy-demo.us-east-1.rds.amazonaws.com -u admin_user -p
Enter password:
...

mysql> select @@aurora_server_id;
+--------------------+
| @@aurora_server_id |
+--------------------+
| instance-9814      |
+--------------------+
1 row in set (0.01 sec)

mysql>
[1]+  Stopped                 mysql -h the-proxy.proxy-demo.us-east-1.rds.amazonaws.com
 -u admin_user -p
$ # Initially, instance-9814 is the writer.
$ aws rds failover-db-cluster --db-cluster-identifier cluster-56-2019-11-14-1399
JSON output
$ # After a short time, the console shows that the failover operation is complete.
$ # Now instance-8898 is the writer.
$ fg
mysql -h the-proxy.proxy-demo.us.us-east-1.rds.amazonaws.com -u admin_user -p

mysql> select @@aurora_server_id;
+--------------------+
| @@aurora_server_id |
+--------------------+
| instance-8898      |
+--------------------+
1 row in set (0.01 sec)

mysql>
[1]+  Stopped                 mysql -h the-proxy.proxy-demo.us-east-1.rds.amazonaws.com
 -u admin_user -p
$ aws rds failover-db-cluster --db-cluster-identifier cluster-56-2019-11-14-1399
JSON output
$ # After a short time, the console shows that the failover operation is complete.
$ # Now instance-9814 is the writer again.
$ fg
mysql -h the-proxy.proxy-demo.us-east-1.rds.amazonaws.com -u admin_user -p

mysql> select @@aurora_server_id;
+--------------------+
```

```
| @@aurora_server_id |
+--------------------+
| instance-9814      |
+--------------------+
1 row in set (0.01 sec)
+---------------+---------------+
| Variable_name | Value         |
+---------------+---------------+
| hostname      | ip-10-1-3-178 |
+---------------+---------------+
1 row in set (0.02 sec)
```

Example Menyesuaikan pengaturan max_connections untuk klaster DB Aurora

Contoh ini menunjukkan cara menyesuaikan pengaturan `max_connections` untuk klaster DB Aurora MySQL. Untuk melakukannya, buat grup parameter klaster DB Anda sendiri berdasarkan pengaturan parameter default untuk klaster yang kompatibel dengan MySQL 5.7. Tentukan nilai untuk pengaturan `max_connections`, dengan mengganti formula yang menetapkan nilai default. Kaitkan grup parameter klaster DB dengan klaster DB Anda.

```
export REGION=us-east-1
export CLUSTER_PARAM_GROUP=rds-proxy-mysql-57-max-connections-demo
export CLUSTER_NAME=rds-proxy-mysql-57

aws rds create-db-parameter-group --region $REGION \
  --db-parameter-group-family aurora-mysql5.7 \
  --db-parameter-group-name $CLUSTER_PARAM_GROUP \
  --description "Aurora MySQL 5.7 cluster parameter group for RDS Proxy demo."

aws rds modify-db-cluster --region $REGION \
  --db-cluster-identifier $CLUSTER_NAME \
  --db-cluster-parameter-group-name $CLUSTER_PARAM_GROUP

echo "New cluster param group is assigned to cluster:"
aws rds describe-db-clusters --region $REGION \
  --db-cluster-identifier $CLUSTER_NAME \
  --query '*[*].{DBClusterParameterGroup:DBClusterParameterGroup}'

echo "Current value for max_connections:"
aws rds describe-db-cluster-parameters --region $REGION \
  --db-cluster-parameter-group-name $CLUSTER_PARAM_GROUP \
  --query '*[*].{ParameterName:ParameterName,ParameterValue:ParameterValue}' \
  --output text | grep "^max_connections"
```

```
echo -n "Enter number for max_connections setting: "
read answer

aws rds modify-db-cluster-parameter-group --region $REGION --db-cluster-parameter-
group-name $CLUSTER_PARAM_GROUP \
  --parameters "ParameterName=max_connections,ParameterValue=$
$answer,ApplyMethod=immediate"

echo "Updated value for max_connections:"
aws rds describe-db-cluster-parameters --region $REGION \
  --db-cluster-parameter-group-name $CLUSTER_PARAM_GROUP \
  --query '*[*].{ParameterName:ParameterName,ParameterValue:ParameterValue}' \
  --output text | grep "^max_connections"
```

Pemecahan masalah untuk Proksi RDS

Setelah itu, Anda dapat menemukan ide pemecahan masalah untuk beberapa masalah Proxy RDS umum dan informasi tentang CloudWatch log untuk RDS Proxy.

Dalam log Proksi RDS, setiap entri diawali dengan nama titik akhir proksi terkait. Nama ini bisa berupa nama yang Anda tentukan untuk titik akhir yang ditentukan pengguna. Atau, bisa menjadi nama khusus `default` untuk titik akhir default proksi yang melakukan permintaan baca/tulis. Untuk informasi selengkapnya tentang titik akhir proksi, lihat Bekerja dengan titik akhir Proksi Amazon RDS.

Topik

- Memverifikasi konektivitas untuk proksi
- Masalah dan solusi umum

Memverifikasi konektivitas untuk proksi

Anda dapat menggunakan perintah berikut untuk memverifikasi bahwa semua komponen seperti proksi, basis data, dan instans komputasi dalam koneksi dapat berkomunikasi satu sama lain.

Periksa proxy itu sendiri menggunakan describe-db-proxiesperintah. Periksa juga kelompok target terkait menggunakan perintah describe-db-proxy-target-groups. Periksa apakah detail target cocok dengan instans DB RDS yang ingin dikaitkan dengan proksi. Gunakan perintah seperti berikut ini.

```
aws rds describe-db-proxies --db-proxy-name $DB_PROXY_NAME
```

```
aws rds describe-db-proxy-target-groups --db-proxy-name $DB_PROXY_NAME
```

Untuk mengonfirmasi bahwa proxy dapat terhubung ke database yang mendasarinya, periksa target yang ditentukan dalam grup target menggunakan describe-db-proxy-targetsperintah. Gunakan perintah seperti berikut.

```
aws rds describe-db-proxy-targets --db-proxy-name $DB_PROXY_NAME
```

Output dari describe-db-proxy-targetsperintah termasuk `TargetHealth` bidang. Anda dapat memeriksa kolom `State`, `Reason`, dan `Description` di dalam `TargetHealth` untuk memeriksa apakah proksi dapat berkomunikasi dengan instans DB acuan.

- Nilai `State` dari `AVAILABLE` mengindikasikan bahwa proksi dapat terhubung ke instans DB.
- Nilai `State` dari `UNAVAILABLE` mengindikasikan masalah koneksi sementara atau permanen. Dalam kasus ini, periksa kolom `Reason` dan `Description`. Misalnya, jika `Reason` memiliki nilai `PENDING_PROXY_CAPACITY`, coba hubungkan lagi setelah proksi menyelesaikan operasi penskalaan. Jika `Reason` memiliki nilai `UNREACHABLE`, `CONNECTION_FAILED`, atau `AUTH_FAILURE`, gunakan penjelasan dari kolom `Description` untuk membantu Anda mendiagnosis masalah.
- Kolom `State` muungkin memiliki nilai `REGISTERING` dalam waktu yang singkat sebelum berubah ke `AVAILABLE` atau `UNAVAILABLE`.

Jika perintah Netcat berikut (`nc`) berhasil, Anda dapat mengakses titik akhir proksi dari instans EC2 atau sistem lain tempat Anda masuk. Perintah ini melaporkan kegagalan jika Anda tidak berada di VPC yang sama dengan proksi dan basis data terkait. Anda mungkin dapat masuk secara langsung ke basis data tanpa berada di VPC yang sama. Akan tetapi, Anda tidak dapat masuk ke proksi kecuali Anda berada di VPC yang sama.

```
nc -zx MySQL_proxy_endpoint 3306

nc -zx PostgreSQL_proxy_endpoint 5432
```

Anda dapat menggunakan perintah berikut untuk memastikan bahwa instans EC2 Anda memiliki properti yang dibutuhkan. Khususnya, VPC untuk instans EC2 harus sama dengan VPC untuk yang terhubung dengan proksi.

```
aws ec2 describe-instances --instance-ids your_ec2_instance_id
```

Periksa rahasia Secrets Manager yang digunakan oleh proksi.

```
aws secretsmanager list-secrets
aws secretsmanager get-secret-value --secret-id your_secret_id
```

Pastikan kolom `SecretString` yang ditampilkan oleh `get-secret-value` dienkode sebagai string JSON yang mencakup kolom `username` dan `password`. Contoh berikut menunjukkan format kolom `SecretString`.

```
{
  "ARN": "some_arn",
  "Name": "some_name",
  "VersionId": "some_version_id",
  "SecretString": '{"username":"some_username","password":"some_password"}',
  "VersionStages": [ "some_stage" ],
  "CreatedDate": some_timestamp
}
```

Masalah dan solusi umum

Bagian ini menjelaskan beberapa masalah umum dan solusi potensial saat menggunakan Proksi RDS.

Setelah menjalankan perintah `aws rds describe-db-proxy-targets` CLI, jika deskripsi `TargetHealth` menyatakan `Proxy does not have any registered credentials`, verifikasi hal-hal berikut:

- Ada kredensial yang terdaftar bagi pengguna untuk mengakses proksi.
- Peran IAM untuk mengakses rahasia Secrets Manager yang digunakan oleh proksi valid.

Anda mungkin mengalami peristiwa RDS berikut saat membuat atau terhubung ke proksi DB.

Kategori	ID peristiwa RDS	Deskripsi
kegagalan	RDS-EVENT-0243	RDS gagal menyediakan kapasitas untuk proksi karena tidak ada alamat IP yang cukup yang tersedia di subnet Anda. Untuk memperbai

Kategori	ID peristiwa RDS	Deskripsi
		ki masalah ini, pastikan subnet Anda memiliki jumlah minimum alamat IP yang tidak digunakan. Untuk menentuka n jumlah yang direkomen dasikan untuk kelas instans Anda, lihat Perencanaan untuk kapasitas alamat IP.
kegagalan	RDS-EVENT-0275	*RDS membatasi beberapa koneksi ke nama proxy DB.* Jumlah permintaan koneksi simultan dari klien ke proxy telah melampaui batas.

Anda mungkin mengalami masalah berikut saat membuat proksi baru atau terhubung ke proksi.

Kesalahan	Penyebab atau solusi
`403: The security token included in the request is invalid`	Pilih peran IAM yang sudah ada, bukan memilih untuk membuat peran baru.

Anda mungkin mengalami masalah berikut saat terhubung ke proksi MySQL.

Kesalahan	Penyebab atau solusi
`ERROR 1040 (HY000): Connections rate limit`	Tingkat permintaan koneksi dari klien ke proksi telah melampaui batas.

Kesalahan	Penyebab atau solusi
`exceeded (limit_value )`	
`ERROR 1040 (HY000): IAM authentication rate limit exceeded`	Jumlah permintaan simultan dengan autentikasi IAM dari klien ke proksi telah melampaui batas.
`ERROR 1040 (HY000): Number simultaneous connections exceeded (limit_value )`	Jumlah permintaan koneksi simultan dari klien ke proksi telah melampaui batas.
`ERROR 1045 (28000): Access denied for user 'DB_USER'@'%' (using password: YES)`	Rahasia Secrets Manager yang digunakan oleh proksi tidak cocok dengan nama pengguna dan kata sandi dari pengguna basis data yang sudah ada. Perbarui kredensial dalam rahasia Secrets Manager, atau pastikan pengguna basis data ada dan memiliki kata sandi yang sama dengan yang ada dalam rahasia.
`ERROR 1105 (HY000): Unknown error`	Terjadi kesalahan yang tidak diketahui.
`ERROR 1231 (42000): Variable ''character_set_client'' can't be set to the value of value`	Nilai yang diatur untuk parameter `character_set_client` tidak valid. Misalnya, nilai `ucs2` tidak valid karena dapat merusak server MySQL Anda.

Kesalahan	Penyebab atau solusi
`ERROR 3159 (HY000): This RDS Proxy requires TLS connections.`	Anda mengaktifkan pengaturan Wajibkan Keamanan Lapisan Pengangkutan dalam proksi, tetapi koneksi Anda menyertakan parameter `ssl-mode=DISABLED` dalam klien MySQL. Lakukan salah satu dari langkah berikut: • Nonaktifkan pengaturan Wajibkan Keamanan Lapisan Pengangkutan untuk proksi. • Hubungkan ke basis data menggunakan pengaturan minimum `ssl-mode=REQUIRED` dalam klien MySQL.
`ERROR 2026 (HY000): SSL connection error: Internal Server Error`	Proses jabat tangan TLS dengan proksi gagal. Beberapa kemungkinan alasannya meliputi: • SSL dibutuhkan, tetapi server tidak mendukungnya. • Terjadi kesalahan server internal. • Terjadi jabat tangan yang buruk.
`ERROR 9501 (HY000): Timed-out waiting to acquire database connection`	Waktu tunggu proksi untuk memperoleh koneksi basis data habis. Beberapa kemungkinan alasannya meliputi: • Proksi tidak dapat membangun koneksi basis data karena koneksi maksimum sudah tercapai • Proksi tidak dapat membangun koneksi basis data karena basis data tidak tersedia.

Anda mungkin mengalami masalah berikut saat terhubung ke proksi PostgreSQL.

Kesalahan	Penyebab	Solusi
`IAM authentication is allowed only with SSL connections.`	Pengguna mencoba terhubung ke basis data menggunakan autentikasi IAM dengan pengaturan `sslmode=disable` dalam klien PostgreSQL.	Pengguna harus terhubung ke basis data menggunakan pengaturan minimum `sslmode=require` dalam klien PostgreSQL. Untuk informasi lebih lanjut, lihat

Kesalahan	Penyebab	Solusi
		dokumentasi Dukungan PostgreSQL SSL.
`This RDS Proxy requires TLS connections.`	Pengguna mengaktifkan opsi Wajibkan Keamanan Lapisan Pengangkutan, tetapi mencoba untuk terhubung dengan `sslmode=disable` dalam klien PostgreSQL.	Untuk memperbaiki kesalahan ini, lakukan salah satu tindakan berikut: • Nonaktifkan opsi Wajibkan Keamanan Lapisan Pengangkutan proksi. • Hubungkan ke basis data menggunakan pengaturan minimum `sslmode=allow` dalam klien PostgreSQL.
`IAM authentication failed for user` *`user_name`* `. Check the IAM token for this user and try again.`	Kesalahan ini mungkin terjadi karena alasan berikut: • Klien memberikan nama pengguna IAM yang salah. • Klien memberikan token otorisasi IAM yang salah untuk pengguna. • Klien menggunakan kebijakan IAM yang tidak memiliki izin yang dibutuhkan. • Klien memberikan token otorisasi IAM yang kedaluwarsa untuk pengguna.	Untuk memperbaiki kesalahan ini, lakukan tindakan berikut: 1. Pastikan pengguna IAM yang disediakan sudah ada. 2. Pastikan token otorisasi IAM adalah milik pengguna IAM yang disediakan. 3. Pastikan kebijakan IAM memiliki izin yang memadai untuk RDS. 4. Periksa validitas token otorisasi IAM yang digunakan.

Kesalahan	Penyebab	Solusi
`This RDS proxy has no credentials for the role role_name . Check the credentials for this role and try again.`	Tidak ada rahasia Secrets Manager untuk peran ini.	Tambahkan rahasia Secrets Manager untuk peran ini. Untuk informasi selengkapnya, lihat Menyiapkan AWS Identity and Access Management kebijakan (IAM).
`RDS supports only IAM, MD5, or SCRAM authentication.`	Klien basis data yang digunakan untuk terhubung ke proksi menggunakan mekanisme autentikasi yang saat ini tidak didukung oleh proksi.	Jika Anda tidak menggunakan autentikasi IAM, gunakan autentikasi kata sandi MD5 atau SCRAM.
`A user name is missing from the connection startup packet. Provide a user name for this connection.`	Klien basis data yang digunakan untuk terhubung ke proksi tidak mengirimkan nama pengguna saat mencoba membangun koneksi.	Pastikan untuk menentukan sebuah nama pengguna saat menyiapkan koneksi ke proksi menggunakan klien PostgreSQL pilihan Anda.
`Feature not supported: RDS Proxy supports only version 3.0 of the PostgreSQL messaging protocol.`	Klien PostgreSQL yang digunakan untuk terhubung ke proksi menggunakan protokol yang lebih lama dari 3.0.	Gunakan klien PostgreSQL yang lebih baru yang mendukung protokol olah pesan 3.0. Jika Anda menggunakan CLI `psql` PostgreSQL, gunakan versi yang lebih baru atau yang setara dengan 7.4.

Kesalahan	Penyebab	Solusi
`Feature not supported: RDS Proxy currently doesn't support streaming replication mode.`	Klien PostgreSQL yang digunakan untuk terhubung ke proksi mencoba menggunakan mode replikasi streaming, yang saat ini tidak didukung oleh Proksi RDS.	Nonaktifkan mode replikasi streaming dalam klien PostgreSQL yang digunakan untuk menghubungkan.
`Feature not supported: RDS Proxy currently doesn't support the option` *`option_name`* `.`	Melalui pesan startup, klien PostgreSQL yang digunakan untuk terhubung ke proksi meminta opsi yang saat ini tidak didukung oleh Proksi RDS.	Nonaktifkan opsi yang ditampilkan sebagai tidak didukung dari pesan di atas dalam klien PostgreSQL yang digunakan untuk menghubungkan.
`The IAM authentication failed because of too many competing requests.`	Jumlah permintaan simultan dengan autentikasi IAM dari klien ke proksi telah melampaui batas.	Kurangi tingkat pembangunan koneksi menggunakan autentikasi IAM dari klien PostgreSQL.
`The maximum number of client connections to the proxy exceeded` *`number_value`* `.`	Jumlah permintaan koneksi simultan dari klien ke proksi telah melampaui batas.	Kurangi jumlah koneksi aktif dari klien PostgreSQL ke proksi RDS ini.
`Rate of connection to proxy exceeded` *`number_value`* `.`	Tingkat permintaan koneksi dari klien ke proksi telah melampaui batas.	Kurangi tingkat pembangunan koneksi dari klien PostgreSQL.
`The password that was provided for the role` *`role_name`* `is wrong.`	Kata sandi untuk peran ini tidak cocok dengan rahasia Secrets Manager.	Periksa rahasia peran ini dalam Secrets Manager untuk melihat apakah kata sandi sama dengan yang digunakan dalam klien PostgreSQL Anda.

Kesalahan	Penyebab	Solusi
`The IAM authentication failed for the role role_name . Check the IAM token for this role and try again.`	Terjadi masalah dengan token IAM yang digunakan untuk autentikasi IAM.	Buat token autentikasi baru dan gunakan dalam koneksi baru.
`IAM is allowed only with SSL connections.`	Klien mencoba untuk terhubung menggunakan autentikasi IAM, tetapi SSL tidak diaktifkan.	Aktifkan SSL dalam klien PostgreSQL.
`Unknown error.`	Terjadi kesalahan yang tidak diketahui.	Hubungi Dukungan AWS untuk menyelidiki masalah ini.
`Timed-out waiting to acquire database connection.`	Waktu tunggu proksi untuk memperoleh koneksi basis data habis. Beberapa kemungkinan alasannya meliputi: • Proksi tidak dapat membangun koneksi basis data karena koneksi maksimum sudah tercapai. • Proksi tidak dapat membangun koneksi basis data karena basis data tidak tersedia.	Kemungkinan solusinya sebagai berikut: • Periksa target untuk melihat apakah basis data tidak tersedia. • Periksa apakah ada eksekusi transaksi dan/atau kueri yang berjalan lama. Transaksi atau kueri ini bisa menggunakan koneksi basis data dari kumpulan koneksi dalam waktu yang lama.

Kesalahan	Penyebab	Solusi
`Request returned an error: database_error .`	Koneksi basis data yang dibangun dari proksi menampilkan kesalahan.	Solusinya bergantung dari kesalahan basis data tertentu. Salah satu contohnya adalah: `Request returned an error: database "your-database-name" does not exist`. Ini berarti bahwa nama basis data yang ditentukan tidak ada di server basis data. Atau, nama pengguna yang digunakan sebagai sebuah nama basis data (jika nama basis data tidak ditentukan) tidak ada di server.

Penggunaan Proksi RDS dengan AWS CloudFormation

Anda dapat menggunakan Proksi RDS dengan AWS CloudFormation. Hal ini membantu Anda membuat grup sumber daya terkait. Grup semacam ini bisa mencakup proksi yang dapat terhubung ke instans DB Amazon RDS. Dukungan Proksi RDS dalam AWS CloudFormation melibatkan dua jenis registri baru: `DBProxy` dan `DBProxyTargetGroup`.

Daftar berikut menunjukkan contoh templat AWS CloudFormation untuk Proksi RDS.

```
Resources:
 DBProxy:
   Type: AWS::RDS::DBProxy
   Properties:
     DBProxyName: CanaryProxy
     EngineFamily: MYSQL
     RoleArn:
      Fn::ImportValue: SecretReaderRoleArn
     Auth:
       - {AuthScheme: SECRETS, SecretArn: !ImportValue ProxySecret, IAMAuth: DISABLED}
     VpcSubnetIds:
```

```
      Fn::Split: [",", "Fn::ImportValue": SubnetIds]

ProxyTargetGroup:
  Type: AWS::RDS::DBProxyTargetGroup
  Properties:
    DBProxyName: CanaryProxy
    TargetGroupName: default
    DBInstanceIdentifiers:
      - Fn::ImportValue: DBInstanceName
  DependsOn: DBProxy
```

Untuk informasi selengkapnya tentang sumber daya dalam sampel ini, lihat DBProxydan DBProxyTargetGroup.

Untuk informasi selengkapnya tentang sumber daya yang dapat dibuat menggunakan AWS CloudFormation, lihat Referensi jenis sumber daya RDS.

Menggunakan integrasi nol-ETL Amazon RDS dengan Amazon Redshift (pratinjau)

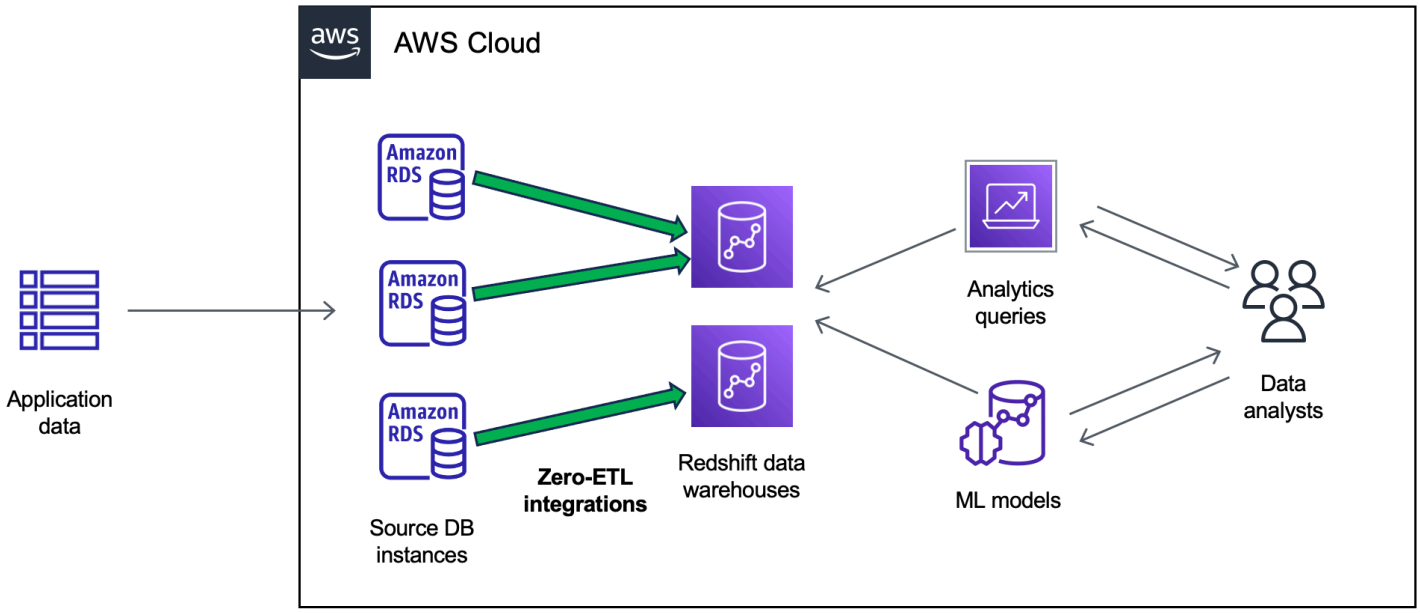
Ini adalah dokumentasi prarilis untuk integrasi nol-ETL Amazon RDS dengan Amazon Redshift, yang sedang dalam rilis pratinjau. Dokumentasi dan fitur dapat berubah. Sebaiknya gunakan fitur ini hanya dalam lingkungan pengujian, bukan dalam lingkungan produksi. Untuk syarat dan ketentuan pratinjau, lihat Beta dan Pratinjau dalam Persyaratan Layanan AWS.

Integrasi nol-ETL Amazon RDS dengan Amazon Redshift memungkinkan analisis hampir waktu nyata dan machine learning (ML) menggunakan Amazon Redshift pada data transaksional berukuran petabyte dari RDS. Extract, transform, and load (ETL) adalah proses menggabungkan data dari berbagai sumber ke dalam repositori pusat yang besar.

Integrasi nol-ETL membuat data dalam basis data RDS Anda Kluster Aurora waktu dekat. Setelah data tersebut berada di Amazon Redshift, Anda dapat memberi daya pada beban kerja analitik, ML, dan AI menggunakan kemampuan bawaan Amazon Redshift, seperti pembelajaran mesin, tampilan terwujud, berbagi data, akses gabungan ke beberapa penyimpanan data dan data lake, serta integrasi dengan Amazon, Amazon, dan lainnya. SageMaker QuickSight Layanan AWS

Untuk membuat integrasi Nol-ETL, Anda menentukan instans DB AZ tunggal atau multi-AZ cluster Aurora DB sumbernya, dan gudang data Amazon Redshift sebagai target. Integrasi ini mereplikasi data dari basis data sumber ke gudang data target.

Diagram berikut menggambarkan fungsi ini:



Integrasi memantau kondisi pipeline data dan memulihkan dari masalah jika memungkinkan. Anda dapat membuat integrasi dari beberapa basis data RDS klaster ke dalam satu namespace Amazon Redshift, memungkinkan Anda memperoleh wawasan di beberapa aplikasi.

Topik

Manfaat

Integrasi nol-ETL RDS dengan Amazon Redshift memiliki manfaat berikut:

- Membantu Anda memperoleh wawasan menyeluruh dari berbagai sumber data.
- Menghilangkan kebutuhan untuk membangun dan memelihara pipeline data yang kompleks yang melakukan operasi extract, transform, and load (ETL). Integrasi nol-ETL menghilangkan tantangan yang muncul dalam membangun dan mengelola pipeline dengan menyediakan dan mengelolanya untuk Anda.
- Mengurangi beban dan biaya operasional, serta membantu Anda fokus pada peningkatan aplikasi Anda.
- Memungkinkan Anda memanfaatkan analitik Amazon Redshift dan kemampuan ML untuk memperoleh wawasan dari data transaksional dan data lainnya, guna merespons secara efektif peristiwa kritis dan sensitif terhadap waktu.

Konsep utama

Saat mulai menggunakan integrasi nol-ETL, pertimbangkan konsep berikut ini:

Integrasi

Pipa data yang dikelola sepenuhnya yang secara otomatis mereplikasi data dan skema transaksional dari database RDS Aurora gudang data Amazon Redshift.

Database sumber

Basis data RDS cluster tempat data direplikasi. Anda dapat menentukan instans DB AZ Tunggal atau Multi-AZ. Beberapa basis data sumber dapat menulis ke target yang sama. Ada beberapa batasan pengaturan untuk basis data sumber, yang diuraikan dalamthe section called “Batasan pratinjau”.

Gudang data target

Gudang data Amazon Redshift tempat tujuan data direplikasi. Ada dua jenis gudang data: gudang data klaster terprovisi dan gudang data nirserver. Gudang data klaster terprovisi adalah kumpulan sumber daya komputasi yang disebut simpul, yang diatur ke dalam grup yang disebut klaster. Gudang data nirserver terdiri dari grup kerja yang menyimpan sumber daya komputasi, serta ruang nama yang menampung objek basis data dan pengguna. Kedua gudang data ini menjalankan mesin Amazon Redshift dan berisi satu atau beberapa basis data.

Untuk informasi selengkapnya, lihat Arsitektur sistem gudang data dalam Panduan Developer Amazon Redshift.

Batasan pratinjau

Batasan berikut berlaku pada integrasi nol-ETL RDS dengan Amazon Redshift.

Topik

Batasan umum

- basis data sumber harus berada di Wilayah yang sama dengan gudang data Amazon Redshift target.
- Anda tidak dapat mengganti nama jika memiliki integrasi yang ada.
- Anda tidak dapat menghapus database yang memiliki integrasi yang ada. Anda harus menghapus semua integrasi yang terkait terlebih dahulu.
- Jika Anda menghentikan basis data sumber, beberapa transaksi terakhir mungkin tidak direplikasi ke gudang data target sampai Anda melanjutkan database.
- Anda tidak dapat menghapus integrasi jika database sumber dihentikan.
- Amazon RDS hanya mendukung deployment instans DB AZ Tunggal dan Multi-AZ sebagai sumber integrasi. Amazon RDS saat ini tidak mendukung klaster DB Multi-AZ.
- Integrasi nol-ETL saat ini tidak mendukung pemfilteran data.
- Jika database Anda adalah sumber penerapan biru/hijau, lingkungan biru dan hijau tidak dapat memiliki integrasi nol-ETL selama peralihan. Anda harus menghapus integrasi tersebut terlebih dahulu dan beralih, lalu membuat ulang integrasi.
- Saat Anda pertama kali membuat integrasi, atau ketika tabel sedang disinkronkan ulang, seeding data dari sumber ke target dapat memakan waktu 20-25 menit atau lebih tergantung ukuran basis data sumber. Penundaan ini dapat menyebabkan peningkatan lag replika.
- Beberapa jenis data tidak didukung. Untuk daftar jenis data yang didukung, lihat the section called “Perbedaan jenis data”.
- Referensi kunci asing dengan pembaruan tabel yang telah ditentukan sebelumnya tidak didukung. Secara khusus, `ON DELETE` dan `ON UPDATE` aturan tidak didukung dengan`CASCADE`,`SET NULL`, dan `SET DEFAULT` tindakan. Mencoba membuat atau memperbarui tabel dengan referensi tersebut ke tabel lain akan menempatkan tabel ke dalam keadaan gagal.

- Transaksi XA tidak didukung.
- Pengidentifikasi objek (termasuk nama basis data, nama tabel, nama kolom, dan lainnya) hanya dapat berisi karakter alfanumerik, angka, $, dan _ (garis bawah).

Batasan RDS for MySQL

- Database sumber Anda harus menjalankan RDS untuk MySQL versi 8.0.28 atau lebih tinggi.
- Integrasi nol-ETL mengandalkan pencatatan log biner MySQL (binlog) untuk mengambil perubahan data yang sedang berlangsung. Sebaiknya jangan gunakan pemfilteran data berbasis binlog, karena hal ini dapat menyebabkan inkonsistensi data antara basis data sumber dan target.
- Tabel sistem, tabel sementara, dan tampilan RDS for MySQL tidak direplikasi ke Amazon Redshift.
- Integrasi nol-ETL didukung hanya untuk basis data yang dikonfigurasi untuk menggunakan mesin penyimpanan InnoDB.
- Klaster DB sumber tidak dapat dikonfigurasi dengan Otoritas Sertifikat (CA) `rds-ca-ecc384-g1`.
- Operasi partisi `ALTER TABLE` menyebabkan tabel Anda melakukan sinkronisasi ulang untuk memuat ulang data dari RDS ke Amazon Redshift. Tabel tidak akan tersedia untuk kueri saat disinkronkan ulang.

Batasan Amazon Redshift

Untuk mengetahui daftar batasan Amazon Redshift yang terkait dengan integrasi nol-ETL, lihat Pertimbangan dalam Panduan Manajemen Amazon Redshift.

Kuota

Akun Anda memiliki kuota berikut yang terkait dengan integrasi nol-ETL RDS dengan Amazon Redshift. Kecuali ditentukan lain, masing-masing kuota ditentukan untuk setiap Wilayah.

Nama	Default	Deskripsi
Integrasi	100	Jumlah total integrasi dalam sebuah Akun AWS.
Integrasi per gudang data target	50	Jumlah integrasi yang mengirim data ke satu gudang data Amazon Redshift target.

Nama	Default	Deskripsi
Integrasi per instans sumber	1	Jumlah integrasi yang mengirimkan data dari instans DB sumber tunggal.

Selain itu, Amazon Redshift menempatkan batasan tertentu pada jumlah tabel yang diizinkan di setiap instans DB atau simpul klaster. Untuk informasi selengkapnya, lihat Kuota dan batasan di Amazon Redshift dalam Panduan Manajemen Amazon Redshift.

Wilayah yang Didukung

Integrasi RDS Zero-ETL dengan Amazon Redshift tersedia dalam subset. Wilayah AWS Untuk mengetahui daftar Wilayah yang didukung, lihat the section called “Integrasi nol-ETL ”.

Mulai menggunakan integrasi nol-ETL Amazon RDS dengan Amazon Redshift

> Ini adalah dokumentasi prarilis untuk integrasi nol-ETL Amazon RDS dengan Amazon Redshift, yang sedang dalam rilis pratinjau. Dokumentasi dan fitur dapat berubah. Sebaiknya gunakan fitur ini hanya dalam lingkungan pengujian, bukan dalam lingkungan produksi. Untuk syarat dan ketentuan pratinjau, lihat Beta dan Pratinjau dalam Persyaratan Layanan AWS.

Sebelum Anda membuat integrasi nol-ETL dengan Amazon Redshift, konfigurasikan database RDS Kluster Aurora izin yang diperlukan. Selama pengaturan, Anda akan menyelesaikan langkah-langkah berikut:

1. Buat grup parameter DB kustom.
2. Buat basis data sumber .
3. Buat gudang data Amazon Redshift target.

Setelah Anda menyelesaikan tugas-tugas ini, lanjutkan ke the section called “Membuat integrasi nol-ETL”.

Langkah 1: Buat grup parameter DB kustom

Integrasi nol-ETL Amazon RDS dengan Amazon Redshift memerlukan nilai spesifik untuk parameter DB yang mengontrol pencatatan log biner (binlog). Untuk mengonfigurasi logging biner, Anda harus terlebih dahulu membuat grup parameter DB kustom, dan kemudian mengaitkannya dengan database sumber.

Buat grup parameter DB kustom dengan pengaturan berikut . Untuk petunjuk cara membuat grup parameter, lihat the section called “Bekerja dengan grup parameter DB”.

- `binlog_format=ROW`
- `binlog_row_image=full`
- `binlog_checksum=NONE`

Selain itu, pastikan parameter `binlog_row_value_options` tidak diatur ke `PARTIAL_JSON`.

Langkah 2: Buat basis data sumber

database ini akan menjadi sumber replikasi data ke Amazon Redshift.

database harus menjalankan RDS untuk MySQL versi 8.0.28 atau lebih tinggi Aurora MySQL versi 3.05 (kompatibel dengan MySQL 8.0.32) atau lebih tinggi. Untuk petunjuk cara membuat . the section called “Membuat instans DB”

Di bagian Konfigurasi tambahan, ubah grup parameter DB default ke grup parameter kustom yang Anda buat pada langkah sebelumnya.

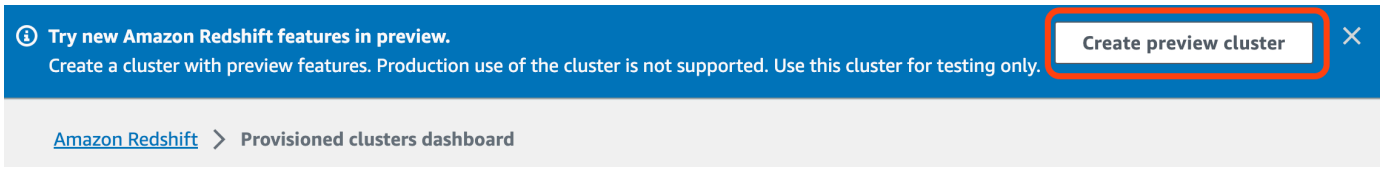
> **Note**
>
> Untuk petunjuk, lihat the section called “Mem-boot ulang instans DB”.

Selain itu, pastikan bahwa backup otomatis diaktifkan pada database. Untuk informasi selengkapnya, lihat the section called “Mengaktifkan pencadangan otomatis”.

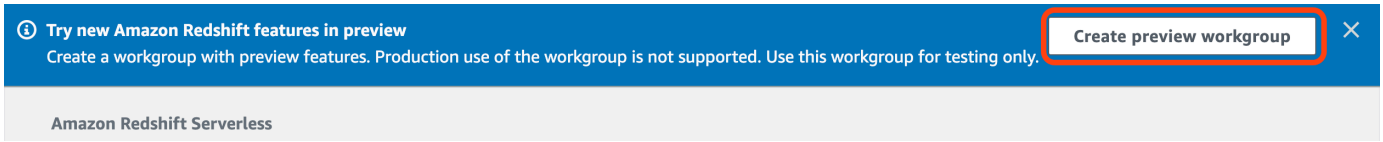
Langkah 3: Buat gudang data Amazon Redshift

Setelah membuat basis data sumber, Anda harus membuat dan mengonfigurasi gudang data target di Amazon Redshift. Gudang data harus memenuhi persyaratan berikut:

- Dibuat dalam pratinjau
  - Untuk membuat klaster terprovisi dalam pratinjau, pilih Buat klaster pratinjau dari banner di dasbor klaster terprovisi. Untuk informasi selengkapnya, lihat Membuat klaster pratinjau.



    Saat membuat klaster, tetapkan Jalur pratinjau ke `preview_2023`.
  - Untuk membuat grup kerja Redshift Nirserver dalam pratinjau, pilih Buat grup kerja pratinjau dari banner di dasbor Nirserver. Untuk informasi selengkapnya, lihat Membuat grup kerja pratinjau.



- Menggunakan jenis simpul RA3 (`ra3.16xlarge`, `ra3.4xlarge`, atau `ra3.xlplus`) dengan setidaknya dua node, atau Redshift Nirserver
- Terenkripsi (jika menggunakan klaster yang disediakan). Untuk informasi selengkapnya, lihat Enkripsi basis data Amazon Redshift.

Untuk petunjuk cara membuat gudang data, lihat Membuat klaster untuk klaster terprovisi, atau Membuat grup kerja dengan ruang nama untuk Redshift Nirserver.

Aktifkan kepekaan huruf besar/kecil di gudang data

Agar integrasi berhasil, parameter kepekaan huruf besar/kecil (`enable_case_sensitive_identifier`) harus diaktifkan untuk gudang data. Secara default, kepekaan huruf besar/kecil dinonaktifkan di semua klaster terprovisi dan grup kerja Redshift Nirserver.

Untuk mengaktifkan kepekaan huruf besar/kecil, lakukan langkah-langkah berikut bergantung pada jenis gudang data Anda:

- Klaster terprovisi – Untuk mengaktifkan kepekaan huruf besar/kecil pada klaster terprovisi, buat grup parameter kustom dengan parameter `enable_case_sensitive_identifier` diaktifkan. Kemudian, hubungkan grup parameter dengan klaster. Untuk petunjuknya, lihat Mengelola grup parameter menggunakan konsol atau Mengonfigurasi nilai parameter menggunakan AWS CLI.

> **Note**
>
> Ingatlah untuk mem-boot ulang klaster setelah Anda mengaitkan grup parameter kustom dengannya.

- Grup kerja Nirserver – Untuk mengaktifkan kepekaan huruf besar/kecil di grup kerja Redshift Nirserver, Anda harus menggunakan AWS CLI. Konsol Amazon Redshift saat ini tidak mendukung modifikasi nilai parameter Redshift Nirserver. Kirim permintaan update-workgroup berikut:

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

  Anda tidak perlu mem-boot ulang grup kerja setelah Anda mengubah nilai parameternya.

Konfigurasikan otorisasi untuk gudang data

Setelah Anda membuat gudang data, Anda harus mengkonfigurasi sumber RDS database cluster sebagai sumber integrasi resmi. Untuk petunjuknya, lihat Mengonfigurasi otorisasi untuk gudang data Amazon Redshift Anda.

Langkah selanjutnya

Dengan basis data RDS sumber cluster dan gudang data target Amazon Redshift, Anda sekarang dapat membuat integrasi nol-ETL dan mereplikasi data. Untuk petunjuk, lihat the section called “Membuat integrasi nol-ETL”.

Membuat integrasi nol-ETL Amazon RDS dengan Amazon Redshift

Ini adalah dokumentasi prarilis untuk integrasi nol-ETL Amazon RDS dengan Amazon Redshift, yang sedang dalam rilis pratinjau. Dokumentasi dan fitur dapat berubah. Sebaiknya gunakan fitur ini hanya dalam lingkungan pengujian, bukan dalam lingkungan produksi. Untuk syarat dan ketentuan pratinjau, lihat Beta dan Pratinjau dalam Persyaratan Layanan AWS.

Saat membuat integrasi Amazon RDS Zero-ETL, Anda menentukan cluster target. Anda juga dapat menyesuaikan pengaturan enkripsi dan menambahkan tag. Amazon RDS menciptakan integrasi antara cluster sumber dan targetnya. Setelah integrasi aktif, data apa pun yang Anda masukkan ke dalam database sumber akan direplikasi ke target Amazon Redshift yang dikonfigurasi.

Topik

Prasyarat

Sebelum membuat integrasi Nol-ETL, Anda harus membuat cluster DB instans Single-AZ atau multi-AZ DB sumber dan gudang data Amazon target. Anda juga harus mengizinkan replikasi ke gudang data dengan menambahkan database sebagai sumber integrasi resmi.

Untuk petunjuk cara menyelesaikan setiap langkah ini, lihat the section called “Mulai menggunakan integrasi nol-ETL”.

Izin yang diperlukan

Izin IAM tertentu diperlukan untuk membuat integrasi nol-ETL. Setidaknya, Anda memerlukan izin untuk melakukan tindakan berikut:

-
- Lihat dan hapus semua integrasi nol-ETL.
- Buat integrasi masuk ke gudang data target. Anda tidak memerlukan izin ini jika akun yang sama memiliki gudang data Amazon Redshift dan akun ini merupakan prinsipal yang diotorisasi untuk gudang data tersebut. Untuk informasi tentang menambahkan prinsipal resmi, lihat Mengonfigurasi otorisasi untuk gudang data Amazon Redshift Anda.

Contoh kebijakan berikut menunjukkan izin hak akses paling rendah yang diperlukan untuk membuat dan mengelola integrasi. Anda mungkin tidak memerlukan izin persis ini jika pengguna atau peran Anda memiliki izin yang lebih luas, seperti kebijakan terkelola `AdministratorAccess`.

> **Note**
>
> Amazon Resource Name (ARN) Redshift memiliki format berikut ini. Perhatikan penggunaan garis miring ke depan (`/`), bukan titik dua (`:`), sebelum UUID ruang nama nirserver.
>
> - Klaster terprovisi – `arn:aws:redshift:{region}:{account-id}:namespace:namespace-uuid`
> - Nirserver – `arn:aws:redshift-serverless:{region}:{account-id}:namespace/namespace-uuid`

Contoh kebijakan

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "rds:CreateIntegration"
        ],
        "Resource": [
            "arn:aws:rds:{region}:{account-id}:db:source-db",
            "arn:aws:rds:{region}:{account-id}:integration:*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "rds:DescribeIntegrations"
        ],
        "Resource": ["*"]
    },
    {
        "Effect": "Allow",
        "Action": [
            "rds:DeleteIntegration"
        ],
        "Resource": [
            "arn:aws:rds:{region}:{account-id}:integration:*"
        ]
    },
    {
```

```
        "Effect": "Allow",
        "Action": [
            "redshift:CreateInboundIntegration"
        ],
        "Resource": [
            "arn:aws:redshift:{region}:{account-id}:namespace:namespace-uuid"
        ]
    }]
}
```

Memilih gudang data target di akun yang berbeda

Jika Anda berencana untuk menentukan gudang data Amazon Redshift target yang ada di gudang lain Akun AWS, Anda harus membuat peran yang memungkinkan pengguna di akun saat ini mengakses sumber daya di akun target. Untuk informasi selengkapnya, lihat Menyediakan akses ke pengguna IAM di pengguna lain Akun AWS yang Anda miliki.

Peran harus memiliki izin berikut, yang memungkinkan pengguna melihat klaster terprovisi Amazon Redshift dan ruang nama Redshift Nirserver di akun target.

Izin dan kebijakan kepercayaan yang diperlukan

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

Peran harus memiliki kebijakan kepercayaan berikut ini, yang menentukan ID akun target.

```
{
   "Version":"2012-10-17",
```

```
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "AWS": "arn:aws:iam::{external-account-id}:root"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

Untuk petunjuk cara membuat peran, lihat Membuat peran menggunakan kebijakan kepercayaan kustom.

Membuat integrasi nol-ETL

Anda dapat membuat integrasi nol-ETL menggunakan AWS Management Console, AWS CLI, atau API RDS.

Secara default, RDS for MySQL akan langsung melakukan purging file log biner. Karena integrasi nol-ETL bergantung pada log biner untuk mereplikasi data dari sumber ke target, periode retensi untuk database sumber harus setidaknya satu jam. Segera setelah Anda membuat integrasi, Amazon RDS memeriksa periode penyimpanan file log biner untuk database sumber yang dipilih. Jika nilai saat ini adalah 0, Amazon RDS secara otomatis mengubahnya menjadi 1. Jika tidak, nilainya tetap sama.

Konsol

Untuk membuat integrasi nol-ETL

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi kiri, pilih Integrasi nol-ETL.
3. Pilih Buat integrasi nol-ETL.
4. Untuk Pengidentifikasi integrasi, masukkan nama untuk integrasi. Nama dapat memiliki maksimal 63 karakter alfanumerik dan dapat menyertakan tanda hubung.
5. Pilih Selanjutnya.
6. Untuk Sumber, pilih berasal. instance harus menjalankan RDS untuk MySQL versi 8.0.28 atau lebih tinggi Aurora MySQL versi 3.05 atau lebih tinggi).

> **Note**
>
> , RDS memberi tahu Anda jika parameter DB tidak dikonfigurasi dengan benar. Jika Anda menerima pesan ini, Anda dapat memilih Perbaiki untuk saya, atau mengonfigurasinya secara manual. Untuk petunjuk cara memperbaikinya secara manual, lihat the section called “Langkah 1: Buat grup parameter DB kustom”.
> Memodifikasi parameter DB membutuhkan boot ulang. Sebelum Anda dapat membuat integrasi, reboot harus lengkap dan nilai parameter baru harus berhasil diterapkan ke database.

7. Setelah database sumber Anda berhasil dikonfigurasi, pilih Berikutnya.
8. Untuk Target, lakukan hal berikut:

    1. (Opsional) Untuk menggunakan target Amazon Redshift yang berbeda Akun AWS , pilih Tentukan akun yang berbeda. Kemudian, masukkan ARN peran IAM dengan izin untuk menampilkan gudang data Anda. Untuk petunjuk cara membuat peran IAM, lihat the section called “Memilih gudang data target di akun yang berbeda”.
    2. Anda dapat memilih klaster Amazon Redshift terprovisi atau ruang nama Redshift Nirserver sebagai target.

> **Note**
>
> RDS akan memberi tahu Anda jika pengaturan kebijakan sumber daya atau kepekaan huruf besar/kecil untuk gudang data yang ditentukan tidak dikonfigurasi dengan benar. Jika Anda menerima pesan ini, Anda dapat memilih Perbaiki untuk saya, atau mengonfigurasinya secara manual. Untuk petunjuk cara memperbaikinya secara manual, lihat Mengaktifkan kepekaan huruf besar/kecil untuk gudang data Anda dan Mengonfigurasi otorisasi untuk gudang data Anda dalam Panduan Manajemen Amazon Redshift.
> Modifikasi kepekaan huruf besar/kecil untuk klaster Redshift terprovisi memerlukan boot ulang. Sebelum Anda dapat membuat integrasi, boot ulang harus selesai dan nilai parameter baru harus berhasil diterapkan ke klaster.
> Jika sumber dan target yang Anda pilih berada di Akun AWS yang berbeda, maka Amazon RDS tidak dapat memperbaiki pengaturan ini untuk Anda. Anda harus menavigasi ke akun lain dan memperbaikinya secara manual di Amazon Redshift.

9. Setelah gudang data target Anda dikonfigurasi dengan benar, pilih Berikutnya.
10. (Opsional) Untuk Tag, tambahkan satu atau beberapa tag ke integrasi. Untuk informasi selengkapnya, lihat the section called “Memberi tag pada sumber daya RDS”.
11. Untuk Enkripsi, tentukan cara enkripsi integrasi Anda. Secara default, RDS mengenkripsi semua integrasi dengan file. Kunci milik AWS Untuk memilih kunci yang dikelola pelanggan, aktifkan Sesuaikan pengaturan enkripsi dan pilih kunci KMS yang akan digunakan untuk enkripsi. Untuk informasi selengkapnya, lihat the section called “Mengenkripsi sumber daya Amazon RDS”.

    > **Note**
    >
    > Jika Anda menentukan kunci KMS kustom, kebijakan kunci harus mengizinkan tindakan `kms:CreateGrant` untuk prinsipal layanan Amazon Redshift (`redshift.amazonaws.com`). Untuk informasi selengkapnya, lihat Membuat kebijakan kunci di Panduan Developer AWS Key Management Service .

    Secara opsional, tambahkan konteks enkripsi. Untuk informasi lebih lanjut, lihat Konteks enkripsi di Panduan Developer AWS Key Management Service .
12. Pilih Selanjutnya.
13. Tinjau pengaturan integrasi Anda dan pilih Buat integrasi nol-ETL. Dibutuhkan sekitar 30 menit agar integrasi menjadi aktif.

    Jika pembuatan gagal, lihat the section called “Saya tidak dapat membuat integrasi nol-ETL” untuk langkah-langkah pemecahan masalah.

Integrasi memiliki status `Creating` ketika sedang dibuat dan gudang data Amazon Redshift target memiliki status `Modifying`. Selama waktu ini, Anda tidak dapat mengueri gudang data atau membuat perubahan konfigurasi apa pun di dalamnya.

Ketika integrasi berhasil dibuat, status integrasi dan gudang data Amazon Redshift target berubah menjadi `Active`.

AWS CLI

Untuk membuat integrasi nol-ETL menggunakan AWS CLI, gunakan perintah create-integration dengan opsi berikut:

- `--integration-name` – Tentukan nama untuk integrasi.

- `--source-arn`— Tentukan ARN dari cluster yang akan menjadi sumber integrasi.
- `--target-arn` – Tentukan ARN gudang data Amazon Redshift yang akan menjadi target integrasi.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds create-integration \
    --integration-name my-integration \
    --source-arn arn:aws:rds:{region}:{account-id}:my-cluster \
    --target-arn arn:aws:redshift:{region}:{account-id}:namespace:namespace-uuid
```

Untuk Windows:

```
aws rds create-integration ^
    --integration-name my-integration ^
    --source-arn arn:aws:rds:{region}:{account-id}:my-cluster ^
    --target-arn arn:aws:redshift:{region}:{account-id}:namespace:namespace-uuid
```

API RDS

Untuk membuat integrasi nol-ETL menggunakan API Amazon RDS, gunakan operasi `CreateIntegration` dengan parameter berikut ini:

- `IntegrationName` – Tentukan nama untuk integrasi.
- `SourceArn`— Tentukan ARN dari cluster yang akan menjadi sumber integrasi.
- `TargetArn` – Tentukan ARN gudang data Amazon Redshift yang akan menjadi target integrasi.

Langkah selanjutnya

Setelah berhasil membuat integrasi nol-ETL, Anda harus membuat basis data tujuan dalam klaster atau grup kerja Amazon Redshift target Anda. Kemudian, Anda dapat mulai menambahkan data ke database RDS sumber cluster dan menanyakannya di Amazon Redshift. Untuk petunjuknya, lihat Membuat basis data tujuan di Amazon Redshift.

Menambahkan data ke database RDS sumber cluster dan menanyakannya di Amazon Redshift

Ini adalah dokumentasi prarilis untuk integrasi nol-ETL Amazon RDS dengan Amazon Redshift, yang sedang dalam rilis pratinjau. Dokumentasi dan fitur dapat berubah. Sebaiknya gunakan fitur ini hanya dalam lingkungan pengujian, bukan dalam lingkungan produksi. Untuk syarat dan ketentuan pratinjau, lihat Beta dan Pratinjau dalam Persyaratan Layanan AWS.

Untuk menyelesaikan pembuatan integrasi nol-ETL yang mereplikasi data dari Amazon RDS ke dalam Amazon Redshift, Anda harus membuat basis data tujuan di Amazon Redshift.

Pertama, hubungkan ke klaster atau grup kerja Amazon Redshift Anda dan buat basis data dengan referensi ke pengidentifikasi integrasi Anda. Kemudian, Anda dapat menambahkan data ke basis data RDS sumber Anda cluster dan melihatnya direplikasi di Amazon Redshift.

Topik

Buat basis data tujuan di Amazon Redshift

Sebelum Anda dapat mulai mereplikasi data ke Amazon Redshift, setelah Anda membuat integrasi, Anda harus membuat basis data tujuan di gudang data target Anda. Basis data tujuan ini harus menyertakan referensi ke pengidentifikasi integrasi. Anda dapat menggunakan konsol Amazon Redshift atau Editor kueri v2 untuk membuat basis data.

Untuk petunjuk cara membuat basis data tujuan, lihat Membuat basis data tujuan di Amazon Redshift.

Tambahkan data ke database sumber

Tambahkan beberapa data ke database RDS Kluster yang ingin Anda tiru ke gudang data Amazon Redshift Anda.

> **Note**
>
> Ada perbedaan antara jenis data di Amazon RDS dan Amazon Redshift. Untuk tabel pemetaan jenis data, lihat the section called “Perbedaan jenis data”.

Pertama, sambungkan ke database sumber menggunakan klien MySQL pilihan Anda. Untuk petunjuk, lihat the section called “Menghubungkan ke instans DB yang menjalankan MySQL”.

Kemudian, buat tabel dan masukkan urutan data sampel.

> **Important**
>
> Pastikan tabel memiliki kunci primer. Jika tidak, tabel tidak dapat direplikasi ke gudang data target.

Contoh berikut menggunakan utilitas MySQL Workbench.

```
CREATE DATABASE my_db;

USE my_db;

CREATE TABLE books_table (ID int NOT NULL, Title VARCHAR(50) NOT NULL, Author
 VARCHAR(50) NOT NULL,
Copyright INT NOT NULL, Genre VARCHAR(50) NOT NULL, PRIMARY KEY (ID));

INSERT INTO books_table VALUES (1, 'The Shining', 'Stephen King', 1977, 'Supernatural
 fiction');
```

Kueri data Anda di Amazon Redshift

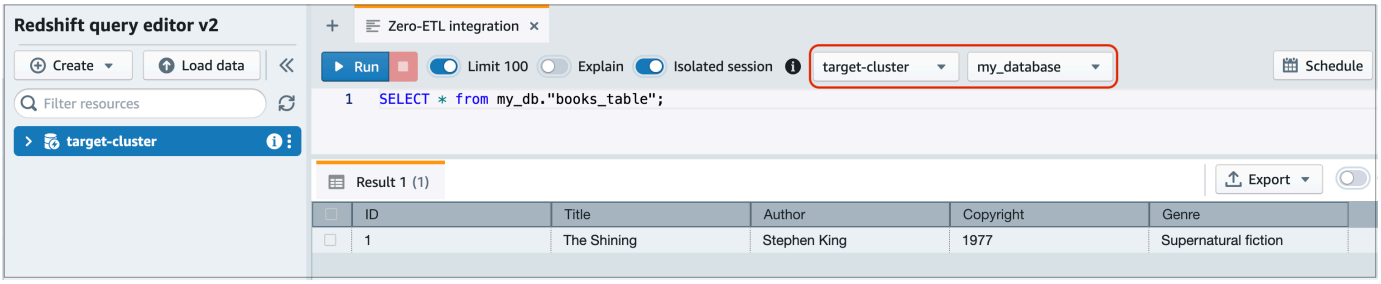
Setelah Anda menambahkan data ke database RDS cluster , itu direplikasi ke Amazon Redshift dan siap untuk ditanyakan.

Untuk mengueri data yang direplikasi

1. Buka konsol Amazon Redshift dan pilih Editor kueri v2 dari panel navigasi kiri.

2. Hubungkan ke klaster atau grup kerja Anda dan pilih basis data tujuan Anda (yang Anda buat dari integrasi) dari menu dropdown (destination_database dalam contoh ini). Untuk petunjuk cara membuat basis data tujuan, lihat Membuat basis data tujuan di Amazon Redshift.
3. Jalankan perintah berikut untuk memilih semua data dari tabel yang Anda buat di basis data RDS sumber DB cluster:

```
SELECT * from my_db."books_table";
```



- *my_db* adalah nama skema basis data RDS.
- *books_table* adalah nama tabel RDS.

Anda juga dapat mengueri data menggunakan klien baris perintah:

```
destination_database=# select * from my_db."books_table";

 ID |       Title |        Author |   Copyright |                 Genre |  txn_seq |
 txn_id
----+-------------+---------------+-------------+-----------------------+----------
+--------+
  1 | The Shining |  Stephen King |        1977 |   Supernatural fiction |        2 |
 12192
```

> **Note**
>
> Untuk kepekaan huruf besar/kecil, gunakan tanda kutip ganda (" ") untuk nama skema, tabel, dan kolom. Untuk informasi selengkapnya, lihat enable_case_sensitive_identifier.

Perbedaan jenis data antara basis data RDS dan Amazon Redshift

Tabel berikut menunjukkan pemetaan jenis data RDS for MySQL ke jenis data Amazon Redshift yang sesuai. Amazon RDS saat ini hanya mendukung jenis data ini untuk integrasi nol-ETL.

Jika tabel di database sumber Anda menyertakan tipe data yang tidak didukung, tabel akan tidak sinkron dan tidak dapat dikonsumsi oleh target Amazon Redshift. Streaming dari sumber ke target berlanjut, tetapi tabel dengan jenis data yang tidak didukung tidak tersedia. Untuk memperbaiki tabel dan membuatnya tersedia di Amazon Redshift, Anda harus mengembalikan perubahan yang melanggar secara manual, lalu menyegarkan integrasi dengan menjalankan `ALTER DATABASE...INTEGRATION REFRESH`.

RDS for MySQL

Jenis data RDS for MySQL	Jenis data Amazon Redshift	Deskripsi	Batasan
INT	INTEGER	Bilangan bulat empat byte bertanda	
SMALLINT	SMALLINT	Bilangan bulat dua byte bertanda	
TINYINT	SMALLINT	Bilangan bulat dua byte bertanda	
MEDIUMINT	INTEGER	Bilangan bulat empat byte bertanda	
BIGINT	BIGINT	Bilangan bulat delapan byte bertanda	

Jenis data RDS for MySQL	Jenis data Amazon Redshift	Deskripsi	Batasan
INT UNSIGNED	BIGINT	Bilangan bulat delapan byte bertanda	
TINYINT UNSIGNED	SMALLINT	Bilangan bulat dua byte bertanda	
MEDIUMINT UNSIGNED	INTEGER	Bilangan bulat empat byte bertanda	
BIGINT UNSIGNED	DECIMAL(20,0)	Numerik persis dari presisi yang dapat dipilih	
DECIMAL(p,s) = NUMERIC(p,s)	DECIMAL(p,s)	Numerik persis dari presisi yang dapat dipilih	Presisi lebih besar dari 38 dan penskalaan lebih besar dari 37 tidak didukung
DECIMAL(p,s) UNSIGNED = NUMERIC(p,s) UNSIGNED	DECIMAL(p,s)	Numerik persis dari presisi yang dapat dipilih	Presisi lebih besar dari 38 dan penskalaan lebih besar dari 37 tidak didukung
FLOAT4/REAL	REAL	Angka floating-point presisi tunggal	

Jenis data RDS for MySQL	Jenis data Amazon Redshift	Deskripsi	Batasan
FLOAT4/REAL UNSIGNED	REAL	Angka floating-point presisi tunggal	
DOUBLE/REAL/FLOAT8	DOUBLE PRECISION	Angka floating-point presisi ganda	
DOUBLE/REAL/FLOAT8 UNSIGNED	DOUBLE PRECISION	Angka floating-point presisi ganda	
BIT(n)	VARBYTE(8)	Nilai biner dengan panjang variabel	
BINER (n)	VARBYTE(n)	Nilai biner dengan panjang variabel	
VARBINER (n)	VARBYTE(n)	Nilai biner dengan panjang variabel	
CHAR(n)	VARCHAR(n)	Nilai string dengan panjang variabel	
VARCHAR(n)	VARCHAR(n)	Nilai string dengan panjang variabel	
TEXT	VARCHAR(65535)	Nilai string dengan panjang variabel hingga 65535 byte	

Jenis data RDS for MySQL	Jenis data Amazon Redshift	Deskripsi	Batasan
TINYTEXT	VARCHAR(255)	Nilai string dengan panjang variabel hingga 255 byte	
ENUM	VARCHAR(1020)	Nilai string dengan panjang variabel hingga 1020 byte	
SET	VARCHAR(1020)	Nilai string dengan panjang variabel hingga 1020 byte	
DATE	DATE	Tanggal kalender (tahun, bulan, hari)	
DATETIME	TIMESTAMP	Tanggal dan waktu (tanpa zona waktu)	
TIMESTAMP(p)	TIMESTAMP	Tanggal dan waktu (tanpa zona waktu)	
TIME	VARCHAR(18)	Nilai string dengan panjang variabel hingga 18 byte	

Jenis data RDS for MySQL	Jenis data Amazon Redshift	Deskripsi	Batasan
YEAR	VARCHAR(4)	Nilai string dengan panjang variabel hingga 4 byte	
JSON	SUPER	Data atau dokumen semi-terstruktur sebagai nilai	

Melihat dan memantau integrasi nol-ETL Amazon RDS dengan Amazon Redshift

Ini adalah dokumentasi prarilis untuk integrasi nol-ETL Amazon RDS dengan Amazon Redshift, yang sedang dalam rilis pratinjau. Dokumentasi dan fitur dapat berubah. Sebaiknya gunakan fitur ini hanya dalam lingkungan pengujian, bukan dalam lingkungan produksi. Untuk syarat dan ketentuan pratinjau, lihat Beta dan Pratinjau dalam Persyaratan Layanan AWS.

Anda dapat melihat detail integrasi nol-ETL Amazon RDS untuk melihat informasi konfigurasi dan statusnya saat ini. Anda juga dapat memantau status integrasi dengan mengueri tampilan sistem tertentu di Amazon Redshift. Selain itu, Amazon Redshift menerbitkan metrik terkait integrasi tertentu ke Amazon, yang dapat Anda lihat dalam konsol CloudWatch Amazon Redshift.

Topik

- Melihat integrasi
- Memantau integrasi menggunakan tabel sistem
- Integrasi pemantauan dengan Amazon EventBridge

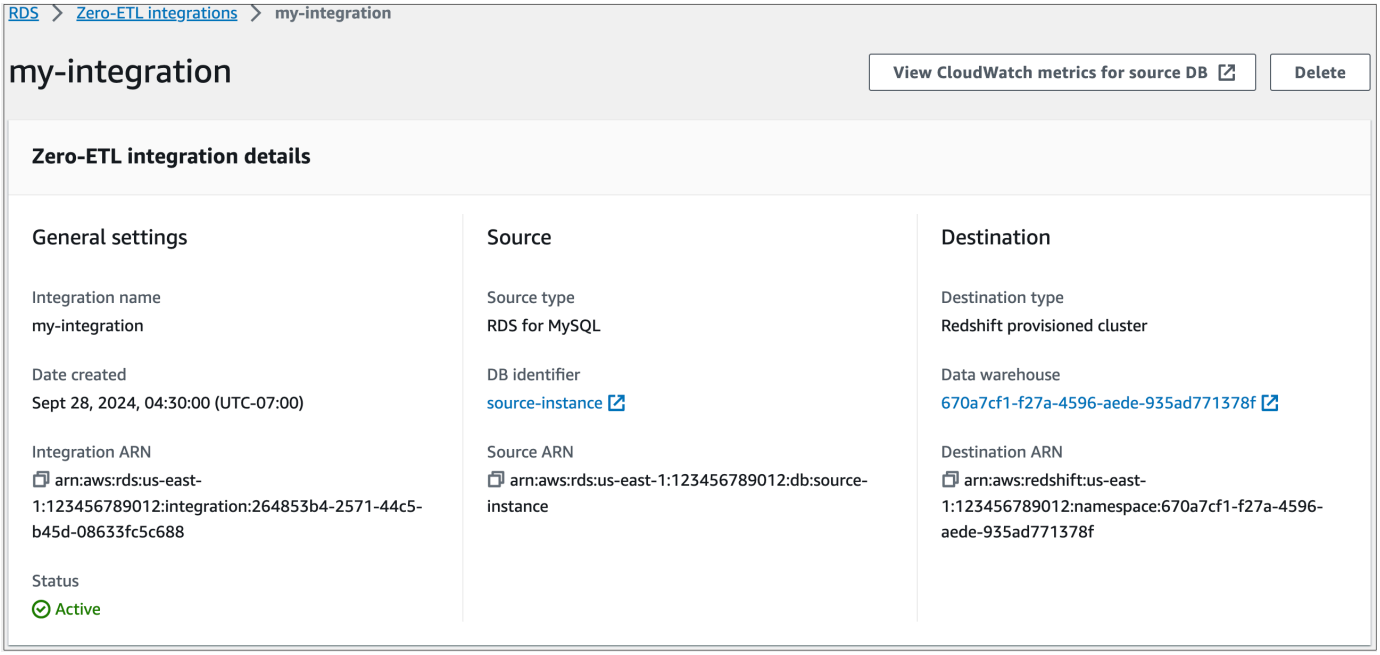
Melihat integrasi

Anda dapat melihat integrasi Amazon RDS Zero-ETL dengan Amazon Redshift menggunakan,, atau RDS AWS Management Console API. AWS CLI

Konsol

Untuk melihat detail integrasi nol-ETL

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi sebelah kiri, pilih Integrasi nol-ETL.
3. Pilih integrasi untuk melihat detail lebih lanjut tentangnya, seperti basis data sumber dan gudang data target.



Integrasi dapat memiliki status berikut:

- `Creating` – Integrasi sedang dibuat.
- `Active` – Integrasi sedang mengirimkan data transaksional ke gudang data target.
- `Syncing` – Integrasi 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.

- `Needs attention` – Integrasi mengalami peristiwa atau kesalahan yang memerlukan intervensi manual untuk menyelesaikannya. Untuk memperbaiki masalah, ikuti petunjuk dalam pesan kesalahan di halaman detail integrasi.
- `Failed` – Integrasi mengalami peristiwa atau kesalahan yang tidak dapat dipulihkan dan tidak dapat diperbaiki. Anda harus menghapus dan membuat ulang integrasi.
- `Deleting` – Integrasi sedang dihapus.

AWS CLI

Untuk melihat semua integrasi nol-ETL di akun saat ini menggunakan AWS CLI, gunakan perintah deskripsi-integrasi dan tentukan opsi. `--integration-identifier`

Example

Untuk Linux, macOS, atau Unix:

```
aws rds describe-integrations \
    --integration-identifier ee605691-6c47-48e8-8622-83f99b1af374
```

Untuk Windows:

```
aws rds describe-integrations ^
    --integration-identifier ee605691-6c47-48e8-8622-83f99b1af374
```

API RDS

Untuk melihat integrasi nol-ETL menggunakan API Amazon RDS, gunakan operasi `DescribeIntegrations` dengan parameter `IntegrationIdentifier`.

Memantau integrasi menggunakan tabel sistem

Amazon Redshift memiliki tabel dan tampilan sistem yang berisi informasi tentang bagaimana sistem berfungsi. Anda dapat mengueri tabel dan tampilan sistem ini dengan cara yang sama seperti Anda akan mengueri tabel basis data lainnya. Untuk informasi selengkapnya tentang tabel dan tampilan sistem di Amazon Redshift, lihat Referensi tabel sistem dalam Panduan Developer Basis Data Amazon Redshift.

Anda dapat mengueri tampilan dan tabel sistem berikut untuk mendapatkan informasi tentang integrasi nol-ETL Anda dengan Amazon Redshift:

- SVV_INTEGRATION – Menyediakan detail konfigurasi untuk integrasi Anda.
- SVV_INTEGRATION_TABLE_STATE – Menjelaskan status setiap tabel dalam integrasi.
- SYS_INTEGRATION_TABLE_STATE_CHANGE – Menampilkan log perubahan status tabel untuk integrasi.
- SYS_INTEGRATION_ACTIVITY – Menyediakan informasi tentang proses integrasi yang selesai.

Semua CloudWatch metrik Amazon terkait integrasi berasal dari Amazon Redshift. Untuk informasi selengkapnya, lihat Memantau integrasi nol-ETL dalam Panduan Manajemen Amazon Redshift. Saat ini, Amazon RDS Aurora tidak mempublikasikan metrik integrasi apa pun. CloudWatch

Integrasi pemantauan dengan Amazon EventBridge

Amazon Redshift mengirimkan peristiwa terkait integrasi ke Amazon. EventBridge Untuk daftar peristiwa dan ID peristiwa terkait, lihat notifikasi peristiwa integrasi nol-ETL dengan Amazon di Panduan Manajemen Pergeseran Merah EventBridge Amazon.

Menghapus integrasi nol-ETL Amazon RDS dengan Amazon Redshift

> Ini adalah dokumentasi prarilis untuk integrasi nol-ETL Amazon RDS dengan Amazon Redshift, yang sedang dalam rilis pratinjau. Dokumentasi dan fitur dapat berubah. Sebaiknya gunakan fitur ini hanya dalam lingkungan pengujian, bukan dalam lingkungan produksi. Untuk syarat dan ketentuan pratinjau, lihat Beta dan Pratinjau dalam Persyaratan Layanan AWS.

Data transaksional Anda tidak dihapus dari Amazon RDS atau Amazon Redshift, tetapi Amazon RDS tidak mengirim data baru ke Amazon Redshift.

Anda hanya dapat menghapus integrasi ketika integrasi ini memiliki status `Active`, `Failed`, `Syncing`, atau `Needs attention`.

Anda dapat menghapus integrasi nol-ETL menggunakan AWS Management Console,, atau RDS API AWS CLI.

Konsol

Untuk menghapus integrasi nol-ETL

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi sebelah kiri, pilih Integrasi nol-ETL.
3. Pilih integrasi nol-ETL yang ingin Anda hapus.
4. Pilih Tindakan, Hapus, dan konfirmasi penghapusan.

AWS CLI

Untuk menghapus integrasi nol-ETL, gunakan perintah delete-integration dan tentukan opsi `--integration-identifier`.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds delete-integration \
    --integration-identifier ee605691-6c47-48e8-8622-83f99b1af374
```

Untuk Windows:

```
aws rds delete-integration ^
    --integration-identifier ee605691-6c47-48e8-8622-83f99b1af374
```

API RDS

Untuk menghapus integrasi nol-ETL menggunakan API Amazon RDS, gunakan operasi `DeleteIntegration` dengan parameter `IntegrationIdentifier`.

Memecahkan masalah integrasi nol-ETL Amazon RDS dengan Amazon Redshift

Ini adalah dokumentasi prarilis untuk integrasi nol-ETL Amazon RDS dengan Amazon Redshift, yang sedang dalam rilis pratinjau. Dokumentasi dan fitur dapat berubah. Sebaiknya gunakan

fitur ini hanya dalam lingkungan pengujian, bukan dalam lingkungan produksi. Untuk syarat dan ketentuan pratinjau, lihat Beta dan Pratinjau di Persyaratan Layanan AWS.

Anda dapat memeriksa status integrasi nol-ETL dengan mengueri tabel sistem SVV_INTEGRATION di Amazon Redshift. Jika kolom `state` memiliki nilai `ErrorState`, artinya ada sesuatu yang salah. Untuk informasi selengkapnya, lihat the section called “Pemantauan menggunakan tabel sistem”.

Gunakan informasi berikut untuk memecahkan masalah umum integrasi nol-ETL Amazon RDS dengan Amazon Redshift.

Topik

- Saya tidak dapat membuat integrasi nol-ETL
- Integrasi saya dalam status Syncing permanen
- Satu atau beberapa tabel Amazon Redshift saya memerlukan sinkronisasi ulang

Saya tidak dapat membuat integrasi nol-ETL

Jika Anda tidak dapat membuat integrasi nol-ETL, pastikan hal berikut sudah benar untuk instans DB sumber Anda:

- sumber Anda menjalankan RDS untuk MySQL versi 8.0.28 atau lebih tinggi Aurora MySQL versi 3.05 (kompatibel dengan MySQL 8.0.32) atau lebih tinggi. Untuk memvalidasi versi mesin, pilih tab Konfigurasi untuk instans DB dan periksa Versi mesin.
- Anda mengonfigurasi parameter DB dengan benar. Jika parameter yang diperlukan tidak diatur dengan benar atau tidak terkait dengan instans DB, pembuatan akan gagal. Lihat the section called “Langkah 1: Buat grup parameter DB kustom”.

Selain itu, pastikan hal-hal berikut ini sudah benar untuk gudang data target Anda:

- Kepekaan huruf besar/kecil diaktifkan. Lihat Mengaktifkan kepekaan huruf besar/kecil untuk gudang data Anda.
- Anda menambahkan pengguna utama resmi dan sumber integrasi yang benar. Untuk petunjuknya, lihat Mengonfigurasi otorisasi untuk gudang data Amazon Redshift Anda.

Integrasi saya dalam status **`Syncing`** permanen

Integrasi Anda mungkin secara konsisten menunjukkan status `Syncing` jika Anda mengubah nilai salah satu parameter DB yang diperlukan.

Untuk memperbaiki masalah ini, periksa nilai parameter dalam grup parameter yang terkait dengan instans DB sumber, dan pastikan nilai tersebut cocok dengan nilai yang diperlukan. Untuk informasi selengkapnya, lihat the section called “Langkah 1: Buat grup parameter DB kustom”.

Jika Anda memodifikasi parameter, pastikan untuk melakukan boot ulang instans DB untuk menerapkan perubahan.

Satu atau beberapa tabel Amazon Redshift saya memerlukan sinkronisasi ulang

Untuk menjalankan perintah tertentu di instans DB sumber, Anda mungkin perlu menyinkronkan ulang tabel Anda. Dalam kasus ini, tampilan sistem SVV_INTEGRATION_TABLE_STATE menunjukkan `table_state` dari `ResyncRequired`. Artinya, integrasi harus memuat ulang data sepenuhnya untuk tabel tersebut dari MySQL ke Amazon Redshift.

Tabel akan memasuki status `Syncing` saat mulai disinkronkan ulang. Anda tidak perlu mengambil tindakan manual apa pun untuk menyinkronkan ulang tabel. Saat data tabel disinkronkan ulang, Anda mungkin tidak dapat mengaksesnya di Amazon Redshift.

Berikut ini adalah beberapa contoh operasi yang dapat menempatkan tabel ke dalam status `ResyncRequired`, dan beberapa kemungkinan alternatif untuk dipertimbangkan.

Operasi	Contoh	Alternatif
Menambahkan kolom ke posisi tertentu	`ALTER TABLE table_name ADD COLUMN column_name INTEGER NOT NULL first;`	Amazon Redshift tidak mendukung penambahan kolom ke posisi tertentu menggunakan kata kunci `first` atau

Operasi	Contoh	Alternatif
		`after`. Jika urutan kolom dalam tabel target tidak penting, tambahkan kolom ke akhir tabel menggunakan perintah yang lebih sederhana: `ALTER TABLE table_name ADD COLUMN column_name column_type ;`

Operasi	Contoh	Alternatif

Menambahkan kolom stempel waktu dengan `CURRENT_TIMESTAMP` default	`ALTER TABLE table_name`
	`  ADD COLUMN column_name  TIMESTAMP`
	`  NOT NULL DEFAULT CURRENT_TIMESTAMP;`
		Nilai `CURRENT_TIMESTAMP` untuk baris tabel yang ada dihitung oleh RDS for MySQL dan tidak dapat disimulasikan di Amazon Redshift tanpa sinkronisasi ulang data tabel penuh.
		
		Jika memungkinkan, alihkan nilai default ke konstanta literal seperti `2023-01-01 00:00:15` untuk menghindari latensi dalam ketersediaan tabel.

Operasi	Contoh	Alternatif
Melakukan beberapa operasi kolom dalam satu perintah	`ALTER TABLE table_name ADD COLUMN column_1, RENAME COLUMN column_2 TO column_3;`	Pertimbangkan untuk membagi perintah menjadi dua operasi terpisah, `ADD` dan `RENAME`, yang tidak memerlukan sinkronisasi ulang.

Amazon RDS for Db2

Amazon RDS mendukung instans DB yang menjalankan edisi IBM Db2 berikut:

- Db2 Standard Edition
- Db2 Advanced Edition

Amazon RDS mendukung instans DB yang menjalankan versi Db2 berikut:

- Db2 11.5

Untuk informasi selengkapnya tentang dukungan versi kecil, lihat Versi-versi Db2 pada Amazon RDS.

Sebelum membuat instans DB, selesaikan langkah-langkah di bagian Menyiapkan Amazon RDS dalam panduan pengguna ini. Saat Anda membuat instans DB menggunakan pengguna utama, pengguna tersebut mendapatkan otoritas `DBADM`, dengan beberapa batasan. Gunakan pengguna ini untuk tugas administratif seperti membuat akun basis data tambahan. Anda tidak dapat menggunakan otoritas tingkat instans `SYSADM`, `SYSCTRL`, `SYSMAINT`, atau otoritas tingkat basis data `SECADM`.

Anda dapat membuat berikut ini:

- Instans DB
- Snapshot DB
- P oint-in-time mengembalikan
- Pencadangan penyimpanan otomatis
- Pencadangan penyimpanan manual

Anda dapat menggunakan instans DB yang menjalankan Db2 di cloud privat virtual (VPC). Anda juga dapat menambahkan fitur ke instans DB RDS for Db2 dengan mengaktifkan berbagai opsi. Amazon RDS mendukung deployment Multi-AZ untuk RDS for Db2 sebagai solusi failover dengan ketersediaan tinggi.

> **Important**
>
> Untuk memberikan pengalaman layanan terkelola, Amazon RDS tidak memberikan akses shell ke instans DB. Hal tersebut juga membatasi akses ke prosedur dan tabel sistem tertentu yang membutuhkan hak istimewa tinggi. Anda dapat mengakses basis data Anda menggunakan klien SQL standar seperti IBM Db2 CLP. Namun, Anda tidak dapat mengakses host secara langsung dengan menggunakan Telnet atau Secure Shell (SSH).

Topik

Ikhtisar Db2 di Amazon RDS

Anda dapat membaca bagian-bagian berikut untuk mendapatkan ikhtisar Db2 di Amazon RDS.

Topik

- Parameter-parameter RDS for Db2
- Pemeriksaan EBCDIC untuk database Db2 di Amazon RDS

Fitur-fitur RDS for Db2

Amazon RDS for Db2 mendukung sebagian besar fitur dan kemampuan basis data IBM Db2. Beberapa fitur mungkin memiliki dukungan terbatas atau privilese yang dibatasi. Lihat informasi yang lebih lengkap tentang fitur-fitur basis data Db2 untuk versi Db2 tertentu dalam dokumentasi IBM Db2.

Anda dapat memfilter fitur-fitur Amazon RDS baru pada halaman Apa yang Baru dengan Basis Data?. Untuk Produk, pilih Amazon RDS. Kemudian, Anda dapat mencari dengan menggunakan kata kunci seperti**`Db2 2023`**.

> **Note**
>
> Daftar berikut tidak lengkap.

Topik

- Fitur-fitur yang didukung di RDS for Db2
- Fitur-fitur yang tidak didukung di RDS for Db2

Fitur-fitur yang didukung di RDS for Db2

RDS untuk Db2 mendukung fitur yang mencakup fitur asli IBM Db2 dan fitur yang merupakan inti dari Amazon RDS.

Fitur-fitur asli bagi IBM Db2

RDS for Db2 mendukung semua fitur basis data Db2 berikut:

- Pembuatan basis data standar yang menggunakan set kode, kolasi, ukuran halaman, dan wilayah buatan pelanggan. Gunakan prosedur tersimpan rdsadmin.create_database Amazon RDS.
- Penambahan, penghapusan, atau pengubahan pengguna dan grup lokal. Gunakan prosedur-prosedur tersimpan Amazon RDS untuk Memberikan dan mencabut privilese.
- Pembuatan peran dengan prosedur tersimpan rdsadmin.create_role Amazon RDS.

- Dukungan untuk tabel tersusun baris standar.
- Dukungan untuk beban kerja analitis bagi tabel tersusun kolom.
- Kemampuan mendefinisikan fitur-fitur kompatibilitas Db2 seperti Oracle dan MySQL.
- Support untuk prosedur tersimpan eksternal Java berbasis.
- Support untuk enkripsi data dalam perjalanan dengan menggunakan SSL/TLS.
- Pemantauan status basis data (`ALIVE`, `DOWN`, `STORAGE_FULL`, `UNKNOWN`, dan `STANDBY_CONNECTABLE`).
- Pemulihan database offline atau online Linux (LE) yang disediakan pelanggan. Gunakan prosedur-prosedur tersimpan Amazon RDS untuk Mengelola basis data.
- Penerapan log arsip Db2 yang disediakan pelanggan untuk menjaga database tetap disinkronkan dengan database Db2 yang dikelola sendiri. Gunakan prosedur-prosedur tersimpan Amazon RDS untuk Mengelola basis data.
- Support untuk audit tingkat instans dan tingkat database Db2.
- Dukungan untuk federasi homogen.
- Kemampuan memuatkan tabel dari file data di Amazon Simple Storage Service (Amazon S3).
- Otorisasi yang diberikan kepada pengguna, grup atau peran, seperti`CONNECT`,`SYSMON`,`ACCESSCTRL`,`DATAACCESS`,`SQLADM`,`WLMADM`, `EXPLAINLOAD`, atau `IMPLICIT_SCHEMA`

Fitur-fitur inti bagi Amazon RDS

RDS for Db2 mendukung fitur-fitur inti Amazon RDS berikut:

- Grup parameter khusus untuk ditetapkan ke instance DB.
- Pembuatan, modifikasi, dan penghapusan instans DB.
- Pemulihan cadangan database offline atau online Linux (LE) Db2 yang dikelola sendiri.

> **Note**
>
> Agar dapat memulihkan cadangan, jangan berikan nama ke basis data saat Anda membuat instans basis data. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.

- Support jenis penyimpanan gp3, io2, dan io1.

- Penggunaan AWS Managed Microsoft AD untuk Kerberos otentikasi, dan otorisasi grup LDAP untuk RDS untuk Db2.
- Modifikasi grup keamanan, port, jenis instans, penyimpanan, periode retensi cadangan, dan pengaturan lain untuk instans Db2 yang ada.
- Perlindungan penghapusan untuk instans DB.
- point-in-time Pemulihan Lintas Wilayah (PITR).
- Penggunaan AWS Key Management Service (AWS KMS) untuk enkripsi penyimpanan dan enkripsi saat istirahat.
- Instans DB multi-AZ dengan satu siaga untuk ketersediaan tinggi.
- Reboot instance DB.
- Pembaruan untuk menguasai kata sandi.
- Pemulihan instans DB ke waktu tertentu.
- Backup dan restorasi instans DB dengan menggunakan backup tingkat penyimpanan.
- Mulai dan hentikan instans DB.
- Pemeliharaan instans DB.

Fitur-fitur yang tidak didukung di RDS for Db2

RDS for Db2 tidak mendukung fitur-fitur basis data Db2 berikut:

- `SYSADM`,`SECADM`, dan `SYSMAINT` akses untuk pengguna master.
- Prosedur tersimpan eksternal ditulis dalam C, C ++, atau Cobol.
- Beberapa instans Db2 DB pada satu host.
- Beberapa database Db2 pada RDS tunggal untuk instance Db2 DB.
- Plugin GSS-API eksternal untuk otentikasi.
- Plugin pihak ketiga eksternal untuk mencadangkan atau memulihkan database Db2.
- Multi-node massively parallel processing (MPP), seperti. IBM Db2 Warehouse
- IBM Db2 pureScale.
- Pemulihan Bencana Ketersediaan Tinggi (HADR).
- Enkripsi basis data asli.
- Federasi heterogen untuk Db2.

- Cross-Region point-in-time-recovery (PITR) untuk backup terenkripsi.

Versi-versi Db2 pada Amazon RDS

Untuk Db2, nomor versi berbentuk major.minor.build.revision, misalnya, 11.5.9.0.sb000000.r1. Implementasi versi kami cocok dengan Db2.

utama

Nomor versi utama adalah bilangan bulat dan bagian fraksional pertama dari nomor versi, misalnya, 11.5. Perubahan versi dianggap besar jika nomor versi utama berubah — misalnya, dari versi 11.5 ke 12.1.

kecil

Nomor versi minor adalah bagian ketiga dan keempat dari nomor versi, misalnya, 9.0 di 11.5.9.0. Bagian ketiga menunjukkan modpack Db2, misalnya, 9 di 9.0. Bagian keempat menunjukkan fixpack Db2, misalnya, 0 di 9.0. Perubahan versi dianggap kecil jika modpack Db2 atau fixpack Db2 berubah—misalnya, beralih dari versi 11.5.9.0 ke 11.5.9.1, atau dari 11.5.9.0 ke 11.5.10.0, dengan pengecualian untuk menyediakan pembaruan tabel katalog. (Amazon RDS menangani pengecualian ini.)

membangun

Nomor build adalah bagian kelima dari nomor versi, misalnya, sb00000000 di 11.5.9.0.sb00000000. Nomor build di mana bagian angka adalah semua nol menunjukkan build standar. Nomor build di mana bagian angka tidak semua nol menunjukkan build khusus. Nomor bangun berubah jika ada perbaikan keamanan atau bangun khusus versi Db2 yang ada. Perubahan nomor build juga menunjukkan bahwa Amazon RDS secara otomatis menerapkan versi minor baru.

revisi

Nomor revisi adalah bagian keenam dari nomor versi, misalnya, r1 di 11.5.9.0.sb000000.r1. Revisi adalah revisi Amazon RDS untuk rilis Db2 yang ada. Perubahan nomor revisi menunjukkan bahwa Amazon RDS secara otomatis menerapkan versi minor baru.

Topik

Versi-versi kecil Db2 yang didukung di Amazon RDS

Tabel berikut menunjukkan versi-versi kecil Db2 yang saat ini didukung oleh Amazon RDS.

> **Note**
>
> Tanggal yang berupa hanya bulan dan tahun merupakan perkiraan, dan akan diperbarui dengan tanggal persisnya saat diketahui.

Versi mesin Db2	Tanggal rilis IBM	Tanggal rilis RDS	Tanggal akhir dukungan standar RDS
11.5			
11.5.9.0	15 November 2023	27 November 2023	

Anda dapat menentukan sebarang versi Db2 yang saat ini didukung ketika membuat instans basis data baru. Anda dapat menentukan versi utama (seperti Db2 11.5) dan sebarang versi kecil yang didukung untuk versi utama itu. Jika tidak ada versi yang ditentukan, Amazon RDS menetapkan bawaan ke versi yang didukung, biasanya versi terbaru. Jika versi utama ditentukan tetapi versi kecilnya tidak, Amazon RDS menetapkan bawaan ke rilis terbaru versi utama yang telah Anda tentukan. Untuk melihat daftar versi yang didukung, serta versi bawaan untuk instans basis data yang baru dibuat, gunakan perintah describe-db-engine-versions AWS Command Line Interface (AWS CLI).

Misalnya, untuk memerinci versi-versi mesin yang didukung bagi RDS for Db2, jalankan perintah AWS CLI berikut. Ganti *region* dengan Wilayah AWS Anda.

Untuk Linux, macOS, atau Unix:

```
aws rds describe-db-engine-versions \
    --filters Name=engine,Values=db2-ae,db2-se \
    --query "DBEngineVersions[].{Engine:Engine, EngineVersion:EngineVersion,
 DBParameterGroupFamily:DBParameterGroupFamily}" \
    --region region
```

Untuk Windows:

```
aws rds describe-db-engine-versions ^
    --filters Name=engine,Values=db2-ae,db2-se ^
    --query "DBEngineVersions[].{Engine:Engine, EngineVersion:EngineVersion,
 DBParameterGroupFamily:DBParameterGroupFamily}" ^
    --region region
```

Perintah ini menghasilkan output yang serupa dengan contoh berikut:

```
[
    {
    "Engine": "db2-ae",
    "EngineVersion": "11.5.9.0.sb00000000.r1",
    "DBParameterGroupFamily": "db2-ae-11.5"
    },
    {
    "Engine": "db2-se",
    "EngineVersion": "11.5.9.0.sb00000000.r1",
    "DBParameterGroupFamily": "db2-se-11.5"
    }
]
```

Versi Db2 bawaan mungkin berbeda-beda menurut Wilayah AWS. Untuk membuat instans basis data dengan versi kecil tertentu, tentukan versi kecil selama pembuatan instans basis data. Anda dapat menentukan versi default untuk mesin Wilayah AWS for `db2-ae` dan `db2-se` database dengan menjalankan `describe-db-engine-versions` perintah. Contoh berikut mengembalikan versi default untuk `db2-ae` di AS Timur (Virginia N.).

Untuk Linux, macOS, atau Unix:

```
aws rds describe-db-engine-versions \
    --default-only --engine db2-ae \
    --query "DBEngineVersions[].{Engine:Engine, EngineVersion:EngineVersion,
 DBParameterGroupFamily:DBParameterGroupFamily}" \
    --region us-east-1
```

Untuk Windows:

```
aws rds describe-db-engine-versions ^
    --default-only --engine db2-ae ^
    --query "DBEngineVersions[].{Engine:Engine, EngineVersion:EngineVersion,
 DBParameterGroupFamily:DBParameterGroupFamily}" ^
```

```
    --region us-east-1
```

Perintah ini menghasilkan output yang serupa dengan contoh berikut:

```
[
    {
    "Engine": "db2-ae",
    "EngineVersion": "11.5.9.0.sb00000000.r1",
    "DBParameterGroupFamily": "db2-ae-11.5"
    }
]
```

Dengan Amazon RDS, Anda mengendalikan waktu untuk memutakhirkan instans Db2 Anda ke versi utama baru yang didukung oleh Amazon RDS. Anda dapat mempertahankan kompatibilitas dengan versi Db2 tertentu, menguji versi baru dengan aplikasi Anda sebelum dikerahkan dalam produksi, dan melakukan pemutakhiran versi utama pada waktu yang paling pas dengan jadwal Anda.

Saat pemutakhiran versi minor otomatis diaktifkan, Amazon RDS secara otomatis memutakhirkan instans DB Anda ke versi minor Db2 baru karena didukung oleh Amazon RDS. Proses patching ini terjadi selama periode pemeliharaan terjadwal Anda. Anda dapat memodifikasi instans DB untuk mengaktifkan atau menonaktifkan peningkatan versi kecil otomatis.

Kecuali untuk versi Db2 11.5.9.1 dan 11.5.10.0, peningkatan otomatis ke versi minor Db2 baru mencakup peningkatan otomatis ke build dan revisi baru. Untuk 11.5.9.1 dan 11.5.10.0, tingkatkan versi minor secara manual.

Jika Anda memilih untuk tidak melakukan peningkatan terjadwal otomatis, Anda dapat melakukan peningkatan manual ke rilis versi kecil yang didukung dengan mengikuti prosedur yang sama seperti untuk pembaruan versi utama. Lihat informasinya di Meng-upgrade versi mesin instans DB.

Versi-versi utama Db2 yang didukung di Amazon RDS

Versi-versi utama RDS for Db2 tersedia di bawah dukungan standar setidaknya sampai akhir dukungan (dasar) IBM untuk versi IBM yang bersangkutan. Tabel berikut memerinci tanggal-tanggal yang dapat Anda gunakan untuk merencanakan siklus pengujian dan pemutakhiran. Jika Amazon memperpanjang dukungan untuk suatu versi RDS for Db2 lebih lama daripada yang dinyatakan semula, kami merencanakan untuk memperbarui tabel ini agar mencerminkan tanggal yang lebih belakangan.

Anda dapat menggunakan tanggal berikut untuk merencanakan siklus pengujian dan peningkatan Anda.

> **Note**
>
> Tanggal yang hanya terdiri atas bulan dan tahun merupakan perkiraan dan akan diperbarui dengan tanggal persis ketika diketahui.

Versi utama Db2	Tanggal rilis IBM	Tanggal rilis RDS	Akhir dukungan (dasar) IBM	Akhir dukungan (diperpan jang) IBM	Tanggal akhir dukungan standar RDS
Db2 11.5	27 Juni 2019	27 November 2023	30 September 2025	4 tahun setelah akhir dukungan	

Opsi-opsi pelisensian RDS for Db2

Amazon RDS for Db2 memiliki satu opsi pelisensian: Bawa Lisensi Sendiri (BYOL).

Bawa Lisensi Sendiri

Dalam model BYOL, Anda menggunakan lisensi basis data Db2 milik Anda yang ada untuk mengerahkan basis data di Amazon RDS. Periksa bahwa Anda memiliki lisensi basis data Db2 yang tepat untuk kelas instans basis data dan edisi basis data Db2 yang ingin Anda jalankan. Anda juga harus mengikuti kebijakan IBM untuk pelisensian perangkat lunak basis data IBM di lingkungan komputasi cloud.

> **Note**
>
> Instans DB multi-AZ adalah siaga dingin karena database Db2 diinstal tetapi tidak berjalan. Standbys tidak dapat dibaca, dijalankan, atau melayani permintaan. Untuk informasi selengkapnya, lihat informasi IBM Db2 perizinan di situs web IBM.

Dalam model ini, Anda terus menggunakan akun dukungan IBM aktif Anda, dan menghubungi IBM secara langsung untuk permintaan layanan basis data Db2. Jika Anda memiliki AWS Support akun dengan dukungan kasus, Anda dapat menghubungi AWS Support untuk masalah Amazon RDS.

Amazon Web Services dan IBM memiliki proses dukungan multivendor untuk kasus-kasus yang memerlukan bantuan dari kedua organisasi.

Amazon RDS mendukung model BYOL untuk Db2 Standard Edition dan Db2 Advanced Edition.

Topik

- ID IBM
- Menambahkan ID IBM ke grup parameter
- Integrasi dengan AWS License Manager

ID IBM

Dalam model BYOL, Anda memerlukan IBM Customer ID dan IBM Site ID Anda untuk membuat, mengubah, atau memulihkan instans basis data RDS for Db2. Anda harus membuat grup parameter kustom dengan IBM Customer ID dan IBM Site ID Anda sebelum membuat instans basis data RDS for Db2. Untuk informasi selengkapnya, lihat Menambahkan ID IBM ke grup parameter. Anda dapat menjalankan beberapa instans basis data RDS for Db2 dengan IBM Customer IDs dan IBM Site IDs yang berbeda dalam Akun AWS atau Wilayah AWS yang sama.

Important

Jika Anda adalah IBM Db2 pelanggan yang sudah ada, Anda dapat menemukan IBM Customer ID dan Anda IBM Site ID di sertifikat Bukti Hak Anda dariIBM. Untuk informasi selengkapnya, lihat petunjuk tentang cara melihat situs web Anda IBM Customer ID dan IBM Site ID di situs web IBM.

Jika Anda adalah IBM Db2 pelanggan baru, Anda harus terlebih dahulu membeli lisensi perangkat lunak Db2 dari. IBM Setelah Anda membeli lisensi perangkat lunak Db2, Anda akan menerima Bukti Hak dariIBM, yang mencantumkan Anda dan AndaIBM Customer ID. IBM Site ID

Jika kami tidak dapat memverifikasi lisensi Anda oleh Anda IBM Customer ID dan AndaIBM Site ID, kami dapat menghentikan instans DB apa pun yang berjalan dengan lisensi yang tidak diverifikasi ini.

Menambahkan ID IBM ke grup parameter

Karena Anda tidak dapat mengubah grup parameter default, Anda harus membuat grup parameter kustom dan lalu mengubahnya agar menyertakan nilai-nilai untuk IBM Customer ID dan IBM Site ID

Anda. Lihat informasi yang lebih lengkap tentang grup parameter di Bekerja dengan grup parameter DB dalam instance DB.

> **Important**
>
> Anda harus membuat grup parameter kustom dengan IBM Customer ID dan IBM Site ID Anda sebelum membuat instans basis data RDS for Db2.

Gunakan setelan parameter dalam tabel berikut.

Parameter	Nilai
`rds.ibm_customer_id`	`<your IBM Customer ID>`
`rds.ibm_site_id`	`<your IBM Site ID>`
`ApplyMethod`	`immediate` , `pending-reboot`

Parameter-parameter ini dinamis, yang berarti bahwa setiap perubahannya akan berlaku dengan serta-merta dan Anda tidak perlu mem-boot ulang instans basis data. Jika tidak ingin perubahan berlaku dengan serta-merta, Anda dapat mengatur `ApplyMethod` ke `pending-reboot` dan menjadwalkan perubahan ini agar dibuat selama jendela pemeliharaan.

Anda dapat membuat dan mengubah grup parameter kustom dengan menggunakan AWS Management Console, AWS CLI, atau Amazon RDS API.

Konsol

Untuk menambahkan IBM Customer ID dan IBM Site ID Anda ke grup parameter

1. Buat grup parameter basis data baru. Lihat informasi yang lebih lengkap tentang cara membuat grup parameter basis data di Membuat grup parameter DB.
2. Ubah grup parameter yang Anda buat. Lihat informasi yang lebih lengkap tentang mengubah grup parameter di Memodifikasi parameter dalam grup parameter DB.

AWS CLI

Untuk menambahkan IBM Customer ID dan IBM Site ID Anda ke grup parameter

1. Buat grup parameter kustom dengan menjalankan perintah create-db-parameter-group.

   Sertakan opsi-opsi yang diperlukan berikut:

   - `--db-parameter-group-name` – Nama untuk grup parameter yang sedang Anda buat.
   - `--db-parameter-group-family` – Edisi mesin dan versi utama Db2. Nilai-nilai yang valid: `db2-se-11.5`, `db2-ae-11.5`.
   - `--description` – Deskripsi untuk grup parameter ini.

   Lihat informasi yang lebih lengkap tentang cara membuat grup parameter basis data di Membuat grup parameter DB.
2. Ubah parameter-parameter dalam grup parameter kustom yang Anda buat dengan menjalankan perintah modify-db-parameter-group.

   Sertakan opsi-opsi yang diperlukan berikut:

   - `--db-parameter-group-name` – Nama grup parameter yang Anda buat.
   - `--parameters` – Array nama parameter, nilai parameter, dan metode aplikasi untuk pembaruan parameter.

   Lihat informasi yang lebih lengkap tentang mengubah grup parameter di Memodifikasi parameter dalam grup parameter DB.

RDS API

Untuk menambahkan IBM Customer ID dan IBM Site ID Anda ke grup parameter

1. Buat grup parameter basis data kustom dengan menggunakan operasi CreateDBParameterGroup API Amazon RDS.

   Sertakan parameter-parameter yang diperlukan berikut:

   - `DBParameterGroupName`
   - `DBParameterGroupFamily`

- `Description`

Lihat informasi yang lebih lengkap tentang cara membuat grup parameter basis data di Membuat grup parameter DB.

2. Ubah parameter-parameter dalam grup parameter kustom yang Anda buat dengan menggunakan operasi ModifyDBParameterGroup API RDS.

   Sertakan parameter-parameter yang diperlukan berikut:

   - `DBParameterGroupName`
   - `Parameters`

   Lihat informasi yang lebih lengkap tentang mengubah grup parameter di Memodifikasi parameter dalam grup parameter DB.

Anda kini siap untuk membuat instans basis data dan melampirkan grup parameter kustom ke instans itu. Lihat informasi yang lebih lengkap di Membuat instans DB Amazon RDS dan Mengaitkan grup parameter DB dengan instans DB.

Integrasi dengan AWS License Manager

Untuk membantu memantau RDS untuk penggunaan lisensi Db2 dalam model BYOL, AWS License Managerterintegrasi dengan RDS untuk Db2. License Manager mendukung pelacakan RDS untuk edisi mesin Db2 berdasarkan CPU virtual (VCPU). Anda juga dapat menggunakan License Manager AWS Organizations untuk mengelola semua akun organisasi Anda secara terpusat.

Tabel berikut menunjukkan filter informasi produk untuk RDS untuk Db2.

Filter	Nama	Deskripsi
Edisi Mesin	`db2-se`	Edisi Standar Db2
	`db2-ae`	Edisi Lanjutan Db2

Untuk melacak penggunaan lisensi RDS Anda untuk instans Db2 DB, Anda dapat membuat lisensi yang dikelola sendiri. Dalam hal ini, sumber daya RDS untuk Db2 yang cocok dengan filter informasi

produk secara otomatis dikaitkan dengan lisensi yang dikelola sendiri. Penemuan RDS untuk instans Db2 DB dapat memakan waktu hingga 24 jam.

Konsol

Untuk membuat lisensi yang dikelola sendiri untuk melacak penggunaan lisensi RDS Anda untuk instans Db2 DB

1. Kunjungi https://console.aws.amazon.com/license-manager/.
2. Buat lisensi yang dikelola sendiri.

   Untuk petunjuk, lihat Membuat lisensi yang dikelola sendiri di Panduan AWS License Manager Pengguna.

   Tambahkan aturan untuk Filter Informasi Produk RDS dalam panel Informasi Produk.

   Untuk informasi selengkapnya, lihat ProductInformationdi Referensi AWS License Manager API.

AWS CLI

Untuk membuat lisensi yang dikelola sendiri dengan menggunakan AWS CLI, panggil create-license-configurationperintah. Gunakan parameter `--cli-input-json` atau `--cli-input-yaml` untuk meneruskan parameter ke perintah.

Example

Kode berikut membuat lisensi yang dikelola sendiri untuk Db2 Standard Edition.

```
aws license-manager create-license-configuration --cli-input-json file://rds-db2-
se.json
```

Berikut adalah sampel file `rds-db2-se.json` yang digunakan dalam contoh.

```
{
    "Name": "rds-db2-se",
    "Description": "RDS Db2 Standard Edition",
    "LicenseCountingType": "vCPU",
    "LicenseCountHardLimit": false,
    "ProductInformationList": [
        {
            "ResourceType": "RDS",
```

```
            "ProductInformationFilterList": [
                {
                    "ProductInformationFilterName": "Engine Edition",
                    "ProductInformationFilterValue": ["db2-se"],
                    "ProductInformationFilterComparator": "EQUALS"
                }
            ]
        }
    ]
}
```

Untuk informasi selengkapnya tentang informasi produk, lihat Penemuan otomatis inventaris sumber daya dalam Panduan Pengguna AWS License Manager .

Untuk informasi selengkapnya tentang *--cli-input* parameter, lihat Menghasilkan AWS CLI kerangka dan parameter input dari file input JSON atau YAMM di Panduan Pengguna.AWS CLI

Kelas-kelas instans RDS for Db2

Penghitungan dan kapasitas memori instans basis data ditentukan oleh kelas instansnya. Kelas instans basis data yang Anda butuhkan bergantung pada daya pemrosesan dan kebutuhan memori Anda.

Kelas-kelas instans RDS for Db2 yang didukung

Kelas-kelas instans RDS for Db2 yang didukung merupakan subkumpulan kelas-kelas instans basis data Amazon RDS. Lihat daftar lengkap kelas instans Amazon RDS di Kelas instans DB .

Tabel berikut memerinci kelas-kelas instans yang didukung untuk basis data Db2 11.5.9.0.

Edisi Db2	Versi Db2 11.5.9.0
Db2 Standard Edition Bawa Lisensi Sendiri (BYOL)	Kelas-kelas instans tujuan umum dengan prosesor Intel Xeon Scalable Generasi ke-3, penyimpanan SSD, dan pengoptimalan jaringan
	db.m6idn.large–db.m6idn.8xlarge
	Kelas-kelas instans tujuan umum yang ditenagai oleh prosesor Intel Xeon Scalable generasi ke-3

Edisi Db2	Versi Db2 11.5.9.0
	db.m6in.large–db.m6in.8xlarge
	Kelas-kelas instans tujuan umum
	db.m6i.large–db.m6i.8xlarge
	Kelas-kelas instans teroptimalkan secara memori dengan SSD berbasis NVMe lokal, ditenagai oleh prosesor Intel Xeon Scalable generasi ke-3
	db.x2iedn.xlarge
	Kelas-kelas instans teroptimalkan secara memori yang ditenagai oleh prosesor Intel Xeon Scalable generasi ke-3
	db.r6idn.large–db.r6idn.4xlarge
	Kelas-kelas instans teroptimalkan secara memori yang ditenagai oleh prosesor Intel Xeon Scalable generasi ke-3
	db.r6in.large–db.r6in.4xlarge
	Kelas-kelas instans teroptimalkan secara memori
	db.r6i.large–db.r6i.4xlarge
	Kelas-kelas instans dengan kinerja dapat dilonjakkan
	db.t3.small–db.t3.2xlarge
Db2 Advanced Edition Bawa Lisensi Sendiri (BYOL)	Kelas-kelas instans tujuan umum dengan prosesor Intel Xeon Scalable Generasi ke-3, penyimpanan SSD, dan pengoptimalan jaringan
	db.m6idn.large–db.m6idn.8xlarge
	Kelas-kelas instans tujuan umum yang ditenagai oleh prosesor Intel Xeon Scalable generasi ke-3

Edisi Db2	Versi Db2 11.5.9.0
	db.m6in.large–db.m6in.8xlarge
	Kelas-kelas instans tujuan umum
	db.m6i.12xlarge–db.m6i.32xlarge
	Kelas-kelas instans teroptimalkan secara memori dengan SSD berbasis NVMe lokal, ditenagai oleh prosesor Intel Xeon Scalable generasi ke-3
	db.x2iedn.2xlarge–db.x2iedn.32xlarge
	Kelas-kelas instans teroptimalkan secara memori yang ditenagai oleh prosesor Intel Xeon Scalable generasi ke-3
	db.r6idn.8xlarge–db.r6idn.32xlarge
	Kelas-kelas instans teroptimalkan secara memori yang ditenagai oleh prosesor Intel Xeon Scalable generasi ke-3
	db.r6in.8xlarge–db.r6in.32xlarge
	Kelas-kelas instans teroptimalkan secara memori
	db.r6i.8xlarge–db.r6i.32xlarge

Parameter-parameter RDS for Db2

RDS for Db2 mendukung pengubahan parameter-parameter manajer basis data (tingkat instans) dan parameter-parameter registri Db2 melalui grup parameter. Parameter-parameter basis data hanya dapat diubah melalui prosedur tersimpan rdsadmin.update_db_param.

Secara bawaan, instans basis data RDS for Db2 menggunakan grup parameter basis data yang spesifik untuk basis data dan instans basis data Db2. Grup parameter ini berisi parameter-parameter untuk mesin basis data IBM Db2. Lihat informasi tentang cara menangani grup parameter dan mengatur parameter di Bekerja dengan grup parameter.

Parameter-parameter RDS for Db2 diatur ke nilai bawaan mesin penyimpanan yang Anda pilih. Lihat informasi yang lebih lengkap tentang aneka parameter Db2 di Parameter-parameter konfigurasi basis data Db2 dalam dokumentasi IBM Db2.

Anda dapat melihat parameter yang tersedia untuk versi Db2 tertentu menggunakan AWS Management Console atau AWS Command Line Interface ()AWS CLI. Lihat informasi tentang cara menampilkan di konsol parameter-parameter dalam grup parameter Db2 di Melihat nilai parameter untuk grup parameter DB.

Dengan menggunakan AWS CLI, Anda dapat melihat parameter untuk versi Db2 dengan menjalankan perintah. describe-engine-default-parameters Tentukan salah satu dari nilai-nilai berikut untuk opsi `--db-parameter-group-family`:

- `db2-ae-11.5`
- `db2-se-11.5`

Misalnya, untuk melihat parameter-parameter untuk Db2 Standard Edition 11.5, jalankan perintah berikut.

```
aws rds describe-engine-default-parameters --db-parameter-group-family db2-se-11.5
```

Perintah ini menghasilkan output yang serupa dengan contoh berikut.

```
{
    "EngineDefaults": {
        "Parameters": [
            {
                "ParameterName": "agent_stack_sz",
                "ParameterValue": "1024",
                "Description": "You can use this parameter to determine the amount of
 memory that is allocated by Db2 for each agent thread stack.",
                "Source": "engine-default",
                "ApplyType": "static",
                "DataType": "integer",
                "AllowedValues": "256-32768",
                "IsModifiable": false
            },
            {
                "ParameterName": "agentpri",
                "ParameterValue": "-1",
```

```
                "Description": "This parameter controls the priority given to all
 agents and to other database manager instance processes and threads by the operating
 system scheduler. This priority determines how CPU time is allocated to the database
 manager processes, agents, and threads relative to other processes and threads running
 on the machine.",
                "Source": "engine-default",
                "ApplyType": "static",
                "DataType": "integer",
                "AllowedValues": "1-99",
                "IsModifiable": false
            },
            ...
        ]
    }
}
```

Untuk memerinci hanya parameter-parameter yang dapat diubah untuk Db2 Standard Edition 11.5, jalankan perintah berikut:

Untuk Linux, macOS, atau Unix:

```
aws rds describe-engine-default-parameters \
    --db-parameter-group-family db2-se-11.5 \
    --query 'EngineDefaults.Parameters[?IsModifiable==`true`].
{ParameterName:ParameterName, DefaultValue:ParameterValue}'
```

Untuk Windows:

```
aws rds describe-engine-default-parameters ^
    --db-parameter-group-family db2-se-11.5 ^
    --query 'EngineDefaults.Parameters[?IsModifiable==`true`].
{ParameterName:ParameterName, DefaultValue:ParameterValue}'
```

Topik

- Menentukan parameter yang dapat diubah
- Mengubah parameter

Menentukan parameter yang dapat diubah

Untuk menentukan parameter-parameter manajer basis data, basis data, dan registri yang dapat Anda ubah, jalankan perintah berikut.

1. Connect ke database Db2 Anda. Dalam contoh berikut, ganti *database_name*, *master_username*, dan *master_password* dengan informasi Anda sendiri.

```
db2 "connect to database_name user master_username using master_password"
```

2. Temukan versi Db2 yang didukung.

```
db2 "select service_level, fixpack_num from table(sysproc.env_get_inst_info()) as
 instanceinfo"
```

3. Lihat parameter-parameter untuk versi Db2 tertentu.

   - Lihat parameter-parameter konfigurasi manajer basis data. Periksa grup parameter yang dilampirkan ke instans DB Anda dengan menggunakan AWS Management Console atau dengan menjalankan perintah berikut:

```
db2 "select cast(substr(name,1,24) as varchar(24)) as name, case
    when value_flags = 'NONE' then '' else value_flags end flags,
    cast(substr(value,1,64) as varchar(64)) as current_value
    from sysibmadm.dbmcfg
    order by name asc with UR"
```

   - Lihat semua parameter konfigurasi basis data Anda.

```
db2 "select cast(substr(name,1,24) as varchar(24)) as name, case
    when value_flags = 'NONE' then '' else value_flags end flags,
    cast(substr(value,1,64) as varchar(64)) as current_value
    from table(db_get_cfg(null)) order by name asc, member asc with UR"
```

   - Lihat variabel-variabel registri yang saat ini ditetapkan.

```
db2 "select cast(substr(reg_var_name,1,50) as varchar(50)) as reg_var_name,
    cast(substr(reg_var_value,1,50) as varchar(50)) as reg_var_value,
    level from table(env_get_reg_variables(null))
    order by reg_var_name,member with UR"
```

   - Lihat daftar semua variabel registri yang didukung.

```
db2 "select cast(substr(reg_var_name,1,50) as varchar(50)) as reg_var_name,
    cast(substr(reg_var_value,1,50) as varchar(50)) as reg_var_value,
    level from table(env_get_reg_variables(null,1))
    order by reg_var_name,member with UR"
```

Mengubah parameter

Anda dapat mengubah parameter-parameter manajer basis data dan registri dalam grup parameter kustom. Pertama, buat grup parameter kustom, lalu ubah parameter-parameter dalam grup parameter kustom tersebut. Untuk informasi selengkapnya, lihat Bekerja dengan grup parameter DB dalam instance DB.

Untuk mengubah parameter-parameter basis data, jalankan perintah berikut.

1. Hubungi basis data `rdsadmin`. Dalam contoh berikut, ganti *`master_username`* dan *`master_password`* dengan informasi Anda.

   ```
   db2 "connect to rdsadmin user master_username using master_password"
   ```

2. Ubah parameter-parameter basis data dengan memanggil prosedur tersimpan `rdsadmin.update_db_param`. Lihat informasi yang lebih lengkap di rdsadmin.update_db_param.

   ```
   db2 "call rdsadmin.update_db_param(
       'database_name',
       'parameter_to_modify',
       'changed_value')"
   ```

Pemeriksaan EBCDIC untuk database Db2 di Amazon RDS

RDS untuk Db2 mendukung pemeriksaan EBCDIC untuk database Db2. Anda hanya dapat menentukan urutan pemeriksaan EBCDIC untuk database saat membuat database menggunakan prosedur tersimpan Amazon the section called “rdsadmin.create_database” RDS.

Saat Anda membuat RDS untuk instans Db2 DB dengan menggunakanAWS Management Console,AWS CLI, atau RDS API, Anda dapat menentukan nama database. Jika Anda menentukan nama database, Amazon RDS membuat database dengan pemeriksaan default. `SYSTEM` Jika Anda

perlu membuat database dengan pemeriksaan EBCDIC, jangan tentukan nama database saat Anda membuat instance DB.

Pengumpulan untuk database di RDS untuk Db2 diatur pada saat pembuatan dan tidak dapat diubah. Jika Anda menentukan nama database ketika Anda membuat instance DB dan Anda ingin database dengan pemeriksaan EBCDIC, hapus instans DB dan buat yang baru.

Untuk membuat database Db2 dengan pemeriksaan EBCDIC

1. Buat RDS untuk instans Db2 DB tanpa menentukan nama database dengan menggunakanAWS Management Console,AWS CLI, atau RDS API. Untuk informasi selengkapnya, lihat Membuat instans DB.
2. Buat database Db2 dan atur opsi pemeriksaan ke nilai EBCDIC dengan memanggil prosedur yang disimpan. `rdsadmin.create_database` Untuk informasi selengkapnya, lihat rdsadmin.create_database.

> **Important**
>
> Setelah Anda membuat database menggunakan prosedur tersimpan, Anda tidak dapat mengubah urutan pemeriksaan. Jika Anda ingin database menggunakan urutan pemeriksaan yang berbeda, jatuhkan database dengan memanggil prosedur yang the section called “rdsadmin.drop_database” disimpan. Kemudian, buat database dengan urutan pemeriksaan yang diperlukan.

Prasyarat untuk membuat instans basis data RDS for Db2

Butir-butir berikut adalah prasyarat sebelum membuat instans basis data.

Topik

Akun Administrator

Saat membuat instans basis data, Anda harus menetapkan akun administrator untuk instans itu. Amazon RDS memberikan wewenang `ACCESSCTRL` kepada akun administrator basis data lokal ini.

Akun administrator memiliki berbagai karakteristik, kemampuan, dan keterbatasan berikut:

- Seorang pengguna lokal dan bukan Akun AWS.
- Tidak memiliki otoritas tingkat instans Db2 seperti `SYSADM`, `SYSMAINT`, atau `SYSCTRL`.
- Tidak dapat menghentikan atau memulai instans Db2.
- Tidak dapat mengedrop basis data Db2 jika Anda menentukan nama saat membuat instans basis data.
- Memiliki akses penuh ke basis data Db2 yang meliputi tabel dan tampilan katalog.
- Dapat membuat pengguna dan grup lokal dengan menggunakan prosedur tersimpan Amazon RDS.
- Dapat memberikan dan mencabut otoritas dan privilese.

Akun administrator dapat melakukan tugas-tugas berikut:

- Membuat, mengubah, atau menghapus instans basis data.
- Membuat cuplikan basis data.
- Memulai point-in-time mengembalikan.
- Membuat cadangan otomatis cuplikan basis data.
- Membuat cadangan manual cuplikan basis data.
- Menggunakan fitur-fitur Amazon RDS lainnya.

Pertimbangan tambahan

Sebelum membuat instance DB, pertimbangkan item berikut:

- Setiap instans basis data RDS for Db2 dapat menampung satu basis data Db2.
- Nama basis data awal
  - Jika Anda tidak memberikan nama basis data saat membuat instans basis data, Amazon RDS tidak membuat basis data.
  - Jangan berikan nama database dalam keadaan berikut:
    - Anda ingin menggunakan prosedur tersimpan Amazon RDS untuk membuat atau menjatuhkan database.
    - Anda ingin membuat database yang menggunakan urutan pemeriksaan EBCDIC. Untuk informasi selengkapnya, lihat Pemeriksaan EBCDIC untuk database Db2 di Amazon RDS.
    - Anda ingin memulihkan cadangan dari Amazon S3.
    - Anda bermigrasi dari AIX atauWindows. Untuk informasi selengkapnya, lihat Migrasi satu kali dari lingkungan AIX atau Windows ke Linux.
- Dalam model Bawa Lisensi Sendiri (BYOL), Anda harus membuat dahulu sebuah grup parameter kustom yang berisi IBM Customer ID dan IBM Site ID. Lihat informasi yang lebih lengkap di Bawa Lisensi Sendiri.

Menghubungkan ke instans DB RDS untuk Db2 Anda

Setelah Amazon RDS menyediakan instans DB RDS untuk Db2, Anda dapat menggunakan aplikasi klien SQL standar apa pun untuk terhubung ke instans DB. Karena Amazon RDS adalah layanan terkelola, Anda tidak dapat masuk sebagai`SYSADM`, `SYSCTRL`, `SECADM`, atau`SYSMAINT`.

Anda dapat terhubung ke instans DB yang menjalankan mesin basis data IBM Db2 dengan menggunakan IBM Db2 CLP, IBM CLPPlus, DBeaver, atauIBM Db2 Data Management Console.

Topik

Menemukan titik akhir instans basis data RDS for Db2 Anda

Setiap instans basis data Amazon RDS memiliki titik akhir, dan setiap titik akhir memiliki nama dan nomor porta DNS untuk instans basis data. Untuk menghubungkan instans basis Anda dengan aplikasi klien SQL, Anda memerlukan nama dan nomor porta DNS untuk instans basis data itu.

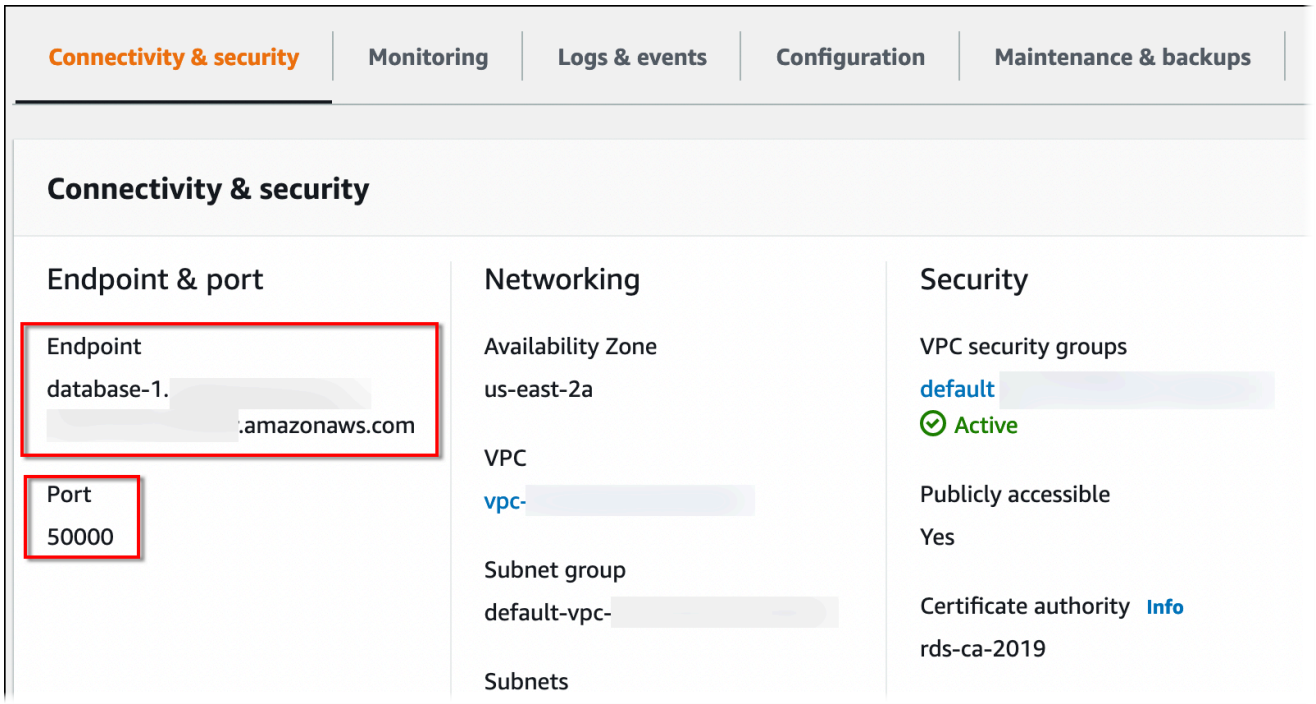
Anda dapat menemukan titik akhir instans basis data dengan menggunakan AWS Management Console atau AWS CLI.

Konsol

Untuk menemukan titik akhir suatu instans basis data RDS for Db2

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di sudut kanan atas konsol, pilih Wilayah AWS untuk instans basis data Anda.
3. Temukan nama dan nomor porta DNS untuk instans basis data RDS for Db2 Anda.
    a. Pilih Basis data untuk menampilkan daftar instans basis data Anda.
    b. Pilih nama instans basis data RDS for Db2 untuk menampilkan detail instans.

c. Pada tab Konektivitas dan keamanan, salin titik akhir. Selain itu, catat nomor porta. Anda memerlukan titik akhir dan nomor porta untuk menghubungi instans basis data.



AWS CLI

Untuk menemukan titik akhir suatu instans basis data RDS for Db2, jalan perintah describe-db-instances. Dalam contoh berikut, ganti *database-1* dengan nama instans basis data Anda.

Untuk Linux, macOS, atau Unix:

```
aws rds describe-db-instances \
    --db-instance-identifier database-1 \
    --query 'DBInstances[].
{DBInstanceIdentifier:DBInstanceIdentifier,DBName:DBName,Endpoint:Endpoint}' \
    --output json
```

Untuk Windows:

```
aws rds describe-db-instances ^
    --db-instance-identifier database-1 ^
    --query 'DBInstances[].
{DBInstanceIdentifier:DBInstanceIdentifier,DBName:DBName,Endpoint:Endpoint}' ^
    --output json
```

Perintah ini menghasilkan output yang serupa dengan contoh berikut. Baris `Address` di dalam output berisi nama DNS.

```
[
    {
        "DBInstanceIdentifier": "database-1",
        "DBName": "DB2DB",
        "Endpoint": {
            "Address": "database-1.123456789012.us-east-2.amazonaws.com",
            "Port": 50000,
            "HostedZoneId": "Z2OC4A7DETW6VH"
        }
    }
]
```

Menghubungi instans basis data RDS for Db2 dengan IBM Db2 CLP

Anda dapat menggunakan utilitas baris perintah seperti IBM Db2 CLP untuk menghubungi instans basis data Amazon RDS for Db2. Utilitas ini adalah bagian dari IBM Data Server Runtime Client. Untuk mengunduh klien dari IBM Fix Central, lihat Paket Klien Server Data IBM Versi 11.5 Mod 8 Fix Pack 0 dalam Dukungan IBM.

Topik

- Terminologi
- Menginstal klien
- Menghubungi instans basis data
- Memecahkan masalah koneksi dengan instans basis data RDS for Db2 Anda

Terminologi

Istilah-istilah berikut membantu menjelaskan perintah-perintah yang digunakan saat menghubungi instans basis data RDS for Db2 Anda.

buat katalog simpul tcpip

> Perintah ini mendaftarkan simpul basis data jauh di klien Db2 lokal, yang membuat simpul dapat diakses oleh aplikasi klien. Untuk membuat katalog sebuah simpul, Anda memberikan informasi seperti nama host server, nomor porta, dan protokol komunikasi. Simpul yang dikatalogkan kemudian mewakili server target tempat berada satu atau beberapa basis data jauh. Lihat

informasi yang lebih lengkap di perintah CATALOG TCPIP/TCPIP4/TCPIP6 NODE dalam dokumentasi IBM Db2.

buat katalog basis data

Perintah ini mendaftarkan basis data jauh di klien Db2 lokal, yang membuat basis data dapat diakses oleh aplikasi klien. Untuk membuat katalog sebuah basis data, Anda memberikan informasi seperti alias basis data, simpul tempat basis data berada, dan jenis autentikasi yang diperlukan untuk menghubungi basis data. Lihat informasi yang lebih lengkap di perintah CATALOG DATABASE dalam dokumentasi IBM Db2.

Menginstal klien

Setelah itudownloading the package for Linux, instal klien menggunakan hak root atau administrator.

> **Note**
>
> Untuk menginstal klien pada AIX atau Windows, ikuti prosedur yang sama, tetapi sesuaikan perintah untuk sistem operasi Anda.

Untuk menginstal klien di Linux

1. Jalankan **`./db2_install -f sysreq`** dan pilih **`yes`** untuk menerima lisensi.
2. Pilih lokasi untuk menginstal klien.
3. Jalankan **`clientInstallDir/instance/db2icrt -s client`***`instance_name`*. Ganti *`instance_name`* dengan pengguna sistem operasi yang valid di Linux. Di Linux, nama instans basis data Db2 dikaitkan dengan nama pengguna sistem operasi.

   Perintah ini membuat direktori **`sqllib`** di bawah direktori home pengguna yang ditunjuk di Linux.

Menghubungi instans basis data

Untuk menghubungi instans basis data RDS for Db2, Anda memerlukan nama dan nomor porta DNS. Lihat informasi tentang cara menemukan info itu di Menemukan titik akhir. Anda juga perlu mengetahui nama basis data, nama pengguna master, dan kata sandi master yang Anda tentukan saat membuat instans basis data RDS for Db2. Lihat informasi yang lebih lengkap tentang cara menemukan info itu di Membuat instans DB.

Untuk menghubungi instans basis data RDS for Db2 dengan IBM Db2 CLP

1. Masuk dengan nama pengguna yang Anda tentukan selama instalasi klien IBM Db2 CLP.
2. Buat katalog instans basis data RDS for Db2 Anda. Dalam contoh berikut, ganti *node_name*, *dns_name*, dan *port* dengan nama untuk simpul dalam katalog lokal, nama DNS untuk instans basis data Anda, dan nomor porta.

   ```
   db2 catalog TCPIP node node_name remote dns_name server port
   ```

   Contoh

   ```
   db2 catalog TCPIP node remnode remote database-1.123456789012.us-
   east-1.amazonaws.com server 50000
   ```

3. Buat katalog basis data `rdsadmin` dan basis data Anda. Ini akan memungkinkan Anda menghubungi basis data `rdsadmin` untuk melakukan tugas-tugas administratif dengan menggunakan prosedur tersimpan Amazon RDS. Untuk informasi selengkapnya, lihat Mengadministrasikan instans basis data RDS for Db2 Anda.

   Dalam contoh berikut, ganti *database_alias*, *node_name*, dan *database_name* dengan alias untuk basis data ini, nama simpul yang ditentukan pada langkah sebelumnya, dan nama basis data Anda. `server_encrypt`mengenkripsi nama pengguna dan kata sandi Anda saat melalui jaringan.

   ```
   db2 catalog database rdsadmin [ as database_alias ] at node node_name
    authentication server_encrypt

   db2 catalog database database_name [ as database_alias ] at node node_name
    authentication server_encrypt
   ```

   Contoh

   ```
   db2 catalog database rdsadmin at node remnode authentication server_encrypt

   db2 catalog database testdb as rdsdb2 at node remnode authentication server_encrypt
   ```

4. Hubungi basis data RDS for Db2 Anda. Dalam contoh berikut, ganti *rds_database_alias*, *master_username*, dan *master_password* dengan nama basis data Anda, nama pengguna master, dan kata sandi master instans basis data RDS for Db2 Anda.

```
db2 connect to rds_database_alias user master_username using master_password
```

Perintah ini menghasilkan output yang serupa dengan contoh berikut:

```
Database Connection Information

    Database server        = DB2/LINUXX8664 11.5.9.0
    SQL authorization ID   = ADMIN
    Local database alias   = TESTDB
```

5. Jalankan kueri dan lihat hasil. Contoh berikut menunjukkan pernyataan SQL yang memilih basis data yang Anda buat.

```
db2 "select current server from sysibm.dual"
```

Perintah ini menghasilkan output yang serupa dengan contoh berikut:

```
1
    ------------------
    TESTDB

    1 record(s) selected.
```

Memecahkan masalah koneksi dengan instans basis data RDS for Db2 Anda

Jika Anda menerima kesalahan `NULLID` berikut, itu biasanya menunjukkan bahwa versi klien Anda dan versi server RDS for Db2 tidak cocok. Lihat versi-versi klien Db2 yang didukung di Kombinasi klien, driver, dan level server yang didukung dalam dokumentasi IBM Db2.

```
db2 "select * from syscat.tables"
SQL0805N Package "NULLID.SQLC2O29 0X4141414141454A69" was not found.
SQLSTATE=51002
```

Setelah menerima kesalahan ini, Anda harus mengikat paket dari klien Db2 lama Anda ke versi server Db2 yang didukung oleh RDS for Db2.

Untuk mengikat paket dari klien Db2 yang lebih lama pada server Db2 yang lebih baru

1. Temukan file-file pengikat pada mesin klien. File-file ini biasanya terletak di direktori bnd dari jalur instalasi klien Db2 dan memiliki ekstensi .bnd.
2. Hubungi server Db2. Dalam contoh berikut, ganti *database_name* dengan nama basis data Db2 Anda. Ganti *master_username* dan *master_password* dengan informasi Anda. Pengguna ini memiliki otoritas DBADM.

   ```
   db2 connect to database_name user master_username using master_password
   ```

3. Jalankan perintah bind untuk mengikat paket.

   a. Arahkan ke direktori tempat file pengikat berada di mesin klien.

   b. Jalankan perintah bind untuk setiap file.

   Opsi-opsi berikut diperlukan:

   - blocking all – Mengikat semua paket dalam file pengikat pada satu permintaan basis data.
   - grant public – Memberikan izin ke public untuk mengeksekusi paket.
   - sqlerror continue – Menetapkan bahwa proses bind berlanjut sekalipun terjadi kesalahan.

   Lihat informasi yang lebih lengkap tentang perintah bind di perintah BIND dalam dokumentasi IBM Db2.
4. Periksa bahwa pengikatan berhasil dengan mengueri tampilan katalog syscat.package atau memeriksa pesan yang dihasilkan setelah perintah bind.

Lihat informasi yang lebih lengkap di Daftar File Pengikatan dan Nama Paket Db2 v11.5 dalam Dukungan IBM.

Menghubungi instans basis data RDS for Db2 dengan IBM CLPPlus

Anda dapat menggunakan utilitas seperti IBM CLPPlus untuk menghubungi instans basis data Amazon RDS for Db2. Utilitas ini adalah bagian dari IBM Data Server Runtime Client. Untuk mengunduh klien dari IBM Fix Central, lihat Paket Klien Server Data IBM Versi 11.5 Mod 8 Fix Pack 0 dalam Dukungan IBM.

> **Important**
>
> Sebaiknya jalankan IBM CLPPlus pada sebuah sistem operasi yang mendukung antarmuka pengguna grafis seperti macOS, Windows, atau Linux dengan Desktop. Jika menjalankan Linux tanpa antarmuka, gunakan opsi -nw dengan perintah CLPPlus.

Topik

Menginstal klien

Setelah mengunduh paket untuk Linux, instal klien.

> **Note**
>
> Untuk menginstal klien pada AIX atau Windows, ikuti prosedur yang sama, tetapi sesuaikan perintah untuk sistem operasi Anda.

Untuk menginstal klien di Linux

1. Jalankan **`./db2_install`**.
2. Jalankan **`clientInstallDir/instance/db2icrt -s client`***`instance_name`*. Ganti *`instance_name`* dengan pengguna sistem operasi yang valid di Linux. Di Linux, nama instans basis data Db2 dikaitkan dengan nama pengguna sistem operasi.

   Perintah ini membuat direktori **`sqllib`** di bawah direktori home pengguna yang ditunjuk di Linux.

Menghubungi instans basis data

Untuk menghubungi instans basis data RDS for Db2, Anda memerlukan nama dan nomor porta DNS. Lihat informasi tentang cara menemukan info itu di Menemukan titik akhir. Anda juga perlu mengetahui nama basis data, nama pengguna master, dan kata sandi master yang Anda tentukan saat membuat instans basis data RDS for Db2. Lihat informasi yang lebih lengkap tentang cara menemukan info itu di Membuat instans DB.

Untuk menghubungi instans basis data RDS for Db2 dengan IBM CLPPlus

1. Tinjau sintaks perintah. Dalam contoh berikut, ganti *clientDir* dengan lokasi tempat klien diinstal.

```
cd clientDir/bin
    ./clpplus -h
```

2. Konfigurasikan server Db2 Anda. Dalam contoh berikut, ganti *dns_name*, *database_name*, *endpoint*, dan *port* dengan nama DNS, nama basis data, titik akhir, dan porta untuk instans basis data RDS for Db2 Anda. Untuk informasi selengkapnya, lihat Menemukan titik akhir instans basis data RDS for Db2 Anda.

```
db2cli writecfg add -dsn dns_name -database database_name -host endpoint -port port
 -parameter "Authentication=SERVER_ENCRYPT"
```

3. Hubungi instans basis data RDS for Db2 Anda. Dalam contoh berikut, ganti *master_username* dan *dns_name* dengan nama pengguna master dan nama DNS.

```
./clpplus -nw master_username@dns_name
```

4. Jendela Java Shell terbuka. Masukkan kata sandi master untuk instans Db2 basis data Anda.

> **Note**
>
> Jika jendela Java Shell tidak membuka, jalankan **`./clpplus -nw`** untuk menggunakan jendela baris perintah yang sama.

```
Enter password: *********
```

Koneksi terbentuk dan menghasilkan output yang serupa dengan contoh berikut:

```
Database Connection Information :
---------------------------------
Hostname = database-1.abcdefghij.us-east-1.rds.amazonaws.com
Database server = DB2/LINUXX8664  SQL110590
SQL authorization ID = admin
Local database alias = DB2DB
```

```
Port = 50000
```

5. Jalankan kueri dan lihat hasil. Contoh berikut menunjukkan pernyataan SQL yang memilih basis data yang Anda buat.

```
SQL > select current server from sysibm.dual;
```

Perintah ini menghasilkan output yang serupa dengan contoh berikut:

```
1
    --------------------
    DB2DB
    SQL>
```

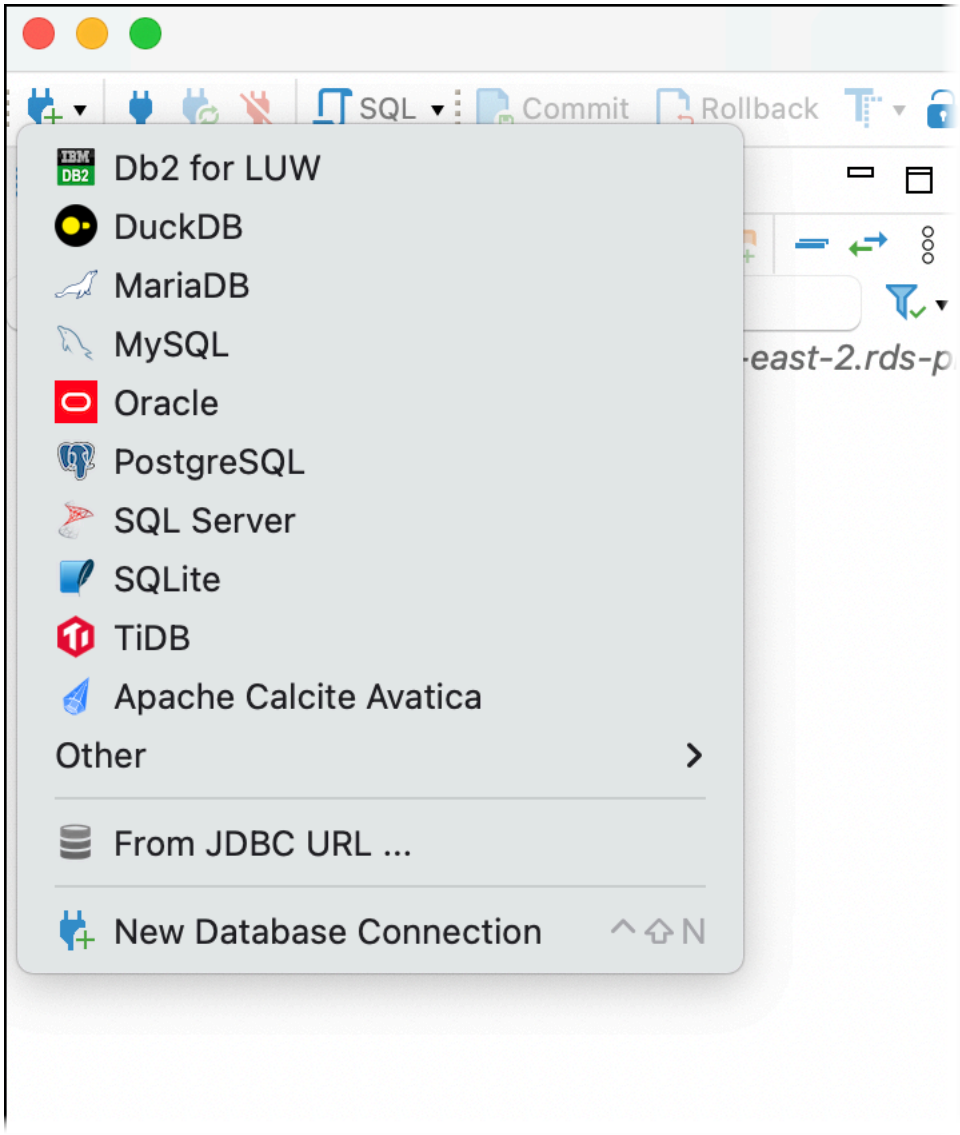
Menghubungi instans basis data RDS for Db2 dengan DBeaver

Anda dapat menggunakan alat-alat pihak ketiga seperti DBeaver untuk menghubungi instans basis data Amazon RDS for Db2. Untuk mengunduh utilitas ini, lihat Komunitas DBeaver.

Untuk menghubungi instans basis data RDS for Db2, Anda memerlukan nama dan nomor porta DNS. Lihat informasi tentang cara menemukan info itu di Menemukan titik akhir. Anda juga perlu mengetahui nama basis data, nama pengguna master, dan kata sandi master yang Anda tentukan saat membuat instans basis data RDS for Db2. Lihat informasi yang lebih lengkap tentang cara menemukan info itu di Membuat instans DB.

Untuk menghubungi instans basis data RDS for Db2 dengan DBeaver

1. Mulai DBeaver.
2. Pilih ikon Koneksi Baru di bilah alat, lalu pilih Db2 for LUW.



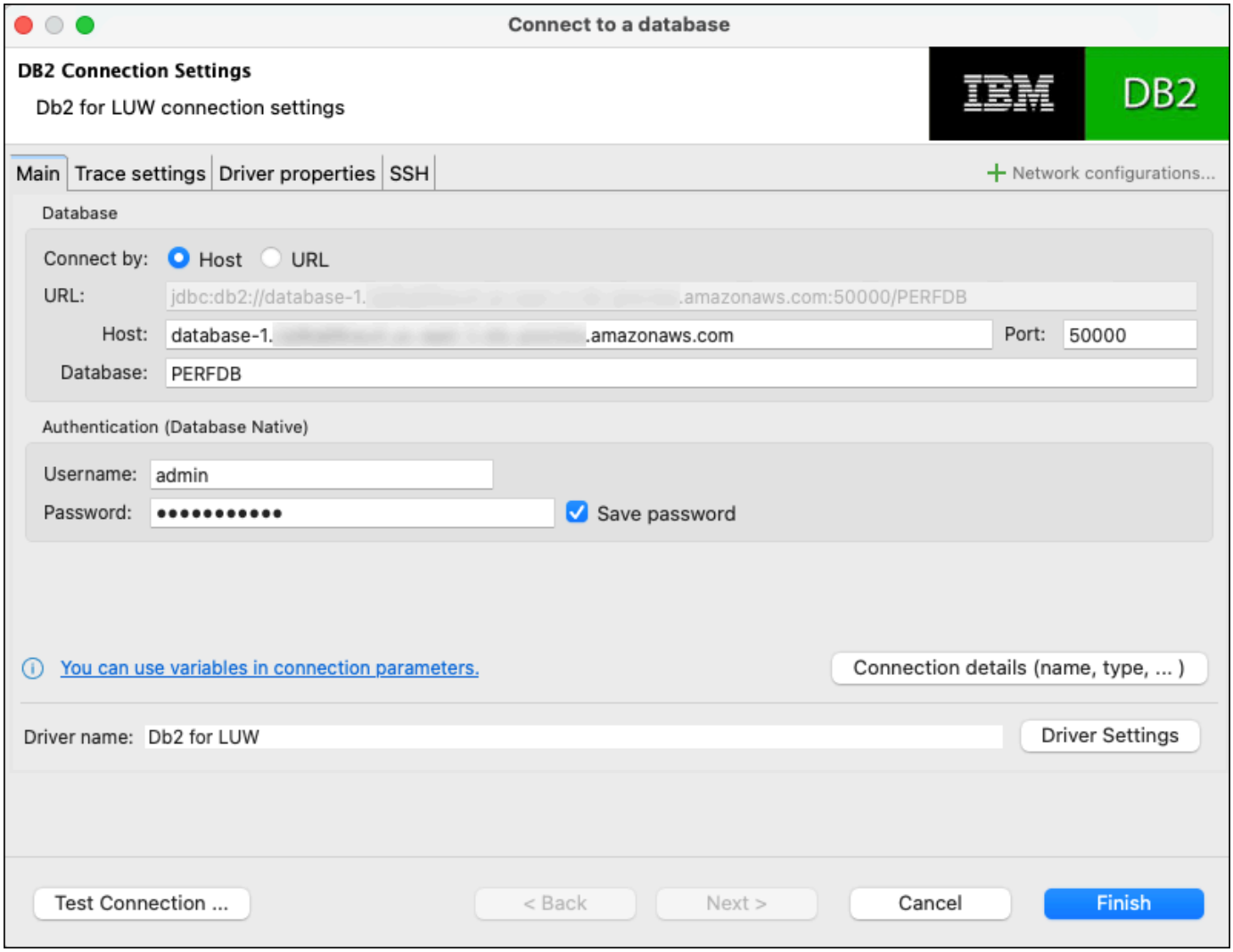
3. Di jendela Hubungi basis data, berikan informasi instans basis data RDS for Db2 Anda.

   a. Masukkan informasi berikut:

      - Untuk Host, masukkan nama DNS instans basis data.
      - Untuk Porta, masukkan nomor porta untuk instans basis data.
      - Untuk Basis Data, masukkan nama basis data.
      - Untuk Nama Pengguna, masukkan nama administrator basis data untuk instans basis data.
      - Untuk Kata Sandi, masukkan kata sandi administrator basis data untuk instans basis data.

   b. Pilih Simpan kata sandi.

   c. Pilih Pengaturan Driver.



4. Di jendela Edit Driver, tentukan properti keamanan tambahan.

   a. Pilih tab Properti driver.

   b. Tambahkan dua Properti Pengguna.

      i. Buka menu konteks (klik kanan), lalu pilih Tambahkan properti baru.

      ii. Untuk Nama Properti, tambahkan encryptionAlgorithm, lalu pilih Oke.

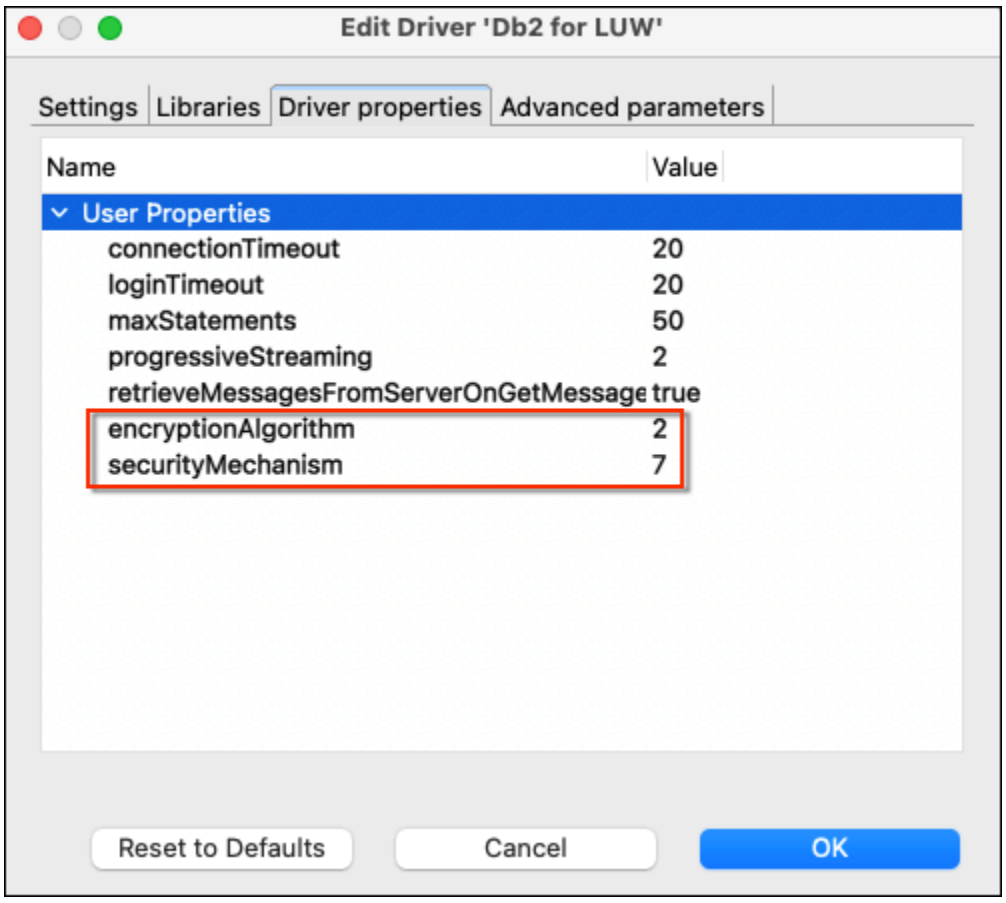
      iii. Dengan baris encryptionAlgorithm dipilih, pilih kolom Nilai dan tambahkan 2.

      iv. Buka menu konteks (klik kanan), lalu pilih Tambahkan properti baru.

      v. Untuk Nama Properti, tambahkan securityMechanism, lalu pilih Oke.

      vi. Dengan baris securityMechanism dipilih, pilih kolom Nilai dan tambahkan 7.

   c. Pilih OK.

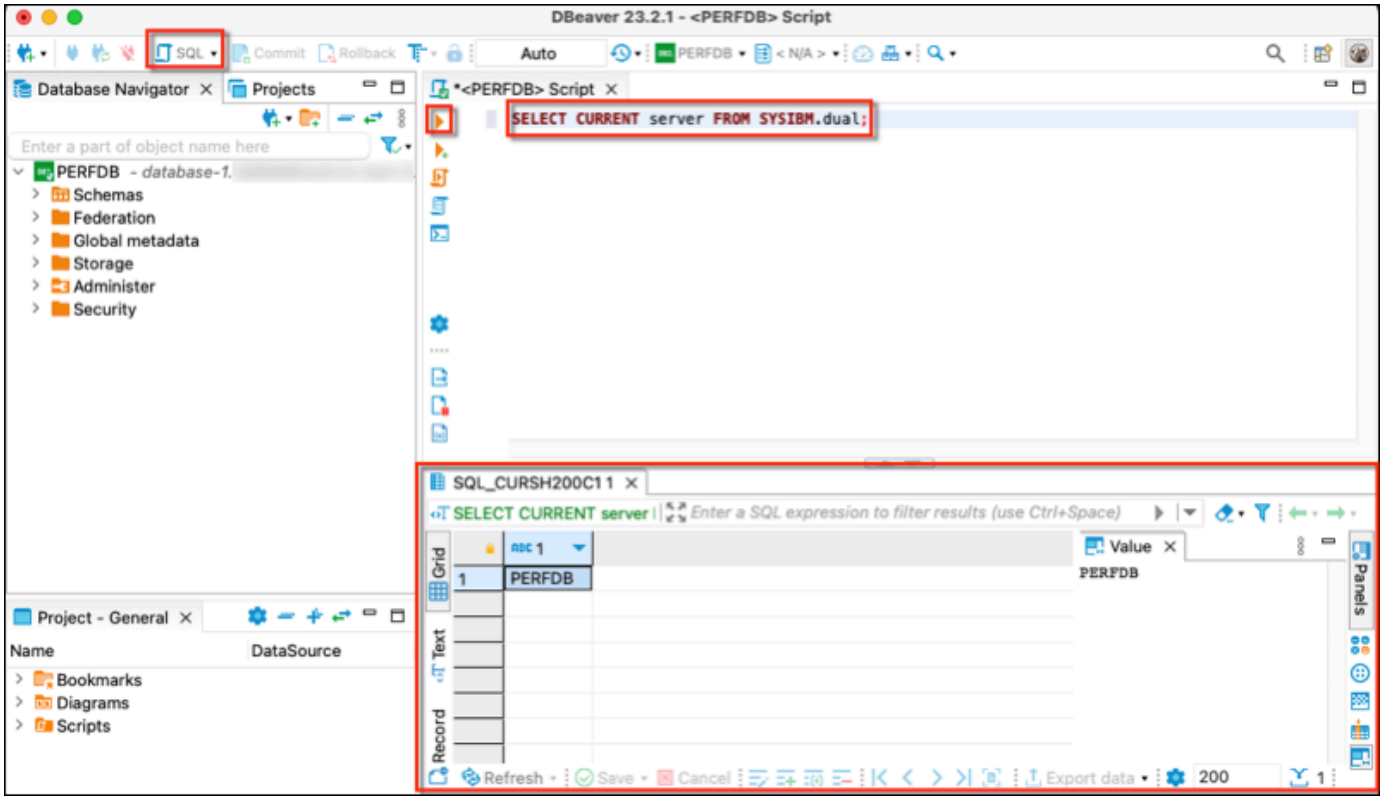


5. Di jendela Hubungi basis data, pilih Uji Koneksi. Jika Anda belum menginstal driver JBDC Db2 di komputer Anda, maka driver diunduh secara otomatis.
6. Pilih Oke.
7. Pilih Selesai.
8. Di tab Navigasi Basis Data, pilih nama basis data. Anda kini dapat menjelajahi objek-objek.

Anda kini siap untuk menjalankan perintah SQL.

Untuk menjalankan perintah SQL dan melihat hasilnya

1. Di menu atas, pilih SQL. Menu ini membuka panel skrip SQL.
2. Di panel Skrip, masukkan perintah SQL.
3. Untuk menjalankan perintah, pilih tombol Jalankan kueri SQL.
4. Di panel hasil SQL, lihat hasil kueri SQL Anda.



Menghubungi instans basis data RDS for Db2 dengan IBM Db2 Data Management Console

Anda dapat menghubungi instans basis data Amazon RDS for Db2 Anda dengan IBM Db2 Data Management Console. IBM Db2 Data Management Console dapat mengadministrasikan dan memantau beberapa instans basis data RDS for Db2. Untuk mengunduh utilitas ini, lihat rilis-rilis IBM Db2 Data Management Console Versi 3.1x di Dukungan IBM.

IBM Db2 Data Management Console mensyaratkan basis data Db2 repositori menyimpan metadata dan metrik kinerja, tetapi tidak dapat membuat secara otomatis repositori untuk RDS for Db2.

Anda harus membuat dahulu basis data repositori untuk memantau satu atau beberapa instans basis data RDS for Db2. Lalu, hubungkan dengan instans basis data RDS for Db2 Anda dengan IBM Db2 Data Management Console.

Topik

- Membuat basis data repositori untuk memantau instans basis data
- Menghubungi instans basis data RDS for Db2 dengan IBM Db2 Data Management Console

Membuat basis data repositori untuk memantau instans basis data

Anda dapat menggunakan instans basis data RDS for Db2 berukuran tepat yang ada sebagai repositori bagi IBM Db2 Data Management Console untuk memantau instans basis data RDS for Db2 yang lain. Namun, karena pengguna admin tidak memiliki otoritas SYSCTRL untuk membuat kolam penyangga dan ruang tabel, penggunaan pembuatan repositori IBM Db2 Data Management Console untuk membuat basis data repositori gagal. Alih-alih, Anda harus membuat basis data repositori untuk memantau instans basis data RDS for Db2 Anda. Anda dapat membuat basis data repositori dengan dua cara. Anda dapat membuat secara manual kolam penyangga, ruang tabel, dan objek untuk sebuah repositori IBM Db2 Data Management Console. Atau Anda dapat membuat instans Amazon EC2 terpisah guna menjadi host repositori IBM Db2 Data Management Console.

Topik

Membuat secara manual kolam penyangga, ruang tabel, dan objek

Untuk membuat kolam penyangga, ruang tabel, dan objek untuk digunakan IBM Db2 Data Management Console

1. Izinkan privilese untuk kolam penyangga dan ruang tabel.

   a. Buat perubahan pada skrip, terutama untuk kolam penyangga dan ruang tabel. Lihat informasi yang lebih lengkap di Mengonfigurasikan basis data repositori dalam dokumentasi IBM Db2 Data Management Console.

   b. Hubungi basis data `rdsadmin`. Dalam contoh berikut, ganti *`master_username`* dan *`master_password`* dengan informasi Anda sendiri.

   ```
   db2 connect to rdadmin user master_username using master_password
   ```

   c. Buat kolam penyangga bagi IBM Db2 Data Management Console. Dalam contoh berikut, ganti *`database_name`* dengan nama repositori yang Anda buat bagi IBM Db2 Data Management Console guna memantau instans basis data RDS for Db2 Anda.

   ```
   db2 "call rdsadmin.create_bufferpool('database_name',
        'BP4CONSOLE', 1000, 'Y', 'Y', 16384)"
   ```

d. Buat ruang tabel untuk IBM Db2 Data Management Console. Dalam contoh berikut, ganti *database_name* dengan nama repositori yang Anda buat bagi IBM Db2 Data Management Console guna memantau instans basis data RDS for Db2 Anda.

```
db2 "call rdsadmin.create_tablespace('database_name',
     'TS4CONSOLE', 'BP4CONSOLE', 16384)"
```

e. Buat ruang tabel sementara untuk IBM Db2 Data Management Console. Dalam contoh berikut, ganti *database_name* dengan nama repositori yang Anda buat bagi IBM Db2 Data Management Console guna memantau instans basis data RDS for Db2 Anda.

```
db2 "call rdsadmin.create_tablespace('database_name',
    'TS4CONSOLE_TEMP', 'BP4CONSOLE', 16384, 0, 0, 'T')"
```

2. Buat objek-objek IBM Db2 Data Management Console secara manual. Lihat informasi yang lebih lengkap di Mengonfigurasikan basis data repositori dalam dokumentasi IBM Db2 Data Management Console.

Membuat instans Amazon EC2 untuk menjadi host repositori IBM Db2 Data Management Console

Anda dapat membuat instans Amazon Elastic Compute Cloud (Amazon EC2) terpisah untuk menjadi host repositori IBM Db2 Data Management Console. Lihat informasi tentang cara membuat instans Amazon EC2 di Tutorial: Memulai instans Linux Amazon EC2 dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Menghubungi instans basis data RDS for Db2 dengan IBM Db2 Data Management Console

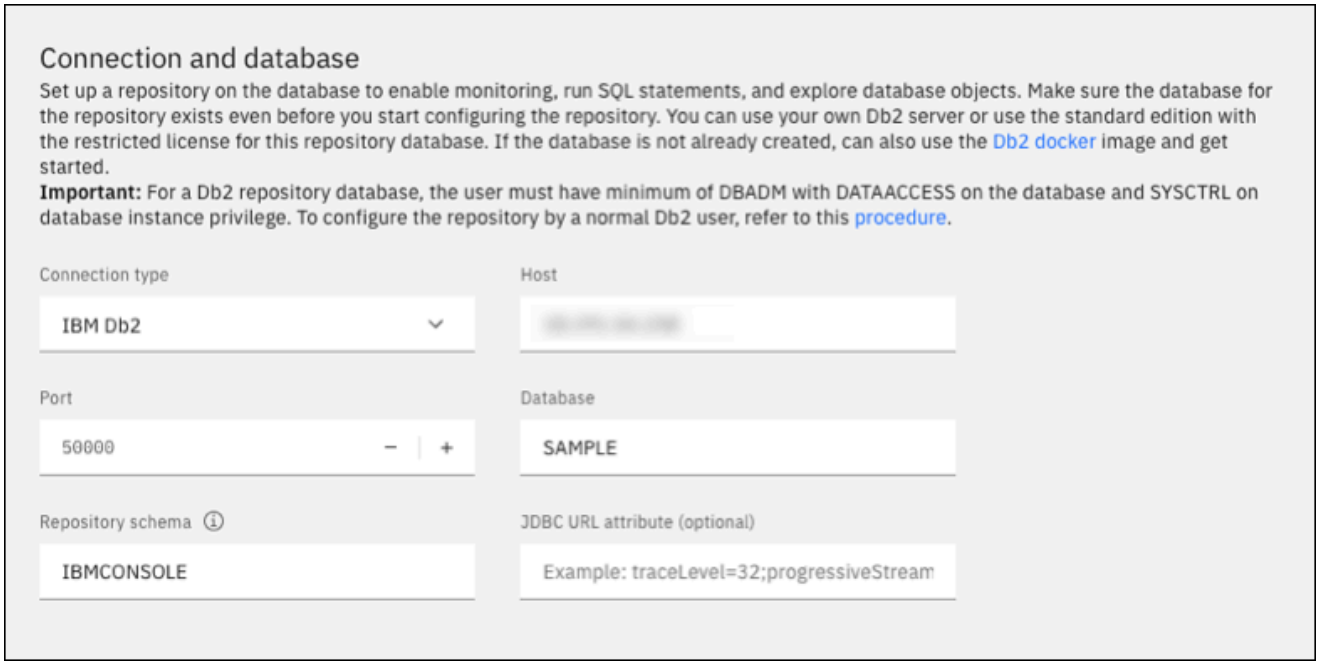
Untuk menghubungi instans basis data RDS for Db2, Anda memerlukan nama dan nomor porta DNS. Lihat informasi tentang cara menemukan info itu di Menemukan titik akhir. Anda juga perlu mengetahui nama basis data, nama pengguna master, dan kata sandi master yang Anda tentukan saat membuat instans basis data RDS for Db2. Lihat informasi yang lebih lengkap tentang cara menemukan info itu di Membuat instans DB. Jika Anda menghubungi melalui internet, izinkan lalu lintas ke porta basis data. Untuk informasi selengkapnya, lihat Membuat instans DB.

Untuk menghubungi instans basis data RDS for Db2 dengan IBM Db2 Data Management Console

1. Mulai IBM Db2 Data Management Console.
2. Konfigurasikan repositori.

a. Di bagian Koneksi dan basis data, masukkan informasi berikut untuk instans basis data RDS for Db2 Anda:

   - Untuk Host, masukkan nama DNS instans basis data.
   - Untuk Porta, masukkan nomor porta untuk instans basis data.
   - Untuk Basis Data, masukkan nama basis data.



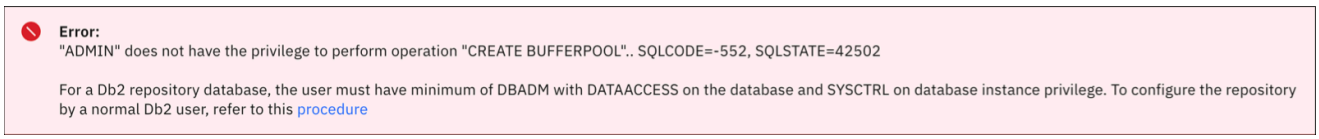
b. Di bagian Keamanan dan kredensial, masukkan informasi berikut untuk instans basis data RDS for Db2 Anda:

   - Untuk Jenis keamanan, pilih Pengguna dan kata sandi terenkripsi.
   - Untuk Nama Pengguna, masukkan nama administrator basis data untuk instans basis data.
   - Untuk Kata Sandi, masukkan kata sandi administrator basis data untuk instans basis data.

c. Pilih Uji koneksi.

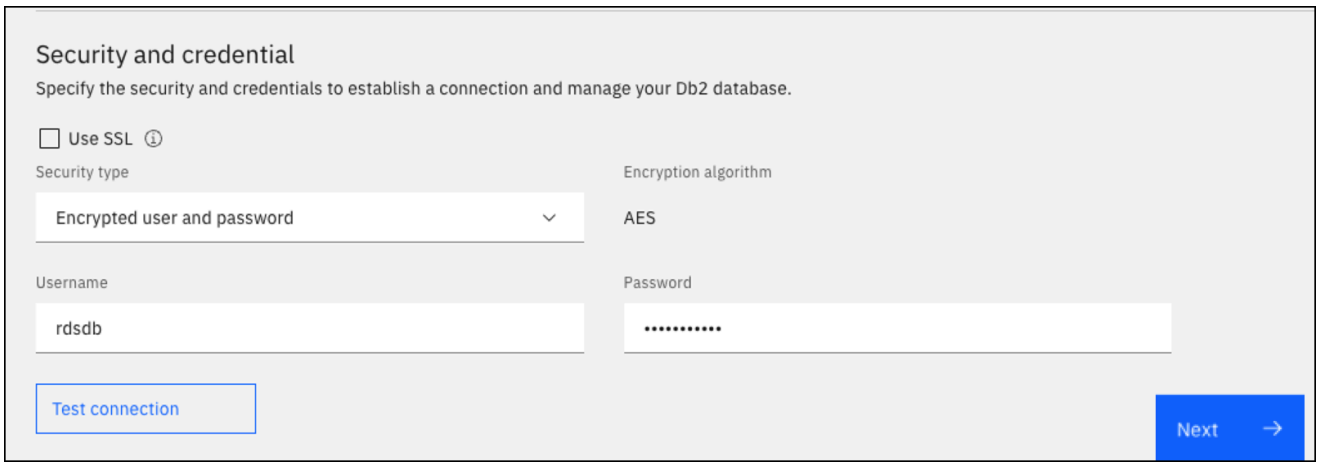
> **Note**
>
> Jika koneksi tidak berhasil, pastikan bahwa porta basis data terbuka melalui aturan masuk grup keamanan. Untuk informasi selengkapnya, lihat Pertimbangan-pertimbangan untuk grup keamanan.

Pesan kesalahan berikut menunjukkan bahwa pengguna admin yang menghubungi instans basis data RDS for Db2 tidak memiliki privilese untuk membuat kolam penyangga atau ruang tabel. Pesan ini juga menunjukkan bahwa untuk basis data repositori Db2, pengguna harus memiliki `DBADM` dan `DATACCESS` pada basis data. Pengguna juga harus memiliki `SYSCTRL` pada privilese database-instance.



Pastikan bahwa Anda membuat kolam penyangga, ruang tabel, dan objek untuk repositori IBM Db2 Data Management Console guna memantau instans basis data RDS for Db2 Anda. Atau, Anda dapat menggunakan instans basis data Amazon EC2 Db2 untuk menjadi host repositori IBM Db2 Data Management Console guna memantau instans basis data RDS for Db2 Anda. Untuk informasi selengkapnya, lihat Membuat basis data repositori untuk memantau instans basis data.

d. Setelah Anda berhasil menguji koneksi, pilih Berikutnya.

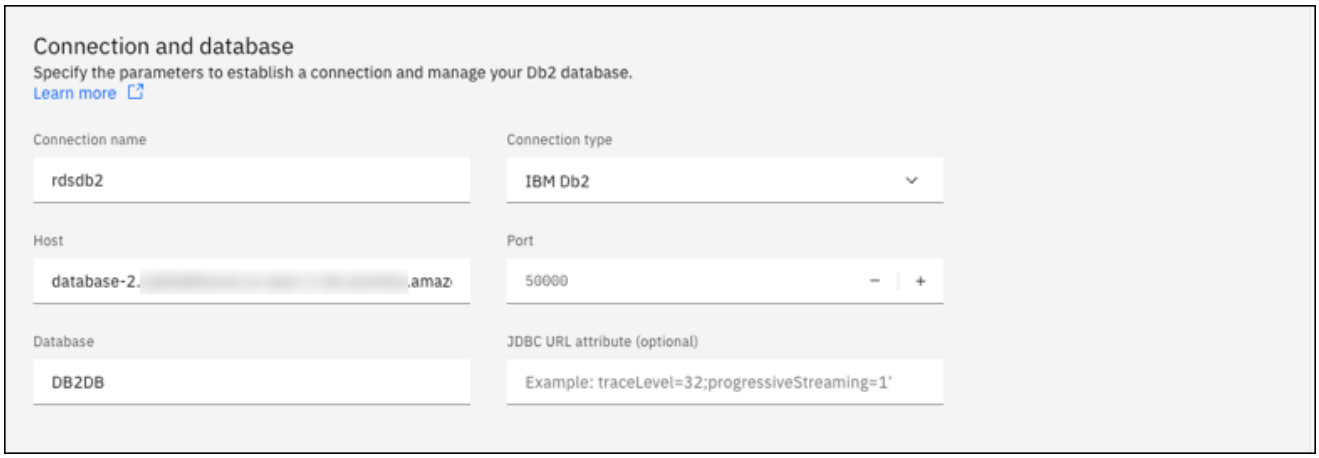


3. Di jendela Atur keikutsertaan statistik pemantauan peristiwa, pilih Berikutnya.

4. (Opsional) Tambahkan koneksi baru. Jika Anda ingin menggunakan instans basis data RDS for Db2 yang berbeda untuk administrasi dan pemantauan, maka tambahkan koneksi dengan instans basis data RDS for Db2 nonrepositori.

   a. Di bagian Koneksi dan basis data, masukkan informasi berikut untuk instans basis data RDS for Db2 yang akan digunakan untuk administrasi dan pemantauan:

      - Untuk Nama koneksi, masukkan pengenal basis data Db2.
      - Untuk Host, masukkan nama DNS instans basis data.
      - Untuk Porta, masukkan nomor porta untuk instans basis data.
      - Untuk Basis Data, masukkan nama basis data.



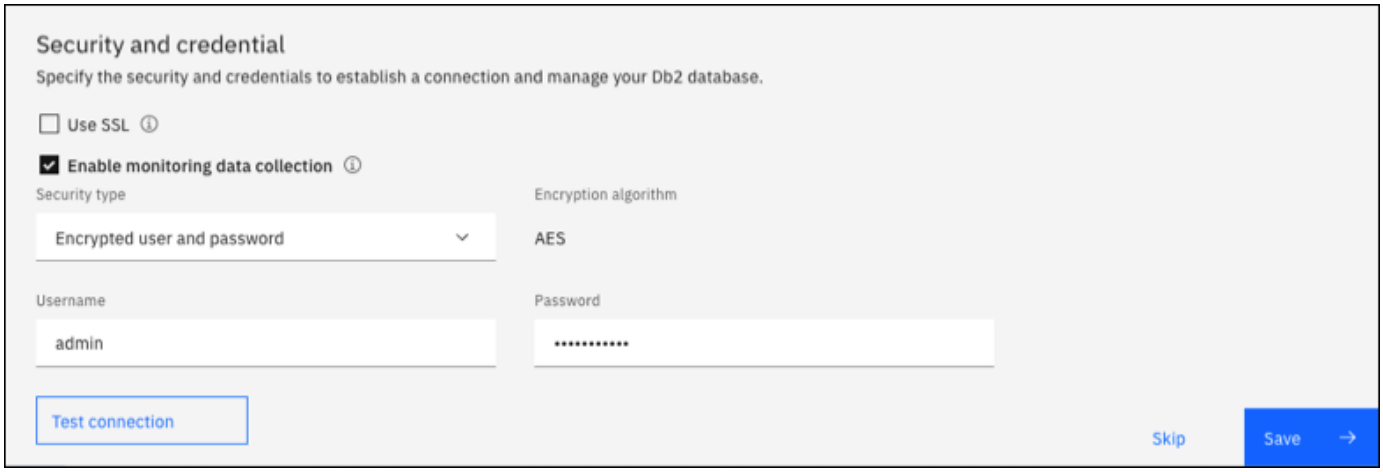
   b. Di bagian Keamanan dan kredensial, pilih Aktifkan pengumpulan data pemantauan.

   c. Masukkan informasi berikut untuk instans basis data RDS for Db2 Anda:

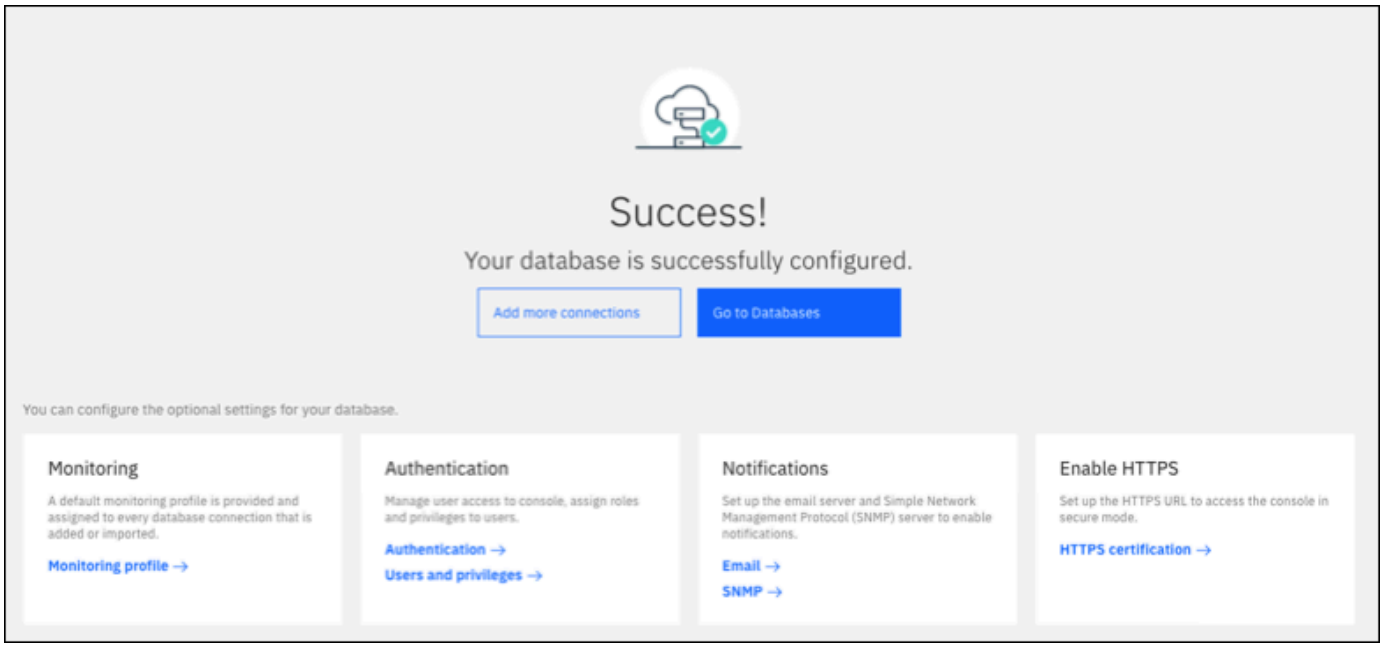
      - Untuk Nama Pengguna, masukkan nama administrator basis data untuk instans basis data.
      - Untuk Kata Sandi, masukkan kata sandi administrator basis data untuk instans basis data.

   d. Pilih Uji koneksi.

   e. Setelah Anda berhasil menguji koneksi, pilih Simpan.



Setelah koneksi ditambahkan, muncul jendela yang mirip dengan jendela berikut. Jendela ini menunjukkan bahwa basis data Anda berhasil dikonfigurasikan.



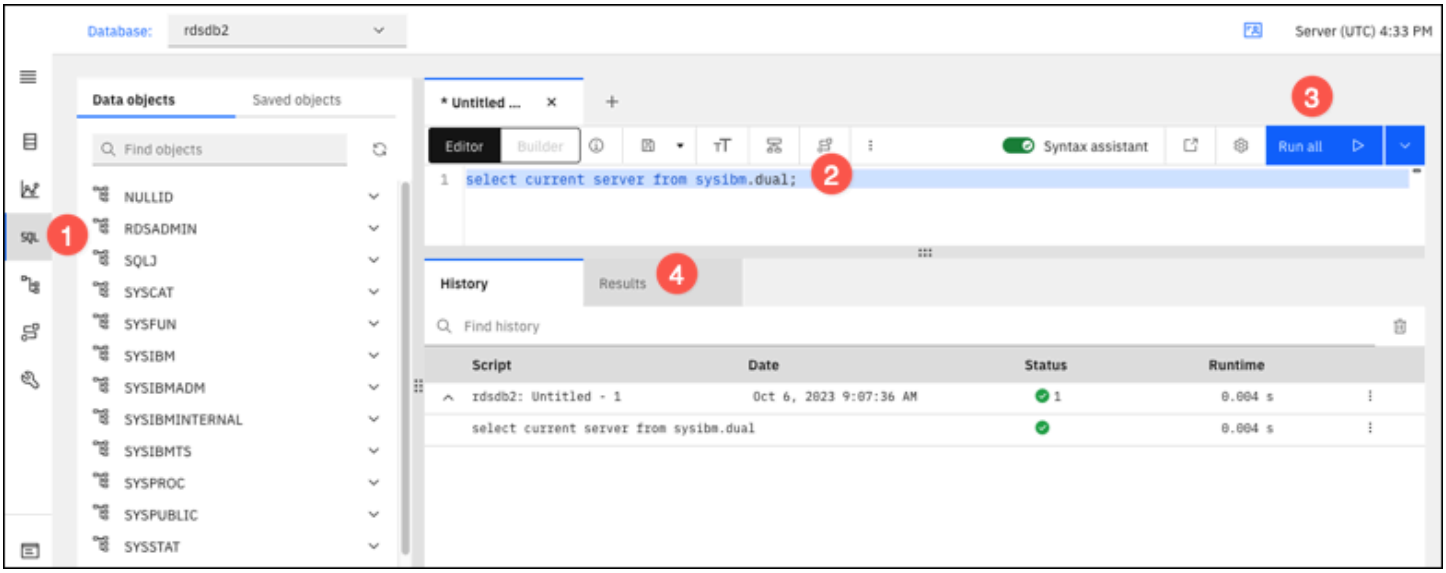
5. Pilih Tuju Basis Data. Muncul jendela basis data yang serupa dengan jendela berikut. Jendela ini adalah dasbor yang menampilkan metrik, status, dan koneksi.

Anda kini dapat mulai menggunakan IBM Db2 Data Management Console untuk melakukan jenis-jenis tugas berikut:

- Mengelola beberapa instans basis data RDS for Db2.
- Menjalankan perintah SQL.
- Menjelajahi, membuat, atau mengubah data dan objek basis data.
- Membuat pernyataan `EXPLAIN PLAN` di SQL.
- Menyetel kueri.

Untuk menjalankan perintah SQL dan melihat hasilnya

1. Di bilah navigasi kiri, pilih SQL.
2. Masukkan perintah SQL.
3. Pilih Jalankan semua.
4. Untuk melihat hasilnya, pilih tab Hasil.



Pertimbangan-pertimbangan untuk grup keamanan

Agar Anda dapat menghubungi instans basis data RDS for Db2, instans itu harus dikaitkan dengan sebuah grup keamanan yang berisi alamat-alamat IP dan konfigurasi jaringan yang diperlukan. Instans basis data RDS for Db2 Anda mungkin menggunakan grup keamanan bawaan. Jika Anda menetapkan grup keamanan nonkonfigurasi bawaan saat membuat instans basis data RDS for Db2, tembok api/firewall akan mencegah koneksi internet. Lihat informasi tentang cara membuat grup keamanan baru di Mengontrol akses dengan grup keamanan.

Setelah grup keamanan baru dibuat, ubah instans basis data Anda untuk mengaitkannya dengan grup keamanan itu. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Anda dapat meningkatkan keamanan dengan menggunakan SSL untuk mengenkripsi koneksi dengan instans basis data Anda. Lihat informasi yang lebih lengkap di Menggunakan SSL/TLS dengan instans basis data RDS for Db2.

Mengamankan koneksi instans basis data RDS for Db2

Amazon RDS for Db2 mendukung cara-cara meningkatkan keamanan untuk instans basis data RDS for Db2 Anda.

Topik

Menggunakan SSL/TLS dengan instans basis data RDS for Db2

SSL adalah sebuah protokol standar industri untuk mengamankan koneksi jaringan antara klien dan server. Setelah SSL versi 3.0, namanya diubah menjadi TLS, tetapi kami masih sering merujuk ke protokol ini dengan SSL. Amazon RDS mendukung enkripsi SSL untuk instans basis data RDS for Db2. Dengan SSL/TLS, Anda dapat mengenkripsi koneksi antara klien aplikasi dan instans basis data RDS for Db2 Anda. Dukungan SSL/TLS tersedia di semua Wilayah AWS untuk RDS for Db2.

Untuk mengaktifkan enkripsi SSL/TLS bagi instans basis data RDS for Db2, tambahkan opsi SSL Db2 ke grup parameter yang terkait dengan instans basis data. Amazon RDS menggunakan porta kedua, sebagaimana diperlukan oleh Db2, untuk koneksi SSL/TLS. Melakukan hal ini memungkinkan komunikasi baik teks jelas maupun berenkripsi SSL terjadi serentak antara instans basis data dan klien Db2. Misalnya, Anda dapat menggunakan porta dengan komunikasi teks jelas untuk berkomunikasi dengan sumber daya lain di dalam VPC sambil menggunakan porta komunikasi berenkripsi SSL untuk berkomunikasi dengan sumber daya di luar VPC.

Topik

Membuat koneksi SSL/TLS

Untuk membuat koneksi SSL/TLS, pilih otoritas sertifikat (CA), unduh bundel sertifikat untuk semua Wilayah AWS, dan tambahkan parameter-parameter ke grup parameter kustom.

Langkah 1: Pilih CA dan unduh sertifikat

Pilih otoritas sertifikat (CA) dan unduh bundel sertifikat untuk semua Wilayah AWS. Untuk informasi selengkapnya, lihat .

Langkah 2: Perbarui parameter-parameter dalam grup parameter kustom

> **Important**
>
> Jika Anda menggunakan model Bawa Lisensi Sendiri (BYOL) untuk RDS for Db2, ubah grup parameter kustom yang Anda buat untuk IBM Customer ID dan IBM Site ID Anda. Jika Anda menggunakan model pelisensian yang lain untuk RDS for Db2, maka ikuti prosedur untuk menambahkan parameter ke grup parameter kustom. Untuk informasi selengkapnya, lihat Opsi-opsi pelisensian RDS for Db2.

Anda tidak dapat mengubah grup parameter bawaan untuk instans basis data RDS for Db2. Oleh karena itu, Anda harus membuat grup parameter kustom, mengubahnya, dan lalu melampirkannya pada instans basis data RDS for Db2 Anda. Lihat informasi yang lebih lengkap tentang grup parameter di Bekerja dengan grup parameter DB dalam instance DB.

Gunakan setelan parameter dalam tabel berikut.

Parameter	Nilai
`DB2COMM`	`TCPIP,SSL`
`SSL_SVCENAME`	`<any port number except the number used for the non-SSL port>`

Untuk memperbarui parameter-parameter dalam grup parameter kustom

1. Buat grup parameter kustom dengan menjalankan perintah create-db-parameter-group.

   Sertakan opsi-opsi yang diperlukan berikut:

   - `--db-parameter-group-name` – Nama untuk grup parameter yang sedang Anda buat.
   - `--db-parameter-group-family` – Edisi mesin dan versi utama Db2. Nilai-nilai yang valid: `db2-se-11-5`, `db2-ae-11.5`.

- `--description` – Deskripsi untuk grup parameter ini.

Lihat informasi yang lebih lengkap tentang cara membuat grup parameter basis data di Membuat grup parameter DB.

2. Ubah parameter-parameter dalam grup parameter kustom yang Anda buat dengan menjalankan perintah modify-db-parameter-group.

   Sertakan opsi-opsi yang diperlukan berikut:

   - `--db-parameter-group-name` – Nama grup parameter yang Anda buat.
   - `--parameters` – Larik nama parameter, nilai parameter, dan metode aplikasi untuk pembaruan parameter.

   Lihat informasi yang lebih lengkap tentang mengubah grup parameter di Memodifikasi parameter dalam grup parameter DB.

3. Kaitkan grup parameter dengan instans basis data RDS for Db2 Anda. Untuk informasi selengkapnya, lihat Mengaitkan grup parameter DB dengan instans DB.

Hubungi server basis data Db2 Anda

Petunjuk untuk menghubungi server basis data Db2 Anda bersifat spesifik bahasa.

Java

Untuk menghubungi server basis data Db2 Anda dengan menggunakan Java

1. Mengunduh driver JDBC. Lihat informasi yang lebih lengkap di Versi dan Unduhan Driver JDBC Db2 dalam dokumentasi Dukungan IBM.
2. Buat file skrip shell dengan isi berikut. Skrip ini menambahkan semua sertifikat dari bundel ke sebuah Java KeyStore.

> **Important**
>
> Periksa bahwa `keytool` ada di jalur dalam skrip sehingga skrip dapat menemukannya. Jika klien Db2 digunakan, Anda dapat menemukan `keytool` di bawah `~sqlib/java/jdk64/jre/bin`.

```
#!/bin/bash
PEM_FILE=$1
PASSWORD=$2
KEYSTORE=$3
# number of certs in the PEM file
CERTS=$(grep 'END CERTIFICATE' $PEM_FILE| wc -l)
for N in $(seq 0 $(($CERTS - 1))); do
    ALIAS="${PEM_FILE%.*}-$N"
    cat $PEM_FILE |
    awk "n==$N { print }; /END CERTIFICATE/ { n++ }" |
    keytool -noprompt -import -trustcacerts -alias $ALIAS -keystore $KEYSTORE -
storepass $PASSWORD
done
```

3. Untuk menjalankan skrip shell dan mengimpor file PEM beserta bundel sertifikat ke dalam Java KeyStore, jalankan perintah berikut. Ganti *shell_file_name.sh* dengan nama file skrip shell Anda dan *pasword* dengan kata sandi untuk Java KeyStore Anda.

```
./shell_file_name.sh global-bundle.pem password truststore.jks
```

4. Untuk menghubungi server Db2 Anda, jalankan perintah berikut. Ganti penampung-penampung nilai berikut dalam contoh dengan informasi instans basis data RDS for Db2 Anda.

   - *ip_address* – Alamat IP untuk titik akhir instans basis data Anda.
   - *port* – Nomor porta untuk koneksi SSL. Ini boleh berupa sebarang nomor porta selain nomor yang digunakan untuk porta non-SSL.
   - *database_name* – Nama basis data dalam instans basis data Anda.
   - *master_username* – Nama pengguna master untuk instans basis data Anda.
   - *master_password* – Kata sandi master untuk instans basis data Anda.

```
export trustStorePassword=MyPassword
java -cp ~/dsdriver/jdbc_sqlj_driver/linuxamd64/db2jcc4.jar \
com.ibm.db2.jcc.DB2Jcc -url \
"jdbc:db2://ip_address:port/database_name:\
sslConnection=true;sslTrustStoreLocation=\
~/truststore.jks;\
sslTrustStorePassword=${trustStorePassword};\
```

```
sslVersion=TLSv1.2;\
encryptionAlgorithm=2;\
securityMechanism=7;" \
-user master_username -password master_password
```

Node.Js

Untuk menghubungi server basis data Db2 Anda dengan menggunakan Node.js

1. Instal driver node-ibm_db. Lihat informasi yang lebih lengkap di Menginstal driver node-ibm_db pada sistem-sistem Linux dan UNIX dalam dokumentasi IBM Db2.
2. Buat file JavaScript berdasarkan isi berikut. Ganti penampung-penampung nilai berikut dalam contoh dengan informasi instans basis data RDS for Db2 Anda.

   - *ip_address* – Alamat IP untuk titik akhir instans basis data Anda.
   - *master_username* – Nama pengguna master untuk instans basis data Anda.
   - *master_password* – Kata sandi master untuk instans basis data Anda.
   - *database_name* – Nama basis data dalam instans basis data Anda.
   - *port* – Nomor porta untuk koneksi SSL. Ini boleh berupa sebarang nomor porta selain nomor yang digunakan untuk porta non-SSL.

```
var ibmdb = require("ibm_db");
const hostname = "ip_address";
const username = "master_username";
const password = "master_password";
const database = "database_name";
const port = "port";
const certPath = "/root/qa-bundle.pem";
ibmdb.open("DRIVER={DB2};DATABASE=" + database + ";HOSTNAME=" +
 hostname + ";UID=" + username + ";PWD=" + password + ";PORT=" + port +
 ";PROTOCOL=TCPIP;SECURITY=SSL;SSLServerCertificate=" + certPath + ";", function
 (err, conn){
 if (err) return console.log(err);
 conn.close(function () {
 console.log('done');
 });
});
```

3. Untuk menjalankan file JavaScript, jalankan perintah berikut.

```
node ssl-test.js
```

Python

Untuk menghubungi server basis data Db2 Anda dengan menggunakan Python

1. Buat file Python dengan isi berikut. Ganti penampung-penampung nilai berikut dalam contoh dengan informasi instans basis data RDS for Db2 Anda.

   - *port* – Nomor porta untuk koneksi SSL. Ini boleh berupa sebarang nomor porta selain nomor yang digunakan untuk porta non-SSL.
   - *master_username* – Nama pengguna master untuk instans basis data Anda.
   - *master_password* – Kata sandi master untuk instans basis data Anda.
   - *database_name* – Nama basis data dalam instans basis data Anda.
   - *ip_address* – Alamat IP untuk titik akhir instans basis data Anda.

```
import click
import ibm_db
import sys

port = port;
master_user_id = "master_username" # Master id used to create your DB instance
master_password = "master_password" # Master password used to create your DB
 instance
db_name = "database_name" # If not given "db-name'
vpc_customer_private_ip = "ip_address" # Hosts end points - Customer private IP
 Addressicert_path = "/root/ssl/global-bundle.pem" # cert path

@click.command()
@click.option("--path", help="certificate path")
def db2_connect(path):

    try:
        conn =
 ibm_db.connect(f"DATABASE={db_name};HOSTNAME={vpc_customer_private_ip};PORT={port};

 PROTOCOL=TCPIP;UID={master_user_id};PWD={master_password};SECURITY=ssl;SSLServerCertifi
 "", "")
```

```
        try:
            ibm_db.exec_immediate(conn, 'create table tablename (a int);')
            print("Query executed successfully")
        except Exception as e:
            print(e)
        finally:
            ibm_db.close(conn)
            sys.exit(1)
    except Exception as ex:
        print("Trying to connect...")

if __name__ == "__main__":
    db2_connect()
```

2. Buat skrip shell berikut, yang menjalankan file Python yang Anda buat. Ganti *python_file_name.py* dengan nama file skrip Python Anda.

```
#!/bin/bash
PEM_FILE=$1
# number of certs in the PEM file
CERTS=$(grep 'END CERTIFICATE' $PEM_FILE| wc -l)

for N in $(seq 0 $(($CERTS - 1))); do
    ALIAS="${PEM_FILE%.*}-$N"
    cert=`cat $PEM_FILE | awk "n==$N { print }; /END CERTIFICATE/ { n++ }"`
    cat $PEM_FILE | awk "n==$N { print }; /END CERTIFICATE/ { n++ }" >
 $ALIAS.pem
    python3 <python_file_name.py> --path $ALIAS.pem
    output=`echo $?`
    if [ $output == 1 ]; then
        break
    fi
done
```

3. Untuk mengimpor file PEM beserta bundel sertifikat dan menjalankan skrip shell, jalankan perintah berikut. Ganti *shell_file_name.sh* dengan nama file skrip shell Anda.

```
./shell_file_name.sh global-bundle.pem
```

Menggunakan autentikasi Kerberos untuk RDS for Db2

Anda kini dapat menggunakan autentikasi Kerberos untuk mengautentikasi pengguna saat menghubungi instans basis data Amazon RDS for Db2 Anda. Instans basis data Anda bekerja dengan AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD) untuk mengaktifkan autentikasi Kerberos. Ketika pengguna mengautentikasi dengan instans basis data RDS for Db2 yang digabungkan dengan domain tepercaya, permintaan autentikasi diteruskan ke direktori yang Anda buat dengan AWS Directory Service. Lihat informasi yang lebih lengkap di Apakah AWS Directory Service? dalam Panduan Administrasi AWS Directory Service.

Pertama, buat direktori AWS Managed Microsoft AD untuk menyimpan kredensial pengguna. Kemudian, tambahkan domain dan informasi lain dari direktori AWS Managed Microsoft AD Anda ke instans basis data RDS for Db2 Anda. Saat pengguna mengautentikasi dengan instans basis data RDS for Db2, permintaan autentikasi diteruskan ke direktori AWS Managed Microsoft AD.

Menyimpan semua kredensial Anda di direktori yang sama dapat menghemat waktu dan tenaga Anda. Dengan pendekatan ini, Anda memiliki sebuah tempat terpusat untuk menyimpan dan mengelola kredensial bagi beberapa instans basis data. Menggunakan direktori juga dapat meningkatkan profil keamanan keseluruhan Anda.

Topik

Kawasan dan ketersediaan versi

Ketersediaan dan dukungan fitur bervariasi di antara versi-versi spesifik setiap mesin basis data, dan di antara Wilayah AWS. Lihat informasi yang lebih lengkap tentang ketersediaan versi dan kawasan RDS for Db2 dengan autentikasi Kerberos di Autentikasi Kerberos.

> **Note**
>
> Autentikasi Kerberos tidak didukung untuk kelas instans basis data yang sudah diusangkan untuk instans basis data RDS for Db2. Lihat informasi yang lebih lengkap di Kelas-kelas instans RDS for Db2.

Ikhtisar autentikasi Kerberos untuk instans basis data RDS for Db2

Untuk menyiapkan autentikasi Kerberos bagi instans basis data RDS for Db2, selesaikan langkah-langkah umum berikut, yang nanti akan dijelaskan lebih terperinci:

1. Gunakan AWS Managed Microsoft AD untuk membuat direktori AWS Managed Microsoft AD. Anda dapat menggunakanAWS Management Console, AWS Command Line Interface (AWS CLI), atau AWS Directory Service untuk membuat direktori. Lihat informasi yang lebih lengkap di Membuat direktori AWS Managed Microsoft AD Anda dalam Panduan Administrasi AWS Directory Service.
2. Buat sebuah peran AWS Identity and Access Management (IAM) yang menggunakan kebijakan IAM terkelola `AmazonRDSDirectoryServiceAccess`. Peran IAM memungkinkan Amazon RDS melakukan panggilan ke direktori Anda.

   Agar peran IAM mengizinkan akses, titik akhir AWS Security Token Service (AWS STS) harus diaktifkan dalam Wilayah AWS yang benar untuk Akun AWS Anda. Titik akhir AWS STS akan aktif secara bawaan di semua Wilayah AWS, dan Anda dapat menggunakannya tanpa tindakan lebih lanjut. Lihat informasi yang lebih lengkap di Mengaktifkan dan menonaktifkan AWS STS di Wilayah AWS dalam Panduan Pengguna IAM.
3. Buat atau ubah instans basis data RDS for Db2 dengan menggunakan AWS Management Console, AWS CLI, atau RDS API dengan salah satu metode berikut:
   - Buat instans basis data RDS for Db2 baru dengan menggunakan konsol, perintah create-db-instance, atau operasi API CreateDBInstance. Untuk petunjuk, lihat Membuat instans DB Amazon RDS.
   - Ubah instans basis data RDS for Db2 yang ada dengan menggunakan konsol, perintah modify-db-instance, atau operasi API ModifyDBInstance. Lihat petunjuknya di Memodifikasi instans DB Amazon RDS.
   - Pulihkan instans basis data RDS for Db2 dari cuplikan basis data dengan menggunakan konsol, perintah restore-db-instance-from-db-snapshot, atau operasi API RestoreDBInstanceFromDBSnapshot. Lihat petunjuknya di Memulihkan dari snapshot DB.

- Pulihkan instans RDS untuk Db2 DB ke point-in-time menggunakan konsol, restore-db-instance-to-point-in-timeperintah, atau operasi API. RestoreDBInstanceToPointInTime Untuk petunjuk, lihat Memulihkan instans DB dengan waktu yang ditentukan.

Anda dapat menemukan instans basis data di Amazon Virtual Private Cloud (VPC) yang sama dengan direktori atau di Akun AWS atau VPC yang berbeda. Saat Anda membuat atau mengubah instans basis data RDS for Db2, lakukan tugas-tugas berikut:

- Sediakan pengidentifikasi domain (pengidentifikasi `d-*`) yang dihasilkan saat Anda membuat direktori.
- Beri nama juga peran IAM yang Anda buat.
- Periksa bahwa grup keamanan instans basis data dapat menerima lalu lintas masuk dari grup keamanan direktori.

4. Konfigurasikan klien Db2 Anda, dan periksa bahwa lalu lintas dapat mengalir antara host klien dan AWS Directory Service untuk porta-porta berikut:
   - TCP/UDP porta 53 - DNS
   - TCP 88 – autentikasi Kerberos
   - TCP 389 – LDAP
   - TCP 464 – autentikasi Kerberos

Menyiapkan autentikasi Kerberos untuk basis data RDS for Db2

Anda menggunakan AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD) untuk mengatur autentikasi Kerberos bagi instans basis data RDS for Db2. Untuk mengatur autentikasi Kerberos, ikuti langkah-langkah ini:

Topik

Langkah 1: Buat sebuah direktori dengan menggunakan AWS Managed Microsoft AD

AWS Directory Service membuat Active Directory terkelola penuh di AWS Cloud. Saat Anda membuat direktori AWS Managed Microsoft AD, AWS Directory Service membuat dua pengontrol domain dan server DNS untuk Anda. Server-server direktori dibuat di subnet yang berbeda di VPC. Redundansi ini membantu memastikan bahwa direktori Anda tetap dapat diakses meskipun terjadi kegagalan.

Saat Anda membuat direktori AWS Managed Microsoft AD, AWS Directory Service melakukan tugas-tugas berikut untuk Anda:

- Menyiapkan Active Directory di dalam VPC Anda.
- Membuat akun administrator direktori dengan nama pengguna `Admin` dan kata sandi yang ditentukan. Anda menggunakan akun ini untuk mengelola direktori Anda.

  > **Important**
  >
  > Pastikan untuk menyimpan kata sandi ini. AWS Directory Service tidak menyimpan kata sandi ini, dan kata sandi ini tidak dapat diambil atau diatur ulang.

- Membuat grup keamanan untuk pengontrol direktori. Grup keamanan harus mengizinkan komunikasi dengan instans basis data RDS for Db2.

Saat Anda meluncurkan AWS Directory Service for Microsoft Active Directory, AWS membuat unit organisasi (OU) yang berisi semua objek direktori Anda. OU ini, yang memiliki nama NetBIOS yang Anda masukkan saat membuat direktori, terletak di domain akar. Domain akar dimiliki dan dikelola oleh AWS.

Akun `Admin` yang dibuat dengan direktori AWS Managed Microsoft AD Anda memiliki izin untuk tugas-tugas administratif paling umum bagi OU Anda:

- Membuat, memperbarui, atau menghapus pengguna
- Menambahkan sumber daya ke domain Anda seperti server file atau cetak, lalu menetapkan izin untuk sumber daya tersebut kepada pengguna di OU Anda
- Membuat OU dan kontainer tambahan
- Melimpahkan kewenangan
- Memulihkan objek-objek yang dihapus dari Keranjang Sampah Active Directory.

- Menjalankan modul-modul Active Directory dan Domain Name Service (DNS) untuk Windows PowerShell di AWS Directory Service.

Akun `Admin` juga memiliki hak melakukan aktivitas-aktivitas selingkup domain berikut:

- Mengelola konfigurasi DNS (menambahkan, menghapus, atau memperbarui rekam, zona, dan penerus).
- Melihat log peristiwa DNS.
- Melihat log peristiwa keamanan.

Untuk membuat direktori dengan AWS Managed Microsoft AD

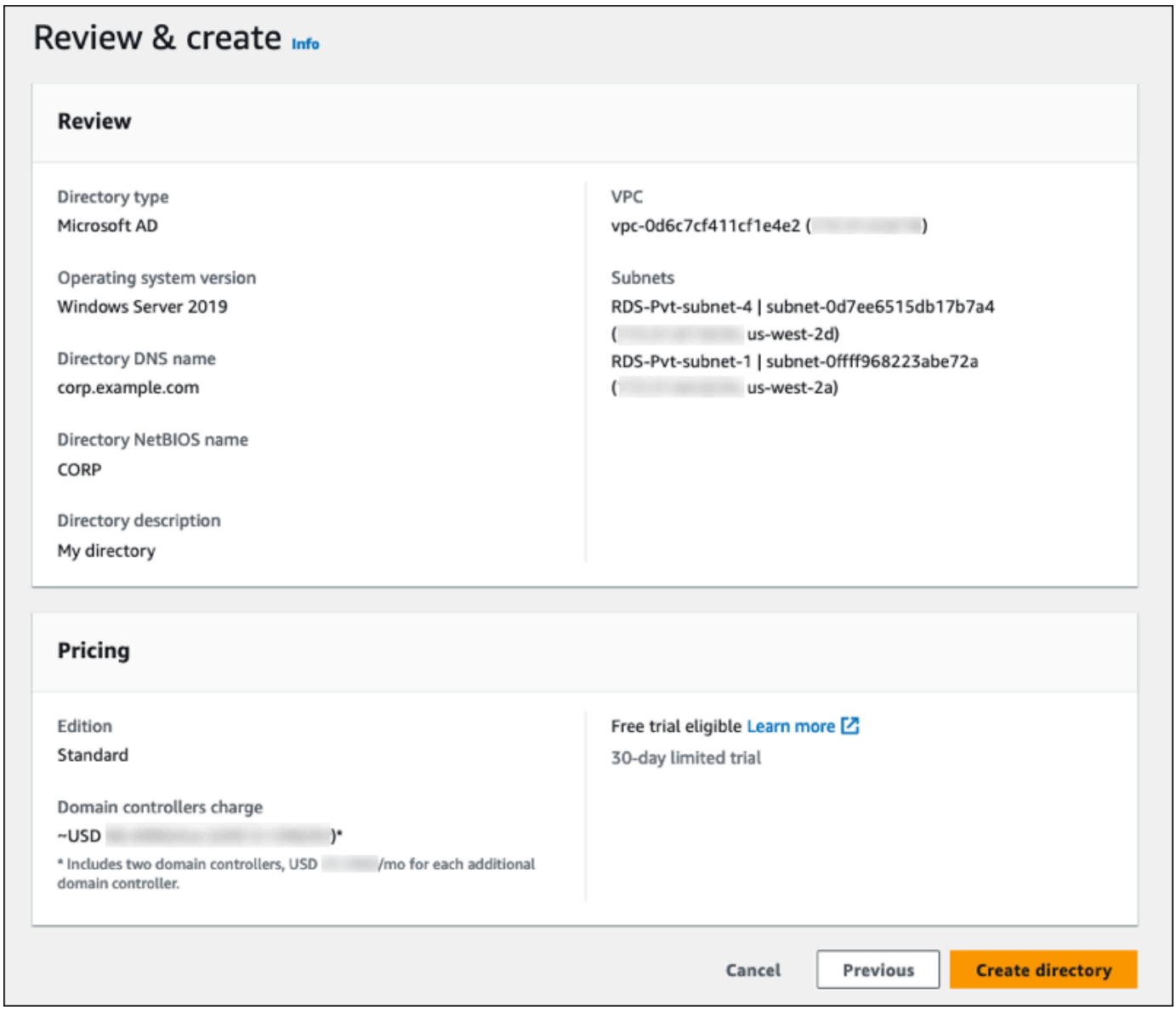
1. Masuk ke AWS Management Console dan buka konsol AWS Directory Service di https://console.aws.amazon.com/directoryservicev2/.
2. Pilih Siapkan direktori.
3. Pilih AWS Managed Microsoft AD. AWS Managed Microsoft AD adalah satu-satunya opsi yang saat ini didukung untuk digunakan dengan Amazon RDS.
4. Pilih Berikutnya.
5. Di halaman Masukkan informasi direktori, berikan informasi berikut:
   - Edisi – Pilih edisi yang memenuhi kebutuhan Anda.
   - Nama DNS direktori – Nama berkualifikasi penuh untuk direktori, seperti `corp.example.com`.
   - Nama NetBIOS direktori – Nama pendek opsional untuk direktori, seperti `CORP`.
   - Deskripsi direktori – Deskripsi opsional untuk direktori.
   - Kata sandi admin – Kata sandi untuk administrator direktori. Proses pembuatan direktori menciptakan akun administrator dengan nama pengguna `Admin` dan kata sandi ini.

     Kata sandi administrator direktori tidak boleh menyertakan kata “admin.” Kata sandi bersifat peka kapital dan harus terdiri atas 8–64 karakter. Kata sandi juga harus berisi setidaknya satu karakter dari tiga di antara empat kategori berikut:
     - Huruf kecil (a-z)
     - Huruf besar (A-Z)
     - Angka (0–9)
     - Karakter nonalfanumerik (~!@#$%^&*_-+=`|\(){}[]:;"'<>,.?/)

- Ulangi kata sandi – Ketik ulang kata sandi administrator.

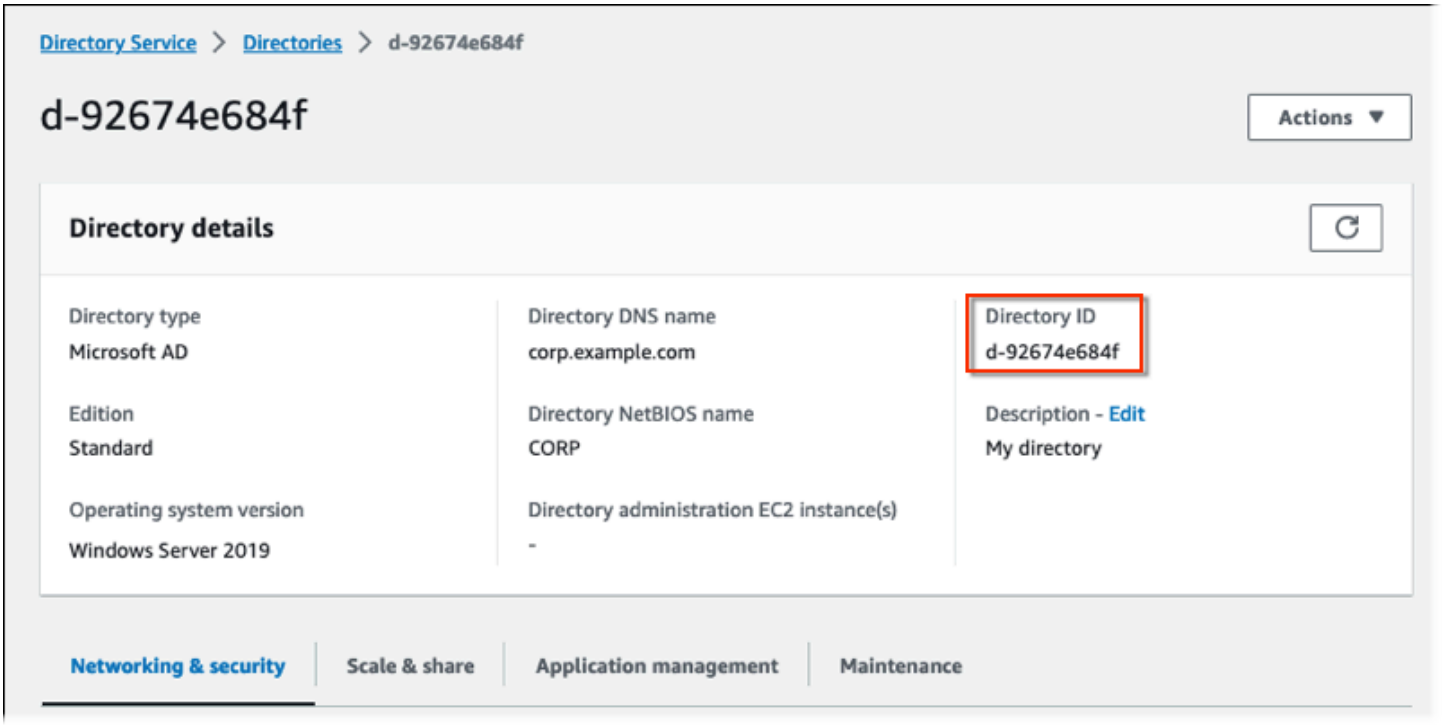
> **Important**
>
> Pastikan untuk menyimpan kata sandi ini. AWS Directory Service tidak menyimpan kata sandi ini, dan kata sandi ini tidak dapat diambil atau diatur ulang.

6. Pilih Berikutnya.
7. Di halaman Pilih VPC dan subnet, berikan informasi berikut:
   - VPC – Pilih VPC untuk direktori. Anda dapat membuat instans basis data RDS for Db2 dalam VPC yang sama ini atau dalam VPC yang berbeda.
   - Subnet – Pilih subnet untuk server direktori. Kedua subnet harus berada di Zona Ketersediaan yang berbeda.
8. Pilih Berikutnya.
9. Tinjau informasi direktori. Jika perubahan diperlukan, pilih Sebelumnya dan buat perubahan. Jika informasi sudah benar, pilih Buat direktori.



Pembuatan direktori memerlukan waktu beberapa menit. Setelah direktori berhasil dibuat, nilai Status berubah menjadi Aktif.

Untuk melihat informasi tentang direktori Anda, pilih ID direktori di ID Direktori. Buat catatan tentang nilai ID Direktori. Anda memerlukan nilai ini saat membuat atau mengubah instans basis data RDS for Db2.



Langkah 2: Buat peran IAM untuk Amazon RDS untuk mengakses AWS Directory Service

Agar Amazon RDS memanggil AWS Directory Service untuk Anda, akun Akun AWS Anda memerlukan peran IAM yang menggunakan kebijakan IAM terkelola `AmazonRDSDirectoryServiceAccess`. Peran ini memungkinkan Amazon RDS melakukan panggilan ke AWS Directory Service.

Saat Anda membuat instans basis data dengan menggunakan AWS Management Console dan akun pengguna konsol Anda memiliki izin `iam:CreateRole`, konsol akan membuat secara otomatis peran IAM yang diperlukan. Dalam hal ini, nama perannya adalah `rds-directoryservice-kerberos-access-role`. Jika tidak, Anda harus membuat peran IAM secara manual. Saat Anda membuat peran IAM ini, pilih `Directory Service`, lalu lampirkan kebijakan terkelola AWS `AmazonRDSDirectoryServiceAccess` ke peran itu.

Lihat informasi yang lebih lengkap tentang membuat peran IAM untuk sebuah layanan di Membuat peran untuk melimpahkan izin ke layanan AWS dalam Panduan Pengguna IAM.

> **Note**
>
> Peran IAM yang digunakan untuk autentikasi Windows untuk RDS for Microsoft SQL Server tidak dapat digunakan untuk RDS for Db2.

Sebagai alternatif untuk penggunaan kebijakan terkelola `AmazonRDSDirectoryServiceAccess`, Anda dapat membuat kebijakan dengan izin-izin yang diperlukan. Dalam hal ini, peran IAM harus memiliki kebijakan kepercayaan IAM berikut:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "directoryservice.rds.amazonaws.com",
          "rds.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Peran ini juga harus memiliki kebijakan peran IAM berikut:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ds:DescribeDirectories",
        "ds:AuthorizeApplication",
        "ds:UnauthorizeApplication",
        "ds:GetAuthorizedApplicationDetails"
      ],
    "Effect": "Allow",
    "Resource": "*"
    }
  ]
}
```

Langkah 3: Buat dan konfigurasikan pengguna

Anda dapat membuat pengguna dengan alat Active Directory Users and Computers. Inilah salah satu alat Active Directory Domain Services dan Active Directory Lightweight Directory Services. Lihat informasi yang lebih lengkap di Menambahkan Pengguna dan Komputer ke domain Active Directory dalam dokumentasi Microsoft. Dalam hal ini, pengguna adalah orang atau entitas lain, seperti komputer, yang merupakan bagian dari domain dan yang identitasnya dipelihara di dalam direktori.

Untuk membuat pengguna di direktori AWS Directory Service, Anda harus menghubungi instans Amazon EC2 berbasis Windows yang merupakan anggota direktori AWS Directory Service. Pada saat yang sama, Anda harus masuk sebagai pengguna yang memiliki privilese membuat pengguna. Lihat informasi yang lebih lengkap di Membuat pengguna dalam Panduan Administrasi AWS Directory Service.

Langkah 4: Buat grup admin RDS for Db2 di AWS Managed Microsoft AD

RDS for Db2 tidak mendukung autentikasi Kerberos untuk pengguna master atau dua pengguna yang dicadangkan Amazon RDS `rdsdb` dan `rdsadmin`. Sebagai gantinya, Anda perlu membuat grup baru yang bernama `masterdba` di AWS Managed Microsoft AD. Lihat informasi yang lebih lengkap di Membuat Akun Grup Active Directory dalam dokumentasi Microsoft. Semua pengguna yang Anda tambahkan ke grup ini akan memiliki privilese pengguna master.

Setelah Anda mengaktifkan autentikasi Kerberos, pengguna master kehilangan peran `masterdba`. Akibatnya, pengguna master tidak akan dapat mengakses keanggotaan grup pengguna lokal instans kecuali Anda menonaktifkan autentikasi Kerberos. Untuk terus menggunakan pengguna master dengan login kata sandi, buat pengguna di AWS Managed Microsoft AD dengan nama yang sama dengan pengguna master. Lalu, tambahkan pengguna itu ke grup `masterdba`.

Langkah 5: Buat atau ubah instans basis data RDS for Db2

Buat atau ubah instans basis data RDS for Db2 untuk digunakan dengan direktori Anda. Anda dapat menggunakanAWS Management Console, AWS CLI, atau RDS API untuk mengaitkan instans basis data dengan direktori. Anda dapat melakukannya dengan salah satu cara berikut:

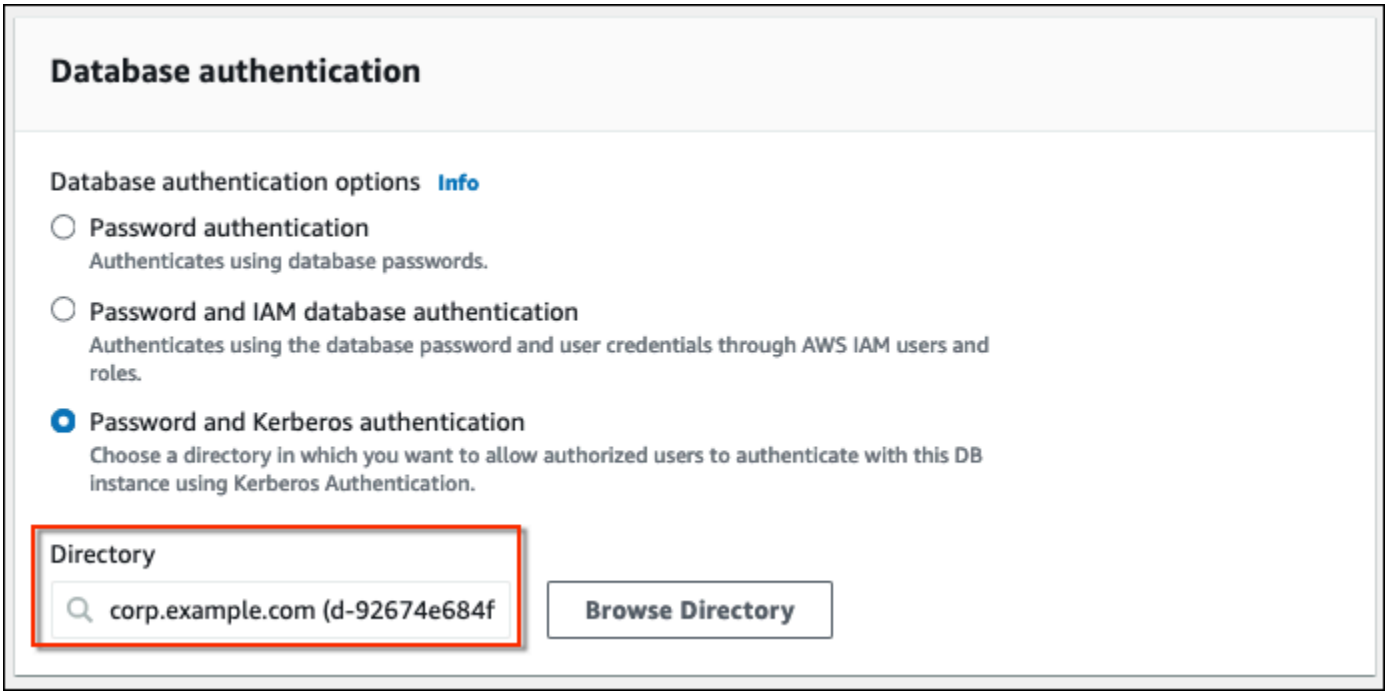
- Buat instans basis data RDS for Db2 baru dengan menggunakan konsol, perintah create-db-instance, atau operasi API CreateDBInstance. Lihat petunjuknya di Membuat instans DB Amazon RDS.

- Ubah instans basis data RDS for Db2 yang ada dengan menggunakan konsol, perintah modify-db-instance, atau operasi API ModifyDBInstance. Lihat petunjuknya di Memodifikasi instans DB Amazon RDS.
- Pulihkan instans basis data RDS for Db2 dari cuplikan basis data dengan menggunakan konsol, perintah restore-db-instance-from-db-snapshot, atau operasi API RestoreDBInstanceFromDBSnapshot. Lihat petunjuknya di Memulihkan dari snapshot DB.
- Pulihkan instans RDS untuk Db2 DB ke point-in-time menggunakan konsol, restore-db-instance-to-point-in-timeperintah, atau operasi API. RestoreDBInstanceToPointInTime Lihat petunjuknya di Memulihkan instans DB dengan waktu yang ditentukan.

Autentikasi Kerberos hanya didukung untuk instans basis data RDS for Db2 dalam VPC. Instans basis data boleh berada dalam VPC yang sama dengan direktori, atau dalam VPC yang berbeda. Instans basis data harus menggunakan grup keamanan yang memungkinkan data masuk dan keluar di dalam VPC direktori, sehingga instans basis data dapat berkomunikasi dengan direktori.

Konsol

Saat Anda menggunakan konsol untuk membuat, mengubah, atau memulihkan instans basis data, pilih Autentikasi kata sandi dan Kerberos di bagian Autentikasi basis data. Kemudian, pilih Telusuri direktori. Pilih direktori atau pilih Buat direktori untuk menggunakan Directory Service.



AWS CLI

Saat Anda menggunakan AWS CLI, parameter-parameter berikut diperlukan untuk instans basis data agar dapat menggunakan direktori yang Anda buat:

- Untuk parameter `--domain`, gunakan pengidentifikasi domain (pengidentifikasi "d-*") yang dihasilkan saat Anda membuat direktori.
- Untuk parameter `--domain-iam-role-name`, gunakan peran yang Anda buat dengan menggunakan kebijakan IAM terkelola `AmazonRDSDirectoryServiceAccess`.

Contoh berikut mengubah instans basis data untuk menggunakan direktori. Ganti penampung nilai berikut dalam contoh dengan nilai-nilai Anda sendiri:

- *`db_instance_name`* – Nama instans basis data RDS for Db2 Anda.
- *`directory_id`* – ID direktori AWS Directory Service for Microsoft Active Directory yang Anda buat.
- *`role_name`* – Nama peran IAM yang Anda buat.

```
aws rds modify-db-instance --db-instance-identifier db_instance_name --domain
 d-directory_id --domain-iam-role-name role_name
```

> **Important**
>
> Jika Anda mengubah instans basis data untuk mengaktifkan autentikasi Kerberos, but ulang instans basis data setelah membuat perubahan.

Langkah 6: Konfigurasikan klien Db2

Untuk mengonfigurasikan klien Db2

1. Buat file /etc/krb5.conf (atau setara) untuk menunjuk ke domain.

> **Note**
>
> Untuk sistem operasi Windows, buat file C:\windows\krb5.ini.

2. Periksa bahwa lalu lintas dapat mengalir antara host klien dan AWS Directory Service. Gunakan utilitas jaringan seperti Netcat untuk tugas-tugas berikut:

   a. Periksa lalu lintas atas DNS untuk porta 53.

   b. Periksa lalu lintas atas TCP/UDP untuk porta 53 dan untuk Kerberos, yang mencakup porta-porta 88 dan 464 untuk AWS Directory Service.

3. Periksa bahwa trafik dapat mengalir antara host klien dan instans basis data melalui porta basis data. Anda dapat menggunakan perintah `db2` untuk menghubungkan dan mengakses basis data.

Contoh berikut adalah /etc/krb5.conf konten file untuk: AWS Managed Microsoft AD

```
[libdefaults]
default_realm = EXAMPLE.COM
[realms]
EXAMPLE.COM = {
kdc = example.com
admin_server = example.com
}
[domain_realm]
.example.com = EXAMPLE.COM
example.com = EXAMPLE.COM
```

Mengelola instans basis data dalam domain

Anda dapat menggunakan AWS Management Console, AWS CLI, atau RDS API untuk mengelola instans basis data Anda dan hubungannya dengan Microsoft Active Directory Anda. Misalnya, Anda dapat mengaitkan Active Directory untuk mengaktifkan autentikasi Kerberos. Anda juga dapat menghapus kaitan untuk Active Directory guna menonaktifkan autentikasi Kerberos. Anda juga dapat memindahkan instans basis data agar diautentikasi secara eksternal oleh satu Microsoft Active Directory ke yang lain.

Misalnya, dengan perintah CLI modify-db-instance, Anda dapat melakukan tindakan-tindakan berikut:

- Coba kembali pengaktifan autentikasi Kerberos untuk keanggotaan yang gagal dengan menentukan ID direktori keanggotaan saat ini untuk opsi `--domain`.
- Nonaktifkan autentikasi Kerberos pada instans basis data dengan menentukan `none` opsi `--domain`.

- Pindahkan instans basis data dari satu domain ke domain lain dengan menentukan pengidentifikasi domain dari domain baru untuk opsi `--domain`.

Memahami keanggotaan domain

Setelah Anda membuat atau mengubah instans basis data, instans akan menjadi anggota domain. Anda dapat melihat status keanggotaan domain di konsol atau dengan menjalankan perintah CLI describe-db-instances. Status instans basis data dapat berupa salah satu nilai berikut:

- `kerberos-enabled` – Instans basis data telah mengaktifkan autentikasi Kerberos.
- `enabling-kerberos` – AWS sedang dalam proses mengaktifkan autentikasi Kerberos pada instans basis data ini.
- `pending-enable-kerberos` – Pengaktifan autentikasi Kerberos tertunda pada instans basis data ini.
- `pending-maintenance-enable-kerberos` – AWS akan mencoba mengaktifkan autentikasi Kerberos pada instans basis data selama jendela pemeliharaan terjadwal berikutnya.
- `pending-disable-kerberos` – Penonaktifan autentikasi Kerberos tertunda pada instans basis data ini.
- `pending-maintenance-disable-kerberos` – AWS akan mencoba menonaktifkan autentikasi Kerberos pada instans basis data selama jendela pemeliharaan terjadwal berikutnya.
- `enable-kerberos-failed` – Masalah konfigurasi mencegah AWS mengaktifkan autentikasi Kerberos pada instans basis data. Perbaiki masalah konfigurasi sebelum menerbitkan ulang perintah untuk mengubah instans basis data.
- `disabling-kerberos` – AWS sedang dalam proses menonaktifkan autentikasi Kerberos pada instans basis data ini.

Permintaan untuk mengaktifkan autentikasi Kerberos dapat gagal karena masalah konektivitas jaringan atau peran IAM yang salah. Dalam beberapa kasus, upaya mengaktifkan autentikasi Kerberos mungkin gagal saat Anda membuat atau memodifikasi instans basis data. Jika ini terjadi, periksa bahwa Anda menggunakan peran IAM yang benar, dan lalu ubah instans basis data untuk bergabung dengan domain.

Menghubungi RDS for Db2 dengan autentikasi Kerberos

Untuk menghubungi RDS for Db2 dengan autentikasi Kerberos

1. Pada penggugah/prompt perintah, jalankan perintah berikut. Dalam contoh berikut, ganti *nama pengguna* dengan nama pengguna Microsoft Active Directory Anda.

```
kinit username
```

2. Jika instans basis data RDS for Db2 menggunakan VPC yang dapat diakses oleh publik, tambahkan alamat IP untuk titik akhir instans basis data Anda ke file `/etc/hosts` Anda di klien Amazon EC2. Misalnya, perintah berikut memperoleh alamat IP, lalu memasukkannya ke file `/etc/hosts`.

```
% dig +short Db2-endpoint.AWS-Region.rds.amazonaws.com
;; Truncated, retrying in TCP mode.
ec2-34-210-197-118.AWS-Region.compute.amazonaws.com.
34.210.197.118

% echo "34.210.197.118  Db2-endpoint.AWS-Region.rds.amazonaws.com" >> /etc/hosts
```

3. Gunakan perintah berikut untuk masuk ke instans basis data RDS for Db2 yang terkait dengan Active Directory. Ganti *database_name* dengan nama basis data RDS for Db2 Anda.

```
db2 connect to database_name
```

Mengadministrasikan instans basis data RDS for Db2 Anda

Topik ini membahas tugas-tugas pengelolaan umum yang Anda lakukan dengan instans basis data RDS for Db2. Beberapa tugas sama untuk semua instans basis data Amazon RDS. Tugas-tugas lain bersifat khusus untuk RDS for Db2.

Tugas-tugas berikut bersifat umum untuk semua basis data RDS. Ada juga tugas-tugas khusus untuk RDS for Db2, seperti menghubungi basis data RDS for Db2 dengan klien SQL standar.

Area tugas	Dokumentasi terkait
Kelas instans, penyimpanan, dan PIOPS Jika Anda membuat instans produksi, pelajari cara kerja kelas instans, tipe penyimpanan, dan IOPS yang tersedia di Amazon RDS.	Kelas instans DB Jenis penyimpanan Amazon RDS
Deployment Multi-AZ Instans DB produksi harus menggunakan deployment Multi-AZ. Deployment Multi-AZ memberikan peningkatan ketersediaan, ketahanan data, dan toleransi kesalahan untuk instans DB.	Mengonfigurasi dan mengelola deployment Multi-AZ
Amazon VPC Jika Akun AWS Anda memiliki cloud privat virtual (VPC) bawaan, instans basis data Anda dibuat secara otomatis di dalam VPC bawaan. Jika akun Anda tidak memiliki VPC bawaan, dan Anda menginginkan instans basis data dalam VPC, buat VPC dan grup subnet sebelum Anda membuat instans.	Bekerja dengan klaster DB dalam VPC
Grup keamanan Secara default, instans DB menggunakan firewall yang mencegah akses. Pastikan Anda membuat grup keamanan dengan alamat IP dan konfigurasi jaringan yang benar untuk mengakses instans DB.	Mengontrol akses dengan grup keamanan

Area tugas	Dokumentasi terkait
Grup parameter Karena instans basis data RDS for Db2 mengharuskan Anda menambahkan parameter-parameter `rds.ibm_customer_id` dan `rds.ibm_site_id`, buat grup parameter sebelum Anda membuat instans basis data. Jika instans basis data Anda membutuhkan parameter-parameter basis data spesifik yang lain, tambahkan juga semua parameter itu ke grup parameter ini sebelum Anda membuat instans basis data.	Menambahkan ID IBM ke grup parameter Bekerja dengan grup parameter
Menghubungi instans basis data Setelah membuat grup keamanan dan mengaitkannya dengan instans basis data, Anda dapat menghubungi instans basis data dengan sebarang aplikasi klien SQL standar seperti IBM Db2 CLP.	Menghubungkan ke instans DB RDS untuk Db2 Anda
Pencadangan dan pemulihan Anda dapat mengonfigurasikan instans basis data Anda agar mengambil cadangan penyimpanan otomatis, atau mengambil cuplikan penyimpanan manual, lalu memulihkan instans dari cadangan atau cuplikan itu.	Mencadangkan, memulihkan, dan mengekspor data

Area tugas	Dokumentasi terkait
Pemantauan Anda dapat memantau instans basis data RDS for Db2 dengan IBM Db2 Data Management Console. Anda juga dapat memantau instans RDS untuk Db2 DB dengan menggunakan metrik CloudWatch Amazon RDS, peristiwa, dan pemantauan yang disempurnakan.	Menghubungi instans basis data RDS for Db2 dengan IBM Db2 Data Management Console Melihat metrik di konsol Amazon RDS Melihat peristiwa Amazon RDS Memantau metrik OS dengan Pemantauan yang Disempurnakan
File log Anda dapat mengakses file log untuk instans basis data RDS for Db2.	Memantau file log Amazon RDS

Topik

- Melakukan tugas-tugas sistem umum untuk instans basis data RDS for Db2
- Melakukan tugas-tugas basis data umum untuk instans basis data Amazon RDS for Db2

Melakukan tugas-tugas sistem umum untuk instans basis data RDS for Db2

Anda dapat melakukan tugas-tugas administrator basis data umum tertentu yang terkait dengan sistem di instans basis data Amazon RDS Anda yang menjalankan Db2. Untuk memberikan pengalaman layanan terkelola, Amazon RDS tidak memberikan akses shell ke instans basis data, dan membatasi akses ke sejumlah prosedur dan tabel sistem tertentu yang memerlukan privilese lanjut.

Topik

- Membuat titik akhir basis data kustom
- Memberikan dan mencabut privilese

- Melampirkan pada instans basis data RDS for Db2 jauh

Membuat titik akhir basis data kustom

Ketika bermigrasi ke RDS for Db2, Anda dapat menggunakan URL titik akhir basis data kustom untuk meminimalkan perubahan pada aplikasi Anda. Misalnya, jika `db2.example.com` digunakan sebagai rekam DNS Anda saat ini, Anda dapat menambahkannya ke Amazon Route 53. Di Route 53, Anda dapat menggunakan zona-zona yang di-hosting privat untuk memetakan titik akhir basis data DNS Anda saat ini ke titik akhir basis data RDS for Db2. Untuk menambahkan rekam `A` atau `CNAME` kustom bagi titik akhir basis data Amazon RDS, lihat Mendaftarkan dan mengelola domain dengan menggunakan Amazon Route 53 dalam Panduan Pengembang Amazon Route 53.

> **Note**
>
> Jika domain Anda tidak dapat ditransfer ke Route 53, Anda dapat meminta penyedia DNS Anda membuat rekam `CNAME` bagi URL titik akhir basis data RDS for Db2. Bacalah dokumentasi penyedia DNS Anda.

Memberikan dan mencabut privilese

Pengguna mendapatkan akses ke basis data melalui keanggotaan dalam grup yang dilampirkan pada basis data. Jika Anda menghapus semua grup yang dilampirkan pada basis data dari seorang pengguna, maka pengguna itu tidak dapat menghubungi basis data.

Gunakan prosedur-prosedur berikut untuk memberikan dan mencabut privilese untuk mengontrol akses ke basis data Anda.

Prosedur-prosedur ini menggunakan IBM Db2 CLP yang berjalan pada mesin lokal untuk menghubungi instans basis data RDS for Db2. Pastikan untuk membuat katalog simpul TCPIP dan basis data untuk menghubungi instans basis data RDS for Db2 yang berjalan di mesin lokal Anda. Untuk informasi selengkapnya, lihat Menghubungi instans basis data RDS for Db2 dengan IBM Db2 CLP.

Topik

- Memberikan akses ke basis data Anda kepada pengguna
- Mengubah kata sandi pengguna
- Menambahkan grup ke pengguna

- Menghapus grup dari pengguna
- Menghapus pengguna
- Memerinci pengguna
- Membuat peran
- Memberikan peran
- Pemberian otorisasi database
- Membatalkan otorisasi basis data

Memberikan akses ke basis data Anda kepada pengguna

Untuk memberikan akses ke basis data Anda kepada seorang pengguna

1. Hubungi basis data `rdsadmin` dengan menggunakan nama pengguna master dan kata sandi master untuk instans basis data RDS for Db2 Anda. Dalam contoh berikut, ganti *`master_username`* dan *`master_password`* dengan informasi Anda sendiri.

   ```
   db2 connect to rdsadmin user master_username using master_password
   ```

   Perintah ini menghasilkan output yang serupa dengan contoh berikut:

   ```
   Database Connection Information

   Database server        = DB2/LINUXX8664 11.5.8.0
   SQL authorization ID   = ADMIN
   Local database alias   = RDSADMIN
   ```

2. Tambahkan pengguna ke daftar otorisasi Anda dengan memanggil `rdsadmin.add_user`. Untuk informasi selengkapnya, lihat rdsadmin.add_user.

   ```
   db2 "call rdsadmin.add_user(
       'username',
       'password',
       'group_name,group_name')"
   ```

3. (Opsional) Tambahkan grup lain ke pengguna dengan memanggil `rdsadmin.add_groups`. Untuk informasi selengkapnya, lihat rdsadmin.add_groups.

   ```
   db2 "call rdsadmin.add_groups(
   ```

```
    'username',
    'group_name,group_name')"
```

4. Tegaskan otoritas yang tersedia untuk pengguna. Dalam contoh berikut, ganti *rds_database_alias*, *master_user*, dan *master_password* dengan informasi Anda sendiri. Juga, ganti *username* dengan nama pengguna si pengguna.

```
db2 terminate
db2 connect to rds_database_alias user master_user using master_password
db2 "SELECT SUBSTR(AUTHORITY,1,20) AUTHORITY, D_USER, D_GROUP, D_PUBLIC
       FROM TABLE (SYSPROC.AUTH_LIST_AUTHORITIES_FOR_AUTHID ('username', 'U') ) AS
 T
       ORDER BY AUTHORITY"
```

   Perintah ini menghasilkan output yang serupa dengan contoh berikut:

```
AUTHORITY            D_USER D_GROUP D_PUBLIC
-------------------- ------ ------- --------
ACCESSCTRL           N      N       N
BINDADD              N      N       N
CONNECT              N      N       N
CREATETAB            N      N       N
CREATE_EXTERNAL_ROUT N      N       N
CREATE_NOT_FENCED_RO N      N       N
CREATE_SECURE_OBJECT N      N       N
DATAACCESS           N      N       N
DBADM                N      N       N
EXPLAIN              N      N       N
IMPLICIT_SCHEMA      N      N       N
LOAD                 N      N       N
QUIESCE_CONNECT      N      N       N
SECADM               N      N       N
SQLADM               N      N       N
SYSADM               *      N       *
SYSCTRL              *      N       *
SYSMAINT             *      N       *
SYSMON               *      N       *
WLMADM               N      N       N
```

5. Berikan peran-peran RDS for Db2 `ROLE_NULLID_PACKAGES`, `ROLE_TABLESPACES`, dan `ROLE_PROCEDURES` ke grup tempat Anda menambahkan pengguna.

> **Note**
>
> Kami membuat instans basis data RDS for Db2 dalam mode `RESTRICTIVE`. Oleh karena itu, peran-peran RDS for Db2 `ROLE_NULLID_PACKAGES`, `ROLE_TABLESPACES`, dan `ROLE_PROCEDURES` memberikan privilese eksekusi pada paket-paket `NULLID` untuk IBM Db2 CLP dan Dynamic SQL. Peran-peran ini juga memberi pengguna privilese pada ruang tabel.

a. Connect ke database Db2 Anda. Dalam contoh berikut, ganti *`database_name`*, *`master_user`*, dan *`master_password`* dengan informasi Anda sendiri.

```
db2 connect to database_name user master_user using master_password
```

b. Berikan peran `ROLE_NULLED_PACKAGES` kepada sebuah grup. Dalam contoh berikut, ganti *`group_name`* dengan nama grup yang ingin Anda tambahi peran.

```
db2 "grant role ROLE_NULLID_PACKAGES to group group_name"
```

c. Berikan peran `ROLE_TABLESPACES` kepada grup yang sama. Dalam contoh berikut, ganti *`group_name`* dengan nama grup yang ingin Anda tambahi peran.

```
db2 "grant role ROLE_TABLESPACES to group group_name"
```

d. Berikan peran `ROLE_PROCEDURES` kepada grup yang sama. Dalam contoh berikut, ganti *`group_name`* dengan nama grup yang ingin Anda tambahi peran.

```
db2 "grant role ROLE_PROCEDURES to group group_name"
```

6. Berikan otoritas-otoritas `connect`, `bindadd`, `createtab`, dan `IMPLICIT_SCHEMA` kepada grup yang Anda tambahi pengguna. Dalam contoh berikut, ganti *`group_name`* dengan nama grup kedua yang ingin Anda tambahi pengguna.

```
db2 "grant usage on workload SYSDEFAULTUSERWORKLOAD to public"
db2 "grant connect, bindadd, createtab, implicit_schema on database to
 group group_name"
```

7. Ulangi langkah 4 hingga 6 untuk setiap grup lain yang Anda tambahi pengguna.

8. Coba akses pengguna dengan menghubungkan sebagai pengguna, membuat tabel, memasukkan nilai ke dalam tabel, dan mengambil data dari tabel. Dalam contoh berikut, ganti *rds_database_alias*, *username*, dan *password* dengan nama basis data dan nama pengguna dan kata sandi si pengguna.

```
db2 connect to rds_database_alias user username using password
db2 "create table t1(c1 int not null)"
db2 "insert into t1 values (1),(2),(3),(4)"
db2 "select * from t1"
```

Mengubah kata sandi pengguna

Untuk mengubah kata sandi seorang pengguna

1. Hubungi basis data `rdsadmin` dengan menggunakan nama pengguna master dan kata sandi master untuk instans basis data RDS for Db2 Anda. Dalam contoh berikut, ganti *master_username* dan *master_password* dengan informasi Anda sendiri.

```
db2 connect to rdsadmin user master_username using master_password
```

2. Ubah kata sandi dengan memanggil `rdsadmin.change_password`. Untuk informasi selengkapnya, lihat rdsadmin.change_password.

```
db2 "call rdsadmin.change_password(
    'username',
    'new_password')"
```

Menambahkan grup ke pengguna

Untuk menambahkan grup ke seorang pengguna

1. Hubungi basis data `rdsadmin` dengan menggunakan nama pengguna master dan kata sandi master untuk instans basis data RDS for Db2 Anda. Dalam contoh berikut, ganti *master_username* dan *master_password* dengan informasi Anda sendiri.

```
db2 connect to rdsadmin user master_username using master_password
```

2. Tambahkan grup ke pengguna dengan memanggil `rdsadmin.add_groups`. Untuk informasi selengkapnya, lihat rdsadmin.add_groups.

```
db2 "call rdsadmin.add_groups(
    'username',
    'group_name,group_name')"
```

Menghapus grup dari pengguna

Untuk menghapus grup dari seorang pengguna

1. Hubungi basis data `rdsadmin` dengan menggunakan nama pengguna master dan kata sandi master untuk instans basis data RDS for Db2 Anda. Dalam contoh berikut, ganti *master_username* dan *master_password* dengan informasi Anda sendiri.

```
db2 connect to rdsadmin user master_username using master_password
```

2. Hapus grup dengan memanggil `rdsadmin.remove_groups`. Untuk informasi selengkapnya, lihat rdsadmin.remove_groups.

> **Warning**
>
> Jika Anda menghapus semua grup yang dilampirkan pada basis data dari seorang pengguna, maka pengguna itu tidak dapat menghubungi basis data. Ini karena Amazon RDS memberikan wewenang kepada grup, bukan pengguna.

```
db2 "call rdsadmin.remove_groups(
    'username',
    'group_name,group_name')"
```

Menghapus pengguna

Untuk menghapus seorang pengguna dari daftar otorisasi

1. Hubungi basis data `rdsadmin` dengan menggunakan nama pengguna master dan kata sandi master untuk instans basis data RDS for Db2 Anda. Dalam contoh berikut, ganti *master_username* dan *master_password* dengan informasi Anda sendiri.

```
db2 connect to rdsadmin user master_username using master_password
```

2. Hapus pengguna dari daftar otorisasi dengan memanggil `rdsadmin.remove_user`. Untuk informasi selengkapnya, lihat rdsadmin.remove_user.

```
db2 "call rdsadmin.remove_user('username')"
```

Memerinci pengguna

Untuk memerinci pengguna pada sebuah daftar otorisasi, panggil prosedur tersimpan `rdsadmin.list_users`. Untuk informasi selengkapnya, lihat rdsadmin.list_users.

```
db2 "call rdsadmin.list_users()"
```

Membuat peran

Anda dapat menggunakan prosedur tersimpan rdsadmin.create_role untuk membuat peran.

Untuk membuat grup

1. Hubungi basis data `rdsadmin`. Dalam contoh berikut, ganti *master_username* dan *master_password* dengan informasi Anda.

```
db2 connect to rdsadmin user master_username using master_password
```

2. Atur Db2 ke konten keluaran.

```
db2 set serveroutput on
```

3. Buat peran. Untuk informasi selengkapnya, lihat the section called “rdsadmin.create_role”.

```
db2 "call rdsadmin.create_role(
    'database_name',
    'role_name')"
```

4. Setel Db2 agar tidak menampilkan konten.

```
db2 set serveroutput off
```

Memberikan peran

Anda dapat menggunakan prosedur rdsadmin.grant_role tersimpan untuk menetapkan peran ke peran, pengguna, atau grup.

Untuk menetapkan peran

1. Hubungi basis data `rdsadmin`. Dalam contoh berikut, ganti *`master_username`* dan *`master_password`* dengan informasi Anda.

```
db2 connect to rdsadmin user master_username using master_password
```

2. Atur Db2 ke konten keluaran.

```
db2 set serveroutput on
```

3. Tetapkan peran. Untuk informasi selengkapnya, lihat the section called “rdsadmin.grant_role”.

```
db2 "call rdsadmin.grant_role(
    'database_name',
    'role_name',
    'grantee',
    'admin_option')"
```

4. Setel Db2 agar tidak menampilkan konten.

```
db2 set serveroutput off
```

Pemberian otorisasi database

Pengguna master, yang memiliki `DBADM` otorisasi, dapat memberikan`DBADM`,`ACCESSCTRL`, atau `DATAACCESS` otorisasi untuk peran, pengguna, atau grup.

Untuk memberikan otorisasi database

1. Hubungi basis data `rdsadmin` dengan menggunakan nama pengguna master dan kata sandi master untuk instans basis data RDS for Db2 Anda. Dalam contoh berikut, ganti *`master_username`* dan *`master_password`* dengan informasi Anda sendiri.

```
db2 connect to rdsadmin user master_username using master_password
```

2. Berikan akses pengguna dengan meneleponrdsadmin.dbadm_grant. Untuk informasi selengkapnya, lihat rdsadmin.dbadm_grant.

```
db2 "call rdsadmin.dbadm_grant(
    ?,
    'database_name',
    'authorization',
    'grantee')"
```

Contoh kasus penggunaan

Prosedur berikut memandu Anda melalui pembuatan peran, memberikan DBADM otorisasi untuk peran, dan menetapkan peran kepada pengguna.

Untuk membuat peran, berikan **DBADM** otorisasi, dan tetapkan peran ke pengguna

1. Hubungi basis data rdsadmin dengan menggunakan nama pengguna master dan kata sandi master untuk instans basis data RDS for Db2 Anda. Dalam contoh berikut, ganti *master_username* dan *master_password* dengan informasi Anda sendiri.

```
db2 connect to rdsadmin user master_username using master_password
```

2. Buat peran yang dipanggil PROD_ROLE untuk database yang disebutTESTDB. Untuk informasi selengkapnya, lihat rdsadmin.create_role.

```
db2 "call rdsadmin.create_role(
    'TESTDB',
    'PROD_ROLE')"
```

3. Tetapkan peran ke pengguna yang dipanggilPROD_USER. PROD_USEROtorisasi admin diberikan untuk menetapkan peran. Untuk informasi selengkapnya, lihat rdsadmin.grant_role.

```
db2 "call rdsadmin.grant_role(
    ?,
    'TESTDB',
    'PROD_ROLE',
    'USER PROD_USER',
    'Y')"
```

4. (Opsional) Berikan otorisasi atau hak istimewa tambahan. Contoh berikut memberikan `DBADM` otorisasi untuk peran bernama `PROD_ROLE` untuk database yang disebut. `FUNDPROD` Untuk informasi selengkapnya, lihat rdsadmin.dbadm_grant.

```
db2 "call rdsadmin.dbadm_grant(
    ?,
    'FUNDPROD',
    'DBADM',
    'ROLE PROD_ROLE')"
```

5. Hentikan sesi Anda.

```
db2 terminate
```

6. Hubungi basis data `testdb` dengan menggunakan nama pengguna master dan kata sandi master untuk instans basis data RDS for Db2 Anda. Dalam contoh berikut, ganti *`master_username`* dan *`master_password`* dengan informasi Anda sendiri.

```
db2 connect to testdb user master_username using master_password
```

7. Tambahkan lebih banyak otorisasi ke peran.

```
db2 "grant connect, implicit_schema on database to role PROD_ROLE"
```

Membatalkan otorisasi basis data

Pengguna master, yang memiliki `DBADM` otorisasi, dapat mencabut, `DBADMACCESSCTRL`, atau `DATAACCESS` otorisasi dari peran, pengguna, atau grup.

Untuk mencabut otorisasi basis data

1. Hubungi basis data `rdsadmin` dengan menggunakan nama pengguna master dan kata sandi master untuk instans basis data RDS for Db2 Anda. Dalam contoh berikut, ganti *`master_username`* dan *`master_password`* dengan informasi Anda sendiri.

```
db2 connect to rdsadmin user master_username using master_password
```

2. Cabut akses pengguna dengan menelepon. `rdsadmin.dbadm_revoke` Untuk informasi selengkapnya, lihat rdsadmin.dbadm_revoke.

```
db2 "call rdsadmin.dbadm_revoke(
    ?,
    'database_name',
    'authorization',
    'grantee')"
```

Melampirkan pada instans basis data RDS for Db2 jauh

Untuk melampirkan pada instans basis data RDS for Db2 jauh

1. Jalankan sesi IBM Db2 CLP sisi klien. Lihat informasi tentang membuat katalog instans basis data dan basis data RDS for Db2 di Menghubungi instans basis data RDS for Db2 dengan IBM Db2 CLP. Catat nama pengguna master dan kata sandi master untuk instans basis data RDS for Db2 Anda.
2. Lampirkan pada instans basis data RDS for Db2. Dalam contoh berikut, ganti *node_name*, *master_username*, dan *master_password* dengan nama simpul TCPIP yang Anda buatkan katalog dan nama pengguna master dan kata sandi master untuk instans basis data RDS for Db2 Anda.

```
db2 attach to node_name user master_username using master_password
```

Setelah melampirkan pada instans basis data RDS for Db2 jauh, Anda dapat menjalankan perintah berikut dan perintah-perintah `get snapshot` lainnya. Lihat informasi yang lebih lengkap di perintah GET SNAPSHOT dalam dokumentasi IBM Db2.

```
db2 list applications
db2 get snapshot for all databases
db2 get snapshot for database manager
db2 get snapshot for all applications
```

Melakukan tugas-tugas basis data umum untuk instans basis data Amazon RDS for Db2

Anda dapat melakukan tugas-tugas DBA umum tertentu yang terkait dengan basis data di instans basis data RDS for Db2 Anda. Untuk memberikan pengalaman layanan terkelola, Amazon RDS tidak

memberikan akses shell ke instans basis data. Selain itu, pengguna master tidak dapat menjalankan perintah atau utilitas yang membutuhkan otoritas `SYSADM`, `SYSMAINT`, atau `SYSCTRL`.

Topik

- Mengelola kolam penyangga
- Mengelola penyimpanan
- Mengelola ruang tabel
- Menghasilkan laporan kinerja

Mengelola kolam penyangga

Anda dapat membuat, mengubah, atau mengedrop kolam penyangga untuk basis data RDS for Db2. Membuat, mengubah, atau mengedrop kolam penyangga membutuhkan otoritas `SYSADMIN` dengan tingkat lebih tinggi, yang tidak tersedia untuk pengguna master. Sebagai gantinya, gunakan prosedur-prosedur tersimpan Amazon RDS.

Anda juga dapat menguras kolam penyangga.

Topik

- Membuat kolam penyangga
- Mengubah kolam penyangga
- Mengedrop kolam penyangga
- Menguras kolam penyangga

Membuat kolam penyangga

Untuk membuat kolam penyangga bagi basis data RDS for Db2 Anda, panggil prosedur tersimpan `rdsadmin.create_bufferpool`. Lihat informasi yang lebih lengkap di pernyataan CREATE BUFFERPOOL dalam dokumentasi IBM Db2.

Untuk membuat kolam penyangga

1. Hubungi basis data `rdsadmin` dengan menggunakan nama pengguna master dan kata sandi master untuk instans basis data RDS for Db2 Anda. Dalam contoh berikut, ganti *`master_username`* dan *`master_password`* dengan informasi Anda sendiri.

```
db2 "connect to rdsadmin user master_user using master_password"
```

2. Buat kolam penyangga dengan memanggil `rdsadmin.create_bufferpool`. Untuk informasi selengkapnya, lihat rdsadmin.create_bufferpool.

```
db2 "call rdsadmin.create_bufferpool(
    'database_name',
    'buffer_pool_name',
    buffer_pool_size,
    'immediate',
    'automatic',
    page_size,
    number_block_pages,
    block_size)"
```

Mengubah kolam penyangga

Untuk mengubah kolam penyangga bagi basis data RDS for Db2 Anda, panggil prosedur tersimpan `rdsadmin.alter_bufferpool`. Lihat informasi yang lebih lengkap di pernyataan ALTER BUFFERPOOL dalam dokumentasi IBM Db2.

Untuk mengubah kolam penyangga

1. Hubungi basis data `rdsadmin` dengan menggunakan nama pengguna master dan kata sandi master untuk instans basis data RDS for Db2 Anda. Dalam contoh berikut, ganti *master_username* dan *master_password* dengan informasi Anda sendiri.

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. Ubah kolam penyangga dengan memanggil `rdsadmin.alter_bufferpool`. Untuk informasi selengkapnya, lihat rdsadmin.alter_bufferpool.

```
db2 "call rdsadmin.alter_bufferpool(
    'database_name',
    'buffer_pool_name',
    buffer_pool_size,
    'immediate',
    'automatic',
    change_number_blocks,
    number_block_pages,
```

```
    block_size)"
```

Mengedrop kolam penyangga

Untuk mengedrop kolam penyangga bagi basis data RDS for Db2 Anda, panggil prosedur tersimpan `rdsadmin.drop_bufferpool`. Lihat informasi yang lebih lengkap di Mengedrop kolam penyangga dalam dokumentasi IBM Db2.

> **Important**
>
> Pastikan bahwa tidak ada ruang tabel yang ditetapkan untuk kolam penyangga yang ingin Anda drop.

Untuk mengedrop kolam penyangga

1. Hubungi basis data `rdsadmin` dengan menggunakan nama pengguna master dan kata sandi master untuk instans basis data RDS for Db2 Anda. Dalam contoh berikut, ganti *master_username* dan *master_password* dengan informasi Anda sendiri.

   ```
   db2 "connect to rdsadmin user master_user using master_password"
   ```

2. Drop kolam penyangga dengan memanggil `rdsadmin.drop_bufferpool`. Untuk informasi selengkapnya, lihat rdsadmin.drop_bufferpool.

   ```
   db2 "call rdsadmin.drop_bufferpool(
       'database_name',
       'buffer_pool_name')"
   ```

Menguras kolam penyangga

Anda dapat menguras kolam penyangga untuk memaksakan sebuah titik cek sehingga RDS for Db2 menulis halaman dari memori ke penyimpanan.

> **Note**
>
> Anda tidak perlu menguras kolam penyangga. Db2 menulis log secara sinkron sebelum menuntaskan/commit transaksi. Halaman kotor mungkin masih ada di kolam penyangga,

tetapi Db2 menulisnya ke penyimpanan secara asinkron. Walaupun sistem mati secara tidak terduga, ketika Anda memulai ulang basis data, Db2 melakukan secara otomatis pemulihan kemacetan itu. Selama pemulihan kemacetan, Db2 menulis perubahan yang tuntas/commit ke basis data atau menggulirkan balik perubahan untuk transaksi yang tidak tuntas.

Untuk menguras kolam penyangga

1. Connect ke database Db2 Anda menggunakan nama pengguna master dan kata sandi utama untuk RDS Anda untuk instans Db2 DB. Dalam contoh berikut, ganti *rds_database_alias*, *master_username*, dan *master_password* dengan informasi Anda sendiri.

```
db2 connect to rds_database_alias user master_username using master_password
```

2. Kuras kolam penyangga.

```
db2 flush bufferpools all
```

Mengelola penyimpanan

Db2 menggunakan penyimpanan otomatis untuk mengelola penyimpanan fisik bagi objek basis data seperti tabel, indeks, dan file sementara. Alih-alih mengalokasikan ruang penyimpanan secara manual dan melacak jalur penyimpanan yang digunakan, penyimpanan otomatis memungkinkan sistem Db2 membuat dan mengelola jalur penyimpanan sebagaimana dibutuhkan. Ini dapat menyederhanakan administrasi basis data Db2 dan mengurangi kemungkinan kesalahan karena kealpaan manusia. Lihat informasi yang lebih lengkap di Penyimpanan otomatis dalam dokumentasi IBM Db2.

Dengan RDS for Db2, Anda dapat menambah secara dinamis ukuran penyimpanan dengan ekspansi otomatis volume logis dan sistem file. Untuk informasi selengkapnya, lihat Menggunakan penyimpanan untuk instans DB Amazon RDS.

Mengelola ruang tabel

Anda dapat membuat, mengubah, atau mengedrop ruang tabel untuk basis data RDS for Db2. Membuat, mengubah, atau mengedrop ruang tabel memerlukan otoritas `SYSADM` dengan tingkat lebih tinggi, yang tidak tersedia untuk pengguna master. Sebagai gantinya, gunakan prosedur-prosedur tersimpan Amazon RDS.

Topik

- Membuat ruang tabel
- Mengubah ruang tabel
- Mengedrop ruang tabel
- Memeriksa status ruang tabel
- Menghasilkan informasi terperinci tentang ruang tabel
- Memerinci status dan grup penyimpanan bagi ruang tabel
- Memerinci ruang tabel sebuah tabel
- Memerinci kontainer ruang tabel

Membuat ruang tabel

Untuk membuat ruang tabel bagi basis data RDS for Db2 Anda, panggil prosedur tersimpan `rdsadmin.create_tablespace`. Lihat informasi yang lebih lengkap di pernyataan CREATE TABLESPACE dalam dokumentasi IBM Db2.

> **Important**
>
> Untuk membuat ruang tabel, Anda harus memiliki kolam penyangga dengan ukuran halaman yang sama untuk dikaitkan dengan ruang tabel. Untuk informasi selengkapnya, lihat Mengelola kolam penyangga.

Untuk membuat ruang tabel

1. Hubungi basis data `rdsadmin` dengan menggunakan nama pengguna master dan kata sandi master untuk instans basis data RDS for Db2 Anda. Dalam contoh berikut, ganti *`master_username`* dan *`master_password`* dengan informasi Anda sendiri.

   ```
   db2 "connect to rdsadmin user master_username using master_password"
   ```

2. Buat ruang tabel dengan memanggil `rdsadmin.create_tablespace`. Untuk informasi selengkapnya, lihat rdsadmin.create_tablespace.

   ```
   db2 "call rdsadmin.create_tablespace(
       'database_name',
   ```

```
    'tablespace_name',
    'buffer_pool_name',
    tablespace_initial_size,
    tablespace_increase_size,
    'tablespace_type')"
```

Mengubah ruang tabel

Untuk mengubah ruang tabel bagi basis data RDS for Db2 Anda, panggil prosedur tersimpan `rdsadmin.alter_tablespace`. Anda dapat menggunakan prosedur tersimpan ini untuk mengubah kolam penyangga sebuah ruang tabel, menurunkan tanda air tinggi, atau membawa daring ruang tabel. Lihat informasi yang lebih lengkap di pernyataan ALTER TABLESPACE dalam dokumentasi IBM Db2.

Untuk mengubah ruang tabel

1. Hubungi basis data `rdsadmin` dengan menggunakan nama pengguna master dan kata sandi master untuk instans basis data RDS for Db2 Anda. Dalam contoh berikut, ganti *master_username* dan *master_password* dengan informasi Anda sendiri.

   ```
   db2 "connect to rdsadmin user master_username using master_password"
   ```

2. Ubah ruang tabel dengan memanggil `rdsadmin.alter_tablespace`. Untuk informasi selengkapnya, lihat rdsadmin.alter_tablespace.

   ```
   db2 "call rdsadmin.alter_tablespace(
       'database_name',
       'tablespace_name',
       'buffer_pool_name',
       buffer_pool_size,
       tablespace_increase_size,
       'max_size', 'reduce_max',
       'reduce_stop',
       'reduce_value',
       'lower_high_water',
       'lower_high_water_stop',
       'switch_online')"
   ```

Mengedrop ruang tabel

Untuk mengedrop ruang tabel bagi basis data RDS for Db2 Anda, panggil prosedur tersimpan `rdsadmin.drop_tablespace`. Sebelum Anda mengedrop ruang tabel, drop dahulu semua objek di ruang tabel seperti tabel, indeks, atau objek besar (LOB). Lihat informasi yang lebih lengkap di Mengedrop ruang tabel dalam dokumentasi IBM Db2.

Untuk mengedrop ruang tabel

1. Hubungi basis data `rdsadmin` dengan menggunakan nama pengguna master dan kata sandi master untuk instans basis data RDS for Db2 Anda. Dalam contoh berikut, ganti *`master_username`* dan *`master_password`* dengan informasi Anda sendiri.

   ```
   db2 "connect to rdsadmin user master_username using master_password"
   ```

2. Drop ruang tabel dengan memanggil `rdsadmin.drop_tablespace`. Untuk informasi selengkapnya, lihat rdsadmin.drop_tablespace.

   ```
   db2 "call rdsadmin.drop_tablespace(
       'database_name',
       'tablespace_name')"
   ```

Memeriksa status ruang tabel

Anda dapat memeriksa status ruang tabel dengan menggunakan perintah `cast`.

Untuk memeriksa status ruang tabel

1. Connect ke database Db2 Anda menggunakan nama pengguna master dan kata sandi utama untuk RDS Anda untuk instans Db2 DB. Dalam contoh berikut, ganti *`rds_database_alias`*, *`master_username`*, dan *`master_password`* dengan informasi Anda sendiri.

   ```
   db2 connect to rds_database_alias user master_username using master_password
   ```

2. Menghasilkan output ringkasan.

   Untuk output ringkasan:

   ```
   db2 "select cast(tbsp_id as smallint) as tbsp_id,
   cast(tbsp_name as varchar(35)) as tbsp_name,
   ```

```
cast(tbsp_type as varchar(3)) as tbsp_type,
cast(tbsp_state as varchar(10)) as state,
cast(tbsp_content_type as varchar(8)) as contents from
 table(mon_get_tablespace(null,-1)) order by tbsp_id"
```

Menghasilkan informasi terperinci tentang ruang tabel

Untuk menghasilkan informasi terperinci tentang ruang tabel

1. Connect ke database Db2 Anda menggunakan nama pengguna master dan kata sandi utama untuk RDS Anda untuk instans Db2 DB. Dalam contoh berikut, ganti *rds_database_alias*, *master_username*, dan *master_password* dengan informasi Anda sendiri.

   ```
   db2 connect to rds_database_alias user master_username using master_password
   ```

2. Menghasilkan perincian semua ruang tabel dalam basis data untuk satu anggota atau semua anggota.

   Untuk satu anggota:

   ```
   db2 "select cast(member as smallint) as member,
   cast(tbsp_id as smallint) as tbsp_id,
   cast(tbsp_name as varchar(35)) as tbsp_name,
   cast(tbsp_type as varchar(3)) as tbsp_type,
   cast(tbsp_state as varchar(10)) as state,
   cast(tbsp_content_type as varchar(8)) as contents,
   cast(tbsp_total_pages as integer) as total_pages,
   cast(tbsp_used_pages as integer) as used_pages,
   cast(tbsp_free_pages as integer) as free_pages,
   cast(tbsp_page_top as integer) as page_hwm,
   cast(tbsp_page_size as integer) as page_sz,
   cast(tbsp_extent_size as smallint) as extent_sz,
   cast(tbsp_prefetch_size as smallint) as prefetch_sz,
   cast(tbsp_initial_size as integer) as initial_size,
   cast(tbsp_increase_size_percent as smallint) as increase_pct,
   cast(storage_group_name as varchar(12)) as stogroup from
    table(mon_get_tablespace(null,-1)) order by member, tbsp_id "
   ```

   Untuk semua anggota:

   ```
   db2 "select cast(member as smallint) as member
   ```

```
cast(tbsp_id as smallint) as tbsp_id,
cast(tbsp_name as varchar(35)) as tbsp_name,
cast(tbsp_type as varchar(3)) as tbsp_type,
cast(tbsp_state as varchar(10)) as state,
cast(tbsp_content_type as varchar(8)) as contents,
cast(tbsp_total_pages as integer) as total_pages,
cast(tbsp_used_pages as integer) as used_pages,
cast(tbsp_free_pages as integer) as free_pages,
cast(tbsp_page_top as integer) as page_hwm,
cast(tbsp_page_size as integer) as page_sz,
cast(tbsp_extent_size as smallint) as extent_sz,
cast(tbsp_prefetch_size as smallint) as prefetch_sz,
cast(tbsp_initial_size as integer) as initial_size,
cast(tbsp_increase_size_percent as smallint) as increase_pct,
cast(storage_group_name as varchar(12)) as stogroup from
 table(mon_get_tablespace(null,-2)) order by member, tbsp_id "
```

Memerinci status dan grup penyimpanan bagi ruang tabel

Untuk memerinci status dan grup penyimpanan bagi ruang tabel, jalankan pernyataan SQL berikut:

```
db2 "SELECT varchar(tbsp_name, 30) as tbsp_name,
                  varchar(TBSP_STATE, 30) state,
                  tbsp_type,
                  varchar(storage_group_name,30) storage_group
FROM TABLE(MON_GET_TABLESPACE('',-2)) AS t"
```

Memerinci ruang tabel sebuah tabel

Untuk memerinci ruang tabel sebuah tabel, jalankan pernyataan SQL berikut. Dalam contoh berikut, ganti *SCHEMA_NAME* dan *TABLE_NAME* dengan nama-nama skema dan tabel Anda.

```
db2 "SELECT
   VARCHAR(SD.TBSPACE,30) AS DATA_SPACE,
   VARCHAR(SL.TBSPACE,30) AS LONG_SPACE,
   VARCHAR(SI.TBSPACE,30) AS INDEX_SPACE
 FROM
   SYSCAT.DATAPARTITIONS P
   JOIN SYSCAT.TABLESPACES SD ON SD.TBSPACEID = P.TBSPACEID
   LEFT JOIN SYSCAT.TABLESPACES SL ON SL.TBSPACEID = P.LONG_TBSPACEID
   LEFT JOIN SYSCAT.TABLESPACES SI ON SI.TBSPACEID = P.INDEX_TBSPACEID
 WHERE
```

```
    TABSCHEMA = 'SCHEMA_NAME'
AND TABNAME   = 'TABLE_NAME'"
```

Memerinci kontainer ruang tabel

Untuk memerinci kontainer ruang tabel bagi sebuah ruang tabel

1. Connect ke database Db2 Anda menggunakan nama pengguna master dan kata sandi utama untuk RDS Anda untuk instans Db2 DB. Dalam contoh berikut, ganti *rds_database_alias*, *master_username*, dan *master_password* dengan informasi Anda sendiri.

   ```
   db2 connect to rds_database_alias user master_username using master_password
   ```

2. Menghasilkan daftar semua kontainer ruang tabel dalam basis data atau kontainer ruang tabel tertentu.

   Untuk semua kontainer ruang tabel:

   ```
   db2 "select cast(member as smallint) as member,
   cast(tbsp_name as varchar(35)) as tbsp_name,
   cast(container_id as smallint) as id,
   cast(container_name as varchar(60)) as container_path, container_type as type from
    table(mon_get_container(null,-2)) order by member,tbsp_id,container_id"
   ```

   Untuk kontainer ruang tabel tertentu:

   ```
   db2 "select cast(member as smallint) as member,
   cast(tbsp_name as varchar(35)) as tbsp_name,
   cast(container_id as smallint) as id,
   cast(container_name as varchar(60)) as container_path, container_type as type from
    table(mon_get_container('TBSP_1',-2)) order by member, tbsp_id,container_id"
   ```

Menghasilkan laporan kinerja

Anda dapat menghasilkan laporan kinerja dengan prosedur atau skrip. Lihat informasi tentang penggunaan prosedur di Prosedur DBSUMMARY - Menghasilkan laporan ringkasan metrik kinerja sistem dan aplikasi dalam dokumentasi IBM Db2.

Db2 menyertakan file `db2mon.sh` dalam direktori `~sqllib/sample/perf`-nya. Menjalankan skrip akan menghasilkan laporan metrik SQL yang berbiaya rendah dan ekstensif. Untuk mengunduh file `db2mon.sh` dan file skrip terkait, lihat direktori perf di repositori GitHub db2-samples IBM.

Untuk menghasilkan laporan kinerja dengan skrip

1. Connect ke database Db2 Anda menggunakan nama pengguna master dan kata sandi utama untuk RDS Anda untuk instans Db2 DB. Dalam contoh berikut, ganti *`master_username`* dan *`master_password`* dengan informasi Anda sendiri.

   ```
   db2 connect to rdsadmin user master_username using master_password
   ```

2. Buat kolam penyangga bernama `db2monbp` dengan ukuran halaman 4096 dengan memanggil `rdsadmin.create_bufferpool`. Untuk informasi selengkapnya, lihat rdsadmin.create_bufferpool.

   ```
   db2 "call rdsadmin.create_bufferpool('database_name','db2monbp',4096)"
   ```

3. Buat ruang tabel sementara bernama `db2montmptbsp` yang menggunakan kolam penyangga `db2monbp` dengan memanggil `rdsadmin.create_tablespace`. Untuk informasi selengkapnya, lihat rdsadmin.create_tablespace.

   ```
   db2 "call rdsadmin.create_tablespace('database_name',\
       'db2montmptbsp','db2monbp',4096,1000,100,'T')"
   ```

4. Buka skrip `db2mon.sh`, dan ubah baris tentang menghubungi basis data.

   a. Hapus baris berikut.

      ```
      db2 -v connect to $dbName
      ```

   b. Ganti baris pada langkah sebelumnya dengan baris berikut. Dalam contoh berikut, ganti *`master_username`* dan *`master_password`* dengan nama pengguna master dan kata sandi master untuk instans basis data RDS for Db2 Anda.

      ```
      db2 -v connect to $dbName user master_username using master_password
      ```

5. Ubah ke direktori tempat skrip terletak. Dalam contoh berikut, ganti *`directory`* dengan nama direktori tempat skrip terletak.

```
cd directory
```

6. Jalankan skrip db2mon.sh untuk menghasilkan laporan pada interval tertentu. Dalam contoh berikut, ganti *rds_database_alias* dan *seconds* dengan nama basis data Anda dan jumlah detik (0 hingga 3600) yang menyelang pembuatan laporan.

```
./db2mon.sh rds_database_alias seconds | tee -a db2mon.out
```

Mengintegrasikan instans basis data RDS for Db2 dengan Amazon S3

Anda dapat mentransfer file antara RDS untuk instans Db2 DB dan bucket Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) dengan prosedur tersimpan Amazon RDS. Untuk informasi selengkapnya, lihat Referensi prosedur tersimpan RDS for Db2.

> **Note**
>
> Instans DB dan bucket Amazon S3 Anda harus berada di Wilayah AWS yang sama.

Agar RDS for Db2 yang berintegrasi dengan Amazon S3, instans basis data Anda harus memiliki akses ke bucket Amazon S3 tempat RDS for Db2 berada. Jika saat ini Anda tidak memiliki bucket S3, buat bucket.

Topik

- Langkah 1: Buat kebijakan IAM
- Langkah 2: Buat peran IAM dan lampirkan kebijakan IAM Anda
- Langkah 3: Tambahkan peran IAM ke instans basis data RDS for Db2

Langkah 1: Buat kebijakan IAM

Pada langkah ini, Anda membuat kebijakan AWS Identity and Access Management (IAM) dengan izin yang diperlukan untuk mentransfer file dari bucket Amazon S3 ke instans RDS DB Anda. Langkah ini beranggapan bahwa Anda telah membuat bucket S3. Lihat informasi yang lebih lengkap di Membuat bucket dalam Panduan Pengguna Amazon S3.

Sebelum Anda membuat kebijakan, catat potongan-potongan informasi berikut:

- Amazon Resource Name (ARN) untuk bucket Anda
- ARN untuk kunci AWS Key Management Service (AWS KMS) Anda, jika bucket Anda menggunakan SSE-KMS atau SSE-S3 enkripsi.

Buat kebijakan IAM yang mencakup izin-izin berikut:

```
"kms:GenerateDataKey",
"kms:Decrypt",
"s3:PutObject",
"s3:GetObject",
"s3:AbortMultipartUpload",
"s3:ListBucket",
"s3:DeleteObject",
"s3:GetObjectVersion",
"s3:ListMultipartUploadParts"
```

Anda dapat membuat kebijakan IAM dengan menggunakan AWS Management Console atau AWS Command Line Interface (AWS CLI).

Konsol

Untuk membuat kebijakan IAM untuk mengizinkan Amazon RDS mengakses bucket Amazon S3

1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi, pilih Kebijakan.
3. Pilih Buat kebijakan, lalu pilih JSON.
4. Tambahkan tindakan berdasarkan layanan. Untuk mentransfer file dari bucket Amazon S3 ke Amazon RDS, Anda harus memilih izin bucket dan izin objek.
5. Perluas Sumber Daya. Anda harus menentukan sumber daya bucket dan objek.
6. Pilih Berikutnya.
7. Untuk Nama kebijakan, masukkan nama untuk kebijakan ini.
8. (Opsional) Untuk Deskripsi, masukkan deskripsi untuk kebijakan ini.
9. Pilih Buat kebijakan.

AWS CLI

Untuk membuat kebijakan IAM guna mengizinkan Amazon RDS mengakses bucket Amazon S3 Anda

1. Jalankan perintah create-policy. Dalam contoh berikut, ganti *iam_policy_name* dan *s3_bucket_name* dengan nama untuk kebijakan IAM Anda dan nama bucket Amazon S3 tempat basis data RDS for Db2 Anda berada.

   Untuk Linux, macOS, atau Unix:

```
aws iam create-policy \
    --policy-name iam_policy_name \
    --policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Action": [
            "kms:GenerateDataKey",
            "kms:Decrypt",
            "s3:PutObject",
            "s3:GetObject",
            "s3:AbortMultipartUpload",
            "s3:ListBucket",
            "s3:DeleteObject",
            "s3:GetObjectVersion",
            "s3:ListMultipartUploadParts"
          ],
          "Resource": [
            "arn:aws:s3:::s3_bucket_name/*",
            "arn:aws:s3:::s3_bucket_name"
          ]
        }
      ]
    }'
```

Untuk Windows:

```
aws iam create-policy ^
    --policy-name iam_policy_name ^
    --policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
            "Action": [
              "s3:PutObject",
              "s3:GetObject",
              "s3:AbortMultipartUpload",
              "s3:ListBucket",
              "s3:DeleteObject",
              "s3:GetObjectVersion",
```

```
            "s3:ListMultipartUploadParts"
          ],
          "Resource": [
            "arn:aws:s3:::s3_bucket_name/*",
            "arn:aws:s3:::s3_bucket_name"
          ]
        }
      ]
    }'
```

2. Setelah kebijakan dibuat, catat ARN kebijakan. Anda memerlukan ARN untuk Langkah 2: Buat peran IAM dan lampirkan kebijakan IAM Anda.

Lihat informasi tentang pembuatan kebijakan IAM di Membuat kebijakan IAM dalam Panduan Pengguna IAM.

Langkah 2: Buat peran IAM dan lampirkan kebijakan IAM Anda

Langkah ini beranggapan bahwa Anda telah membuat kebijakan IAM di Langkah 1: Buat kebijakan IAM. Pada langkah ini, Anda membuat peran IAM untuk instans basis data RDS for Db2 dan kemudian melampirkan kebijakan IAM ke peran itu.

Anda dapat membuat peran IAM untuk instans DB Anda dengan menggunakan AWS Management Console atau. AWS CLI

Konsol

Untuk membuat peran IAM dan melampirkan kebijakan IAM padanya

1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi, silakan pilih Peran.
3. Pilih Buat peran.
4. Untuk jenis entitas tepercaya, pilih Layanan AWS.
5. Untuk Layanan atau kasus penggunaan, pilih RDS, lalu pilih RDS –Tambah Peran ke Basis Data.
6. Pilih Berikutnya.
7. Untuk Kebijakan izin, cari dan pilih nama kebijakan IAM yang Anda buat.
8. Pilih Berikutnya.

9. Untuk Nama peran, masukkan nama peran.
10. (Opsional) Untuk Deskripsi, masukkan deskripsi untuk peran baru ini.
11. Pilih Buat peran.

AWS CLI

Untuk membuat peran IAM dan melampirkan kebijakan IAM padanya

1. Jalankan perintah create-role. Dalam contoh berikut, ganti *iam_role_name* dengan nama untuk peran IAM Anda.

   Untuk Linux, macOS, atau Unix:

```
aws iam create-role \
    --role-name iam_role_name \
    --assume-role-policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
            "Service": "rds.amazonaws.com"
          },
          "Action": "sts:AssumeRole"
        }
      ]
    }'
```

   Untuk Windows:

```
aws iam create-role ^
    --role-name iam_role_name ^
    --assume-role-policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
            "Service": "rds.amazonaws.com"
          },
          "Action": "sts:AssumeRole"
```

```
        }
      ]
    }'
```

2. Setelah peran dibuat, catat ARN peran tersebut. Anda memerlukan ARN untuk Langkah 3: Tambahkan peran IAM ke instans basis data RDS for Db2.
3. Jalankan perintah attach-role-policy. Dalam contoh berikut, ganti *iam_policy_arn* dengan ARN dari kebijakan IAM yang Anda buat di Langkah 1: Buat kebijakan IAM. Ganti *iam_role_name* dengan nama peran IAM yang baru saja Anda buat.

   Untuk Linux, macOS, atau Unix:

   ```
   aws iam attach-role-policy \
      --policy-arn iam_policy_arn \
      --role-name iam_role_name
   ```

   Untuk Windows:

   ```
   aws iam attach-role-policy ^
      --policy-arn iam_policy_arn ^
      --role-name iam_role_name
   ```

Lihat informasi yang lebih lengkap di Membuat peran untuk melimpahkan izin ke pengguna IAM dalam Panduan Pengguna IAM.

Langkah 3: Tambahkan peran IAM ke instans basis data RDS for Db2

Pada langkah ini, Anda menambahkan peran IAM ke instans basis data RDS for Db2. Perhatikan persyaratan berikut:

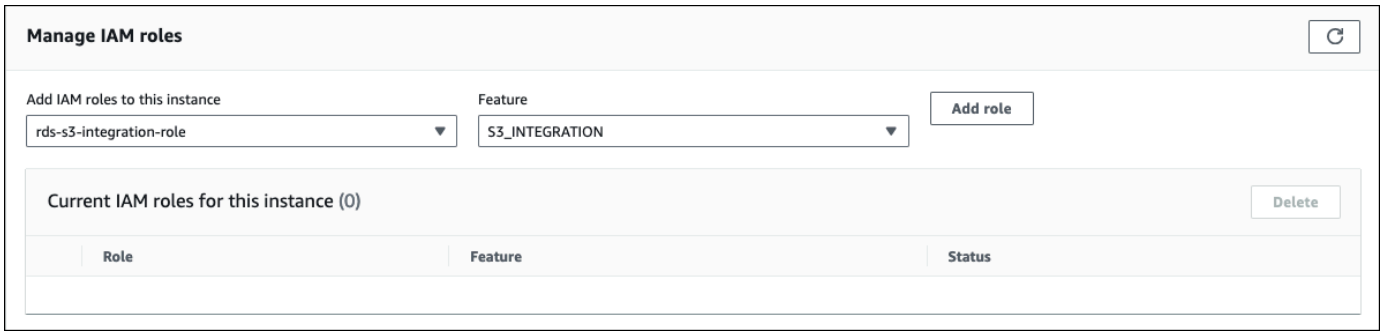
- Anda harus memiliki akses ke peran IAM dengan kebijakan izin Amazon S3 yang disyaratkan terlampir padanya.
- Anda hanya dapat mengaitkan satu peran IAM dengan instans basis data RDS for Db2 Anda pada setiap saat.
- Instans basis data RDS for Db2 Anda harus dalam keadaan Tersedia.

Anda dapat menambahkan peran IAM ke instans DB Anda dengan menggunakan AWS Management Console atau. AWS CLI

Konsol

Untuk menambahkan peran IAM ke instans basis data RDS for Db2

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data.
3. Pilih nama instans basis data RDS for Db2 Anda.
4. Pada tab Konektivitas dan keamanan, gulir turun ke bagian Kelola peran IAM di bagian bawah halaman.
5. Untuk Tambahkan peran IAM ke instans ini, pilih peran yang Anda buat di Langkah 2: Buat peran IAM dan lampirkan kebijakan IAM Anda.
6. Untuk Fitur, pilih S3_INTEGRATION.
7. Pilih Tambahkan peran.



AWS CLI

Untuk menambahkan peran IAM ke instans basis data RDS for Db2 Anda, jalankan perintah add-role-to-db-instance. Dalam contoh berikut, ganti *db_instance_name* dan *iam_role_arn* dengan nama instans basis data Anda dan ARN peran IAM yang Anda buat di Langkah 2: Buat peran IAM dan lampirkan kebijakan IAM Anda.

Untuk Linux, macOS, atau Unix:

```
aws rds add-role-to-db-instance \
    --db-instance-identifier db_instance_name \
    --feature-name S3_INTEGRATION \
    --role-arn iam_role_arn \
```

Untuk Windows:

```
aws rds add-role-to-db-instance ^
    --db-instance-identifier db_instance_name ^
    --feature-name S3_INTEGRATION ^
    --role-arn iam_role_arn ^
```

Untuk memastikan bahwa peran berhasil ditambahkan ke instans basis data RDS for Db2 Anda, jalankan perintah describe-db-instances. Dalam contoh berikut, ganti *db_instance_name* dengan nama instans basis data Anda.

Untuk Linux, macOS, atau Unix:

```
aws rds describe-db-instances \
    --filters "Name=db-instance-id,Values=db_instance_name" \
    --query 'DBInstances[].AssociatedRoles'
```

Untuk Windows:

```
aws rds describe-db-instances ^
    --filters "Name=db-instance-id,Values=db_instance_name" ^
    --query 'DBInstances[].AssociatedRoles'
```

Perintah ini menghasilkan output yang serupa dengan contoh berikut:

```
[
    [
        {
            "RoleArn": "arn:aws:iam::0123456789012:role/rds-db2-s3-role",
            "FeatureName": "S3_INTEGRATION",
            "Status": "ACTIVE"
        }
    ]
]
```

Memigrasikan data ke Db2 di Amazon RDS

Anda dapat memigrasikan basis data Db2 kelolaan sendiri ke RDS for Db2 dengan menggunakan AWS atau alat Db2 asli.

Topik

Pendekatan migrasi yang menggunakan AWS

Anda dapat melakukan migrasi satu kali database Db2 dariLinux,AIX, atau lingkungan Windows ke Amazon RDS untuk Db2. Untuk meminimalkan waktu henti, Anda dapat melakukan migrasi waktu henti nyaris nol. Anda juga dapat melakukan migrasi sinkron melalui replikasi atau penggunaan. AWS Database Migration Service

Untuk migrasi satu kali untuk database Linux Db2 berbasis, Amazon RDS hanya mendukung backup offline dan online. Amazon RDS tidak mendukung incremental dan Delta backup. Untuk migrasi mendekati nol untuk basis data Db2 Linux berbasis, Amazon RDS memerlukan cadangan online. Kami menyarankan Anda menggunakan backup online untuk migrasi downtime mendekati nol dan backup offline untuk migrasi yang dapat menangani downtime.

Topik

Migrasi satu kali dari lingkungan Linux ke Linux

Dengan pendekatan migrasi ini, Anda mencadangkan basis data Db2 kelolaan sendiri ke bucket Amazon S3. Kemudian, Anda menggunakan prosedur tersimpan Amazon RDS untuk memulihkan database Db2 Anda ke Amazon RDS untuk instans Db2 DB. Lihat informasi yang lebih lengkap tentang penggunaan Amazon S3 di Mengintegrasikan instans basis data RDS for Db2 dengan Amazon S3 .

Topik

Keterbatasan dan rekomendasi untuk menggunakan pemulihan asli

Keterbatasan dan rekomendasi berikut berlaku untuk penggunaan pemulihan asli:

- Amazon RDS hanya mendukung backup offline dan online untuk native restore. Amazon RDS tidak mendukung inkremental atau Delta backup.
- Anda tidak dapat memulihkan dari bucket Amazon S3 di Wilayah AWS yang berbeda dari Wilayah tempat instans RDS untuk Db2 DB Anda berada.
- Anda tidak dapat memulihkan basis data jika instans basis data RDS for Db2 sudah berisi basis data.
- Amazon S3 membatasi ukuran file yang diunggah ke bucket Amazon S3 hingga 5 TB. Jika file cadangan basis data Anda melebihi 5 TB, maka bagi file itu menjadi beberapa file yang lebih kecil.
- Amazon RDS tidak mendukung rutinitas eksternal yang tidak berpagar, pemulihan inkremental, atau pemulihan Delta.
- Anda tidak dapat memulihkan dari sebuah basis data sumber terenkripsi, tetapi dapat memulihkan ke sebuah instans basis data Amazon RDS terenkripsi.

Saat Anda memulihkan basis data Anda, cadangan akan disalin dan diekstrak pada instans basis data RDS for Db2 Anda. Sebaiknya sediakan ruang penyimpanan untuk instans basis data RDS for Db2 Anda dengan ukuran yang sama dengan atau lebih besar daripada jumlah ukuran cadangan ditambah ukuran basis data asli pada disk.

Ukuran maksimal basis data yang dipulihkan adalah ukuran basis data maksimal yang didukung dikurangi ukuran cadangan. Misalnya, jika ukuran basis data maksimal yang didukung adalah 64 TiB dan ukuran cadangan adalah 30 TiB, maka ukuran maksimal basis data yang dipulihkan adalah 34 TiB.

```
64 TiB - 30 TiB = 34 TiB
```

Menyiapkan pencadangan dan pemulihan asli

Untuk pencadangan dan pemulihan asli, Anda memerlukan AWS komponen berikut:

- Bucket Amazon S3 untuk menyimpan file cadangan Anda: Unggah file cadangan apa pun yang ingin Anda migrasikan ke Amazon RDS. Kami menyarankan Anda menggunakan backup offline untuk migrasi yang dapat menangani downtime. Jika sudah memiliki bucket S3, Anda dapat menggunakannya. Jika Anda tidak memiliki bucket S3, lihat Membuat bucket dalam Panduan Pengguna Amazon S3.

  > **Note**
  >
  > Jika database Anda besar dan akan membutuhkan waktu lama untuk mentransfer ke ember S3, Anda dapat memesan AWS Snow Family perangkat dan meminta AWS untuk melakukan pencadangan. Setelah Anda menyalin file ke perangkat dan mengembalikannya ke tim Snow Family, tim akan mentransfer citra cadangan Anda ke bucket S3 Anda. Lihat informasi yang lebih lengkap dalam dokumentasi AWS Snow Family.

- Peran IAM untuk mengakses bucket S3: Jika Anda sudah memiliki peran IAM, Anda dapat menggunakan peran tersebut. Jika Anda tidak memiliki peran, lihat Langkah 2: Buat peran IAM dan lampirkan kebijakan IAM Anda.
- Kebijakan IAM dengan hubungan kepercayaan dan izin yang dilampirkan pada peran IAM Anda: Untuk informasi selengkapnya, lihat. Langkah 1: Buat kebijakan IAM
- Peran IAM ditambahkan ke RDS Anda untuk instans Db2 DB: Untuk informasi selengkapnya, lihat. Langkah 3: Tambahkan peran IAM ke instans basis data RDS for Db2

Memulihkan basis data Db2 Anda

Setelah menyiapkan pencadangan dan pemulihan asli, Anda siap untuk memulihkan basis data Db2 ke instans basis data RDS for Db2.

Untuk memulihkan basis data Db2 ke instans basis data RDS for Db2

1. Hubungi instans basis data RDS for Db2 Anda. Untuk informasi selengkapnya, lihat Menghubungkan ke instans DB RDS untuk Db2 Anda.
2. Pulihkan basis data Anda dengan memanggil `rdsadmin.restore_database`. Lihat informasi yang lebih lengkap di rdsadmin.restore_database.

Migrasi waktu henti nyaris nol untuk basis data Db2 berbasis Linux

Dengan pendekatan migrasi ini, Anda memigrasikan basis data Db2 berbasis Linux dari satu basis data Db2 kelolaan sendiri (sumber) ke Amazon RDS for Db2. Pendekatan ini menghasilkan pemadaman atau waktu henti minimal atau nol bagi aplikasi atau pengguna. Pendekatan ini mencadangkan basis data Anda dan memulihkannya dengan pemutaran ulang log, yang membantu mencegah gangguan pada operasi yang sedang berjalan dan menyediakan ketersediaan tinggi basis data Anda.

Untuk mencapai migrasi waktu henti nyaris nol, RDS for Db2 mengimplementasikan pemulihan dengan pemutaran ulang log. Pendekatan ini mengambil cadangan basis data Db2 berbasis Linux kelolaan sendiri Anda dan memulihkannya di server RDS for Db2. Dengan prosedur-prosedur tersimpan Amazon RDS, Anda lalu menerapkan log transaksi selanjutnya untuk memutakhirkan basis data.

Topik

Keterbatasan dan rekomendasi migrasi waktu henti nyaris nol

Keterbatasan berikut berlaku pada penggunaan migrasi waktu henti nyaris nol:

- Amazon RDS memerlukan cadangan online untuk migrasi downtime mendekati nol. Ini karena Amazon RDS menyimpan database Anda dalam status tertunda rollforward saat Anda mengunggah log transaksi yang diarsipkan. Untuk informasi selengkapnya, lihat the section called “Memigrasikan basis data Db2 Anda”.
- Anda tidak dapat memulihkan dari bucket Amazon S3 di Wilayah AWS yang berbeda dari Wilayah tempat instans RDS untuk Db2 DB Anda berada.
- Anda tidak dapat memulihkan basis data jika instans basis data RDS for Db2 sudah berisi basis data.
- Amazon S3 membatasi ukuran file yang diunggah ke bucket S3 pada 5 TB. Jika file cadangan basis data Anda melebihi 5 TB, maka bagi file itu menjadi beberapa file yang lebih kecil.
- Amazon RDS tidak mendukung rutinitas eksternal yang tidak berpagar, pemulihan inkremental, atau pemulihan Delta.

- Anda tidak dapat memulihkan dari sebuah basis data sumber terenkripsi, tetapi dapat memulihkan ke sebuah instans basis data Amazon RDS terenkripsi.

Ketika Anda memulihkan database Anda, Amazon RDS menyalin cadangan Anda dan kemudian mengekstraknya pada RDS Anda untuk instans Db2 DB. Sebaiknya sediakan ruang penyimpanan untuk instans basis data RDS for Db2 Anda dengan ukuran yang sama dengan atau lebih besar daripada jumlah ukuran cadangan ditambah ukuran basis data asli pada disk.

Ukuran maksimal basis data yang dipulihkan adalah ukuran basis data maksimal yang didukung dikurangi ukuran cadangan. Misalnya, jika ukuran basis data maksimal yang didukung adalah 64 TiB dan ukuran cadangan adalah 30 TiB, maka ukuran maksimal basis data yang dipulihkan adalah 34 TiB.

```
64 TiB - 30 TiB = 34 TiB
```

Menyiapkan migrasi waktu henti nyaris nol

Untuk migrasi downtime mendekati nol, Anda memerlukan komponen berikut: AWS

- Bucket Amazon S3 untuk menyimpan file cadangan Anda: Unggah file cadangan apa pun yang ingin Anda migrasikan ke Amazon RDS. Amazon RDS memerlukan cadangan online untuk migrasi downtime mendekati nol. Jika sudah memiliki bucket S3, Anda dapat menggunakannya. Jika Anda tidak memiliki bucket S3, lihat Membuat bucket dalam Panduan Pengguna Amazon S3.

  > **Note**
  >
  > Jika database Anda besar dan akan membutuhkan waktu lama untuk mentransfer ke ember S3, Anda dapat memesan AWS Snow Family perangkat dan meminta AWS untuk melakukan pencadangan. Setelah Anda menyalin file ke perangkat dan mengembalikannya ke tim Snow Family, tim akan mentransfer citra cadangan Anda ke bucket S3 Anda. Lihat informasi yang lebih lengkap dalam dokumentasi AWS Snow Family.

- Peran IAM untuk mengakses bucket S3: Jika Anda sudah memiliki peran AWS Identity and Access Management (IAM), Anda dapat menggunakan peran tersebut. Jika Anda tidak memiliki peran, lihat Langkah 2: Buat peran IAM dan lampirkan kebijakan IAM Anda.
- Kebijakan IAM dengan hubungan kepercayaan dan izin yang dilampirkan pada peran IAM Anda: Untuk informasi selengkapnya, lihat. Langkah 1: Buat kebijakan IAM

- Peran IAM ditambahkan ke RDS Anda untuk instans Db2 DB: Untuk informasi selengkapnya, lihat. Langkah 3: Tambahkan peran IAM ke instans basis data RDS for Db2

Memigrasikan basis data Db2 Anda

Setelah menyiapkan migrasi waktu henti nyaris nol, Anda siap untuk memigrasikan basis data Db2 ke instans basis data RDS for Db2.

Untuk melakukan migrasi waktu henti nyaris nol

1. Lakukan pencadangan daring basis data sumber Anda. Lihat informasi yang lebih lengkap di perintah BACKUP DATABASE dalam dokumentasi IBM Db2.
2. Salin cadangan basis data ke sebuah bucket Amazon S3. Lihat informasi tentang cara menggunakan Amazon S3 dalam Panduan Pengguna Amazon Simple Storage Service.
3. Hubungi server `rdsadmin` dengan *`master_username`* dan *`master_password`* untuk instans basis data RDS for Db2 Anda.

   ```
   db2 connect to rdsadmin user master_username using master_password
   ```

4. Pulihkan cadangan pada server RDS for Db2 dengan memanggil `rdsadmin.restore_database`. Atur `backup_type` ke `ONLINE`. Untuk informasi selengkapnya, lihat rdsadmin.restore_database.
5. Salin log arsip Anda dari server sumber ke bucket S3. Lihat informasi yang lebih lengkap di Pengelogan arsip dalam dokumentasi IBM Db2.
6. Terapkan log arsip sebanyak yang diperlukan dengan memanggil `rdsadmin.rollforward_database`. Atur `complete_rollforward` ke `FALSE` untuk menjaga basis data dalam keadaan `ROLL-FORWARD PENDING`. Untuk informasi selengkapnya, lihat rdsadmin.rollforward_database.
7. Setelah Anda menerapkan semua log arsip, bawa daring basis data dengan memanggil `rdsadmin.complete_rollforward`. Untuk informasi selengkapnya, lihat rdsadmin.complete_rollforward.
8. Alihkan koneksi aplikasi ke server RDS for Db2 dengan memperbarui titik akhir aplikasi Anda untuk basis data atau dengan memperbarui titik akhir DNS agar mengarahkan lalu lintas ke server RDS for Db2. Anda juga dapat menggunakan fitur perutean ulang klien otomatis Db2 pada basis data Db2 kelolaan sendiri Anda dengan titik akhir basis data RDS for Db2. Lihat informasi yang lebih lengkap di Deskripsi dan penyiapan perutean ulang klien otomatis dalam dokumentasi IBM Db2.

9. (Opsional) Matikan basis data sumber Anda.

Migrasi satu kali dari lingkungan AIX atau Windows ke Linux

Dengan pendekatan migrasi ini, Anda menggunakan alat-alat Db2 asli untuk mencadangkan basis data Db2 kelolaan sendiri ke bucket Amazon S3. Alat-alat Db2 asli meliputi utilitas `export`, perintah sistem `db2move`, atau perintah sistem `db2look`. Basis data Db2 Anda dapat kelolaan sendiri atau di Amazon Elastic Compute Cloud (Amazon EC2). Anda dapat memindahkan data dari sistem AIX atau Windows Anda ke bucket Amazon S3. Lalu, menggunakan klien Db2 untuk memuatkan data secara langsung dari bucket S3 ke basis data RDS for Db2 Anda. Waktu henti bergantung pada ukuran basis data Anda. Lihat informasi yang lebih lengkap tentang penggunaan Amazon S3 di Mengintegrasikan instans basis data RDS for Db2 dengan Amazon S3 .

Untuk memigrasikan basis data Db2 ke RDS for Db2

1. Buat persiapan untuk mencadangkan basis data Anda. Konfigurasikan jumlah penyimpanan yang cukup untuk menampung cadangan pada sistem Db2 kelolaan sendiri.
2. Cadangkan basis data Anda.
   a. Jalankan perintah sistem db2look untuk mengekstrak file bahasa definisi data (DDL) untuk semua objek.
   b. Jalankan utilitas ekspor Db2, perintah sistem db2move, atau pernyataan CREATE EXTERNAL TABLE untuk menurunkan data tabel Db2 ke penyimpanan di sistem Db2 Anda.
3. Pindahkan cadangan Anda ke bucket Amazon S3. Untuk informasi selengkapnya, lihat Mengintegrasikan instans basis data RDS for Db2 dengan Amazon S3 .

   > **Note**
   >
   > Jika database Anda besar dan akan membutuhkan waktu lama untuk mentransfer ke ember S3, Anda dapat memesan AWS Snow Family perangkat dan meminta AWS untuk melakukan pencadangan. Setelah Anda menyalin file ke perangkat dan mengembalikannya ke tim Snow Family, tim akan mentransfer citra cadangan Anda ke bucket S3 Anda. Lihat informasi yang lebih lengkap dalam dokumentasi AWS Snow Family.

4. Gunakan klien Db2 untuk memuatkan data secara langsung dari bucket S3 ke basis data RDS for Db2 Anda.

Migrasi sinkron dari ke lingkungan LinuxLinux

Dengan pendekatan migrasi ini, Anda menyiapkan replikasi antara basis data Db2 kelolaan sendiri dan instans basis data RDS for Db2. Perubahan yang dibuat pada basis data kelolaan sendiri bereplikasi ke instans basis data RDS for Db2 dalam waktu nyaris nyata. Pendekatan ini dapat memberikan ketersediaan yang sinambung dan meminimalkan waktu henti selama proses migrasi.

Menggunakan AWS Database Migration Service (AWS DMS)

Anda dapat menggunakan AWS DMS untuk migrasi satu kali dan kemudian menyinkronkan dari Db2 di Linux, Unix, dan Windows ke Amazon RDS untuk Db2. Untuk informasi lebih lanjut, lihat Apa itu AWS Database Migration Service? .

Alat Db2 asli

Anda dapat menggunakan beberapa alat, utilitas, dan perintah Db2 asli untuk memindahkan data dari basis data Db2 ke basis data Amazon RDS for Db2. Untuk menggunakan semua alat Db2 asli ini, Anda harus dapat menghubungkan mesin klien Anda dengan instans basis data RDS for Db2. Untuk informasi selengkapnya, lihat Menghubungkan mesin klien dengan instans basis data RDS for Db2.

Nama alat	Kasus penggunaan	Keterbatasan
db2look	Menyalin metadata dari basis data Db2 kelolaan sendiri ke basis data RDS for Db2.	• Anda harus mengubah sintaks untuk membuat kolam penyangga, membuat ruang tabel, dan membuat peran agar sesuai dengan sintaks yang digunakan oleh Prosedur tersimpan RDS for Db2.
Perintah IMPORT	Memigrasikan tabel kecil dan tabel dengan objek besar (LOB) dari mesin klien ke instans basis data RDS for Db2.	• Lebih lambat daripada utilitas `LOAD` karena operasi pengelogan `INSERT` dan `DELETE`. • Kinerja buruk dengan bandwidth jaringan terbatas.

Nama alat	Kasus penggunaan	Keterbatasan
Utilitas INGEST	Mengalirkan data terus-menerus dari file dan pipa tanpa objek besar (LOB) pada mesin klien ke instans basis data RDS for Db2. Mendukung operasi-operasi `INSERT` dan `MERGE`.	• Tidak dapat mengalirkan file data yang berisi LOB. Gunakan perintah `IMPORT` sebagai gantinya. • Konektivitas diperlukan antara basis data Db2 kelolaan sendiri dan basis data RDS for Db2.
Perintah INSERT	Menyalin data dalam tabel kecil dari basis data Db2 kelolaan sendiri ke basis data RDS for Db2.	• Konektivitas diperlukan antara basis data Db2 kelolaan sendiri dan basis data RDS for Db2. • Kinerja buruk dengan bandwidth jaringan terbatas.
Perintah LOAD	Memigrasikan tabel kecil tanpa objek besar (LOB) dari mesin klien ke instans basis data RDS for Db2.	• Tidak dapat memigrasikan file data yang berisi LOB. Gunakan perintah `IMPORT` sebagai gantinya. • Kinerja buruk dengan bandwidth jaringan terbatas.

Menghubungkan mesin klien dengan instans basis data RDS for Db2

Untuk menggunakan segala alat Db2 asli untuk memindahkan data dari basis data Db2 ke basis data Amazon RDS for Db2, Anda harus menghubungkan dahulu mesin klien Anda dengan instans basis data RDS for Db2.

Mesin klien dapat berupa:

- Instans Amazon Elastic Compute Cloud (Amazon EC2) pada Linux, Windows, atau macOS. Instans ini semestinya berada dalam cloud privat virtual (VPC) yang sama dengan instans basis data RDS for Db2 Anda, AWS Cloud9 atau AWS CloudShell.

- Instans Db2 kelolaan sendiri di instans Amazon EC2. Instans semestinya berada dalam VPC yang sama.
- Instans Db2 kelolaan sendiri di instans Amazon EC2. Instans boleh berada di VPC yang berbeda jika Anda mengaktifkan perekanan/peering VPC. Lihat informasi yang lebih lengkap di Membuat koneksi peering VPC dalam Panduan Peering VPC Amazon Virtual Private Cloud.
- Mesin lokal yang menjalankan Linux, Windows, atau macOS dalam lingkungan kelolaan sendiri. Anda harus memiliki konektivitas publik dengan RDS for Db2 atau mengaktifkan konektivitas VPN antara instans Db2 kelolaan sendiri dan AWS.

Untuk menghubungkan mesin klien dengan instans basis data RDS for Db2, masuk ke mesin klien Anda dengan IBM Db2 Data Management Console. Lihat informasi yang lebih lengkap di Membuat instans DB Amazon RDS dan IBM Db2 Data Management Console.

Anda dapat menggunakan AWS Database Migration Service (AWS DMS) untuk menjalankan kueri terhadap basis data, menjalankan rencana eksekusi SQL, dan memantau basis data. Lihat informasi yang lebih lengkap di Apakah Layanan Migrasi Basis Data AWS? dalam Panduan Pengguna AWS Database Migration Service.

Setelah berhasil menghubungkan mesin klien dengan instans basis data RDS for Db2, Anda siap untuk menggunakan semua alat Db2 asli untuk menyalin data. Untuk informasi selengkapnya, lihat Alat Db2 asli.

Alat db2look

`db2look` adalah alat Db2 asli yang mengekstrak file bahasa definisi data (DDL), objek, otorisasi, konfigurasi, WLM, dan tata letak basis data. Anda dapat menggunakan `db2look` untuk menyalin metadata dari basis data dari basis data Db2 kelolaan sendiri ke basis data RDS for Db2. Lihat informasi yang lebih lengkap di Meniru basis data dengan menggunakan db2look dalam dokumentasi IBM Db2.

Untuk menyalin metadata basis data

1. Jalankan alat `db2look` pada sistem Db2 kelolaan sendiri untuk mengekstrak file DDL. Dalam contoh berikut, ganti *`database_name`* dengan nama basis data Db2 Anda.

   ```
   db2look -d database_name -e -l -a -f -wlm -cor -createdb -printdbcfg -o db2look.sql
   ```

2. Jika mesin klien Anda memiliki akses ke basis data sumber (Db2 kelolaan sendiri) dan instans basis data RDS for Db2, Anda dapat membuat file `db2look.sql` di mesin klien dengan

melampirkan secara langsung pada instans jarak jauh. Lalu, buat katalog instans Db2 kelolaan sendiri jauh.

a. Buat katalog simpul. Dalam contoh berikut, ganti *dns_ip_address* dan *porta* dengan nama DNS atau alamat IP dan nomor porta basis data Db2 kelolaan sendiri.

```
db2 catalog tcpip node srcnode REMOTE dns_ip_address server port
```

b. Buat katalog basis data. Dalam contoh berikut, ganti *source_database_name* dan *source_database_alias* dengan nama basis data Db2 kelolaan sendiri dan alias yang ingin Anda gunakan untuk basis data ini.

```
db2 catalog database source_database_name as source_database_alias at node
 srcnode \
    authentication server_encrypt
```

c. Lampirkan ke basis data sumber. Dalam contoh berikut, ganti *source_database_alias*, *user_id*, dan *user_password* dengan alias yang Anda buat pada langkah di atas dan ID pengguna dan kata sandi untuk basis data Db2 kelolaan sendiri.

```
db2look -d source_database_alias -i user_id -w user_password -e -l -a -f -wlm \
    -cor -createdb -printdbcfg -o db2look.sql
```

3. Jika Anda tidak dapat mengakses basis data Db2 kelolaan sendiri jauh dari mesin klien, salin file `db2look.sql` ke mesin klien. Kemudian, buat katalog instans basis data RDS for Db2.

a. Buat katalog simpul. Dalam contoh berikut, ganti *dns_ip_address* dan *porta* dengan nama DNS atau alamat IP dan nomor porta instans basis data RDS for Db2.

```
db2 catalog tcpip node remnode REMOTE dns_ip_address server port
```

b. Buat katalog basis data. Dalam contoh berikut, ganti *rds_database_name* dan *rds_database_alias* dengan nama basis data RDS for Db2 dan alias yang ingin Anda gunakan untuk basis data ini.

```
db2 catalog database rds_database_name as rds_database_alias at node remnode \
    authentication server_encrypt
```

c. Buat katalog basis data admin yang mengelola RDS for Db2. Anda tidak dapat menggunakan basis data ini untuk menyimpan data apa pun.

```
db2 catalog database rdsadmin as rdsadmin at node remnode authentication
 server_encrypt
```

4. Buat kolam penyangga dan ruang tabel. Administrator tidak memiliki privilese untuk membuat kolam penyangga atau ruang tabel. Namun, Anda dapat menggunakan prosedur tersimpan Amazon RDS untuk membuat keduanya.

   a. Temukan nama dan definisi kolam penyangga dan ruang tabel dalam file `db2look.sql`.

   b. Hubungi Amazon RDS dengan menggunakan nama pengguna master dan kata sandi master untuk instans basis data RDS for Db2 Anda. Dalam contoh berikut, ganti *`master_username`* dan *`master_password`* dengan informasi Anda sendiri.

   ```
   db2 connect to rdsadmin user master_username using master_password
   ```

   c. Buat kolam penyangga dengan memanggil `rdsadmin.create_bufferpool`. Untuk informasi selengkapnya, lihat rdsadmin.create_bufferpool.

   ```
   db2 "call rdsadmin.create_bufferpool(
       'database_name',
       'buffer_pool_name',
       buffer_pool_size,
       'immediate',
       'automatic',
       page_size,
       number_block_pages,
       block_size)"
   ```

   d. Buat ruang tabel dengan memanggil `rdsadmin.create_tablespace`. Untuk informasi selengkapnya, lihat rdsadmin.create_tablespace.

   ```
   db2 "call rdsadmin.create_tablespace(
       'database_name',
       'tablespace_name',
       'buffer_pool_name',
       tablespace_initial_size,
       tablespace_increase_size,
       'tablespace_type')"
   ```

   e. Ulangi langkah c atau d untuk setiap kolam penyangga atau ruang tabel yang ingin Anda tambahkan.

f. Akhiri koneksi Anda.

```
db2 terminate
```

5. Buat tabel dan objek.

a. Hubungi basis data RDS for Db2 Anda dengan menggunakan nama pengguna master dan kata sandi master untuk instans basis data RDS for Db2 Anda. Dalam contoh berikut, ganti *rds_database_name*, *master_username*, dan *master_password* dengan informasi Anda sendiri.

```
db2 connect to rds_database_name user master_username using master_password
```

b. Jalankan file `db2look.sql`.

```
db2 -tvf db2look.sql
```

c. Akhiri koneksi Anda.

```
db2 terminate
```

Perintah IMPORT dengan mesin klien

Anda dapat menggunakan perintah `IMPORT` dari mesin klien untuk mengimpor data Anda ke server Amazon RDS for Db2.

> **Important**
>
> Metode perintah `IMPORT` berguna untuk memigrasikan tabel kecil dan tabel yang menyertakan objek besar (LOB). Perintah `IMPORT` lebih lambat daripada utilitas `LOAD` karena operasi-operasi pengelogan `INSERT` dan `DELETE`. Jika bandwidth jaringan Anda antara mesin klien dan RDS for Db2 terbatas, sebaiknya gunakan pendekatan migrasi yang berbeda. Untuk informasi selengkapnya, lihat Alat Db2 asli.

Untuk mengimpor data ke server RDS for Db2

1. Masuk ke mesin klien Anda dengan IBM Db2 Data Management Console. Untuk informasi selengkapnya, lihat Menghubungi instans basis data RDS for Db2 dengan IBM Db2 Data Management Console.
2. Buat katalog basis data RDS for Db2 pada mesin klien.

   a. Buat katalog simpul. Dalam contoh berikut, ganti *dns_ip_address* dan *porta* dengan nama DNS atau alamat IP dan nomor porta basis data Db2 kelolaan sendiri.

   ```
   db2 catalog tcpip node srcnode REMOTE dns_ip_address server port
   ```

   b. Buat katalog basis data. Dalam contoh berikut, ganti *source_database_name* dan *source_database_alias* dengan nama basis data Db2 kelolaan sendiri dan alias yang ingin Anda gunakan untuk basis data ini.

   ```
   db2 catalog database source_database_name as source_database_alias at node
    srcnode \
       authentication server_encrypt
   ```

3. Lampirkan ke basis data sumber. Dalam contoh berikut, ganti *source_database_alias*, *user_id*, dan *user_password* dengan alias yang Anda buat pada langkah di atas dan ID pengguna dan kata sandi untuk basis data Db2 kelolaan sendiri.

   ```
   db2look -d source_database_alias -i user_id -w user_password -e -l -a -f -wlm \
       -cor -createdb -printdbcfg -o db2look.sql
   ```

4. Hasilkan file data dengan menggunakan perintah `EXPORT` pada sistem Db2 kelolaan sendiri Anda. Dalam contoh berikut, ganti *directory* dengan direktori pada mesin klien tempat file data Anda berada. Ganti *file_name* dan *table_name* dengan nama file data dan nama tabel.

   ```
   db2 "export to /directory/file_name.txt of del lobs to /directory/lobs/ \
       modified by coldel\| select * from table_name"
   ```

5. Hubungi basis data RDS for Db2 Anda dengan menggunakan nama pengguna master dan kata sandi master untuk instans basis data RDS for Db2 Anda. Dalam contoh berikut, ganti *rds_database_alias*, *master_username*, dan *master_password* dengan informasi Anda sendiri.

```
db2 connect to rds_database_alias user master_username using master_password
```

6. Gunakan perintah `IMPORT` untuk mengimpor data dari file pada mesin klien ke dalam basis data RDS for Db2 jauh. Lihat informasi yang lebih lengkap di perintah IMPORT dalam dokumentasi IBM Db2. Dalam contoh berikut, ganti *`directory`* dan *`file_name`* dengan direktori pada mesin klien tempat file data Anda berada dan nama file data itu. Ganti *`SCHEMA_NAME`* dan *`TABLE_NAME`* dengan nama skema dan tabel Anda.

```
db2 "IMPORT from /directory/file_name.tbl OF DEL LOBS FROM /directory/lobs/ \
    modified by coldel\| replace into SCHEMA_NAME.TABLE_NAME"
```

7. Akhiri koneksi Anda.

```
db2 terminate
```

Utilitas INGEST

Anda dapat menggunakan utilitas `INGEST` untuk mengalirkan dengan kontinu data dari file dan pipa pada mesin klien ke instans basis data Amazon RDS for Db2 target. Utilitas `INGEST` mendukung operasi-operasi `INSERT` dan `MERGE`. Lihat informasi yang lebih lengkap di utilitas Ingest dalam dokumentasi IBM Db2.

Karena utilitas `INGEST` mendukung nama panggilan, Anda dapat menggunakannya untuk mentransfer data dari basis data Db2 kelolaan sendiri ke basis data RDS for Db2. Pendekatan ini berfungsi selama ada konektivitas jaringan di antara kedua basis data.

> **Important**
>
> Utilitas `INGEST` tidak mendukung objek besar (LOB). Gunakan perintah `IMPORT` sebagai gantinya.

Untuk menggunakan fitur `RESTARTABLE` dalam utilitas `INGEST`, jalankan perintah berikut pada basis data RDS for Db2.

```
db2 "call sysproc.sysinstallobjects('INGEST','C',NULL,NULL)"
```

Perintah INSERT dari basis data Db2 kelolaan sendiri ke basis data Amazon RDS for Db2

Anda dapat menggunakan perintah `INSERT` dari server Db2 kelolaan sendiri untuk memasukkan data Anda ke dalam basis data RDS for Db2. Dengan pendekatan migrasi ini, Anda menggunakan nama panggilan untuk instans basis data RDS for Db2 jauh. Basis data Db2 kelolaan sendiri Anda (sumber) harus dapat menghubungi basis data RDS for Db2 (target).

> **Important**
>
> Metode perintah `INSERT` berguna untuk memigrasikan tabel kecil. Jika bandwidth jaringan antara basis data Db2 kelolaan sendiri Anda dan basis data RDS for Db2 terbatas, sebaiknya gunakan pendekatan migrasi yang berbeda. Untuk informasi selengkapnya, lihat Alat Db2 asli.

Untuk menyalin data dari basis data Db2 kelolaan sendiri ke basis data RDS for Db2

1. Buat katalog instans basis data RDS for Db2 pada instans Db2 kelolaan sendiri.

   a. Buat katalog simpul. Dalam contoh berikut, ganti *dns_ip_address* dan *porta* dengan nama DNS atau alamat IP dan nomor porta basis data Db2 kelolaan sendiri.

   ```
   db2 catalog tcpip node remnode REMOTE dns_ip_address SERVER port
   ```

   b. Buat katalog basis data. Dalam contoh berikut, ganti *rds_database_name* dengan nama basis data pada instans basis data RDS for Db2 Anda.

   ```
   db2 catalog database rds_database_name as remdb at node remnode \
       authentication server_encrypt
   ```

2. Aktifkan federasi pada instans Db2 kelolaan sendiri. Dalam contoh berikut, ganti *source_database_name* dengan nama basis data Anda pada instans Db2 kelolaan sendiri.

   ```
   db2 update dbm cfg using FEDERATED YES source_database_name
   ```

3. Buat tabel pada instans basis data RDS for Db2.

   a. Buat katalog simpul. Dalam contoh berikut, ganti *dns_ip_address* dan *porta* dengan nama DNS atau alamat IP dan nomor porta basis data Db2 kelolaan sendiri.

```
db2 catalog tcpip node srcnode REMOTE dns_ip_address server port
```

b. Buat katalog basis data. Dalam contoh berikut, ganti *source_database_name* dan *source_database_alias* dengan nama basis data Db2 kelolaan sendiri dan alias yang ingin Anda gunakan untuk basis data ini.

```
db2 catalog database source_database_name as source_database_alias at node
 srcnode \
    authentication server_encrypt
```

4. Lampirkan ke basis data sumber. Dalam contoh berikut, ganti *source_database_alias*, *user_id*, dan *user_password* dengan alias yang Anda buat pada langkah di atas dan ID pengguna dan kata sandi untuk basis data Db2 kelolaan sendiri.

```
db2look -d source_database_alias -i user_id -w user_password -e -l -a -f -wlm \
    -cor -createdb -printdbcfg -o db2look.sql
```

5. Siapkan federasi, dan buat nama panggilan untuk tabel basis data RDS for Db2 pada instans Db2 kelolaan sendiri.

a. Hubungi basis data lokal Anda. Dalam contoh berikut, ganti *source_database_name* dengan nama basis data pada instans Db2 kelolaan sendiri Anda.

```
db2 connect to source_database_name
```

b. Buat pembungkus untuk mengakses sumber data Db2.

```
db2 create wrapper drda
```

c. Tentukan sumber data pada basis data federasi. Dalam contoh berikut, ganti *admin* dan *admin_password* dengan kredensial Anda untuk instans Db2 kelolaan sendiri Anda. Ganti *rds_database_name* dengan nama basis data pada instans basis data RDS for Db2 Anda.

```
db2 "create server rdsdb2 type DB2/LUW version '11.5.9.0' \
    wrapper drda authorization "admin" password "admin_password" \
    options( dbname 'rds_database_name', node 'remnode')"
```

d. Petakan pengguna pada kedua basis data. Dalam contoh berikut, ganti *master_username* dan *master_password* dengan kredensial Anda untuk instans basis data RDS for Db2 Anda.

```
db2 "create user mapping for user server rdsdb2 \
    options (REMOTE_AUTHID 'master_username', REMOTE_PASSWORD
 'master_password')"
```

e. Periksa koneksi dengan server RDS for Db2.

```
db2 set passthru rdsdb2
```

f. Buat nama panggilan untuk tabel di basis data RDS for Db2 jauh. Dalam contoh berikut, ganti *NICKNAME* dan *TABLE_NAME* dengan nama panggilan untuk tabel dan nama tabel.

```
db2 create nickname REMOTE.NICKNAME for RDSDB2.TABLE_NAME.NICKNAME
```

6. Masukkan data ke dalam tabel di basis data RDS for Db2 jauh. Gunakan nama panggilan dalam pernyataan `select` di tabel lokal dalam instans Db2 kelolaan sendiri. Dalam contoh berikut, ganti *NICKNAME* dan *TABLE_NAME* dengan nama panggilan untuk tabel dan nama tabel.

```
db2 "INSERT into REMOTE.NICKNAME select * from RDS2DB2.TABLE_NAME.NICKNAME"
```

Perintah LOAD dengan mesin klien

Anda dapat menggunakan perintah `LOAD CLIENT` untuk memuatkan data dari file ke server RDS for Db2. Karena tidak ada konektivitas SSH ke server Amazon RDS for Db2, Anda dapat menggunakan perintah `LOAD CLIENT` pada server Db2 kelolaan sendiri atau mesin klien Db2 Anda.

Important

Metode perintah `LOAD` berguna untuk memigrasikan tabel kecil. Jika bandwidth jaringan Anda antara klien dan RDS for Db2 terbatas, sebaiknya gunakan pendekatan migrasi yang berbeda. Lihat informasi yang lebih lengkap di Alat Db2 asli.
Jika file data Anda menyertakan rujukan ke nama file objek besar (LOB), maka perintah `LOAD` tidak akan berfungsi karena objek besar harus berada di server Db2. Jika Anda mencoba

memuatkan LOB dari mesin klien ke server RDS for Db2, Anda akan menerima kesalahan SQL3025N. Gunakan perintah IMPORT sebagai gantinya.

Untuk memuatkan data ke server RDS for Db2

1. Masuk ke mesin klien Anda dengan IBM Db2 Data Management Console. Untuk informasi selengkapnya, lihat Menghubungi instans basis data RDS for Db2 dengan IBM Db2 Data Management Console.
2. Buat katalog basis data RDS for Db2 pada mesin klien.

   a. Buat katalog simpul. Dalam contoh berikut, ganti *dns_ip_address* dan *porta* dengan nama DNS atau alamat IP dan nomor porta basis data Db2 kelolaan sendiri.

   ```
   db2 catalog tcpip node srcnode REMOTE dns_ip_address server port
   ```

   b. Buat katalog basis data. Dalam contoh berikut, ganti *source_database_name* dan *source_database_alias* dengan nama basis data Db2 kelolaan sendiri dan alias yang ingin Anda gunakan untuk basis data ini.

   ```
   db2 catalog database source_database_name as source_database_alias at node
    srcnode \
       authentication server_encrypt
   ```

3. Lampirkan ke basis data sumber. Dalam contoh berikut, ganti *source_database_alias*, *user_id*, dan *user_password* dengan alias yang Anda buat pada langkah di atas dan ID pengguna dan kata sandi untuk basis data Db2 kelolaan sendiri.

   ```
   db2look -d source_database_alias -i user_id -w user_password -e -l -a -f -wlm \
       -cor -createdb -printdbcfg -o db2look.sql
   ```

4. Hasilkan file data dengan menggunakan perintah EXPORT pada sistem Db2 kelolaan sendiri Anda. Dalam contoh berikut, ganti *directory* dengan direktori pada mesin klien tempat file data Anda berada. Ganti *file_name* dan *TABLE_NAME* dengan nama file data dan nama tabel.

   ```
   db2 "export to /directory/file_name.txt of del modified by coldel\| \
       select * from TPCH.TABLE_NAME"
   ```

5. Hubungi basis data RDS for Db2 Anda dengan menggunakan nama pengguna master dan kata sandi master untuk instans basis data RDS for Db2 Anda. Dalam contoh berikut, ganti

rds_database_alias, *master_username*, dan *master_password* dengan informasi Anda sendiri.

```
db2 connect to rds_database_alias user master_username using master_password
```

6. Gunakan perintah LOAD untuk memuatkan data dari file pada mesin klien ke basis data RDS for Db2 jauh. Lihat informasi yang lebih lengkap di perintah LOAD dalam dokumentasi IBM Db2. Dalam contoh berikut, ganti *directory* dengan direktori pada mesin klien tempat file data Anda berada. Ganti *file_name* dan *TABLE_NAME* dengan nama file data dan nama tabel.

```
db2 "LOAD CLIENT from /directory/file_name.txt \
    modified by coldel\| replace into TPCH.TABLE_NAME \
    nonrecoverable without prompting"
```

7. Akhiri koneksi Anda.

```
db2 terminate
```

Pilihan untuk RDS untuk instans Db2 DB

Berikut ini menunjukkan opsi, atau fitur tambahan, yang tersedia untuk instans Amazon RDS yang menjalankan mesin Db2 DB. Untuk mengaktifkan opsi ini, Anda dapat menambahkannya ke grup opsi khusus, lalu mengaitkan grup opsi dengan instans DB Anda. Untuk informasi selengkapnya tentang cara menggunakan grup opsi, lihat Menggunakan grup opsi.

Amazon RDS mendukung opsi berikut untuk Db2:

Opsi	ID Opsi
Pencatatan audit Db2	`DB2_AUDIT`

Pencatatan audit Db2

Dengan pencatatan audit Db2, Amazon RDS mencatat aktivitas database, termasuk pengguna yang masuk ke database dan kueri yang dijalankan terhadap database. RDS mengunggah log audit yang telah selesai ke bucket Amazon S3 Anda, menggunakan peran (IAM) AWS Identity and Access Management yang Anda berikan.

Topik

Menyiapkan pencatatan audit Db2

Untuk mengaktifkan audit logging untuk database RDS untuk Db2, Anda mengaktifkan `DB2_AUDIT` opsi pada RDS untuk instans Db2 DB. Kemudian, konfigurasikan kebijakan audit untuk mengaktifkan fitur untuk database tertentu. Untuk mengaktifkan opsi pada RDS untuk instans Db2 DB, Anda mengonfigurasi pengaturan opsi untuk opsi tersebut. `DB2_AUDIT` Anda melakukannya dengan memberikan Amazon Resource Names (ARN) untuk bucket Amazon S3 Anda dan peran IAM dengan izin untuk mengakses bucket Anda.

Untuk mengatur pencatatan audit Db2 untuk database RDS untuk Db2, selesaikan langkah-langkah berikut.

Topik

Langkah 1: Buat bucket Amazon S3.

Jika Anda belum melakukannya, buat bucket Amazon S3 tempat Amazon RDS dapat mengunggah RDS Anda untuk file log audit database Db2. Pembatasan berikut berlaku untuk bucket S3 yang Anda gunakan sebagai target untuk file audit:

- Itu harus sama Wilayah AWS dengan RDS Anda untuk instans Db2 DB.
- Bucket S3 tidak boleh dibuka untuk umum.
- Bucket S3 tidak dapat menggunakan Kunci Objek S3.
- Pemilik bucket juga harus menjadi pemilik peran IAM.

Untuk mempelajari cara membuat bucket Amazon S3, lihat Membuat bucket di Panduan Pengguna Amazon S3.

Setelah mengaktifkan pencatatan audit, Amazon RDS secara otomatis mengirimkan log dari instans DB Anda ke lokasi berikut:

- Log tingkat instans DB - *bucket_name*/db2-audit-logs/*dbi_resource_id*/*date_time_utc*/
- Log tingkat basis data - *bucket_name*/db2-audit-logs/*dbi_resource_id*/*date_time_utc*/*db_name*/

Catat Nama Sumber Daya Amazon (ARN) untuk bucket Anda. Informasi ini diperlukan untuk menyelesaikan langkah-langkah selanjutnya.

Langkah 2: Buat kebijakan IAM

Buat kebijakan IAM dengan izin yang diperlukan untuk mentransfer file log audit dari instans DB ke bucket Amazon S3. Langkah ini mengasumsikan bahwa Anda memiliki bucket S3.

Sebelum Anda membuat kebijakan, kumpulkan informasi berikut:

- ARN untuk ember Anda.
- ARN untuk kunci AWS Key Management Service (AWS KMS) Anda, jika bucket Anda menggunakan SSE-KMS enkripsi.

Buat kebijakan IAM yang mencakup izin-izin berikut:

```
"s3:ListBucket",
 "s3:GetBucketACL",
 "s3:GetBucketLocation",
 "s3:PutObject",
 "s3:ListMultipartUploadParts",
 "s3:AbortMultipartUpload",
 "s3:ListAllMyBuckets"
```

> **Note**
>
> Amazon RDS memerlukan `s3:ListAllMyBuckets` tindakan secara internal untuk memverifikasi bahwa yang sama Akun AWS memiliki bucket S3 dan RDS untuk instans Db2 DB.

Jika bucket Anda menggunakan SSE-KMS enkripsi, sertakan juga izin berikut:

```
"kms:GenerateDataKey",
 "kms:Decrypt"
```

Anda dapat membuat kebijakan IAM dengan menggunakan AWS Management Console atau AWS Command Line Interface (AWS CLI).

Konsol

Untuk membuat kebijakan IAM untuk mengizinkan Amazon RDS mengakses bucket Amazon S3

1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi, pilih Kebijakan.
3. Pilih Buat kebijakan, lalu pilih JSON.
4. Di Tambahkan tindakan, filter berdasarkan S3. Tambahkan akses ListBucket, GetBucketAcl, dan GetBucketLocation.
5. Untuk Menambahkan sumber daya, pilih Tambah. Untuk jenis Resource, pilih bucket, lalu masukkan nama bucket Anda. Kemudian, pilih Tambahkan sumber daya.
6. Pilih Tambahkan pernyataan baru.
7. Di Tambahkan tindakan, filter berdasarkan S3. Tambahkan akses PutObject, ListMultipartUploadParts, dan AbortMultipartUpload.

8. Untuk Menambahkan sumber daya, pilih Tambah. Untuk jenis Sumber Daya, pilih objek, dan masukkan *nama bucket Anda/**. Kemudian, pilih Tambahkan sumber daya.
9. Pilih Tambahkan pernyataan baru.
10. Di Tambahkan tindakan, filter berdasarkan S3. Tambahkan akses ListAllMyBuckets.
11. Untuk Menambahkan sumber daya, pilih Tambah. Untuk jenis Sumber Daya, pilih Semua Sumber Daya. Kemudian, pilih Tambahkan sumber daya.
12. Jika Anda menggunakan kunci KMS Anda sendiri untuk mengenkripsi data:
    1. Pilih Tambahkan pernyataan baru.
    2. Di Tambahkan tindakan, filter berdasarkan KMS. Tambahkan akses GenerateDataKeydan Dekripsi.
    3. Untuk Menambahkan sumber daya, pilih Tambah. Untuk jenis Sumber Daya, pilih Semua Sumber Daya. Kemudian, pilih Tambahkan sumber daya.
13. Pilih Berikutnya.
14. Untuk Nama kebijakan, masukkan nama untuk kebijakan ini.
15. (Opsional) Untuk Deskripsi, masukkan deskripsi untuk kebijakan ini.
16. Pilih Buat kebijakan.

AWS CLI

Untuk membuat kebijakan IAM guna mengizinkan Amazon RDS mengakses bucket Amazon S3 Anda

1. Jalankan perintah create-policy. Dalam contoh berikut, ganti *iam_policy_name dan s3_bucket_name dengan nama untuk kebijakan IAM Anda dan nama bucket Amazon* S3 target Anda.

   Untuk Linux, macOS, atau Unix:

```
aws iam create-policy \
    --policy-name iam_policy_name \
    --policy-document '{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Sid": "Statement1",
                "Effect": "Allow",
                "Action": [
```

```
                    "s3:ListBucket",
                    "s3:GetBucketAcl",
                    "s3:GetBucketLocation"
                ],
                "Resource": [
                    "arn:aws:s3:::s3_bucket_name"
                ]
            },
            {
                "Sid": "Statement2",
                "Effect": "Allow",
                "Action": [
                    "s3:PutObject",
                    "s3:ListMultipartUploadParts",
                    "s3:AbortMultipartUpload"
                ],
                "Resource": [
                    "arn:aws:s3:::s3_bucket_name/*"
                ]
            },
            {
                "Sid": "Statement3",
                "Effect": "Allow",
                "Action": [
                    "s3:ListAllMyBuckets"
                ],
                "Resource": [
                    "*"
                ]
            },
            {
                "Sid": "Statement4",
                "Effect": "Allow",
                "Action": [
                    "kms:GenerateDataKey",
                    "kms:Decrypt"
                ],
                "Resource": [
                    "*"
                ]
            }
        ]
}'
```

Untuk Windows:

```
aws iam create-policy ^
    --policy-name iam_policy_name ^
    --policy-document '{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Sid": "Statement1",
                "Effect": "Allow",
                "Action": [
                    "s3:ListBucket",
                    "s3:GetBucketAcl",
                    "s3:GetBucketLocation"
                ],
                "Resource": [
                    "arn:aws:s3:::s3_bucket_name"
                ]
            },
            {
                "Sid": "Statement2",
                "Effect": "Allow",
                "Action": [
                    "s3:PutObject",
                    "s3:ListMultipartUploadParts",
                    "s3:AbortMultipartUpload"
                ],
                "Resource": [
                    "arn:aws:s3:::s3_bucket_name/*"
                ]
            },
            {
                "Sid": "Statement3",
                "Effect": "Allow",
                "Action": [
                    "s3:ListAllMyBuckets"
                ],
                "Resource": [
                    "*"
                ]
            },
            {
                "Sid": "Statement4",
```

```
                "Effect": "Allow",
                "Action": [
                    "kms:GenerateDataKey",
                    "kms:Decrypt"
                ],
                "Resource": [
                    "*"
                ]
            }
        ]
}'
```

2. Setelah kebijakan dibuat, catat ARN kebijakan. Anda memerlukan ARN untuk Langkah 3: Buat peran IAM dan lampirkan kebijakan IAM Anda.

Lihat informasi tentang pembuatan kebijakan IAM di Membuat kebijakan IAM dalam Panduan Pengguna IAM.

Langkah 3: Buat peran IAM dan lampirkan kebijakan IAM Anda

Langkah ini mengasumsikan bahwa Anda membuat kebijakan IAM di. Langkah 2: Buat kebijakan IAM Pada langkah ini, Anda membuat peran IAM untuk RDS Anda untuk instans Db2 DB dan kemudian melampirkan kebijakan IAM Anda ke peran tersebut.

Anda dapat membuat peran IAM untuk instans DB Anda dengan menggunakan konsol atau. AWS CLI

Konsol

Untuk membuat peran IAM dan melampirkan kebijakan IAM padanya

1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi, silakan pilih Peran.
3. Pilih Buat peran.
4. Untuk jenis entitas tepercaya, pilih Layanan AWS.
5. Untuk Service atau use case, pilih RDS, lalu pilih RDS — Add Role to Database.
6. Pilih Berikutnya.
7. Untuk Kebijakan izin, cari dan pilih nama kebijakan IAM yang Anda buat.

8. Pilih Berikutnya.
9. Untuk Nama peran, masukkan nama peran.
10. (Opsional) Untuk Deskripsi, masukkan deskripsi untuk peran baru ini.
11. Pilih Buat peran.

AWS CLI

Untuk membuat peran IAM dan melampirkan kebijakan IAM padanya

1. Jalankan perintah create-role. Dalam contoh berikut, ganti *iam_role_name* dengan nama untuk peran IAM Anda.

   Untuk Linux, macOS, atau Unix:

```
aws iam create-role \
    --role-name iam_role_name \
    --assume-role-policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
            "Service": "rds.amazonaws.com"
          },
          "Action": "sts:AssumeRole"
        }
      ]
    }'
```

   Untuk Windows:

```
aws iam create-role ^
    --role-name iam_role_name ^
    --assume-role-policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
            "Service": "rds.amazonaws.com"
```

```
          },
          "Action": "sts:AssumeRole"
        }
      ]
    }'
```

2. Setelah peran dibuat, perhatikan ARN dari peran ini. Anda membutuhkan ARN ini untuk langkah selanjutnya,. Langkah 4: Konfigurasikan grup opsi untuk pencatatan audit Db2
3. Jalankan perintah attach-role-policy. Dalam contoh berikut, ganti *iam_policy_arn* dengan ARN dari kebijakan IAM yang Anda buat di Langkah 2: Buat kebijakan IAM. Ganti *iam_role_name* dengan nama peran IAM yang baru saja Anda buat.

   Untuk Linux, macOS, atau Unix:

   ```
   aws iam attach-role-policy \
      --policy-arn iam_policy_arn \
      --role-name iam_role_name
   ```

   Untuk Windows:

   ```
   aws iam attach-role-policy ^
      --policy-arn iam_policy_arn ^
      --role-name iam_role_name
   ```

Lihat informasi yang lebih lengkap di Membuat peran untuk melimpahkan izin ke pengguna IAM dalam Panduan Pengguna IAM.

Langkah 4: Konfigurasikan grup opsi untuk pencatatan audit Db2

Proses untuk menambahkan opsi pencatatan audit Db2 ke RDS untuk instans Db2 DB adalah sebagai berikut:

1. Buat grup opsi baru, atau salin atau ubah grup opsi yang sudah ada.
2. Tambahkan dan konfigurasikan semua opsi yang diperlukan.
3. Kaitkan grup opsi dengan instans DB.

Setelah Anda menambahkan opsi pencatatan audit Db2, Anda tidak perlu memulai ulang instans DB Anda. Begitu grup opsi aktif, Anda dapat membuat audit dan menyimpan log audit di bucket S3.

Untuk menambah dan mengonfigurasi pencatatan audit Db2 pada grup opsi instans DB

1. Pilih salah satu cara berikut:
   - Gunakan grup opsi yang sudah ada.
   - Buat grup opsi DB kustom, dan gunakan grup opsi itu. Untuk informasi selengkapnya, lihat Membuat grup opsi.
2. Tambahkan opsi DB2_AUDIT ke grup opsi, dan konfigurasikan pengaturan opsi. Untuk informasi cara menambahkan aturan selengkapnya, lihat Menambahkan opsi ke grup opsi.
   - Untuk IAM_ROLE_ARN, masukkan ARN dari peran IAM yang Anda buat. the section called "Buat peran IAM dan lampirkan kebijakan IAM Anda"
   - Untuk S3_BUCKET_ARN, masukkan ARN bucket S3 yang akan digunakan untuk log audit Db2 Anda. Bucket harus berada di Wilayah yang sama dengan RDS Anda untuk instans Db2 DB. Kebijakan yang terkait dengan peran IAM yang Anda masukkan harus mengizinkan operasi yang diperlukan pada sumber daya ini.
3. Terapkan grup opsi ke instans DB baru atau yang sudah ada. Pilih salah satu cara berikut:
   - Jika Anda membuat instans DB baru, terapkan grup opsi ketika Anda meluncurkan instans.
   - Di instans DB yang sudah ada, terapkan grup opsi dengan mengubah instans lalu memberikan grup opsi baru. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Langkah 5: Konfigurasikan kebijakan audit

Untuk mengonfigurasi kebijakan audit untuk database RDS untuk Db2 Anda, sambungkan ke `rdsadmin` database menggunakan nama pengguna utama dan kata sandi utama untuk RDS Anda untuk instans Db2 DB. Kemudian, panggil prosedur `rdsadmin.configure_db_audit` tersimpan dengan nama DB database Anda dan nilai parameter yang berlaku.

Contoh berikut menghubungkan ke database dan mengkonfigurasi kebijakan audit untuk `testdb` dengan kategori AUDIT, CHECKING, OBJMAINT, SECMAINT, SYSADMIN, dan VALIDATE. Nilai status `BOTH` mencatat keberhasilan dan kegagalan, dan `ERROR TYPE` secara `NORMAL` default. Untuk informasi selengkapnya tentang cara menggunakan prosedur tersimpan ini, lihatthe section called "rdsadmin.configure_db_audit".

```
db2 "connect to rdsadmin user master_user using master_password"
db2 "call rdsadmin.configure_db_audit('testdb', 'ALL', 'BOTH', ?)"
```

Langkah 6: Periksa konfigurasi audit

Untuk memastikan kebijakan audit Anda diatur dengan benar, periksa status konfigurasi audit Anda.

Untuk memeriksa konfigurasi, sambungkan ke `rdsadmin` database menggunakan nama pengguna utama dan kata sandi master untuk RDS Anda untuk instans Db2 DB. Kemudian, jalankan pernyataan SQL berikut dengan nama DB database Anda. Dalam contoh berikut, nama DB adalah *`testdb`*.

```
db2 "select task_id, task_type, database_name, lifecycle,
    varchar(bson_to_json(task_input_params), 500) as task_params,
    cast(task_output as varchar(500)) as task_output
    from table(rdsadmin.get_task_status(null,'testdb','CONFIGURE_DB_AUDIT'))"

Sample Output

TASK_ID              TASK_TYPE            DATABASE_NAME       LIFECYCLE
-------------------- -------------------- --------------- -------------
                   2  CONFIGURE_DB_AUDIT            DB2DB       SUCCESS

... continued ...
TASK_PARAMS
------------------------------------------------------------
{ "AUDIT_CATEGORY" : "ALL", "CATEGORY_SETTING" : "BOTH" }

... continued ...
                                          TASK_OUTPUT
--------------------------------------------------
2023-12-22T20:27:03.029Z Task execution has started.

2023-12-22T20:27:04.285Z Task execution has completed successfully.
```

Mengelola pencatatan audit Db2

Setelah menyiapkan pencatatan audit Db2, Anda dapat mengubah kebijakan audit untuk database tertentu, atau menonaktifkan pencatatan audit di tingkat database atau untuk seluruh instans DB. Anda juga dapat mengubah bucket Amazon S3 tempat file log Anda diunggah.

Topik

- Memodifikasi kebijakan audit Db2
- Memodifikasi lokasi file log Anda

- Menonaktifkan pencatatan audit Db2

Memodifikasi kebijakan audit Db2

Untuk memodifikasi kebijakan audit untuk RDS tertentu untuk database Db2, jalankan prosedur yang `rdsadmin.configure_db_audit` disimpan. Dengan prosedur tersimpan ini, Anda dapat mengubah kategori, pengaturan kategori, dan konfigurasi jenis kesalahan kebijakan audit. Untuk informasi selengkapnya, lihat the section called “rdsadmin.configure_db_audit”.

Memodifikasi lokasi file log Anda

Untuk mengubah bucket Amazon S3 tempat file log Anda diunggah, lakukan salah satu hal berikut:

- Ubah grup opsi saat ini yang dilampirkan ke RDS Anda untuk instans Db2 DB — Perbarui `S3_BUCKET_ARN` pengaturan untuk `DB2_AUDIT` opsi untuk menunjuk ke bucket baru. Juga, pastikan untuk memperbarui kebijakan IAM yang dilampirkan ke peran IAM yang ditentukan oleh `IAM_ROLE_ARN` pengaturan di grup opsi terlampir. Kebijakan IAM ini harus menyediakan bucket baru Anda dengan izin akses yang diperlukan. Untuk informasi tentang izin yang diperlukan dalam kebijakan IAM, lihat. Buat kebijakan IAM
- Lampirkan RDS Anda untuk instans Db2 DB ke grup opsi yang berbeda - Ubah instans DB Anda untuk mengubah grup opsi yang dilampirkan padanya. Pastikan bahwa grup opsi baru dikonfigurasi dengan benar `S3_BUCKET_ARN` dan `IAM_ROLE_ARN` pengaturan. Untuk informasi tentang cara mengkonfigurasi pengaturan ini untuk `DB2_AUDIT` opsi, lihatKonfigurasikan grup opsi.

Saat Anda memodifikasi grup opsi, pastikan Anda segera menerapkan perubahan. Untuk informasi selengkapnya, lihat the section called “Memodifikasi instans DB”.

Menonaktifkan pencatatan audit Db2

Untuk menonaktifkan pencatatan audit Db2, lakukan salah satu hal berikut:

- Nonaktifkan pencatatan audit untuk instans RDS untuk Db2 DB - Ubah instans DB Anda dan hapus grup opsi dengan `DB2_AUDIT` opsi darinya. Untuk informasi selengkapnya, lihat the section called “Memodifikasi instans DB”.
- Nonaktifkan pencatatan audit untuk database tertentu — Hentikan pencatatan audit dan hapus kebijakan audit `rdsadmin.disable_db_audit` dengan memanggil nama DB database Anda. Untuk informasi selengkapnya, lihat the section called “rdsadmin.disable_db_audit”.

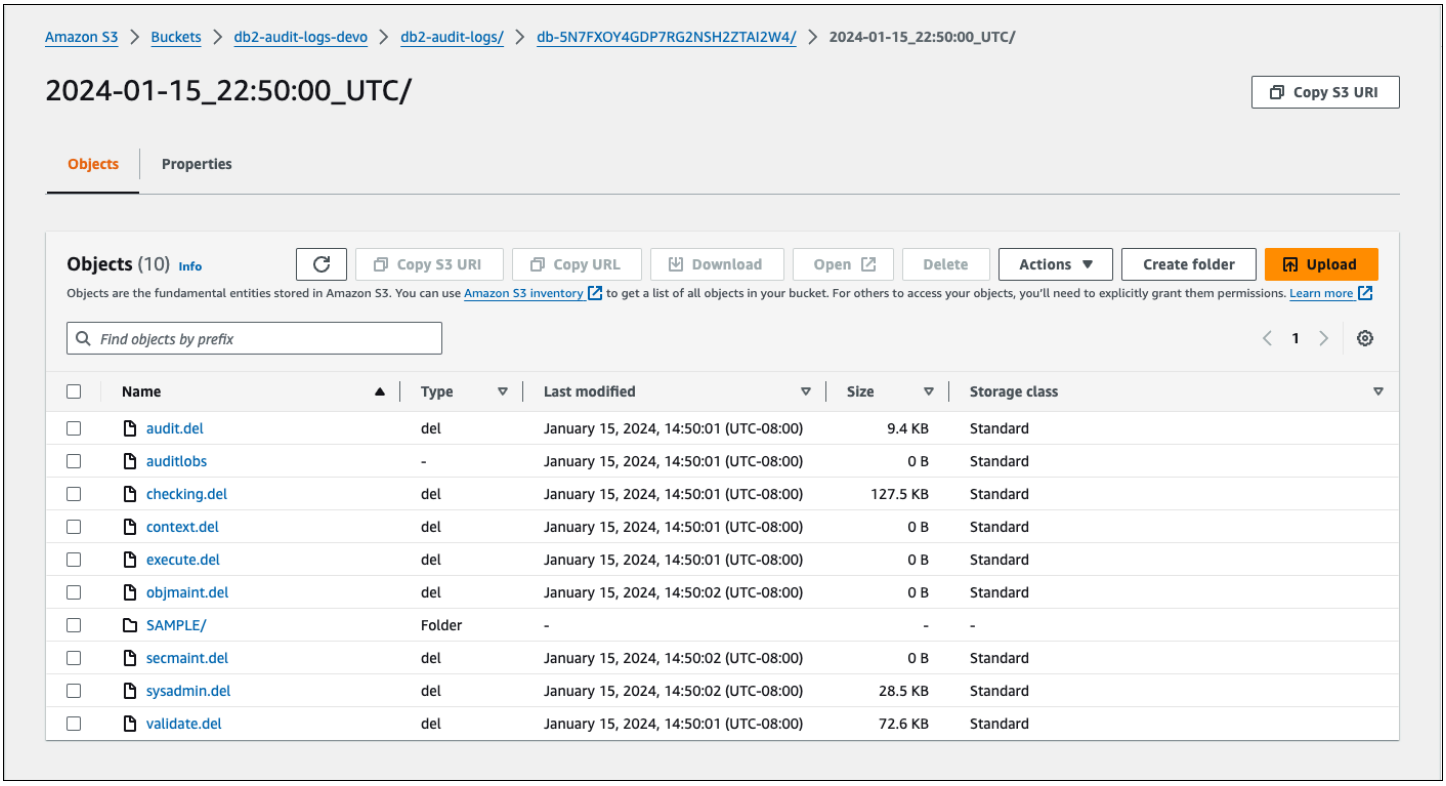
```
db2 "call rdsadmin.disable_db_audit(
    'db_name')"
```

Melihat log audit

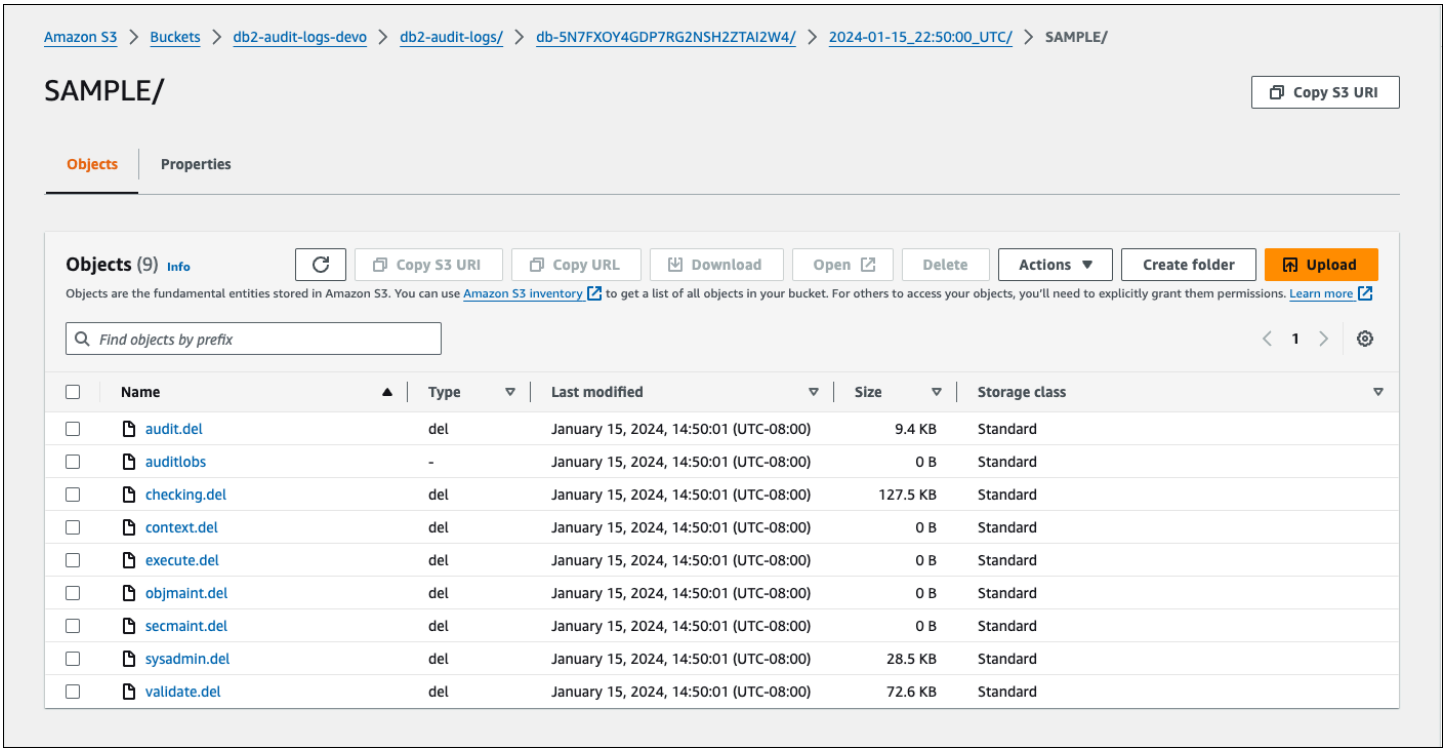
Setelah mengaktifkan pencatatan audit Db2, tunggu setidaknya satu jam sebelum melihat data audit di bucket Amazon S3 Anda. Amazon RDS secara otomatis mengirimkan log dari RDS Anda untuk instans Db2 DB ke lokasi berikut:

- Log tingkat instans DB - *bucket_name*/db2-audit-logs/*dbi_resource_id*/*date_time_utc*/
- Log tingkat basis data - *bucket_name*/db2-audit-logs/*dbi_resource_id*/*date_time_utc*/*db_name*/

Contoh tangkapan layar berikut dari konsol Amazon S3 menunjukkan daftar folder untuk RDS untuk file log tingkat instans Db2 DB.



Contoh screenshot berikut dari konsol Amazon S3 menunjukkan file log tingkat database untuk RDS untuk instans Db2 DB.



Pemecahan masalah pencatatan audit Db2

Gunakan informasi berikut untuk memecahkan masalah umum dengan pencatatan audit Db2.

Tidak dapat mengonfigurasi kebijakan audit

Jika memanggil prosedur yang disimpan `rdsadmin.configure_db_audit` mengembalikan kesalahan, bisa jadi grup opsi dengan `DB2_AUDIT` opsi tidak terkait dengan RDS untuk instans Db2 DB. Ubah instance DB untuk menambahkan grup opsi, lalu coba panggil prosedur tersimpan lagi. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Tidak ada data di bucket Amazon S3

Jika data pencatatan hilang dari bucket Amazon S3, periksa hal berikut:

- Bucket Amazon S3 berada di Wilayah yang sama dengan RDS Anda untuk instans Db2 DB.
- Peran yang Anda tentukan dalam pengaturan `IAM_ROLE_ARN` opsi dikonfigurasi dengan izin yang diperlukan untuk mengunggah log ke bucket Amazon S3 Anda. Untuk informasi selengkapnya, lihat Buat kebijakan IAM.
- ARN untuk pengaturan `S3_BUCKET_ARN` opsi `IAM_ROLE_ARN` dan benar dalam grup opsi yang terkait dengan RDS Anda untuk instans Db2 DB. Untuk informasi selengkapnya, lihat Konfigurasikan grup opsi.

Anda dapat memeriksa status tugas konfigurasi logging audit Anda dengan menghubungkan ke database dan menjalankan pernyataan SQL. Untuk informasi selengkapnya, lihat Periksa konfigurasi audit.

Anda juga dapat memeriksa acara untuk mengetahui lebih lanjut tentang mengapa log mungkin hilang. Untuk informasi tentang cara melihat acara, lihatthe section called “Melihat log, peristiwa, dan aliran di konsol Amazon RDS”.

Prosedur tersimpan eksternal untuk RDS untuk Db2

Anda dapat membuat rutinitas eksternal dan mendaftarkannya dengan RDS Anda untuk database Db2 sebagai prosedur tersimpan eksternal. Saat ini, RDS untuk Db2 hanya mendukung rutinitas berbasis Java untuk prosedur tersimpan eksternal.

Prosedur tersimpan eksternal berbasis Java

Prosedur tersimpan eksternal berbasis Java adalah rutinitas Java eksternal yang Anda daftarkan dengan RDS Anda untuk database Db2 sebagai prosedur tersimpan eksternal.

Topik

Batasan untuk prosedur tersimpan eksternal berbasis Java

Sebelum Anda mengembangkan rutinitas eksternal Anda, pertimbangkan batasan dan batasan berikut.

Untuk membuat rutinitas eksternal Anda, pastikan untuk menggunakan Java Development Kit (JDK) yang disediakan oleh Db2. Untuk informasi selengkapnya, lihat dukungan perangkat lunak Java untuk produk database Db2.

Program Java Anda dapat membuat file hanya di `/tmp` direktori, dan Amazon RDS tidak mendukung pengaktifan izin executable atau Set User ID (SUID) pada file-file ini. Program Java Anda juga tidak dapat menggunakan panggilan sistem soket atau panggilan sistem berikut:

- _sysctl
- acct
- afs_syscall
- bpf
- capset
- chown
- chroot

- create_module
- delete_module
- fanotify_init
- fanotify_mark
- finit_module
- fsconfig
- fsopen
- fspick
- get_kernel_syms
- getpmsg
- init_module
- mount
- move_mount
- nfsservctl
- open_by_handle_at
- open_tree
- pivot_root
- putpmsg
- query_module
- quotactl
- reboot
- security
- setdomainname
- setfsuid
- sethostname
- sysfs
- tuxcall
- umount2
- uselib

- ustat
- vhangup
- vserver

Untuk pembatasan tambahan pada rutinitas eksternal untuk Db2, lihat Pembatasan rutinitas eksternal dalam dokumentasi. IBM Db2

Mengkonfigurasi prosedur tersimpan eksternal berbasis Java

Untuk mengkonfigurasi prosedur tersimpan eksternal, buat file.jar dengan rutinitas eksternal Anda, instal pada database RDS untuk Db2 Anda, dan kemudian daftarkan sebagai prosedur tersimpan eksternal.

Topik

- Langkah 1: Aktifkan prosedur tersimpan eksternal
- Langkah 2: Instal file.jar dengan rutinitas eksternal Anda
- Langkah 3: Daftarkan prosedur tersimpan eksternal
- Langkah 4: Validasi prosedur tersimpan eksternal

Langkah 1: Aktifkan prosedur tersimpan eksternal

Untuk mengaktifkan prosedur tersimpan eksternal, dalam grup parameter kustom yang terkait dengan instans DB Anda, setel parameter `db2_alternate_authz_behaviour` ke salah satu nilai berikut:

- `EXTERNAL_ROUTINE_DBADM`— Secara implisit memberikan izin kepada pengguna, grup, atau peran apa pun dengan `DBADM` otoritas. `CREATE_EXTERNAL_ROUTINE`
- `EXTERNAL_ROUTINE_DBAUTH`— Memungkinkan pengguna dengan `DBADM` wewenang untuk memberikan `CREATE_EXTERNAL_ROUTINE` izin kepada pengguna, grup, atau peran apa pun. Dalam hal ini, tidak ada pengguna, grup, atau peran yang secara implisit diberikan izin ini, bahkan pengguna dengan `DBADM` otoritas.

Untuk informasi selengkapnya tentang pengaturan ini, lihat pernyataan GRANT (otoritas database) dalam IBM Db2 dokumentasi.

Anda dapat membuat dan mengubah grup parameter kustom dengan menggunakan AWS Management Console, AWS CLI, atau Amazon RDS API.

Konsol

Untuk mengkonfigurasi parameter db2_alternate_authz_behavior dalam grup parameter kustom

1. Jika Anda ingin menggunakan grup parameter DB kustom yang berbeda dari yang digunakan instans DB Anda, buat grup parameter DB baru. Pastikan bahwa grup parameter kustom baru menyertakan IBM ID untuk opsi lisensi Bring Your Own License (BYOL). Untuk informasi tentang ID ini, lihatthe section called “ID IBM”. Lihat informasi yang lebih lengkap tentang cara membuat grup parameter basis data di Membuat grup parameter DB.
2. Tetapkan nilai untuk `db2_alternate_authz_behaviour` parameter dalam grup parameter kustom Anda. Lihat informasi yang lebih lengkap tentang mengubah grup parameter di Memodifikasi parameter dalam grup parameter DB.

AWS CLI

Untuk mengkonfigurasi parameter db2_alternate_authz_behavior dalam grup parameter kustom

1. Jika Anda ingin menggunakan grup parameter DB kustom yang berbeda dari yang digunakan instans DB Anda, buat grup parameter khusus dengan menjalankan create-db-parameter-groupperintah. Pastikan bahwa grup parameter kustom baru menyertakan IBM ID untuk opsi lisensi Bring Your Own License (BYOL). Untuk informasi tentang ID ini, lihatthe section called “ID IBM”.

   Sertakan opsi-opsi yang diperlukan berikut:

   - `--db-parameter-group-name` – Nama untuk grup parameter yang sedang Anda buat.
   - `--db-parameter-group-family` – Edisi mesin dan versi utama Db2. Nilai yang valid adalah `db2-se-11.5` dan `db2-ae-11.5`.
   - `--description` – Deskripsi untuk grup parameter ini.

   Lihat informasi yang lebih lengkap tentang cara membuat grup parameter basis data di Membuat grup parameter DB.

   Contoh berikut menunjukkan cara membuat grup parameter kustom bernama `MY_EXT_SP_PARAM_GROUP` untuk keluarga grup parameter`db2-se-11.5`.

   Untuk Linux, macOS, atau Unix:

```
aws rds create-db-parameter-group \
--region us-east-1 \
--db-parameter-group-name MY_EXT_SP_PARAM_GROUP \
--db-parameter-group-family db2-se-11.5 \
--description "test db2 external routines"
```

Untuk Windows:

```
aws rds create-db-parameter-group ^
--region us-east-1 ^
--db-parameter-group-name MY_EXT_SP_PARAM_GROUP ^
--db-parameter-group-family db2-se-11.5 ^
--description "test db2 external routines"
```

2. Ubah `db2_alternate_authz_behaviour` parameter dalam grup parameter kustom Anda dengan menjalankan modify-db-parameter-groupperintah.

   Sertakan opsi-opsi yang diperlukan berikut:

   - `--db-parameter-group-name` – Nama grup parameter yang Anda buat.
   - `--parameters` – Array nama parameter, nilai parameter, dan metode aplikasi untuk pembaruan parameter.

   Lihat informasi yang lebih lengkap tentang mengubah grup parameter di Memodifikasi parameter dalam grup parameter DB.

   Contoh berikut menunjukkan cara memodifikasi grup parameter `MY_EXT_SP_PARAM_GROUP` dengan menetapkan nilai `db2_alternate_authz_behaviour` ke`EXTERNAL_ROUTINE_DBADM`.

   Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name MY_EXT_SP_PARAM_GROUP \
    --parameters
 "ParameterName='db2_alternate_authz_behaviour',ParameterValue='EXTERNAL_ROUTINE_DBADM',App
```

   Untuk Windows:

```
aws rds modify-db-parameter-group ^
    --db-parameter-group-name MY_EXT_SP_PARAM_GROUP ^
    --parameters
 "ParameterName='db2_alternate_authz_behaviour',ParameterValue='EXTERNAL_ROUTINE_DBADM',App
```

API RDS

Untuk mengkonfigurasi parameter db2_alternate_authz_behavior dalam grup parameter kustom

1. Jika Anda ingin menggunakan grup parameter DB kustom yang berbeda dari yang digunakan instans DB Anda, buat grup parameter DB baru dengan menggunakan CreateDBParameterGroupoperasi Amazon RDS API. Pastikan bahwa grup parameter kustom baru menyertakan ID IBM untuk opsi lisensi Bring Your Own License (BYOL). Untuk informasi tentang ID ini, lihatthe section called “ID IBM”.

   Sertakan parameter-parameter yang diperlukan berikut:

   - `DBParameterGroupName`
   - `DBParameterGroupFamily`
   - `Description`

   Lihat informasi yang lebih lengkap tentang cara membuat grup parameter basis data di Membuat grup parameter DB.
2. Ubah `db2_alternate_authz_behaviour` parameter dalam grup parameter kustom yang Anda buat dengan menggunakan ModifyDBParameterGroupoperasi RDS API.

   Sertakan parameter-parameter yang diperlukan berikut:

   - `DBParameterGroupName`
   - `Parameters`

   Lihat informasi yang lebih lengkap tentang mengubah grup parameter di Memodifikasi parameter dalam grup parameter DB.

Langkah 2: Instal file.jar dengan rutinitas eksternal Anda

Setelah Anda membuat rutinitas Java Anda, buat file.jar dan kemudian jalankan `db2 "call sqlj.install_jar('file:`*`file_path`*`',`*`jar_ID`*`)"` untuk menginstalnya pada database RDS untuk Db2 Anda.

Contoh berikut menunjukkan cara membuat rutin Java dan menginstalnya pada database RDS untuk Db2. Contoh termasuk kode sampel untuk rutinitas sederhana yang dapat Anda gunakan untuk menguji proses. Contoh ini membuat asumsi berikut:

- Kode Java dikompilasi pada server tempat Db2 diinstal. Ini adalah praktik terbaik karena tidak mengkompilasi dengan JDK yang disediakan IBM dapat mengakibatkan kesalahan yang tidak dapat dijelaskan.
- Server memiliki database RDS untuk Db2 yang dikatalogkan secara lokal.

Jika Anda ingin mencoba proses dengan kode contoh berikut, salin dan kemudian simpan ke file bernama`MYJAVASP.java`.

```
import java.sql.*;
public class MYJAVASP
{
public static void my_JAVASP (String inparam) throws SQLException, Exception
{
try
{
// Obtain the calling context's connection details.
Connection myConn = DriverManager.getConnection("jdbc:default:connection");
String myQuery = "INSERT INTO TEST.TEST_TABLE VALUES (?, CURRENT DATE)";
PreparedStatement myStmt = myConn.prepareStatement(myQuery);
myStmt.setString(1, inparam);
myStmt.executeUpdate();
}
catch (SQLException sql_ex)
{
throw sql_ex;
}
catch (Exception ex)
{
throw ex;
}
}
```

Perintah berikut mengkompilasi rutin Java.

```
~/sqllib/java/jdk64/bin/javac MYJAVASP.java
```

Perintah berikut membuat file.jar.

```
~/sqllib/java/jdk64/bin/jar cvf MYJAVASP.jar MYJAVASP.class
```

Perintah berikut terhubung ke database bernama `MY_DB2_DATABASE` dan menginstal file.jar.

```
db2 "connect to MY_DB2_DATABASE user master_username using master_password"

db2 "call sqlj.install_jar('file:/tmp/MYJAVASP.jar','MYJAVASP')"
db2 "call sqlj.refresh_classes()"
```

Langkah 3: Daftarkan prosedur tersimpan eksternal

Setelah Anda menginstal file.jar pada RDS Anda untuk database Db2, daftarkan sebagai prosedur tersimpan dengan menjalankan perintah or. `db2 CREATE PROCEDURE` `db2 REPLACE PROCEDURE`

Contoh berikut menunjukkan cara menghubungkan ke database dan mendaftarkan rutinitas Java yang dibuat pada langkah sebelumnya sebagai prosedur tersimpan.

```
db2 "connect to MY_DB2_DATABASE user master_username using master_password"

create procedure TESTSP.MYJAVASP (in input char(6))
specific myjavasp
dynamic result sets 0
deterministic
language java
parameter style java
no dbinfo
fenced
threadsafe
modifies sql data
program type sub
external name 'MYJAVASP!my_JAVASP';
```

Langkah 4: Validasi prosedur tersimpan eksternal

Gunakan langkah-langkah berikut untuk menguji sampel prosedur tersimpan eksternal yang terdaftar pada langkah sebelumnya.

Untuk memvalidasi prosedur tersimpan eksternal

1. Buat tabel seperti `TEST.TEST_TABLE` pada contoh berikut.

```
db2 "create table TEST.TEST_TABLE(C1 char(6), C2 date)"
```

2. Panggil prosedur tersimpan eksternal yang baru. Panggilan mengembalikan status`0`.

```
db2 "call TESTSP.MYJAVASP('test')"
Return Status = 0
```

3. Kueri tabel yang Anda buat di langkah 1 untuk memverifikasi hasil panggilan prosedur yang disimpan.

```
db2 "SELECT * from TEST.TEST_TABLE"
```

   Query menghasilkan output yang mirip dengan contoh berikut:

```
C1     C2
------ ----------
test   02/05/2024
```

Masalah dan batasan yang diketahui untuk Amazon RDS untuk Db2

Item berikut adalah masalah dan batasan yang diketahui untuk bekerja dengan Amazon RDS untuk Db2.

Topik

Batasan otentikasi

Amazon RDS disetel `DB2AUTH` ke`JCC_ENFORCE_SECMEC`. Karena tidak `JCC_ENFORCE_SECMEC` dapat dimodifikasi, Amazon RDS memberlakukan enkripsi kata sandi pada koneksi JDBC.

Referensi prosedur tersimpan RDS for Db2

Topik-topik ini menjelaskan prosedur-prosedur tersimpan sistem yang tersedia untuk instans Amazon RDS yang menjalankan mesin RDS for Db2. Pengguna master harus menjalankan prosedur-prosedur ini.

Topik

Memberikan dan mencabut privilese

Prosedur-prosedur tersimpan berikut memberikan dan mencabut privilese.

Topik

rdsadmin.create_role

Menciptakan peran.

Sintaks

```
db2 "call rdsadmin.create_role(
    'database_name',
    'role_name')"
```

Parameter-parameter

Parameter-parameter berikut diperlukan:

database_name

Nama database yang akan dijalankan perintah. Tipe datanya `varchar`.

role_name

Nama peran yang ingin Anda buat. Tipe datanya `varchar`.

Catatan penggunaan

Lihat informasi tentang pemeriksaan status pembuatan peran di rdsadmin.get_task_status.

Contoh-contoh

Contoh berikut menciptakan peran berjudul `MY_ROLE` untuk basis data `DB2DB`.

```
db2 "call rdsadmin.create_role(
    'DB2DB',
    'MY_ROLE')"
```

rdsadmin.grant_role

Menetapkan peran ke peran, pengguna, atau grup.

Sintaks

```
db2 "call rdsadmin.grant_role(
    ?,
    'database_name',
    'role_name',
    'grantee',
    'admin_option')"
```

Parameter-parameter

Parameter output berikut diperlukan:

?

Penanda parameter yang menampilkan pengenal unik untuk tugas tersebut. Parameter ini hanya menerima?.

Parameter input berikut diperlukan:

database_name

Nama database yang akan dijalankan perintah. Tipe datanya `varchar`.

role_name

Nama peran yang ingin Anda buat. Tipe datanya `varchar`.

penerima hibah

Peran, pengguna, atau grup untuk menerima otorisasi. Tipe datanya `varchar`. Nilai yang valid:`ROLE`,`USER`,`GROUP`,`PUBLIC`.

Format harus nilai diikuti dengan nama. Pisahkan beberapa nilai dan nama dengan koma. Contoh: 'USER *user1*, *user2*, GROUP *group1*, *group2*'. Ganti nama dengan informasi Anda sendiri.

Parameter input berikut adalah opsional:

admin_option

Menentukan apakah penerima hibah `ROLE` memiliki `DBADM` otorisasi untuk menetapkan peran. Tipe datanya `char`. Nilai default-nya `N`.

Catatan penggunaan

Untuk informasi tentang memeriksa status penetapan peran, lihatrdsadmin.get_task_status.

Contoh-contoh

Contoh berikut menetapkan peran yang dipanggil `ROLE_TEST` untuk database `TESTDB` untuk peran yang dipanggil`role1`, pengguna dipanggil`user1`, dan grup yang dipanggil`group1`. `ROLE_TEST`diberikan otorisasi admin untuk menetapkan peran.

```
db2 "call rdsadmin.grant_role(
    ?,
    'TESTDB',
    'ROLE_TEST',
    'ROLE role1, USER user1, GROUP group1',
    'Y')"
```

Contoh berikut menetapkan peran yang dipanggil `ROLE_TEST` untuk database `TESTDB` untuk`PUBLIC`. `ROLE_TEST`tidak diberikan otorisasi admin untuk menetapkan peran.

```
db2 "call rdsadmin.grant_role(
    ?,
    'TESTDB',
    'ROLE_TEST',
```

```
    'PUBLIC')"
```

rdsadmin.add_user

Menambahkan pengguna ke daftar otorisasi.

Sintaks

```
db2 "call rdsadmin.add_user(
    'username',
    'password',
    'group_name,group_name')"
```

Parameter-parameter

Parameter-parameter berikut diperlukan:

username

Nama pengguna seorang pengguna. Tipe datanya `varchar`.

password

Kata sandi seorang pengguna. Tipe datanya `varchar`.

Parameter berikut bersifat opsional:

group_name

Nama grup yang ingin Anda tambahi pengguna. Tipe datanya `varchar`. Defaultnya adalah string kosong atau null.

Catatan penggunaan

Anda dapat menambahkan seorang pengguna ke satu atau beberapa grup dengan memisahkan nama-nama grup dengan koma.

Anda dapat membuat grup saat membuat pengguna baru, atau saat menambahkan grup ke pengguna yang ada. Anda tidak dapat membuat grup dengan sendirinya.

> **Note**
>
> Jumlah maksimum pengguna yang dapat Anda tambahkan dengan menelepon `rdsadmin.add_user` adalah 5.000.

Lihat informasi tentang pemeriksaan status penambahan pengguna di rdsadmin.get_task_status.

Contoh-contoh

Contoh berikut membuat pengguna dipanggil `jorge_souza` dan menetapkan pengguna ke grup yang dipanggil `sales` dan`inside_sales`.

```
db2 "call rdsadmin.add_user(
    'jorge_souza',
    '*******',
    'sales,inside_sales')"
```

rdsadmin.change_password

Mengubah kata sandi pengguna.

Sintaks

```
db2 "call rdsadmin.change_password(
    'username',
    'new_password')"
```

Parameter-parameter

Parameter-parameter berikut diperlukan:

username

Nama pengguna seorang pengguna. Tipe datanya `varchar`.

new_password

Kata sandi baru untuk si pengguna. Tipe datanya `varchar`.

Catatan penggunaan

Lihat informasi tentang pemeriksaan status perubahan kata sandi di rdsadmin.get_task_status.

Contoh-contoh

Contoh berikut mengubah kata sandi untuk `jorge_souza`.

```
db2 "call rdsadmin.change_password(
    'jorge_souza',
    '*******')"
```

rdsadmin.list_users

Memerinci pengguna pada daftar otorisasi.

Sintaks

```
db2 "call rdsadmin.list_users()"
```

Catatan penggunaan

Lihat informasi tentang pemeriksaan status pemerincian pengguna di rdsadmin.get_task_status.

rdsadmin.remove_user

Menghapus pengguna dari daftar otorisasi.

Sintaks

```
db2 "call rdsadmin.remove_user('username')"
```

Parameter-parameter

Parameter berikut diperlukan:

username

Nama pengguna seorang pengguna. Tipe datanya `varchar`.

Catatan penggunaan

Lihat informasi tentang pemeriksaan status penghapusan pengguna di rdsadmin.get_task_status.

Contoh-contoh

Contoh berikut menghapus `jorge_souza` dari kemampuan mengakses basis data dalam instans basis data RDS for Db2.

```
db2 "call rdsadmin.remove_user('jorge_souza')"
```

rdsadmin.add_groups

Menambahkan grup ke pengguna.

Sintaks

```
db2 "call rdsadmin.add_groups(
    'username',
    'group_name,group_name')"
```

Parameter-parameter

Parameter-parameter berikut diperlukan:

`username`

Nama pengguna seorang pengguna. Tipe datanya `varchar`.

`group_name`

Nama grup yang ingin Anda tambahi pengguna. Tipe datanya `varchar`. Default-nya adalah string kosong.

Catatan penggunaan

Anda dapat menambahkan satu atau beberapa grup ke seorang pengguna dengan memisahkan nama-nama grup dengan koma. Lihat informasi tentang pemeriksaan status penambahan grup di rdsadmin.get_task_status.

Contoh-contoh

Contoh berikut menambahkan grup-grup `direct_sales` dan `b2b_sales` ke pengguna `jorge_souza`.

```
db2 "call rdsadmin.add_groups(
    'jorge_souza',
```

```
    'direct_sales,b2b_sales')"
```

rdsadmin.remove_groups

Menghapus grup dari serang pengguna.

Sintaks

```
db2 "call rdsadmin.remove_groups(
    'username',
    'group_name,group_name')"
```

Parameter-parameter

Parameter-parameter berikut diperlukan:

username

Nama pengguna seorang pengguna. Tipe datanya `varchar`.

group_name

Nama grup yang darinya Anda ingin menghapus pengguna. Tipe datanya `varchar`.

Catatan penggunaan

Anda dapat menghapus satu atau beberapa grup dari seorang pengguna dengan memisahkan nama-nama grup dengan koma.

Lihat informasi tentang pemeriksaan status penghapusan grup di rdsadmin.get_task_status.

Contoh-contoh

Contoh berikut menghapus grup-grup `direct_sales` dan `b2b_sales` dari pengguna `jorge_souza`.

```
db2 "call rdsadmin.remove_groups(
    'jorge_souza',
    'direct_sales,b2b_sales')"
```

rdsadmin.dbadm_grant

Memberikan`DBADM`,`ACCESSCTRL`, atau `DATAACCESS` otorisasi untuk peran, pengguna, atau grup.

Sintaks

```
db2 "call rdsadmin.dbadm_grant(
    ?,
    'database_name',
    'authorization',
    'grantee')"
```

Parameter-parameter

Parameter output berikut diperlukan:

?

Penanda parameter yang menampilkan pengenal unik untuk tugas tersebut. Parameter ini hanya menerima?.

Parameter input berikut diperlukan:

database_name

Nama database yang akan dijalankan perintah. Tipe datanya `varchar`.

otorisasi

Jenis otorisasi untuk diberikan. Tipe datanya `varchar`. Nilai-nilai yang valid: `DBADM`, `ACCESSCTRL`, `DATAACCESS`.

Pisahkan beberapa jenis dengan koma.

penerima hibah

Peran, pengguna, atau grup untuk menerima otorisasi. Tipe datanya `varchar`. Nilai-nilai yang valid: `ROLE`, `USER`, `GROUP`.

Format harus nilai diikuti dengan nama. Pisahkan beberapa nilai dan nama dengan koma. Contoh: 'USER *user1*, *user2*, GROUP *group1*, *group2*'. Ganti nama dengan informasi Anda sendiri.

Catatan penggunaan

Peran untuk menerima akses harus ada.

Untuk informasi tentang memeriksa status pemberian akses admin database, lihatrdsadmin.get_task_status.

Contoh-contoh

Contoh berikut memberikan akses admin database ke database yang diberi nama TESTDB untuk peran ROLE_DBA tersebut.

```
db2 "call rdsadmin.dbadm_grant(
    ?,
    'TESTDB',
    'DBADM',
    'ROLE ROLE_DBA')"
```

Contoh berikut memberikan akses admin database ke database bernama TESTDB untuk user1 dangroup1.

```
db2 "call rdsadmin.dbadm_grant(
    ?,
    'TESTDB',
    'DBADM',
    'USER user1, GROUP group1')"
```

Contoh berikut memberikan akses admin database ke database bernama TESTDB untukuser1,, user2group1, dangroup2.

```
db2 "call rdsadmin.dbadm_grant(
    ?,
    'TESTDB',
    'DBADM',
    'USER user1, user2, GROUP group1, group2')"
```

rdsadmin.dbadm_revoke

MencabutDBADM,ACCESSCTRL, atau DATAACCESS otorisasi dari peran, pengguna, atau grup.

Sintaks

```
db2 "call rdsadmin.dbadm_revoke(
    ?,
    'database_name,
```

```
    'authorization',
    'grantee')"
```

Parameter-parameter

Parameter output berikut diperlukan:

?

Pengidentifikasi unik untuk tugas tersebut. Parameter ini hanya menerima?.

Parameter input berikut diperlukan:

database_name

Nama database yang akan dijalankan perintah. Tipe datanya `varchar`.

otorisasi

Jenis otorisasi untuk dicabut. Tipe datanya `varchar`. Nilai-nilai yang valid: `DBADM`, `ACCESSCTRL`, `DATAACCESS`.

Pisahkan beberapa jenis dengan koma.

penerima hibah

Peran, pengguna, atau grup untuk mencabut otorisasi dari. Tipe datanya `varchar`. Nilai-nilai yang valid: `ROLE`, `USER`, `GROUP`.

Format harus nilai diikuti dengan nama. Pisahkan beberapa nilai dan nama dengan koma. Contoh: 'USER *user1*, *user2*, GROUP *group1*, *group2*'. Ganti nama dengan informasi Anda sendiri.

Catatan penggunaan

Untuk informasi tentang memeriksa status pembatalan akses admin database, lihat. rdsadmin.get_task_status

Contoh-contoh

Contoh berikut mencabut akses admin database ke database yang diberi nama `TESTDB` untuk peran `ROLE_DBA` tersebut.

```
db2 "call rdsadmin.dbadm_revoke(
    ?,
    'TESTDB',
    'DBADM',
    'ROLE ROLE_DBA')"
```

Contoh berikut mencabut akses admin database ke database bernama TESTDB untuk user1 dangroup1.

```
db2 "call rdsadmin.dbadm_revoke(
    ?,
    'TESTDB',
    'DBADM',
    'USER user1, GROUP group1')"
```

Contoh berikut mencabut akses admin database ke database bernama TESTDB untukuser1,, user2group1, dangroup2.

```
db2 "call rdsadmin.dbadm_revoke(
    ?,
    'TESTDB',
    'DBADM',
    'USER user1, user2, GROUP group1, group2')"
```

Mengelola kolam penyangga

Prosedur-prosedur tersimpan berikut mengelola kolam penyangga.

Topik

rdsadmin.create_bufferpool

Membuat kolam penyangga.

Sintaks

```
db2 "call rdsadmin.create_bufferpool(
    'database_name',
    'buffer_pool_name',
    buffer_pool_size,
    'immediate',
    'automatic',
    page_size,
    number_block_pages,
    block_size)"
```

Parameter-parameter

Parameter-parameter berikut diperlukan:

database_name

Nama database untuk menjalankan perintah pada. Tipe datanya `varchar`.

buffer_pool_name

Nama kolam penyangga yang akan dibuat. Tipe datanya `varchar`.

Parameter berikut ini bersifat opsional:

buffer_pool_size

Ukuran kolam penyangga berupa jumlah halaman. Tipe datanya `integer`. Nilai default-nya `-1`.

immediate

Menentukan apakah perintah berjalan dengan serta-merta. Tipe datanya `char`. Nilai default-nya Y.

automatic

Menentukan apakah akan mengatur kolam buffer untuk otomatis. Tipe datanya `char`. Nilai default-nya Y.

page_size

Ukuran halaman kolam penyangga. Tipe datanya `integer`. Nilai yang valid:`4096`,`8192`,`16384`,`32768`. Nilai default-nya `8192`.

number_block_pages

Jumlah halaman blok di kolam penyangga. Tipe datanya `integer`. Nilai default-nya `0`.

block_size

Ukuran blok untuk halaman blok. Tipe datanya `integer`. Nilai-nilai yang valid: `2` sampai `256`. Nilai default-nya `32`.

Catatan penggunaan

Lihat informasi tentang pemeriksaan status pembuatan kolam penyangga di rdsadmin.get_task_status.

Contoh-contoh

Contoh berikut membuat kolam penyangga bernama `BP8` untuk basis data bernama `TESTDB` dengan parameter-parameter default, sehingga kolam penyangga menggunakan ukuran halaman 8 KB.

```
db2 "call rdsadmin.create_bufferpool(
    'TESTDB',
    BP8')"
```

Contoh berikut membuat kumpulan buffer `BP16` yang disebut database `TESTDB` yang menggunakan ukuran halaman 16 KB dengan jumlah halaman awal 1.000 dan diatur ke otomatis. Db2 segera

menjalankan perintah. Jika Anda menggunakan jumlah halaman awal -1, maka Db2 akan menggunakan alokasi halaman otomatis.

```
db2 "call rdsadmin.create_bufferpool(
    'TESTDB',
    'BP16',
    1000,
    'Y',
    'Y',
    16384)"
```

Contoh berikut membuat kolam buffer dipanggil BP16 untuk database yang disebutTESTDB. Buffer pool ini memiliki ukuran halaman 16 KB dengan jumlah halaman awal 10.000. Db2 menjalankan perintah segera menggunakan 500 halaman blok dengan ukuran blok 512.

```
db2 "call rdsadmin.create_bufferpool(
    'TESTDB',
    'BP16',
    10000,
    'Y',
    'Y',
    16384,
    500,
    512)"
```

rdsadmin.alter_bufferpool

Mengubah kolam penyangga.

Sintaks

```
db2 "call rdsadmin.alter_bufferpool(
    'database_name',
    'buffer_pool_name',
    buffer_pool_size,
    'immediate',
    'automatic',
    change_number_blocks,
    number_block_pages,
    block_size)"
```

Parameter-parameter

Parameter-parameter berikut diperlukan:

database_name

Nama database untuk menjalankan perintah pada. Tipe datanya `varchar`.

buffer_pool_name

Nama kolam penyangga untuk diubah. Tipe datanya `varchar`.

buffer_pool_size

Ukuran kolam penyangga berupa jumlah halaman. Tipe datanya `integer`.

Parameter berikut ini bersifat opsional:

immediate

Menentukan apakah perintah berjalan dengan serta-merta. Tipe datanya `char`. Nilai default-nya `Y`.

automatic

Menentukan apakah akan mengatur kolam buffer untuk otomatis. Tipe datanya `char`. Nilai default-nya `N`.

change_number_blocks

Menentukan apakah ada perubahan jumlah halaman blok di kolam penyangga. Tipe datanya `char`. Nilai default-nya `N`.

number_block_pages

Jumlah halaman blok di kolam penyangga. Tipe datanya `integer`. Nilai default-nya `0`.

block_size

Ukuran blok untuk halaman blok. Tipe datanya `integer`. Nilai-nilai yang valid: `2` sampai `256`. Nilai default-nya `32`.

Catatan penggunaan

Lihat informasi tentang pemeriksaan status pengubahan kolam penyangga di rdsadmin.get_task_status.

Contoh-contoh

Contoh berikut mengubah kumpulan buffer yang dipanggil BP16 untuk database yang dipanggil TESTDB untuk non-otomatis, dan mengubah ukuran menjadi 10.000 halaman. Db2 segera menjalankan perintah ini.

```
db2 "call rdsadmin.alter_bufferpool(
    'TESTDB',
    'BP16',
    10000,
    'Y',
    'N')"
```

rdsadmin.drop_bufferpool

Mengedrop kolam penyangga.

Sintaks

```
db2 "call rdsadmin.drop_bufferpool(
    'database_name',
    'buffer_pool_name'"
```

Parameter-parameter

Parameter-parameter berikut diperlukan:

database_name

Nama basis data yang memiliki kolam penyangga. Tipe datanya varchar.

buffer_pool_name

Nama kolam penyangga untuk dijatuhkan. Tipe datanya varchar.

Catatan penggunaan

Lihat informasi tentang pemeriksaan status pengedropan kolam penyangga di rdsadmin.get_task_status.

Contoh-contoh

Contoh berikut mengedrop kolam penyangga bernama BP16 untuk basis data bernama TESTDB.

```
db2 "call rdsadmin.drop_bufferpool(
    'TESTDB',
    'BP16')"
```

Mengelola basis data

Prosedur-prosedur tersimpan berikut mengelola basis data.

Topik

rdsadmin.create_database

Membuat basis data.

Sintaks

```
db2 "call rdsadmin.create_database('database_name')"
```

Parameter-parameter

> **Note**
>
> Prosedur tersimpan ini tidak memvalidasi kombinasi parameter yang diperlukan. Saat Anda memanggil rdsadmin.get_task_status, fungsi buatan pengguna ini dapat menghasilkan kesalahan karena kombinasi `database_codeset`, `database_territory`, dan `database_collation` yang tidak valid. Lihat informasi yang lebih lengkap di Choosing the code page, territory, and collation for your database dalam dokumentasi IBM Db2.

Parameter berikut diperlukan:

`database_name`

Nama basis data yang akan dibuat. Tipe datanya `varchar`.

Parameter berikut ini bersifat opsional:

database_page_size

Ukuran halaman default basis data. Nilai yang valid:4096,8192,16384,32768. Tipe datanya integer. Nilai default-nya 8192.

> **Important**
>
> Amazon RDS mendukung atomisitas tulis untuk halaman 4 KiB, 8 KiB, dan 16 KiB. Sebaliknya, 32 halaman KiB berisiko robek, atau sebagian data ditulis ke meja. Jika Anda menggunakan 32 halaman KiB, kami sarankan Anda mengaktifkan point-in-time pemulihan dan pencadangan otomatis. Jika tidak, Anda berisiko tidak dapat pulih dari halaman yang sobek. Lihat informasi yang lebih lengkap di the section called “Pengantar cadangan” dan the section called “Memulihkan instans DB dengan waktu yang ditentukan”.

database_code_set

Set kode untuk basis data. Tipe datanya varchar. Nilai default-nya UTF-8.

database_territory

Kode negara dua huruf untuk basis data. Tipe datanya varchar. Nilai default-nya US.

database_collation

Urutan pemeriksaan yang menentukan bagaimana string karakter yang disimpan dalam database diurutkan dan dibandingkan. Tipe datanya varchar.

Nilai yang valid:

- COMPATIBILITY— Urutan pemeriksaan IBM Db2 Versi 2.
- EBCDIC_819_037— Halaman kode ISO Latin, pemeriksaan; CCSID 037 (EBCDIC US English).
- EBCDIC_819_500— Halaman kode ISO Latin, pemeriksaan; CCSID 500 (EBCDIC International).
- EBCDIC_850_037— Halaman kode Latin ASCII, pemeriksaan; CCSID 037 (EBCDIC US English).

- `EBCDIC_850_500`— Halaman kode Latin ASCII, pemeriksaan; CCSID 500 (EBCDIC International).
- `EBCDIC_932_5026`— Halaman kode ASCII Jepang, pemeriksaan; CCSID 037 (EBCDIC US English).
- `EBCDIC_932_5035`— Halaman kode ASCII Jepang, pemeriksaan; CCSID 500 (EBCDIC International).
- `EBCDIC_1252_037`— Halaman kode Windows Latin, pemeriksaan; CCSID 037 (EBCDIC US English).
- `EBCDIC_1252_500`— Halaman kode Windows Latin, pemeriksaan; CCSID 500 (EBCDIC International).
- `IDENTITY`— Pemeriksaan default. String dibandingkan byte untuk byte.
- `IDENTITY_16BIT`— Skema Pengkodean Kompatibilitas untuk urutan pemeriksaan UTF-16:8-bit (CESU-8). Untuk informasi selengkapnya, lihat Laporan Teknis Unicode #26 di situs web Unicode Consortium.
- `NLSCHAR`— Hanya untuk digunakan dengan halaman kode Thailand (CP874).
- `SYSTEM`— Jika Anda menggunakan`SYSTEM`, database mengambil urutan pemeriksaan secara otomatis untuk `database_codeset` dan. `database_territory`

Nilai default-nya `IDENTITY`.

Selain itu, RDS untuk Db2 mendukung kelompok pengumpulan berikut: dan. `language-aware-collation` `locale-sensitive-collation` Untuk informasi selengkapnya, lihat Memilih pemeriksaan untuk database Unicode dalam dokumentasi. IBM Db2

`database_autoconfigure_str`

Sintaks `AUTOCONFIGURE` perintah, misalnya, '`AUTOCONFIGURE APPLY DB`'. Tipe datanya `varchar`. Defaultnya adalah string kosong atau null.

Lihat informasi yang lebih lengkap di AUTOCONFIGURE command dalam dokumentasi IBM Db2.

Catatan penggunaan

Anda dapat membuat basis data dengan memanggil `rdsadmin.create_database` jika Anda tidak menentukan nama basis data saat membuat instans basis data RDS for Db2 dengan menggunakan konsol Amazon RDS atau AWS CLI. Untuk informasi selengkapnya, lihat Membuat instans DB.

Pertimbangan-pertimbangan khusus:

- PerintahCREATE DATABASE yang dikirim ke instans Db2 menggunakan opsi RESTRICTIVE.
- RDS untuk Db2 hanya menggunakan. AUTOMATIC STORAGE
- RDS for Db2 menggunakan nilai-nilai default untuk NUMSEGS dan DFT_EXTENT_SZ.
- RDS for Db2 menggunakan enkripsi penyimpanan dan tidak mendukung enkripsi basis data.

Lihat informasi yang lebih lengkap tentang semua pertimbangan ini di CREATE DATABASE command dalam dokumentasi IBM Db2.

Sebelum memanggil rdsadmin.create_database, Anda harus menghubungi basis data rdsadmin. Dalam contoh berikut, ganti *master_username* dan *master_password* dengan informasi instans basis data RDS for Db2 Anda.

```
db2 connect to rdsadmin user master_username using master_password
```

Lihat informasi tentang pemeriksaan status pembuatan basis data di rdsadmin.get_task_status.

Contoh-contoh

Contoh berikut membuat database yang disebut TESTJP dengan kombinasi yang benar dari parameter database_code_set, database_territory, dan database_collation untuk Jepang.

```
db2 "call rdsadmin.create_database('TESTJP', 4096, 'IBM-437', 'JP', 'SYSTEM')"
```

rdsadmin.drop_database

Mengedrop basis data.

Sintaks

```
db2 "call rdsadmin.drop_database('database_name')"
```

Parameter-parameter

Parameter berikut diperlukan:

database_name

Nama basis data yang akan didrop. Tipe datanya varchar.

Catatan penggunaan

Anda dapat menjatuhkan database dengan menelepon `rdsadmin.drop_database` hanya jika kondisi berikut terpenuhi:

- Anda tidak menentukan nama basis data saat membuat instans basis data RDS for Db2 dengan menggunakan konsol Amazon RDS atau AWS CLI. Untuk informasi selengkapnya, lihat Membuat instans DB.
- Anda membuat basis data dengan memanggil prosedur tersimpan the section called "rdsadmin.create_database".
- Anda memulihkan basis data dari citra offline atau cadangan dengan memanggil prosedur tersimpan the section called "rdsadmin.restore_database".

Sebelum memanggil `rdsadmin.drop_database`, Anda harus menghubungi basis data `rdsadmin`. Dalam contoh berikut, ganti *`master_username`* dan *`master_password`* dengan informasi instans basis data RDS for Db2 Anda.

```
db2 connect to rdsadmin user master_username using master_password
```

Lihat informasi tentang pemeriksaan status pengedropan basis data di rdsadmin.get_task_status.

Contoh-contoh

Contoh berikut mengedrop basis data bernama `TESTDB`.

```
db2 "call rdsadmin.drop_database('TESTDB')"
```

Contoh-contoh respons

Jika Anda melewatkan nama database yang salah, maka prosedur yang disimpan mengembalikan contoh respons berikut.

```
SQL0438N Application raised error or warning with diagnostic text: "Cannot
drop database. Database with provided name does not exist". SQLSTATE=99993
```

Jika Anda membuat database menggunakan konsol Amazon RDS atau AWS CLI, maka prosedur yang disimpan mengembalikan contoh respons berikut.

```
Return Status = 0
```

Setelah menerima `Return Status = 0`, panggil prosedur tersimpan the section called "rdsadmin.get_task_status". Respons yang mirip dengan contoh berikut menjelaskan status.

```
1 ERROR DROP_DATABASE RDSDB 2023-10-10-16.33.03.744122 2023-10-10-16.33.30.143797 -
 2023-10-10-16.33.30.098857 Task execution has started.
2023-10-10-16.33.30.143797 Caught exception during executing task id 1, Aborting task.
Reason Dropping database created via rds CreateDBInstance api is not allowed.
Only database created using rdsadmin.create_database can be dropped
```

rdsadmin.update_db_param

Memperbarui parameter-parameter basis data.

Sintaks

```
db2 "call rdsadmin.update_db_param(
    'database_name',
    'parameter_to_modify',
    'changed_value')"
```

Parameter-parameter

Parameter-parameter berikut diperlukan:

database_name

Nama database untuk menjalankan tugas. Tipe datanya `varchar`.

parameter_to_modify

Nama parameter yang akan diubah. Tipe datanya `varchar`. Untuk informasi selengkapnya, lihat Parameter-parameter RDS for Db2.

changed_value

Nilai untuk mengubah nilai parameter. Tipe datanya `varchar`.

Catatan penggunaan

Lihat informasi tentang pemeriksaan status pembaruan parameter basis data di rdsadmin.get_task_status.

Contoh-contoh

Contoh berikut memperbarui parameter `archretrydelay` ke `100` untuk basis data bernama `TESTDB`.

```
db2 "call rdsadmin.update_db_param(
    'TESTDB',
    'archretrydelay',
    '100')";
```

Contoh berikut menunda validasi objek yang dibuat pada basis data bernama `TESTDB` untuk menghindari pemeriksaan dependensi.

```
db2 "call rdsadmin.update_db_param(
    'TESTDB',
    'auto_reval',
    'deferred_force')"
```

rdsadmin.restore_database

Memulihkan basis data.

Sintaks

```
db2 "call rdsadmin.restore_database(
    ?,
    'database_name',
    's3_bucket_name',
    's3_prefix',
    restore_timestamp,
    'backup_type')"
```

Parameter-parameter

Parameter output berikut diperlukan:

?

Pasar parameter yang mengeluarkan pesan kesalahan. Parameter ini hanya menerima?.

Parameter input berikut diperlukan:

database_name

Nama basis data yang akan dipulihkan. Nama ini harus sama dengan nama basis data dalam citra cadangan. Tipe datanya `varchar`.

s3_bucket_name

Nama bucket Amazon S3 tempat cadangan Anda berada. Tipe datanya `varchar`.

s3_prefix

Awalan yang digunakan untuk pencocokan file selama pengunduhan. Tipe datanya `varchar`.

Jika parameter ini kosong, maka semua file di bucket Amazon S3 akan diunduh. Berikut ini adalah contoh awalan.

```
backupfolder/SAMPLE.0.rdsdb.DBPART000.20230615010101
```

restore_timestamp

Stempel waktu citra cadangan basis data. Tipe datanya `varchar`.

Stempel waktu disertakan dalam nama file cadangan. Misalnya, `20230615010101` adalah stempel waktu untuk nama file `SAMPLE.0.rdsdb.DBPART000.20230615010101.001`.

backup_type

Jenis cadangan. Tipe datanya `varchar`. Nilai-nilai yang valid: `OFFLINE`, `ONLINE`.

Gunakan `ONLINE` untuk migrasi waktu henti nyaris nol. Untuk informasi selengkapnya, lihat Migrasi waktu henti nyaris nol untuk basis data Db2 berbasis Linux.

Catatan penggunaan

Anda dapat memulihkan basis data dengan memanggil `rdsadmin.restore_database` jika Anda tidak menentukan nama basis data saat membuat instans basis data RDS for Db2 dengan menggunakan konsol Amazon RDS atau AWS CLI. Untuk informasi selengkapnya, lihat Membuat instans DB.

Sebelum memulihkan database, Anda harus menyediakan ruang penyimpanan untuk RDS Anda untuk instans Db2 DB yang sama dengan atau lebih besar dari jumlah ukuran cadangan Anda dan database Db2 asli pada disk. Saat Anda memulihkan cadangan, Amazon RDS mengekstrak file cadangan pada instans basis data RDS for Db2 Anda.

Setiap file cadangan harus 5 TB atau lebih kecil. Jika file cadangan melebihi 5 TB, Anda harus membagi file cadangan tersebut ke dalam beberapa file yang lebih kecil.

Untuk memastikan bahwa prosedur tersimpan `rdsadmin.restore_database` memulihkan semua file, jangan sertakan akhiran nomor file setelah stempel waktu dalam nama file. Misalnya, *`s3_prefix`* `backupfolder/SAMPLE.0.rdsdb.DBPART000.20230615010101` memulihkan file-file berikut:

```
SAMPLE.0.rdsdb.DBPART000.20230615010101.001
SAMPLE.0.rdsdb.DBPART000.20230615010101.002
SAMPLE.0.rdsdb.DBPART000.20230615010101.003
SAMPLE.0.rdsdb.DBPART000.20230615010101.004
SAMPLE.0.rdsdb.DBPART000.20230615010101.005
```

Lihat informasi tentang pemeriksaan status pemulihan basis data Anda di rdsadmin.get_task_status.

Lihat cara membawa online basis data dan menerapkan log transaksi tambahan setelah memulihkan basis data di rdsadmin.rollforward_database.

Contoh-contoh

Contoh berikut memulihkan cadangan offline dengan satu atau beberapa file yang memiliki *`backupfolder/SAMPLE.0.rdsdb.DBPART000.20230615010101s3_prefix`*.

```
db2 "call rdsadmin.restore_database(
    ?,
    'SAMPLE',
    'myS3bucket',
    'backupfolder/SAMPLE.0.rdsdb.DBPART000.20230615010101',
    20230615010101,
    'OFFLINE')"
```

rdsadmin.rollforward_database

Membawa online basis data dan menerapkan log transaksi tambahan setelah memulihkan basis data dengan memanggil rdsadmin.restore_database.

Sintaks

```
db2 "call rdsadmin.rollforward_database(
    ?,
    'database_name',
```

```
    's3_bucket_name',
    s3_prefix,
    'rollfoward_to_option',
    'complete_rollforward')"
```

Parameter-parameter

Parameter output berikut diperlukan:

?

Penanda parameter yang mengeluarkan pesan kesalahan. Parameter ini hanya menerima?.

Parameter input berikut diperlukan:

database_name

Nama database untuk melakukan operasi pada. Tipe datanya `varchar`.

s3_bucket_name

Nama bucket Amazon S3 tempat cadangan Anda berada. Tipe datanya `varchar`.

s3_prefix

Awalan yang digunakan untuk pencocokan file selama pengunduhan. Tipe datanya `varchar`.

Jika parameter ini kosong, maka semua file di bucket S3 akan diunduh. Contoh berikut adalah contoh awalan.

```
backupfolder/SAMPLE.0.rdsdb.DBPART000.20230615010101
```

Parameter input berikut adalah opsional:

rollforward_to_option

Titik yang ingin Anda tuju untuk pengguliran maju. Tipe datanya `varchar`. Nilai-nilai yang valid: `END_OF_LOGS`, `END_OF_BACKUP`. Nilai default-nya `END OF LOGS`.

complete_rollforward

Menentukan apakah menyelesaikan proses pengguliran maju. Tipe datanya `varchar`. Nilai default-nya `TRUE`.

Jika TRUE, maka setelah selesai, basis data akan online dan dapat diakses. Jika FALSE, maka basis data tetap dalam keadaan ROLL-FORWARD PENDING.

Catatan penggunaan

Setelah memanggil rdsadmin.restore_database, Anda harus memanggil rollforward_database untuk menerapkan log arsip dari bucket S3. Anda juga dapat menggunakan prosedur tersimpan ini untuk memulihkan log transaksi tambahan setelah memanggil rdsadmin.restore_database.

Jika Anda mengatur complete_rollforward ke FALSE, maka basis data Anda dalam keadaan ROLL-FORWARD PENDING dan offline. Untuk membawa database online, Anda harus menelponrdsadmin.complete_rollforward.

Lihat informasi tentang pemeriksaan status pengguliran maju basis data di rdsadmin.get_task_status.

Contoh-contoh

Contoh berikut menggulir maju ke cadangan online basis data dengan log transaksi dan lalu membawa online basis data itu.

```
db2 "call rdsadmin.rollforward_database(
    ?,
    null,
    null,
    'END_OF_LOGS',
    'TRUE')"
```

Contoh berikut bergulir ke cadangan online database tanpa log transaksi, dan kemudian membawa database online.

```
db2 "call rdsadmin.rollforward_database(
    ?,
    'TESTDB',
    'S3Bucket',
    'logsfolder/,
    'END_OF_BACKUP',
    'TRUE')"
```

Contoh berikut bergulir ke cadangan online database dengan log transaksi, dan kemudian tidak membawa database online.

```
db2 "call rdsadmin.rollforward_database(
    ?,
    'TESTDB',
    null,
    'onlinebackup/TESTDB',
    'END_OF_LOGS',
    'FALSE')"
```

Contoh berikut bergulir ke cadangan online database dengan log transaksi tambahan, dan kemudian tidak membawa database online.

```
db2 "call rdsadmin.rollforward_database(
    ?,
    'TESTDB',
    'S3Bucket',
    'logsfolder/S0000155.LOG',
    'END_OF_LOGS',
    'FALSE')"
```

rdsadmin.complete_rollforward

Membawa online basis data dari keadaan `ROLL-FORWARD PENDING`.

Sintaks

```
db2 "call rdsadmin.complete_rollforward(
    ?,
    'database_name')"
```

Parameter-parameter

Parameter output berikut diperlukan:

?

Penanda parameter yang mengeluarkan pesan kesalahan. Parameter ini hanya menerima?.

Parameter input berikut diperlukan:

database_name

Nama basis data yang ingin Anda bawa online. Tipe datanya `varchar`.

Catatan penggunaan

Jika Anda menelepon rdsadmin.rollforward_database dengan `complete_rollforward` set to`FALSE`, maka database Anda dalam `ROLL-FORWARD PENDING` keadaan dan offline. Untuk menyelesaikan proses pengguliran maju dan membawa online basis data, panggil `rdsadmin.complete_rollforward`.

Lihat informasi tentang memeriksa status penyelesaian proses pengguliran maju di rdsadmin.get_task_status.

Contoh-contoh

Contoh berikut membawa online basis data TESTDB.

```
db2 "call rdsadmin.complete_rollfoward(
    ?,
    'TESTDB')"
```

Mengelola ruang tabel

Prosedur-prosedur tersimpan berikut mengelola ruang tabel.

Topik

rdsadmin.create_tablespace

Membuat ruang tabel.

Sintaks

```
db2 "call rdsadmin.create_tablespace(
    'database_name',
    'tablespace_name',
    'buffer_pool_name',
    tablespace_page_size,
    tablespace_initial_size,
    tablespace_increase_size,
    'tablespace_type')"
```

Parameter-parameter

Parameter-parameter berikut diperlukan:

database_name

Nama database untuk membuat tablespace di. Tipe datanya `varchar`.

tablespace_name

Nama tablespace yang akan dibuat. Tipe datanya `varchar`.

Parameter berikut ini bersifat opsional:

buffer_pool_name

Nama kolam buffer untuk menetapkan tablespace. Tipe datanya `varchar`. Default-nya adalah string kosong.

> **Important**
>
> Anda harus sudah memiliki kolam penyangga dengan ukuran halaman yang sama untuk dikaitkan dengan ruang tabel.

tablespace_page_size

Ukuran halaman tablespace dalam byte. Tipe datanya `integer`. Nilai yang valid:`4096,8192,16384,32768`. Defaultnya adalah ukuran halaman yang digunakan saat Anda membuat database dengan menelepon[rdsadmin.create_database](#).

> **Important**
>
> Amazon RDS mendukung atomisitas tulis untuk halaman 4 KiB, 8 KiB, dan 16 KiB. Sebaliknya, 32 halaman KiB berisiko robek, atau sebagian data ditulis ke meja. Jika Anda menggunakan 32 halaman KiB, kami sarankan Anda mengaktifkan point-in-time pemulihan dan pencadangan otomatis. Jika tidak, Anda berisiko tidak dapat pulih dari halaman yang sobek. Lihat informasi yang lebih lengkap di [the section called "Pengantar cadangan"](#) dan [the section called "Memulihkan instans DB dengan waktu yang ditentukan"](#).

tablespace_initial_size

Ukuran awal ruang tabel dalam kilobyte (KB). Tipe datanya `integer`. Nilai yang valid: 48 atau lebih tinggi. Default-nya adalah null.

Jika Anda tidak menetapkan nilai, Db2 menetapkan nilai yang sesuai untuk Anda.

> **Note**
>
> Parameter ini tidak berlaku untuk ruang tabel sementara karena sistem mengelola ruang tabel sementara.

tablespace_increase_size

Persentase yang digunakan untuk menambah ruang tabel apabila penuh. Tipe datanya `integer`. Nilai-nilai yang valid: 1–100. Default-nya adalah null.

Jika Anda tidak menetapkan nilai, Db2 menetapkan nilai yang sesuai untuk Anda.

> **Note**
>
> Parameter ini tidak berlaku untuk ruang tabel sementara karena sistem mengelola ruang tabel sementara.

tablespace_type

Ukuran ruang tabel. Tipe datanya `char`. Nilai yang valid: U (untuk data pengguna) atau T (untuk data sementara). Nilai default-nya U.

Catatan penggunaan

RDS untuk Db2 selalu membuat database besar untuk data.

Lihat informasi tentang pemeriksaan status pembuatan ruang tabel di rdsadmin.get_task_status.

Contoh-contoh

Contoh berikut membuat tablespace dipanggil SP8 dan menetapkan kumpulan buffer dipanggil BP8 untuk database yang dipanggil. TESTDB Tablespace memiliki ukuran halaman tablespace awal 4.096 byte, tablespace awal 1.000 KB, dan peningkatan ukuran tabel disetel ke 50%.

```
db2 "call rdsadmin.create_tablespace(
    'TESTDB',
    'SP8',
    'BP8',
    4096,
    1000,
    50)"
```

Contoh berikut menciptakan tablespace sementara yang disebutSP8. Ini menetapkan kumpulan buffer BP8 yang disebut berukuran 8 KiB untuk database yang disebut. TESTDB

```
db2 "call rdsadmin.create_tablespace(
    'TESTDB',
    'SP8',
    'BP8',
    8192,
    NULL,
    NULL,
    'T')"
```

rdsadmin.alter_tablespace

Mengubah ruang tabel.

Sintaks

```
db2 "call rdsadmin.alter_tablespace(
    'database_name',
    'tablespace_name',
    'buffer_pool_name',
    tablespace_increase_size,
    'max_size',
    'reduce_max',
    'reduce_stop',
    'reduce_value',
    'lower_high_water',
    'lower_high_water_stop',
    'switch_online')"
```

Parameter-parameter

Parameter-parameter berikut diperlukan:

database_name

Nama basis data yang menggunakan ruang tabel. Tipe datanya `varchar`.

tablespace_name

Nama tablespace untuk diubah. Tipe datanya `varchar`.

Parameter berikut ini bersifat opsional:

buffer_pool_name

Nama kolam buffer untuk menetapkan tablespace. Tipe datanya `varchar`. Default-nya adalah string kosong.

> **Important**
>
> Anda harus sudah memiliki kolam penyangga dengan ukuran halaman yang sama untuk dikaitkan dengan ruang tabel.

tablespace_increase_size

Persentase yang digunakan untuk menambah ruang tabel apabila penuh. Tipe datanya `integer`. Nilai-nilai yang valid: `1`–`100`. Nilai default-nya `0`.

max_size

Ukuran maksimum ruang tabel. Tipe datanya `varchar`. Nilai-nilai yang valid: *`integer`* `K` | `M` | `G`, atau `NONE`. Nilai default-nya `NONE`.

reduce_max

Menentukan apakah akan mengurangi tanda air tinggi ke batas maksimumnya. Tipe datanya `char`. Nilai default-nya `N`.

reduce_stop

Menentukan apakah akan menyela perintah `reduce-max` atau `reduce-value` sebelumnya. Tipe datanya `char`. Nilai default-nya `N`.

reduce_value

Jumlah atau persentase yang digunakan untuk mengurangi tanda air tinggi ruang tabel. Tipe datanya `varchar`. Nilai-nilai yang valid: *`integer`* `KiB`| `MiB` | `GiB`, atau `1`–`100`. Default `N`.

lower_high_water

Menentukan apakah akan menjalankan `ALTER TABLESPACE LOWER HIGH WATER MARK` perintah. Tipe datanya `char`. Nilai default-nya `N`.

lower_high_water_stop

Menentukan apakah akan menjalankan `ALTER TABLESPACE LOWER HIGH WATER MARK STOP` perintah. Tipe datanya `char`. Nilai default-nya `N`.

switch_online

Menentukan apakah akan menjalankan `ALTER TABLESPACE SWITCH ONLINE` perintah. Tipe datanya `char`. Nilai default-nya `N`.

Catatan penggunaan

Lihat informasi tentang pemeriksaan status pengubahan ruang tabel di rdsadmin.get_task_status.

Contoh-contoh

Contoh berikut mengubah ruang tabel bernama `SP8` dan menetapkan kolam penyangga bernama `BP8` untuk basis data bernama `TESTDB` untuk menurunkan tanda air tinggi.

```
db2 "call rdsadmin.alter_tablespace(
    'TESTDB',
    'SP8',
    'BP8',
    NULL,
    NULL,
    'Y')"
```

rdsadmin.drop_tablespace

Mengedrop ruang tabel.

Sintaks

```
db2 "call rdsadmin.drop_tablespace(
    'database_name',
    'tablespace_name')"
```

Parameter-parameter

Parameter-parameter berikut diperlukan:

database_name

Nama basis data yang memiliki ruang tabel. Tipe datanya `varchar`.

tablespace_name

Nama tablespace yang akan dijatuhkan. Tipe datanya `varchar`.

Catatan penggunaan

Lihat informasi tentang pemeriksaan status pengedropan ruang tabel di rdsadmin.get_task_status.

Contoh-contoh

Contoh berikut mengedrop ruang tabel bernama SP8 untuk basis data bernama TESTDB.

```
db2 "call rdsadmin.drop_tablespace(
    'TESTDB',
    'SP8')"
```

Mengelola kebijakan audit

Prosedur tersimpan berikut mengelola kebijakan audit untuk RDS untuk database Db2 yang menggunakan audit logging. Untuk informasi selengkapnya, lihat the section called “Pencatatan audit Db2”.

Topik

rdsadmin.configure_db_audit

Mengkonfigurasi kebijakan audit untuk RDS untuk database Db2 yang ditentukan oleh db_name. Jika kebijakan yang Anda konfigurasikan tidak ada, memanggil prosedur tersimpan ini akan membuatnya. Jika kebijakan ini memang ada, memanggil prosedur tersimpan ini akan mengubahnya dengan nilai parameter yang Anda berikan.

Sintaksis

```
db2 "call rdsadmin.configure_db_audit(
    'db_name',
    'category',
    'category_setting',
    '?')"
```

Parameter-parameter

Parameter-parameter berikut diperlukan.

db_nama

Nama DB dari database RDS untuk Db2 untuk mengonfigurasi kebijakan audit untuk. Tipe datanya `varchar`.

kategori

Nama kategori untuk mengonfigurasi kebijakan audit ini. Tipe datanya `varchar`. Berikut ini adalah nilai yang valid untuk parameter ini:

- `ALL`Dengan`ALL`, Amazon RDS tidak termasuk`CONTEXT`,`EXECUTE`, atau `ERROR` kategori.

- AUDIT
- CHECKING
- CONTEXT
- ERROR
- EXECUTE— Anda dapat mengkonfigurasi kategori ini dengan data atau tanpa data. Dengan data berarti juga mencatat nilai data input yang disediakan untuk variabel host dan penanda parameter apa pun. Defaultnya adalah tanpa data. Untuk informasi selengkapnya, lihat deskripsi parameter *category_setting* dan parameter. the section called “Contoh-contoh”
- OBJMAINT
- SECMAINT
- SYSADMIN
- VALIDATE

Untuk informasi selengkapnya tentang kategori ini, lihat IBM Db2dokumentasi.

category_setting

Pengaturan untuk kategori audit yang ditentukan. Tipe datanya varchar.

Tabel berikut menunjukkan nilai pengaturan kategori yang valid untuk setiap kategori.

Kategori	Pengaturan kategori yang valid
ALL	BOTH|FAILURE|SUCCESS|NONE
AUDIT	
CHECKING	
CONTEXT	
OBJMAINT	
SECMAINT	
SYSADMIN	
VALIDATE	

Kategori	Pengaturan kategori yang valid
ERROR	AUDIT|NORMAL . Defaultnya adalahNORMAL.
EXECUTE	BOTH,WITH|BOTH,WITHOUT|FAILURE,WITH|FAILURE,WITHOUT|SUCCESS,WITH|SUCCESS,WITHOUT|NONE

Catatan penggunaan

Sebelum Anda meneleponrdsadmin.configure_db_audit, pastikan instans RDS untuk Db2 DB dengan database tempat Anda mengonfigurasi kebijakan audit dikaitkan dengan grup opsi yang memiliki opsi. DB2_AUDIT Untuk informasi selengkapnya, lihat the section called “Menyiapkan pencatatan audit Db2”.

Setelah mengonfigurasi kebijakan audit, Anda dapat memeriksa status konfigurasi audit untuk database dengan mengikuti langkah-langkahnyaPeriksa konfigurasi audit.

ALLMenentukan category parameter tidak termasukCONTEXT,EXECUTE, atau ERROR kategori. Untuk menambahkan kategori ini ke kebijakan audit Anda, hubungi rdsadmin.configure_db_audit secara terpisah dengan setiap kategori yang ingin Anda tambahkan. Untuk informasi selengkapnya, lihat the section called “Contoh-contoh”.

Contoh-contoh

Contoh berikut membuat atau memodifikasi kebijakan audit untuk database bernamaTESTDB. Dalam contoh 1 hingga 5, jika ERROR kategori sebelumnya tidak dikonfigurasi, kategori ini disetel ke NORMAL (default). Untuk mengubah pengaturan ituAUDIT, ikutiExample 6: Specifying the ERROR category.

Contoh 1: Menentukan kategori **ALL**

```
db2 "call rdsadmin.configure_db_audit('TESTDB', 'ALL', 'BOTH', ?)"
```

Dalam contoh, panggilan mengkonfigurasiAUDIT,,CHECKING,OBJMAINT, SECMAINTSYSADMIN, dan VALIDATE kategori dalam kebijakan audit. Menentukan BOTH berarti bahwa peristiwa yang berhasil dan gagal akan diaudit untuk masing-masing kategori ini.

Contoh 2: Menentukan **EXECUTE** kategori dengan data

```
db2 "call rdsadmin.configure_db_audit('TESTDB', 'EXECUTE', 'SUCCESS,WITH', ?)"
```

Dalam contoh, panggilan mengonfigurasi `EXECUTE` kategori dalam kebijakan audit. Menentukan `SUCCESS,WITH` berarti bahwa log untuk kategori ini hanya akan mencakup peristiwa yang berhasil, dan akan menyertakan nilai data input yang disediakan untuk variabel host dan penanda parameter.

Contoh 3: Menentukan **`EXECUTE`** kategori tanpa data

```
db2 "call rdsadmin.configure_db_audit('TESTDB', 'EXECUTE', 'FAILURE,WITHOUT', ?)"
```

Dalam contoh, panggilan mengonfigurasi `EXECUTE` kategori dalam kebijakan audit. Menentukan `FAILURE,WITHOUT` berarti bahwa log untuk kategori ini hanya akan mencakup peristiwa yang gagal, dan tidak akan menyertakan nilai data input yang disediakan untuk variabel host dan penanda parameter.

Contoh 4: Menentukan **`EXECUTE`** kategori tanpa acara status

```
db2 "call rdsadmin.configure_db_audit('TESTDB', 'EXECUTE', 'NONE', ?)"
```

Dalam contoh, panggilan mengonfigurasi `EXECUTE` kategori dalam kebijakan audit. Menentukan `NONE` berarti tidak ada acara dalam kategori ini yang akan diaudit.

Contoh 5: Menentukan kategori **`OBJMAINT`**

```
db2 "call rdsadmin.configure_db_audit('TESTDB', 'OBJMAINT', 'NONE', ?)"
```

Dalam contoh, panggilan mengonfigurasi `OBJMAINT` kategori dalam kebijakan audit. Menentukan `NONE` berarti tidak ada acara dalam kategori ini yang akan diaudit.

Contoh 6: Menentukan kategori **`ERROR`**

```
db2 "call rdsadmin.configure_db_audit('TESTDB', 'ERROR', 'AUDIT', ?)"
```

Dalam contoh, panggilan mengonfigurasi `ERROR` kategori dalam kebijakan audit. Menentukan `AUDIT` berarti bahwa semua kesalahan, termasuk kesalahan yang terjadi dalam pencatatan audit itu sendiri, ditangkap dalam log. Jenis kesalahan default adalah`NORMAL`. Dengan`NORMAL`, kesalahan yang dihasilkan oleh audit diabaikan dan hanya `SQLCODE` s untuk kesalahan yang terkait dengan operasi yang dilakukan ditangkap.

rdsadmin.disable_db_audit

Menghentikan pencatatan audit untuk database RDS untuk Db2 yang ditentukan oleh *db_name* dan menghapus kebijakan audit yang dikonfigurasi untuknya.

> **Note**
>
> Prosedur tersimpan ini hanya menghapus kebijakan audit yang dikonfigurasi dengan meneleponthe section called “rdsadmin.configure_db_audit”.

Sintaksis

```
db2 "call rdsadmin.disable_db_audit('db_name')"
```

Parameter-parameter

Parameter-parameter berikut diperlukan.

db_nama

Nama DB dari database RDS untuk Db2 untuk menonaktifkan audit logging untuk. Tipe datanya `varchar`.

Catatan penggunaan

Panggilan `rdsadmin.disable_db_audit` tidak menonaktifkan pencatatan audit untuk RDS untuk instans Db2 DB. Untuk menonaktifkan pencatatan audit pada tingkat instans DB, hapus grup opsi dari instans DB. Untuk informasi selengkapnya, lihat Menonaktifkan pencatatan audit Db2.

Contoh-contoh

Contoh berikut menonaktifkan audit logging untuk database bernama`TESTDB`.

```
db2 "call rdsadmin.disable_db_audit('TESTDB')"
```

Referensi fungsi buatan pengguna RDS for Db2

Topik-topik ini menjelaskan fungsi buatan pengguna yang tersedia untuk instans Amazon RDS yang menjalankan mesin RDS for Db2.

Topik

Memeriksa status tugas

Anda dapat menggunakan fungsi `rdsadmin.get_task_status` buatan pengguna untuk memeriksa status tugas-tugas berikut. Daftar ini tidak lengkap.

- Membuat, mengubah, atau mengedrop kolam penyangga
- Membuat, mengubah, atau mengedrop ruang tabel
- Membuat atau mengedrop basis data
- Memulihkan cadangan basis data dari Amazon S3
- Menggulir maju log basis data dari Amazon S3

rdsadmin.get_task_status

Menghasilkan status tugas.

Sintaks

```
db2 "select task_id, task_type, database_name, lifecycle,
    varchar(bson_to_json(task_input_params), 500) as task_params,
    cast(task_output as varchar(500)) as task_output
    from table(rdsadmin.get_task_status(task_id,'database_name','task_type'))"
```

Parameter-parameter

Parameter-parameter berikut bersifat opsional. Jika Anda tidak memberikan parameter sama sekali, fungsi buatan pengguna ini akan menghasilkan status semua tugas untuk semua basis data. Amazon RDS mempertahankan riwayat tugas selama 35 hari.

`task_id`

ID tugas yang sedang dijalankan. ID ini dihasilkan saat Anda menjalankan sebuah tugas. Default: `0`.

`database_name`

Nama basis data yang menjalankan tugas.

`task_type`

Jenis tugas untuk dikueri. Nilai yang valid: `ADD_GROUPSADD_USER,ALTER_BUFFERPOOL,ALTER_TABLESPACE,CHANGE_PASSWORD,COMPLETE_RO`

Contoh-contoh

Contoh berikut menampilkan kolom-kolom yang dihasilkan saat `rdsadmin.get_task_status` dipanggil.

```
db2 "describe select * from table(rdsadmin.get_task_status())"
```

Contoh berikut memerinci status semua tugas.

```
db2 "select task_id, task_type, database_name, lifecycle,
    varchar(bson_to_json(task_input_params), 500) as task_params,
    cast(task_output as varchar(500)) as task_output
    from table(rdsadmin.get_task_status(null,null,null))"
```

Contoh berikut memerinci status tugas tertentu.

```
db2 "select task_id, task_type, database_name,
    varchar(bson_to_json(task_input_params), 500) as task_params
    from table(rdsadmin.get_task_status(1,null,null))"
```

Contoh berikut memerinci status tugas dan basis data tertentu.

```
db2 "select task_id, task_type, database_name,
    varchar(bson_to_json(task_input_params), 500) as task_params
    from table(rdsadmin.get_task_status(2,'SAMPLE',null))"
```

Contoh berikut memerinci status semua tugas `ADD_GROUPS`.

```
db2 "select task_id, task_type, database_name,
    varchar(bson_to_json(task_input_params), 500) as task_params
    from table(rdsadmin.get_task_status(null,null,'add_groups'))"
```

Contoh berikut memerinci status semua tugas untuk basis data tertentu.

```
db2 "select task_id, task_type, database_name,
    varchar(bson_to_json(task_input_params), 500) as task_params
    from table(rdsadmin.get_task_status(null,'testdb', null))"
```

Contoh berikut menghasilkan nilai-nilai JSON berupa kolom.

```
db2 "select varchar(r.task_type,25) as task_type, varchar(r.lifecycle,10) as lifecycle,
 r.created_at, u.* from
    table(rdsadmin.get_task_status(null,null,'restore_db')) as r,
 json_table(r.task_input_params, 'strict $' columns(s3_prefix varchar(500)
    null on empty, s3_bucket_name varchar(500) null on empty) error on error ) as U"
```

Respons

Fungsi buatan pengguna `rdsadmin.get_task_status` menghasilkan kolom-kolom berikut:

`TASK_ID`

ID tugas.

`TASK_TYPE`

Bergantung pada parameter-parameter input.

- `ADD_GROUPS` – Menambahkan grup.
- `ADD_USER` – Menambahkan pengguna.
- `ALTER_BUFFERPOOL` – Mengubah kolam penyangga.
- `ALTER_TABLESPACE` – Mengubah ruang tabel.
- `CHANGE_PASSWORD` – Mengubah kata sandi pengguna.
- `COMPLETE_ROLLFORWARD` – Menyelesaikan tugas `rdsadmin.rollforward_database` dan mengaktifkan basis data.
- `CREATE_BUFFERPOOL` – Membuat kolam penyangga.
- `CREATE_DATABASE` – Membuat basis data.
- `CREATE_ROLE` – Membuat peran Db2 untuk pengguna.
- `CREATE_TABLESPACE` – Membuat ruang tabel.
- `DROP_BUFFERPOOL` – Mengedrop kolam penyangga.
- `DROP_DATABASE` – Menghapus basis data.
- `DROP_TABLESPACE` – Mengedrop ruang tabel.
- `LIST_USERS` – Memerinci semua pengguna.
- `REMOVE_GROUPS` – Menghapus grup.
- `REMOVE_USER` – Menghapus pengguna.
- `RESTORE_DB` – Memulihkan basis data lengkap.

- ROLLFORWARD_DB_LOG – Melakukan tugas `rdsadmin.rollforward_database` pada log basis data.
- ROLLFORWARD_STATUS – Menghasilkan status tugas `rdsadmin.rollforward_database`.
- UPDATE_DB_PARAM – Memperbarui parameter data.

DATABASE_NAME

Nama basis data yang terkait dengan tugas.

COMPLETED_WORK_BYTES

Jumlah byte yang dipulihkan oleh tugas.

DURATION_MINS

Waktu yang dibutuhkan untuk menyelesaikan tugas.

LIFECYCLE

Status tugas. Status yang mungkin:

- CREATED – Setelah tugas diserahkan ke Amazon RDS, Amazon RDS menetapkan statusnya ke CREATED.
- IN_PROGRESS – Setelah tugas dimulai, Amazon RDS menetapkan statusnya ke IN_PROGRESS. Diperlukan hingga 5 menit agar status berubah dari CREATED ke IN_PROGRESS.
- SUCCESS – Setelah tugas selesai, Amazon RDS menetapkan statusnya ke SUCCESS.
- ERROR – Jika tugas pemulihan gagal, Amazon RDS menetapkan statusnya ke ERROR. Lihat informasi yang lebih lengkap tentang kesalahan ini di TASK_OUPUT.

CREATED_BY

authid yang membuat perintah.

CREATED_AT

Tanggal dan waktu ketika tugas dibuat.

LAST_UPDATED_AT

Tanggal dan waktu ketika tugas terakhir diperbarui.

TASK_INPUT_PARAMS

Parameter-parameter berbeda-beda berdasarkan jenis tugas. Semua parameter input disajikan berupa objek JSON. Misalnya, kunci JSON untuk tugas RESTORE_DB adalah sebagai berikut:

- DBNAME
- RESTORE_TIMESTAMP
- S3_BUCKET_NAME
- S3_PREFIX

TASK_OUTPUT

Informasi tambahan tentang tugas. Jika terjadi kesalahan selama pemulihan asli, kolom ini mencakup informasi tentang kesalahan itu.

Contoh respons

Contoh respons berikut menunjukkan bahwa basis data bernama TESTJP berhasil dibuat. Lihat informasi yang lebih lengkap di prosedur tersimpan the section called “rdsadmin.create_database”.

```
`1 SUCCESS CREATE_DATABASE RDSDB 2023-10-24-18.32.44.962689 2023-10-24-18.34.50.038523
 1 TESTJP { "CODESET" : "IBM-437", "TERRITORY" : "JP", "COLLATION" : "SYSTEM",
 "AUTOCONFIGURE_CMD" : "", "PAGESIZE" : 4096 }
2023-10-24-18.33.30.079048 Task execution has started.

2023-10-24-18.34.50.038523 Task execution has completed successfully`.
```

Contoh respons berikut menjelaskan alasan gagalnya pengedropan basis data. Lihat informasi yang lebih lengkap di prosedur tersimpan the section called “rdsadmin.drop_database”.

```
1 ERROR DROP_DATABASE RDSDB 2023-10-10-16.33.03.744122 2023-10-10-16.33.30.143797 -
 2023-10-10-16.33.30.098857 Task execution has started.
2023-10-10-16.33.30.143797 Caught exception during executing task id 1, Aborting task.
Reason Dropping database created via rds CreateDBInstance api is not allowed.
Only database created using rdsadmin.create_database can be dropped
```

Contoh respon berikut menunjukkan keberhasilan pemulihan database. Lihat informasi yang lebih lengkap di prosedur tersimpan the section called “rdsadmin.restore_database”.

```
1 RESTORE_DB  SAMPLE  SUCCESS

{ "S3_BUCKET_NAME" : "mybucket", "S3_PREFIX" :
 "SAMPLE.0.rdsdb3.DBPART000.20230413183211.001", "RESTORE_TIMESTAMP" :
 "20230413183211", "BACKUP_TYPE" : "offline" }
```

```
2023-11-06-18.31.03.115795 Task execution has started.
2023-11-06-18.31.04.300231 Preparing to download
2023-11-06-18.31.08.368827 Download complete. Starting Restore
2023-11-06-18.33.13.891356 Task Completed Successfully
```

Amazon RDS for MariaDB

Amazon RDS mendukung instans DB yang menjalankan versi MariaDB berikut:

- MariaDB 10.11
- MariaDB 10.6
- MariaDB 10.5
- MariaDB 10.4
- MariaDB 10.3 (akhir dukungan standar RDS dijadwalkan pada 23 Oktober 2023)

Untuk informasi selengkapnya tentang dukungan versi kecil, lihat Versi-versi MariaDB pada Amazon RDS.

Untuk membuat instans DB MariaDB, gunakan alat manajemen atau antarmuka Amazon RDS. Anda kemudian dapat menggunakan alat Amazon RDS untuk melakukan tindakan manajemen untuk instans DB. Tindakan tersebut termasuk seperti berikut:

- Mengonfigurasi ulang atau mengubah ukuran instans DB
- Mengizinkan koneksi ke instans DB
- Membuat dan memulihkan dari cadangan atau snapshot
- Membuat Sekunder Multi-AZ
- Membuat replika baca
- Memantau performa instans DB Anda

Untuk menyimpan dan mengakses data dalam instans DB Anda, gunakan utilitas dan aplikasi MariaDB standar.

MariaDB tersedia di semua Wilayah AWS. Untuk informasi selengkapnya tentang Wilayah AWS, lihat Wilayah, Zona Ketersediaan, dan Zona Lokal.

Anda dapat menggunakan basis data Amazon RDS for MariaDB untuk membangun aplikasi yang mematuhi HIPAA. Anda dapat menyimpan informasi terkait perawatan kesehatan, termasuk informasi kesehatan yang dilindungi (PHI), berdasarkan Perjanjian Rekan Bisnis (BAA) dengan AWS. Untuk informasi selengkapnya, lihat Kepatuhan HIPAA. AWS Layanan dalam Cakupan telah dinilai sepenuhnya oleh auditor pihak ketiga dan menghasilkan sertifikasi, pengesahan kepatuhan, atau

Otoritas untuk Beroperasi (ATO). Untuk informasi selengkapnya, lihat layanan AWS dalam cakupan berdasarkan program kepatuhan.

Sebelum membuat instans DB, selesaikan langkah-langkah di Menyiapkan Amazon RDS. Saat Anda membuat instans DB, pengguna master RDS mendapatkan hak istimewa DBA, dengan beberapa batasan. Gunakan akun ini untuk tugas administratif seperti membuat akun basis data tambahan.

Anda dapat membuat berikut ini:

- Instans DB
- Snapshot DB
- Pemulihan titik waktu
- Pencadangan otomatis
- Pencadangan manual

Anda dapat menggunakan instans DB yang menjalankan MariaDB di dalam cloud privat virtual (VPC) berdasarkan Amazon VPC. Anda juga dapat menambahkan fitur ke instans DB MariaDB Anda dengan mengaktifkan berbagai opsi. Amazon RDS mendukung deployment multi-AZ untuk MariaDB sebagai solusi failover dengan ketersediaan tinggi.

> **Important**
>
> Untuk memberikan pengalaman layanan terkelola, Amazon RDS tidak memberikan akses shell ke instans DB. Hal tersebut juga membatasi akses ke prosedur dan tabel sistem tertentu yang membutuhkan hak istimewa tingkat lanjut. Anda dapat mengakses basis data Anda menggunakan klien SQL standar seperti klien mysql. Namun, Anda tidak dapat mengakses host secara langsung dengan menggunakan Telnet atau Secure Shell (SSH).

Topik

- Dukungan fitur MariaDB di Amazon RDS
- Versi-versi MariaDB pada Amazon RDS
- Menghubungkan ke instans DB yang menjalankan mesin basis data MariaDB
- Mengamankan koneksi instans DB MariaDB
- Meningkatkan performa kueri RDS for MariaDB dengan Amazon RDS Optimized Reads
- Meningkatkan performa penulisan dengan Amazon RDS Optimized Writes for MariaDB

- Meningkatkan mesin DB MariaDB
- Mengimpor data ke instans basis data MariaDB
- Menggunakan replikasi MariaDB di Amazon RDS
- Opsi untuk mesin basis data MariaDB
- Parameter untuk MariaDB
- Memigrasikan data dari snapshot DB MySQL ke instans DB MariaDB
- MariaDB di referensi Amazon RDS SQL
- Zona waktu lokal untuk instans basis data MariaDB
- Masalah umum dan batasan untuk RDS for MariaDB

Dukungan fitur MariaDB di Amazon RDS

RDS for MariaDB mendukung sebagian besar fitur dan kemampuan MariaDB. Beberapa fitur mungkin memiliki dukungan terbatas atau privilese yang dibatasi.

Anda dapat memfilter fitur-fitur Amazon RDS baru pada halaman Apa yang Baru dengan Basis Data?. Untuk Produk, pilih Amazon RDS. Lalu, cari dengan menggunakan kata kunci seperti **`MariaDB 2023`**.

> **Note**
>
> Daftar-daftar berikut tidak lengkap.

Topik

- Dukungan fitur MariaDB di versi-versi utama Amazon RDS for MariaDB
- Mesin penyimpanan yang didukung untuk MariaDB di Amazon RDS
- Penghangatan cache untuk MariaDB di Amazon RDS
- Fitur-fitur MariaDB yang didukung oleh Amazon RDS

Dukungan fitur MariaDB di versi-versi utama Amazon RDS for MariaDB

Di bagian-bagian berikut, temukan informasi tentang dukungan fitur MariaDB pada versi-versi utama Amazon RDS for MariaDB:

Topik

Lihat informasi tentang versi-versi kecil Amazon RDS for MariaDB yang didukung di Versi-versi MariaDB pada Amazon RDS.

Dukungan MariaDB 10.11 di Amazon RDS

Amazon RDS mendukung fitur-fitur baru berikut untuk instans basis data Anda yang menjalankan MariaDB versi 10.11 atau lebih tinggi.

- Pengaya Pemeriksaan Pengulangan Penggunaan Kata Sandi – Anda dapat menggunakan pengaya Pemeriksaan Pengulangan Penggunaan Kata Sandi MariaDB untuk mencegah pengguna mengulang penggunaan kata sandi dan untuk mengatur periode retensi kata sandi. Lihat informasi yang lebih lengkap di Pengaya Pemeriksaan Pengulangan Penggunaan Kata Sandi.
- Otorisasi GRANT TO PUBLIC – Anda dapat memberikan privilese kepada semua pengguna yang memiliki akses ke server Anda. Lihat informasi yang lebih lengkap di GRANT TO PUBLIC.
- Pemisahan privilese-privilese SUPER dan ADMIN HANYA BACA – Anda dapat menghapus privilese ADMIN HANYA BACA dari semua pengguna, bahkan dari pengguna yang pernah memiliki privilese SUPER.
- Keamanan – Anda kini dapat mengatur opsi `--ssl` sebagai bawaan untuk klien MariaDB Anda. MariaDB tidak lagi menonaktifkan SSL secara diam-diam jika konfigurasinya salah.
- Perintah dan fungsi SQL – Anda kini dapat menggunakan perintah `SHOW ANALYZE FORMAT=JSON` dan fungsi `ROW_NUMBER`, `SFORMAT`, dan`RANDOM_BYTES`. `SFORMAT` memungkinkan pemformatan string dan diaktifkan secara bawaan. Anda dapat mengonversi partisi ke tabel dan tabel ke partisi dalam satu perintah. Ada juga beberapa perbaikan seputar fungsi-fungsi `JSON_*()`. Fungsi-fungsi `DES_ENCRYPT` dan `DES_DECRYPT` diusangkan untuk versi 10.10 dan yang lebih tinggi. Lihat informasi yang lebih lengkap di SFORMAT.
- Penyempurnaan InnoDB - Penyempurnaan ini mencakup butir-butir berikut:
  - Peningkatan kinerja dalam log redo untuk mengurangi amplifikasi tulis dan meningkatkan konkurensi.

  - Kemampuan bagi Anda untuk mengubah ruang tabel undo tanpa menginisialisasikan ulang direktori data. Peningkatan ini mengurangi sisihan umum/overhead bidang kontrol. Mesin ini membutuhkan pemulaian ulang, tetapi tidak memerlukan inisialisasi ulang setelah mengubah ruang tabel undo.
  - Dukungan untuk `CHECK TABLE … EXTENDED` dan untuk indeks menurun secara internal.
  - Perbaikan untuk penyisipan massal.
- Perubahan Binlog – Perubahan ini mencakup butir-butir berikut:
  - Mengelog `ALTER` dalam dua fase untuk mengurangi latensi replikasi. Parameter `binlog_alter_two_phase` dinonaktifkan secara bawaan, tetapi dapat diaktifkan melalui grup parameter.
  - Mengelog `explicit_defaults_for_timestamp`.
  - Tidak lagi mengelog `INCIDENT_EVENT` jika transaksi dapat digulirkan balik dengan selamat.
- Peningkatan replikasi – Instans basis data MariaDB versi 10.11 menggunakan replikasi GTID secara bawaan jika master mendukungnya. Juga, `Seconds_Behind_Master` lebih cermat.
- Klien – Anda dapat menggunakan opsi-opsi baris perintah baru untuk `mysqlbinglog` dan `mariadb-dump`. Anda dapat menggunakan `mariadb-dump` untuk membuang dan memulihkan data historis.
- Tata versi/versioning sistem – Anda dapat mengubah riwayat. MariaDB membuat secara otomatis partisi baru.
- DDL Atomis – `CREATE OR REPLACE` kini atomis. Entah pernyataan itu berhasil atau benar-benar dibalik.
- Penulisan log redo – Log redo menulis secara asinkron.
- Fungsi tersimpan — Fungsi tersimpan kini mendukung parameter-parameter `IN`, `OUT`, dan `INOUT` yang sama dengan yang di dalam prosedur tersimpan.
- Parameter diusangkan atau dihapus – Parameter-parameter berikut diusangkan atau dihapus untuk instans basis data MariaDB versi 10.11:
  - innodb_change_buffering
  - innodb_disallow_writes
  - innodb_log_write_ahead_size
  - innodb_prefix_index_cluster_optimization
  - keep_files_on_create
  - old

- Parameter dinamis – Parameter-parameter berikut kini dinamis untuk instans basis data MariaDB versi 10.11:
  - innodb_log_file_size
  - innodb_write_io_threads
  - innodb_read_io_threads
- Nilai bawaan baru untuk parameter – Parameter-parameter berikut memiliki nilai bawaan baru untuk instans basis data MariaDB versi 10.11:
  - Nilai bawaan parameter explicit_defaults_for_timestamp berubah dari `OFF` ke `ON`.
  - Nilai bawaan parameter optimizer_prune_level berubah dari 1 ke 2.
- Nilai valid baru untuk parameter – Parameter-parameter berikut memiliki nilai valid baru untuk instans basis data MariaDB versi 10.11:
  - Nilai-nilai yang valid untuk parameter old digabungkan ke dalam nilai-nilai untuk parameter old_mode.
  - Nilai-nilai yang valid untuk parameter histogram_type kini meliputi `JSON_HB`.
  - Rentang nilai yang valid untuk parameter innodb_log_buffer_size kini `262144` hingga `4294967295` (256 KB hingga 4096 MB).
  - Rentang nilai yang valid untuk parameter innodb_log_file_size kini `4194304` hingga `512GB` (4 MB hingga 512 GB).
  - Nilai-nilai yang valid untuk parameter optimizer_prune_level kini meliputi 2.
- Parameter baru – Parameter-parameter berikut adalah baru untuk instans basis data MariaDB versi 10.11:
  - Parameter binlog_alter_two_phase dapat meningkatkan kinerja replikasi.
  - Parameter log_slow_min_examined_row_limit dapat meningkatkan kinerja.
  - Parameter log_slow_query dan parameter log_slow_query_file masing-masing adalah alias untuk `slow_query_log` dan `slow_query_log_file`.
  - optimizer_extra_pruning_depth
  - system_versioning_insert_history

Lihat daftar semua fitur dan dokumentasi dalam informasi berikut di situs web MariaDB.

Versi	Perubahan dan peningkatan	Catatan perilisan
MariaDB 10.7	Perubahan dan peningkatan di MariaDB 10.7	Catatan rilis - MariaDB seri 10.7
MariaDB 10.8	Perubahan dan peningkatan di MariaDB 10.8	Catatan rilis - MariaDB seri 10.8
MariaDB 10.9	Perubahan dan peningkatan di MariaDB 10.9	Catatan rilis - MariaDB seri 10.9
MariaDB 10.10	Perubahan dan peningkatan di MariaDB 10.10	Catatan rilis - MariaDB seri 10.10
MariaDB 10.11	Perubahan dan peningkatan di MariaDB 10.11	Catatan rilis - MariaDB seri 10.11

Lihat daftar fitur yang tidak didukung di Fitur-fitur MariaDB yang didukung oleh Amazon RDS.

Dukungan MariaDB 10.6 di Amazon RDS

Amazon RDS mendukung fitur-fitur baru berikut untuk instans basis data Anda yang menjalankan MariaDB versi 10.6 atau lebih tinggi:

- Mesin penyimpanan MyRocks – Anda dapat menggunakan mesin penyimpanan MyRocks dengan RDS for MariaDB untuk mengoptimalkan konsumsi penyimpanan aplikasi web kinerja tinggi intensif tulis. Lihat informasi yang lebih lengkap di Mesin penyimpanan yang didukung untuk MariaDB di Amazon RDS dan MyRocks.
- Autentikasi basis data AWS Identity and Access Management (IAM) – Anda dapat menggunakan autentikasi basis data IAM untuk keamanan yang lebih kuat dan pengelolaan terpusat terhadap koneksi dengan instans basis data MariaDB Anda. Lihat informasi yang lebih lengkap di Autentikasi basis data IAM untuk MariaDB, MySQL, dan PostgreSQL.
- Opsi pemutakhiran – Anda kini dapat memutakhirkan ke RDS for MariaDB versi 10.6 dari sebarang rilis utama sebelumnya (10.3, 10.4, 10.5). Anda juga dapat memulihkan cuplikan dari instans basis data MySQL 5.6 atau 5.7 yang ada ke instans MariaDB 10.6. Lihat informasi yang lebih lengkap di Meningkatkan mesin DB MariaDB.

- Replikasi tertunda – Anda kini dapat mengatur periode waktu yang dapat dikonfigurasikan bagi replika baca untuk tertinggal di belakang basis data sumber. Dalam konfigurasi replikasi MariaDB standar, ada penundaan replikasi minimal antara sumber dan replika. Dengan replikasi tertunda, Anda dapat menetapkan penundaan disengaja sebagai strategi untuk pemulihan bencana. Lihat informasi yang lebih lengkap di Mengonfigurasi replikasi tertunda dengan MariaDB.
- Kompatibilitas Oracle PL/SQL – Dengan menggunakan RDS for MariaDB versi 10.6, Anda dapat memigrasikan dengan lebih mudah aplikasi Oracle lama Anda ke Amazon RDS. Lihat informasi yang lebih lengkap di SQL_MODE=ORACLE.
- DDL Atomis - Pernyataan-pernyataan bahasa data dinamis (DDL) Anda dapat selamat secara relatif dari kemacetan dengan RDS for MariaDB versi 10.6. `CREATE TABLE`,`ALTER TABLE`,`RENAME TABLE`,`DROP TABLE`, `DROP DATABASE` dan pernyataan-pernyataan DDL terkait kini atomis. Entah pernyataan itu berhasil, atau benar-benar dibalik. Lihat informasi yang lebih lengkap di DDL Atomis.
- Penyempurnaan lain – Penyempurnaan ini mencakup fungsi `JSON_TABLE` untuk mengubah data JSON ke format relasional dalam SQL, dan pemuatan data tabel kosong yang lebih cepat dengan Innodb. Penyempurnaan juga meliputi `sys_schema` baru untuk analisis dan pemecahan masalah, penyempurnaan pengoptimal untuk mengabaikan indeks yang tidak terpakai, dan peningkatan kinerja. Lihat informasi yang lebih lengkap di JSON_TABLE.
- Nilai bawaan baru untuk parameter – Parameter-parameter berikut memiliki nilai bawaan baru untuk instans basis data MariaDB versi 10.6:
  - Nilai bawaan untuk parameter-parameter berikut telah berubah dari `utf8` ke `utf8mb3`:
    - character_set_client
    - character_set_connection
    - character_set_results
    - character_set_system

    Meskipun nilai-nilai bawaan telah berubah untuk semua parameter ini, tidak ada perubahan fungsional. Lihat informasi yang lebih lengkap di Set Karakter dan Kolasi yang Didukung dalam dokumentasi MariaDB.
  - Nilai bawaan parameter collation_connection telah berubah dari `utf8_general_ci` ke `utf8mb3_general_ci`. Meskipun nilai bawaan telah berubah untuk parameter ini, tidak ada perubahan fungsional.
  - Nilai bawaan parameter old_mode telah berubah dari unset ke `UTF8_IS_UTF8MB3`. Meskipun nilai bawaan telah berubah untuk parameter ini, tidak ada perubahan fungsional.

Lihat daftar semua fitur MariaDB 10.6 dan dokumentasinya di Perubahan dan peningkatan dalam MariaDB 10.6 dan Catatan rilis - seri MariaDB 10.6 di situs web MariaDB.

Lihat daftar fitur yang tidak didukung di Fitur-fitur MariaDB yang didukung oleh Amazon RDS.

Dukungan MariaDB 10.5 di Amazon RDS

Amazon RDS mendukung fitur-fitur baru berikut untuk instans basis data Anda yang menjalankan MariaDB versi 10.5 atau lebih baru:

- Penyempurnaan InnoDB – MariaDB versi 10.5 menyertakan penyempurnaan InnoDB. Lihat informasi yang lebih lengkap di InnoDB: Peningkatan Kinerja, dll. dalam dokumentasi MariaDB.
- Pembaruan skema kinerja – MariaDB versi 10.5 mencakup pembaruan skema kinerja. Lihat informasi yang lebih lengkap di Pembaruan Skema Kinerja Agar Sesuai Dengan Instrumentasi dan Tabel MySQL 5.7 dalam dokumentasi MariaDB.
- Satu file dalam log redo InnoDB – Dalam versi MariaDB sebelum versi 10.5, nilai parameter `innodb_log_files_in_group` telah ditetapkan ke 2. Pada MariaDB versi 10.5, nilai parameter ini diatur ke 1.

  Jika Anda memutakhirkan dari versi lebih lama ke MariaDB versi 10.5, dan Anda tidak mengubah parameter, nilai parameter `innodb_log_file_size` tidak berubah. Namun, itu berlaku untuk satu file log, bukan dua. Hasilnya adalah instans basis data MariaDB versi 10.5 Anda yang dimutakhirkan menggunakan setengah ukuran log redo yang digunakannya sebelum pemutakhiran. Perubahan ini dapat memberikan dampak kinerja yang nyata. Untuk mengatasi masalah ini, Anda dapat menggandakan nilai parameter `innodb_log_file_size`. Lihat informasi tentang cara mengubah parameter di Memodifikasi parameter dalam grup parameter DB.
- Perintah SHOW SLAVE STATUS tidak didukung – Dalam versi MariaDB sebelum versi 10.5, perintah `SHOW SLAVE STATUS` memerlukan privilese `REPLICATION SLAVE`. Pada MariaDB versi 10.5, perintah `SHOW REPLICA STATUS` yang setara memerlukan privilese `REPLICATION REPLICA ADMIN`. Privilese baru ini tidak diberikan kepada pengguna master RDS.

  Alih-alih menggunakan perintah `SHOW REPLICA STATUS`, jalankan prosedur tersimpan `mysql.rds_replica_status` baru untuk menghasilkan informasi yang serupa. Lihat informasi yang lebih lengkap di mysql.rds_replica_status.
- Perintah SHOW RELAYLOG EVENTS tidak didukung – Dalam versi MariaDB sebelum versi 10.5, perintah `SHOW RELAYLOG EVENTS` memerlukan privilese `REPLICATION SLAVE`. Pada MariaDB versi 10.5, perintah ini memerlukan privilese `REPLICATION REPLICA ADMIN`. Privilese baru ini tidak diberikan kepada pengguna master RDS.

- Nilai bawaan baru untuk parameter – Parameter-parameter berikut memiliki nilai bawaan baru untuk instans basis data MariaDB versi 10.5:
  - Nilai bawaan parameter max_connections telah berubah menjadi `LEAST({DBInstanceClassMemory/25165760},12000)`. Lihat informasi tentang fungsi parameter `LEAST` di Fungsi parameter DB.
  - Nilai bawaan parameter innodb_adaptive_hash_index telah berubah menjadi `OFF` (`0`).
  - Nilai bawaan parameter innodb_checksum_algorithm telah berubah menjadi `full_crc32`.
  - Nilai bawaan parameter innodb_log_file_size telah berubah menjadi 2 GB.

Lihat daftar semua fitur MariaDB 10.5 dan dokumentasinya di Perubahan dan peningkatan dalam MariaDB 10.5 dan Catatan rilis - seri MariaDB 10.5 di situs web MariaDB.

Lihat daftar fitur yang tidak didukung di Fitur-fitur MariaDB yang didukung oleh Amazon RDS.

Dukungan MariaDB 10.4 di Amazon RDS

Amazon RDS mendukung fitur-fitur baru berikut untuk instans basis data Anda yang menjalankan MariaDB versi 10.4 atau lebih baru:

- Penyempurnaan keamanan akun pengguna – Peningkatan Kedaluwarsa kata sandi dan penguncian akun
- Penyempurnaan pengoptimal – Fitur jejak pengoptimal
- Penyempurnaan InnoDB – Dukungan DROP COLUMN instan dan ekstensi `VARCHAR` instan untuk `ROW_FORMAT=DYNAMIC` dan `ROW_FORMAT=COMPACT`
- Parameter baru – Meliputi tcp_nodelay, tls_versi, dan gtid_cleanup_batch_size

Lihat daftar semua fitur MariaDB 10.4 dan dokumentasinya di Perubahan dan peningkatan dalam MariaDB 10.4 dan Catatan rilis - seri MariaDB 10.4 di situs web MariaDB.

Lihat daftar fitur yang tidak didukung di Fitur-fitur MariaDB yang didukung oleh Amazon RDS.

Dukungan MariaDB 10.3 di Amazon RDS

Amazon RDS mendukung fitur-fitur baru berikut untuk instans basis data Anda yang menjalankan MariaDB versi 10.3 atau lebih baru:

- Kompatibilitas Oracle – pengurai kompatibilitas PL/SQL, urutan, INTERSECT dan EXCEPT untuk melengkapi UNION, deklarasi-deklarasi TYPE OF dan ROW TYPE OF baru, dan kolom yang tidak terlihat
- Pemrosesan data temporal – Tabel berversi sistem untuk mengueri keadaan dahulu dan kini basis data
- Fleksibilitas – Agregat buatan pengguna, kompresi kolom yang independen terhadap penyimpanan, dan dukungan protokol proksi untuk merelai alamat IP klien ke server
- Kemudahan dikelola – Operasi ADD COLUMN instan dan operasi bahasa definisi data (DDL) cepat hentikan bila gagal/fail fast

Lihat daftar semua fitur MariaDB 10.3 dan dokumentasinya di Perubahan dan peningkatan dalam MariaDB 10.3 dan Catatan rilis - seri MariaDB 10.3 di situs web MariaDB.

Lihat daftar fitur yang tidak didukung di Fitur-fitur MariaDB yang didukung oleh Amazon RDS.

Mesin penyimpanan yang didukung untuk MariaDB di Amazon RDS

RDS for MariaDB mendukung mesin-mesin penyimpanan berikut.

Topik

- Mesin penyimpanan InnoDB
- Mesin penyimpanan MyRocks

Mesin-mesin penyimpanan lain saat ini tidak didukung oleh RDS for MariaDB.

Mesin penyimpanan InnoDB

Meskipun MariaDB mendukung banyak mesin penyimpanan dengan berbagai kemampuan, tidak semuanya dioptimalkan untuk pemulihan dan ketahanan data. InnoDB adalah mesin penyimpanan yang dianjurkan untuk instans basis data MariaDB di Amazon RDS. Fitur Amazon RDS seperti Pemulihan Titik Waktu dan pemulihan cuplikan memerlukan mesin penyimpanan yang dapat dipulihkan dan hanya didukung untuk mesin penyimpanan yang disarankan untuk versi MariaDB.

Lihat informasi yang lebih lengkap di InnoDB.

Mesin penyimpanan MyRocks

Mesin penyimpanan MyRocks tersedia dalam RDS for MariaDB versi 10.6 dan lebih tinggi. Sebelum menggunakan mesin penyimpanan MyRocks di basis data produksi, kami menyarankan supaya Anda melakukan pembandingan dan pengujian menyeluruh untuk memverifikasi semua potensi manfaat dibandingkan dengan InnoDB untuk kasus penggunaan Anda.

Grup parameter bawaan untuk MariaDB versi 10.6 mencakup parameter-parameter MyRocks. Lihat informasi yang lebih lengkap di Parameter untuk MariaDB dan Bekerja dengan grup parameter.

Untuk membuat tabel yang menggunakan mesin penyimpanan MyRocks, tentukan `ENGINE=RocksDB` dalam pernyataan `CREATE TABLE`. Contoh berikut akan membuat tabel yang menggunakan mesin penyimpanan MyRocks.

```
CREATE TABLE test (a INT NOT NULL, b CHAR(10)) ENGINE=RocksDB;
```

Kami sangat menganjurkan agar Anda tidak menjalankan transaksi yang mencakup tabel-tabel InnoDB dan MyRocks. MariaDB tidak menjamin ACID (atomisitas, konsistensi, isolasi, ketahanan) untuk transaksi melintas mesin penyimpanan. Meskipun memiliki tabel InnoDB serentak dengan tabel MyRocks dalam satu instans basis data dapat dilakukan, pendekatan ini tidak kami anjurkan kecuali selama migrasi dari satu mesin penyimpanan ke mesin penyimpanan yang lain. Ketika tabel InnoDB dan tabel MyRocks berada di satu instans basis data, setiap mesin penyimpanan memiliki kolam penyangga sendiri, yang dapat menyebabkan kinerja menurun.

MyRocks tidak mendukung isolasi `SERIALIZABLE` atau kunci celah. Jadi, secara umum, Anda tidak dapat menggunakan MyRocks dengan replikasi berbasis pernyataan. Lihat informasi yang lebih lengkap di MyRocks dan Replikasi.

Saat ini, Anda hanya dapat mengubah parameter-parameter MyRocks berikut:

- `rocksdb_block_cache_size`
- `rocksdb_bulk_load`
- `rocksdb_bulk_load_size`
- `rocksdb_deadlock_detect`
- `rocksdb_deadlock_detect_depth`
- `rocksdb_max_latest_deadlocks`

Mesin penyimpanan MyRocks dan mesin penyimpanan InnoDB dapat bersaing memperoleh memori berdasarkan setelan untuk parameter-parameter `rocksdb_block_cache_size` dan `innodb_buffer_pool_size`. Dalam beberapa kasus, Anda mungkin hanya bermaksud menggunakan mesin penyimpanan MyRocks pada instans basis data tertentu. Jika demikian, sebaiknya atur `innodb_buffer_pool_size minimal` parameter ke nilai minimal dan atur `rocksdb_block_cache_size` setinggi mungkin.

Anda dapat mengakses file log MyRocks dengan menggunakan operasi-operasi `DescribeDBLogFiles` dan `DownloadDBLogFilePortion`.

Lihat informasi yang lebih lengkap tentang MyRocks di MyRocks di situs web MariaDB.

Penghangatan cache untuk MariaDB di Amazon RDS

Penghangatan cache InnoDB dapat memberikan raihan kinerja untuk instans basis data MariaDB Anda dengan menyimpan status saat ini dari kolam penyangga ketika instans basis data dimatikan, lalu memuatkan ulang kolam penyangga itu dari informasi tersimpan ketika instans basis data dimulai. Pendekatan ini mengabaikan kebutuhan bagi kolam penyangga untuk "menghangat" dari penggunaan basis data normal dan malah mengisi dahulu kolam penyangga dengan halaman untuk kueri-kueri umum yang diketahui. Lihat informasi yang lebih lengkap tentang penghangatan cache di Menguras dan memulihkan kolam penyangga dalam dokumentasi MariaDB.

Penghangatan cache diaktifkan secara bawaan pada instans basis data MariaDB 10.3 dan yang lebih tinggi. Untuk mengaktifkannya, atur parameter-parameter `innodb_buffer_pool_dump_at_shutdown` dan `innodb_buffer_pool_load_at_startup` ke 1 dalam grup parameter untuk instans basis data Anda. Mengubah nilai parameter-parameter ini di grup parameter akan memengaruhi semua instans basis data MariaDB yang menggunakan grup parameter itu. Untuk mengaktifkan penghangatan cache bagi instans basis data MariaDB tertentu, Anda mungkin perlu membuat grup parameter baru bagi instans itu. Lihat informasi tentang grup parameter di Bekerja dengan grup parameter.

Penghangatan cache terutama memberikan manfaat kinerja ke instans basis data yang menggunakan penyimpanan standar. Jika menggunakan penyimpanan PIOPS, Anda biasanya tidak melihat manfaat kinerja yang kentara.

> **Important**
>
> Jika instans basis data MariaDB tidak mati secara normal, seperti saat pindah saat gagal/failover, status kolam penyangga tidak disimpan ke disk. Dalam kasus ini, MariaDB

memuatkan file kolam penyangga apa pun yang tersedia saat instans basis data dimulai ulang. Tidak ada kerugian yang timbul, tetapi kolam penyangga yang dipulihkan mungkin tidak mencerminkan status terbaru kolam penyangga itu sebelum pemulaian ulang. Untuk memastikan bahwa Anda memiliki status terbaru kolam penyangga yang tersedia untuk menghangatkan cache saat pemulaian ulang, sebaiknya kuras kolam penyangga secara berkala "atas permintaan." Anda dapat menguras atau mengisi kolam penyangga atas permintaan.

Anda dapat membuat peristiwa untuk menguras kolam penyangga secara otomatis dan pada interval berkala. Misalnya, pernyataan berikut membuat peristiwa bernama `periodic_buffer_pool_dump` yang menguras kolam penyangga setiap jam.

```
CREATE EVENT periodic_buffer_pool_dump
   ON SCHEDULE EVERY 1 HOUR
   DO CALL mysql.rds_innodb_buffer_pool_dump_now();
```

Lihat informasi yang lebih lengkap di Peristiwa dalam dokumentasi MariaDB.

Menguras dan mengisi kolam penyangga atas permintaan

Anda dapat menyimpan dan memuatkan cache atas permintaan dengan menggunakan prosedur tersimpan berikut:

- Untuk menguras keadaan saat ini kolam penyangga ke disk, panggil prosedur tersimpan mysql.rds_innodb_buffer_pool_dump_now.
- Untuk memuatkan keadaan tersimpan kolam penyangga dari disk, panggil prosedur tersimpan mysql.rds_innodb_buffer_pool_load_now.
- Untuk membatalkan operasi pemuatan yang sedang berlangsung, panggil prosedur tersimpan mysql.rds_innodb_buffer_pool_load_abort.

Fitur-fitur MariaDB yang didukung oleh Amazon RDS

Fitur-fitur MariaDB berikut tidak didukung di Amazon RDS:

- Mesin penyimpanan S3
- Pengaya autentikasi – GSSAPI
- Pengaya autentikasi – Unix Socket

- Pengaya enkripsi AWS Key Management
- Replikasi tertunda untuk versi-versi MariaDB di bawah 10.6
- Enkripsi MariaDB asli saat rehat untuk InnoDB dan Aria

  Anda dapat mengaktifkan enkripsi saat rehat untuk instans basis data MariaDB dengan mengikuti petunjuk di Mengenkripsi sumber daya Amazon RDS.
- HandlerSocket
- Jenis tabel JSON untuk versi-versi MariaDB di bawah 10.6
- MariaDB ColumnStore
- MariaDB Galera Cluster
- Replikasi multisumber
- Mesin penyimpanan MyRocks untuk versi-versi MariaDB di bawah 10.6
- Pengaya validasi kata sandi, `simple_password_check`, dan `cracklib_password_check`
- Mesin penyimpanan Spider
- Mesin penyimpanan Sphinx
- Mesin penyimpanan TokuDB
- Atribut-atribut objek khusus mesin penyimpanan, sebagaimana dijelaskan di Atribut-atribut Tabel/Bidang/Indeks baru yang ditentukan mesin dalam dokumentasi MariaDB
- Enkripsi tabel dan ruang tabel
- Pengaya Pengelolaan Kunci Hashicorp
- Menjalankan dua pemutakhiran secara paralel

Untuk memberikan pengalaman layanan terkelola, Amazon RDS tidak memberikan akses shell ke instans basis data, dan membatasi akses ke sejumlah prosedur dan tabel sistem tertentu yang memerlukan privilese lanjut. Amazon RDS mendukung akses ke basis data di instans basis data dengan menggunakan aplikasi klien SQL standar. Amazon RDS tidak mengizinkan akses host langsung ke instans basis data dengan menggunakan Telnet, Secure Shell (SSH), atau Windows Remote Desktop Connection.

Versi-versi MariaDB pada Amazon RDS

Untuk MariaDB, nomor versi disusun sebagai versi X.Y.Z. Dalam terminologi Amazon RDS, X.Y menunjukkan versi utama, dan Z adalah nomor versi kecil. Untuk implementasi Amazon RDS, perubahan versi dianggap besar jika nomor versi utama berubah, misalnya, dari versi 10.5 ke 10.6. Perubahan versi dianggap kecil jika hanya nomor versi minor yang berubah, misalnya dari versi 10.6.14 ke 10.6.16.

Topik

Versi-versi kecil MariaDB yang didukung di Amazon RDS

Amazon RDS saat ini mendukung versi-versi kecil MariaDB berikut.

> **Note**
>
> Tanggal yang berupa hanya bulan dan tahun merupakan perkiraan, dan akan diperbarui dengan tanggal persisnya saat diketahui.

Versi mesin MariaDB	Tanggal rilis komunitas	Tanggal rilis RDS	Tanggal akhir dukungan standar RDS
10.11			
10.11.7	7 Februari 2024	26 Februari 2024	Maret 2025
10.11.6	13 November 2023	12 Desember 2023	Maret 2025
10.11.5	14 Agustus 2023	7 September 2023	September 2024

Versi mesin MariaDB	Tanggal rilis komunitas	Tanggal rilis RDS	Tanggal akhir dukungan standar RDS
10.11.4	7 Juni 2023	21 Agustus 2023	September 2024
10.6			
10.6.17	7 Februari 2024	26 Februari 2024	Maret 2025
10.6.16	13 November 2023	12 Desember 2023	Maret 2025
10.6.15	14 Agustus 2023	7 September 2023	September 2024
10.6.14	7 Juni 2023	22 Juni 2023	September 2024
10.6.13	10 Mei 2023	15 Juni 2023	September 2024
10.5			
10.5.24	7 Februari 2024	26 Februari 2024	Maret 2025
10.5.23	13 November 2023	12 Desember 2023	Maret 2025
10.5.22	14 Agustus 2023	7 September 2023	September 2024
10.5.21	7 Juni 2023	22 Juni 2023	September 2024
10.5.20	10 Mei 2023	15 Juni 2023	September 2024
10.4			
10.4.33	7 Februari 2024	26 Februari 2024	Agustus 2024
10.4.32	13 November 2023	12 Desember 2023	Agustus 2024
10.4.31	14 Agustus 2023	7 September 2023	Agustus 2024
10.4.30	7 Juni 2023	22 Juni 2023	Agustus 2024
10.4.29	10 Mei 2023	15 Juni 2023	Agustus 2024

Anda dapat menentukan versi MariaDB mana pun yang saat ini didukung ketika membuat instans basis data baru. Anda dapat menentukan versi utama (seperti MariaDB 10.5) dan sebarang versi kecil yang didukung untuk versi utama itu. Jika tidak ada versi yang ditentukan, Amazon RDS menjadikan default sebuah versi yang didukung, biasanya versi terbaru. Jika versi utama ditentukan tetapi versi kecil tidak, Amazon RDS menjadikan menetapkan default ke rilis versi utama terbaru yang telah Anda tentukan. Untuk melihat daftar versi yang didukung, serta default untuk instans DB yang baru dibuat, gunakan perintah. `describe-db-engine-versions` AWS CLI

Misalnya, untuk menampilkan daftar versi-versi mesin yang didukung untuk RDS for MariaDB, jalankan perintah CLI berikut:

```
aws rds describe-db-engine-versions --engine mariadb --query "*[].
{Engine:Engine,EngineVersion:EngineVersion}" --output text
```

Versi MariaDB default mungkin berbeda-beda menurut Wilayah AWS. Untuk membuat instans basis data dengan versi kecil tertentu, tentukan versi kecil selama pembuatan instans basis data. Anda dapat menentukan versi minor default untuk Wilayah AWS menggunakan AWS CLI perintah berikut:

```
aws rds describe-db-engine-versions --default-only --engine mariadb
 --engine-version major-engine-version --region region --query "*[].
{Engine:Engine,EngineVersion:EngineVersion}" --output text
```

Ganti *major-engine-version*dengan versi mesin utama, dan ganti *wilayah* dengan Wilayah AWS. Misalnya, AWS CLI perintah berikut mengembalikan versi mesin minor MariaDB default untuk versi utama 10,5 dan AS Barat (Oregon Wilayah AWS) (us-barat-2):

```
aws rds describe-db-engine-versions --default-only --engine mariadb --engine-version
 10.5 --region us-west-2 --query "*[].{Engine:Engine,EngineVersion:EngineVersion}" --
output text
```

Versi-versi utama MariaDB yang didukung di Amazon RDS

Versi-versi utama RDS for MariaDB tetap tersedia setidaknya sampai akhir pemakaian komunitas untuk versi komunitas yang bersangkutan. Anda dapat menggunakan tanggal-tanggal berikut untuk merencanakan siklus pengujian dan pemutakhiran. Jika Amazon memperpanjang dukungan untuk versi RDS for MariaDB lebih lama daripada yang dinyatakan semula, kami merencanakan untuk memperbarui tabel ini guna mencerminkan tanggal yang mundur itu.

> **Note**
>
> Tanggal yang berupa hanya bulan dan tahun merupakan perkiraan, dan akan diperbarui dengan tanggal persisnya saat diketahui.

Versi utama MariaDB	Tanggal rilis komunitas	Tanggal rilis RDS	Tanggal akhir pemakaian komunitas	Tanggal akhir dukungan standar RDS
MariaDB 10.11	16 Februari 2023	21 Agustus 2023	16 Februari 2028	Februari 2028
MariaDB 10.6	6 Juli 2021	3 Februari 2022	6 Juli 2026	Juli 2026
MariaDB 10.5	24 Juni 2020	21 Januari 2021	24 Juni 2025	Juni 2025
MariaDB 10.4	18 Juni 2019	6 April 2020	18 Juni 2024	Agustus 2024

Akhir dukungan standar MariaDB 10.3 RDS

Pada 23 Oktober 2023, Amazon RDS memulai proses akhir dukungan standar RDS untuk MariaDB versi 10.3 dengan menggunakan jadwal berikut, yang mencakup rekomendasi pemutakhiran. Kami menyarankan agar Anda memutakhirkan semua instans basis data MariaDB 10.3 ke MariaDB 10.6 sesegera mungkin. Untuk informasi selengkapnya, lihat Meningkatkan mesin DB MariaDB.

Tindakan atau rekomendasi	Tanggal
Kami menyarankan agar Anda memutakhirkan secara manual instans-instans basis data MariaDB 10.3 Anda ke versi pilihan Anda. Anda dapat memutakhirkan secara langsung ke MariaDB versi 10.6 atau lebih tinggi.	Sekarang–23 Oktober 2023
Kami menyarankan agar Anda memutakhirkan cuplikan MariaDB 10.3 ke versi pilihan Anda.	Sekarang–23 Oktober 2023

Tindakan atau rekomendasi	Tanggal
Anda tidak bisa lagi membuat instans basis data MariaDB 10.3 baru. Anda masih dapat membuat replika baca instans basis data MariaDB 10.3 yang ada dan mengubahn ya dari deployment AZ Tunggal ke deployment Multi-AZ.	23 Agustus 2023
Amazon RDS memulai pemutakhiran otomatis untuk instans basis data MariaDB 10.3 Anda ke versi 10.6.	23 Oktober 2023
Amazon RDS memulai pemutakhiran otomatis ke versi 10.6 untuk setiap instans basis data MariaDB 10.3 yang dipulihkan dari cuplikan.	23 Oktober 2023
Amazon RDS memutakhirkan secara otomatis instans basis data MariaDB 10.3 yang tersisa ke versi 10.6 entah berada dalam jendela pemeliharaan terjadwal ataupun tidak.	23 Januari 2024

Akhir dukungan standar MariaDB 10.2 RDS

Pada 15 Oktober 2022, Amazon RDS memulai proses akhir dukungan standar RDS untuk MariaDB versi 10.2 dengan menggunakan jadwal berikut, yang mencakup rekomendasi pemutakhiran. Kami menyarankan agar Anda memutakhirkan semua instans basis data MariaDB 10.2 ke MariaDB 10.3 atau lebih tinggi sesegera mungkin. Untuk informasi selengkapnya, lihat Meningkatkan mesin DB MariaDB.

Tindakan atau rekomendasi	Tanggal
Kami menyarankan agar Anda memutakhirkan secara manual instans-instans basis data MariaDB 10.2 Anda ke versi pilihan Anda. Anda dapat memutakhirkan secara langsung ke MariaDB versi 10.3 atau 10.6.	Sekarang–15 Oktober 2022

Tindakan atau rekomendasi	Tanggal
Kami menyarankan agar Anda memutakhirkan secara manual cuplikan MariaDB 10.2 ke versi pilihan Anda.	Sekarang–15 Oktober 2022
Anda tidak bisa lagi membuat instans basis data MariaDB 10.2 baru. Anda masih dapat membuat replika baca instans basis data MariaDB 10.2 yang ada dan mengubahn ya dari deployment AZ Tunggal ke deployment Multi-AZ.	15 Juli 2022
Amazon RDS memulai pemutakhiran otomatis untuk instans basis data MariaDB 10.2 Anda ke versi 10.3.	15 Oktober 2022
Amazon RDS memulai pemutakhiran otomatis ke versi 10.3 untuk setiap instans basis data MariaDB 10.2 yang dipulihkan dari cuplikan.	15 Oktober 2022
Amazon RDS memutakhirkan secara otomatis instans basis data MariaDB 10.2 yang tersisa ke versi 10.3 entah berada dalam jendela pemeliharaan terjadwal ataupun tidak.	15 Januari 2023

Lihat informasi yang lebih lengkap tentang akhir dukungan standar RDS untuk Amazon RDS for MariaDB 10.2 di Pengumuman: Tanggal Akhir Pemakaian Amazon Relational Database Service (Amazon RDS) for MariaDB 10.2 adalah 15 Oktober 2022.

Versi-versi yang dihentikan untuk Amazon RDS for MariaDB

Amazon RDS for MariaDB versi 10.0, 10.1, dan 10.2 dihentikan.

Lihat informasi tentang kebijakan pengusangan Amazon RDS for MariaDB, lihat Tanya Jawab Umum Amazon RDS.

Menghubungkan ke instans DB yang menjalankan mesin basis data MariaDB

Setelah Amazon RDS menyediakan instans DB, Anda dapat menggunakan aplikasi klien MariaDB standar atau utilitas guna terhubung ke instans tersebut. Dalam string koneksi, tentukan alamat Sistem Nama Domain (DNS) dari titik akhir instans DB sebagai parameter host. Tentukan juga nomor port dari titik akhir instans DB sebagai parameter port.

Anda dapat terhubung ke instans DB Amazon RDS for MariaDB menggunakan alat seperti klien baris perintah MySQL. Untuk mengetahui informasi selengkapnya tentang cara menggunakan klien baris perintah MySQL, lihat alat baris perintah mysql di dokumentasi MySQL. Satu aplikasi berbasis GUI yang dapat Anda gunakan untuk terhubung adalah Heidi. Untuk mengetahui informasi selengkapnya, lihat halaman Mengunduh HeidiSQL. Untuk mengetahui informasi tentang cara menginstal MySQL (termasuk klien baris perintah MySQL), lihat Menginstal dan meningkatkan MySQL.

Sebagian besar distribusi Linux menyertakan klien MariaDB, bukan klien Oracle MySQL. Untuk menginstal klien baris perintah MySQL di Amazon Linux 2023, jalankan perintah berikut:

```
sudo dnf install mariadb105
```

Untuk menginstal klien baris perintah MySQL di Amazon Linux 2, jalankan perintah berikut:

```
sudo yum install mariadb
```

Untuk menginstal klien baris perintah MySQL di sebagian besar distribusi Linux berbasis DEB, jalankan perintah berikut.

```
apt-get install mariadb-client
```

Untuk memeriksa versi klien baris perintah MySQL Anda, jalankan perintah berikut.

```
mysql --version
```

Untuk membaca dokumentasi MySQL untuk versi klien Anda saat ini, jalankan perintah berikut.

```
man mysql
```

Untuk terhubung ke instans DB dari luar cloud privat virtual (VPC) berdasarkan Amazon VPC, instans DB harus dapat diakses secara publik. Selain itu, akses harus diberikan menggunakan aturan masuk grup keamanan instans DB, dan persyaratan lain harus terpenuhi. Untuk mengetahui informasi selengkapnya, lihat Tidak dapat terhubung ke instans DB Amazon RDS.

Anda dapat menggunakan enkripsi SSL pada koneksi ke instans DB MariaDB. Untuk mengetahui informasinya, lihat Menggunakan SSL/TLS dengan instans basis data MariaDB.

Topik

- Menemukan informasi koneksi untuk instans DB MariaDB
- Menghubungkan dari klien baris perintah MySQL (tidak terenkripsi)
- Memecahkan masalah koneksi ke instans DB MariaDB Anda

Menemukan informasi koneksi untuk instans DB MariaDB

Informasi koneksi untuk instans DB mencakup titik akhir, port, dan pengguna basis datanya yang valid, seperti pengguna utama. Sebagai contoh, anggaplah bahwa nilai titik akhir adalah `mydb.123456789012.us-east-1.rds.amazonaws.com`. Dalam hal ini, nilai port adalah `3306`, dan pengguna basis data adalah `admin`. Dengan informasi ini, Anda menentukan nilai-nilai berikut dalam string koneksi:

- Untuk host atau nama host, atau nama DNS, tentukan `mydb.123456789012.us-east-1.rds.amazonaws.com`.
- Untuk port, tentukan `3306`.
- Untuk pengguna, tentukan `admin`.

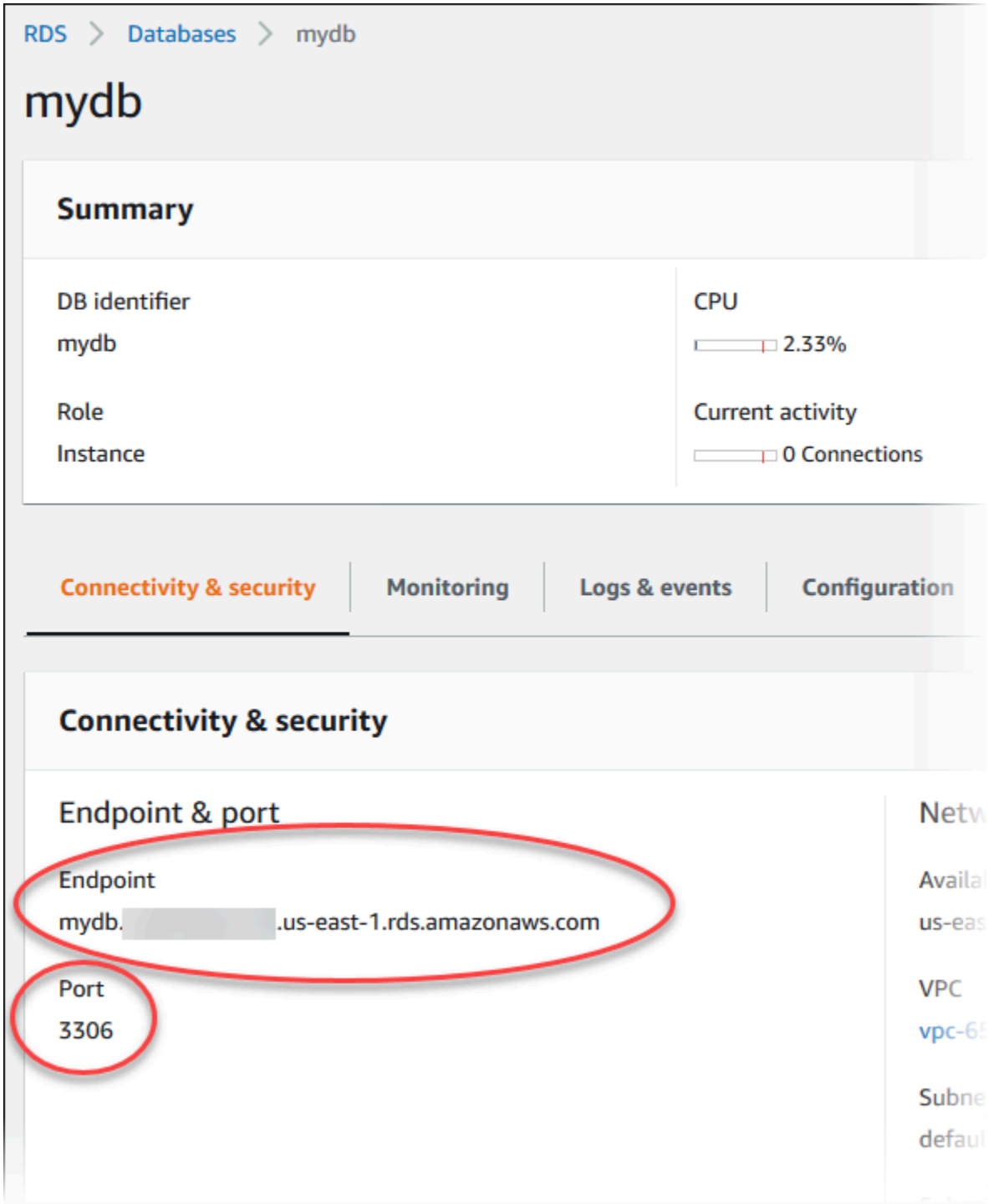
Untuk terhubung ke instans DB, gunakan klien apa saja untuk mesin DB MariaDB. Misalnya, Anda dapat menggunakan klien baris perintah MySQL atau MySQL Workbench.

Untuk menemukan informasi koneksi untuk instans DB, Anda dapat menggunakan describe-db-instancesperintah, AWS Command Line Interface (AWS CLI)AWS Management Console, atau operasi Amazon RDS API DescribedBInstances untuk mencantumkan detailnya.

Konsol

Untuk menemukan informasi koneksi instans DB di AWS Management Console

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data untuk menampilkan daftar instans DB Anda.
3. Pilih nama instans DB MariaDB untuk menampilkan detailnya.
4. Di tab Konektivitas & keamanan, salin titik akhir. Selain itu, catat nomor porta. Anda memerlukan titik akhir dan nomor port untuk terhubung ke instans DB.



5. Jika Anda perlu menemukan nama pengguna utama, pilih tab Konfigurasi dan lihat nilai Nama pengguna utama.

AWS CLI

Untuk menemukan informasi koneksi untuk instance MariaDB dengan menggunakan, panggil AWS CLI perintah. describe-db-instances Dalam panggilan tersebut, buat kueri untuk ID instans DB, titik akhir, port, dan nama pengguna utama.

Untuk Linux, macOS, atau Unix:

```
aws rds describe-db-instances \
  --filters "Name=engine,Values=mariadb" \
  --query "*[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername]"
```

Untuk Windows:

```
aws rds describe-db-instances ^
  --filters "Name=engine,Values=mariadb" ^
  --query "*[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername]"
```

Output Anda akan terlihat seperti berikut ini.

```
[
    [
        "mydb1",
        "mydb1.123456789012.us-east-1.rds.amazonaws.com",
        3306,
        "admin"
    ],
    [
        "mydb2",
        "mydb2.123456789012.us-east-1.rds.amazonaws.com",
        3306,
        "admin"
    ]
]
```

API RDS

Untuk menemukan informasi koneksi instans DB dengan menggunakan API Amazon RDS, panggil operasi DescribeDBInstances. Dalam output, temukan nilai untuk alamat titik akhir, port titik akhir, dan nama pengguna utama.

Menghubungkan dari klien baris perintah MySQL (tidak terenkripsi)

> **Important**
>
> Hanya gunakan koneksi MySQL yang tidak terenkripsi saat klien dan server berada di VPC yang sama dan jaringan tepercaya. Untuk mengetahui informasi tentang cara menggunakan koneksi terenkripsi, lihat Menghubungkan dari klien baris perintah MySQL dengan SSL/TLS (terenkripsi).

Untuk terhubung ke instans DB menggunakan klien baris perintah MySQL, masukkan perintah berikut pada prompt perintah di komputer klien. Melakukan langkah ini akan menghubungkan Anda ke basis data di instans DB MariaDB. Lakukan penggantian nama DNS (titik akhir) instans DB Anda untuk *<endpoint>*, dan nama pengguna utama yang Anda gunakan untuk *<mymasteruser>*. Masukkan kata sandi utama yang Anda gunakan saat dimintai kata sandi.

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

Setelah memasukkan kata sandi untuk pengguna, Anda akan melihat output yang terlihat seperti berikut ini.

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.6.10-MariaDB-log Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]>
```

Memecahkan masalah koneksi ke instans DB MariaDB Anda

Dua penyebab umum kegagalan koneksi ke instans DB baru adalah sebagai berikut:

- Instans DB dibuat menggunakan grup keamanan yang tidak mengotorisasi koneksi dari perangkat atau instans Amazon EC2 tempat aplikasi atau utilitas MariaDB berjalan. Instans DB harus memiliki grup keamanan VPC yang mengotorisasi koneksi. Untuk mengetahui informasi selengkapnya, lihat Amazon VPC dan Amazon RDS.

Anda dapat menambahkan atau mengedit aturan masuk di grup keamanan. Untuk Sumber, pilih IP Saya. Pilihan ini akan mengizinkan akses ke instans DB dari alamat IP yang terdeteksi di browser Anda.

- Instans DB dibuat menggunakan port default 3306, dan perusahaan Anda memiliki aturan firewall yang memblokir koneksi ke port tersebut dari perangkat di jaringan perusahaan Anda. Untuk memperbaiki kegagalan ini, buat ulang instans dengan port yang berbeda.

Untuk mengetahui informasi selengkapnya tentang masalah koneksi, lihat Tidak dapat terhubung ke instans DB Amazon RDS.

Mengamankan koneksi instans DB MariaDB

Anda dapat mengelola keamanan instans DB MariaDB Anda.

Topik

- Keamanan MariaDB di Amazon RDS
- Mengenkripsi dengan SSL/TLS koneksi klien dengan instans basis data MariaDB
- Memperbarui aplikasi untuk terhubung ke instans MariaDB menggunakan sertifikat SSL/TLS baru

Keamanan MariaDB di Amazon RDS

Keamanan untuk instans basis data MariaDB dikelola pada tiga tingkat:

- AWS Identity and Access Management mengendalikan siapa yang dapat melakukan tindakan pengelolaan Amazon RDS pada instans basis data. Jika Anda menghubungi AWS dengan menggunakan kredensial IAM, akun IAM Anda harus memiliki kebijakan IAM yang memberikan izin yang disyaratkan untuk menjalankan operasi pengelolaan Amazon RDS. Lihat informasi yang lebih lengkap di Manajemen identitas dan akses untuk Amazon RDS.
- Saat membuat instans basis data, Anda menggunakan grup keamanan VPC untuk mengendalikan perangkat dan instans Amazon EC2 yang boleh membuka koneksi dengan titik akhir dan porta instans basis data. Koneksi ini dapat dibuat dengan menggunakan Lapisan Soket Aman (SSL) dan Keamanan Lapisan Pengangkutan (TLS). Selain itu, aturan tembok api/firewall di perusahaan Anda dapat mengendalikan apakah perangkat yang berjalan di perusahaan Anda boleh membuka koneksi dengan instans basis data.
- Setelah koneksi dibuka untuk instans basis data MariaDB, autentikasi masuk dan izin diterapkan dengan cara yang sama dengan pada instans MariaDB mandiri. Perintah-perintah seperti `CREATE USER`, `RENAME USER`, `GRANT`, `REVOKE` dan `SET PASSWORD` bekerja sebagaimana pada basis data mandiri, seperti mengubah langsung tabel skema basis data.

Saat Anda membuat instans basis data Amazon RDS, pengguna master memiliki privilese-privilese bawaan berikut:

- `alter`
- `alter routine`
- `create`

- `create routine`
- `create temporary tables`
- `create user`
- `create view`
- `delete`
- `drop`
- `event`
- `execute`
- `grant option`
- `index`
- `insert`
- `lock tables`
- `process`
- `references`
- `reload`

  Privilese ini terbatas pada instans basis data MariaDB. Privilese ini tidak memberikan akses ke operasi-operasi FLUSH LOGS atau FLUSH TABLES WITH READ LOCK.
- `replication client`
- `replication slave`
- `select`
- `show databases`
- `show view`
- `trigger`
- `update`

Lihat informasi yang lebih lengkap tentang semua privilese ini di Pengelolaan akun pengguna dalam dokumentasi MariaDB.

> **Note**
>
> Meskipun Anda dapat menghapus pengguna master di instans basis data, hal itu tidak kami sarankan. Untuk membuat ulang pengguna master, gunakan API `ModifyDBInstance` atau

`modify-db-instance` AWS CLI dan tetapkan kata sandi pengguna master baru dengan parameter yang sesuai. Jika tidak ada dalam instans, pengguna master akan dibuat dengan kata sandi yang ditentukan.

Untuk menyediakan layanan pengelolaan bagi setiap instans basis data, pengguna `rdsadmin` dibuat saat instans basis data dibuat. Mencoba mengedrop, mengganti nama, mengubah kata sandi, atau mengubah privilese untuk akun `rdsadmin` Anda akan menghasilkan kesalahan.

Untuk memungkinkan pengelolaan instans basis data, perintah-perintah `kill` dan `kill_query` standar telah dibatasi. Perintah-perintah Amazon RDS `mysql.rds_kill`, `mysql.rds_kill_query`, dan `mysql.rds_kill_query_id` disediakan untuk digunakan pada MariaDB dan juga MySQL sehingga Anda dapat mengakhiri sesi atau kueri pengguna pada instans basis data.

Mengenkripsi dengan SSL/TLS koneksi klien dengan instans basis data MariaDB

Lapisan Soket Aman (Secure Sockets Layer, SSL) adalah protokol standar industri untuk mengamankan koneksi jaringan antara klien dan server. Setelah SSL versi 3.0, namanya diubah menjadi Keamanan Lapisan Pengangkutan (TLS). Amazon RDS mendukung enkripsi SSL/TLS untuk instans basis data MariaDB. Dengan SSL/TLS, Anda dapat mengenkripsi koneksi antara klien aplikasi dan instans basis data MariaDB. Dukungan SSL/TLS tersedia di semua. Wilayah AWS

Topik

- Menggunakan SSL/TLS dengan instans basis data MariaDB
- Mewajibkan SSL/TLS untuk semua koneksi dengan instans basis data MariaDB
- Menghubungkan dari klien baris perintah MySQL dengan SSL/TLS (terenkripsi)

Menggunakan SSL/TLS dengan instans basis data MariaDB

Amazon RDS membuat sertifikat SSL/TLS dan menginstal sertifikat tersebut pada instans DB ketika Amazon RDS menyediakan instans. Sertifikat ini ditandatangani oleh otoritas sertifikat. Sertifikat SSL/TLS mencakup titik akhir instans DB sebagai Nama Umum (Common Name, CN) untuk sertifikat SSL/TLS guna menghalangi serangan spoofing.

Sertifikat SSL/TLS yang dibuat oleh Amazon RDS adalah entitas root tepercaya dan semestinya berfungsi dalam sebagian besar kasus, tetapi mungkin gagal jika aplikasi Anda tidak menerima rantai sertifikat. Jika aplikasi Anda tidak menerima rantai sertifikat, Anda mungkin perlu menggunakan sertifikat perantara untuk terhubung dengan Wilayah AWS Anda. Misalnya, Anda harus menggunakan sertifikat perantara untuk terhubung ke AWS GovCloud (US) Wilayah menggunakan SSL/TLS.

Untuk informasi tentang mengunduh sertifikat, lihat . Lihat informasi yang lebih lengkap tentang cara menggunakan TLS/SSL dengan MySQL di Memperbarui aplikasi untuk terhubung ke instans MariaDB menggunakan sertifikat SSL/TLS baru.

Amazon RDS for MariaDB mendukung Transport Layer Security (TLS) versi 1.3, 1.2, 1.1, dan 1.0. Dukungan TLS tergantung pada versi minor MariaDB. Tabel berikut menunjukkan dukungan TLS untuk MariaDB versi minor.

Versi TLS	MariaDB 10.11	MariaDB 10.6	MariaDB 10.5	MariaDB 10.4
TLS 1.3	Semua versi minor	Semua versi minor	Semua versi minor	Semua versi minor
TLS 1.2	Semua versi minor	Semua versi minor	Semua versi minor	Semua versi minor
TLS 1.1	10.11.6 dan lebih rendah	10.6.16 dan lebih rendah	10.5.23 dan lebih rendah	10.4.32 dan lebih rendah
TLS 1.0	10.11.6 dan lebih rendah	10.6.16 dan lebih rendah	10.5.23 dan lebih rendah	10.4.32 dan lebih rendah

Anda dapat meminta koneksi SSL/TLS untuk akun pengguna tertentu. Misalnya, Anda dapat menggunakan salah satu pernyataan berikut, sesuai dengan versi MariaDB, untuk mewajibkan koneksi SSL/TLS pada akun pengguna `encrypted_user`.

Gunakan pernyataan berikut.

```
ALTER USER 'encrypted_user'@'%' REQUIRE SSL;
```

Lihat informasi yang lebih lengkap tentang koneksi SSL/TLS dengan MariaDB di Securing Connections for Client and Server dalam dokumentasi MariaDB.

Mewajibkan SSL/TLS untuk semua koneksi dengan instans basis data MariaDB

Gunakan parameter `require_secure_transport` untuk mewajibkan bahwa semua koneksi pengguna dengan instans basis data MariaDB Anda menggunakan SSL/TLS. Secara default, parameter `require_secure_transport` diatur ke `OFF`. Anda dapat mengatur parameter `require_secure_transport` ke `ON` guna mewajibkan SSL/TLS untuk koneksi dengan instans basis data Anda.

> **Note**
>
> Parameter `require_secure_transport` hanya didukung untuk MariaDB versi 10.5 dan lebih tinggi.

Anda dapat mengatur nilai parameter `require_secure_transport` dengan memperbarui grup parameter basis data untuk instans basis data Anda. Anda tidak perlu mem-boot ulang instans DB agar perubahan berlaku.

Saat parameter `require_secure_transport` diatur ke `ON` untuk suatu instans DB, klien basis data dapat terhubung dengannya jika dapat membentuk koneksi terenkripsi. Jika tidak, pesan kesalahan yang serupa dengan yang berikut ini ditampilkan kepada klien:

```
ERROR 1045 (28000): Access denied for user 'USER'@'localhost' (using password: YES |
 NO)
```

Lihat informasi tentang pengaturan parameter di Memodifikasi parameter dalam grup parameter DB.

Lihat informasi yang lebih lengkap tentang parameter `require_secure_transport` dalam dokumentasi MariaDB.

Menghubungkan dari klien baris perintah MySQL dengan SSL/TLS (terenkripsi)

Parameter-parameter program klien `mysql` sedikit berbeda jika Anda menggunakan MySQL versi 5.7, MySQL versi 8.0, atau versi MariaDB.

Untuk mengetahui versi yang Anda miliki, jalankan perintah `mysql` dengan opsi `--version`. Dalam contoh berikut, output menunjukkan bahwa program klien berasal dari MariaDB.

```
$ mysql --version
mysql  Ver 15.1 Distrib 10.5.15-MariaDB, for osx10.15 (x86_64) using readline 5.1
```

Sebagian besar distribusi Linux, seperti Amazon Linux, CentOS, SUSE, dan Debian telah mengganti MySQL dengan MariaDB, dan versi `mysql` di dalamnya adalah dari MariaDB.

Untuk terhubung dengan instans DB Anda menggunakan SSL/TLS, ikuti langkah-langkah ini:

Untuk terhubung dengan instans DB dengan SSL/TLS menggunakan klien baris perintah MySQL

1. Unduh sertifikat root yang berfungsi untuk semua Wilayah AWS.

   Untuk informasi tentang mengunduh sertifikat, lihat .

2. Gunakan klien baris perintah MySQL untuk terhubung dengan instans DB melalui enkripsi SSL/TLS. Untuk parameter `-h`, ganti nama (titik akhir) DNS untuk instans DB Anda. Untuk parameter `--ssl-ca`, ganti nama file sertifikat SSL/TLS. Untuk parameter `-P`, ganti port untuk instans DB Anda. Untuk parameter `-u`, ganti nama pengguna dari pengguna basis data yang valid, seperti pengguna master. Masukkan kata sandi pengguna master ketika diminta.

   Contoh berikut menunjukkan cara meluncurkan klien dengan menggunakan parameter `--ssl-ca` yang memakai klien MariaDB:

   ```
   mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=global-bundle.pem --ssl -P 3306 -u myadmin -p
   ```

   Untuk mewajibkan bahwa koneksi SSL/TLS memeriksa titik akhir instans basis data terhadap titik akhir dalam sertifikat SSL/TLS, masukkan perintah berikut:

   ```
   mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=global-bundle.pem --ssl-verify-server-cert -P 3306 -u myadmin -p
   ```

   Contoh berikut menunjukkan cara meluncurkan klien dengan menggunakan parameter `--ssl-ca` yang memakai klien MySQL 5.7 atau lebih tinggi:

   ```
   mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=global-bundle.pem --ssl-mode=REQUIRED -P 3306 -u myadmin -p
   ```

3. Masukkan kata sandi pengguna master ketika diminta.

Anda semestinya melihat output yang serupa berikut.

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.6.10-MariaDB-log Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]>
```

Memperbarui aplikasi untuk terhubung ke instans MariaDB menggunakan sertifikat SSL/TLS baru

Sejak 13 Januari 2023, Amazon RDS telah menerbitkan serifikat Otoritas Sertifikat (CA) baru untuk terhubung ke instans DB RDS menggunakan Lapisan Soket Aman atau Keamanan Lapisan Pengangkutan (SSL/TLS). Setelah itu, Anda dapat menemukan informasi tentang pembaruan aplikasi untuk menggunakan sertifikat baru.

Topik ini dapat membantu menentukan apakah aplikasi Anda memerlukan verifikasi sertifikat untuk terhubung ke instans DB Anda.

> **Note**
>
> Beberapa aplikasi dikonfigurasi untuk terhubung ke MariaDB hanya jika aplikasi tersebut berhasil memverifikasi sertifikat pada server. Untuk aplikasi tersebut, Anda harus memperbarui penyimpanan kepercayaan aplikasi klien untuk menyertakan sertifikat CA baru. Anda dapat menentukan mode SSL berikut: `disabled`, `preferred`, dan `required`. Saat Anda menggunakan mode SSL `preferred` dan sertifikat CA tidak ada atau tidak diperbarui, koneksi kembali tidak menggunakan SSL dan masih berhasil terhubung.
> Sebaiknya hindari mode `preferred`. Dalam mode `preferred`, jika koneksi menghadapi sertifikat yang tidak valid, koneksi berhenti menggunakan enkripsi dan melanjutkan tanpa enkripsi.

Setelah Anda memperbarui sertifikat CA di penyimpanan kepercayaan aplikasi klien, Anda dapat merotasi sertifikat di instans DB Anda. Sebaiknya Anda menguji prosedur ini di lingkungan pengembangan dan penahapan sebelum menerapkannya di lingkungan produksi Anda.

Untuk informasi selengkapnya tentang rotasi sertifikat, lihat Merotasi sertifikat SSL/TLS. Untuk informasi selengkapnya tentang cara mengunduh sertifikat, lihat . Untuk informasi tentang menggunakan SSL/TLS dengan instans DB MariaDB, lihat Menggunakan SSL/TLS dengan instans basis data MariaDB.

Topik

- Menentukan apakah klien memerlukan verifikasi sertifikat agar dapat terhubung
- Memperbarui penyimpanan kepercayaan aplikasi Anda
- Contoh kode Java untuk membangun koneksi SSL

Menentukan apakah klien memerlukan verifikasi sertifikat agar dapat terhubung

Anda dapat memeriksa apakah klien JDBC dan klien MySQL memerlukan verifikasi sertifikat untuk terhubung.

JDBC

Contoh MySQL Connector/J 8.0 berikut menunjukkan satu cara untuk memeriksa properti koneksi JDBC aplikasi untuk menentukan apakah koneksi yang berhasil memerlukan sertifikat yang valid. Untuk informasi selengkapnya tentang semua opsi koneksi JDBC untuk MySQL, lihat Properti konfigurasi di dokumentasi MySQL.

Saat menggunakan MySQL Connector/J 8.0, koneksi SSL memerlukan verifikasi terhadap sertifikat CA server jika properti koneksi Anda memiliki `sslMode` yang diatur ke `VERIFY_CA` atau `VERIFY_IDENTITY`, seperti pada contoh berikut.

```
Properties properties = new Properties();
properties.setProperty("sslMode", "VERIFY_IDENTITY");
properties.put("user", DB_USER);
properties.put("password", DB_PASSWORD);
```

> **Note**
>
> Jika Anda menggunakan MySQL Java Connector v5.1.38 atau yang lebih baru, atau MySQL Java Connector v8.0.9 atau yang lebih baru untuk terhubung ke basis data Anda, meski Anda belum mengonfigurasi aplikasi secara eksplisit untuk menggunakan SSL/TLS saat terhubung

ke basis data Anda, driver klien ini akan menggunakan SSL/TLS secara default. Selain itu, saat menggunakan SSL/TLS, aplikasi akan melakukan verifikasi sertifikat parsial dan gagal terhubung jika sertifikat server basis data kedaluwarsa.
Tentukan kata sandi selain prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

MySQL

Contoh Klien MySQL berikut menunjukkan dua cara untuk memeriksa koneksi MySQL skrip untuk menentukan apakah koneksi yang berhasil memerlukan sertifikat yang valid. Untuk informasi selengkapnya tentang semua opsi koneksi dengan Klien MySQL, lihat Konfigurasi sisi klien untuk koneksi terenkripsi di dokumentasi MySQL.

Saat menggunakan Klien MySQL 5.7 atau MySQL 8.0, koneksi SSL memerlukan verifikasi terhadap sertifikat CA server jika, untuk opsi `--ssl-mode`, Anda menentukan `VERIFY_CA` atau `VERIFY_IDENTITY`, seperti pada contoh berikut.

```
mysql -h mysql-database.rds.amazonaws.com -uadmin -ppassword --ssl-ca=/tmp/ssl-cert.pem
 --ssl-mode=VERIFY_CA
```

Saat menggunakan Klien MySQL 5.6, koneksi SSL memerlukan verifikasi terhadap sertifikat CA server jika Anda menentukan opsi `--ssl-verify-server-cert`, seperti pada contoh berikut.

```
mysql -h mysql-database.rds.amazonaws.com -uadmin -ppassword --ssl-ca=/tmp/ssl-cert.pem
 --ssl-verify-server-cert
```

Memperbarui penyimpanan kepercayaan aplikasi Anda

Untuk informasi tentang pembaruan penyimpanan kepercayaan untuk aplikasi MySQL, lihat Menggunakan TLS/SSL dengan MariaDB Connector/J di dokumentasi MariaDB.

Untuk informasi tentang cara mengunduh sertifikat root, lihat .

Untuk contoh skrip yang mengimpor sertifikat, lihat Contoh skrip untuk mengimpor sertifikat ke trust store Anda.

> **Note**
>
> Saat memperbarui penyimpanan kepercayaan, Anda dapat mempertahankan sertifikat lama selain menambahkan sertifikat baru.

Jika Anda menggunakan driver JDBC MariaDB Connector/J dalam aplikasi, atur properti berikut dalam aplikasi.

```
System.setProperty("javax.net.ssl.trustStore", certs);
System.setProperty("javax.net.ssl.trustStorePassword", "password");
```

Saat Anda memulai aplikasi, atur properti berikut.

```
java -Djavax.net.ssl.trustStore=/path_to_truststore/MyTruststore.jks -
Djavax.net.ssl.trustStorePassword=my_truststore_password com.companyName.MyApplication
```

> **Note**
>
> Tentukan kata sandi selain prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

Contoh kode Java untuk membangun koneksi SSL

Contoh kode berikut menunjukkan cara menyiapkan koneksi SSL menggunakan JDBC.

```
private static final String DB_USER = "admin";

        private static final String DB_USER = "user name";
        private static final String DB_PASSWORD = "password";
        // This key store has only the prod root ca.
        private static final String KEY_STORE_FILE_PATH = "file-path-to-keystore";
        private static final String KEY_STORE_PASS = "keystore-password";
```

```
    public static void main(String[] args) throws Exception {
        Class.forName("org.mariadb.jdbc.Driver");

        System.setProperty("javax.net.ssl.trustStore", KEY_STORE_FILE_PATH);
        System.setProperty("javax.net.ssl.trustStorePassword", KEY_STORE_PASS);

        Properties properties = new Properties();
        properties.put("user", DB_USER);
        properties.put("password", DB_PASSWORD);


        Connection connection = DriverManager.getConnection("jdbc:mysql://ssl-mariadb-
public.cni62e2e7kwh.us-east-1.rds.amazonaws.com:3306?useSSL=true",properties);
        Statement stmt=connection.createStatement();

        ResultSet rs=stmt.executeQuery("SELECT 1 from dual");

        return;
    }
```

Important

Setelah Anda menentukan bahwa koneksi database Anda menggunakan SSL/TLS dan telah memperbarui toko kepercayaan aplikasi Anda, Anda dapat memperbarui database Anda untuk menggunakan sertifikat 2048-g1. rds-ca-rsa Untuk mengetahui petunjuknya, lihat langkah 3 dalam Memperbarui sertifikat CA Anda dengan memodifikasi instans atau cluster DB.

Tentukan kata sandi selain prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

Meningkatkan performa kueri RDS for MariaDB dengan Amazon RDS Optimized Reads

Anda dapat mempercepat pemrosesan kueri untuk RDS for MariaDB dengan Amazon RDS Optimized Reads. Instans DB RDS for MariaDB yang menggunakan RDS Optimized Reads dapat memproses kueri hingga 2x lebih cepat dibandingkan dengan instans DB yang tidak menggunakannya.

Topik

Ikhtisar RDS Optimized Reads

Saat Anda menggunakan instans DB RDS for MariaDB yang mengaktifkan RDS Optimized Reads, performa kueri instans DB Anda akan lebih cepat melalui penggunaan penyimpanan instans. Penyimpanan instans menyediakan penyimpanan tingkat blok sementara untuk instans Anda. Penyimpanan terletak di solid state drive (SSD) Non-Volatile Memory Express (NVMe) yang secara fisik terpasang ke server host. Penyimpanan ini dioptimalkan untuk latensi rendah, performa I/O acak tinggi, dan throughput baca berurutan tinggi.

RDS Optimized Reads diaktifkan secara default ketika instans DB menggunakan kelas instans DB dengan penyimpanan instans, seperti db.m5d atau db.m6gd. Dengan RDS Optimized Reads, beberapa objek sementara disimpan di penyimpanan instans. Objek sementara ini termasuk file sementara internal, tabel sementara internal pada disk, file peta memori, dan file cache log biner (binlog). Untuk informasi selengkapnya tentang penyimpanan instans, lihat Penyimpanan instans Amazon EC2 dalam Panduan Pengguna Amazon Elastic Compute Cloud untuk Instans Linux.

Beban kerja yang menghasilkan objek sementara di MariaDB untuk pemrosesan kueri dapat memanfaatkan penyimpanan instans untuk mempercepat pemrosesan kueri. Jenis beban kerja ini mencakup kueri yang melibatkan pengurutan, agregasi hash, penggabungan dengan beban tinggi,

Ekspresi Tabel Umum (CTE), dan kueri pada kolom yang tidak diindeks. Volume penyimpanan instans ini memberikan IOPS dan performa yang lebih tinggi, terlepas dari konfigurasi penyimpanan yang digunakan untuk penyimpanan Amazon EBS secara persisten. Karena RDS Optimized Reads memindahkan beban operasi pada objek sementara ke penyimpanan instans, operasi input/output per detik (IOPS) atau throughput penyimpanan persisten (Amazon EBS) kini dapat digunakan untuk operasi pada objek persisten. Operasi ini mencakup pembacaan dan penulisan file data biasa, dan operasi mesin latar belakang, seperti flushing dan penggabungan buffer sisipan.

> **Note**
>
> Snapshot RDS manual dan otomatis hanya berisi file mesin untuk objek persisten. Objek sementara yang dibuat di penyimpanan instans tidak disertakan dalam snapshot RDS.

Kasus penggunaan RDS Optimized Reads

Jika Anda memiliki beban kerja yang sangat bergantung pada objek sementara, seperti tabel atau file internal, untuk eksekusi kueri, Anda dapat memperoleh manfaat dengan mengaktifkan RDS Optimized Reads. Kasus penggunaan berikut ini adalah kandidat untuk RDS Optimized Reads:

- Aplikasi yang menjalankan kueri analitis dengan ekspresi tabel umum (CTE) yang kompleks, tabel turunan, dan operasi pengelompokan
- Replika baca yang melayani lalu lintas baca padat dengan kueri yang tidak dioptimalkan
- Aplikasi yang menjalankan kueri pelaporan berdasarkan permintaan atau dinamis yang melibatkan operasi yang kompleks, seperti kueri dengan klausa `GROUP BY` dan `ORDER BY`
- Beban kerja yang menggunakan tabel sementara internal untuk pemrosesan kueri

  Anda dapat memantau variabel status mesin `created_tmp_disk_tables` untuk menentukan jumlah tabel sementara berbasis disk yang dibuat pada instans DB Anda.
- Aplikasi yang membuat tabel sementara dalam jumlah besar, baik secara langsung maupun dalam prosedur, untuk menyimpan hasil sementara
- Kueri basis data yang melakukan pengelompokan atau pengurutan pada kolom yang tidak diindeks

Praktik terbaik RDS Optimized Reads

Gunakan praktik terbaik RDS Optimized Reads berikut:

- Tambahkan logika coba lagi untuk kueri hanya baca jika terjadi kegagalan karena penyimpanan instans penuh selama eksekusi.
- Pantau ruang penyimpanan yang tersedia di penyimpanan instans dengan CloudWatch metrik`FreeLocalStorage`. Jika penyimpanan instans hampir penuh karena beban kerja pada instans DB, modifikasi instans DB untuk menggunakan kelas instans DB yang lebih besar.
- Jika memori instans DB sudah memadai tetapi masih mencapai batas penyimpanan pada penyimpanan instans, tingkatkan nilai `binlog_cache_size` untuk mempertahankan entri binlog khusus sesi dalam memori. Konfigurasi ini akan mencegah penulisan entri binlog ke file cache binlog sementara pada disk.

  Parameter `binlog_cache_size` dibuat per sesi. Anda dapat mengubah nilai untuk setiap sesi baru. Pengaturan untuk parameter ini dapat meningkatkan pemanfaatan memori pada instans DB selama beban kerja mencapai puncaknya. Oleh karena itu, pertimbangkan untuk meningkatkan nilai parameter berdasarkan pola beban kerja aplikasi Anda dan memori yang tersedia pada instans DB.
- Gunakan nilai default `MIXED` untuk `binlog_format`. Tergantung ukuran transaksi, mengatur `binlog_format` ke `ROW` dapat menghasilkan file cache binlog berukuran besar pada penyimpanan instans.
- Jangan melakukan perubahan besar-besaran dalam satu transaksi. Transaksi seperti ini dapat menghasilkan file cache binlog berukuran besar pada penyimpanan instans dan dapat menyebabkan masalah ketika penyimpanan instans penuh. Pertimbangkan untuk membagi penulisan menjadi beberapa transaksi kecil guna meminimalkan penggunaan penyimpanan untuk file cache binlog.

Menggunakan RDS Optimized Reads

Saat Anda menyediakan instans DB RDS for MariaDB dengan salah satu kelas instans DB berikut dalam deployment instans DB Satu AZ atau deployment instans DB Multi-AZ, instans DB akan otomatis menggunakan RDS Optimized Reads.

Untuk mengaktifkan RDS Optimized Reads, lakukan salah satu tindakan berikut:

- Buat instans DB RDS for MariaDB menggunakan salah satu kelas instans DB berikut. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.
- Modifikasi instans DB RDS for MariaDB untuk menggunakan salah satu kelas instans DB berikut. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

RDS Optimized Reads tersedia di semua Wilayah AWS tempat satu atau lebih kelas instans DB dengan penyimpanan SSD NVMe lokal didukung. Untuk informasi tentang kelas instans DB, lihat the section called “Kelas instans DB”.

Ketersediaan kelas instans DB berbeda untuk Wilayah AWS. Untuk menentukan apakah kelas instans DB didukung secara spesifik Wilayah AWS, lihatthe section called “Menentukan dukungan kelas instans DB di Wilayah AWS”.

Jika Anda tidak ingin menggunakan RDS Optimized Reads, modifikasi instans DB Anda agar tidak menggunakan kelas instans DB yang mendukung fitur tersebut.

Memantau instans DB yang menggunakan RDS Optimized Reads

Anda dapat memantau instans DB yang menggunakan Bacaan yang Dioptimalkan RDS dengan metrik berikut: CloudWatch

- `FreeLocalStorage`
- `ReadIOPSLocalStorage`
- `ReadLatencyLocalStorage`
- `ReadThroughputLocalStorage`
- `WriteIOPSLocalStorage`
- `WriteLatencyLocalStorage`
- `WriteThroughputLocalStorage`

Metrik ini menyediakan data tentang penyimpanan instans, IOPS, dan throughput yang tersedia. Untuk informasi selengkapnya tentang metrik ini, lihat Metrik CloudWatch tingkat instans Amazon untuk Amazon RDS.

Batasan RDS Optimized Reads

Batasan berikut berlaku untuk RDS Optimized Reads:

- RDS Optimized Reads didukung untuk versi RDS for MariaDB berikut:
  - 10.11.4 dan versi 10.11 yang lebih tinggi
  - 10.6.7 dan versi 10.6 yang lebih tinggi
  - 10.5.16 dan versi 10.5 yang lebih tinggi

  - 10.4.25 dan versi 10.4 yang lebih tinggi

  Untuk informasi tentang versi RDS for MariaDB, lihat Versi-versi MariaDB pada Amazon RDS.
- Anda tidak dapat mengubah lokasi objek sementara ke penyimpanan persisten (Amazon EBS) pada kelas instans DB yang mendukung RDS Optimized Reads.
- Saat pencatatan log biner pada instans DB diaktifkan, ukuran transaksi maksimum dibatasi oleh ukuran penyimpanan instans. Di MariaDB, setiap sesi yang membutuhkan penyimpanan lebih banyak daripada nilai `binlog_cache_size` transaksi penulisan akan berubah menjadi file cache binlog sementara, yang dibuat pada penyimpanan instans.
- Transaksi dapat gagal ketika penyimpanan instans penuh.

Meningkatkan performa penulisan dengan Amazon RDS Optimized Writes for MariaDB

Anda dapat meningkatkan performa transaksi penulisan dengan RDS Optimized Writes for MariaDB. Ketika basis data RDS for MariaDB Anda menggunakan RDS Optimized Writes, RDS dapat mencapai throughput transaksi penulisan hingga dua kali lebih tinggi.

Topik

Ikhtisar RDS Optimized Writes

Saat RDS Optimized Writes diaktifkan, basis data RDS for MariaDB Anda hanya menulis sekali ketika melakukan flashing data ke penyimpanan yang kuat tanpa memerlukan buffer penulisan ganda. basis data terus melindungi properti ACID untuk menghasilkan transaksi basis data yang andal dan meningkatkan performa.

basis data relasional, seperti MariaDB, menyediakan properti ACID seperti atomisitas, konsistensi, isolasi, dan daya tahan untuk transaksi basis data yang andal. Untuk membantu menyediakan properti ini, MariaDB menggunakan tempat penyimpanan data yang disebut buffer penulisan ganda guna mencegah kesalahan penulisan sebagian halaman. Kesalahan ini terjadi ketika terjadi kegagalan perangkat keras saat basis data sedang memperbarui halaman, seperti dalam kasus pemadaman listrik. basis data MariaDB dapat mendeteksi penulisan sebagian halaman dan memulihkan dengan salinan halaman di buffer penulisan ganda. Selain memberikan perlindungan, teknik ini juga menambah jumlah operasi penulisan. Untuk informasi selengkapnya tentang buffer penulisan ganda MariaDB, lihat Buffer Penulisan Ganda InnoDB dalam dokumentasi MariaDB.

Dengan mengaktifkan RDS Optimized Writes, basis data RDS for MariaDB hanya menulis sekali ketika melakukan flashing data ke penyimpanan tahan lama tanpa menggunakan buffer penulisan ganda. RDS Optimized Writes berguna jika Anda menjalankan beban kerja penulisan berat pada basis data RDS for MariaDB. Contoh basis data dengan beban kerja penulisan berat antara lain basis data yang mendukung pembayaran digital, perdagangan finansial, dan aplikasi game.

basis data ini dijalankan pada kelas instans DB yang menggunakan AWS Nitro System. Karena konfigurasi perangkat keras dalam sistem ini, basis data dapat menulis halaman 16-KiB secara langsung ke file data secara andal dan kuat dalam satu langkah. AWS Nitro System mendukung RDS Optimized Writes.

Anda dapat mengatur `rds.optimized_writes` parameter basis data baru untuk mengontrol fitur RDS Optimized Writes untuk basis data RDS for MariaDB. Akses parameter ini di grup parameter DB pada RDS for MariaDB untuk versi berikut:

- 10.11.4 dan versi 10.11 yang lebih tinggi
- 10.6.10 dan versi 10.6 yang lebih tinggi

Tetapkan parameter menggunakan nilai berikut:

- `AUTO` – Aktifkan RDS Optimized Writes jika didukung oleh basis data. Nonaktifkan RDS Optimized Writes jika tidak didukung basis data. Ini adalah pengaturan default.
- `OFF` – Nonaktifkan RDS Optimized Writes meski didukung oleh basis data.

Jika Anda memigrasikan basis data RDS for MariaDB yang dikonfigurasi untuk menggunakan RDS Optimized Writes ke kelas instans DB yang tidak mendukung fitur tersebut, RDS secara otomatis menonaktifkan RDS Optimized Writes untuk basis data tersebut.

Saat RDS Optimized Writes dinonaktifkan, basis data akan menggunakan buffer penulisan ganda MariaDB.

Untuk menentukan apakah basis data RDS for MariaDB menggunakan RDS Optimized Writes, lihat nilai saat ini parameter `innodb_doublewrite` basis data. Jika basis data menggunakan RDS Optimized Writes, parameter ini diatur ke `FALSE` (`0`).

Menggunakan RDS Optimized Writes

Anda dapat mengaktifkan RDS Optimized Writes saat membuat basis data RDS for MariaDB dengan konsol RDS, AWS CLI, atau API RDS. RDS Optimized Writes diaktifkan secara otomatis ketika kedua kondisi berikut berlaku selama pembuatan basis data:

- Anda menentukan versi mesin DB dan kelas instans DB yang mendukung RDS Optimized Writes.
  - RDS Optimized Writes didukung untuk versi RDS for MariaDB berikut:
    - 10.11.4 dan versi 10.11 yang lebih tinggi

- 10.6.10 dan versi 10.6 yang lebih tinggi

  Untuk informasi tentang versi RDS for MariaDB, lihat Versi-versi MariaDB pada Amazon RDS.

- RDS Optimized Writes didukung untuk basis data RDS for MariaDB yang menggunakan kelas instans DB berikut:
  - db.m7g
  - db.m6g
  - db.m6gd
  - db.m6i
  - db.m5
  - db.m5d
  - db.r7g
  - db.r6g
  - db.r6gd
  - db.r6i
  - db.r5
  - db.r5b
  - db.r5d
  - db.x2idn
  - db.x2iedn

  Untuk informasi tentang kelas instans DB, lihat the section called “Kelas instans DB”.

  Ketersediaan kelas instans DB untuk Wilayah AWS berbeda-beda. Untuk mengetahui dukungan kelas instans DB pada suatu Wilayah AWS, lihat the section called “Menentukan dukungan kelas instans DB di Wilayah AWS”.

- Dalam grup parameter yang terkait dengan basis data, parameter `rds.optimized_writes` diatur ke `AUTO`. Dalam grup parameter default, parameter ini selalu diatur ke `AUTO`.

Jika Anda ingin menggunakan versi mesin DB dan kelas instans DB yang mendukung RDS Optimized Writes, tetapi tidak ingin menggunakan fitur ini, tentukan grup parameter khusus saat Anda membuat basis data. Dalam grup parameter ini, atur parameter `rds.optimized_writes` ke `OFF`. Agar nantinya basis data menggunakan RDS Optimized Writes, Anda dapat mengatur parameter

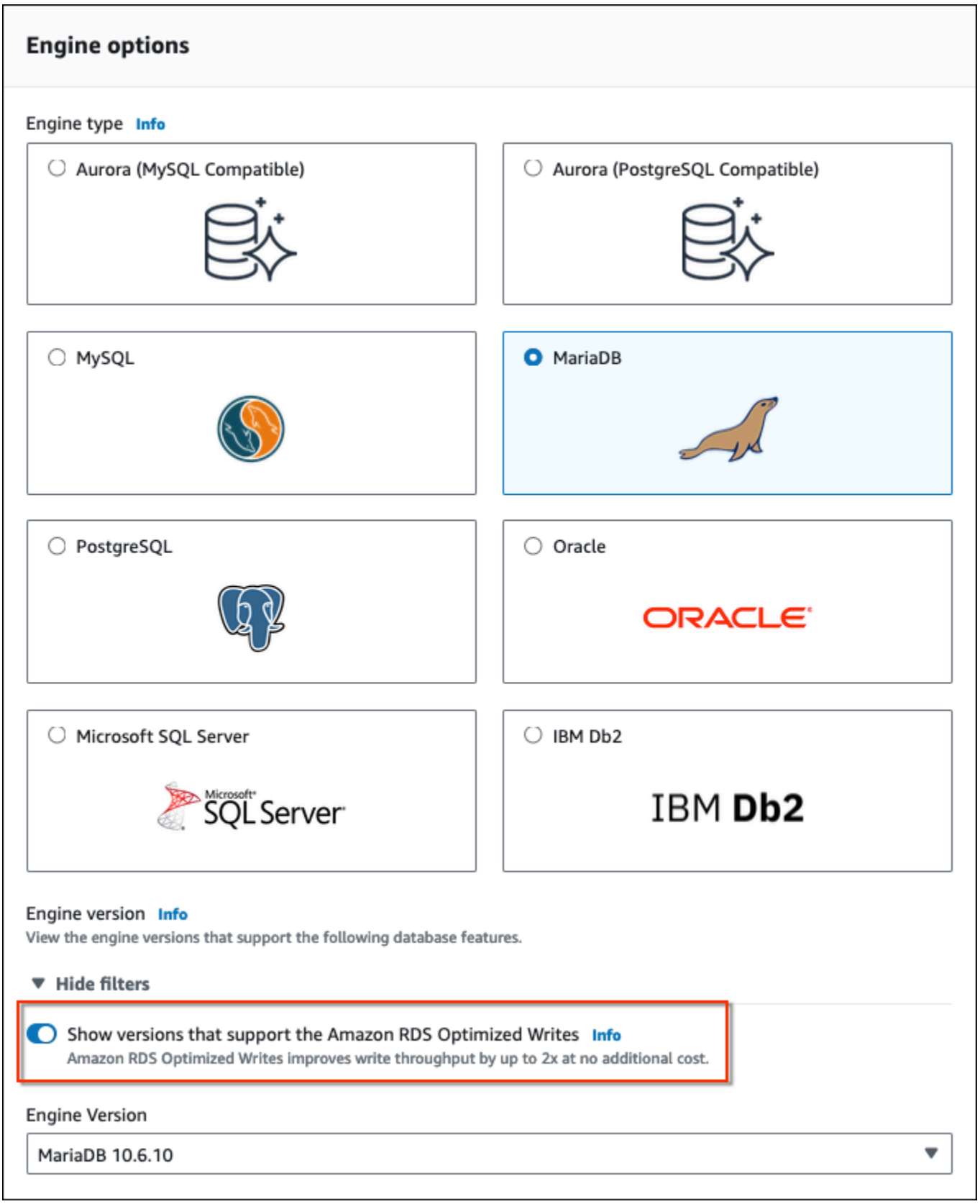
ke AUT0 untuk mengaktifkannya. Untuk informasi tentang pembuatan grup parameter khusus dan pengaturan parameter, lihat Bekerja dengan grup parameter.

Untuk informasi tentang pembuatan instans DB, lihat Membuat instans DB Amazon RDS.

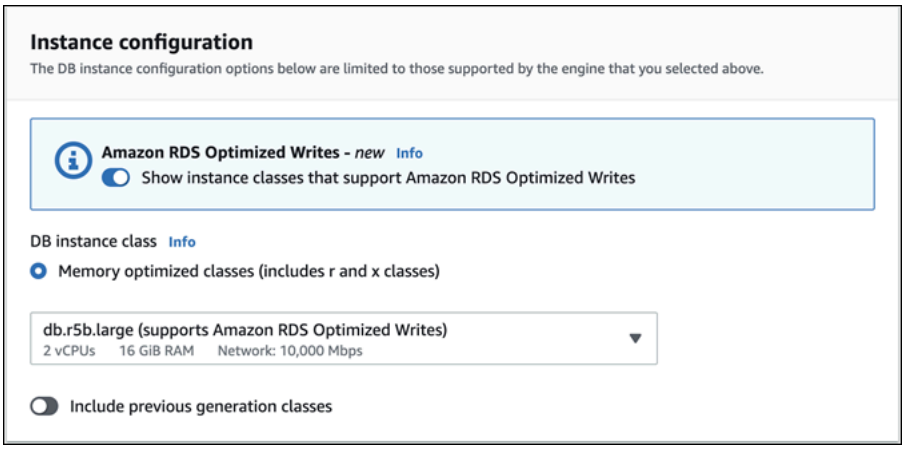
Konsol

Saat menggunakan konsol RDS untuk membuat basis data RDS for MariaDB, Anda dapat memfilter versi mesin DB dan kelas instans DB yang mendukung RDS Optimized Writes. Setelah mengaktifkan filter, Anda dapat memilih versi mesin DB dan kelas instans DB yang tersedia.

Untuk memilih versi mesin DB yang mendukung RDS Optimized Writes, filter versi mesin DB RDS for MariaDB yang mendukungnya di Versi mesin, lalu pilih versi.


Engine options
Engine type Info
Aurora (MySQL Compatible)
Aurora (PostgreSQL Compatible)
MySQL
MariaDB
PostgreSQL
Oracle
ORACLE
Microsoft SQL Server
Microsoft SQL Server
IBM Db2
IBM Db2
Engine version Info
View the engine versions that support the following database features.
Hide filters
Show versions that support the Amazon RDS Optimized Writes Info
Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.
Engine Version
MariaDB 10.6.10

Di bagian Konfigurasi instans, gunakan filter untuk menemukan kelas instans DB yang mendukung RDS Optimized Writes, lalu pilih kelas instans DB.



Setelah menentukan pilihan ini, Anda dapat memilih pengaturan lain sesuai kebutuhan dan menyelesaikan pembuatan basis data RDS for MariaDB dengan konsol.

AWS CLI

Untuk membuat instance DB dengan menggunakanAWS CLI, gunakan create-db-instanceperintah. Pastikan nilai `--engine-version` dan `--db-instance-class` mendukung RDS Optimized Writes. Selain itu, pastikan parameter `rds.optimized_writes` untuk grup parameter yang terkait dengan instans DB telah diatur ke AUT0. Contoh ini mengaitkan grup parameter default dengan instans DB.

Example Membuat instans DB yang menggunakan RDS Optimized Writes

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --engine mariadb \
    --engine-version 10.6.10 \
    --db-instance-class db.r5b.large \
    --manage-master-user-password \
    --master-username admin \
    --allocated-storage 200
```

Untuk Windows:

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

```
    --engine mariadb ^
    --engine-version 10.6.10 ^
    --db-instance-class db.r5b.large ^
    --manage-master-user-password ^
    --master-username admin ^
    --allocated-storage 200
```

API RDS

Anda dapat membuat instans DB menggunakan operasi CreateDBInstance. Saat Anda menggunakan operasi ini, pastikan nilai `EngineVersion` dan `DBInstanceClass` mendukung RDS Optimized Writes. Selain itu, pastikan parameter `rds.optimized_writes` untuk grup parameter yang terkait dengan instans DB telah diatur ke `AUTO`.

Mengaktifkan RDS Optimized Writes pada basis data yang sudah ada

Untuk mengubah basis data RDS for MariaDB yang sudah ada untuk mengaktifkan RDS Optimized Writes, basis data harus dibuat dengan versi mesin DB dan kelas instans DB yang didukung. Selain itu, basis data harus sudah dibuat setelah RDS Optimized Writes dirilis pada 7 Maret 2023, karena konfigurasi sistem file yang diperlukan tidak kompatibel dengan basis data yang dibuat sebelum dirilis. Jika kondisi ini terpenuhi, Anda dapat mengaktifkan RDS Optimized Writes dengan mengatur parameter `rds.optimized_writes` ke `AUTO`.

Jika basis data Anda tidak dibuat dengan versi mesin, kelas instans, atau konfigurasi sistem file yang didukung, Anda dapat menggunakan Deployment Blue/Green RDS untuk bermigrasi ke konfigurasi yang didukung. Sambil membuat deployment blue/green, lakukan hal berikut:

- Pilih Aktifkan Optimized Writes pada basis data hijau, lalu tentukan versi mesin dan kelas instans DB yang mendukung RDS Optimized Writes. Untuk daftar versi mesin dan kelas instans yang didukung, lihat the section called “Menggunakan dengan basis data baru”.
- Di bagian Penyimpanan, pilih Tingkatkan konfigurasi sistem file penyimpanan. Opsi ini meningkatkan basis data ke konfigurasi sistem file dasar yang kompatibel.

Saat Anda membuat deployment blue/green, jika parameter `rds.optimized_writes` diatur ke `AUTO`, RDS Optimized Writes akan secara otomatis diaktifkan pada lingkungan hijau. Anda kemudian dapat beralih antara deployment blue/green, yang mendukung lingkungan hijau sebagai lingkungan produksi yang baru.

Untuk informasi selengkapnya, lihat the section called “Membuat deployment blue/green”.

Batasan RDS Optimized Writes

Saat Anda memulihkan basis data RDS for MariaDB dari snapshot, Anda hanya bisa mengaktifkan RDS Optimized Writes untuk basis data jika semua kondisi berikut terpenuhi:

- Snapshot dibuat dari basis data yang mendukung RDS Optimized Writes.
- Snapshot dibuat dari basis data yang dibuat setelah RDS Optimized Writes dirilis.
- Snapshot dikembalikan ke basis data yang mendukung RDS Optimized Writes.
- basis data yang dipulihkan berkaitan dengan grup parameter yang parameter `rds.optimized_writes`-nya diatur ke `AUTO`.

Meningkatkan mesin DB MariaDB

Ketika Amazon RDS mendukung versi baru mesin basis data, Anda dapat meningkatkan instans DB Anda ke versi baru. Ada dua jenis peningkatan untuk instans DB MariaDB: peningkatan versi mayor dan versi minor.

Peningkatan versi mayor dapat berisi perubahan basis data yang tidak memiliki kompatibilitas mundur dengan aplikasi yang ada. Oleh karena itu, Anda harus melakukan peningkatan versi mayor untuk instans DB Anda secara manual. Anda dapat memulai peningkatan versi mayor dengan mengubah instans DB Anda. Namun, sebelum Anda melakukan peningkatan versi mayor, kami sarankan agar Anda mengikuti petunjuk dalam Peningkatan versi mayor untuk MariaDB.

Sebaliknya, tingkatkan versi minor hanya menyertakan perubahan yang kompatibel dengan aplikasi yang ada. Anda dapat memulai peningkatan versi minor secara manual dengan memodifikasi instans DB Anda. Atau Anda dapat mengaktifkan opsi Peningkatan versi minor otomatis saat membuat atau memodifikasi instans DB. Tindakan ini akan membuat instans DB Anda secara otomatis ditingkatkan setelah pengujian Amazon RDS dan menyetujui versi baru. Untuk informasi tentang melakukan peningkatan, lihat Meng-upgrade versi mesin instans DB.

Jika instans DB MariaDB Anda menggunakan replika baca, Anda harus meningkatkan semua replika baca sebelum meningkatkan instans sumber. Jika instans DB Anda ada dalam deployment Multi-AZ, replika penulis dan siaga akan ditingkatkan. Instans DB Anda mungkin tidak tersedia hingga peningkatan selesai.

Untuk informasi selengkapnya tentang versi yang didukung MariaDB dan manajemen versi, lihat Versi-versi MariaDB pada Amazon RDS.

Peningkatan mesin basis data memerlukan waktu henti. Durasi waktu henti bervariasi berdasarkan ukuran instans DB Anda.

> **Tip**
>
> Anda dapat meminimalkan waktu henti yang diperlukan untuk peningkatan instans DB dengan menggunakan deployment blue/green. Untuk informasi selengkapnya, lihat Menggunakan Deployment Blue/Green Amazon RDS untuk pembaruan basis data.

Topik

- Gambaran umum peningkatan

- Nomor versi MariaDB
- Nomor versi RDS
- Peningkatan versi mayor untuk MariaDB
- Meningkatkan instans DB MariaDB
- Peningkatan versi minor otomatis untuk MariaDB
- Menggunakan replika baca untuk mengurangi waktu henti saat meningkatkan basis data MariaDB

Gambaran umum peningkatan

Saat Anda menggunakan AWS Management Console untuk memutakhirkan instans DB, ini menunjukkan target pemutakhiran yang valid untuk instans DB. Anda juga dapat menggunakan AWS CLI perintah berikut untuk mengidentifikasi target pemutakhiran yang valid untuk instans DB:

Untuk Linux, macOS, atau Unix:

```
aws rds describe-db-engine-versions \
  --engine mariadb \
  --engine-version version-number \
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Untuk Windows:

```
aws rds describe-db-engine-versions ^
  --engine mariadb ^
  --engine-version version-number ^
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Misalnya, untuk mengidentifikasi target pemutakhiran yang valid untuk instance MariaDB versi 10.5.17 DB, jalankan perintah berikut: AWS CLI

Untuk Linux, macOS, atau Unix:

```
aws rds describe-db-engine-versions \
  --engine mariadb \
  --engine-version 10.5.17 \
```

```
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Untuk Windows:

```
aws rds describe-db-engine-versions ^
  --engine mariadb ^
  --engine-version 10.5.17 ^
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Amazon RDS mengambil dua atau lebih snapshot DB selama proses peningkatan. Amazon RDS mengambil hingga dua snapshot dari instans DB sebelum melakukan perubahan peningkatan. Jika peningkatan tidak berfungsi untuk basis data Anda, Anda dapat memulihkan salah satu snapshot ini untuk membuat instans DB yang menjalankan versi lama. Amazon RDS mengambil snapshot lain dari instans DB saat peningkatan selesai. Amazon RDS mengambil snapshot ini terlepas dari apakah AWS Backup mengelola cadangan untuk instans DB.

> **Note**
>
> Amazon RDS hanya mengambil snapshot DB jika Anda telah mengatur periode retensi cadangan untuk instans DB Anda ke angka yang lebih besar dari 0. Untuk mengubah periode retensi cadangan Anda, lihat Memodifikasi instans DB Amazon RDS.

Setelah peningkatan selesai, Anda tidak dapat kembali ke versi mesin basis data sebelumnya. Jika Anda ingin kembali ke versi yang lebih lama, pulihkan snapshot DB pertama yang diambil untuk membuat instans DB baru.

Anda mengontrol waktu untuk meningkatkan instans DB Anda ke versi baru yang didukung oleh Amazon RDS. Tingkat kontrol ini membantu Anda menjaga kompatibilitas dengan versi basis data spesifik dan menguji versi baru untuk aplikasi Anda sebelum menerapkannya dalam produksi. Saat Anda siap, Anda dapat melakukan peningkatan versi pada waktu yang paling cocok dengan jadwal Anda.

Jika instans DB Anda menggunakan replikasi baca, Anda harus meningkatkan semua replikasi baca sebelum meningkatkan instans sumber.

Jika instans DB Anda ada dalam deployment Multi-AZ, instans DB primer dan siaga akan ditingkatkan. Instans DB primer dan siaga ditingkatkan pada saat yang sama dan Anda akan

mengalami pemadaman hingga peningkatan selesai. Waktu pemadaman bervariasi berdasarkan mesin basis data, versi mesin, dan ukuran instans DB Anda.

Nomor versi MariaDB

Urutan penomoran versi untuk RDS untuk mesin database MariaDB baik dalam bentuk major.minor.patch.yyyymmdd atau major.minor.patch, misalnya, 10.11.5.R2.20231201 atau 10.4.30. Format yang digunakan tergantung pada versi mesin MariaDB.

mayor

Nomor versi utama adalah bilangan bulat dan bagian fraksional pertama dari nomor versi, misalnya, 10.11. Peningkatan versi mayor akan meningkatkan bagian mayor dari nomor versi. Misalnya, upgrade dari 10.5 .20 ke 10.6.12 adalah upgrade versi utama, di mana 10.5 dan 10.6 adalah nomor versi utama.

kecil

Nomor versi minor adalah bagian ketiga dari nomor versi, misalnya, 5 di 10.11.5.

tambalan

Patch adalah bagian keempat dari nomor versi, misalnya, R2 di 10.11.5.R2. Versi patch RDS mencakup perbaikan bug penting yang ditambahkan ke versi minor setelah dirilis.

YYYMMDD

Tanggal adalah bagian kelima dari nomor versi, misalnya, 20231201 di 10.11.5.R2.20231201. Versi tanggal RDS adalah patch keamanan yang mencakup perbaikan keamanan penting yang ditambahkan ke versi minor setelah dirilis. Itu tidak termasuk perbaikan apa pun yang mungkin mengubah perilaku mesin.

Versi utama	Versi minor	Skema penamaan
10.11	≥ 5	Instans DB baru menggunakan major.minor.patch.YYMMDD, misalnya, 10.11.5.R2.20231201. Instans DB yang ada mungkin menggunakan major.minor.patch, misalnya, 10.11.5.R2,

Versi utama	Versi minor	Skema penamaan
		hingga versi mayor atau minor Anda berikutnya ditingkatkan.
	< 5	Instans DB yang ada menggunakan major.minor.patch, misalnya, 10.11.4.R2.
10.6	≥ 14	Instans DB baru menggunakan major.minor.patch.YYMMDD, misalnya, 10.6.14.R2.20231201. Instans DB yang ada mungkin menggunakan major.minor.patch, misalnya, 10.6.14.R2, hingga versi mayor atau minor Anda berikutnya ditingkatkan.
	< 14	Instans DB yang ada menggunakan major.minor.patch, misalnya, 10.6.13.R2.
10.5	≥ 21	Instans DB baru menggunakan major.minor.patch.YYMMDD, misalnya, 10.5.21.R2.20231201. Instans DB yang ada mungkin menggunakan major.minor.patch, misalnya, 10.5.21.R2, hingga peningkatan versi mayor atau minor berikutnya.
	< 21	Instans DB yang ada menggunakan major.minor.patch, misalnya, 10.5.20.R2.

Versi utama	Versi minor	Skema penamaan
10.4	≥ 30	Instans DB baru menggunakan major.minor.patch.YYMMDD, misalnya, 10.4.30.R2.20231201. Instans DB yang ada mungkin menggunakan major.minor.patch, misalnya, 10.4.30.R2, hingga versi mayor atau minor Anda berikutnya ditingkatkan.
	< 30	Instans DB yang ada menggunakan major.minor.patch, misalnya, 10.4.29.R2.

Nomor versi RDS

Nomor versi RDS menggunakan skema *major*.*minor*.*patch* atau *major*.*minor*.*patch*.*YYYYMMDD* penamaan. Versi patch RDS mencakup perbaikan bug penting yang ditambahkan ke versi minor setelah dirilis. Versi tanggal RDS (*YYMMDD*) adalah patch keamanan. Patch keamanan tidak menyertakan perbaikan apa pun yang dapat mengubah perilaku mesin.

Untuk mengidentifikasi nomor versi Amazon RDS untuk basis data Anda, Anda harus terlebih dahulu membuat ekstensi `rds_tools` dengan menggunakan perintah berikut:

```
CREATE EXTENSION rds_tools;
```

Anda dapat mengetahui nomor versi RDS dari database RDS untuk MariaDB Anda dengan kueri SQL berikut:

```
mysql> select mysql.rds_version();
```

Misalnya, query RDS untuk MariaDB 10.6.14 database mengembalikan output berikut:

```
+---------------------+
| mysql.rds_version() |
+---------------------+
```

```
| 10.6.14.R2.20231201  |
+----------------------+
1 row in set (0.01 sec)
```

Peningkatan versi mayor untuk MariaDB

Peningkatan versi mayor dapat berisi perubahan basis data yang tidak memiliki kompatibilitas mundur dengan aplikasi yang ada. Akibatnya, Amazon RDS tidak menerapkan peningkatan versi mayor secara otomatis. Anda harus memodifikasi instans DB Anda secara manual. Kami menyarankan Anda untuk menguji peningkatan apa pun secara menyeluruh sebelum menerapkannya ke instans produksi Anda.

Amazon RDS mendukung peningkatan di tempat berikut untuk versi mayor mesin basis data MariaDB:

- Versi MariaDB mana pun ke MariaDB 10.11
- Versi MariaDB mana pun ke MariaDB 10.6
- MariaDB 10.4 ke MariaDB 10.5
- MariaDB 10.3 ke MariaDB 10.4

Untuk melakukan peningkatan versi mayor ke versi MariaDB yang lebih rendah dari 10.6, tingkatkan ke setiap versi mayor secara berurutan. Misalnya, untuk meningkatkan dari versi 10.3 ke versi 10.5, tingkatkan dengan urutan sebagai berikut: 10.3 ke 10.4 lalu 10.4 ke 10.5.

Jika Anda menggunakan grup parameter kustom, dan melakukan peningkatan versi mayor, Anda harus menentukan grup parameter default untuk versi mesin DB baru atau membuat grup parameter kustom Anda sendiri untuk versi mesin DB baru. Pengaitan grup parameter baru dengan instans DB akan memerlukan boot ulang basis data yang dimulai oleh pelanggan setelah peningkatan selesai. Status grup parameter instans akan menampilkan `pending-reboot` jika instans perlu di-boot ulang untuk menerapkan perubahan grup parameter. Status grup parameter suatu instans dapat dilihat di AWS Management Console atau dengan menggunakan panggilan "describe" seperti `describe-db-instances`.

Meningkatkan instans DB MariaDB

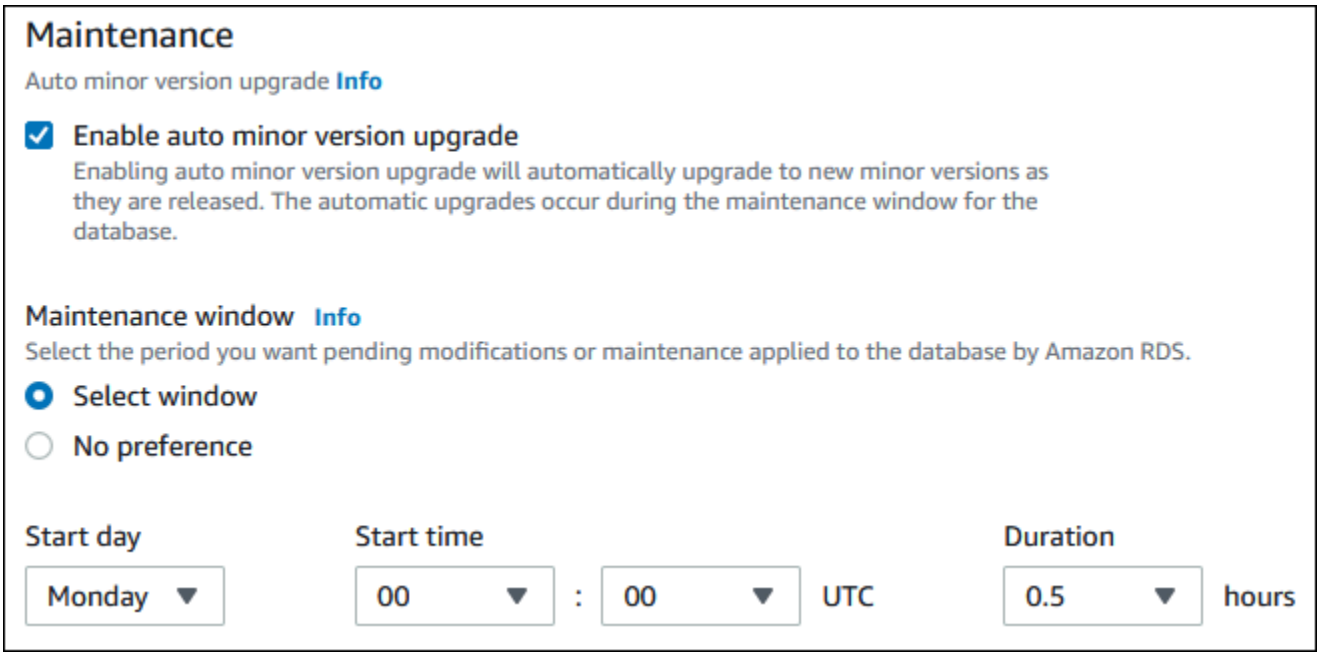
Untuk informasi tentang peningkatan instans DB MariaDB secara manual atau otomatis, lihat Meng-upgrade versi mesin instans DB.

Peningkatan versi minor otomatis untuk MariaDB

Jika Anda menentukan pengaturan berikut saat membuat atau memodifikasi instans DB, Anda dapat melakukan peningkatan instans DB secara otomatis.

- Pengaturan Peningkatan versi minor otomatis diaktifkan.
- Pengaturan Periode retensi cadangan lebih besar dari 0.

Di AWS Management Console, pengaturan ini berada di bawah Konfigurasi tambahan. Gambar berikut menunjukkan pengaturan Peningkatan versi minor otomatis.



Untuk informasi selengkapnya tentang pengaturan ini, lihat Pengaturan untuk instans DB.

Untuk beberapa RDS untuk versi utama MariaDB di Wilayah AWS beberapa, satu versi minor ditunjuk oleh RDS sebagai versi upgrade otomatis. Setelah versi minor diuji dan disetujui oleh Amazon RDS, tingkatkan versi minor terjadi secara otomatis selama periode pemeliharaan Anda. RDS tidak secara otomatis menetapkan versi minor yang lebih baru sebagai versi peningkatan otomatis. Sebelum RDS menetapkan versi peningkatan otomatis yang lebih baru, beberapa kriteria dipertimbangkan, seperti yang berikut ini:

- Masalah keamanan yang diketahui
- Bug dalam versi komunitas MariaDB
- Stabilitas armada secara keseluruhan sejak versi minor dirilis

> **Note**
>
> Support untuk menggunakan TLS versi 1.0 dan 1.1 telah dihapus dimulai dengan versi minor tertentu dari MariaDB. Untuk informasi tentang versi minor MariaDB yang didukung, lihat. the section called “Dukungan SSL/TLS”

Anda dapat menggunakan AWS CLI perintah berikut untuk menentukan versi target pemutakhiran minor otomatis saat ini untuk versi minor MariaDB tertentu secara spesifik. Wilayah AWS

Untuk Linux, macOS, atau Unix:

```
aws rds describe-db-engine-versions \
--engine mariadb \
--engine-version minor-version \
--region region \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
--output text
```

Untuk Windows:

```
aws rds describe-db-engine-versions ^
--engine mariadb ^
--engine-version minor-version ^
--region region ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output text
```

Misalnya, AWS CLI perintah berikut menentukan target pemutakhiran minor otomatis untuk MariaDB minor versi 10.5.16 di AS Timur (Ohio) (us-timur-2). Wilayah AWS

Untuk Linux, macOS, atau Unix:

```
aws rds describe-db-engine-versions \
--engine mariadb \
--engine-version 10.5.16 \
--region us-east-2 \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
```

```
--output table
```

Untuk Windows:

```
aws rds describe-db-engine-versions ^
--engine mariadb ^
--engine-version 10.5.16 ^
--region us-east-2 ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output table
```

Output Anda akan seperti yang berikut ini.

```
---------------------------------
|    DescribeDBEngineVersions    |
+--------------+-----------------+
|  AutoUpgrade |  EngineVersion  |
+--------------+-----------------+
|  True        |  10.5.17        |
|  False       |  10.5.18        |
|  False       |  10.5.19        |
|  False       |  10.6.5         |
|  False       |  10.6.7         |
|  False       |  10.6.8         |
|  False       |  10.6.10        |
|  False       |  10.6.11        |
|  False       |  10.6.12        |
+--------------+-----------------+
```

Dalam contoh ini, nilai `AutoUpgrade` adalah `True` untuk MariaDB versi 10.5.17. Jadi, target peningkatan minor otomatis adalah MariaDB versi 10.5.17, yang disorot pada output.

Instans DB MariaDB secara otomatis ditingkatkan selama periode pemeliharaan Anda jika kriteria berikut terpenuhi:

- Pengaturan Peningkatan versi minor otomatis diaktifkan.
- Pengaturan Periode retensi cadangan lebih besar dari 0.
- Instans DB menjalankan versi mesin DB minor yang lebih rendah dari versi minor peningkatan otomatis saat ini.

Untuk informasi selengkapnya, lihat Meng-upgrade versi mesin minor secara otomatis.

Menggunakan replika baca untuk mengurangi waktu henti saat meningkatkan basis data MariaDB

Dalam kebanyakan kasus, deployment blue/green adalah opsi terbaik untuk mengurangi waktu henti saat meningkatkan instans DB MariaDB. Untuk informasi selengkapnya, lihat Menggunakan Deployment Blue/Green Amazon RDS untuk pembaruan basis data.

Jika Anda tidak dapat menggunakan deployment blue/green dan instans DB MariaDB Anda saat ini sedang digunakan dengan aplikasi produksi, Anda dapat menggunakan prosedur berikut untuk meningkatkan versi basis data untuk instans DB Anda. Prosedur ini dapat mengurangi jumlah waktu henti untuk aplikasi Anda.

Dengan menggunakan replika baca, Anda dapat melakukan sebagian besar langkah-langkah pemeliharaan terlebih dahulu dan meminimalkan perubahan yang diperlukan selama pemadaman sebenarnya. Dengan teknik ini, Anda dapat menguji dan mempersiapkan instans DB baru tanpa membuat perubahan pada instans DB Anda yang sudah ada.

Prosedur berikut menunjukkan contoh peningkatan dari MariaDB versi 10.5 ke MariaDB versi 10.6. Anda dapat menggunakan langkah umum yang sama untuk peningkatan ke versi mayor lainnya.

Untuk meningkatkan basis data MariaDB saat instans DB sedang digunakan

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Buat sebuah replika baca dari instans DB MariaDB 10.5 Anda. Proses ini membuat salinan yang dapat ditingkatkan dari basis data Anda. Mungkin ada replika baca lainnya dari instans DB tersebut.
    a. Pada konsol, pilih Basis data, lalu pilih instans DB yang ingin Anda tingkatkan.
    b. Untuk Tindakan, pilih Buat replika baca.
    c. Berikan nilai untuk Pengidentifikasi instans DB untuk replika baca Anda dan pastikan bahwa Kelas instans DB dan pengaturan lainnya sudah sesuai dengan instans DB MariaDB 10.5 Anda.
    d. Pilih Buat replika baca.
3. (Opsional) Ketika replika baca telah dibuat dan Status menunjukkan Tersedia, ubah replika baca menjadi deployment Multi-AZ dan aktifkan pencadangan.

Secara default, replika baca dibuat sebagai deployment AZ Tunggal dengan pencadangan dinonaktifkan. Karena replika baca ini akhirnya akan menjadi instans DB produksi, praktik terbaiknya adalah mengonfigurasi deployment Multi-AZ dan mengaktifkan pencadangan sekarang.

a. Pada konsol, pilih Basis data, lalu pilih replika baca yang baru saja Anda buat.
b. Pilih Ubah.
c. Untuk Deployment Multi-AZ, pilih Buat instans siaga.
d. Untuk Periode Retensi Cadangan, pilih nilai selain nol positif, misalnya 3 hari, lalu pilih Lanjutkan.
e. Untuk Penjadwalan modifikasi, pilih Terapkan segera.
f. Pilih Modifikasi instans DB.

4. Saat Status replika baca menunjukkan Tersedia, tingkatkan replika baca ke MariaDB 10.6.

a. Pada konsol, pilih Basis data, lalu pilih replika baca yang baru saja Anda buat.
b. Pilih Ubah.
c. Untuk Versi mesin DB, pilih MariaDB versi 10.6 sebagai target peningkatan, lalu pilih Lanjutkan.
d. Untuk Penjadwalan modifikasi, pilih Terapkan segera.
e. Pilih Modifikasi instans DB untuk memulai peningkatan.

5. Ketika pemutakhiran selesai dan Status menunjukkan Tersedia, verifikasi bahwa replika baca yang ditingkatkan adalah up-to-date dengan sumber MariaDB 10.5 instans DB. Untuk memverifikasi, hubungkan ke replika baca dan jalankan perintah `SHOW REPLICA STATUS`. Jika `Seconds_Behind_Master` bidangnya`0`, maka replikasi adalah up-to-date.

> **Note**
>
> Versi sebelumnya dari MariaDB menggunakan `SHOW SLAVE STATUS`, bukan `SHOW REPLICA STATUS`. Jika Anda menggunakan MariaDB sebelum versi 10.6, gunakan `SHOW SLAVE STATUS`.

6. (Opsional) Buat replika baca dari replika baca Anda.

Jika Anda ingin instans DB memiliki replika baca setelah dipromosikan menjadi instans DB mandiri, Anda dapat membuat replika baca sekarang.

a. Pada konsol, pilih Basis data, lalu pilih replika baca yang baru saja Anda tingkatkan.

b. Untuk Tindakan, pilih Buat replika baca.

c. Berikan nilai untuk Pengidentifikasi instans DB untuk replika baca Anda dan pastikan bahwa Kelas instans DB dan pengaturan lainnya sudah sesuai dengan instans DB MariaDB 10.5 Anda.

d. Pilih Buat replika baca.

7. (Opsional) Konfigurasikan grup parameter kustom DB untuk replika baca.

   Jika Anda ingin instans DB menggunakan grup parameter kustom setelah dipromosikan menjadi instans DB mandiri, Anda dapat membuat grup parameter DB sekarang dan mengaitkannya dengan replika baca.

   a. Buat grup parameter DB kustom untuk MariaDB 10.6. Untuk petunjuk, lihat Membuat grup parameter DB.

   b. Modifikasi parameter yang ingin Anda ubah dalam grup parameter DB yang baru saja Anda buat. Untuk petunjuk, lihat Memodifikasi parameter dalam grup parameter DB.

   c. Pada konsol, pilih Basis data, lalu pilih replika baca.

   d. Pilih Ubah.

   e. Untuk Grup parameter DB, pilih grup parameter MariaDB 10.6 yang baru Anda buat, lalu pilih Lanjutkan.

   f. Untuk Penjadwalan modifikasi, pilih Terapkan segera.

   g. Pilih Modifikasi instans DB untuk memulai peningkatan.

8. Jadikan replika baca MariaDB 10.6 Anda sebagai instans DB mandiri.

   Important

   Saat Anda mempromosikan replika baca MariaDB 10.6 Anda menjadi instans DB mandiri, replika baca ini bukan lagi merupakan replika dari instans DB MariaDB 10.5 Anda. Kami sarankan Anda mempromosikan replika baca MariaDB selama periode pemeliharaan jika instans DB MariaDB 10.5 sumber Anda berada dalam mode hanya baca dan semua operasi tulis ditangguhkan. Saat promosi selesai, Anda dapat mengarahkan operasi tulis Anda ke instans DB MariaDB 10.6 yang telah ditingkatkan untuk memastikan bahwa tidak ada operasi tulis yang hilang.
   Selain itu, sebelum mempromosikan replika baca MariaDB 10.6, kami sarankan Anda menjalankan semua operasi bahasa definisi data (DDL) yang diperlukan di replika

baca MariaDB 10.6 tersebut. Contohnya adalah membuat indeks. Pendekatan ini akan menghindari efek negatif pada performa replika baca MariaDB 10.6 setelah dipromosikan. Untuk mempromosikan replika baca, gunakan prosedur berikut.

a. Pada konsol, pilih Basis data, lalu pilih replika baca yang baru saja Anda tingkatkan.
b. Untuk Tindakan, pilih Promosikan.
c. Pilih Ya untuk mengaktifkan pencadangan otomatis untuk instans replika baca. Untuk informasi selengkapnya, lihat Pengantar cadangan.
d. Pilih Lanjutkan.
e. Pilih Promosikan Replika Baca.

9. Sekarang Anda memiliki versi peningkatan dari basis data MariaDB Anda. Pada tahap ini, Anda dapat mengarahkan aplikasi Anda ke instans DB MariaDB 10.6 yang baru.

Mengimpor data ke instans basis data MariaDB

Anda dapat menggunakan beberapa teknik untuk mengimpor data ke dalam instans basis data RDS for MariaDB. Pendekatan terbaik bergantung pada sumber data, jumlah data, dan apakah impor dilakukan satu kali atau berlanjut. Jika Anda turut memigrasikan aplikasi bersama data, pertimbangkan juga jumlah waktu henti yang bersedia Anda terima.

Temukan teknik-teknik untuk mengimpor data ke instans basis data RDS for MariaDB dalam tabel berikut.

Sumber	Jumlah data	Satu kali atau berkelanjutan	Waktu henti aplikasi	Teknik	Informasi lain
Instans basis data MariaDB yang ada	Setiap	Satu kali atau berkelanjutan	Minimal	Buat replika baca untuk replikasi berkelanjutan. Dorong replika baca untuk pembuatan satu kali instans DB baru.	Menggunakan replika baca instans DB
Basis data MariaDB atau MySQL yang sudah ada	Kecil	Satu kali	Beberapa	Salin data secara langsung ke instans DB MySQL Anda dengan menggunakan utilitas baris perintah.	Mengimpor data dari basis data MariaDB atau MySQL ke instans DB MariaDB atau MySQL

Sumber	Jumlah data	Satu kali atau berkelanjutan	Waktu henti aplikasi	Teknik	Informasi lain
Data tidak disimpan dalam basis data yang sudah ada	Sedang	Satu kali	Beberapa	Buat file datar dan impor mereka menggunakan pernyataan MySQL`LOAD DATA LOCAL INFILE`.	Mengimpor data dari sumber mana pun ke instans DB MySQL atau MariaDB

Sumber	Jumlah data	Satu kali atau berkelanjutan	Waktu henti aplikasi	Teknik	Informasi lain
Basis data MySQL atau MariaDB yang sudah ada di on-premise atau di Amazon EC2	Setiap	Berkelanjutan	Minimal	Konfigurasikan replikasi dengan basis data MariaDB atau MySQL yang ada sebagai sumber replikasi. Anda dapat mengonfigurasikan replikasi ke dalam instans basis data MariaDB dengan menggunakan pengidentifikasi transaksi global (GTID) MariaDB jika instans eksternal adalah MariaDB versi 10.0.24 atau lebih tinggi, atau menggunakan koordinat log biner untuk instans MySQL atau instans MariaDB pada versi yang lebih lama daripada 10.0.24. GTID MariaDB diterapkan secara berbeda dengan GTID MySQL, yang tidak didukung oleh Amazon RDS.	Mengonfigurasi replikasi posisi file log biner dengan instans sumber eksternal Mengimpor data ke instans DB Amazon RDS MariaDB atau MySQL dengan lebih sedikit waktu henti

Sumber	Jumlah data	Satu kali atau berkelanjutan	Waktu henti aplikasi	Teknik	Informasi lain
Sebarang basis data yang ada	Setiap	Satu kali atau berkelanjutan	Minimal	Gunakan AWS Database Migration Service untuk memigrasikan database dengan downtime minimal dan, untuk banyak mesin DB database, lanjutkan replikasi yang sedang berlangsung.	Apakah AWS Database Migration Service dan Menggunakan basis data yang kompatibel dengan MySQL sebagai target untuk AWS DMS dalam Panduan Pengguna AWS Database Migration Service

> **Note**
>
> Basis data sistem MySQL berisi informasi autentikasi dan otorisasi yang dibutuhkan untuk masuk ke instans basis data Anda dan mengakses datanya. Pengedropan, pengubahan, penggantian nama, atau pemenggalan tabel, data, atau konten lain basis data MySQL dalam instans basis data Anda dapat mengakibatkan kesalahan dan mungkin membuat instans basis data Anda dan datanya tidak dapat diakses. Jika ini terjadi, instans DB dapat dipulihkan dari snapshot menggunakan AWS CLI `restore-db-instance-from-db-snapshot`atau dipulihkan menggunakan `restore-db-instance-to-point-in-time`perintah.

Mengimpor data dari basis data MariaDB atau MySQL ke instans DB MariaDB atau MySQL

Anda juga dapat mengimpor data dari basis data MariaDB atau MySQL yang sudah ada ke instans DB MySQL atau MariaDB. Hal ini dilakukan dengan menyalin basis data dengan mysqldump dan memasukkannya langsung ke dalam instans DB MariaDB atau MySQL. Utilitas baris perintah `mysqldump` umumnya digunakan untuk membuat pencadangan dan mentransfer data dari satu server MySQL atau MariaDB ke server lainnya. Utilitas ini disertakan dalam perangkat lunak klien MySQL dan MariaDB.

> **Note**
>
> Jika Anda mengimpor atau mengekspor data dalam jumlah besar dengan instans MySQL DB, lebih andal dan lebih cepat untuk memindahkan data masuk dan keluar dari Amazon RDS dengan menggunakan file cadangan dan Amazon S3. `xtrabackup` Untuk informasi selengkapnya, lihat Memulihkan cadangan ke instans DB MySQL.

Perintah `mysqldump` yang umum digunakan untuk memindahkan data dari basis data eksternal ke instans DB Amazon RDS adalah seperti berikut.

```
mysqldump -u local_user \
    --databases database_name \
    --single-transaction \
    --compress \
    --order-by-primary  \
    -plocal_password | mysql -u RDS_user \
```

```
        --port=port_number \
        --host=host_name \
        -pRDS_password
```

> **Important**
>
> Pastikan tidak ada spasi di antara opsi `-p` dan kata sandi yang dimasukkan. Tentukan kredensial yang berbeda dari perintah yang ditunjukkan di sini sebagai praktik terbaik keamanan.

Perhatikan rekomendasi dan pertimbangan berikut:

- Jangan sertakan skema berikut dalam file dump: `sys`, `performance_schema`, dan `information_schema`. Utilitas `mysqldump` tidak menyertakan skema tersebut secara default.
- Jika Anda perlu memigrasikan pengguna dan hak istimewa, pertimbangkan untuk menggunakan alat yang menghasilkan bahasa kontrol data (DCL) untuk membuatnya kembali, seperti utilitas. pt-show-grants
- Untuk melakukan impor, pastikan pengguna yang melakukannya memiliki akses ke instans DB. Untuk informasi selengkapnya, lihat Mengontrol akses dengan grup keamanan.

Parameternya adalah sebagai berikut:

- `-u` *`local_user`* – Gunakan untuk menentukan nama pengguna. Saat menggunakan parameter ini untuk pertama kalinya, Anda harus menentukan nama akun pengguna pada basis data MariaDB atau MySQL lokal yang diidentifikasi oleh parameter `--databases`.
- `--databases` *`database_name`* – Gunakan untuk menentukan nama basis data pada instans MariaDB atau MySQL lokal yang ingin Anda impor ke Amazon RDS.
- `--single-transaction` – Gunakan untuk memastikan bahwa semua data yang dimuat dari basis data lokal konsisten dengan satu titik waktu. Jika ada proses lain yang mengubah data saat `mysqldump` membacanya, penggunaan parameter ini dapat membantu menjaga integritas data.
- `--compress` – Gunakan untuk mengurangi konsumsi bandwidth jaringan dengan mengompres data dari basis data lokal sebelum mengirimkannya ke Amazon RDS.
- `--order-by-primary` – Gunakan untuk mengurangi waktu pemuatan dengan mengurutkan setiap tabel data berdasarkan kunci primernya.

- `-p`*`local_password`* – Gunakan untuk menentukan kata sandi. Saat menggunakan parameter ini untuk pertama kalinya, Anda harus menentukan kata sandi untuk akun pengguna yang diidentifikasi oleh parameter `-u`.
- `-u` *`RDS_user`* – Gunakan untuk menentukan nama pengguna. Saat menggunakan parameter ini untuk kedua kalinya, Anda harus menentukan nama akun pengguna pada basis data default untuk instans DB MariaDB atau MySQL yang diidentifikasi oleh parameter `--host`.
- `--port` *`port_number`* – Gunakan untuk menentukan port instans DB MariaDB atau MySQL Anda. Secara default, port ini adalah 3306, kecuali jika Anda mengubah nilainya saat membuat instans.
- `--host` *`host_name`* – Gunakan untuk menentukan nama Sistem Nama Domain (DNS) dari titik akhir instans DB Amazon RDS, misalnya, `myinstance.123456789012.us-east-1.rds.amazonaws.com`. Anda dapat menemukan nilai titik akhir dalam detail instans di Konsol Manajemen Amazon RDS.
- `-p`*`RDS_password`* – Gunakan untuk menentukan kata sandi. Saat menggunakan parameter ini untuk kedua kalinya, Anda harus menentukan kata sandi untuk akun pengguna yang diidentifikasi oleh parameter `-u` kedua.

Pastikan Anda membuat prosedur, pemicu, fungsi, atau peristiwa tersimpan apa pun secara manual di dalam basis data Amazon RDS Anda. Jika objek ini berada di basis data yang Anda salin, jangan sertakan saat Anda menjalankan `mysqldump`. Untuk melakukannya, sertakan parameter berikut ke perintah `mysqldump` Anda: `--routines=0 --triggers=0 --events=0`.

Contoh berikut menyalin basis data sampel `world` pada host lokal ke instans DB MySQL.

Untuk Linux, macOS, atau Unix:

```
sudo mysqldump -u localuser \
    --databases world \
    --single-transaction \
    --compress \
    --order-by-primary  \
    --routines=0 \
    --triggers=0 \
    --events=0 \
    -plocalpassword | mysql -u rdsuser \
        --port=3306 \
        --host=myinstance.123456789012.us-east-1.rds.amazonaws.com \
        -prdspassword
```

Untuk Windows, jalankan perintah berikut pada jendela perintah yang telah dibuka dengan mengeklik kanan Jendela Perintah pada menu program Windows dan memilih Jalankan sebagai administrator:

```
mysqldump -u localuser ^
    --databases world ^
    --single-transaction ^
    --compress ^
    --order-by-primary  ^
    --routines=0 ^
    --triggers=0 ^
    --events=0 ^
    -plocalpassword | mysql -u rdsuser ^
        --port=3306 ^
        --host=myinstance.123456789012.us-east-1.rds.amazonaws.com ^
        -prdspassword
```

> **Note**
>
> Tentukan kredensial yang berbeda dari perintah yang ditunjukkan di sini sebagai praktik terbaik keamanan.

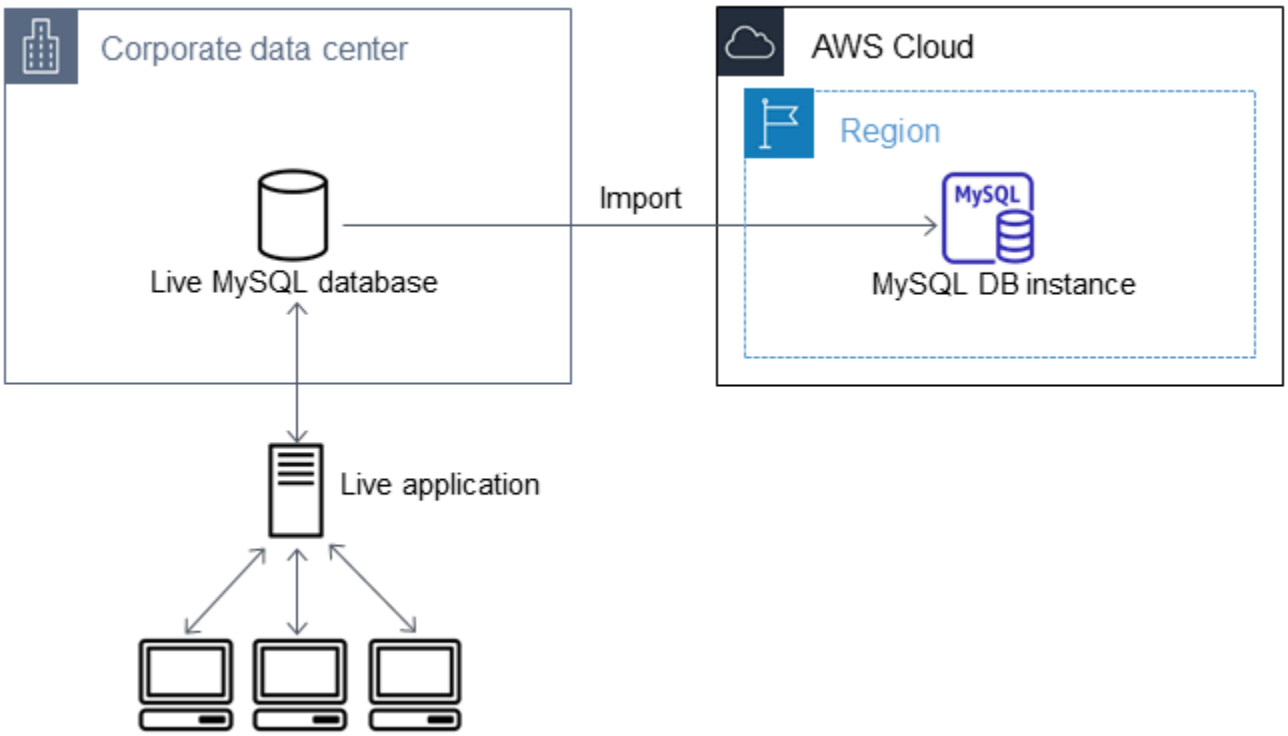
Mengimpor data ke instans DB Amazon RDS MariaDB atau MySQL dengan lebih sedikit waktu henti

Dalam beberapa kasus, mungkin Anda harus mengimpor data dari basis data MariaDB atau MySQL eksternal yang mendukung aplikasi live ke instans DB MariaDB, instans DB MySQL, atau klaster DB Multi-AZ MySQL. Gunakan prosedur berikut untuk meminimalkan dampak terhadap ketersediaan aplikasi. Prosedur ini juga dapat berguna jika Anda menggunakan basis data yang sangat besar. Dengan menggunakan prosedur ini, Anda dapat mengurangi biaya impor dengan mengurangi jumlah data yang dilewatkan di seluruh jaringan AWS.

Dalam prosedur ini, Anda dapat mentransfer salinan data basis data Anda ke instans Amazon EC2 dan mengimpor data ke basis data Amazon RDS baru. Anda kemudian menggunakan replikasi untuk membawa database Amazon RDS up-to-date dengan instans eksternal langsung Anda, sebelum mengarahkan aplikasi Anda ke database Amazon RDS. Lakukan konfigurasi pada replikasi MariaDB berdasarkan pengidentifikasi transaksi global (GTID) jika instans eksternalnya adalah MariaDB 10.0.24 atau yang lebih tinggi dan instans targetnya adalah RDS for MariaDB. Jika tidak, lakukan konfigurasi pada replikasi berdasarkan koordinat log biner. Kami menyarankan replikasi berbasis

GTID jika basis data eksternal Anda mendukungnya karena replikasi berbasis GTID adalah metode yang lebih andal. Untuk informasi selengkapnya, lihat Global transaction ID dalam dokumentasi MariaDB.

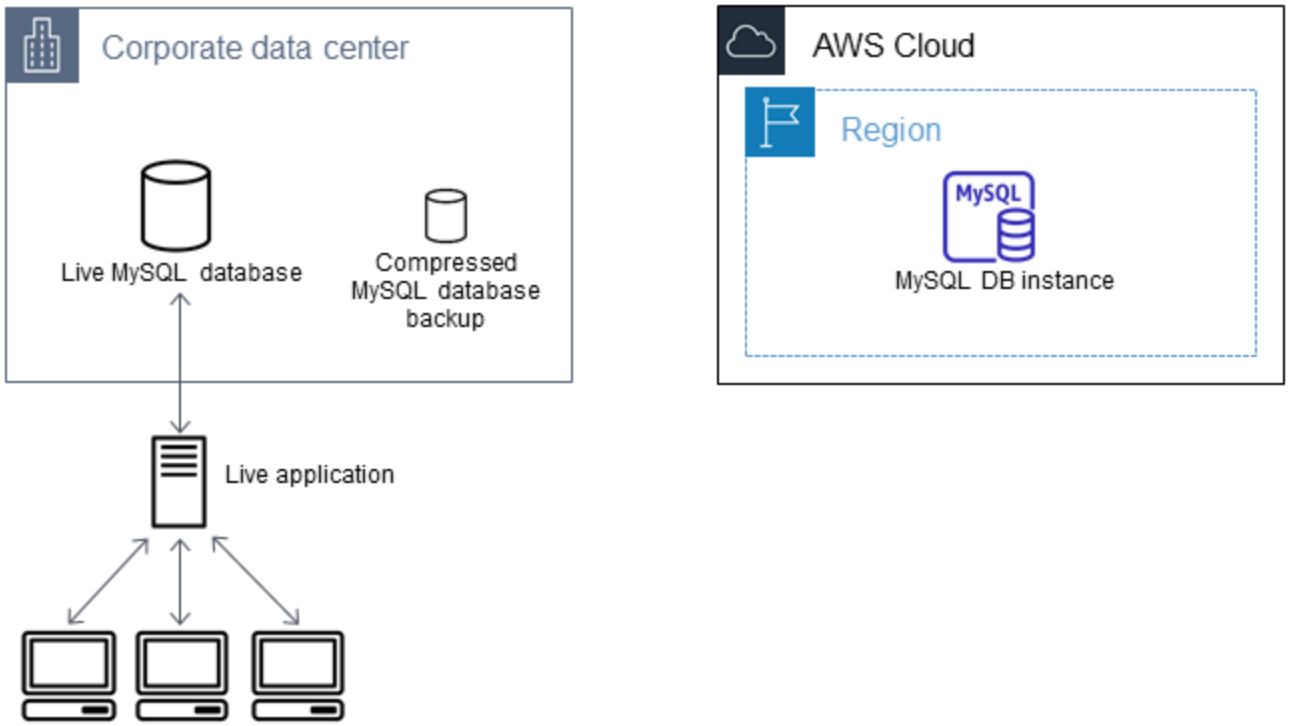
> **Note**
>
> Jika Anda ingin mengimpor data ke instans DB MySQL dan skenario Anda mendukungnya, sebaiknya pindahkan data ke dan dari Amazon RDS dengan menggunakan file cadangan dan Amazon S3. Untuk informasi selengkapnya, lihat Memulihkan cadangan ke instans DB MySQL.



> **Note**
>
> Kami tidak menyarankan Anda untuk menggunakan prosedur ini dengan basis data MySQL sumber dari versi MySQL sebelum versi 5.5 karena ada potensi masalah replikasi. Untuk informasi selengkapnya, lihat Replication compatibility between MySQL versions dalam dokumentasi MySQL.

Membuat salinan basis data yang sudah ada

Langkah pertama dalam proses memigrasikan jumlah data yang besar ke basis data RDS for MariaDB atau RDS for MySQL dengan waktu henti minimal adalah membuat salinan data sumber.



Anda dapat menggunakan utilitas mysqldump untuk membuat cadangan basis data dalam format SQL atau delimited-text. Kami sarankan Anda melakukan uji coba dengan setiap format dalam lingkungan non-produksi untuk melihat metode mana yang dapat meminimalkan jumlah waktu untuk menjalankan mysqldump.

Kami juga sarankan Anda menimbang performa mysqldump dibandingkan dengan manfaat yang ditawarkan dengan menggunakan format delimited-text untuk pemuatan. Pencadangan yang menggunakan format delimited-text akan menciptakan sebuah file teks yang dipisahkan oleh tab untuk setiap tabel yang disalin ke lokasi lain. Untuk mengurangi jumlah waktu yang dibutuhkan untuk mengimpor basis data Anda, Anda dapat memuat file tersebut secara paralel menggunakan perintah `LOAD DATA LOCAL INFILE`. Untuk informasi selengkapnya tentang memilih format mysqldump dan kemudian memuat data, lihat Using mysqldump for backups di dalam dokumentasi MySQL.

Sebelum memulai operasi pencadangan, pastikan Anda mengatur opsi replikasi pada basis data MariaDB atau MySQL yang Anda salin ke Amazon RDS. Opsi replikasi mencakup pengaktifan pencatatan log biner dan pengaturan ID server yang unik. Pengaturan opsi ini menyebabkan server

Anda mulai mencatat log transaksi basis data dan menyiapkannya menjadi sebuah instans replikasi sumber di lain waktu dalam proses ini.

> **Note**
>
> Gunakan opsi `--single-transaction` dengan mysqldump karena opsi ini mencadangkan status konsisten basis data. Untuk memastikan file dump valid, jangan menjalankan pernyataan bahasa definisi data (DDL) saat mysqldump sedang berjalan. Anda dapat menjadwalkan jadwal pemeliharaan untuk operasi ini.
> Jangan sertakan skema berikut dalam file dump: `sys`, `performance_schema`, dan `information_schema`. Utilitas mysqldump mengecualikan skema ini secara default.
> Untuk memigrasikan pengguna dan hak istimewa, pertimbangkan untuk menggunakan alat yang menghasilkan bahasa kontrol data (DCL) untuk membuatnya kembali, seperti utilitas. pt-show-grants

Mengatur opsi replikasi

1. Edit file `my.cnf` (file ini biasanya ada di bawah `/etc`).

   ```
   sudo vi /etc/my.cnf
   ```

   Tambahkan opsi `log_bin` dan `server_id` ke bagian `[mysqld]`. Opsi `log_bin` menyediakan sebuah pengidentifikasi nama file untuk file log biner. Opsi `server_id` menyediakan pengidentifikasi unik untuk server dalam hubungan sumber-replika.

   Contoh berikut menunjukkan bagian `[mysqld]` yang diperbarui dari sebuah file `my.cnf`.

   ```
   [mysqld]
   log-bin=mysql-bin
   server-id=1
   ```

   Untuk informasi selengkapnya, lihat dokumentasi MySQL.

2. Untuk replikasi dengan klaster DB Multi-AZ, atur `ENFORCE_GTID_CONSISTENCY` dan parameter `GTID_MODE` ke `ON`.

   ```
   mysql> SET @@GLOBAL.ENFORCE_GTID_CONSISTENCY = ON;
   ```

```
mysql> SET @@GLOBAL.GTID_MODE = ON;
```

Pengaturan ini tidak diperlukan untuk replikasi dengan instans DB.

3. Mulai ulang layanan `mysql`.

```
sudo service mysqld restart
```

Membuat salinan cadangan basis data yang sudah ada

1. Buat cadangan data Anda menggunakan utilitas mysqldump, dengan menentukan format SQL atau delimited-text.

   Tentukan `--master-data=2` untuk membuat file cadangan yang dapat digunakan untuk memulai replikasi antar server. Untuk informasi selengkapnya, lihat dokumentasi mysqldump.

   Untuk meningkatkan performa dan memastikan integritas data, gunakan opsi `--order-by-primary` dan `--single-transaction` mysqldump.

   Untuk menghindari penyertaan basis data sistem MySQL di dalam cadangan, jangan gunakan opsi `--all-databases` dengan mysqldump. Untuk informasi selengkapnya, lihat Creating a data snapshot using mysqldump dalam dokumentasi MySQL.

   Gunakan chmod sesuai kebutuhan untuk memastikan bahwa direktori tempat file cadangan diciptakan dapat ditulis.

   > **Important**
   >
   > Pada Windows, jalankan jendela perintah sebagai administrator.

   - Untuk membuat output SQL, gunakan perintah berikut.

     Untuk Linux, macOS, atau Unix:

```
sudo mysqldump \
    --databases database_name \
    --master-data=2  \
    --single-transaction \
```

```
    --order-by-primary \
    -r backup.sql \
    -u local_user \
    -p password
```

> **Note**
>
> Tentukan kredensial selain prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

Untuk Windows:

```
mysqldump ^
    --databases database_name ^
    --master-data=2  ^
    --single-transaction ^
    --order-by-primary ^
    -r backup.sql ^
    -u local_user ^
    -p password
```

> **Note**
>
> Tentukan kredensial selain prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

- Untuk membuat output delimited-text, gunakan perintah berikut.

  Untuk Linux, macOS, atau Unix:

```
sudo mysqldump \
    --tab=target_directory \
    --fields-terminated-by ',' \
    --fields-enclosed-by '"' \
    --lines-terminated-by 0x0d0a \
    database_name \
    --master-data=2 \
    --single-transaction \
    --order-by-primary \
```

```
    -p password
```

Untuk Windows:

```
mysqldump ^
    --tab=target_directory ^
    --fields-terminated-by "," ^
    --fields-enclosed-by """ ^
    --lines-terminated-by 0x0d0a ^
    database_name ^
    --master-data=2 ^
    --single-transaction ^
    --order-by-primary ^
    -p password
```

> **Note**
>
> Tentukan kredensial selain prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.
> Pastikan Anda membuat prosedur, pemicu, fungsi, atau peristiwa tersimpan apa pun secara manual di dalam basis data Amazon RDS Anda. Jika Anda memiliki objek-objek tersebut di dalam basis data yang Anda salin, jangan sertakan saat Anda menjalankan mysqldump. Untuk melakukannya, sertakan argumen berikut dengan perintah mysqldump Anda: `--routines=0 --triggers=0 --events=0`.

Saat menggunakan format delimited-text, muncul komentar `CHANGE MASTER TO` saat Anda menjalankan mysqldump. Komentar ini berisi nama dan posisi file log master. Jika instans eksternalnya bukan MariaDB versi 10.0.24 atau yang lebih tinggi, catat nilai untuk `MASTER_LOG_FILE` dan `MASTER_LOG_POS`. Anda memerlukan nilai-nilai ini saat menyiapkan replikasi.

```
-- Position to start replication or point-in-time recovery from
--
-- CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin-changelog.000031',
 MASTER_LOG_POS=107;
```

Jika Anda menggunakan format SQL, Anda dapat memperoleh nama dan posisi file log master pada komentar `CHANGE MASTER TO` tersebut di dalam file cadangan. Jika instans eksternalnya

adalah MariaDB versi 10.0.24 atau yang lebih tinggi, Anda dapat memperoleh GTID pada langkah berikutnya.

2. Jika instans eksternal yang Anda gunakan adalah MariaDB versi 10.0.24 atau yang lebih tinggi, Anda dapat menggunakan replikasi berbasis GTID. Jalankan `SHOW MASTER STATUS` pada instans MariaDB eksternal untuk mendapatkan nama dan posisi file log biner, lalu konversi ke GTID dengan menjalankan `BINLOG_GTID_POS` pada instans MariaDB eksternal.

```
SELECT BINLOG_GTID_POS('binary log file name', binary log file position);
```

Catat GTID yang ditampilkan; Anda membutuhkannya untuk mengonfigurasi replikasi.

3. Kompres data yang disalin untuk mengurangi jumlah sumber daya jaringan yang dibutuhkan untuk menyalin data Anda ke basis data Amazon RDS. Catat ukuran file cadangan. Anda memerlukan informasi ini saat menentukan seberapa besar instans Amazon EC2 yang harus dibuat. Setelah selesai, kompres file cadangan menggunakan GZIP atau utilitas kompresi pilihan Anda.

   - Untuk mengompresi output SQL, gunakan perintah berikut.

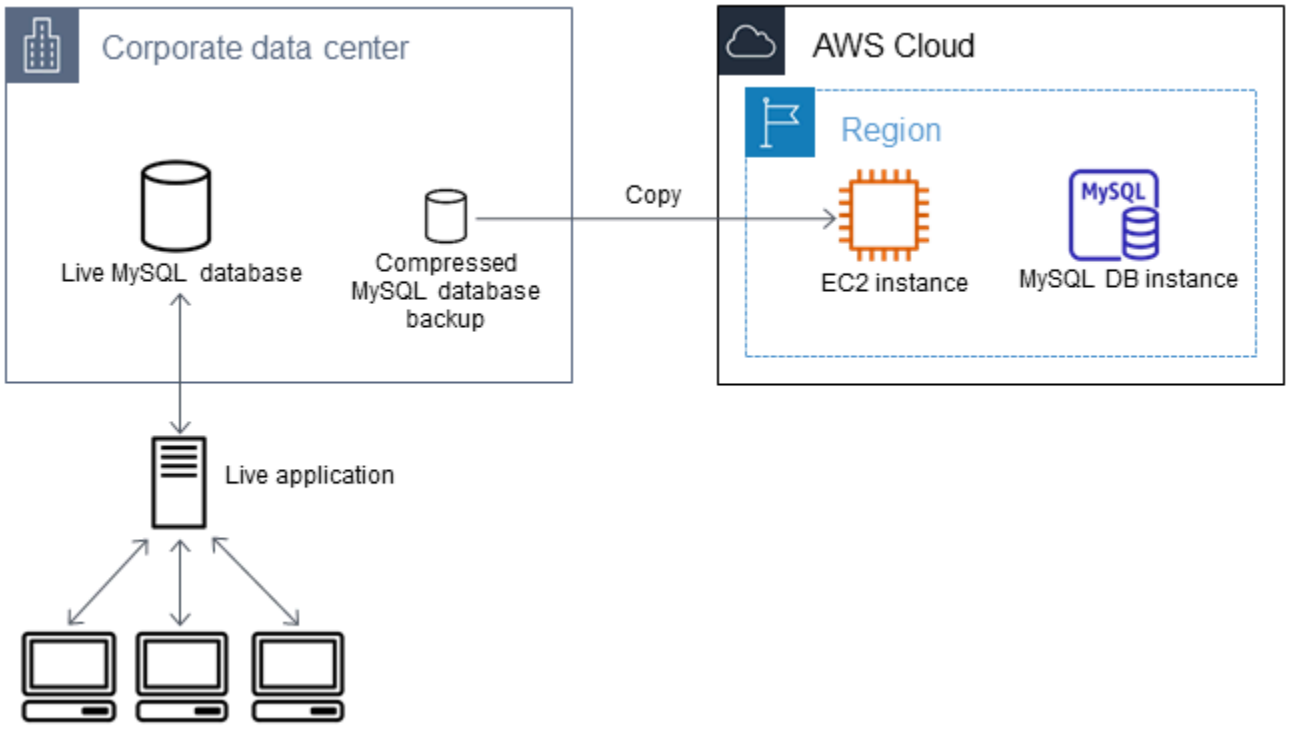
```
gzip backup.sql
```

   - Untuk mengompresi output delimited-text, gunakan perintah berikut.

```
tar -zcvf backup.tar.gz target_directory
```

Buat sebuah instans Amazon EC2 dan salin basis data terkompresi

Penyalinan file cadangan basis data terkompresi ke sebuah instans Amazon EC2 membutuhkan lebih sedikit sumber daya jaringan dibandingkan dengan melakukan penyalinan langsung data tidak terkompresi antar instans basis data. Setelah data Anda berada di Amazon EC2, Anda dapat menyalinnya dari sana langsung ke basis data MariaDB atau MySQL Anda. Agar Anda dapat menghemat biaya sumber daya jaringan, instans Amazon EC2 Anda harus berada di AWS Wilayah yang sama dengan instans Amazon RDS DB Anda. Memiliki instans Amazon EC2 di AWS Wilayah yang sama dengan database Amazon RDS Anda juga mengurangi latensi jaringan selama impor.



Membuat instans Amazon EC2 dan menyalin data Anda

1. Di Wilayah AWS tempat Anda berencana untuk membuat database RDS, buat virtual private cloud (VPC), grup keamanan VPC, dan subnet VPC. Pastikan aturan masuk untuk grup keamanan VPC Anda mengizinkan alamat IP yang dibutuhkan agar aplikasi Anda dapat terhubung ke AWS. Anda dapat menentukan rentang alamat IP (misalnya, `203.0.113.0/24`), atau grup keamanan VPC lainnya. Anda dapat menggunakan Konsol Manajemen Amazon VPC untuk membuat dan mengelola VPC, subnet, dan grup keamanan. Untuk informasi selengkapnya, lihat Mulai menggunakan Amazon VPC dalam Panduan Memulai Amazon Virtual Private Cloud.
2. Buka Konsol Manajemen Amazon EC2 dan pilih AWS Wilayah yang berisi instans Amazon EC2 dan database Amazon RDS Anda. Luncurkan sebuah instans Amazon EC2 menggunakan VPC, subnet, dan grup keamanan yang Anda buat pada Langkah 1. Pastikan Anda memilih tipe instans dengan penyimpanan yang cukup untuk file cadangan basis data Anda saat tidak terkompresi. Untuk detail tentang instans Amazon EC2 lihat Mulai menggunakan instans Linux Amazon EC2 dalam Panduan Pengguna Amazon Elastic Compute Cloud untuk Linux.
3. Untuk terhubung ke basis data Amazon RDS Anda dari instans Amazon EC2 Anda, edit grup keamanan VPC Anda. Tambahkan aturan masuk yang menentukan alamat IP privat instans EC2 Anda. Anda dapat menemukan alamat IP pribadi pada tab Detail dari panel Instans dalam jendela konsol EC2. Untuk mengedit grup keamanan VPC dan menambahkan aturan masuk, pilih Grup Keamanan dalam panel navigasi konsol EC2, pilih grup keamanan Anda, lalu tambahkan aturan

masuk untuk MySQL atau Aurora yang menentukan alamat IP privat instans EC2 Anda. Untuk mempelajari cara menambahkan aturan ke sebuah grup keamanan VPC, lihat Menambahkan dan menghapus aturan di dalam Panduan Pengguna Amazon VPC.

4. Salin file cadangan basis data terkompresi Anda dari sistem lokal ke instans Amazon EC2 Anda. Gunakan `chmod` sesuai kebutuhan untuk memastikan Anda memiliki izin menulis pada direktori target instans Amazon EC2. Anda dapat menggunakan `scp` atau klien Secure Shell (SSH) untuk menyalin file. Berikut adalah contohnya.

```
scp -r -i key pair.pem backup.sql.gz ec2-user@EC2 DNS:/target_directory/backup.sql.gz
```

> **Important**
>
> Pastikan untuk menyalin data sensitif menggunakan protokol transfer jaringan yang aman.

5. Hubungkan ke instans Amazon EC2 Anda dan instal pembaruan terkini dan alat klien MySQL dengan menggunakan perintah berikut.

```
sudo yum update -y
sudo yum install mysql -y
```

Untuk informasi selengkapnya, lihat Membuat koneksi ke instans Anda dalam Panduan Pengguna Amazon Elastic Compute Cloud untuk Linux.

> **Important**
>
> Contoh ini menginstal klien MySQL pada Amazon Machine Image (AMI) untuk distribusi Linux Amazon. Untuk menginstal klien MySQL pada distribusi yang berbeda, seperti Linux Ubuntu atau Red Hat Enterprise, contoh ini tidak berlaku. Untuk informasi tentang penginstalan MySQL, lihat Installing and Upgrading MySQL dalam dokumentasi MySQL.

6. Saat terhubung ke instans Amazon EC2, dekompresi file cadangan basis data Anda. Berikut ini adalah beberapa contohnya.
   - Untuk mendekompresi output SQL, gunakan perintah berikut.

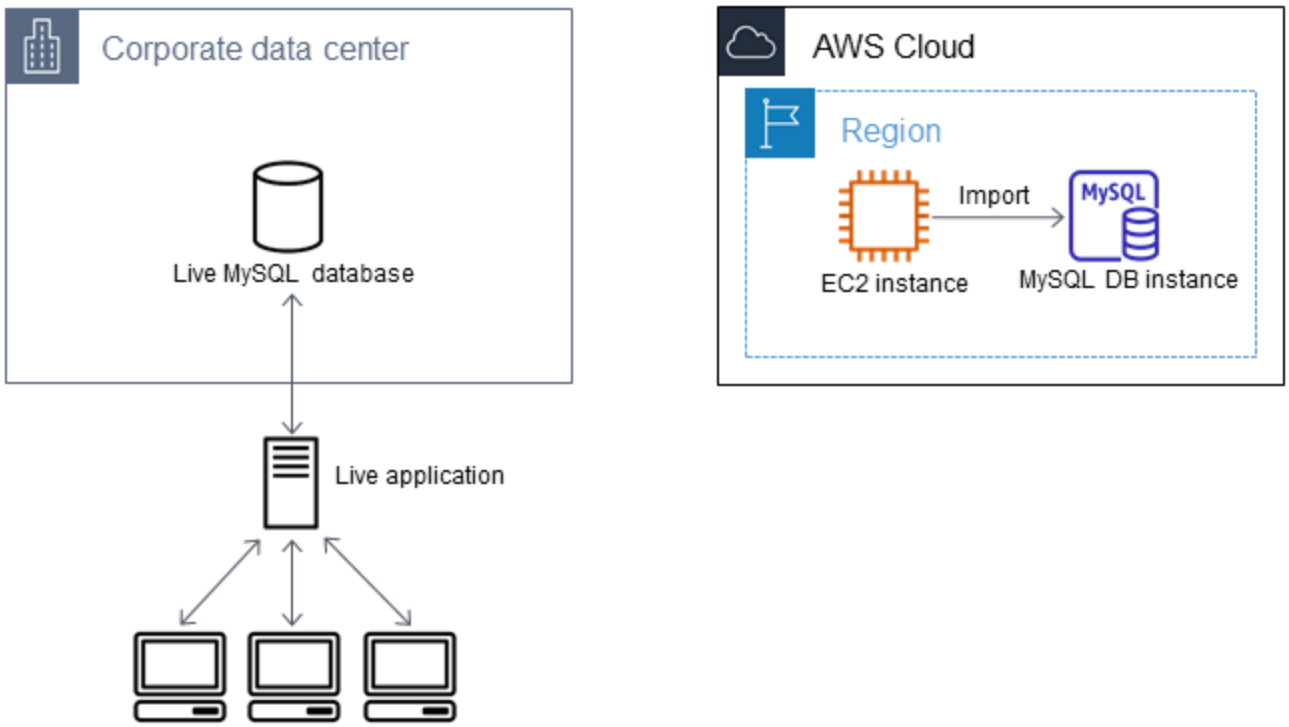
     ```
     gzip backup.sql.gz -d
     ```

   - Untuk mendekompresi output delimited-text, gunakan perintah berikut:

```
tar xzvf backup.tar.gz
```

Buat basis data MySQL atau MariaDB dan impor data dari instans Amazon EC2 Anda

Dengan membuat instance MariaDB, instans MySQL DB, atau kluster DB MySQL Multi-AZ di Wilayah yang AWS sama dengan instans Amazon EC2 Anda, Anda dapat mengimpor file cadangan database dari EC2 lebih cepat daripada melalui internet.



Untuk membuat basis data MariaDB atau MySQL dan mengimpor data Anda

1. Tentukan kelas instans DB dan jumlah ruang penyimpanan yang dibutuhkan untuk mendukung perkiraan beban kerja untuk basis data Amazon RDS ini. Sebagai bagian dari proses ini, putuskan berapa ruang dan kapasitas pemrosesan yang memadai untuk prosedur pemuatan data Anda. Putuskan juga apa yang diperlukan untuk menangani beban kerja produksi. Anda dapat memperkirakan ini berdasarkan ukuran dan sumber daya dari basis data MariaDB atau MySQL sumber. Untuk informasi selengkapnya, lihat Kelas instans DB .
2. Buat instans DB atau cluster DB multi-AZ di AWS Wilayah yang berisi instans Amazon EC2 Anda.

   Untuk membuat klaster DB Multi-AZ MySQL, ikuti petunjuk di Membuat klaster DB Multi-AZ.

Untuk membuat instans DB MariaDB atau MySQL, ikuti petunjuk di Membuat instans DB Amazon RDS dan gunakan pedoman berikut ini:

- Tentukan versi mesin DB yang kompatibel dengan instans DB sumber Anda, seperti berikut:
  - Jika instans sumber Anda adalah MySQL 5.5.x, instans DB Amazon RDS harus MySQL.
  - Jika instans sumber Anda adalah MySQL 5.6.x atau 5.7.x, instans DB Amazon RDS harus MySQL atau MariaDB.
  - Jika instans sumber Anda adalah MySQL 8.0.x, instans DB Amazon RDS harus MySQL 8.0.x.
  - Jika instans sumber Anda adalah MariaDB 5.5 atau yang lebih tinggi, instans DB Amazon RDS harus MariaDB.
- Tentukan cloud privat virtual (VPC) dan grup keamanan VPC yang sama untuk instans Amazon EC2 Anda. Pendekatan ini memastikan bahwa instans Amazon EC2 dan instans Amazon RDS Anda terlihat oleh satu sama lain pada jaringan. Pastikan instans DB Anda dapat diakses publik. Untuk mengatur replikasi dengan basis data sumber Anda yang akan dijelaskan nanti, instans DB Anda harus dapat diakses publik.
- Jangan mengonfigurasikan lebih dari satu Zona Ketersediaan, retensi cadangan, atau replika baca sebelum Anda selesai mengimpor cadangan basis data. Setelah impor selesai, Anda dapat mengonfigurasi Multi-AZ dan retensi cadangan untuk instans produksi.

3. Tinjau opsi konfigurasi default untuk basis data Amazon RDS. Jika grup parameter default untuk basis data tidak memiliki opsi konfigurasi yang Anda inginkan, temukan grup parameter lain atau buat grup parameter baru. Untuk informasi selengkapnya tentang pembuatan grup parameter, lihat Bekerja dengan grup parameter.
4. Hubungkan ke basis data Amazon RDS baru sebagai pengguna master. Buat pengguna yang diperlukan untuk mendukung administrator, aplikasi, dan layanan yang harus mengakses instans. Nama host untuk basis data Amazon RDS adalah nilai Titik akhir untuk instans ini tanpa menyertakan nomor port. Contohnya adalah `mysampledb.123456789012.us-west-2.rds.amazonaws.com`. Anda dapat menemukan nilai titik akhir dalam detail basis data di Konsol Manajemen Amazon RDS.
5. Hubungkan ke instans Amazon EC2 Anda. Untuk informasi selengkapnya, lihat Membuat koneksi ke instans Anda dalam Panduan Pengguna Amazon Elastic Compute Cloud untuk Linux.
6. Hubungkan ke basis data Amazon RDS Anda sebagai sebuah host jarak jauh dari instans Amazon EC2 Anda menggunakan perintah `mysql`. Berikut adalah contohnya.

```
mysql -h host_name -P 3306 -u db_master_user -p
```

Nama host adalah titik akhir basis data Amazon RDS.

7. Pada prompt `mysql`, jalankan perintah `source` dan berikan nama file dump basis data Anda untuk memuat data ke dalam instans DB Amazon RDS:

   - Untuk format SQL, gunakan perintah berikut.

     ```
     mysql> source backup.sql;
     ```

   - Untuk format delimited-text, pertama-tama buat basis data, jika ini bukan basis data default yang Anda buat saat mengatur basis data Amazon RDS.

     ```
     mysql> create database database_name;
     mysql> use database_name;
     ```

     Lalu buat tabel.

     ```
     mysql> source table1.sql
     mysql> source table2.sql
     etc...
     ```

     Lalu impor data.

     ```
     mysql> LOAD DATA LOCAL INFILE 'table1.txt' INTO TABLE table1 FIELDS TERMINATED BY
      ',' ENCLOSED BY '"' LINES TERMINATED BY '0x0d0a';
     mysql> LOAD DATA LOCAL INFILE 'table2.txt' INTO TABLE table2 FIELDS TERMINATED BY
      ',' ENCLOSED BY '"' LINES TERMINATED BY '0x0d0a';
     etc...
     ```

     Untuk meningkatkan performa, Anda dapat melakukan operasi ini secara paralel dari beberapa koneksi sehingga semua tabel Anda akan diciptakan dan kemudian dimuat secara bersamaan.

     > **Note**
     >
     > Jika Anda menggunakan opsi pemformatan data apa pun dengan mysqldump saat Anda pertama kali membuang tabel, pastikan untuk menggunakan opsi yang sama

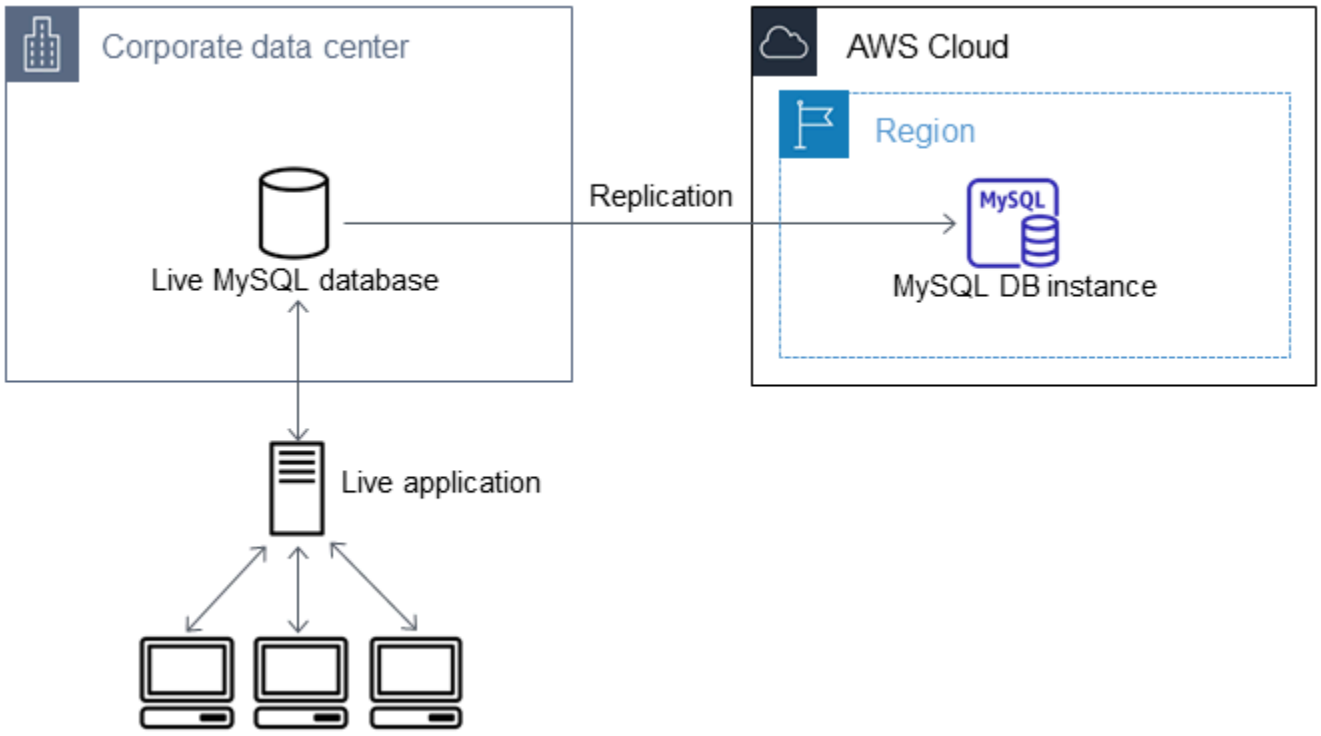
untuk memastikan interpretasi yang tepat dari konten file data. `LOAD DATA LOCAL INFILE`

8. Jalankan `SELECT` kueri sederhana terhadap satu atau dua tabel dalam database yang diimpor untuk memverifikasi bahwa impor berhasil.

Jika Anda tidak lagi memerlukan instans Amazon EC2 yang digunakan dalam prosedur ini, hentikan instans EC2 untuk mengurangi penggunaan sumber daya Anda. AWS Untuk mengakhiri sebuah instans EC2, lihat Mengakhiri instans di Panduan Pengguna Amazon EC2.

Replikasi antara basis data eksternal Anda dan basis data Amazon RDS baru

Basis data sumber Anda kemungkinan diperbarui pada saat menyalin dan mentransfer data ke basis data MariaDB atau MySQL. Dengan demikian, Anda dapat menggunakan replikasi untuk membawa database yang disalin up-to-date dengan database sumber.



Izin yang dibutuhkan untuk memulai replikasi pada basis data Amazon RDS dibatasi dan tidak tersedia untuk pengguna master Amazon RDS Anda. Karena itu, pastikan gunakan perintah Amazon RDS atau perintah mysql.rds_set_external_master_gtid untuk mengonfigurasi replikasi, dan perintah mysql.rds_start_replication untuk memulai replikasi antara basis data live Anda dan basis data Amazon RDS Anda.

Memulai replikasi

Sebelumnya, Anda sudah mengaktifkan pencatatan log biner dan mengatur ID server unik untuk basis data sumber Anda. Sekarang Anda dapat mengatur basis data Amazon RDS Anda sebagai replika dengan basis data live Anda sebagai instans replikasi sumber.

1. Di Konsol Manajemen Amazon RDS, tambahkan alamat IP server yang meng-host basis data sumber ke grup keamanan VPC untuk basis data Amazon RDS. Untuk informasi selengkapnya tentang cara memodifikasi grup keamanan VPC, lihat Grup keamanan untuk VPC Anda dalam Panduan Pengguna Amazon Virtual Private Cloud.

   Anda juga mungkin harus mengonfigurasi jaringan lokal Anda untuk mengizinkan koneksi dari alamat IP basis data Amazon RDS Anda agar klaster DB ini dapat berkomunikasi dengan instans sumber Anda. Untuk menemukan alamat IP basis data Amazon RDS, gunakan perintah `host`.

   ```
   host rds_db_endpoint
   ```

   Nama host adalah nama DNS dari titik akhir basis data Amazon RDS, misalnya, `myinstance.123456789012.us-east-1.rds.amazonaws.com`. Anda dapat menemukan nilai titik akhir dalam detail instans di Konsol Manajemen Amazon RDS.

2. Menggunakan klien pilihan Anda, hubungkan ke instans sumber dan buat pengguna untuk digunakan untuk replikasi. Akun ini digunakan hanya untuk replikasi dan harus dibatasi pada domain Anda untuk meningkatkan keamanan. Berikut adalah contohnya.

   MySQL 5.5, 5.6, dan 5.7

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
   ```

   MySQL 8.0

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED WITH mysql_native_password BY
    'password';
   ```

   > **Note**
   >
   > Tentukan kredensial selain prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

3. Untuk instans sumber, berikan hak istimewa `REPLICATION CLIENT` dan `REPLICATION SLAVE` kepada pengguna replikasi Anda. Misalnya, untuk memberikan hak akses `REPLICATION CLIENT` dan `REPLICATION SLAVE` pada semua basis data untuk pengguna '`repl_user`' bagi domain Anda, jalankan perintah berikut.

   MySQL 5.5, 5.6, dan 5.7

   ```
   GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com'
    IDENTIFIED BY 'password';
   ```

   MySQL 8.0

   ```
   GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
   ```

   > **Note**
   >
   > Tentukan kredensial selain prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

4. Jika Anda menggunakan format SQL untuk menciptakan file cadangan dan instans eksternalnya bukan MariaDB 10.0.24 atau yang lebih tinggi, lihat konten dari file tersebut.

   ```
   cat backup.sql
   ```

   File tersebut menyertakan sebuah komentar `CHANGE MASTER TO` yang berisi nama dan posisi file log master. Komentar ini disertakan dalam file cadangan saat Anda menggunakan opsi `--master-data` dengan mysqldump. Catat nilai untuk `MASTER_LOG_FILE` dan `MASTER_LOG_POS`.

   ```
   --
   -- Position to start replication or point-in-time recovery from
   --

   -- CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin-changelog.000031', MASTER_LOG_POS=107;
   ```

   Jika Anda menggunakan format delimited-text untuk membuat file cadangan dan instans eksternalnya bukan MariaDB 10.0.24 atau yang lebih tinggi, Anda seharusnya sudah memiliki koordinat log biner dari langkah 1 pada prosedur “Membuat salinan cadangan basis data yang sudah ada” dalam topik ini.

Jika instans eksternalnya adalah MariaDB 10.0.24 atau yang lebih tinggi, Anda seharusnya sudah memiliki GTID untuk memulai replikasi dari langkah 2 pada prosedur “Membuat salinan cadangan basis data yang sudah ada” dalam topik ini.

5. Jadikan basis data Amazon RDS sebagai replika. Jika instans eksternalnya bukan MariaDB 10.0.24 atau yang lebih tinggi, hubungkan basis data Amazon RDS sebagai pengguna master dan identifikasi basis data sumber sebagai instans replikasi sumber dengan menggunakan perintah . Gunakan nama file log master dan posisi log master yang Anda tentukan dalam langkah sebelumnya jika Anda memiliki sebuah file cadangan format SQL. Atau gunakan nama dan posisi yang Anda tentukan saat membuat file cadangan jika Anda menggunakan format delimited-text. Berikut adalah contohnya.

```
CALL mysql.rds_set_external_master ('myserver.mydomain.com', 3306,
    'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 0);
```

> **Note**
>
> Tentukan kredensial selain prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

Jika instans eksternalnya adalah MariaDB 10.0.24 atau yang lebih tinggi, hubungkan basis data Amazon RDS sebagai pengguna master dan identifikasi basis data sumber sebagai instans replikasi sumber dengan menggunakan perintah mysql.rds_set_external_master_gtid. Gunakan GTID yang Anda tentukan pada langkah 2 dalam prosedur “Membuat salinan cadangan basis data yang sudah ada” dalam topik ini. Berikut adalah contohnya.

```
CALL mysql.rds_set_external_master_gtid ('source_server_ip_address', 3306,
 'ReplicationUser', 'password', 'GTID', 0);
```

`source_server_ip_address` adalah alamat IP instans replikasi sumber. Alamat DNS privat EC2 saat ini tidak didukung.

> **Note**
>
> Tentukan kredensial selain prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

6. Pada basis data Amazon RDS, terbitkan perintah mysql.rds_start_replication untuk memulai replikasi.

```
CALL mysql.rds_start_replication;
```

7. Pada database Amazon RDS, jalankan perintah SHOW REPLICA STATUS untuk menentukan kapan replika up-to-date dengan instance replikasi sumber. Hasil perintah `SHOW REPLICA STATUS` mencakup bidang `Seconds_Behind_Master`. Ketika `Seconds_Behind_Master` bidang mengembalikan 0, maka replika adalah up-to-date dengan contoh replikasi sumber.

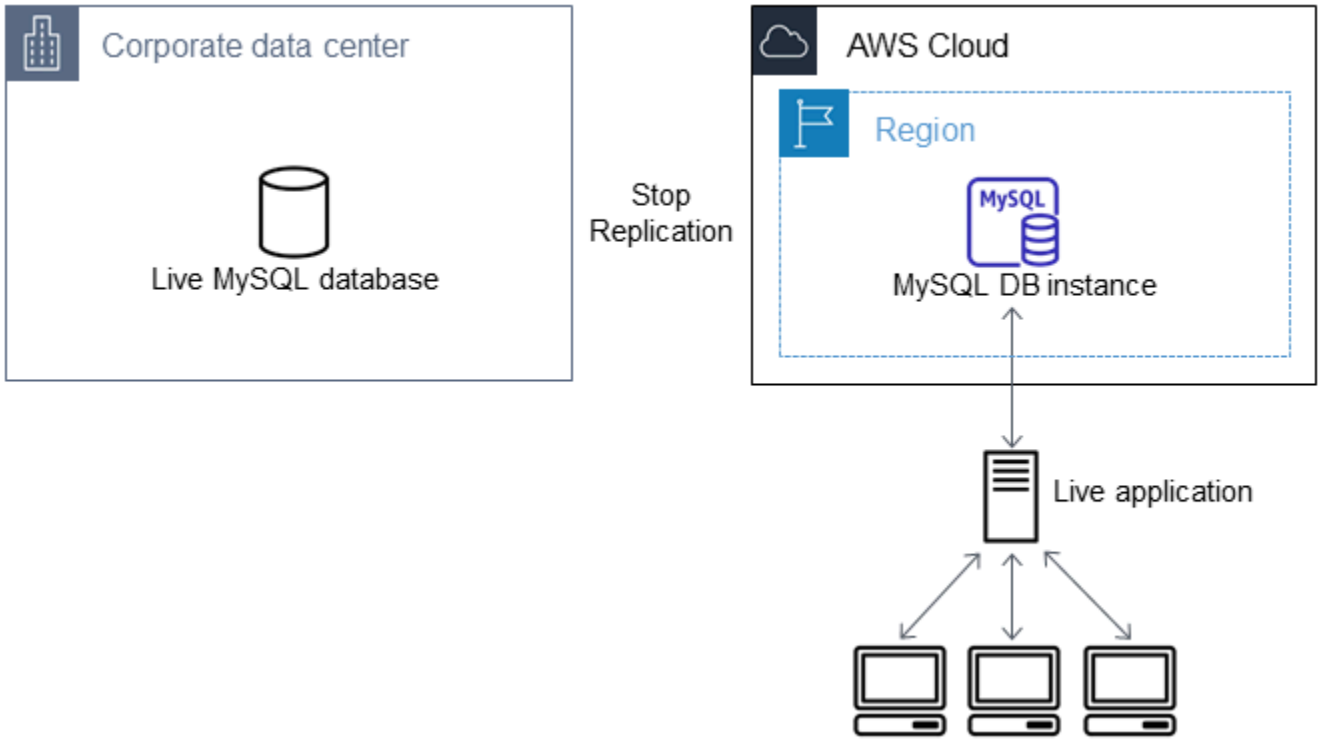
   > **Note**
   >
   > Versi MySQL sebelumnya menggunakan `SHOW SLAVE STATUS`, bukan `SHOW REPLICA STATUS`. Jika Anda menggunakan versi MySQL sebelum 8.0.23, gunakan `SHOW SLAVE STATUS`.

   Untuk instans DB MariaDB 10.5, 10.6, atau 10.11, jalankan prosedur mysql.rds_replica_status, bukan perintah MySQL.

8. Setelah database Amazon RDS up-to-date, aktifkan backup otomatis sehingga Anda dapat memulihkan database tersebut jika diperlukan. Anda dapat mengaktifkan atau memodifikasi pencadangan otomatis untuk basis data Amazon RDS Anda menggunakan Konsol Manajemen Amazon RDS. Untuk informasi selengkapnya, lihat Pengantar cadangan.

Mengalihkan aplikasi live Anda ke instans Amazon RDS Anda

Setelah up-to-date database MariaDB atau MySQL dengan instance replikasi sumber, Anda sekarang dapat memperbarui aplikasi langsung Anda untuk menggunakan instans Amazon RDS.



Mengalihkan aplikasi live Anda ke basis data MariaDB atau MySQL Anda dan menghentikan replikasi

1. Untuk menambahkan grup keamanan VPC untuk basis data Amazon RDS, tambahkan alamat IP server yang meng-host aplikasi. Untuk informasi selengkapnya tentang cara memodifikasi grup keamanan VPC, lihat Grup keamanan untuk VPC Anda dalam Panduan Pengguna Amazon Virtual Private Cloud.
2. Verifikasi bahwa `Seconds_Behind_Master` bidang dalam hasil perintah SHOW REPLICA STATUS adalah 0, yang menunjukkan bahwa replika up-to-date dengan contoh replikasi sumber.

```
SHOW REPLICA STATUS;
```

> **Note**
>
> Versi MySQL sebelumnya menggunakan `SHOW SLAVE STATUS`, bukan `SHOW REPLICA STATUS`. Jika Anda menggunakan versi MySQL sebelum 8.0.23, gunakan `SHOW SLAVE STATUS`.

Untuk instans DB MariaDB 10.5, 10.6, atau 10.11, jalankan prosedur mysql.rds_replica_status, bukan perintah MySQL.

3. Tutup semua koneksi ke sumber setelah transaksi selesai.
4. Perbarui aplikasi Anda untuk menggunakan basis data Amazon RDS. Pembaruan ini biasanya melibatkan perubahan pengaturan koneksi untuk mengidentifikasi nama host dan port basis data Amazon RDS, akun pengguna dan kata sandi untuk terhubung, dan basis data yang digunakan.
5. Hubungkan ke instans DB.

   Untuk klaster DB Multi-AZ, hubungkan ke instans DB penulis.
6. Hentikan replikasi untuk instans Amazon RDS menggunakan perintah mysql.rds_stop_replication.

```
CALL mysql.rds_stop_replication;
```

7. Jalankan perintah mysql.rds_reset_external_master pada basis data Amazon RDS Anda untuk mereset konfigurasi replikasi sehingga instans ini tidak lagi diidentifikasi sebagai replika.

```
CALL mysql.rds_reset_external_master;
```

8. Aktifkan fitur Amazon RDS tambahan seperti dukungan Multi-AZ dan replika baca. Lihat informasi yang lebih lengkap di Mengonfigurasi dan mengelola deployment Multi-AZ dan Menggunakan replika baca instans DB.

Mengimpor data dari sumber mana pun ke instans DB MySQL atau MariaDB

Sebaiknya buat snapshot DB dari instans Amazon RDS DB target sebelum dan sesudah pemuatan data. Snapshot DB Amazon RDS adalah cadangan lengkap instans DB Anda yang dapat digunakan untuk memulihkan instans DB Anda ke status yang diketahui. Saat Anda memulai sebuah snapshot DB, operasi I/O ke instans DB Anda untuk sementara ditangguhkan selama basis data Anda dicadangkan.

Jika perlu, segera buat sebuah snapshot DB sebelum pemuatan agar Anda dapat memulihkan basis data ke statusnya sebelum pemuatan. Dengan snapshot DB yang diambil segera setelah pemuatan, Anda tidak perlu memuat data lagi jika terjadi hal-hal yang tidak diinginkan dan snapshot ini juga dapat digunakan untuk melakukan seeding instans basis data baru.

Daftar berikut menunjukkan langkah-langkah yang harus diambil. Setiap langkah dibahas secara lebih mendetail di bawah.

1. Buat file datar yang berisi data yang akan dimuat.

2. Hentikan aplikasi apa pun yang mengakses instans DB target.
3. Buat sebuah snapshot DB.
4. Pertimbangkan menonaktifkan pencadangan otomatis Amazon RDS.
5. Muat data.
6. Aktifkan pencadangan otomatis lagi.

Langkah 1: Buat file datar yang berisi data yang akan dimuat

Gunakan format umum, seperti CSV (Comma-Separated Values), untuk menyimpan data yang akan dimuat. Setiap tabel harus memiliki file sendiri; Anda tidak dapat menggabungkan data untuk beberapa tabel dalam file yang sama. Beri setiap file nama yang sama dengan tabelnya. Ekstensi file dapat berupa apa pun yang Anda inginkan. Misalnya, jika nama tabelnya `sales`, nama file-nya mungkin `sales.csv` atau `sales.txt`, tetapi bukan `sales_01.csv`.

Jika memungkinkan, urutkan data berdasarkan kunci primer tabel yang dimuat. Tindakan ini secara drastis meningkatkan waktu pemuatan dan meminimalkan kebutuhan penyimpanan disk.

Kecepatan dan efisiensi prosedur ini ditentukan oleh kemampuan untuk mempertahankan ukuran file tetap kecil. Jika ukuran file individual yang tidak terkompresi lebih besar dari 1 GiB, bagilah ke dalam beberapa file dan muat setiap file secara terpisah.

Pada sistem seperti Unix (termasuk Linux), gunakan perintah `split`. Misalnya, perintah berikut membagi file `sales.csv` menjadi beberapa file berukuran kurang dari 1 GiB, yang hanya membagi pada jeda baris (-C 1024m). File-file baru tersebut diberi nama `sales.part_00`, `sales.part_01`, dan seterusnya.

```
split -C 1024m -d sales.csv sales.part_
```

Utilitas yang serupa juga tersedia untuk sistem operasi lain.

Langkah 2: Hentikan aplikasi apa pun yang mengakses instans DB target

Sebelum memulai pemuatan besar, hentikan semua aktivitas aplikasi yang mengakses instans DB target yang akan Anda muati. Kami menyarankan hal ini terutama jika sesi lain akan memodifikasi tabel yang sedang dimuat atau tabel yang menjadi rujukan. Tindakan ini dapat mengurangi risiko pelanggaran pembatasan yang terjadi selama pemuatan dan meningkatkan performa pemuatan.

Tindakan ini juga memungkinkan untuk memulihkan instans DB ke titik tepat sebelum pemuatan tanpa kehilangan perubahan yang dibuat oleh proses yang tidak terlibat dalam pemuatan.

Tentu saja, terkadang ini tidak memungkinkan atau tidak praktis. Jika Anda tidak dapat menghentikan aplikasi dari mengakses instans DB sebelum pemuatan, lakukan langkah-langkah untuk memastikan ketersediaan dan integritas data Anda. Langkah-langkah tertentu yang dibutuhkan dapat bervariasi tergantung kasus penggunaan dan persyaratan situs tertentu.

Langkah 3: Buat snapshot DB

Jika Anda berencana memuat data ke dalam instans DB baru yang tidak berisi data, Anda dapat melompati langkah ini. Sebaliknya, dengan membuat snapshot DB dari instans DB Anda, Anda dapat memulihkan instans DB ke titik tepat sebelum pemuatan, jika memang diperlukan. Seperti yang disebutkan sebelumnya, saat Anda memulai snapshot DB, operasi I/O ke instans DB Anda ditangguhkan selama beberapa menit selama basis data dicadangkan.

Contoh berikut menggunakan AWS CLI `create-db-snapshot` perintah untuk membuat snapshot DB dari `AcmeRDS` instance dan memberikan snapshot DB pengenal. `"preload"`

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-snapshot \
    --db-instance-identifier AcmeRDS \
    --db-snapshot-identifier preload
```

Untuk Windows:

```
aws rds create-db-snapshot ^
    --db-instance-identifier AcmeRDS ^
    --db-snapshot-identifier preload
```

Anda juga dapat menggunakan pemulihan dari fungsionalitas snapshot DB untuk membuat instans DB pengujian untuk melakukan dry run atau untuk membatalkan perubahan yang dibuat selama pemuatan.

Harap diingat bahwa pemulihan basis data dari sebuah snapshot DB akan menciptakan sebuah instans DB baru yang, seperti semua instans DB, memiliki pengidentifikasi yang unik dan titik akhir. Untuk memulihkan instans DB tanpa mengubah titik akhir, pertama-tama hapus instans DB sehingga Anda dapat menggunakan ulang titik akhir.

Misalnya, untuk membuat instans DB untuk dry run atau pengujian lainnya, beri instans DB tersebut pengidentifikasinya sendiri. Dalam contoh ini, AcmeRDS-2" adalah pengidentifikasinya. Contoh ini terhubung ke instans DB menggunakan titik akhir yang terkait dengan AcmeRDS-2.

Untuk Linux, macOS, atau Unix:

```
aws rds restore-db-instance-from-db-snapshot \
    --db-instance-identifier AcmeRDS-2 \
    --db-snapshot-identifier preload
```

Untuk Windows:

```
aws rds restore-db-instance-from-db-snapshot ^
    --db-instance-identifier AcmeRDS-2 ^
    --db-snapshot-identifier preload
```

Untuk menggunakan ulang titik akhir yang sudah ada, pertama-tama hapus instans DB kemudian berikan pengidentifikasi yang sama kepada basis data yang dipulihkan.

Untuk Linux, macOS, atau Unix:

```
aws rds delete-db-instance \
    --db-instance-identifier AcmeRDS \
    --final-db-snapshot-identifier AcmeRDS-Final

aws rds restore-db-instance-from-db-snapshot \
    --db-instance-identifier AcmeRDS \
    --db-snapshot-identifier preload
```

Untuk Windows:

```
aws rds delete-db-instance ^
    --db-instance-identifier AcmeRDS ^
    --final-db-snapshot-identifier AcmeRDS-Final

aws rds restore-db-instance-from-db-snapshot ^
    --db-instance-identifier AcmeRDS ^
    --db-snapshot-identifier preload
```

Contoh sebelumnya mengambil snapshot DB akhir sebuah instans DB sebelum menghapusnya. Ini adalah langkah opsional, tetapi direkomendasikan.

Langkah 4: Pertimbangkan untuk menonaktifkan pencadangan otomatis Amazon RDS

> **Warning**
>
> Jangan mematikan cadangan otomatis jika Anda perlu melakukan point-in-time pemulihan.

Mematikan pencadangan otomatis menghapus semua cadangan yang ada, jadi point-in-time pemulihan tidak mungkin dilakukan setelah pencadangan otomatis dimatikan. Penonaktifan pencadangan otomatis adalah sebuah optimisasi performa dan tidak dibutuhkan untuk pemuatan data. Snapshot DB manual tidak terpengaruh dengan menonaktifkan pencadangan otomatis. Semua snapshot DB manual yang sudah ada tetap tersedia untuk pemulihan.

Penonaktifan pencadangan otomatis mengurangi waktu pemuatan sekitar 25 persen dan mengurangi jumlah ruang penyimpanan yang dibutuhkan selama pemuatan. Jika Anda berencana memuat data ke dalam sebuah instans DB baru yang tidak berisi data, penonaktifan pencadangan adalah cara yang mudah untuk mempercepat pemuatan dan menghindari penggunaan penyimpanan tambahan yang diperlukan untuk pencadangan. Namun, dalam beberapa kasus Anda mungkin berencana untuk melakukan pemuatan ke dalam instans DB yang sudah berisi data. Jika demikian, pertimbangkan manfaat mematikan cadangan terhadap dampak kehilangan kemampuan untuk melakukan. point-in-time-recovery

Instans DB memiliki pencadangan otomatis yang diaktifkan secara default (dengan periode retensi satu hari). Untuk menonaktifkan pencadangan otomatis, atur periode retensi pencadangan ke nol. Setelah pemuatan selesai, Anda dapat mengaktifkan kembali pencadangan dengan mengatur periode retensi pencadangan ke nilai selain nol. Untuk mengaktifkan atau menonaktifkan pencadangan, Amazon RDS mematikan instans DB dan memulainya kembali untuk mengaktifkan atau menonaktifkan pencatatan log MariaDB atau MySQL.

Gunakan AWS CLI `modify-db-instance` perintah untuk mengatur retensi cadangan ke nol dan segera terapkan perubahan. Untuk mengatur periode retensi menjadi nol diperlukan mulai ulang instans DB, jadi tunggu hingga mulai ulang selesai sebelum melanjutkan.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier AcmeRDS \
    --apply-immediately \
    --backup-retention-period 0
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier AcmeRDS ^
    --apply-immediately ^
    --backup-retention-period 0
```

Anda dapat memeriksa status instans DB Anda dengan AWS CLI `describe-db-instances` perintah. Contoh berikut ini menampilkan status instans DB dari instans DB `AcmeRDS`.

```
aws rds describe-db-instances --db-instance-identifier AcmeRDS --query "*[].
{DBInstanceStatus:DBInstanceStatus}"
```

Saat status instans DB adalah `available`, Anda dapat melanjutkan.

Langkah 5: Muat data

Gunakan pernyataan `LOAD DATA LOCAL INFILE` MySQL untuk membaca baris dari file datar Anda ke dalam tabel database.

Contoh berikut menunjukkan cara memuat data dari file bernama `sales.txt` ke dalam tabel bernama `Sales` dalam database.

```
mysql> LOAD DATA LOCAL INFILE 'sales.txt' INTO TABLE Sales FIELDS TERMINATED BY ' '
 ENCLOSED BY '' ESCAPED BY '\\';
Query OK, 1 row affected (0.01 sec)
Records: 1  Deleted: 0  Skipped: 0  Warnings: 0
```

Untuk informasi selengkapnya tentang `LOAD DATA` pernyataan tersebut, lihat dokumentasi MySQL.

Langkah 6: Aktifkan pencadangan otomatis Amazon RDS

Setelah pemuatan selesai, aktifkan pencadangan otomatis Amazon RDS dengan mengatur periode retensi pencadangan kembali ke nilai sebelum pemuatan. Sebagaimana dijelaskan sebelumnya, Amazon RDS akan memulai ulang instans DB, jadi bersiaplah untuk pemadaman singkat.

Contoh berikut menggunakan AWS CLI `modify-db-instance` perintah untuk mengaktifkan backup otomatis untuk instans `AcmeRDS` DB dan mengatur periode retensi menjadi satu hari.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier AcmeRDS \
    --backup-retention-period 1 \
    --apply-immediately
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier AcmeRDS ^
    --backup-retention-period 1 ^
    --apply-immediately
```

Menggunakan replikasi MariaDB di Amazon RDS

Anda biasanya menggunakan replika baca untuk mengonfigurasi replikasi antara instans DB Amazon RDS. Untuk informasi umum tentang replika baca, lihat Menggunakan replika baca instans DB. Untuk informasi spesifik tentang menggunakan replika baca di Amazon RDS for MariaDB, lihat Menggunakan replika baca MariaDB.

Anda juga dapat mengonfigurasi replikasi berdasarkan koordinat log biner untuk instans DB MariaDB. Untuk instans MariaDB, Anda juga dapat mengonfigurasi replikasi berdasarkan ID transaksi global (GTID), yang memberikan keselamatan kerusakan yang lebih baik. Untuk informasi selengkapnya, lihat Mengonfigurasikan replikasi berbasis GTID dengan instans sumber eksternal.

Berikut ini adalah opsi replikasi lain yang tersedia RDS for MariaDB:

- Anda dapat menyiapkan replikasi antara instans DB RDS for MariaDB dan instans MySQL atau MariaDB yang berada di luar Amazon RDS. Untuk informasi tentang cara mengonfigurasi replikasi dengan sumber eksternal, lihat Mengonfigurasi replikasi posisi file log biner dengan instans sumber eksternal.
- Anda dapat mengonfigurasi replikasi untuk mengimpor basis data dari instans MySQL atau MariaDB yang berada di luar Amazon RDS, atau untuk mengekspor basis data ke instans tersebut. Untuk informasi selengkapnya, lihat Mengimpor data ke instans DB Amazon RDS MariaDB atau MySQL dengan lebih sedikit waktu henti dan Mengekspor data dari instans DB MySQL dengan menggunakan replikasi.

Untuk opsi replikasi ini, Anda dapat menggunakan replikasi berbasis baris, replikasi berbasis pernyataan, atau replikasi campuran. Replikasi berbasis baris hanya mereplikasi baris yang diubah yang dihasilkan dari laporan SQL. Replikasi berbasis pernyataan mereplikasi seluruh pernyataan SQL. Replikasi campuran menggunakan replikasi berbasis pernyataan jika memungkinkan, tetapi ke replikasi berbasis baris ketika pernyataan SQL yang tidak aman untuk replikasi berbasis pernyataan dijalankan. Dalam sebagian besar kasus, replikasi campuran direkomendasikan. Format biner log dari DB instans menentukan apakah replikasi berbasis baris, berbasis pernyataan, atau campuran. Untuk informasi mengenai pengaturan format log biner, lihat Format pengelogan biner.

Topik

- Menggunakan replika baca MariaDB
- Mengonfigurasikan replikasi berbasis GTID dengan instans sumber eksternal
- Mengonfigurasi replikasi posisi file log biner dengan instans sumber eksternal

Menggunakan replika baca MariaDB

Setelah itu, Anda bisa menemukan informasi spesifik tentang menggunakan replika baca di Amazon RDS for MariaDB. Untuk mengetahui informasi umum tentang replika baca dan petunjuk penggunaannya, lihat Menggunakan replika baca instans DB.

Topik

Mengonfigurasi replika baca dengan MariaDB

Sebelum instans DB MariaDB dapat berfungsi sebagai sumber replikasi, pastikan untuk mengaktifkan pencadangan otomatis pada instans DB sumber dengan mengatur periode retensi cadangan ke nilai selain 0. Persyaratan ini juga berlaku untuk replika baca yang merupakan instans DB sumber untuk replika baca lain.

Anda dapat membuat hingga 15 replika baca dari satu instans DB dalam Wilayah yang sama. Agar replikasi beroperasi secara efektif, setiap replika baca harus memiliki jumlah sumber daya komputasi dan penyimpanan yang sama seperti instans DB sumber. Jika Anda menskalakan instans DB sumber, replika baca juga perlu diskalakan.

RDS for MariaDB mendukung replika baca berjenjang. Untuk mempelajari cara mengonfigurasi replika baca berjenjang, lihat Menggunakan replika baca berjenjang dengan RDS for MariaDB.

Anda dapat menjalankan beberapa replika baca, membuat dan menghapus tindakan pada saat yang sama yang mereferensikan instans DB sumber yang sama. Saat Anda melakukan tindakan ini, tidak boleh ada lebih dari 15 replika baca untuk setiap instans sumber.

Mengonfigurasi filter replikasi dengan MariaDB

Anda dapat menggunakan filter replikasi untuk mengetahui basis data dan tabel mana yang direplikasi dengan replika baca. Filter replikasi dapat menyertakan basis data dan tabel ke dalam replikasi atau mengecualikan mereka dari replikasi.

Berikut ini adalah beberapa kasus penggunaan untuk replikasi filter:

- Untuk mengurangi ukuran replika baca. Dengan filter replikasi, Anda dapat mengecualikan basis data dan tabel yang tidak diperlukan pada replika baca.
- Untuk mengecualikan basis data dan tabel dari replika baca untuk alasan keamanan.
- Untuk mereplikasi basis data yang berbeda dan tabel untuk kasus penggunaan tertentu di replika baca yang berbeda. Misalnya, Anda mungkin menggunakan replika baca khusus untuk analitik atau penyerpihan.
- Untuk instans DB yang memiliki replika baca berbeda Wilayah AWS, untuk mereplikasi basis data atau tabel yang berbeda secara berbeda Wilayah AWS.

> **Note**
>
> Anda juga dapat menggunakan filter replikasi untuk menentukan basis data dan tabel apa yang direplikasi dengan instans DB MariaDB primer yang dikonfigurasi sebagai replika dalam topologi replikasi masuk. Untuk mengetahui informasi selengkapnya tentang konfigurasi ini, lihat Mengonfigurasi replikasi posisi file log biner dengan instans sumber eksternal.

Topik

- Mengatur parameter filter replikasi untuk RDS for MariaDB
- Keterbatasan filter replikasi untuk RDS for MariaDB
- Contoh filter replikasi untuk RDS for MariaDB
- Melihat filter replikasi untuk replika baca

Mengatur parameter filter replikasi untuk RDS for MariaDB

Untuk mengonfigurasi filter replikasi, atur parameter filter replikasi berikut pada replika baca:

- `replicate-do-db` — Mereplikasi perubahan ke basis data yang ditentukan. Ketika Anda menetapkan parameter ini untuk replika baca, hanya basis data yang ditentukan dalam parameter yang direplikasi.
- `replicate-ignore-db` — Jangan mereplikasi perubahan ke basis data yang ditentukan. Ketika parameter `replicate-do-db` diatur untuk replika baca, parameter ini tidak dievaluasi.
- `replicate-do-table` — Mereplikasi perubahan ke tabel yang ditentukan. Ketika Anda menetapkan parameter ini untuk replika baca, hanya tabel yang ditentukan dalam parameter yang direplikasi. Juga, ketika parameter `replicate-do-db` atau `replicate-ignore-db` diatur, basis data yang mencakup tabel tertentu harus disertakan dalam replikasi dengan replika baca.
- `replicate-ignore-table` — Jangan mereplikasi perubahan ke tabel yang ditentukan. Ketika parameter `replicate-do-table` diatur untuk replika baca, parameter ini tidak dievaluasi.
- `replicate-wild-do-table` — Mereplikasi tabel berdasarkan basis data dan pola nama tabel yang ditentukan. Karakter wildcard `%` dan `_` didukung. Ketika parameter `replicate-do-db` atau `replicate-ignore-db` diatur, pastikan untuk menyertakan basis data yang mencakup tabel tertentu dalam replikasi dengan replika baca.
- `replicate-wild-ignore-table` — Jangan mereplikasi tabel berdasarkan basis data dan pola nama tabel yang ditentukan. Karakter wildcard `%` dan `_` didukung. Ketika parameter `replicate-do-table` atau `replicate-wild-do-table` diatur untuk replika baca, parameter ini tidak dievaluasi.

Parameter dievaluasi sesuai dengan urutannya dalam daftar. Untuk mengetahui informasi selengkapnya tentang cara kerja parameter ini, lihat Dokumentasi MariaDB.

Secara default, masing-masing parameter ini memiliki nilai kosong. Pada setiap replika baca, Anda dapat menggunakan parameter ini untuk mengatur, mengubah, dan menghapus filter replikasi. Ketika Anda menetapkan salah satu parameter ini, pisahkan masing-masing filter dari yang lain dengan koma.

Anda dapat menggunakan karakter wildcard `%` dan `_` dalam parameter `replicate-wild-do-table` dan `replicate-wild-ignore-table`. Parameter wildcard `%` mencocokkan dengan sejumlah karakter, dan wildcard `_` hanya mencocokkan satu karakter.

Format logging biner dari instans DB sumber penting untuk replikasi karena menentukan catatan perubahan data. Pengaturan parameter `binlog_format` menentukan apakah replikasi berbasis baris atau berbasis pernyataan. Untuk informasi selengkapnya, lihat Format pengelogan biner.

> **Note**
>
> Semua pernyataan bahasa definisi data (DDDL) direplikasi sebagai pernyataan, terlepas dari pengaturan `binlog_format` pada instans DB sumber.

Keterbatasan filter replikasi untuk RDS for MariaDB

Keterbatasan tersebut berlaku kepada filter replikasi untuk RDS for MariaDB:

- Setiap parameter filter replikasi memiliki batas 2.000 karakter.
- Koma tidak didukung dalam filter replikasi.
- Opsi MariaDB `binlog_do_db` dan `binlog_ignore_db` untuk filter log biner tidak didukung.
- Filter replikasi tidak mendukung transaksi XA.

  Untuk mengetahui informasi selengkapnya, lihat Pembatasan pada Transaksi XA dalam dokumentasi MySQL.
- Filter replikasi tidak didukung RDS for MariaDB versi 10.2.

Contoh filter replikasi untuk RDS for MariaDB

Untuk mengonfigurasi filter replikasi untuk replika baca, modifikasi parameter filter replikasi dalam grup parameter yang terkait dengan replika baca tersebut.

> **Note**
>
> Anda tidak dapat mengubah grup parameter default. Jika replika baca menggunakan grup parameter default, buat grup parameter baru dan kaitkan dengan replika baca tersebut. Untuk informasi selengkapnya tentang grup parameter DB, lihat Bekerja dengan grup parameter.

Anda dapat mengatur parameter dalam grup parameter menggunakan AWS Management Console, AWS CLI, atau API RDS. Untuk informasi tentang mengatur parameter, lihat Memodifikasi parameter dalam grup parameter DB. Ketika Anda mengatur parameter dalam grup parameter, semua instans DB yang terkait dengan grup parameter tersebut menggunakan pengaturan parameter. Jika Anda mengatur parameter filter replikasi dalam grup parameter, pastikan bahwa grup parameter dikaitkan hanya dengan replika baca. Biarkan parameter filter replikasi kosong untuk instans DB sumber.

Contoh berikut mengatur parameter menggunakan AWS CLI. Contoh ini menetapkan `ApplyMethod` ke `immediate` sehingga perubahan parameter terjadi segera setelah perintah CLI selesai. Jika Anda ingin menerapkan perubahan tertunda setelah replika baca di-reboot, atur `ApplyMethod` ke `pending-reboot`.

Contoh berikut mengatur filter replikasi:

- Including databases in replication
- Including tables in replication
- Including tables in replication with wildcard characters
- Escaping wildcard characters in names
- Excluding databases from replication
- Excluding tables from replication
- Excluding tables from replication using wildcard characters

Example Termasuk basis data dalam replikasi

Contoh berikut menyertakan basis data `mydb1` dan `mydb2` dalam replikasi. Ketika Anda mengatur `replicate-do-db` untuk replika baca, hanya basis data yang ditentukan dalam parameter yang direplikasi.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-do-db", "ParameterValue": "mydb1,mydb2",
 "ApplyMethod":"immediate"}]"
```

Untuk Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-do-db", "ParameterValue": "mydb1,mydb2",
 "ApplyMethod":"immediate"}]"
```

Example Termasuk tabel dalam replikasi

Contoh berikut menyertakan tabel `table1` dan `table2` dalam basis data `mydb1` dalam replikasi.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-do-table", "ParameterValue":
 "mydb1.table1,mydb1.table2", "ApplyMethod":"immediate"}]"
```

Untuk Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-do-table", "ParameterValue":
 "mydb1.table1,mydb1.table2", "ApplyMethod":"immediate"}]"
```

Example Menyertakan tabel dalam replikasi menggunakan karakter wildcard

Contoh berikut menyertakan tabel dengan nama berawalan `orders` dan `returns` dalam basis data `mydb` dalam replikasi.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-wild-do-table", "ParameterValue":
 "mydb.orders%,mydb.returns%", "ApplyMethod":"immediate"}]"
```

Untuk Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-wild-do-table", "ParameterValue":
 "mydb.orders%,mydb.returns%", "ApplyMethod":"immediate"}]"
```

Example Mengeluarkan karakter wildcard dalam nama

Contoh berikut menunjukkan kepada Anda cara menggunakan karakter escape \ untuk mengeluarkan karakter wildcard yang merupakan bagian dari nama.

Asumsikan bahwa Anda memiliki beberapa nama tabel dalam basis data `mydb1` yang dimulai dengan `my_table`, dan Anda ingin menyertakan tabel ini dalam replikasi. Nama tabel meliputi garis bawah,

yang juga merupakan karakter wildcard, sehingga contoh ini melepaskan garis bawah dalam nama tabel.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-wild-do-table", "ParameterValue": "my
\_table%", "ApplyMethod":"immediate"}]"
```

Untuk Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-wild-do-table", "ParameterValue": "my
\_table%", "ApplyMethod":"immediate"}]"
```

Example Mengecualikan basis data dari replikasi

Contoh berikut mengecualikan basis data `mydb1` dan `mydb2` dari replikasi.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-ignore-db", "ParameterValue":
 "mydb1,mydb2", "ApplyMethod":"immediate"}]"
```

Untuk Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-ignore-db", "ParameterValue":
 "mydb1,mydb2", "ApplyMethod":"immediate"}]"
```

Example Mengecualikan tabel dari replikasi

Contoh berikut mengecualikan tabel `table1` dan `table2` dalam basis data `mydb1` dari replikasi.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-ignore-table", "ParameterValue":
 "mydb1.table1,mydb1.table2", "ApplyMethod":"immediate"}]"
```

Untuk Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-ignore-table", "ParameterValue":
 "mydb1.table1,mydb1.table2", "ApplyMethod":"immediate"}]"
```

Example Mengecualikan tabel dari replikasi menggunakan karakter wildcard

Contoh berikut mengecualikan tabel dengan nama berawalan `orders` dan `returns` dalam basis data `mydb` dari replikasi.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-wild-ignore-table", "ParameterValue":
 "mydb.orders%,mydb.returns%", "ApplyMethod":"immediate"}]"
```

Untuk Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-wild-ignore-table", "ParameterValue":
 "mydb.orders%,mydb.returns%", "ApplyMethod":"immediate"}]"
```

Melihat filter replikasi untuk replika baca

Anda dapat melihat filter replikasi untuk replika baca dengan cara berikut:

- Memeriksa pengaturan parameter filter replikasi dalam grup parameter yang terkait dengan replika baca.

  Untuk mengetahui petunjuknya, lihat Melihat nilai parameter untuk grup parameter DB.
- Dalam klien MariaDB, hubungkan ke replika baca dan jalankan pernyataan `SHOW REPLICA STATUS`.

Dalam output, bidang berikut menunjukkan filter replikasi untuk replika baca:

- `Replicate_Do_DB`
- `Replicate_Ignore_DB`
- `Replicate_Do_Table`
- `Replicate_Ignore_Table`
- `Replicate_Wild_Do_Table`
- `Replicate_Wild_Ignore_Table`

Untuk mengetahui informasi selengkapnya tentang bidang ini, lihat Memeriksa Status Replikasi dalam dokumentasi MySQL.

> **Note**
>
> Versi sebelumnya dari MariaDB menggunakan `SHOW SLAVE STATUS`, bukan `SHOW REPLICA STATUS`. Jika Anda menggunakan MariaDB sebelum versi 10.5, gunakan `SHOW SLAVE STATUS`.

Mengonfigurasi replikasi tertunda dengan MariaDB

Anda dapat menggunakan replikasi tertunda sebagai strategi pemulihan bencana. Dengan replikasi tertunda, Anda menentukan jumlah waktu minimum, dalam detik, untuk menunda replikasi dari sumber ke replika baca. Jika terjadi bencana, seperti tabel yang terhapus secara tidak sengaja, Anda menyelesaikan langkah-langkah berikut untuk memulihkan dari bencana dengan cepat:

- Hentikan replikasi ke replika baca sebelum perubahan yang menyebabkan bencana dikirim ke replika tersebut.

  Untuk menghentikan replikasi, gunakan prosedur yang tersimpan di mysql.rds_stop_replication.
- Tingkatkan replika baca menjadi instans DB sumber baru dengan menggunakan petunjuk di Mempromosikan replika baca menjadi instans DB mandiri.

> **Note**
>
> - Replikasi tertunda didukung untuk versi MariaDB 10.6 dan yang lebih tinggi.

- Gunakan prosedur yang tersimpan untuk mengonfigurasi replikasi tertunda. Anda tidak dapat mengonfigurasi replikasi tertunda dengan AWS Management Console, AWS CLI, atau Amazon RDS API.
- Anda dapat menggunakan replikasi berdasarkan pengidentifikasi transaksi global (GTID) dalam konfigurasi replikasi tertunda.

Topik

- Mengonfigurasi replikasi tertunda selama pembuatan replika baca
- Mengubah replikasi tertunda untuk replika baca yang sudah ada
- Mempromosikan replika baca

Mengonfigurasi replikasi tertunda selama pembuatan replika baca

Untuk mengonfigurasi replikasi tertunda untuk replika baca di masa mendatang yang dibuat dari instans DB, jalankan prosedur tersimpan mysql.rds_set_configuration dengan parameter `target delay`.

Untuk mengonfigurasi replikasi tertunda selama pembuatan replika baca

1. Dengan menggunakan klien MariaDB, hubungkan ke instans DB MariaDB untuk menjadi sumber replika baca sebagai pengguna master.
2. Jalankan prosedur tersimpan mysql.rds_set_configuration dengan parameter `target delay`.

   Misalnya, jalankan prosedur tersimpan berikut untuk menentukan bahwa replikasi ditunda setidaknya satu jam (3.600 detik) untuk replika baca yang dibuat dari instans DB saat ini.

   ```
   call mysql.rds_set_configuration('target delay', 3600);
   ```

   > **Note**
   >
   > Setelah menjalankan prosedur tersimpan ini, replika baca apa pun yang Anda buat menggunakan AWS CLI atau API Amazon RDS dikonfigurasi dengan replikasi tertunda selama jumlah detik yang ditentukan.

Mengubah replikasi tertunda untuk replika baca yang sudah ada

Untuk mengubah replikasi tertunda untuk replika baca yang ada, jalankan prosedur tersimpan mysql.rds_set_source_delay.

Untuk mengubah replikasi tertunda untuk replika baca yang sudah ada

1. Dengan menggunakan klien MariaDB, hubungkan ke replika baca sebagai pengguna master.
2. Gunakan prosedur yang tersimpan di mysql.rds_stop_replication untuk menghentikan replikasi.
3. Jalankan prosedur yang tersimpan di mysql.rds_set_source_delay.

   Misalnya, jalankan prosedur tersimpan berikut untuk menentukan bahwa replikasi ke replika baca ditunda setidaknya satu jam (3600 detik).

   ```
   call mysql.rds_set_source_delay(3600);
   ```
4. Gunakan prosedur yang tersimpan di mysql.rds_start_replication untuk memulai replikasi.

Mempromosikan replika baca

Setelah replikasi dihentikan, dalam skenario pemulihan bencana, Anda dapat mempromosikan replika baca menjadi instans DB sumber baru. Untuk mengetahui informasi tentang cara mempromosikan replika baca, lihat Mempromosikan replika baca menjadi instans DB mandiri.

Memperbarui replika baca dengan MariaDB

Replika baca dirancang untuk mendukung kueri baca, tetapi Anda mungkin memerlukan pembaruan sesekali. Misalnya, Anda mungkin perlu menambahkan indeks untuk mempercepat jenis kueri tertentu yang mengakses replika. Anda dapat mengaktifkan pembaruan dengan mengatur parameter `read_only` untuk 0 dalam grup parameter DB untuk replika baca.

Menggunakan deployment replika baca multi-AZ dengan MariaDB

Anda dapat membuat replika baca dari deployment instans DB Multi-AZ atau tunggal-AZ. Anda menggunakan deployment Multi-AZ untuk meningkatkan dan ketersediaan data kritis, tetapi Anda tidak dapat menggunakan sekunder Multi-AZ untuk melayani kueri baca-saja. Sebagai gantinya, Anda dapat membuat replika baca dari instans DB Multi-AZ multi-lalu lintas tinggi untuk mengeluarkan kueri baca-saja. Jika instans sumber dari deployment Multi-AZ gagal karena replika baca sekunder, setiap replika baca terkait akan otomatis untuk menggunakan sumber sekunder

(sekarang primer) sebagai sumber replikasinya. Untuk informasi selengkapnya, lihat Mengonfigurasi dan mengelola deployment Multi-AZ.

Anda dapat membuat replika baca sebagai instans DB Multi-AZ. Amazon RDS membuat instans siaga replika Anda di Zona Ketersediaan lain untuk dukungan failover untuk replika tersebut. Membuat replika baca Anda sebagai instans DB Multi-AZ tidak tergantung pada apakah basis data sumber adalah instans DB Multi-AZ.

Menggunakan replika baca berjenjang dengan RDS for MariaDB

RDS for MariaDB mendukung replika baca berjenjang. Dengan replika baca berjenjang, Anda dapat menskalakan pembacaan tanpa menambahkan overhead ke instans DB RDS for MariaDB Anda.

Dengan replika baca berjenjang, instans DB RDS for MariaDB Anda mengirimkan data ke replika baca pertama dalam rantai. Replika baca tersebut kemudian mengirimkan data ke replika kedua dalam rantai, dan seterusnya. Hasil akhirnya adalah bahwa semua replika baca dalam rantai memiliki perubahan dari instans DB RDS for MariaDB, tetapi tanpa overhead hanya pada instans DB sumber.

Anda dapat membuat serangkaian hingga tiga replika baca dalam rantai dari instans DB RDS for MariaDB sumber. Misalnya, anggaplah bahwa Anda memiliki instans DB RDS for MariaDB, `mariadb-main`. Anda dapat melakukan hal berikut:

- Dimulai dengan `mariadb-main`, buat replika baca pertama dalam rantai, `read-replica-1`.
- Selanjutnya, dari `read-replica-1`, buat replika baca berikutnya dalam rantai, `read-replica-2`.
- Akhirnya, dari `read-replica-2`, buat replika baca ketiga dalam rantai, `read-replica-3`.

Anda tidak dapat membuat replika baca lain di luar replika baca berjenjang ketiga ini dalam seri untuk. `mariadb-main` Serangkaian instans lengkap dari instans DB RDS for MariaDB hingga akhir serangkaian replika baca berjenjang dapat terdiri dari paling banyak empat instans DB.

Agar replika baca berjenjang berfungsi, setiap sumber instans DB RDS for MariaDB harus mengaktifkan pencadangan otomatis. Untuk mengaktifkan pencadangan otomatis pada replika baca, pertama-tama buat replika baca, lalu ubah replika baca untuk mengaktifkan pencadangan otomatis. Untuk informasi selengkapnya, lihat Membuat replika baca.

Seperti halnya replika baca lainnya, Anda dapat mempromosikan replika baca yang merupakan bagian dari kaskade. Mempromosikan replika baca dari dalam rantai replika baca menghilangkan

replika itu dari rantai. Misalnya, misalkan Anda ingin memindahkan sebagian beban kerja dari instans `mariadb-main` DB Anda ke instans baru untuk digunakan oleh departemen akuntansi saja. Dengan asumsi rantai tiga replika baca dari contoh, Anda memutuskan untuk mempromosikan `read-replica-2`. Rantai terpengaruh sebagai berikut:

- Mempromosikan `read-replica-2` menghapusnya dari rantai replikasi.
  - Sekarang menjadi instans DB baca/tulis penuh.
  - Itu terus bereplikasi `read-replica-3`, seperti yang dilakukan sebelum promosi.
- `mariadb-main` Anda terus mereplikasi ke `read-replica-1`.

Untuk mengetahui informasi selengkapnya tentang cara mempromosikan replika baca, lihat Mempromosikan replika baca menjadi instans DB mandiri.

Memantau replika baca MariaDB

Untuk replika baca MariaDB, Anda dapat memantau kelambatan replikasi di Amazon CloudWatch dengan melihat metrik Amazon RDS. `ReplicaLag` Metrik `ReplicaLag` melaporkan nilai dari kolom `Seconds_Behind_Master` dari perintah `SHOW REPLICA STATUS`.

> **Note**
>
> Versi sebelumnya dari MariaDB menggunakan `SHOW SLAVE STATUS`, bukan `SHOW REPLICA STATUS`. Jika Anda menggunakan MariaDB sebelum versi 10.5, gunakan `SHOW SLAVE STATUS`.

Penyebab umum keterlambatan replikasi untuk MariaDB adalah sebagai berikut:

- Pemadaman jaringan.
- Menulis ke tabel dengan indeks pada replika baca. Jika parameter `read_only` tidak diatur ke 0 pada replika baca, hal ini dapat merusak replikasi.
- Gunakan mesin penyimpanan nontransaksional seperti MyISAM. Replikasi hanya didukung untuk mesin penyimpanan InnoDB pada MariaDB.

Saat metrik `ReplicaLag` mencapai 0, replika telah menyamai instans DB sumber. Jika metrik `ReplicaLag` menampilkan -1, replikasi saat ini tidak aktif. `ReplicaLag` = -1 setara dengan `Seconds_Behind_Master` = `NULL`.

Memulai dan menghentikan replikasi dengan replika baca MariaDB

Anda dapat menghentikan dan memulai ulang proses replikasi di instans Amazon RDS DB dengan memanggil prosedur yang disimpan sistem mysql.rds_stop_replication dan mysql.rds_start_replication. Anda dapat melakukan ini saat mereplikasi antara dua instans Amazon RDS untuk operasi jangka panjang seperti membuat indeks besar. Anda juga perlu menghentikan dan memulai replikasi saat mengimpor atau mengekspor basis data. Untuk informasi selengkapnya, lihat Mengimpor data ke basis data Amazon RDS MariaDB atau MySQL dengan lebih sedikit waktu henti dan Mengekspor data dari instans DB MySQL dengan menggunakan replikasi.

Jika replikasi dihentikan selama lebih dari 30 hari berturut-turut, baik secara manual atau karena kesalahan replikasi, Amazon RDS menghentikan replikasi antara instans DB sumber dan semua replika baca. Hal ini dilakukan untuk mencegah peningkatan persyaratan penyimpanan pada instans DB sumber dan waktu failover yang lama. Instans DB replika baca masih tersedia. Namun, replikasi tidak dapat dilanjutkan karena log biner yang diperlukan oleh replika baca dihapus dari instans DB sumber setelah replikasi dihentikan. Anda dapat membuat replika baca baru untuk instans DB sumber untuk memulihkan replikasi.

Pemecahan Masalah kendala replika baca MariaDB

Teknologi replikasi untuk MariaDB bersifat asinkron. Karena mereka tidak sinkron, sesekali `BinLogDiskUsage` meningkatkan instans DB sumber dan `ReplicaLag` pada replika baca diharapkan. Misalnya, volume operasi tulis tinggi ke instans DB sumber dapat terjadi secara paralel. Sebaliknya, operasi ke replika baca diseret menggunakan utas I/O tunggal, yang dapat menyebabkan jeda antara instans sumber dan replika baca. Untuk mengetahui informasi selengkapnya tentang replika hanya baca di dokumentasi MariaDB, buka Ringkasan replikasi.

Anda dapat melakukan beberapa hal untuk mengurangi keterlambatan antara pembaruan ke instans DB sumber dan pembaruan berikutnya ke replika baca, seperti berikut:

- Mengukur replika baca untuk memiliki ukuran penyimpanan dan kelas instans DB yang sebanding dengan instans DB sumber.
- Memastikan kompatibilitas pengaturan parameter di grup parameter DB yang digunakan oleh instans DB sumber dan replika baca. Untuk informasi selengkapnya dan instans, lihat diskusi tentang `max_allowed_packet` nanti di bagian ini.

Amazon RDS memantau status replikasi replika baca Anda dan memperbarui `Replication State` bidang instans replika baca untuk `Error` jika replikasi berhenti karena alasan apa pun. Contohnya

mungkin jika kueri DML yang dijalankan pada replika baca Anda bertentangan dengan pembaruan yang dilakukan pada instans DB sumber.

Anda dapat meninjau perincian kesalahan terkait yang disebabkan oleh mesin MariaDB dengan melihat kolom `Replication Error`. Peristiwa yang menunjukkan status replika baca juga dihasilkan, termasuk RDS-EVENT-0045, RDS-EVENT-0046, dan RDS-EVENT-0047. Untuk mengetahui informasi selengkapnya tentang acara dan berlangganan acara, lihat Menggunakan pemberitahuan peristiwa Amazon RDS. Jika muncul pesan kesalahan MariaDB, periksa kesalahan di dokumentasi pesan kesalahan MariaDB.

Satu masalah umum yang dapat menyebabkan kesalahan replikasi adalah ketika nilai untuk `max_allowed_packet` parameter untuk replika baca lebih kecil dari `max_allowed_packet` untuk instans DB sumber. Parameter `max_allowed_packet` adalah parameter kustom yang dapat Anda atur dalam grup parameter DB yang digunakan untuk menentukan ukuran maksimum kode DML yang dapat dijalankan pada basis data. Dalam beberapa kasus, nilai parameter `max_allowed_packet` dalam grup parameter DB yang terkait dengan instans DB sumber lebih kecil dari nilai parameter `max_allowed_packet` dalam grup parameter DB yang terkait dengan replika baca sumber. Dalam kasus ini, proses replikasi dapat menyebabkan kesalahan (Paket lebih besar dari byte 'maks_allowed_packet') dan menghentikan replikasi. Anda dapat memperbaiki kesalahan dengan memiliki replika sumber dan baca, gunakan grup parameter DB yang sama `max_allowed_packet` nilai parameter.

Situasi umum lainnya yang dapat menyebabkan kesalahan replikasi mencakup hal-hal berikut:

- Menulis ke tabel di replika baca. Jika Anda membuat indeks pada replika baca, parameter `read_only` harus diatur ke 0 untuk membuat indeks. Jika Anda menulis pada replika baca, tindakan ini dapat merusak replikasi.
- Menggunakan mesin penyimpanan non-transaksional seperti replika baca MyISAM. memerlukan mesin penyimpanan transaksional. Replikasi hanya didukung untuk mesin penyimpanan InnoDB pada MariaDB.
- Gunakan kueri nondeterministik yang tidak aman seperti `SYSDATE()`. Untuk informasi selengkapnya, lihat Penentuan laporan yang aman dan tidak aman dalam pengelogan biner.

Jika Anda memutuskan bahwa Anda dapat melewati kesalahan dengan aman, Anda dapat mengikuti langkah-langkah yang dijelaskan dalam Melewati kesalahan replikasi saat ini. Jika tidak, Anda dapat menghapus replika baca dan membuat instans menggunakan pengidentifikasi instans DB yang sama

sehingga titik akhir tetap sama dengan replika baca lama Anda. Jika kesalahan replikasi diperbaiki, `Replication State` berubah menjadi mereplikasi.

Untuk instans DB MariaDB, dalam beberapa kasus, replika baca tidak dapat dialihkan ke yang sekunder jika beberapa kejadian log biner (binlog) tidak di-flush saat kegagalan. Dalam kasus ini, hapus dan buat ulang replika baca secara manual. Anda dapat mengurangi kemungkinan terjadinya hal ini dengan menetapkan nilai parameter berikut: `sync_binlog=1` dan `innodb_flush_log_at_trx_commit=1`. Pengaturan ini dapat mengurangi kinerja, jadi uji dampaknya sebelum menerapkan perubahan di lingkungan produksi.

Mengonfigurasikan replikasi berbasis GTID dengan instans sumber eksternal

Anda dapat menyiapkan replikasi berdasarkan pengidentifikasi transaksi global (GTID) dari instans eksternal MariaDB versi 10.0.24 atau lebih tinggi ke dalam instans basis data RDS for MariaDB. Ikuti pedoman ini saat Anda menyiapkan instans sumber eksternal dan replika di Amazon RDS:

- Pantau peristiwa pindah saat gagal/failover untuk instans basis data RDS for MariaDB yang menjadi replika Anda. Jika terjadi pindah saat gagal/failover, maka instans basis data yang merupakan replika Anda dapat dibuat ulang pada sebuah host baru dengan alamat jaringan yang berbeda. Lihat informasi tentang cara memantau peristiwa pindah saat gagal/failover di Menggunakan pemberitahuan peristiwa Amazon RDS.
- Pertahankan log biner (binlog) pada instans sumber Anda sampai Anda memastikan bahwa log itu telah diterapkan pada replika. Upaya mempertahankan ini memastikan bahwa Anda dapat memulihkan instans sumber jika terjadi kegagalan.
- Aktifkan pencadangan otomatis pada instans basis data MariaDB Anda di Amazon RDS. Mengaktifkan pencadangan otomatis memastikan bahwa Anda dapat memulihkan replika ke titik waktu tertentu jika Anda perlu menyinkronkan ulang instans sumber dan replika Anda. Lihat informasi tentang pencadangan dan pemulihan titik waktu di Mencadangkan, memulihkan, dan mengekspor data.

> **Note**
>
> Izin-izin yang dibutuhkan untuk memulai replikasi pada instans basis data MariaDB bersifat terbatas dan tidak tersedia bagi pengguna master Amazon RDS Anda. Karena itu, Anda harus menggunakan perintah-perintah Amazon RDS mysql.rds_set_external_master_gtid dan

mysql.rds_start_replication untuk menyiapkan replikasi antara basis data aktif dan basis data RDS for MariaDB Anda.

Untuk memulai replikasi antara instans sumber eksternal dan instans basis data MariaDB pada Amazon RDS, gunakan prosedur berikut.

Memulai replikasi

1. Jadikan instans basis data MariaDB sumber bersifat hanya baca:

```
mysql> FLUSH TABLES WITH READ LOCK;
mysql> SET GLOBAL read_only = ON;
```

2. Dapatkan GTID saat ini dari instans MariaDB eksternal. Ini dapat Anda lakukan dengan menggunakan `mysql` atau editor kueri pilihan Anda untuk menjalankan `SELECT @@gtid_current_pos;`.

   GTID berformat `<domain-id>-<server-id>-<sequence-id>`. Sebuah GTID yang lazim terlihat seperti **`0-1234510749-1728`**. Lihat informasi yang lebih lengkap tentang GTID dan komponen-komponennya di ID transaksi global dalam dokumentasi MariaDB.

3. Salin basis data dari instans MariaDB eksternal ke instans basis data MariaDB dengan menggunakan `mysqldump`. Untuk basis data yang sangat besar, Anda mungkin perlu menggunakan prosedur di Mengimpor data ke basis data Amazon RDS MariaDB atau MySQL dengan lebih sedikit waktu henti.

   Untuk Linux, macOS, atau Unix:

```
mysqldump \
    --databases database_name \
    --single-transaction \
    --compress \
    --order-by-primary \
    -u local_user \
    -plocal_password | mysql \
        --host=hostname \
        --port=3306 \
        -u RDS_user_name \
        -pRDS_password
```

Untuk Windows:

```
mysqldump ^
    --databases database_name ^
    --single-transaction ^
    --compress ^
    --order-by-primary \
    -u local_user \
    -plocal_password | mysql ^
        --host=hostname ^
        --port=3306 ^
        -u RDS_user_name ^
        -pRDS_password
```

> **Note**
>
> Pastikan bahwa tidak ada spasi antara opsi `-p` dan kata sandi yang dimasukkan. Tetapkan kata sandi selain penggugah/prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

Gunakan opsi-opsi `--host`, `--user (-u)`, `--port` dan `-p` dalam perintah `mysql` untuk menentukan nama host, nama pengguna, porta, dan kata sandi untuk menghubungi instans basis data MariaDB Anda. Nama host adalah nama DNS dari titik akhir instans basis data MariaDB, misalnya `myinstance.123456789012.us-east-1.rds.amazonaws.com`. Anda dapat menemukan nilai titik akhir dalam detail instans di Konsol Manajemen Amazon RDS.

4. Jadikan lagi instans MariaDB sumber bersifat dapat ditulis.

```
mysql> SET GLOBAL read_only = OFF;
mysql> UNLOCK TABLES;
```

5. Di Konsol Manajemen Amazon RDS, tambahkan alamat IP server yang menjadi host data basis data MariaDB eksternal ke grup keamanan VPC untuk instans basis data MariaDB. Lihat informasi yang lebih lengkap tentang cara mengubah grup keamanan VPC di Grup keamanan untuk VPC dalam Panduan Pengguna Amazon Virtual Private Cloud.

   Alamat IP dapat berubah jika kondisi-kondisi berikut terpenuhi:

- Anda menggunakan alamat IP publik untuk komunikasi antara instans sumber eksternal dan instans basis data.
- Instans sumber eksternal dihentikan dan dimulai ulang.

Jika kedua kondisi ini terpenuhi, periksa alamat IP sebelum menambahkannya.

Mungkin Anda juga harus mengonfigurasikan jaringan lokal Anda untuk mengizinkan koneksi dari alamat IP instans basis data MariaDB agar dapat berkomunikasi dengan instans MariaDB eksternal. Untuk menemukan alamat IP instans basis data MariaDB, gunakan perintah `host`.

```
host db_instance_endpoint
```

Nama host adalah nama DNS titik akhir instans basis data MariaDB.

6. Dengan menggunakan klien pilihan Anda, hubungi instans MariaDB eksternal dan buat akun pengguna MariaDB yang akan digunakan untuk replikasi. Akun ini digunakan hanya untuk replikasi dan harus dibatasi pada domain Anda untuk meningkatkan keamanan. Berikut sebuah contoh.

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
```

> **Note**
>
> Tetapkan kata sandi selain penggugah/prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

7. Untuk instans MariaDB eksternal, berikan privilese-privilese `REPLICATION CLIENT` dan `REPLICATION SLAVE` kepada pengguna replikasi Anda. Misalnya, untuk memberikan privilese-privilese `REPLICATION CLIENT` dan `REPLICATION SLAVE` pada semua basis data kepada pengguna '`repl_user`' bagi domain Anda, keluarkan perintah berikut.

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
```

8. Jadikan instans basis data MariaDB sebagai replika. Hubungi instans basis data MariaDB sebagai pengguna master dan tandai basis data MariaDB eksternal sebagai instans sumber replikasi dengan menggunakan perintah mysql.rds_set_external_master_gtid. Gunakan GTID yang Anda peroleh dalam Langkah 2. Berikut sebuah contoh.

```
CALL mysql.rds_set_external_master_gtid ('mymasterserver.mydomain.com', 3306,
 'repl_user', 'password', 'GTID', 0);
```

> **Note**
>
> Tetapkan kata sandi selain penggugah/prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

9. Pada instans basis data MariaDB, terbitkan perintah mysql.rds_start_replication untuk memulai replikasi.

```
CALL mysql.rds_start_replication;
```

Mengonfigurasi replikasi posisi file log biner dengan instans sumber eksternal

Anda dapat menyiapkan replikasi antara instans DB RDS for MySQL atau MariaDB dan instans MySQL atau MariaDB yang berada di luar Amazon RDS menggunakan replikasi file log biner.

Topik

Sebelum Anda mulai

Anda dapat mengonfigurasi replikasi menggunakan posisi file log biner transaksi yang direplikasi.

Izin yang diperlukan untuk memulai replikasi pada instans DB Amazon RDS dibatasi dan tidak tersedia untuk pengguna master Amazon RDS Anda. Karena itu, pastikan Anda menggunakan perintah dan mysql.rds_start_replication Amazon RDS untuk mengatur replikasi antara basis data live dan basis data Amazon RDS Anda.

Untuk mengatur format pencatatan log biner untuk basis data MySQL atau MariaDB, perbarui parameter `binlog_format`. Jika instans DB Anda menggunakan grup parameter instans DB default, buat grup parameter DB baru untuk mengubah pengaturan `binlog_format`. Kami sarankan Anda menggunakan pengaturan default untuk `binlog_format`, yaitu `MIXED`. Namun, Anda juga

dapat mengatur `binlog_format` ke `ROW` atau `STATEMENT` jika Anda memerlukan format log biner (binlog) tertentu. Boot ulang instans DB Anda agar perubahan diterapkan.

Untuk informasi tentang mengatur parameter `binlog_format`, lihat Mengkonfigurasi pengelogan biner MySQL. Untuk informasi tentang implikasi tipe replikasi MySQL yang berbeda-beda, lihat Keuntungan dan kerugian replikasi berbasis pernyataan dan berbasis baris dalam dokumentasi MySQL.

Mengonfigurasi replikasi posisi file log biner dengan instans sumber eksternal

Ikuti pedoman ini saat Anda menyiapkan instans sumber eksternal dan replika di Amazon RDS:

- Pantau peristiwa failover untuk instans DB Amazon RDS yang merupakan replika Anda. Jika terjadi failover, maka instans DB yang merupakan replika Anda dapat dibuat ulang pada host baru dengan alamat jaringan yang berbeda. Untuk informasi tentang cara pemonitoran peristiwa failover, lihat Menggunakan pemberitahuan peristiwa Amazon RDS.
- Pertahankan binlog pada instans sumber Anda hingga Anda memverifikasi bahwa binlog tersebut telah diterapkan ke replika. Dengan mempertahankannya, Anda dapat memulihkan instans sumber Anda jika terjadi kegagalan.
- Aktifkan pencadangan otomatis pada instans DB Amazon RDS Anda. Dengan mengaktifkan pencadangan otomatis, Anda dapat memulihkan replika ke titik waktu tertentu jika Anda perlu menyinkronkan ulang instans sumber dan replika Anda. Untuk informasi tentang pencadangan dan point-in-time pemulihan, lihat. Mencadangkan, memulihkan, dan mengekspor data

Mengonfigurasi replikasi file log biner dengan Instans sumber eksternal

1. Jadikan instans MySQL atau MariaDB sumber sebagai hanya-baca.

   ```
   mysql> FLUSH TABLES WITH READ LOCK;
   mysql> SET GLOBAL read_only = ON;
   ```

2. Jalankan perintah `SHOW MASTER STATUS` pada instans MySQL atau MariaDB sumber untuk menentukan lokasi binlog.

   Anda menerima output yang mirip dengan contoh berikut.

   ```
   File                      Position
   ----------------------------------
    mysql-bin-changelog.000031      107
   ```

```
-----------------------------------
```

3. Salin basis data dari instans eksternal ke instans DB Amazon RDS menggunakan `mysqldump`. Untuk basis data yang sangat besar, Anda mungkin ingin menggunakan prosedur di Mengimpor data ke basis data Amazon RDS MariaDB atau MySQL dengan lebih sedikit waktu henti.

   Untuk Linux, macOS, atau Unix:

```
mysqldump --databases database_name \
    --single-transaction \
    --compress \
    --order-by-primary \
    -u local_user \
    -plocal_password | mysql \
        --host=hostname \
        --port=3306 \
        -u RDS_user_name \
        -pRDS_password
```

   Untuk Windows:

```
mysqldump --databases database_name ^
    --single-transaction ^
    --compress ^
    --order-by-primary ^
    -u local_user ^
    -plocal_password | mysql ^
        --host=hostname ^
        --port=3306 ^
        -u RDS_user_name ^
        -pRDS_password
```

> **Note**
>
> Pastikan tidak ada spasi di antara opsi `-p` dan sandi yang dimasukkan.

Untuk menentukan nama host, nama pengguna, port, dan kata sandi untuk menghubungkan ke instans Amazon RDS DB Anda, gunakan opsi `--host`, `--user (-u)`, `--port`, dan `-p` dalam perintah `mysql`. Nama host adalah nama Domain Name Service (DNS) dari titik akhir instans DB

Amazon RDS, misalnya `myinstance.123456789012.us-east-1.rds.amazonaws.com`. Anda dapat menemukan nilai titik akhir dalam detail instans di AWS Management Console.

4. Jadikan instans DB MySQL atau MariaDB sumber sebagai writable (dapat diubah) lagi.

```
mysql> SET GLOBAL read_only = OFF;
mysql> UNLOCK TABLES;
```

Untuk informasi lebih lanjut tentang cara membuat cadangan untuk digunakan dengan replikasi, lihat dokumentasi MySQL.

5. Di AWS Management Console, tambahkan alamat IP server yang meng-host basis data eksternal ke grup keamanan cloud privat virtual (VPC) untuk instans DB Amazon RDS. Untuk informasi selengkapnya tentang cara memodifikasi grup keamanan VPC, lihat Grup keamanan untuk VPC Anda dalam Panduan Pengguna Amazon Virtual Private Cloud.

Alamat IP dapat berubah jika kondisi berikut terpenuhi:

- Anda menggunakan alamat IP publik untuk komunikasi antara instans sumber eksternal dan instans basis data.
- Instans sumber eksternal dihentikan dan dimulai ulang.

Jika semua kondisi ini terpenuhi, verifikasikan alamat IP sebelum menambahkannya.

Anda mungkin juga perlu mengonfigurasi jaringan lokal Anda untuk mengizinkan koneksi dari alamat IP instans Amazon RDS DB Anda. Anda melakukan ini agar jaringan lokal Anda dapat berkomunikasi dengan instans MySQL atau MariaDB eksternal Anda. Untuk menemukan alamat IP dari instans Amazon RDS DB, gunakan perintah `host`.

```
host db_instance_endpoint
```

Nama host adalah nama DNS dari titik akhir instans DB Amazon RDS.

6. Menggunakan klien pilihan Anda, hubungkan ke instans eksternal dan buat pengguna untuk digunakan untuk replikasi. Gunakan akun ini semata-mata untuk replikasi dan batasi hanya untuk domain Anda guna meningkatkan keamanan. Berikut adalah contohnya.

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
```

> **Note**
>
> Tentukan kata sandi selain prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

7. Untuk instans eksternal, berikan hak akses `REPLICATION CLIENT` dan `REPLICATION SLAVE` kepada pengguna replikasi Anda. Misalnya, untuk memberikan hak akses `REPLICATION CLIENT` dan `REPLICATION SLAVE` pada semua basis data untuk pengguna '`repl_user`' bagi domain Anda, jalankan perintah berikut.

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
```

8. Jadikan instans Amazon RDS DB sebagai replika. Untuk melakukannya, pertama-tama hubungkan ke instans DB Amazon RDS sebagai pengguna master. Kemudian identifikasi basis data MySQL atau MariaDB eksternal sebagai instans sumber menggunakan perintah . Gunakan nama file log master dan posisi log master yang Anda tentukan pada langkah 2. Berikut adalah contohnya.

```
CALL mysql.rds_set_external_master ('mymasterserver.mydomain.com', 3306,
 'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 0);
```

> **Note**
>
> Di RDS for MySQL, Anda dapat memilih untuk menggunakan replikasi tertunda dengan menjalankan prosedur tersimpan mysql.rds_set_external_master_with_delay sebagai gantinya. Di RDS for MySQL, salah satu alasan menggunakan replikasi tertunda adalah untuk mengaktifkan pemulihan bencana dengan prosedur tersimpan mysql.rds_start_replication_until. Saat ini, RDS untuk MariaDB mendukung replikasi tertunda tetapi tidak mendukung prosedur `mysql.rds_start_replication_until`.

9. Di instans DB Amazon RDS, terbitkan perintah mysql.rds_start_replication untuk memulai replikasi.

```
CALL mysql.rds_start_replication;
```

Opsi untuk mesin basis data MariaDB

Pada bagian berikut ini, Anda dapat menemukan deskripsi untuk opsi, atau fitur tambahan, yang tersedia untuk instans Amazon RDS yang menjalankan Mesin DB MariaDB. Untuk mengaktifkan opsi ini, Anda menambahkannya ke grup opsi kustom, lalu mengaitkankannya dengan instans DB Anda. Untuk informasi selengkapnya tentang cara menggunakan grup opsi, lihat Menggunakan grup opsi.

Amazon RDS mendukung opsi berikut untuk MariaDB:

ID Opsi	Versi mesin
`MARIADB_AUDIT_PLUGIN`	MariaDB 10.3 dan yang lebih tinggi

Dukungan MariaDB Audit Plugin

Amazon RDS mendukung penggunaan MariaDB Audit Plugin di instans basis data MariaDB. MariaDB Plugin Audit mencatat aktivitas basis data seperti pengguna yang masuk ke basis data, kueri yang dijalankan terhadap basis data, dan banyak lagi. Catatan aktivitas basis data disimpan dalam file log.

Pengaturan opsi Audit Plugin

Amazon RDS mendukung pengaturan berikut untuk opsi MariaDB Audit Plugin.

> **Note**
>
> Jika Anda tidak mengonfigurasi pengaturan opsi di konsol RDS, RDS akan menggunakan pengaturan default.

Pengaturan opsi	Nilai valid	Nilai default	Deskripsi
`SERVER_AUDIT_FILE_PATH`	`/rdsdbdata/log/audit/`	`/rdsdbdata/log/audit/`	Lokasi file log. File log berisi catatan aktivitas yang ditentukan dalam `SERVER_AUDIT_EVENTS` . Untuk informasi selengkap

Pengaturan opsi	Nilai valid	Nilai default	Deskripsi
			nya, lihat Melihat dan mencantumkan file log basis data dan File log basis data MariaDB.
SERVER_AUDIT_FILE_ROTATE_SIZE	1–1000000000	1000000	Ukuran dalam byte. Jika ukuran ini tercapai, file akan diputar. Untuk informasi selengkapnya, lihat Ukuran file log.
SERVER_AUDIT_FILE_ROTATIONS	0–100	9	Jumlah rotasi log yang akan disimpan ketika server_audit_output_type=file . Jika diatur ke 0, file log tidak akan pernah diputar. Untuk informasi selengkapnya, lihat Ukuran file log dan Mengunduh file log basis data.

Pengaturan opsi	Nilai valid	Nilai default	Deskripsi
SERVER_AUDIT_EVENTS	CONNECT, QUERY, TABLE, QUERY_DDL , QUERY_DML , QUERY_DML_NO_SELECT , QUERY_DCL	CONNECT, QUERY	Jenis aktivitas yang akan dicatat di log. Penginstalan MariaDB Audit Plugin itu sendiri juga akan dicatat. • CONNECT: Mencatat koneksi yang berhasil dan tidak berhasil ke basis data, dan pemutusan dari basis data. • QUERY: Mencatat teks semua kueri yang dijalankan terhadap basis data. • TABLE: Mencatat tabel yang terpengaruh oleh kueri ketika kueri dijalankan terhadap basis data. • QUERY_DDL : Mirip dengan peristiwa QUERY, tetapi hanya mengembalikan kueri bahasa definisi data (DDL) (CREATE, ALTER, dan sebagainya). • QUERY_DML : Mirip dengan peristiwa QUERY, tetapi hanya mengembalikan kueri bahasa manipulasi data (DML) (INSERT, UPDATE, dan sebagainya, serta SELECT). • QUERY_DML_NO_SELECT : Mirip dengan peristiwa QUERY_DML , tetapi tidak mencatat kueri SELECT. • QUERY_DCL : Mirip dengan peristiwa QUERY, tetapi hanya mengembalikan kueri bahasa kontrol data (DCL) (GRANT, REVOKE, dan sebagainya).

Pengaturan opsi	Nilai valid	Nilai default	Deskripsi
SERVER_AUDIT_INCL_USERS	Beberapa nilai yang dipisahkan koma	Tidak ada	Hanya menyertakan aktivitas dari pengguna tertentu. Secara default, aktivitas dicatat untuk semua pengguna. SERVER_AUDIT_INCL_USERS dan SERVER_AUDIT_EXCL_USERS sama-sama bersifat eksklusif. Jika Anda menambahkan nilai ke SERVER_AUDIT_INCL_USERS , pastikan tidak ada nilai yang ditambahkan ke SERVER_AUDIT_EXCL_USERS .

Pengaturan opsi	Nilai valid	Nilai default	Deskripsi
SERVER_AUDIT_EXCL_USERS	Beberapa nilai yang dipisahkan koma	Tidak ada	Mengecualikan aktivitas dari pengguna tertentu. Secara default, aktivitas dicatat untuk semua pengguna. SERVER_AUDIT_INCL_USERS dan SERVER_AUDIT_EXCL_USERS sama-sama bersifat eksklusif. Jika Anda menambahkan nilai ke SERVER_AUDIT_EXCL_USERS , pastikan tidak ada nilai yang ditambahkan ke SERVER_AUDIT_INCL_USERS . Pengguna rdsadmin membuat kueri basis data setiap detik untuk memeriksa kondisi basis data. Bergantung pada pengaturan Anda yang lain, aktivitas ini mungkin dapat menyebabkan ukuran file log Anda bertambah sangat besar dengan sangat cepat. Jika Anda tidak perlu mencatat aktivitas ini, tambahkan pengguna rdsadmin ke daftar SERVER_AUDIT_EXCL_USERS . **Note** Aktivitas CONNECT selalu dicatat untuk semua pengguna meskipun pengguna ditentukan untuk pengaturan opsi ini.
SERVER_AUDIT_LOGGING	ON	ON	Pencatatan aktif. Satu-satunya nilai yang valid adalah ON. Amazon RDS tidak mendukung penonaktifan pencatatan. Jika Anda ingin menonaktifkan pencatatan, hapus MariaDB Audit Plugin. Untuk informasi selengkapnya, lihat Menghapus MariaDB Audit Plugin.

Pengaturan opsi	Nilai valid	Nilai default	Deskripsi
SERVER_AUDIT_QUERY_LOG_LIMIT	0–2147483647	1024	Batas panjang string kueri dalam sebuah catatan.

Menambahkan MariaDB Audit Plugin

Proses umum untuk menambahkan MariaDB Audit Plugin ke instans DB adalah sebagai berikut:

1. Buat grup opsi baru, atau salin atau ubah grup opsi yang ada.
2. Tambahkan opsi ke grup opsi.
3. Kaitkan grup opsi dengan instans DB.

Setelah menambahkan MariaDB Audit Plugin, Anda tidak perlu memulai ulang instans DB Anda. Setelah grup opsi aktif, audit akan segera dimulai.

Untuk menambahkan MariaDB Audit Plugin

1. Tentukan grup opsi yang ingin Anda gunakan. Anda dapat membuat grup opsi baru atau menggunakan grup opsi yang ada. Jika Anda ingin menggunakan grup opsi yang ada, lewati ke langkah berikutnya. Jika tidak, buat grup opsi DB kustom. Pilih mariadb untuk Mesin, lalu pilih 10.3 atau yang lebih tinggi untuk Versi mesin utama. Untuk informasi selengkapnya, lihat Membuat grup opsi.
2. Tambahkan opsi MARIADB_AUDIT_PLUGIN ke grup opsi, lalu konfigurasi pengaturan opsi. Untuk informasi selengkapnya tentang cara menambahkan opsi, lihat Menambahkan opsi ke grup opsi. Untuk informasi selengkapnya tentang setiap pengaturan, lihat Pengaturan opsi Audit Plugin.
3. Terapkan grup opsi ke instans DB baru atau yang sudah ada.
   - Untuk instans DB baru, Anda menerapkan grup opsi saat Anda meluncurkan instans. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.

- Untuk instans DB yang sudah ada, terapkan grup opsi dengan mengubah instans DB dan melampirkan grup opsi baru. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Melihat dan mengunduh log MariaDB Audit Plugin

Setelah mengaktifkan MariaDB Audit Plugin, Anda dapat mengakses hasilnya di file log dengan cara yang sama seperti Anda mengakses file log berbasis teks lainnya. File log audit terletak di `/rdsdbdata/log/audit/`. Untuk informasi tentang cara melihat file log di konsol, lihat Melihat dan mencantumkan file log basis data. Untuk informasi tentang cara mengunduh file log, lihat Mengunduh file log basis data.

Mengubah pengaturan MariaDB Audit Plugin

Setelah mengaktifkan MariaDB Audit Plugin, Anda dapat mengubah pengaturan untuk plugin. Untuk informasi selengkapnya tentang cara mengubah pengaturan opsi, lihat Memodifikasi pengaturan opsi. Untuk informasi selengkapnya tentang setiap pengaturan, lihat Pengaturan opsi Audit Plugin.

Menghapus MariaDB Audit Plugin

Amazon RDS tidak mendukung penonaktifan log di MariaDB Audit Plugin. Namun, Anda dapat menghapus plugin dari instans DB. Jika Anda menghapus MariaDB Audit Plugin, instans DB dimulai ulang secara otomatis untuk menghentikan audit.

Untuk menghapus MariaDB Audit Plugin dari instans DB, lakukan salah satu hal berikut:

- Hapus opsi MariaDB Audit Plugin dari grup opsi tempatnya berada. Perubahan ini memengaruhi semua instans DB yang menggunakan grup opsi tersebut. Untuk informasi selengkapnya, lihat Menghapus opsi dari grup opsi
- Ubah instans DB dan tentukan grup opsi berbeda yang tidak menyertakan plugin. Perubahan ini memengaruhi instans DB tunggal. Anda dapat menentukan grup opsi default (kosong) atau grup opsi kustom yang berbeda. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Parameter untuk MariaDB

Secara default, instans DB MariaDB menggunakan grup parameter DB yang spesifik untuk basis data MariaDB. Grup parameter ini berisi beberapa, tetapi tidak semua, parameter yang terdapat dalam grup parameter DB Amazon RDS untuk mesin basis data MySQL. Grup parameter ini juga berisi sejumlah parameter khusus MariaDB yang baru. Untuk informasi tentang cara menangani grup parameter dan mengatur parameter, lihat Bekerja dengan grup parameter.

Melihat parameter MariaDB

Parameter RDS for MariaDB diatur ke nilai default mesin penyimpanan yang telah Anda pilih. Untuk informasi selengkapnya tentang parameter MariaDB, lihat dokumentasi MariaDB. Untuk informasi selengkapnya tentang mesin penyimpanan MariaDB, lihat Mesin penyimpanan yang didukung untuk MariaDB di Amazon RDS.

Anda dapat melihat parameter yang tersedia untuk versi RDS for MariaDB tertentu menggunakan konsol RDS atau AWS CLI. Untuk informasi tentang cara melihat parameter dalam grup parameter MariaDB di konsol RDS, lihat Melihat nilai parameter untuk grup parameter DB.

Dengan menggunakan AWS CLI, Anda dapat melihat parameter untuk versi RDS for MariaDB dengan menjalankan perintah `describe-engine-default-parameters`. Tentukan salah satu dari nilai-nilai berikut untuk opsi `--db-parameter-group-family`:

- `mariadb10.11`
- `mariadb10.6`
- `mariadb10.5`
- `mariadb10.4`
- `mariadb10.3`

Misalnya, untuk melihat parameter RDS for MariaDB versi 10.6, jalankan perintah berikut.

```
aws rds describe-engine-default-parameters --db-parameter-group-family mariadb10.6
```

Output Anda akan terlihat serupa dengan yang berikut ini.

```
{
    "EngineDefaults": {
        "Parameters": [
```

```
            {
                "ParameterName": "alter_algorithm",
                "Description": "Specify the alter table algorithm.",
                "Source": "engine-default",
                "ApplyType": "dynamic",
                "DataType": "string",
                "AllowedValues": "DEFAULT,COPY,INPLACE,NOCOPY,INSTANT",
                "IsModifiable": true
            },
            {
                "ParameterName": "analyze_sample_percentage",
                "Description": "Percentage of rows from the table ANALYZE TABLE will
 sample to collect table statistics.",
                "Source": "engine-default",
                "ApplyType": "dynamic",
                "DataType": "float",
                "AllowedValues": "0-100",
                "IsModifiable": true
            },
            {
                "ParameterName": "aria_block_size",
                "Description": "Block size to be used for Aria index pages.",
                "Source": "engine-default",
                "ApplyType": "static",
                "DataType": "integer",
                "AllowedValues": "1024-32768",
                "IsModifiable": false
            },
            {
                "ParameterName": "aria_checkpoint_interval",
                "Description": "Interval in seconds between automatic checkpoints.",
                "Source": "engine-default",
                "ApplyType": "dynamic",
                "DataType": "integer",
                "AllowedValues": "0-4294967295",
                "IsModifiable": true
            },
        ...
```

Untuk hanya mencantumkan parameter yang dapat dimodifikasi untuk RDS for MariaDB versi 10.6, jalankan perintah berikut.

Untuk Linux, macOS, atau Unix:

```
aws rds describe-engine-default-parameters --db-parameter-group-family mariadb10.6 \
   --query 'EngineDefaults.Parameters[?IsModifiable==`true`]'
```

Untuk Windows:

```
aws rds describe-engine-default-parameters --db-parameter-group-family mariadb10.6 ^
   --query "EngineDefaults.Parameters[?IsModifiable==`true`]"
```

Parameter MySQL yang tidak tersedia

Parameter MySQL berikut tidak tersedia dalam grup parameter DB khusus MariaDB:

- bind_address
- binlog_error_action
- binlog_gtid_simple_recovery
- binlog_max_flush_queue_time
- binlog_order_commits
- binlog_row_image
- binlog_rows_query_log_events
- binlogging_impossible_mode
- block_encryption_mode
- core_file
- default_tmp_storage_engine
- div_precision_increment
- end_markers_in_json
- enforce_gtid_consistency
- eq_range_index_dive_limit
- explicit_defaults_for_timestamp
- gtid_executed
- gtid-mode
- gtid_next
- gtid_owned
- gtid_purged

- log_bin_basename
- log_bin_index
- log_bin_use_v1_row_events
- log_slow_admin_statements
- log_slow_slave_statements
- log_throttle_queries_not_using_indexes
- master-info-repository
- optimalr_trace
- optimizer_trace_features
- optimizer_trace_limit
- optimizer_trace_max_mem_size
- optimizer_trace_offset
- relay_log_info_repository
- rpl_stop_slave_timeout
- slave_parallel_workers
- slave_pending_jobs_size_max
- slave_rows_search_algorithms
- storage_engine
- table_open_cache_instances
- timed_mutexes
- transaction_allow_batching
- validate-password
- validate_password_dictionary_file
- validate_password_length
- validate_password_mixed_case_count
- validate_password_number_count
- validate_password_policy
- validate_password_special_char_count

Untuk informasi selengkapnya tentang parameter MySQL, lihat dokumentasi MySQL.

Memigrasikan data dari snapshot DB MySQL ke instans DB MariaDB

Anda dapat memigrasi snapshot DB RDS for MySQL ke instans DB baru yang menjalankan MariaDB menggunakan AWS Management Console, AWS CLI, atau API Amazon RDS. Anda harus menggunakan snapshot DB yang dibuat dari instans DB Amazon RDS yang menjalankan MySQL 5.6 atau 5.7. Untuk mempelajari cara membuat snapshot DB RDS for MySQL, lihat Membuat snapshot DB untuk instans DB Single-AZ.

Memigrasi snapshot tidak memengaruhi instans DB asli tempat snapshot diambil. Anda dapat menguji dan memvalidasi instans DB baru sebelum mengalihkan lalu lintas ke instans DB sebagai pengganti instans DB asli.

Setelah Anda bermigrasi dari MySQL ke MariaDB, instans DB MariaDB dikaitkan dengan grup parameter dan grup opsi DB default. Setelah memulihkan snapshot DB, Anda dapat mengaitkan grup parameter DB kustom dengan instans DB baru. Namun, grup parameter MariaDB memiliki kumpulan variabel sistem berbeda yang dapat dikonfigurasi. Untuk informasi tentang perbedaan antara variabel sistem MySQL dan MariaDB, lihat Perbedaan Variabel Sistem antara MariaDB dan MySQL. Untuk mempelajari tentang grup parameter DB, lihat Bekerja dengan grup parameter. Untuk mempelajari tentang grup opsi, lihat Menggunakan grup opsi.

Melakukan migrasi

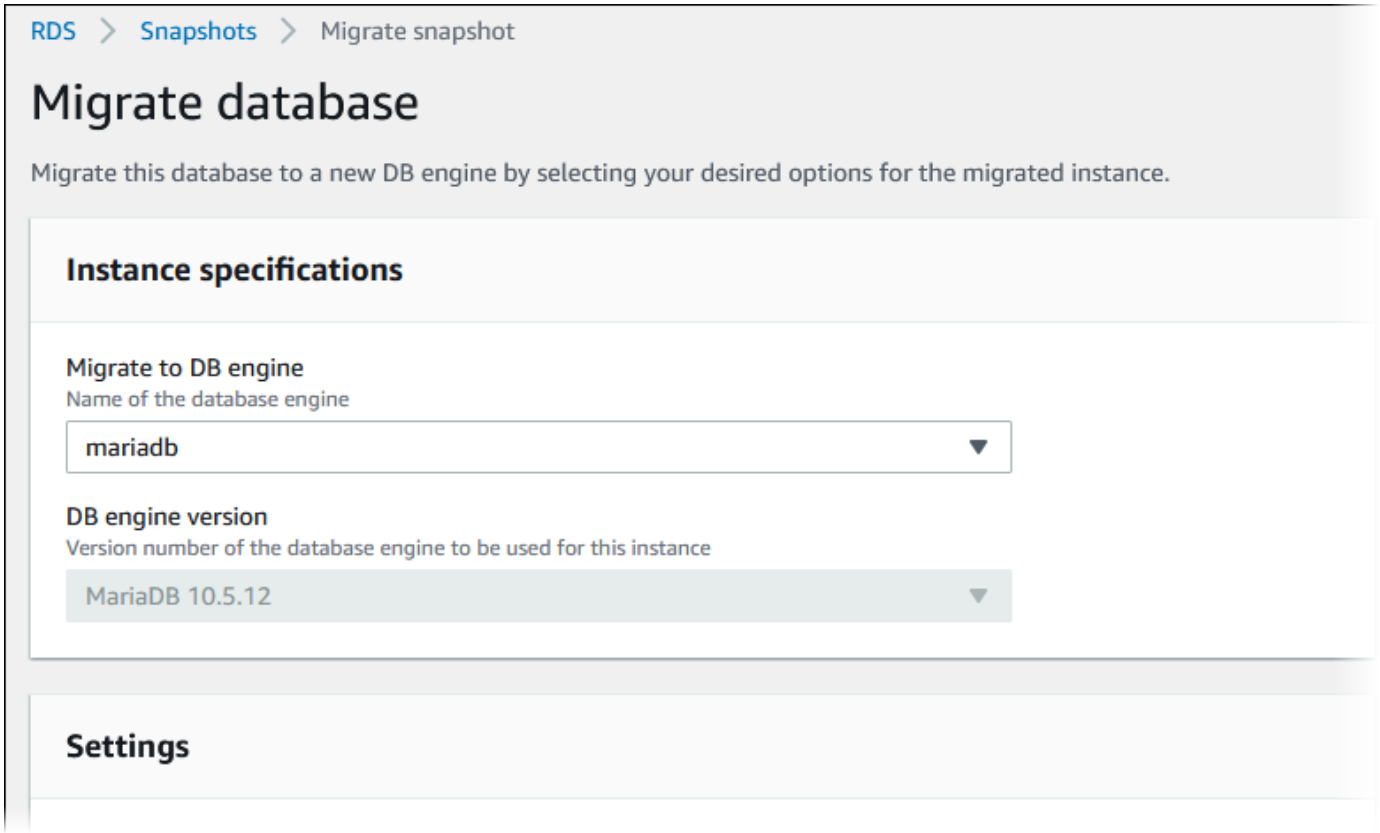
Anda dapat memigrasi snapshot DB RDS for MySQL ke instans DB MariaDB baru menggunakan AWS Management Console, AWS CLI, atau API RDS.

Konsol

Untuk memigrasikan snapshot DB MySQL ke instans DB MariaDB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Snapshot, lalu pilih snapshot DB MySQL yang ingin Anda migrasi.
3. Untuk Tindakan, pilih Migrasikan snapshot. Halaman Migrasikan basis data akan muncul.
4. Untuk Migrasikan ke Mesin DB, pilih mariadb.

   Amazon RDS memilih Versi mesin DB secara otomatis. Anda tidak dapat mengubah versi mesin DB.



5. Untuk bagian yang tersisa, tentukan pengaturan instans DB Anda. Untuk informasi tentang setiap pengaturan, lihat Pengaturan untuk instans DB.
6. Pilih Migrasikan.

AWS CLI

Untuk memigrasikan data dari snapshot DB MySQL ke instans DB MariaDB, gunakan perintah AWS CLI `restore-db-instance-from-db-snapshot` dengan parameter berikut ini:

- -- db-instance-identifier — Nama instans DB untuk membuat dari snapshot DB.
- -- db-snapshot-identifier — Pengidentifikasi untuk snapshot DB untuk dipulihkan dari.
- --engine – Mesin basis data digunakan untuk instans baru.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds restore-db-instance-from-db-snapshot \
    --db-instance-identifier newmariadbinstance \
```

```
    --db-snapshot-identifier mysqlsnapshot \
    --engine mariadb
```

Untuk Windows:

```
aws rds restore-db-instance-from-db-snapshot ^
    --db-instance-identifier newmariadbinstance ^
    --db-snapshot-identifier mysqlsnapshot ^
    --engine mariadb
```

API

Untuk memigrasikan data dari snapshot DB MySQL ke instans DB MariaDB, panggil operasi API Amazon RDS `RestoreDBInstanceFromDBSnapshot`.

Ketidakcocokkan antara MariaDB dan MySQL

Ketidakcocokkan antara MySQL dan MariaDB meliputi hal berikut:

- Anda tidak dapat memigrasi snapshot DB yang dibuat dengan MySQL 8.0 ke MariaDB.
- Jika basis data MySQL sumber menggunakan hash kata sandi SHA256, pastikan untuk mengatur ulang kata sandi pengguna yang sudah di-hash SHA256 sebelum Anda terhubung ke basis data MariaDB. Kode berikut menunjukkan cara mengatur ulang kata sandi yang sudah di-hash SHA256.

  ```
  SET old_passwords = 0;
  UPDATE mysql.user SET plugin = 'mysql_native_password',
  Password = PASSWORD('new_password')
  WHERE (User, Host) = ('master_user_name', %);
  FLUSH PRIVILEGES;
  ```

- Jika akun pengguna master RDS Anda menggunakan hash kata sandi SHA-256, pastikan untuk mengatur ulang kata sandi menggunakan AWS Management Console, perintah `modify-db-instance` AWS CLI, atau Operasi API RDS ModifyDBInstance. Untuk mengetahui informasi tentang cara mengubah instans DB, lihat Memodifikasi instans DB Amazon RDS.
- MariaDB tidak mendukung plugin Memcached. Namun, data yang digunakan oleh plugin Memcached disimpan sebagai tabel InnoDB. Setelah Anda memigrasikan snapshot DB MySQL, Anda dapat mengakses data yang digunakan oleh plugin Memcached menggunakan SQL. Untuk

informasi selengkapnya tentang basis data innodb_memcache, lihat Internal Plugin memcached InnoDB.

MariaDB di referensi Amazon RDS SQL

Pada bagian berikut, Anda dapat menemukan deskripsi prosedur tersimpan sistem yang tersedia untuk instans Amazon RDS yang menjalankan mesin DB MariaDB.

Anda dapat menggunakan prosedur tersimpan sistem yang tersedia untuk instans DB MySQL dan instans DB MariaDB. Prosedur tersimpan ini didokumentasikan di RDS for MySQL. Instans DB MariaDB mendukung semua prosedur tersimpan, kecuali untuk `mysql.rds_start_replication_until` dan `mysql.rds_start_replication_until_gtid`.

Selain itu, prosedur tersimpan sistem berikut hanya didukung untuk instans DB Amazon RDS yang menjalankan MariaDB:

- mysql.rds_replica_status
- mysql.rds_set_external_master_gtid
- mysql.rds_kill_query_id

mysql.rds_replica_status

Menunjukkan status replikasi replika baca MariaDB.

Panggil prosedur ini pada replika baca untuk menampilkan informasi status parameter penting utas replika.

Sintaks

```
CALL mysql.rds_replica_status;
```

Catatan penggunaan

Prosedur ini hanya didukung untuk instans DB MariaDB yang menjalankan MariaDB versi 10.5 dan yang lebih tinggi.

Prosedur ini setara dengan perintah `SHOW REPLICA STATUS`. Perintah ini tidak didukung untuk MariaDB versi 10.5 dan instans DB yang lebih tinggi.

Dalam MariaDB versi sebelumnya, perintah `SHOW SLAVE STATUS` yang setara memerlukan hak istimewa `REPLICATION SLAVE`. Dalam MariaDB versi 10.5 dan yang lebih tinggi, diperlukan hak

istimewa `REPLICATION REPLICA ADMIN`. Untuk melindungi manajemen RDS untuk MariaDB 10.5 dan instans DB yang lebih tinggi, hak istimewa baru ini tidak diberikan kepada pengguna master RDS.

Contoh

Contoh berikut menunjukkan status replika baca MariaDB:

```
call mysql.rds_replica_status;
```

Responsnya terlihat seperti berikut:

```
*************************** 1. row ***************************
               Replica_IO_State: Waiting for master to send event
                    Source_Host: XX.XX.XX.XXX
                    Source_User: rdsrepladmin
                    Source_Port: 3306
                  Connect_Retry: 60
                Source_Log_File: mysql-bin-changelog.003988
            Read_Source_Log_Pos: 405
                 Relay_Log_File: relaylog.011024
                  Relay_Log_Pos: 657
          Relay_Source_Log_File: mysql-bin-changelog.003988
             Replica_IO_Running: Yes
            Replica_SQL_Running: Yes
                Replicate_Do_DB:
            Replicate_Ignore_DB:
             Replicate_Do_Table:
         Replicate_Ignore_Table:
 mysql.rds_sysinfo,mysql.rds_history,mysql.rds_replication_status
        Replicate_Wild_Do_Table:
    Replicate_Wild_Ignore_Table:
                     Last_Errno: 0
                     Last_Error:
                   Skip_Counter: 0
            Exec_Source_Log_Pos: 405
                Relay_Log_Space: 1016
                Until_Condition: None
                 Until_Log_File:
                  Until_Log_Pos: 0
             Source_SSL_Allowed: No
             Source_SSL_CA_File:
             Source_SSL_CA_Path:
```

```
                Source_SSL_Cert:
              Source_SSL_Cipher:
                 Source_SSL_Key:
          Seconds_Behind_Master: 0
  Source_SSL_Verify_Server_Cert: No
                  Last_IO_Errno: 0
                  Last_IO_Error:
                 Last_SQL_Errno: 0
                 Last_SQL_Error:
    Replicate_Ignore_Server_Ids:
               Source_Server_Id: 807509301
                 Source_SSL_Crl:
             Source_SSL_Crlpath:
                     Using_Gtid: Slave_Pos
                    Gtid_IO_Pos: 0-807509301-3980
        Replicate_Do_Domain_Ids:
    Replicate_Ignore_Domain_Ids:
                  Parallel_Mode: optimistic
                      SQL_Delay: 0
            SQL_Remaining_Delay: NULL
      Replica_SQL_Running_State: Reading event from the relay log
             Replica_DDL_Groups: 15
Replica_Non_Transactional_Groups: 0
    Replica_Transactional_Groups: 3658
1 row in set (0.000 sec)

Query OK, 0 rows affected (0.000 sec)
```

mysql.rds_set_external_master_gtid

Mengonfigurasi replikasi berbasis GTID dari instans MariaDB yang berjalan secara eksternal ke Amazon RDS ke instans DB MariaDB. Prosedur tersimpan ini hanya didukung jika instans MariaDB eksternal memiliki versi 10.0.24 atau lebih tinggi. Saat menyiapkan replikasi yang salah satu atau kedua instansnya tidak mendukung pengidentifikasi transaksi global (GTID) MariaDB, gunakan .

Menggunakan GTID untuk replikasi menyediakan fitur keselamatan jatuh yang tidak ditawarkan oleh replikasi log biner, jadi kami menyarankan Anda menggunakannya saat fitur ini didukung dalam replikasi.

Sintaks

```
CALL mysql.rds_set_external_master_gtid(
```

```
  host_name
  , host_port
  , replication_user_name
  , replication_user_password
  , gtid
  , ssl_encryption
);
```

Parameter

host_name

String. Nama host atau alamat IP instans MariaDB yang berjalan secara eksternal ke Amazon RDS yang akan menjadi instans sumber.

host_port

Integer. Port yang digunakan oleh instans MariaDB yang berjalan secara eksternal ke Amazon RDS yang akan dikonfigurasi sebagai instans sumber. Jika konfigurasi jaringan Anda menyertakan replikasi port SSH yang mengonversi nomor port, tentukan nomor port yang diekspos oleh SSH.

replication_user_name

String. ID pengguna dengan izin `REPLICATION SLAVE` dalam instans DB MariaDB yang akan dikonfigurasi sebagai replika baca.

replication_user_password

String. Kata sandi ID pengguna yang ditentukan dalam `replication_user_name`.

gtid

String. ID transaksi global pada instans sumber untuk memulai replikasi.

Anda dapat menggunakan `@@gtid_current_pos` untuk mendapatkan GTID saat ini jika instans sumber telah dikunci saat Anda mengonfigurasi replikasi, sehingga log biner tidak berubah di antara titik-titik ketika Anda mendapatkan GTID dan ketika replikasi dimulai.

Atau, jika Anda menggunakan `mysqldump` versi 10.0.13 atau yang lebih tinggi untuk mengisi instans replika sebelum memulai replikasi, Anda dapat mendapatkan posisi GTID di output menggunakan opsi `--master-data` atau `--dump-slave`. Jika Anda tidak menggunakan `mysqldump` versi 10.0.13 atau yang lebih tinggi, Anda dapat menjalankan `SHOW MASTER STATUS` atau menggunakan opsi `mysqldump` yang sama untuk mendapatkan nama dan posisi

file log biner, lalu mengonversinya ke GTID dengan menjalankan `BINLOG_GTID_POS` di luar instans MariaDB:

```
SELECT BINLOG_GTID_POS('<binary log file name>', <binary log file position>);
```

Untuk informasi selengkapnya tentang implementasi GTID MariaDB, buka ID transaksi global dalam dokumentasi MariaDB.

ssl_encryption

Nilai yang menentukan apakah enkripsi Secure Socket Layer (SSL) digunakan pada koneksi replikasi. 1 menentukan untuk menggunakan enkripsi SSL, 0 menentukan untuk tidak menggunakan enkripsi. Default-nya adalah 0.

> **Note**
>
> Opsi `MASTER_SSL_VERIFY_SERVER_CERT` tidak didukung. Opsi ini diatur ke 0, yang berarti koneksi dienkripsi, tetapi sertifikat tidak diverifikasi.

Catatan penggunaan

Prosedur `mysql.rds_set_external_master_gtid` harus dijalankan oleh pengguna master. Prosedur ini harus dijalankan pada instans DB MariaDB yang Anda konfigurasikan sebagai replika instans MariaDB yang berjalan secara eksternal ke Amazon RDS. Sebelum menjalankan `mysql.rds_set_external_master_gtid`, Anda harus sudah mengonfigurasi instans MariaDB yang berjalan secara eksternal ke Amazon RDS sebagai instans sumber. Untuk informasi selengkapnya, lihat Mengimpor data ke instans basis data MariaDB.

> **Warning**
>
> Jangan gunakan `mysql.rds_set_external_master_gtid` untuk mengelola replikasi antara dua instans DB Amazon RDS. Gunakan hanya saat mereplikasi instans MariaDB yang berjalan secara eksternal ke RDS. Untuk informasi tentang pengelolaan replikasi antara instans DB Amazon RDS, lihat Menggunakan replika baca instans DB.

Setelah memanggil `mysql.rds_set_external_master_gtid` untuk mengonfigurasi instans DB Amazon RDS sebagai replika baca, Anda dapat memanggil mysql.rds_start_replication pada replika

baca untuk memulai proses replikasi. Anda dapat memanggil mysql.rds_reset_external_master untuk menghapus konfigurasi replika baca.

Saat `mysql.rds_set_external_master_gtid` dipanggil, Amazon RDS mencatat waktu, pengguna, dan tindakan "set master" di tabel `mysql.rds_history` dan `mysql.rds_replication_status`.

Contoh

Ketika dijalankan pada instans DB MariaDB, contoh berikut mengonfigurasinya sebagai replika instans MariaDB yang berjalan secara eksternal ke Amazon RDS.

```
call mysql.rds_set_external_master_gtid
 ('Sourcedb.some.com',3306,'ReplicationUser','SomePassW0rd','0-123-456',0);
```

mysql.rds_kill_query_id

Mengakhiri kueri yang berjalan pada server MariaDB.

Sintaksis

```
CALL mysql.rds_kill_query_id(queryID);
```

Parameter

queryID

Integer. Identitas kueri yang akan diakhiri.

Catatan penggunaan

Untuk menghentikan kueri yang berjalan pada server MariaDB, gunakan prosedur `mysql.rds_kill_query_id` dan teruskan ID kueri tersebut. Untuk mendapatkan ID kueri, kueri MariaDB Tabel PROSESLIST skema informasi, seperti yang ditunjukkan berikut ini:

```
SELECT USER, HOST, COMMAND, TIME, STATE, INFO, QUERY_ID FROM
                INFORMATION_SCHEMA.PROCESSLIST WHERE USER = '<user name>';
```

Koneksi ke server MariaDB dipertahankan.

Contoh

Contoh berikut mengakhiri kueri dengan ID kueri 230040:

```
call mysql.rds_kill_query_id(230040);
```

Zona waktu lokal untuk instans basis data MariaDB

Secara bawaan, zona waktu untuk instans basis data MariaDB adalah Waktu Universal Terkoordinasi (UTC). Anda dapat mengatur zona waktu untuk instans basis data Anda ke zona waktu lokal untuk aplikasi Anda.

Untuk mengatur zona waktu lokal bagi instans basis data, atur parameter `time_zone` di grup parameter bagi instans basis data itu ke salah satu nilai yang didukung yang tercantum belakangan di bagian ini. Saat Anda mengatur parameter `time_zone` untuk grup parameter, semua instans basis data dan replika baca yang menggunakan grup parameter itu berubah untuk menggunakan zona waktu lokal baru. Lihat informasi tentang pengaturan parameter di grup parameter di Bekerja dengan grup parameter.

Setelah Anda mengatur zona waktu lokal, semua koneksi baru ke basis data mencerminkan perubahan itu. Jika ada koneksi terbuka ke basis data Anda ketika Anda mengubah zona waktu lokal, Anda tidak akan melihat pembaruan zona waktu lokal sampai Anda menutup koneksi itu dan membuka koneksi baru.

Anda dapat mengatur zona waktu lokal yang berbeda untuk sebuah instans basis data dan satu atau beberapa replika baca. Untuk melakukannya, gunakan grup parameter yang berbeda untuk instans basis data dan replika dan atur parameter `time_zone` dalam setiap grup parameter ke zona waktu lokal yang berbeda.

Jika Anda mereplikasi di seluruhWilayah AWS, instans basis data sumber dan replika baca menggunakan grup parameter yang berbeda (grup parameter bersifat unik bagi Wilayah AWS). Untuk menggunakan zona waktu lokal yang sama bagi setiap instans, Anda harus mengatur parameter `time_zone` dalam grup-grup parameter instans itu dan replika baca.

Saat Anda memulihkan instans basis data dari cuplikan basis data, zona waktu lokal diatur ke UTC. Anda dapat memperbarui zona waktu ke zona waktu lokal Anda setelah pemulihan selesai. Jika Anda memulihkan instans basis data ke suatu titik waktu, maka zona waktu lokal untuk instans basis data yang dipulihkan adalah setelan zona waktu dari grup parameter instans basis data yang dipulihkan.

Internet Assigned Numbers Authority (IANA) menerbitkan zona-zona waktu baru di https://www.iana.org/time-zones beberapa kali dalam setahun. Setiap kali RDS menerbitkan rilis pemeliharaan kecil baru MariaDB, rilis itu akan dikirimkan beserta data zona waktu terbaru pada saat terbit. Jika Anda menggunakan versi RDS for MariaDB terbaru, Anda akan memiliki data zona waktu terbaru dari RDS. Untuk memastikan bahwa instans basis data Anda memiliki data zona waktu terbaru, sebaiknya mutakhirkan ke versi mesin basis data yang lebih tinggi. Atau, Anda dapat

mengubah secara manual tabel zona waktu dalam instans basis data MariaDB. Untuk itu, Anda dapat menggunakan perintah-perintah SQL atau menjalankan alat mysql_tzinfo_to_sql di klien SQL. Setelah memperbarui data zona waktu secara manual, but ulang instans basis data Anda sehingga perubahan berlaku. RDS tidak mengubah atau mengatur ulang data zona waktu instans basis data yang sedang berjalan. Data zona waktu baru diinstal hanya ketika Anda melakukan pemutakhiran versi mesin basis data.

Anda dapat mengatur zona waktu lokal Anda ke salah satu nilai berikut.

`Africa/Cairo`	`Asia/Riyadh`
`Africa/Casablanca`	`Asia/Seoul`
`Africa/Harare`	`Asia/Shanghai`
`Africa/Monrovia`	`Asia/Singapore`
`Africa/Nairobi`	`Asia/Taipei`
`Africa/Tripoli`	`Asia/Tehran`
`Africa/Windhoek`	`Asia/Tokyo`
`America/Araguaina`	`Asia/Ulaanbaatar`
`America/Asuncion`	`Asia/Vladivostok`
`America/Bogota`	`Asia/Yakutsk`
`America/Buenos_Aires`	`Asia/Yerevan`
`America/Caracas`	`Atlantic/Azores`
`America/Chihuahua`	`Australia/Adelaide`
`America/Cuiaba`	`Australia/Brisbane`
`America/Denver`	`Australia/Darwin`
`America/Fortaleza`	`Australia/Hobart`

America/Guatemala	Australia/Perth
America/Halifax	Australia/Sydney
America/Manaus	Brazil/East
America/Matamoros	Canada/Newfoundland
America/Monterrey	Canada/Saskatchewan
America/Montevideo	Canada/Yukon
America/Phoenix	Europe/Amsterdam
America/Santiago	Europe/Athens
America/Tijuana	Europe/Dublin
Asia/Amman	Europe/Helsinki
Asia/Ashgabat	Europe/Istanbul
Asia/Baghdad	Europe/Kaliningrad
Asia/Baku	Europe/Moscow
Asia/Bangkok	Europe/Paris
Asia/Beirut	Europe/Prague
Asia/Calcutta	Europe/Sarajevo
Asia/Damascus	Pacific/Auckland
Asia/Dhaka	Pacific/Fiji
Asia/Irkutsk	Pacific/Guam
Asia/Jerusalem	Pacific/Honolulu
Asia/Kabul	Pacific/Samoa

Asia/Karachi	US/Alaska
Asia/Kathmandu	US/Central
Asia/Krasnoyarsk	US/Eastern
Asia/Magadan	US/East-Indiana
Asia/Muscat	US/Pacific
Asia/Novosibirsk	UTC

Masalah umum dan batasan untuk RDS for MariaDB

Item berikut adalah masalah dan batasan umum saat menggunakan RDS for MariaDB.

> **Note**
>
> Daftar ini bukanlah daftar lengkap.

Topik

Batas ukuran file MariaDB di Amazon RDS

Untuk instans DB MariaDB, ukuran tabel maksimumnya adalah 16 TB saat menggunakan ruang tabel file-per-table InnoDB. Batasan ini juga membatasi ruang tabel sistem hingga ukuran maksimum sebesar 16 TB. Ruang tabel file-per-table InnoDB (dengan masing-masing tabel dalam ruang tabelnya sendiri) diatur secara default untuk instans DB MariaDB. Batasan ini tidak terkait dengan batas penyimpanan maksimum untuk instans DB MariaDB. Untuk informasi selengkapnya tentang batas penyimpanan, lihat Penyimpanan instans DB Amazon RDS.

Ada kelebihan dan kekurangan dalam menggunakan ruang tabel file-per-table InnoDB, bergantung pada aplikasi Anda. Untuk menentukan pendekatan terbaik bagi aplikasi Anda, lihat Ruang tabel file-per-table dalam dokumentasi MySQL.

Sebaiknya Anda tidak membiarkan tabel berkembang hingga ukuran file maksimum. Secara umum, praktik yang lebih baik adalah membagi data menjadi tabel yang lebih kecil, yang dapat meningkatkan waktu performa dan pemulihan.

Salah satu opsi yang dapat Anda gunakan untuk memecah tabel ke dalam tabel yang lebih kecil adalah partisi. Partisi mendistribusikan porsi tabel besar ke dalam file terpisah berdasarkan aturan yang Anda tentukan. Misalnya, jika menyimpan transaksi berdasarkan tanggal, Anda dapat membuat aturan partisi yang mendistribusikan transaksi lama ke dalam file terpisah menggunakan partisi.

Kemudian, Anda secara berkala dapat mengarsipkan data transaksi historis yang tidak diperlukan aplikasi Anda. Untuk informasi selengkapnya, lihat Partisi dalam dokumentasi MySQL.

Untuk menentukan ukuran semua ruang tabel InnoDB

- Gunakan perintah SQL berikut untuk menentukan apakah salah satu tabel Anda terlalu besar dan merupakan kandidat untuk partisi.

> **Note**
>
> Untuk MariaDB 10.6 dan yang lebih tinggi, kueri ini juga mengembalikan ukuran ruang tabel sistem InnoDB.
> Untuk versi MariaDB sebelum 10.6, Anda tidak dapat menentukan ukuran ruang tabel sistem InnoDB dengan mengkueri tabel sistem. Sebaiknya Anda meningkatkan ke versi yang lebih baru.

```
SELECT SPACE,NAME,ROUND((ALLOCATED_SIZE/1024/1024/1024), 2)
as "Tablespace Size (GB)"
FROM information_schema.INNODB_SYS_TABLESPACES ORDER BY 3 DESC;
```

Untuk menentukan ukuran tabel pengguna non-InnoDB

- Gunakan perintah SQL berikut untuk menentukan apakah ada tabel pengguna non-InnoDB yang terlalu besar.

```
SELECT TABLE_SCHEMA, TABLE_NAME, round(((DATA_LENGTH + INDEX_LENGTH+DATA_FREE)
/ 1024 / 1024/ 1024), 2) As "Approximate size (GB)" FROM information_schema.TABLES
WHERE TABLE_SCHEMA NOT IN ('mysql', 'information_schema', 'performance_schema')
and ENGINE<>'InnoDB';
```

Untuk mengaktifkan ruang tabel file-per-table InnoDB

- Atur parameter `innodb_file_per_table` ke 1 di grup parameter untuk instans DB.

Untuk menonaktifkan ruang tabel file-per-table InnoDB

- Atur parameter `innodb_file_per_table` ke `0` di grup parameter untuk instans DB.

Untuk informasi tentang pembaruan grup parameter, lihat Bekerja dengan grup parameter.

Saat mengaktifkan atau menonaktifkan ruang tabel file-per-table InnoDB, Anda dapat menerbitkan perintah `ALTER TABLE`. Anda dapat menggunakan perintah ini untuk memindahkan tabel dari ruang tabel global ke ruang tabelnya sendiri. Anda juga dapat memindahkan tabel dari ruang tabelnya sendiri ke ruang tabel global. Berikut adalah contohnya.

```
ALTER TABLE table_name ENGINE=InnoDB, ALGORITHM=COPY;
```

Kata yang dicadangkan InnoDB

`InnoDB` adalah kata yang dicadangkan untuk RDS for MariaDB. Anda tidak dapat menggunakan nama ini untuk basis data MariaDB.

Port kustom

Amazon RDS memblokir koneksi ke port kustom 33060 untuk mesin MariaDB. Pilih port yang berbeda untuk mesin MariaDB Anda.

Wawasan Performa

Penghitung InnoDB tidak terlihat di Wawasan Performa untuk RDS for MariaDB versi 10.11 karena komunitas MariaDB tidak lagi mendukungnya.

Amazon RDS for Microsoft SQL Server

Amazon RDS mendukung beberapa versi dan edisi Microsoft SQL Server. Tabel berikut menunjukkan versi minor terbaru yang didukung dari masing-masing versi utama. Untuk daftar lengkap versi, edisi, dan versi mesin RDS yang didukung, lihat Versi Microsoft SQL Server di Amazon RDS.

Versi mayor	Paket Layanan/GDR	Pembaruan kumulatif	Versi minor	Artikel Dasar Pengetahuan	Tanggal rilis
SQL Server 2022	–	CU11	16.0.4105.2	KB5032679	Januari 11, 2024
SQL Server 2019	–	CU25	15.0.4355.3	KB5033688	Februari 15, 2024
SQL Server 2017	GDR	CU31	14.0.3465.1	KB5029376	10 Oktober 2023
SQL Server 2016	SP3 GDR	–	13.0.6435.1	KB5029186	10 Oktober 2023
SQL Server 2014	SP3 GDR	CU4	12.0.6449.1	KB5029185	10 Oktober 2023

Untuk informasi tentang lisensi SQL Server, lihat Melisensikan Microsoft SQL Server di Amazon RDS. Untuk informasi tentang build SQL Server, lihat artikel dukungan Microsoft tentang build SQL Server terbaru.

Dengan Amazon RDS, Anda dapat membuat instans DB dan snapshot DB, point-in-time pemulihan, dan pencadangan otomatis atau manual. Instans DB yang menjalankan SQL Server dapat digunakan di dalam VPC. Anda juga dapat menggunakan Secure Sockets Layer (SSL) untuk menghubungkan ke instans DB yang menjalankan SQL Server, dan Anda dapat menggunakan enkripsi data transparan (TDE) untuk mengenkripsi data diam. Amazon RDS saat ini mendukung deployment

Multi-AZ untuk SQL Server menggunakan SQL Server Database Mirroring (DBM) atau Always On Availability Group (AG) sebagai solusi failover dengan ketersediaan tinggi.

Untuk memberikan pengalaman layanan terkelola, Amazon RDS tidak menyediakan akses shell ke instans DB, dan membatasi akses ke prosedur sistem dan tabel tertentu yang memerlukan hak akses tingkat lanjut. Amazon RDS mendukung akses ke basis data pada instans DB menggunakan aplikasi klien SQL standar seperti Microsoft SQL Server Management Studio. Amazon RDS tidak mengizinkan akses host langsung ke instans DB melalui Telnet, Secure Shell (SSH), atau Windows Remote Desktop Connection. Saat Anda membuat instans DB, pengguna utama ditetapkan ke peran db_owner untuk semua basis data pengguna pada instans tersebut, dan memiliki semua izin tingkat basis data kecuali yang digunakan untuk pencadangan. Amazon RDS mengelola pencadangan untuk Anda.

Sebelum membuat instans DB pertama, Anda harus menyelesaikan langkah-langkah di bagian pengaturan dalam panduan ini. Untuk informasi selengkapnya, lihat Menyiapkan Amazon RDS.

Topik

- Menghubungkan ke instans DB yang menjalankan mesin basis data Microsoft SQL Server
- Menggunakan Active Directory dengan RDS for SQL Server
- Memperbarui aplikasi untuk terhubung ke instans DB Microsoft SQL Server menggunakan sertifikat SSL/TLS baru
- Meng-upgrade mesin DB Microsoft SQL Server
- Mengimpor dan mengekspor basis data SQL Server menggunakan pencadangan dan pemulihan native
- Menggunakan replika baca untuk Microsoft SQL Server di Amazon RDS
- Deployment Multi-AZ untuk Amazon RDS for Microsoft SQL Server
- Fitur tambahan untuk Microsoft SQL di Amazon RDS
- Opsi untuk mesin basis data Microsoft SQL Server
- Tugas DBA umum untuk Microsoft SQL Server

Tugas manajemen umum untuk Microsoft SQL Server di Amazon RDS

Berikut adalah tugas manajemen umum yang Anda lakukan dengan instans DB Amazon RDS for SQL Server, dengan tautan ke dokumentasi yang relevan untuk setiap tugas.

Area tugas	Dokumentasi terkait
Kelas instans, penyimpanan, dan PIOPS Jika membuat instans DB untuk tujuan produksi, Anda harus memahami cara kerja kelas instans, jenis penyimpanan, dan pekerjaan IOPS yang Tersedia di Amazon RDS.	Dukungan kelas instans DB untuk Microsoft SQL Server Jenis penyimpanan Amazon RDS
Deployment Multi-AZ Instans basis data produksi seyogianya menggunakan deployment Multi-AZ. Deployment Multi-AZ memberikan peningkatan ketersediaan, durabilitas data, dan toleransi kesalahan untuk instans DB. Deployment Multi-AZ untuk SQL Server diimplementasikan menggunakan DBM asli SQL Server atau teknologi AG.	Mengonfigurasi dan mengelola deployment Multi-AZ Deployment Multi-AZ menggunakan Microsoft SQL Server Database Mirroring atau Always On Availability Group

Area tugas	Dokumentasi terkait
Amazon Virtual Private Cloud (Amazon VPC) Jika AWS akun Anda memiliki VPC default, maka instans DB Anda secara otomatis dibuat di dalam VPC default. Jika akun Anda tidak memiliki VPC default, dan Anda menginginkan instans DB dalam suatu VPC, Anda harus membuat VPC dan grup subnet sebelum membuat instans DB.	Bekerja dengan klaster DB dalam VPC
Grup keamanan Secara default, instans DB dibuat dengan firewall yang mencegah akses ke instans tersebut. Oleh karena itu, Anda harus membuat grup keamanan dengan alamat IP dan konfigura si jaringan yang benar untuk mengakses instans DB.	Mengontrol akses dengan grup keamanan
Grup parameter Jika instans DB Anda akan membutuhkan parameter basis data tertentu, Anda harus membuat grup parameter sebelum Anda membuat instans DB.	Bekerja dengan grup parameter
Grup opsi Jika instans DB Anda akan membutuhkan opsi basis data tertentu, Anda harus membuat grup opsi sebelum membuat instans DB.	Opsi untuk mesin basis data Microsoft SQL Server
Menghubungkan ke instans DB Setelah membuat grup keamanan dan mengaitkannya ke instans DB, Anda dapat menghubungkan ke instans DB menggunakan aplikasi klien SQL standar seperti Microsoft SQL Server Management Studio.	Menghubungkan ke instans DB yang menjalankan mesin basis data Microsoft SQL Server

Area tugas	Dokumentasi terkait
Pencadangan dan pemulihan Saat membuat instans DB, Anda dapat mengonfigurasinya untuk melakukan pencadangan otomatis. Anda juga dapat mencadangkan dan memulihkan basis data secara manual menggunakan file cadangan penuh (file .bak).	Pengantar cadangan Mengimpor dan mengekspor basis data SQL Server menggunakan pencadangan dan pemulihan native
Pemantauan Anda dapat memantau instans SQL Server DB dengan menggunakan metrik CloudWatch Amazon RDS, peristiwa, dan pemantauan yang disempurnakan.	Melihat metrik di konsol Amazon RDS Melihat peristiwa Amazon RDS
File log Anda dapat mengakses file log untuk instans DB SQL Server Anda.	Memantau file log Amazon RDS File log basis data Microsoft SQL Server

Ada juga tugas administratif lanjutan untuk bekerja dengan instans DB SQL Server. Untuk informasi selengkapnya, lihat dokumentasi berikut ini:

- Tugas DBA umum untuk Microsoft SQL Server.
- Menggunakan AWS Managed Active Directory dengan RDS for SQL Server
- Mengakses basis data tempdb

Batasan untuk instans DB Microsoft SQL Server

Implementasi Amazon RDS dari Microsoft SQL Server pada instans DB memiliki beberapa batasan yang harus Anda ketahui:

- Jumlah maksimum basis data yang didukung pada instans DB bergantung pada jenis kelas instans dan mode ketersediaan—Single-AZ, Multi-AZ Database Mirroring (DBM), atau Multi-AZ Availability Group (AG). Basis data sistem Microsoft SQL Server tidak termasuk dalam batasan ini.

Tabel berikut menunjukkan jumlah maksimum basis data yang didukung untuk setiap jenis kelas instans dan mode ketersediaan. Gunakan tabel ini untuk membantu memutuskan apakah Anda dapat beralih dari satu jenis kelas instans ke kelas instans lainnya, atau dari satu mode ketersediaan ke mode lainnya. Jika instans DB sumber Anda memiliki lebih banyak basis data daripada yang dapat didukung oleh jenis kelas instans target atau mode ketersediaan, modifikasi instans DB akan gagal. Status permintaan Anda dapat dilihat di panel Peristiwa.

Jenis kelas instans	AZ Tunggal	Multi-AZ dengan DBM	Multi-AZ dengan Always On AG
db.*.micro to db.*.medium	30	N/A	N/A
db.*.large	30	30	30
db.*.xlarge to db.*.16xlarge	100	50	75
db.*.24xlarge	100	50	100

* Menunjukkan jenis kelas instans yang berbeda.

Misalnya, instans DB Anda berjalan pada db.*.16xlarge dengan AZ Tunggal dan memiliki 76 basis data. Anda mengubah instans DB untuk meningkatkan ke penggunaan Multi-AZ Always On AG. Peningkatan ini gagal, karena instans DB berisi lebih banyak basis data daripada yang dapat didukung oleh konfigurasi target Anda. Jika Anda meningkatkan jenis kelas instans Anda ke db.*.24xlarge, modifikasi tersebut akan berhasil.

Jika peningkatan gagal, Anda melihat peristiwa dan pesan yang mirip dengan berikut ini:

- Tidak dapat mengubah kelas instans basis data. Instans memiliki 76 basis data, tetapi setelah konversi, hanya akan mendukung 75.
- Tidak dapat mengonversi instans DB menjadi Multi-AZ: instans memiliki 76 basis data, tetapi setelah konversi, hanya akan mendukung 75.

Jika point-in-time pemulihan atau pemulihan snapshot gagal, Anda melihat peristiwa dan pesan yang mirip dengan berikut ini:

  - Instans basi data dimasukkan ke dalam pemulihan yang tidak kompatibel. Instans memiliki 76 basis data, tetapi setelah konversi, hanya akan mendukung 75.
- Port berikut disimpan untuk Amazon RDS, dan Anda tidak dapat menggunakannya saat membuat instans DB: `1234, 1434, 3260, 3343, 3389, 47001,` dan `49152-49156`.
- Koneksi klien dari alamat IP di dalam rentang 169.254.0.0/16 tidak diizinkan. Rentang ini adalah Automatic Private IP Addressing Range (APIPA) yang digunakan untuk alamat tautan lokal.
- SQL Server Standard Edition hanya menggunakan subset dari prosesor yang tersedia jika jumlah prosesor instans DB melebihi batas perangkat lunak (24 core, 4 soket, dan RAM 128 GB). Contohnya adalah kelas instans db.m5.24xlarge dan db.r5.24xlarge.

  Untuk informasi selengkapnya, lihat tabel batas skala pada Editions and supported features of SQL Server 2019 (15.x) dalam dokumentasi Microsoft.
- Amazon RDS for SQL Server tidak mendukung impor data ke basis data msdb.
- Anda tidak dapat mengganti nama basis data pada instans DB dalam deployment Multi-AZ SQL Server.
- Pastikan Anda menggunakan pedoman ini saat menetapkan parameter DB berikut pada RDS for SQL Server:
  - `max server memory (mb)` >= 256 MB
  - `max worker threads` >= (jumlah CPU logis * 7)

  Untuk informasi selengkapnya tentang cara menetapkan parameter DB, lihat Bekerja dengan grup parameter.
- Ukuran penyimpanan maksimum untuk instans DB SQL Server adalah sebagai berikut:
  - Penyimpanan Tujuan Umum (SSD) – 16 TiB untuk semua edisi
  - Penyimpanan IOPS yang Tersedia – 16 TiB untuk semua edisi
  - Penyimpanan Magnetik – 1 TiB untuk semua edisi

  Jika Anda memiliki skenario yang membutuhkan jumlah penyimpanan yang lebih besar, Anda dapat menggunakan sharding di beberapa instans DB untuk mengatasi batas tersebut. Pendekatan ini membutuhkan logika perutean yang bergantung pada data dalam aplikasi yang terhubung ke sistem sharding. Anda dapat menggunakan kerangka kerja sharding yang ada, atau Anda dapat menulis kode kustom untuk mengaktifkan sharding. Jika Anda menggunakan kerangka kerja yang ada, kerangka kerja tersebut tidak dapat menginstal komponen apa pun di server yang sama dengan instans DB.
- Ukuran penyimpanan minimum untuk instans DB SQL Server adalah sebagai berikut:

  - Penyimpanan Tujuan Umum (SSD) – 20 GiB untuk Edisi Enterprise, Standard, Web, dan Express
  - Penyimpanan IOPS yang Tersedia – 20 GiB untuk Edisi Enterprise Standar, dan Web dan Express
  - Penyimpanan Magnetik – 20 GiB untuk Edisi Enterprise Standar, dan Web dan Express
- Amazon RDS tidak mendukung menjalankan layanan ini di server yang sama dengan instans DB RDS Anda:
  - Layanan Kualitas Data
  - Layanan Data Master

  Untuk menggunakan fitur ini, sebaiknya Anda menginstal SQL Server pada instans Amazon EC2, atau menggunakan instans SQL Server on-premise. Dalam kasus ini, instans EC2 atau SQL Server bertindak sebagai server Layanan Data Master untuk instans DB SQL Server Anda di Amazon RDS. Anda dapat menginstal SQL Server pada instans Amazon EC2 dengan penyimpanan Amazon EBS, sesuai dengan kebijakan lisensi Microsoft.
- Karena batasan di Microsoft SQL Server, pemulihan ke suatu titik waktu sebelum berhasil menjalankan `DROP DATABASE` mungkin tidak mencerminkan status basis data tersebut pada saat itu. Misalnya, basis data yang dihapus biasanya dikembalikan ke statusnya hingga 5 menit sebelum perintah `DROP DATABASE` dikeluarkan. Jenis pemulihan ini berarti Anda tidak dapat memulihkan transaksi yang dilakukan selama beberapa menit tersebut pada basis data yang dihapus. Untuk menyiasatinya, Anda dapat mengeluarkan kembali perintah `DROP DATABASE` setelah operasi pemulihan selesai. Menghentikan basis data akan menghapus log transaksi untuk basis data tersebut.
- Untuk SQL Server, Anda membuat basis data setelah membuat instans DB. Nama basis data mengikuti aturan penamaan SQL Server biasa dengan perbedaan berikut:
  - Nama basis data tidak boleh dimulai dengan `rdsadmin`.
  - Nama tersebut tidak boleh diawali atau diakhiri dengan spasi atau tab.
  - Nama tersebut tidak boleh berisi karakter apa pun yang membuat baris baru.
  - Nama tersebut tidak boleh berisi kutipan tunggal (`'`).

Dukungan kelas instans DB untuk Microsoft SQL Server

Komputasi dan kapasitas memori instans DB ditentukan oleh kelas instans DB-nya. Kelas instans DB yang Anda butuhkan tergantung pada kebutuhan daya dan memori pemrosesan Anda. Untuk informasi selengkapnya, lihat Kelas instans DB .

Daftar kelas instans DB berikut yang didukung untuk Microsoft SQL Server disediakan di sini untuk memudahkan Anda. Untuk daftar terbaru, lihat konsol RDS: https://console.aws.amazon.com/rds/.

Tidak semua kelas instans DB tersedia di semua versi minor SQL Server yang didukung. Misalnya, beberapa kelas instans DB yang lebih baru seperti db.r6i tidak tersedia pada versi minor yang lebih lama. Anda dapat menggunakan AWS CLI perintah describe-orderable-db-instance-options untuk mengetahui kelas instans DB mana yang tersedia untuk edisi dan versi SQL Server Anda.

Edisi SQL Server	Rentang dukungan 2022 dan 2019	Rentang dukungan 2017 dan 2016	Rentang dukungan 2014
Enterprise Edition	db.t3.xlarge –db.t3.2xlarge db.r5.xlarge –db.r5.24xlarge db.r5b.xlarge –db.r5b.24xlarge db.r5d.xlarge –db.r5d.24xlarge db.r6i.xlarge –db.r6i.32xlarge db.m5.xlarge –db.m5.24xlarge	db.t3.xlarge –db.t3.2xlarge db.r3.xlarge –db.r3.8xlarge db.r4.xlarge –db.r4.16xlarge db.r5.xlarge –db.r5.24xlarge db.r5b.xlarge –db.r5b.24xlarge db.r5d.xlarge –db.r5d.24xlarge	db.t3.xlarge –db.t3.2xlarge db.r3.xlarge –db.r3.8xlarge db.r4.xlarge –db.r4.8xlarge db.r5.xlarge –db.r5.24xlarge db.r5b.xlarge –db.r5b.24xlarge db.r5d.xlarge –db.r5d.24xlarge

Edisi SQL Server	Rentang dukungan 2022 dan 2019	Rentang dukungan 2017 dan 2016	Rentang dukungan 2014
	db.m5d.xlarge –db.m5d.24xlarge db.m6i.xlarge –db.m6i.32xlarge db.x1.16xlarge –db.x1.32xlarge db.x1e.xlarge –db.x1e.32xlarge db.x2iedn.xlarge –db.x2iedn.32xlarge db.z1d.xlarge –db.z1d.12xlarge	db.r6i.xlarge –db.r6i.32xlarge db.m4.xlarge –db.m4.16xlarge db.m5.xlarge –db.m5.24xlarge db.m5d.xlarge –db.m5d.24xlarge db.m6i.xlarge –db.m6i.32xlarge db.x1.16xlarge –db.x1.32xlarge db.x1e.xlarge –db.x1e.32xlarge db.x2iedn.xlarge –db.x2iedn.32xlarge db.z1d.xlarge –db.z1d.12xlarge	db.r6i.xlarge –db.r6i.32xlarge db.m4.xlarge –db.m4.10xlarge db.m5.xlarge –db.m5.24xlarge db.m5d.xlarge –db.m5d.24xlarge db.m6i.xlarge –db.m6i.32xlarge db.x1.16xlarge –db.x1.32xlarge db.x1e.xlarge –db.x1e.32xlarge db.x2iedn.xlarge –db.x2iedn.32xlarge

Edisi SQL Server	Rentang dukungan 2022 dan 2019	Rentang dukungan 2017 dan 2016	Rentang dukungan 2014
Standarc Edition	db.t3.xla rge –db.t3.2xlarge db.r5.lar ge –db.r5.24xlarge db.r5b.la rge –db.r5b.24 xlarge db.r5d.la rge –db.r5d.24 xlarge db.r6i.la rge –db.r6i.8xlarge db.m5.lar ge –db.m5.24xlarge db.m5d.la rge –db.m5d.24 xlarge db.m6i.la rge –db.m6i.8xlarge db.x1.16x large –db.x1.32x large db.x1e.xl arge –db.x1e.32 xlarge	db.t3.xla rge –db.t3.2xlarge db.r4.lar ge –db.r4.16xlarge db.r5.lar ge –db.r5.24xlarge db.r5b.la rge –db.r5b.24 xlarge db.r5d.la rge –db.r5d.24 xlarge db.r6i.la rge –db.r6i.8xlarge db.m4.lar ge –db.m4.16xlarge db.m5.lar ge –db.m5.24xlarge db.m5d.la rge –db.m5d.24 xlarge db.m6i.la rge –db.m6i.8xlarge db.x1.16x large –db.x1.32x large	db.t3.xla rge –db.t3.2xlarge db.r3.lar ge –db.r3.8xlarge db.r4.lar ge –db.r4.8xlarge db.r5.lar ge –db.r5.24xlarge db.r5b.la rge –db.r5b.24 xlarge db.r5d.la rge –db.r5d.24 xlarge db.r6i.la rge –db.r6i.8xlarge db.m3.med ium –db.m3.2xlarge db.m4.lar ge –db.m4.10xlarge db.m5.lar ge –db.m5.24xlarge db.m5d.la rge –db.m5d.24 xlarge

Edisi SQL Server	Rentang dukungan 2022 dan 2019	Rentang dukungan 2017 dan 2016	Rentang dukungan 2014
	db.x2iedn .xlarge – db.x2iedn .32xlarge db.z1d.la rge – db.z1d.12 xlarge	db.x1e.xl arge – db.x1e.32 xlarge db.x2iedn .xlarge – db.x2iedn .32xlarge db.z1d.la rge – db.z1d.12 xlarge	db.m6i.la rge – db.m6i.8xlarge db.x1.16x large – db.x1.32x large db.x1e.xl arge – db.x1e.32 xlarge db.x2iedn .xlarge – db.x2iedn .32xlarge

Edisi SQL Server	Rentang dukungan 2022 dan 2019	Rentang dukungan 2017 dan 2016	Rentang dukungan 2014
Web Edition	db.t3.small –db.t3.2xlarge db.r5.large –db.r5.4xlarge db.r5b.large –db.r5b.4xlarge db.r5d.large –db.r5d.4xlarge db.r6i.large –db.r6i.4xlarge db.m5.large –db.m5.4xlarge db.m5d.large –db.m5d.4xlarge db.m6i.large –db.m6i.4xlarge db.z1d.large –db.z1d.3xlarge	db.t2.small –db.t2.medium db.t3.small –db.t3.2xlarge db.r4.large –db.r4.2xlarge db.r5.large –db.r5.4xlarge db.r5b.large –db.r5b.4xlarge db.r5d.large –db.r5d.4xlarge db.r6i.large –db.r6i.4xlarge db.m4.large –db.m4.4xlarge db.m5.large –db.m5.4xlarge db.m5d.large –db.m5d.4xlarge db.m6i.large –db.m6i.4xlarge db.z1d.large –db.z1d.3xlarge	db.t2.small –db.t2.medium db.t3.small –db.t3.2xlarge db.r3.large –db.r3.2xlarge db.r4.large –db.r4.2xlarge db.r5.large –db.r5.4xlarge db.r5b.large –db.r5b.4xlarge db.r5d.large –db.r5d.4xlarge db.r6i.large –db.r6i.4xlarge db.m3.medium –db.m3.2xlarge db.m4.large –db.m4.4xlarge db.m5.large –db.m5.4xlarge db.m5d.large –db.m5d.4xlarge

Edisi SQL Server	Rentang dukungan 2022 dan 2019	Rentang dukungan 2017 dan 2016	Rentang dukungan 2014
			db.m6i.la rge –db.m6i.4xlarge
Express Edition	db.t3.mic ro –db.t3.xlarge	db.t2.mic ro –db.t2.medium db.t3.mic ro –db.t3.xlarge	db.t2.mic ro –db.t2.medium db.t3.mic ro –db.t3.xlarge

Keamanan Microsoft SQL Server

Mesin basis data Microsoft SQL Server menggunakan keamanan berbasis peran. Nama pengguna utama yang Anda tentukan saat membuat instans DB adalah login SQL Server Authentication yang merupakan bagian dari peran server tetap `processadmin`, `public`, dan `setupadmin`.

Setiap pengguna yang membuat basis data ditetapkan ke peran db_owner untuk basis data tersebut dan memiliki semua izin tingkat basis data kecuali yang digunakan untuk pencadangan. Amazon RDS mengelola pencadangan untuk Anda.

Peran tingkat server berikut saat ini tidak tersedia di Amazon RDS for SQL Server:

- bulkadmin
- dbcreator
- diskadmin
- securityadmin
- serveradmin
- sysadmin

Izin tingkat server berikut tidak tersedia di instans DB RDS for SQL Server:

- MENGUBAH BASIS DATA
- MENGUBAH PEMBERITAHUAN PERISTIWA

- MENGUBAH SUMBER DAYA
- MENGUBAH PENGATURAN (Anda dapat menggunakan operasi API grup parameter DB untuk mengubah parameter; untuk informasi selengkapnya, lihat Bekerja dengan grup parameter)
- AUTENTIKASI SERVER
- CONTROL_SERVER
- MEMBUAT PEMBERITAHUAN PERISTIWA DDL
- MEMBUAT TITIK AKHIR
- MEMBUAT PERAN SERVER
- MEMBUAT PEMBERITAHUAN PERISTIWA LACAK
- MEMBATALKAN BASIS DATA
- UNIT AKSES EKSTERNAL
- PEMATIAN (Anda dapat menggunakan opsi boot ulang RDS)
- ASSEMBLY TIDAK AMAN
- MENGUBAH GRUP KETERSEDIAAN
- MEMBUAT GRUP KETERSEDIAAN

Dukungan program kepatuhan untuk instans DB Microsoft SQL Server

AWS Layanan dalam lingkup telah sepenuhnya dinilai oleh auditor pihak ketiga dan menghasilkan sertifikasi, pengesahan kepatuhan, atau Otoritas untuk Beroperasi (ATO). Untuk informasi selengkapnya, lihat layanan AWS dalam cakupan berdasarkan program kepatuhan.

Dukungan HIPAA untuk instans DB Microsoft SQL Server

Anda dapat menggunakan basis data Amazon RDS for Microsoft SQL Server untuk membangun aplikasi yang mematuhi HIPAA. Anda dapat menyimpan informasi terkait perawatan kesehatan, termasuk informasi kesehatan yang dilindungi (PHI), berdasarkan Perjanjian Rekan Bisnis (BAA) dengan AWS. Untuk informasi selengkapnya, lihat Kepatuhan HIPAA.

Amazon RDS for SQL Server mendukung HIPAA untuk versi dan edisi berikut:

- SQL Server 2022 Edisi Enterprise, Standard, dan Web
- SQL Server 2019 Edisi Enterprise, Standard, dan Web

- SQL Server 2017 Edisi Enterprise, Standard, dan Web
- SQL Server 2016 Edisi Enterprise, Standard, dan Web
- SQL Server 2014 Edisi Enterprise, Standard, dan Web

Untuk mengaktifkan dukungan HIPAA pada instans DB Anda, siapkan tiga komponen berikut.

Komponen	Detail
Audit	Untuk menyiapkan audit, atur parameter `rds.sqlserver_audit` ke nilai `fedramp_hipaa` . Jika instans DB Anda belum menggunakan grup parameter DB kustom, Anda harus membuat grup parameter kustom dan melampirkannya ke instans DB sebelum dapat mengubah parameter `rds.sqlserver_audit` . Untuk informasi selengkapnya, lihat Bekerja dengan grup parameter.
Enkripsi transportasi	Untuk menyiapkan enkripsi transportasi, paksa semua koneksi ke instans DB Anda untuk menggunakan Secure Sockets Layer (SSL). Untuk informasi selengkapnya, lihat Memaksa koneksi ke instans DB Anda untuk menggunakan SSL.
Enkripsi saat istirahat	Untuk menyiapkan enkripsi saat diam, Anda memiliki dua opsi: 1. Jika menjalankan SQL Server 2014–2022 Enterprise Edition atau 2022 Standard Edition, Anda dapat menggunakan Enkripsi Data Transparan (TDE) untuk menghasilkan enkripsi diam. Untuk informasi selengkapnya, lihat Dukungan untuk Enkripsi Data Transparan di SQL Server. 2. Anda dapat mengatur enkripsi saat istirahat dengan menggunakan AWS Key Management Service (AWS KMS) kunci enkripsi. Untuk informasi selengkapnya, lihat Mengenkripsi sumber daya Amazon RDS.

Dukungan SSL untuk instans DB Microsoft SQL Server

Anda dapat menggunakan SSL untuk mengenkripsi koneksi antara aplikasi Anda dan instans DB Amazon RDS yang menjalankan Microsoft SQL Server. Anda juga dapat memaksa semua koneksi ke instans DB Anda untuk menggunakan SSL. Jika Anda memaksa koneksi untuk menggunakan SSL, hal ini terjadi secara transparan pada klien, dan klien tidak perlu melakukan tindakan apa pun untuk menggunakan SSL.

SSL didukung di semua AWS Wilayah dan untuk semua edisi SQL Server yang didukung. Untuk informasi selengkapnya, lihat Menggunakan SSL dengan instans DB Microsoft SQL Server.

Versi Microsoft SQL Server di Amazon RDS

Anda dapat menentukan versi Microsoft SQL Server yang didukung saat ini ketika membuat instans DB baru. Anda dapat menentukan versi utama Microsoft SQL Server (seperti Microsoft SQL Server 14.00), dan versi minor yang didukung untuk versi utama yang ditentukan. Jika tidak ada versi yang ditentukan, Amazon RDS menetapkan ke versi yang didukung secara default, biasanya versi terbaru. Jika versi utama ditentukan tetapi versi kecil tidak, Amazon RDS menjadikan menetapkan default ke rilis versi utama terbaru yang telah Anda tentukan.

Tabel berikut menunjukkan versi yang didukung untuk semua edisi dan semua AWS Wilayah, kecuali jika disebutkan. Anda juga dapat menggunakan `describe-db-engine-versions` AWS CLI perintah untuk melihat daftar versi yang didukung, serta default untuk instans DB yang baru dibuat.

Versi SQL Server yang didukung di RDS

Versi utama	Versi minor	API RDS **`EngineVersion`** dan CLI **`engine-version`**
SQL Server 2022	16.00.4095.4 (CU10)	`16.00.4095.4.v1`
	16.00.4085.2 (CU9)	`16.00.4085.2.v1`
SQL Server 2019	15.00.4345.5 (CU24)	`15.00.4345.5.v1`
	15.00.4335.1 (CU23)	`15.00.4335.1.v1`
	15.00.4322.2 (CU22)	`15.00.4322.2.v1`
	15.00.4316.3 (CU21)	`15.00.4316.3.v1`

Versi utama	Versi minor	API RDS **EngineVersion** dan CLI **engine-version**
	15.00.4312.2 (CU20)	15.00.4312.2.v1
	15.00.4236.7 (CU16)	15.00.4236.7.v1
	15.00.4198.2 (CU15)	15.00.4198.2.v1
	15.00.4153.1 (CU12)	15.00.4153.1.v1
	15.00.4073.23 (CU8)	15.00.4073.23.v1
	15.00.4043.16 (CU5)	15.00.4043.16.v1
SQL Server 2017	14.00.3465.1 (CU31)	14.00.3465.1.v1
	14.00.3460.9 (CU31)	14.00.3460.9.v1
	14.00.3451.2 (CU30)	14.00.3451.2.v1
	14.00.3421.10 (CU27)	14.00.3421.10.v1
	14.00.3401.7 (CU25)	14.00.3401.7.v1
	14.00.3381.3 (CU23)	14.00.3381.3.v1
	14.00.3356.20 (CU22)	14.00.3356.20.v1
	14.00.3294.2 (CU20)	14.00.3294.2.v1
	14.00.3281.6 (CU19)	14.00.3281.6.v1
SQL Server 2016	13.00.6435.1 (GDR)	13.00.6435.1.v1
	13.00.6430.49 (GDR)	13.00.6430.49.v1
	13.00.6419.1 (SP3 + Perbaikan Terbaru)	13.00.6419.1.v1
	13.00.6300.2 (SP3)	13.00.6300.2.v1

Versi utama	Versi minor	API RDS **EngineVersion** dan CLI **engine-version**
SQL Server 2014	12.00.6449.1 (SP3 CU4 GDR)	12.00.6449.1.v1
	12.00.6444.4 (SP3 CU4 GDR)	12.00.6444.4.v1
	12.00.6439.10 (SP3 CU4 SU)	12.00.6439.10.v1
	12.00.6433.1 (SP3 CU4 SU)	12.00.6433.1.v1
	12.00.6329.1 (SP3 CU4)	12.00.6329.1.v1
	12.00.6293.0 (SP3 CU3)	12.00.6293.0.v1

Manajemen versi di Amazon RDS

Amazon RDS mencakup manajemen versi fleksibel yang memungkinkan Anda mengontrol kapan dan bagaimana patching dan tingkatkan instans DB Anda. Ini memungkinkan Anda untuk melakukan tindakan berikut terhadap mesin DB Anda:

- Menjaga kompatibilitas dengan versi patch mesin basis data.
- Uji versi patch baru untuk memverifikasi bahwa versi-versi tersebut berfungsi dengan aplikasi Anda sebelum menerapkannya dalam produksi.
- Rencanakan dan lakukan peningkatan versi untuk memenuhi perjanjian tingkat layanan dan persyaratan penentuan waktu Anda.

Patching mesin Microsoft SQL Server di Amazon RDS

Amazon RDS secara berkala menggabungkan patch basis data Microsoft SQL Server resmi ke dalam versi mesin instans DB yang khusus untuk Amazon RDS. Untuk informasi selengkapnya tentang patch Microsoft SQL Server di setiap versi mesin, lihat Versi dan dukungan fitur di Amazon RDS.

Saat ini, Anda melakukan semua peningkatan mesin pada instans DB Anda secara manual. Untuk informasi selengkapnya, lihat Meng-upgrade mesin DB Microsoft SQL Server.

Jadwal penghentian untuk versi mesin utama Microsoft SQL Server di Amazon RDS

Tabel berikut menampilkan jadwal penghentian yang direncanakan untuk versi mesin utama Microsoft SQL Server.

Tanggal	Informasi
9 Juli 2024	Microsoft akan menghentikan pembaruan patch penting untuk SQL Server 2014. Unt nya, lihat Microsoft SQL Server 2014 di dokumentasi Microsoft.
1 Juni 2024	Amazon RDS berencana untuk mengakhiri dukungan Microsoft SQL Server 2014 pad Pada saat itu, instans yang masih tersisa akan dijadwalkan untuk dimigrasikan ke SQ versi minor terbaru). Untuk informasi selengkapnya, lihat Pengumuman: Amazon RD mengakhiri dukungan untuk versi utama SQL Server 2014. Untuk menghindari peningkatan otomatis dari Microsoft SQL Server 2014, Anda dapa an pada waktu yang diinginkan. Untuk informasi selengkapnya, lihat Meng-upgrade v
12 Juli 2022	Microsoft akan menghentikan pembaruan patch penting untuk SQL Server 2012. Unt nya, lihat Microsoft SQL Server 2012 di dokumentasi Microsoft.
1 Juni 2022	Amazon RDS berencana untuk mengakhiri dukungan Microsoft SQL Server 2012 pad Pada saat itu, instans yang masih tersisa akan dijadwalkan untuk dimigrasikan ke SQ versi minor terbaru). Untuk informasi selengkapnya, lihat Pengumuman: Amazon RD mengakhiri dukungan untuk versi utama SQL Server 2012. Untuk menghindari peningkatan otomatis dari Microsoft SQL Server 2012, Anda dapa an pada waktu yang diinginkan. Untuk informasi selengkapnya, lihat Meng-upgrade v
1 September 2021	Amazon RDS mulai menonaktifkan pembuatan instans DB RDS for SQL Server baru Microsoft SQL Server 2012. Untuk informasi selengkapnya, lihat Pengumuman: Ama Server mengakhiri dukungan untuk versi utama SQL Server 2012.
12 Juli 2019	Tim Amazon RDS menghentikan dukungan untuk Microsoft SQL Server 2008 R2 pad yang masih tersisa dari Microsoft SQL Server 2008 R2 dimigrasikan ke SQL Server 2 minor terbaru).

Tanggal	Informasi
	Untuk menghindari peningkatan otomatis dari Microsoft SQL Server 2008 R2, Anda c peningkatan pada waktu yang diinginkan. Untuk informasi selengkapnya, lihat Meng- instans DB.
25 April 2019	Sebelum akhir April 2019, Anda tidak lagi dapat membuat basis data Amazon RDS fo menggunakan Microsoft SQL Server 2008R2.

Fitur Microsoft SQL Server di Amazon RDS

Versi SQL Server yang didukung di Amazon RDS dilengkapi fitur-fitur berikut. Secara umum, versi juga dilengkapi dengan fitur dari versi sebelumnya, kecuali dinyatakan lain dalam dokumentasi Microsoft.

Topik

- Fitur Microsoft SQL Server 2022
- Fitur Microsoft SQL Server 2019
- Fitur Microsoft SQL Server 2017
- Fitur Microsoft SQL Server 2016
- Fitur Microsoft SQL Server 2014
- Akhir dukungan Microsoft SQL Server 2012 di Amazon RDS
- Akhir dukungan Microsoft SQL Server 2008 R2 di Amazon RDS

Fitur Microsoft SQL Server 2022

SQL Server 2022 dilengkapi dengan banyak fitur baru, seperti berikut ini:

- Parameter Sensitive Plan Optimization – memungkinkan beberapa paket cache untuk satu pernyataan berparameter, berpotensi mengurangi masalah dengan sniffing parameter.
- SQL Server Ledger – menyediakan kemampuan untuk membuktikan secara kriptografi bahwa data Anda belum diubah tanpa otorisasi.
- Inisialisasi file instan untuk peristiwa pertumbuhan file log transaksi – menghasilkan eksekusi peristiwa pertumbuhan log yang lebih cepat hingga 64MB, termasuk untuk basis data dengan TDE yang diaktifkan.

- Penyempurnaan konkurensi latch halaman sistem – mengurangi pertentangan latch halaman saat mengalokasikan dan membatalkan alokasi halaman dan luasan data, memberikan peningkatan performa yang signifikan pada `tempdb` beban kerja yang berat.

Untuk daftar lengkap fitur SQL Server 2022, lihat What's new in SQL Server 2022 (16.x) di dokumentasi Microsoft.

Untuk daftar fitur yang tidak didukung, lihat Fitur yang tidak didukung dan fitur dengan dukungan terbatas.

Fitur Microsoft SQL Server 2019

SQL Server 2019 dilengkapi dengan banyak fitur baru, seperti berikut ini:

- Pemulihan basis data yang dipercepat (ADR) – Mengurangi waktu pemulihan crash setelah mulai ulang atau rollback transaksi yang berjalan lama.
- Intelligent Query Processing (IQP):
  - Umpan balik pemberian memori mode baris – Mengoreksi pemberian hak akses eksesif secara otomatis, yang sebaliknya akan mengakibatkan pemborosan memori dan pengurangan konkurensi.
  - Mode batch di rowstore – Mengaktifkan eksekusi mode batch untuk beban kerja analitik tanpa memerlukan indeks columnstore.
  - Kompilasi ditangguhkan variabel tabel – Menyempurnakan kualitas paket dan performa keseluruhan untuk kueri yang mereferensikan variabel tabel.
- Performa cerdas:
  - Opsi indeks `OPTIMIZE_FOR_SEQUENTIAL_KEY` – Meningkatkan throughput untuk sisipan konkurensi tinggi ke dalam indeks.
  - Peningkatan skalabilitas titik pemeriksaan tidak langsung – Membantu basis data dengan beban kerja DML berat.
  - Pembaruan Concurrent Page Free Space (PFS) – Memungkinkan penanganan sebagai latch bersama, bukan sebagai latch eksklusif.
- Memantau perbaikan:
  - Jenis tunggu `WAIT_ON_SYNC_STATISTICS_REFRESH` – Menampilkan akumulasi waktu tingkat instans yang dihabiskan untuk operasi refresh statistik sinkron.

  - Konfigurasi cakupan basis data – Mencakup `LIGHTWEIGHT_QUERY_PROFILING` dan `LAST_QUERY_PLAN_STATS`.
  - Fungsi manajemen dinamis (DMF) – Mencakup `sys.dm_exec_query_plan_stats` dan `sys.dm_db_page_info`.
- Peringatan pemotongan panjang – Pesan kesalahan pemotongan data secara default mencakup nama tabel dan kolom serta nilai yang dipotong.
- Pembuatan indeks online yang dapat dilanjutkan kembali – Di SQL Server 2017, hanya pembuatan ulang indeks online yang dapat dilanjutkan kembali yang didukung.

Untuk daftar lengkap fitur SQL Server 2019, lihat What's new in SQL Server 2019 (15.x) di dokumentasi Microsoft.

Untuk daftar fitur yang tidak didukung, lihat Fitur yang tidak didukung dan fitur dengan dukungan terbatas.

Fitur Microsoft SQL Server 2017

SQL Server 2017 dilengkapi dengan banyak fitur baru, seperti berikut ini:

- Pemrosesan kueri adaptif
- Koreksi paket otomatis (fitur penyetelan otomatis)
- GraphDB
- Pembuatan ulang indeks yang dapat dilanjutkan

Untuk daftar lengkap fitur SQL Server 2017, lihat What's new in SQL Server 2017 di dokumentasi Microsoft.

Untuk daftar fitur yang tidak didukung, lihat Fitur yang tidak didukung dan fitur dengan dukungan terbatas.

Fitur Microsoft SQL Server 2016

Amazon RDS mendukung fitur SQL Server 2016 berikut:

- Selalu Dienkripsi
- Dukungan JSON
- Analitik Operasional

- Penyimpanan Kueri
- Tabel Temporal

Untuk daftar lengkap fitur SQL Server 2016, lihat What's new in SQL Server 2016 di dokumentasi Microsoft.

Fitur Microsoft SQL Server 2014

Selain fitur SQL Server 2012 yang didukung, Amazon RDS mendukung pengoptimal kueri baru yang tersedia di SQL Server 2014, dan juga fitur durabilitas tertunda.

Untuk daftar fitur yang tidak didukung, lihat Fitur yang tidak didukung dan fitur dengan dukungan terbatas.

SQL Server 2014 mendukung semua parameter dari SQL Server 2012 dan menggunakan nilai default yang sama. SQL Server 2014 mencakup satu parameter baru, checksum cadangan default. Untuk informasi selengkapnya, lihat How to enable the CHECKSUM option if backup utilities do not expose the option dalam dokumentasi Microsoft.

Akhir dukungan Microsoft SQL Server 2012 di Amazon RDS

SQL Server 2012 telah mencapai akhir dukungannya di Amazon RDS.

RDS meningkatkan semua instans DB yang masih menggunakan SQL Server 2012 ke versi minor terbaru SQL Server 2014. Untuk informasi selengkapnya, lihat Manajemen versi di Amazon RDS.

Akhir dukungan Microsoft SQL Server 2008 R2 di Amazon RDS

SQL Server 2008 R2 telah mencapai akhir dukungannya di Amazon RDS.

RDS meningkatkan semua instans DB yang masih menggunakan SQL Server 2008 R2 ke versi minor terbaru SQL Server 2012. Untuk informasi selengkapnya, lihat Manajemen versi di Amazon RDS.

Dukungan Change Data Capture (CDC) untuk instans DB Microsoft SQL Server

Amazon RDS mendukung Change Data Capture (CDC) untuk instans DB yang menjalankan Microsoft SQL Server. CDC menangkap perubahan yang dibuat pada data di tabel Anda, dan

menyimpan metadata tentang setiap perubahan yang bisa Anda akses nanti. Untuk informasi selengkapnya, lihat Change Data Capture (CDC) di dokumentasi Microsoft.

Amazon RDS mendukung CDC untuk edisi dan versi SQL Server berikut:

- Microsoft SQL Server Enterprise Edition (Semua versi)
- Microsoft SQL Server Standard Edition:
  - 2022
  - 2019
  - 2017
  - 2016 versi 13.00.4422.0 SP1 CU2 dan yang lebih baru

Untuk menggunakan CDC dengan instans DB Amazon RDS Anda, pertama-tama aktifkan atau nonaktifkan CDC di tingkat basis data dengan menggunakan prosedur tersimpan yang ditentukan RDS. Setelah itu, setiap pengguna yang memiliki peran `db_owner` untuk basis data tersebut dapat menggunakan prosedur tersimpan Microsoft native untuk mengontrol CDC pada basis data tersebut. Untuk informasi selengkapnya, lihat Menggunakan pengambilan data perubahan.

Anda dapat menggunakan CDC dan AWS Database Migration Service untuk mengaktifkan replikasi berkelanjutan dari instans SQL Server DB.

Fitur yang tidak didukung dan fitur dengan dukungan terbatas

Fitur-fitur Microsoft SQL Server berikut tidak didukung di Amazon RDS:

- Mencadangkan ke Microsoft Azure Blob Storage
- Ekstensi kumpulan buffer
- Kebijakan kata sandi kustom
- Layanan Kualitas Data
- Pengiriman Log Basis Data
- Snapshot basis data (Amazon RDS hanya mendukung snapshot instans DB)
- Prosedur tersimpan yang diperluas, termasuk xp_cmdshell
- Dukungan FILESTREAM
- Tabel file
- Machine Learning dan R Services (memerlukan akses OS untuk menginstalnya)

- Rencana pemeliharaan
- Pengumpul Data Performa
- Manajemen Berbasis Kebijakan
- PolyBase
- Replikasi
- Resource Governor
- Pemicu tingkat server
- Titik Akhir Service Broker
- Basis data stretch
- Properti basis data YANG DAPAT DIPERCAYA (membutuhkan peran sysadmin)
- Titik akhir T-SQL (semua operasi yang menggunakan CREATE ENDPOINT tidak tersedia)
- WCF Data Services

Fitur Microsoft SQL Server berikut memiliki dukungan terbatas di Amazon RDS:

- Kueri terdistribusi/server tertaut. Untuk informasi selengkapnya, lihat Menerapkan server tertaut dengan Amazon RDS for Microsoft SQL Server.
- Runtime Bahasa Umum (CLR). Di RDS for SQL Server 2016 dan versi yang lebih rendah, CLR didukung dalam mode SAFE dan hanya menggunakan bit assembly. CLR tidak didukung di RDS for SQL Server 2017 dan versi yang lebih tinggi. Untuk informasi selengkapnya, lihat Common Runtime Language Integration di dokumentasi Microsoft.

Fitur berikut tidak didukung di Amazon RDS dengan SQL Server 2022:

- Penangguhan basis data untuk snapshot
- Sumber Data Eksternal
- Pencadangan dan pemulihan ke penyimpanan objek yang kompatibel dengan S3
- Integrasi penyimpanan objek
- TLS 1.3 dan MS-TDS 8.0
- Pembongkaran kompresi cadangan dengan QAT
- SQL Server Analysis Services (SSAS)
- Pencerminan basis data dengan deployment Multi-AZ. SQL Server Always On adalah satu-satunya metode yang didukung dengan deployment Multi-AZ.

Deployment Multi-AZ menggunakan Microsoft SQL Server Database Mirroring atau Always On Availability Group

Amazon RDS mendukung deployment Multi-AZ untuk instans DB yang menjalankan Microsoft SQL Server menggunakan SQL Server Database Mirroring (DBM) atau Always On Availability Groups (AG). Deployment Multi-AZ memberikan peningkatan ketersediaan, durabilitas data, dan toleransi kesalahan untuk instans DB. Jika terjadi pemeliharaan basis data yang direncanakan atau gangguan layanan yang tidak direncanakan, Amazon RDS secara otomatis gagal ke replika up-to-date sekunder sehingga operasi basis data dapat dilanjutkan dengan cepat tanpa intervensi manual. Instans primer dan sekunder menggunakan titik akhir yang sama, yang alamat jaringan fisiknya bertransisi ke replika sekunder pasif sebagai bagian dari proses failover. Anda tidak perlu mengonfigurasi ulang aplikasi saat terjadi failover.

Amazon RDS mengelola failover dengan secara aktif memantau deployment Multi-AZ Anda dan memulai failover ketika masalah pada instans primer terjadi. Failover tidak akan terjadi kecuali instans siaga dan primer sepenuhnya sinkron. Amazon RDS secara aktif mempertahankan deployment Multi-AZ Anda dengan otomatis memperbaiki instans DB yang tidak berkondisi baik dan membangun kembali replikasi sinkron. Anda tidak perlu mengatur apa pun. Amazon RDS menangani instans primer, pemantau, dan siaga untuk Anda. Saat Anda menyiapkan Multi-AZ SQL Server, RDS mengonfigurasi instans sekunder pasif untuk semua basis data pada instans.

Untuk informasi lebih lanjut, lihat Deployment Multi-AZ untuk Amazon RDS for Microsoft SQL Server.

Menggunakan Enkripsi Data Transparan untuk mengenkripsi data diam

Amazon RDS mendukung Enkripsi Data Transparan (TDE) Microsoft SQL Server, yang secara transparan mengenkripsi data yang disimpan. Amazon RDS menggunakan grup opsi untuk mengaktifkan dan mengonfigurasi fitur ini. Untuk informasi selengkapnya tentang opsi TDE, lihat Dukungan untuk Enkripsi Data Transparan di SQL Server.

Fungsi dan prosedur tersimpan Amazon RDS for Microsoft SQL Server

Berikut ini, Anda dapat menemukan daftar prosedur tersimpan dan fungsi Amazon RDS yang membantu mengotomatiskan tugas SQL Server.

Jenis tugas	Prosedur atau fungsi	Penggunaannya
Tugas administrasi	`rds_drop_database`	Menghapus sementara basis data Microsoft SQL Server
	`rds_failover_time`	Menentukan waktu failover terakhir
	`rds_modify_db_name`	Mengganti nama basis data Microsoft SQL Server dalam deployment Multi-AZ
	`rds_read_error_log`	Melihat log kesalahan dan agen
	`rds_set_configuration`	Operasi ini digunakan untuk mengatur berbagai konfigurasi instans DB: • Pengambilan data perubahan untuk instans Multi-AZ • Mengatur periode retensi untuk file pelacakan dan dump • Mengompresi file backup
	`rds_set_database_online`	Melakukan transisi basis data Microsoft SQL Server dari OFFLINE ke ONLINE
	`rds_set_system_database_sync_objects`	Mengaktifkan replikasi pekerjaan SQL Server Agent
	`rds_fn_get_system_database_`	

Jenis tugas	Prosedur atau fungsi	Penggunaannya

	`sync_objects`
	
	`rds_fn_server_object_last_sync_time`	

	`rds_show_configuration`	Untuk melihat nilai yang ditetapkan menggunakan `rds_set_configuration` , lihat topik berikut: • Pengambilan data perubahan untuk instans Multi-AZ • Mengatur periode retensi untuk file pelacakan dan dump
	`rds_shrink_tempdbfile`	Mengurangi basis data tempdb
Change Data Capture (CDC)	`rds_cdc_disable_db`	Menonaktifkan CDC
	`rds_cdc_enable_db`	Mengaktifkan CDC
Database Mail	`rds_fn_sysmail_allitems`	Melihat pesan, log, dan lampiran
	`rds_fn_sysmail_event_log`	Melihat pesan, log, dan lampiran

Jenis tugas	Prosedur atau fungsi	Penggunaannya
	`rds_fn_sysmail_mailattachments`	Melihat pesan, log, dan lampiran
	`rds_sysmail_control`	Operasi ini digunakan dalam memulai dan menghentikan antrean email: • Memulai antrean email • Menghentikan antrean email
	`rds_sysmail_delete_mailitems_sp`	Menghapus pesan
Pencadangan dan pemulihan native	`rds_backup_database`	Membuat backup basis data
	`rds_cancel_task`	Membatalkan tugas
	`rds_finish_restore`	Menyelesaikan pemulihan basis data
	`rds_restore_database`	Memulihkan basis data
	`rds_restore_log`	Memulihkan log

Jenis tugas	Prosedur atau fungsi	Penggunaannya
Transfer file Amazon S3	`rds_delete_from_filesystem`	Menghapus file di instans DB RDS
	`rds_download_from_s3`	Mengunduh file dari bucket Amazon S3 ke instans DB SQL Server
	`rds_gather_file_details`	Menampilkan daftar file di instans DB RDS
	`rds_upload_to_s3`	Mengunggah file dari instans DB SQL Server ke bucket Amazon S3
Microsoft Distributed Transaction Coordinator (MSDTC)	`rds_msdtc_transaction_tracing`	Menggunakan pelacakan transaksi
SQL Server Audit	`rds_fn_get_audit_file`	Melihat log audit

Jenis tugas	Prosedur atau fungsi	Penggunaannya

Enkripsi Data Transparan	`rds_backup_tde_certificate`
	
	`rds_drop_tde_certificate`
	
	`rds_restore_tde_certificate`
	
	`rds_fn_list_user_tde_certificates`
		Dukungan untuk Enkripsi Data Transparan di SQL Server

Jenis tugas	Prosedur atau fungsi	Penggunaannya
Microsoft Business Intelligence (MSBI)	`rds_msbi_task`	Operasi ini digunakan dengan SQL Server Analysis Services (SSAS): • Men-deploy proyek SSAS di Amazon RDS • Menambahkan pengguna domain sebagai administrator basis data • Membuat cadangan basis data SSAS • Memulihkan basis data SSAS Operasi ini juga digunakan dengan SQL Server Integration Services (SSIS): • Izin administratif untuk SSISDB • Melakukan deployment satu proyek SSIS Operasi ini juga digunakan dengan SQL Server Reporting Services (SSRS): • Memberikan akses ke pengguna domain • Mencabut izin tingkat sistem
	`rds_fn_task_status`	Operasi ini menunjukkan status tugas MSBI: • SSAS: Memantau status tugas deployment • SSIS: Memantau status tugas deployment • SSRS: Memantau status tugas
SSIS	`rds_drop_ssis_database`	Menurunkan basis data SSISDB

Jenis tugas	Prosedur atau fungsi	Penggunaannya
	`rds_sqlagent_proxy`	Membuat proksi SSIS
SSRS	`rds_drop_ssrs_databases`	Menghapus basis data SSRS

Zona waktu lokal untuk instans DB Microsoft SQL Server

Zona waktu instans DB Amazon RDS yang menjalankan Microsoft SQL Server diatur secara default. Nilai default saat ini adalah Waktu Universal Terkoordinasi (UTC). Anda dapat mengatur zona waktu instans DB ke zona waktu lokal agar sesuai dengan zona waktu aplikasi Anda.

Anda akan menetapkan zona waktu saat pertama kali membuat instans DB. Anda dapat membuat instans DB dengan menggunakan AWS Management Console, tindakan CreatedBInstance Amazon RDS API, atau perintah. AWS CLI create-db-instance

Jika instans DB Anda adalah bagian dari deployment Multi-AZ (menggunakan SQL Server DBM atau AG), maka ketika Anda gagal, zona waktu Anda akan tetap menjadi zona waktu lokal yang Anda tetapkan. Untuk informasi selengkapnya, lihat Deployment Multi-AZ menggunakan Microsoft SQL Server Database Mirroring atau Always On Availability Group.

Saat Anda meminta point-in-time pemulihan, Anda menentukan waktu untuk memulihkannya. Waktu ditampilkan dalam zona waktu lokal Anda. Untuk informasi selengkapnya, lihat Memulihkan instans DB dengan waktu yang ditentukan.

Berikut ini adalah batasan untuk menetapkan zona waktu lokal pada instans DB:

- Anda tidak dapat mengubah zona waktu Instans DB SQL Server yang sudah ada.
- Anda tidak dapat memulihkan snapshot dari instans DB dalam satu zona waktu ke instans DB dalam zona waktu yang berbeda.
- Kami sangat menyarankan agar Anda tidak memulihkan file cadangan dari satu zona waktu ke zona waktu yang berbeda. Jika memulihkan file cadangan dari satu zona waktu ke zona waktu yang berbeda, Anda harus mengaudit kueri dan aplikasi Anda untuk mengetahui efek dari

perubahan zona waktu. Untuk informasi selengkapnya, lihat Mengimpor dan mengekspor basis data SQL Server menggunakan pencadangan dan pemulihan native.

Zona waktu yang didukung

Anda dapat mengatur zona waktu lokal Anda ke salah satu nilai yang tercantum dalam tabel berikut ini.

Zona waktu yang didukung untuk Amazon RDS di SQL Server

Zona waktu	Offset waktu standar	Deskripsi	Catatan
Waktu Standar Afghanistan	(UTC+04.30)	Kabul	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Alaska	(UTC–09.00)	Alaska	
Waktu Standar Aleutian	(UTC–10.00)	Kepulauan Aleutian	
Waktu Standar Altai	(UTC+07.00)	Barnaul, Gorno-Altaysk	
Waktu Standar Arab	(UTC+03.00)	Kuwait, Riyadh	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Arab	(UTC+04.00)	Abu Dhabi, Muscat	
Waktu Standar Arab	(UTC+03.00)	Bagdad	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Argentina	(UTC–03.00)	Kota Buenos Aires	Zona waktu ini tidak menggunak

Zona waktu	Offset waktu standar	Deskripsi	Catatan
			an waktu musim panas.
Waktu Standar Astrakhan	(UTC+04.00)	Astrakhan, Ulyanovsk	
Waktu Standar Atlantik	(UTC–04.00)	Waktu Atlantik (Kanada)	
Waktu Standar Tengah AUS	(UTC+09.30)	Darwin	Zona waktu ini tidak menggunak an waktu musim panas.
Waktu Standar Barat Tengah Aus	(UTC+08.45)	Eucla	
Waktu Standar Timur AUS	(UTC+10.00)	Canberra, Melbourne, Sydney	
Waktu Standar Azerbaijan	(UTC+04.00)	Baku	
Waktu Standar Azores	(UTC–01.00)	Azores	
Waktu Standar Bahia	(UTC–03.00)	Salvador	
Waktu Standar Bangladesh	(UTC+06.00)	Dhaka	Zona waktu ini tidak menggunak an waktu musim panas.
Waktu Standar Belarusia	(UTC+03.00)	Minsk	Zona waktu ini tidak menggunak an waktu musim panas.
Waktu Standar Bougainville	(UTC+11.00)	Pulau Bougainville	

Zona waktu	Offset waktu standar	Deskripsi	Catatan
Waktu Standar Kanada Pusat	(UTC–06.00)	Saskatchewan	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Cape Verde	(UTC–01.00)	Cabo Verde Is.	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Kaukasus	(UTC+04.00)	Yerevan	
Cen. Waktu Standar Australia	(UTC+09.30)	Adelaide	
Waktu Standar Amerika Tengah	(UTC–06.00)	Amerika Tengah	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Asia Tengah	(UTC+06.00)	Astana	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Brasil Tengah	(UTC–04.00)	Cuiaba	
Waktu Standar Eropa Tengah	(UTC+01.00)	Belgrade, Bratislava, Budapest, Ljubljana, Praha	
Waktu Standar Eropa Tengah	(UTC+01.00)	Sarajevo, Skopje, Warsawa, Zagreb	

Zona waktu	Offset waktu standar	Deskripsi	Catatan
Waktu Standar Pasifik Tengah	(UTC+11.00)	Kepulauan Solomon, Kaledonia Baru	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Tengah	(UTC–06.00)	Waktu Tengah (AS dan Kanada)	
Waktu Standar Tengah (Meksiko)	(UTC–06.00)	Guadalajara, Mexico City, Monterrey	
Waktu Standar Kepulauan Chatham	(UTC+12.45)	Kepulauan Chatham	
Waktu Standar Tiongkok	(UTC+08.00)	Beijing, Chongqing, Hong Kong, Urumqi	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Kuba	(UTC–05.00)	Havana	
Waktu Standar Garis Batas Tanggal	(UTC–12.00)	Garis Batas Tanggal Internasional Bagian Barat	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Afrika Timur	(UTC+03.00)	Nairobi	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Australia Timur	(UTC+10.00)	Brisbane	Zona waktu ini tidak menggunakan waktu musim panas.

Zona waktu	Offset waktu standar	Deskripsi	Catatan
Waktu Standar Eropa Timur	(UTC+02.00)	Chisinau	
Waktu Standar Amerika Selatan Timur	(UTC–03.00)	Brasilia	
Waktu Standar Pulau Paskah	(UTC–06.00)	Pulau Paskah	
Waktu Standar Timur	(UTC–05.00)	Waktu Timur (AS dan Kanada)	
Waktu Standar Timur (Meksiko)	(UTC–05.00)	Chetumal	
Waktu Standar Mesir	(UTC+02.00)	Kairo	
Waktu Standar Ekaterinburg	(UTC+05.00)	Ekaterinburg	
Waktu Standar Fiji	(UTC+12.00)	Fiji	
Waktu Standar FLE	(UTC+02.00)	Helsinki, Kyiv, Riga, Sofia, Tallinn, Vilnius	
Waktu Standar Georgia	(UTC+04.00)	Tbilisi	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar GMT	(UTC)	Dublin, Edinburgh, Lisbon, London	Zona waktu ini tidak sama dengan Greenwich Mean Time. Zona waktu ini menggunakan waktu musim panas.

Zona waktu	Offset waktu standar	Deskripsi	Catatan
Waktu Standar Greenland	(UTC–03.00)	Greenland	
Waktu Standar Greenwich	(UTC)	Monrovia, Reykjavik	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar GTB	(UTC+02.00)	Athena, Bukares	
Waktu Standar Haiti	(UTC–05.00)	Haiti	
Waktu Standar Hawaii	(UTC–10.00)	Hawaii	
Waktu Standar India	(UTC+05.30)	Chennai, Kolkata, Mumbai, New Delhi	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Iran	(UTC+03.30)	Teheran	
Waktu Standar Israel	(UTC+02.00)	Yerusalem	
Waktu Standar Yordania	(UTC+02.00)	Amman	
Waktu Standar Kaliningrad	(UTC+02.00)	Kaliningrad	
Waktu Standar Kamchatka	(UTC+12.00)	Petropavlovsk-Kamchatsky – Lama	
Waktu Standar Korea	(UTC+09.00)	Seoul	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Libya	(UTC+02.00)	Tripoli	

Zona waktu	Offset waktu standar	Deskripsi	Catatan
Waktu Standar Kepulauan Line	(UTC+14.00)	Pulau Kiritimati	
Waktu Standar Lord Howe	(UTC+10.30)	Pulau Lord Howe	
Waktu Standar Magadan	(UTC+11.00)	Magadan	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Magallanes	(UTC–03.00)	Punta Arenas	
Waktu Standar Marquesas	(UTC–09.30)	Kepulauan Marquesas	
Waktu Standar Mauritius	(UTC+04.00)	Port Louis	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Timur Tengah	(UTC+02.00)	Beirut	
Waktu Standar Montevideo	(UTC–03.00)	Montevideo	
Waktu Standar Maroko	(UTC+01.00)	Casablanca	
Waktu Standar Pegunungan	(UTC–07.00)	Waktu Pegunungan (AS dan Kanada)	
Waktu Standar Pegunungan (Meksiko)	(UTC–07.00)	Chihuahua, La Paz, Mazatlan	
Waktu Standar Myanmar	(UTC+06.30)	Yangon (Rangoon)	Zona waktu ini tidak menggunakan waktu musim panas.

Zona waktu	Offset waktu standar	Deskripsi	Catatan
Waktu Standar Asia Tengah Utara	(UTC+07.00)	Novosibirsk	
Waktu Standar Namibia	(UTC+02.00)	Windhoek	
Waktu Standar Nepal	(UTC+05.45)	Kathmandu	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Selandia Baru	(UTC+12.00)	Auckland, Wellington	
Waktu Standar Newfoundland	(UTC–03.30)	Newfoundland	
Waktu Standar Norfolk	(UTC+11.00)	Pulau Norfolk	
Waktu Standar Timur Asia Utara	(UTC+08.00)	Irkutsk	
Waktu Standar Asia Utara	(UTC+07.00)	Krasnoyarsk	
Waktu Standar Korea Utara	(UTC+09.00)	Pyongyang	
Waktu Standar Omsk	(UTC+06.00)	Omsk	
Waktu Standar SA Pasifik	(UTC–03.00)	Santiago	
Waktu Standar Pasifik	(UTC–08.00)	Waktu Pasifik (AS dan Kanada)	
Waktu Standar Pasifik (Meksiko)	(UTC–08.00)	Baja California	

Zona waktu	Offset waktu standar	Deskripsi	Catatan
Waktu Standar Pakistan	(UTC+05.00)	Islamabad, Karachi	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Paraguay	(UTC–04.00)	Asuncion	
Waktu Standar Romance	(UTC+01.00)	Brussels, Copenhagen, Madrid, Paris	
Zona Waktu Rusia 10	(UTC+11.00)	Chokurdakh	
Zona Waktu Rusia 11	(UTC+12.00)	Anadyr, Petropavlovsk-Kamchatsky	
Zona Waktu Rusia 3	(UTC+04.00)	Izhevsk, Samara	
Waktu Standar Rusia	(UTC+03.00)	Moskow, St. Petersburg, Volgograd	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Timur SA	(UTC–03.00)	Cayenne, Fortaleza	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Pasifik SA	(UTC–05.00)	Bogota, Lima, Quito, Rio Branco	Zona waktu ini tidak menggunakan waktu musim panas.

Zona waktu	Offset waktu standar	Deskripsi	Catatan
Waktu Standar Barat SA	(UTC–04.00)	Georgetown, La Paz, Manaus, San Juan	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Saint Pierre	(UTC–03.00)	Saint Pierre dan Miquelon	
Waktu Standar Sakhalin	(UTC+11.00)	Sakhalin	
Waktu Standar Samoa	(UTC+13.00)	Samoa	
Waktu Standar Sao Tome	(UTC+01.00)	Sao Tome	
Waktu Standar Saratov	(UTC+04.00)	Saratov	
Waktu Standar Asia Tenggara	(UTC+07.00)	Bangkok, Hanoi, Jakarta	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Singapura	(UTC+08.00)	Kuala Lumpur, Singapura	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Afrika Selatan	(UTC+02.00)	Harare, Pretoria	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Sri Lanka	(UTC+05.30)	Sri Jayawardenepura	Zona waktu ini tidak menggunakan waktu musim panas.

Zona waktu	Offset waktu standar	Deskripsi	Catatan
Waktu Standar Sudan	(UTC+02.00)	Khartoum	
Waktu Standar Syria	(UTC+02.00)	Damascus	
Waktu Standar Taipei	(UTC+08.00)	Taipei	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Tasmania	(UTC+10.00)	Hobart	
Waktu Standar Tocantins	(UTC–03.00)	Araguaina	
Waktu Standar Tokyo	(UTC+09.00)	Osaka, Sapporo, Tokyo	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Tomsk	(UTC+07.00)	Tomsk	
Waktu Standar Tonga	(UTC+13.00)	Nuku'alofa	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Transbaikal	(UTC+09.00)	Chita	
Waktu Standar Turki	(UTC+03.00)	Istanbul	
Waktu Standar Turks dan Caicos	(UTC–05.00)	Turks dan Caicos	
Waktu Standar Ulaanbaatar	(UTC+08.00)	Ulaanbaatar	Zona waktu ini tidak menggunakan waktu musim panas.

Zona waktu	Offset waktu standar	Deskripsi	Catatan
Waktu Standar Timur AS	(UTC–05.00)	Indiana (Timur)	
Waktu Standar Pegunungan AS	(UTC–07.00)	Arizona	Zona waktu ini tidak menggunakan waktu musim panas.
UTC	UTC	Waktu Universal Terkoordinasi	Zona waktu ini tidak menggunakan waktu musim panas.
UTC–02	(UTC–02.00)	Waktu Universal Terkoordinasi–02	Zona waktu ini tidak menggunakan waktu musim panas.
UTC–08	(UTC–08.00)	Waktu Universal Terkoordinasi–08	
UTC–09	(UTC–09.00)	Waktu Universal Terkoordinasi–09	
UTC–11	(UTC–11.00)	Waktu Universal Terkoordinasi–11	Zona waktu ini tidak menggunakan waktu musim panas.
UTC+12	(UTC+12.00)	Waktu Universal Terkoordinasi+12	Zona waktu ini tidak menggunakan waktu musim panas.
UTC+13	(UTC+13.00)	Waktu Universal Terkoordinasi+13	

Zona waktu	Offset waktu standar	Deskripsi	Catatan
Waktu Standar Venezuela	(UTC–04.00)	Caracas	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Vladivostok	(UTC+10.00)	Vladivostok	
Waktu Standar Volgograd	(UTC+04.00)	Volgograd	
Waktu Standar Australia Barat	(UTC+08.00)	Perth	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Afrika Tengah Barat	(UTC+01.00)	Afrika Tengah Barat	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Eropa Barat	(UTC+01.00)	Amsterdam, Berlin, Bern, Roma, Stockholm, Wina	
Waktu Standar Mongolia Barat	(UTC+07.00)	Hovd	
Waktu Standar Asia Barat	(UTC+05.00)	Ashgabat, Tashkent	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Tepi Barat	(UTC+02.00)	Gaza, Hebron	

Zona waktu	Offset waktu standar	Deskripsi	Catatan
Waktu Standar Pasifik Barat	(UTC+10.00)	Guam, Port Moresby	Zona waktu ini tidak menggunakan waktu musim panas.
Waktu Standar Yakutsk	(UTC+09.00)	Yakutsk	

Melisensikan Microsoft SQL Server di Amazon RDS

Saat Anda menyiapkan instans DB Amazon RDS untuk Microsoft SQL Server, lisensi perangkat lunak akan disertakan.

Ini berarti Anda tidak perlu membeli lisensi SQL Server terpisah. AWS memiliki lisensi untuk perangkat lunak basis data SQL Server. Harga Amazon RDS mencakup lisensi perangkat lunak, sumber daya perangkat keras yang mendasari, dan kemampuan manajemen Amazon RDS.

Amazon RDS mendukung edisi Microsoft SQL Server berikut:

- Perusahaan
- Standar
- Web
- Ekspres

> **Note**
>
> Lisensi untuk Edisi Web SQL Server hanya mendukung halaman web publik dan yang dapat diakses internet, EBSIT, aplikasi web, dan layanan web. Tingkat dukungan ini diperlukan untuk mematuhi hak penggunaan Microsoft. Untuk informasi selengkapnya, lihat ketentuan layanan AWS.

Amazon RDS mendukung deployment Multi-AZ untuk instans DB yang menjalankan Microsoft SQL Server menggunakan SQL Server Database Mirroring (DBM) atau Always On Availability Groups (AG). Tidak ada persyaratan lisensi tambahan untuk deployment multi-AZ. Untuk informasi selengkapnya, lihat Deployment Multi-AZ untuk Amazon RDS for Microsoft SQL Server.

Memulihkan instans DB yang telah dihentikan lisensinya

Amazon RDS mengambil gambar instans DB yang dihentikan lisensinya. Jika instans Anda dihentikan karena masalah lisensi, Anda dapat memulihkannya dari snapshot ke instans DB baru. Instans DB baru memiliki lisensi yang disertakan.

Untuk informasi selengkapnya, lihat Memulihkan instans DB yang telah dihentikan lisensinya.

Pengembangan dan pengujian

Karena persyaratan lisensi, kami tidak dapat menawarkan Edisi Pengembang SQL Server di Amazon RDS. Anda dapat menggunakan Edisi Ekspres untuk banyak kebutuhan pengembangan, pengujian, dan nonproduksi lainnya. Namun demikian, jika Anda memerlukan fitur lengkap dari instalasi SQL Server tingkat perusahaan untuk pengembangan, Anda dapat mengunduh dan menginstal Edisi Pengembang SQL Server di RDS Custom untuk SQL Server menggunakan CEV dengan BYOM. Untuk informasi selengkapnya, lihat Mempersiapkan CEV menggunakan Bawa Media Anda Sendiri (BYOM). Infrastruktur khusus tidak diperlukan untuk Edisi Pengembang. Dengan menggunakan host Anda sendiri, Anda juga mendapatkan akses ke fitur kemampuan program lainnya yang tidak dapat diakses di Amazon RDS. Untuk informasi selengkapnya tentang perbedaan antara edisi SQL Server, lihat Edisi dan fitur yang didukung SQL Server 2019 di dokumentasi Microsoft.

Menghubungkan ke instans DB yang menjalankan mesin basis data Microsoft SQL Server

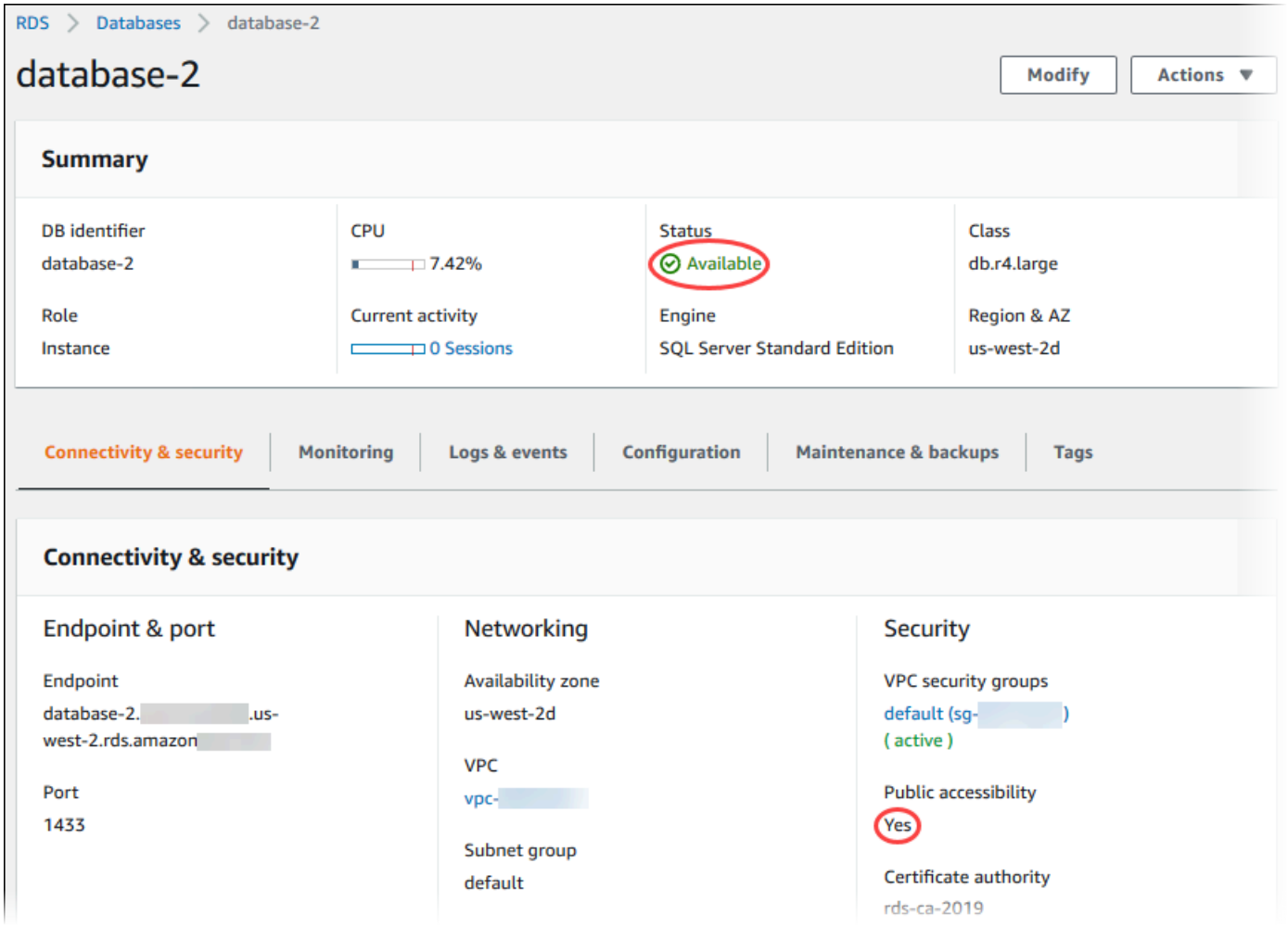
Setelah Amazon RDS menyediakan instans DB, Anda dapat menggunakan aplikasi klien SQL standar guna terhubung ke instans DB. Dalam topik ini, Anda terhubung ke instans DB Anda menggunakan Microsoft SQL Server Management Studio (SSMS) atau SQL Workbench/J.

Sebagai contoh yang akan memandu Anda melalui proses membuat dan menghubungkan ke instans DB sampel, lihat Membuat dan menghubungkan ke instans DB Microsoft SQL Server.

Sebelum Anda menyambungkan

Sebelum dapat terhubung ke instans DB, instans tersebut harus tersedia dan dapat diakses.

1. Pastikan statusnya `available`. Anda dapat memeriksa hal ini pada halaman detail untuk instans Anda di AWS Management Console atau menggunakan perintah AWS CLI describe-db-instances.



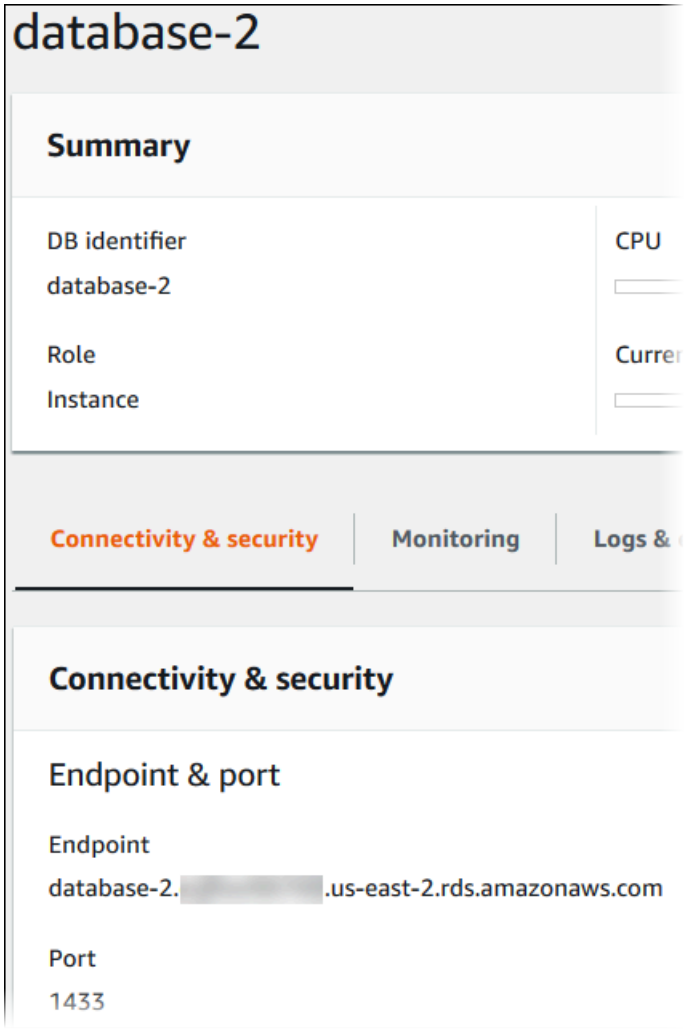
2. Pastikan bahwa instans tersebut dapat diakses sumber Anda. Tergantung pada skenario Anda, instans tersebut mungkin tidak perlu diakses publik. Untuk mengetahui informasi selengkapnya, lihat Amazon VPC dan Amazon RDS.
3. Pastikan bahwa aturan masuk grup keamanan VPC Anda memungkinkan akses ke instans DB Anda. Untuk mengetahui informasi selengkapnya, lihat Tidak dapat terhubung ke instans DB Amazon RDS.

Menemukan nomor port dan titik akhir instans DB

Anda memerlukan titik akhir dan nomor port untuk terhubung ke instans DB.

Untuk menemukan titik akhir dan port

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di sudut kanan atas konsol Amazon RDS, pilih Wilayah AWS untuk instans DB Anda.
3. Temukan nomor port dan (titik akhir) nama Sistem Nama Domain (DNS) untuk instans DB Anda:
    a. Buka konsol RDS dan pilih Basis data untuk menampilkan daftar instans DB Anda.
    b. Pilih nama instans DB SQL Server untuk menampilkan detailnya.
    c. Di tab Konektivitas & keamanan, salin titik akhir.



d. Catat nomor port tersebut.

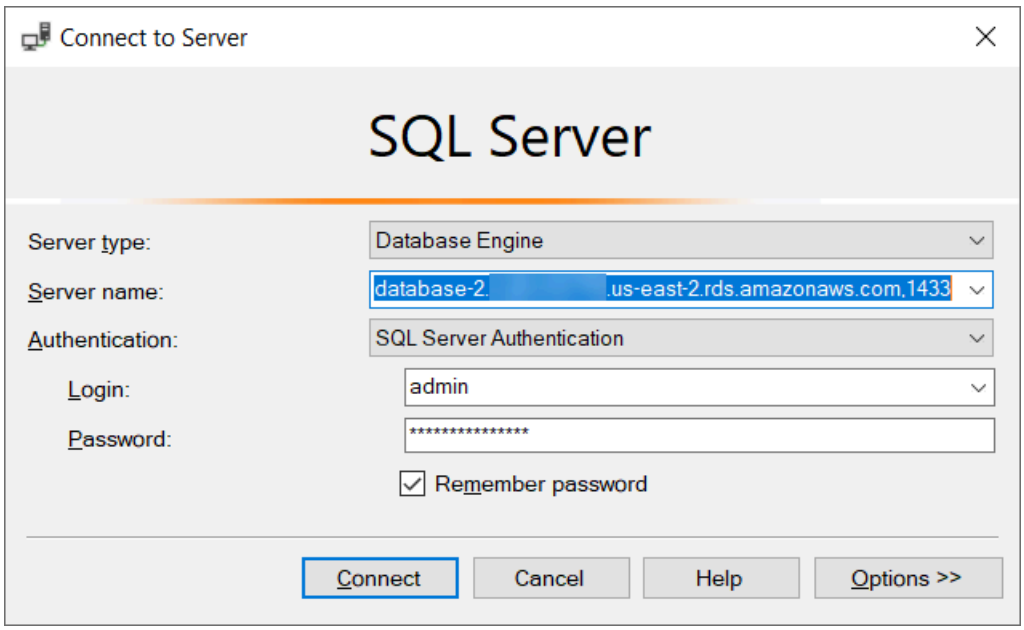
Menghubungkan ke instans DB Anda dengan Microsoft SQL Server Management Studio

Dalam prosedur ini, Anda terhubung ke sampel instans DB menggunakan Microsoft SQL Server Management Studio (SSMS). Untuk mengunduh versi mandiri dari utilitas ini, lihat Mengunduh SQL Server Management Studio (SSMS) dalam dokumentasi Microsoft.

Untuk terhubung ke instans DB menggunakan SSMS

1. Mulai SQL Server Management Studio.

   Lalu, kotak dialog Hubungkan ke Server akan muncul.



2. Berikan informasi untuk instans DB Anda:

   a. Untuk Jenis server, pilih Mesin Basis Data.

   b. Untuk Nama server, masukkan nama DNS (titik akhir) dan nomor port instans DB Anda, yang dipisahkan dengan koma.

      > **Important**
      >
      > Ubah titik dua antara titik akhir dan nomor port menjadi koma.

      Nama server Anda akan terlihat seperti contoh berikut.

      ```
      database-2.cg034itsfake.us-east-1.rds.amazonaws.com,1433
      ```

   c. Untuk Autentikasi, pilih Autentikasi SQL Server.

   d. Untuk Masuk, masukkan nama pengguna utama untuk instans DB Anda.

   e. Untuk Kata sandi, masukkan kata sandi untuk instans DB Anda.

3. Pilih Hubungkan.

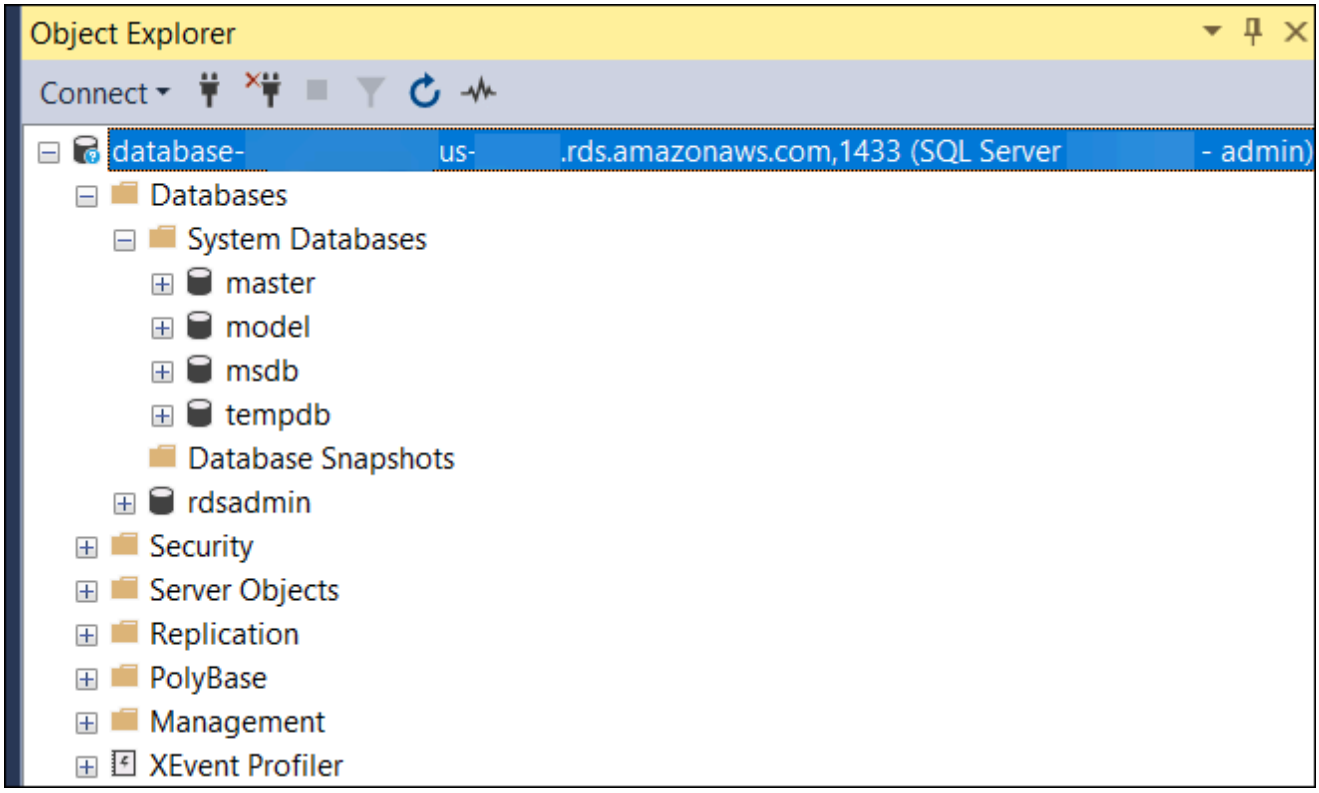
   Setelah beberapa saat, SSMS akan terhubung ke instans DB Anda.

   Jika tidak dapat terhubung ke instans DB Anda, lihat Pertimbangan grup keamanan dan Memecahkan masalah koneksi ke instans DB SQL Server Anda.

4. Instans DB SQL Server Anda hadir dengan data sistem bawaan standar SQL Server (`master`, `model`, `msdb`, dan `tempdb`). Untuk menjelajahi basis data sistem, lakukan tindakan berikut:

   a. Di SSMS, pada menu Lihat, pilih Object Explorer.

   b. Perluas instans DB Anda, perluas Basis Data, lalu perluas Basis Data Sistem.



5. Instans DB SQL Server Anda juga dilengkapi dengan basis data bernama `rdsadmin`. Amazon RDS menggunakan basis data ini untuk menyimpan objek yang digunakan untuk mengelola basis data Anda. Basis data `rdsadmin` juga mencakup prosedur tersimpan yang dapat Anda jalankan untuk melakukan tugas tingkat lanjut. Untuk mengetahui informasi selengkapnya, lihat Tugas DBA umum untuk Microsoft SQL Server.

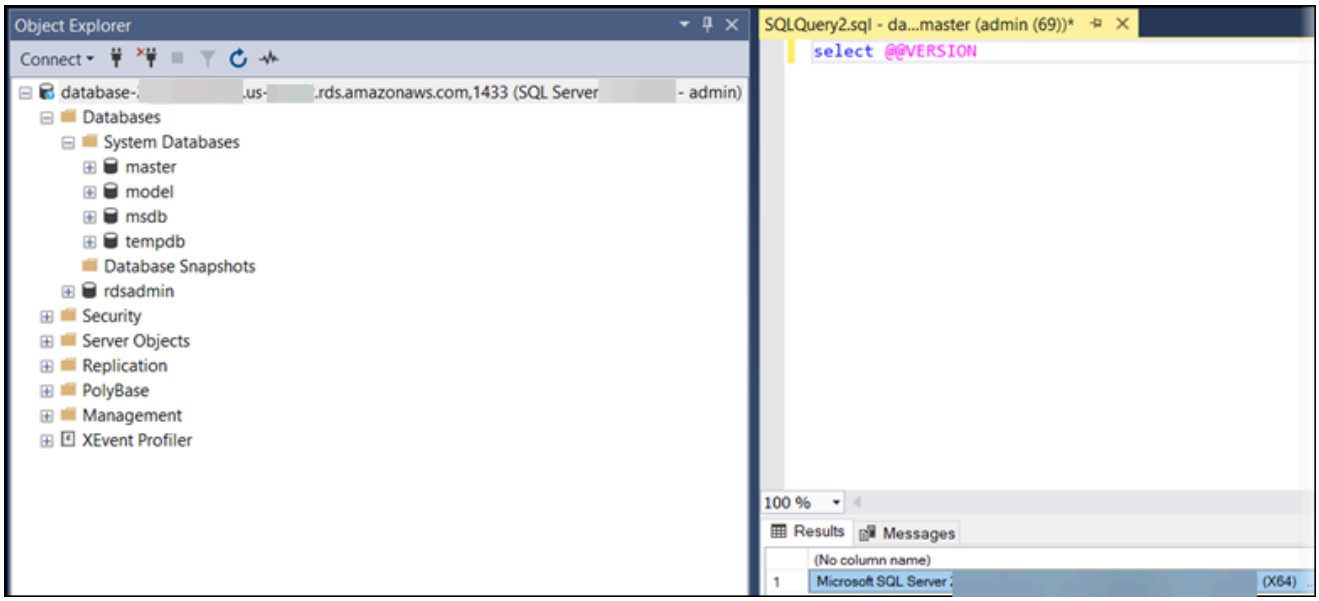
6. Sekarang Anda dapat mulai membuat basis data sendiri dan menjalankan kueri pada instans DB dan basis data Anda seperti biasa. Untuk menjalankan kueri pengujian pada instans DB Anda, lakukan tindakan berikut:

   a. Di SSMS, pada menu File, arahkan ke Baru, lalu pilih Kueri dengan Koneksi Saat Ini.

   b. Masukkan kueri SQL berikut.

```
select @@VERSION
```

c. Jalankan kueri. SSMS mengembalikan versi SQL Server dari instans DB Amazon RDS Anda.



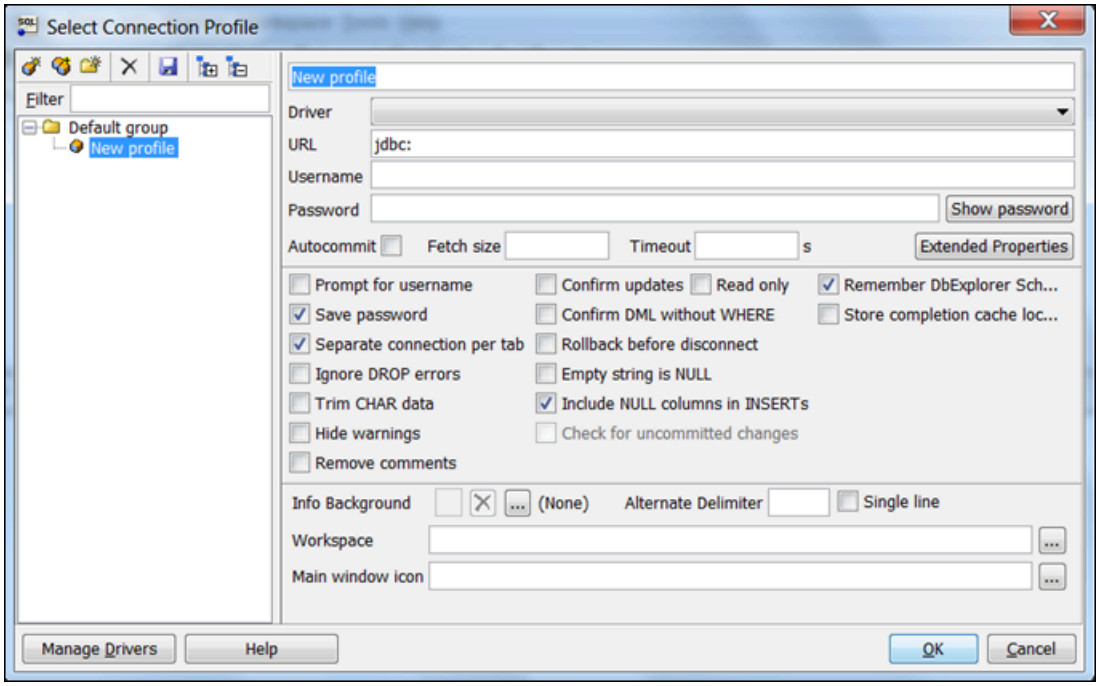
Menghubungkan ke instans DB Anda dengan SQL Workbench/J

Contoh ini menunjukkan cara menghubungkan ke instans DB yang menjalankan mesin basis data Microsoft SQL Server menggunakan alat basis data SQL Workbench/J. Untuk mengunduh SQL Workbench/J, lihat SQL Workbench/J.

SQL Workbench/J menggunakan JDBC untuk terhubung ke instans DB Anda. Anda juga memerlukan driver JDBC untuk SQL Server. Untuk mengunduh driver ini, lihat Driver JDBC Microsoft 4.1 (pratinjau) dan 4.0 untuk SQL Server.

Untuk terhubung ke instans DB menggunakan SQL Workbench/J

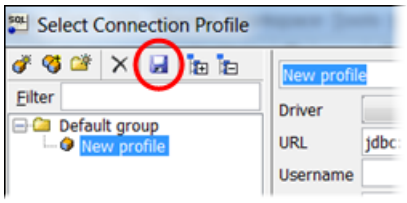
1. Buka SQL Workbench/J. Kotak dialog Pilih Profil Koneksi akan muncul, seperti yang ditunjukkan berikut ini.



2. Di kotak pertama pada bagian atas kotak dialog, masukkan nama untuk profil.
3. Untuk Driver, pilih **`SQL JDBC 4.0`**.
4. Untuk URL, masukkan **`jdbc:sqlserver://`**, lalu masukkan titik akhir instans DB Anda. Misalnya, nilai URL dapat berupa seperti berikut ini.

```
jdbc:sqlserver://sqlsvr-pdz.abcd12340.us-west-2.rds.amazonaws.com:1433
```

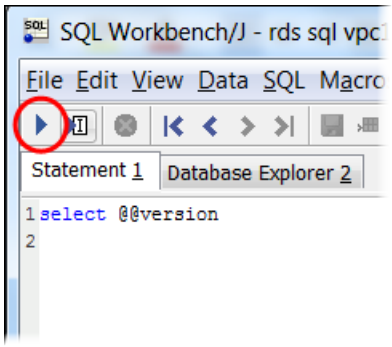
5. Untuk Nama pengguna, masukkan nama pengguna utama untuk instans DB.
6. Untuk Kata sandi, masukkan kata sandi untuk pengguna utama.
7. Pilih ikon simpan di toolbar dialog, seperti yang ditunjukkan berikut.



8. Pilih OKE. Setelah beberapa saat, SQL Workbench/J terhubung ke instans DB Anda. Jika tidak dapat terhubung ke instans DB Anda, lihat Pertimbangan grup keamanan dan Memecahkan masalah koneksi ke instans DB SQL Server Anda.
9. Dalam panel kueri, masukkan kueri SQL berikut.

```
select @@VERSION
```

10. Pilih ikon `Execute` di toolbar, seperti yang ditunjukkan berikut.



Kueri menampilkan informasi versi untuk instans DB Anda, serupa dengan yang berikut.

```
Microsoft SQL Server 2017 (RTM-CU22) (KB4577467) - 14.0.3356.20 (X64)
```

Pertimbangan grup keamanan

Untuk terhubung ke instans DB Anda, instans DB Anda harus dikaitkan dengan grup keamanan. Grup keamanan ini berisi alamat IP dan konfigurasi jaringan yang Anda gunakan untuk mengakses instans DB. Anda mungkin telah mengaitkan instans DB dengan grup keamanan yang sesuai saat Anda membuat instans DB Anda. Jika Anda menetapkan grup keamanan yang tidak dikonfigurasi saat membuat instans DB Anda, firewall instans DB Anda akan mencegah koneksi.

Dalam beberapa kasus, Anda mungkin perlu membuat grup keamanan baru untuk memungkinkan akses. Untuk petunjuk tentang cara membuat grup keamanan baru, lihat Mengontrol akses dengan grup keamanan. Untuk topik yang dapat memandu Anda melalui proses penyiapan aturan grup keamanan VPC Anda, lihat Tutorial: Membuat VPC untuk digunakan dengan instans DB (khusus IPv4).

Setelah Anda membuat grup keamanan baru, ubah instans DB Anda untuk dikaitkan dengan grup keamanan. Untuk mengetahui informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Anda dapat meningkatkan keamanan menggunakan SSL untuk mengenkripsi koneksi ke instans DB Anda. Untuk mengetahui informasi selengkapnya, lihat Menggunakan SSL dengan instans DB Microsoft SQL Server.

Memecahkan masalah koneksi ke instans DB SQL Server Anda

Tabel berikut menunjukkan pesan kesalahan yang dapat terjadi saat mencoba untuk terhubung ke instans DB SQL Server Anda.

Masalah	Saran pemecahan masalah
Tidak dapat membuka koneksi ke SQL Server — Microsoft SQL Server, Kesalahan: 53	Pastikan bahwa Anda menentukan nama server dengan benar. Untuk Nama server, masukkan nama DNS dan nomor port instans DB sampel Anda, dipisahkan dengan koma. **Important** Jika terdapat titik dua di antara nama DNS dan nomor port, ubah titik dua menjadi koma. Nama server Anda akan terlihat seperti contoh berikut. `sample-instance.cg034itsfake.us-east-1.rds.amazonaws.com,1433`
Tidak ada koneksi yang dapat dibuat karena mesin target secara aktif menolaknya — Microsoft SQL Server, Kesalahan: 10061	Anda dapat menghubungi instans DB, tetapi koneksinya ditolak. Masalah ini biasanya disebabkan karena kesalahan menentukan nama pengguna atau kata sandi. Verifikasi nama pengguna dan kata sandi, lalu coba lagi.
Terjadi kesalahan terkait jaringan atau khusus instans saat membuat koneksi ke SQL Server. Server tidak ditemukan atau tidak dapat diakses.. . Waktu operasi tunggu habis – Microsoft SQL Server, Kesalahan: 258	Aturan akses yang diterapkan oleh firewall lokal dan alamat IP yang diizinkan untuk mengakses instans DB mungkin tidak cocok. Kemungkinan besar masalahnya adalah aturan masuk dalam grup keamanan Anda. Untuk mengetahui informasi selengkapnya, lihat Keamanan dalam Amazon RDS.
	Instans basis data Anda harus dapat diakses secara publik. Untuk menghubungakannya dari luar VPC, instans harus memiliki alamat IP publik yang ditetapkan.

> **Note**
>
> Untuk mengetahui informasi selengkapnya tentang masalah koneksi, lihat Tidak dapat terhubung ke instans DB Amazon RDS.

Menggunakan Active Directory dengan RDS for SQL Server

Anda dapat menggabungkan instans DB RDS for SQL Server ke domain Microsoft Active Directory (AD). Domain AD Anda dapat di-host pada AD yang Dikelola AWS di dalam AWS, atau pada AD yang Dikelola Sendiri di lokasi pilihan Anda, termasuk pusat data korporasi Anda, di AWS EC2, atau di penyedia cloud lainnya.

Anda dapat mengautentikasi pengguna domain menggunakan autentikasi NTLM dengan Active Directory yang Dikelola Sendiri. Anda dapat menggunakan autentikasi Kerberos dan NTLM dengan Active Directory yang Dikelola AWS.

Pada bagian berikut, Anda dapat menemukan informasi tentang menggunakan Active Directory yang Dikelola Sendiri dan Active Directory yang Dikelola AWS untuk Microsoft SQL Server di Amazon RDS.

Topik

Menggunakan Directory Active yang Dikelola Sendiri dengan instans DB Amazon RDS for SQL Server

Anda dapat menggabungkan instans DB RDS for SQL Server langsung ke domain Active Directory (AD) yang dikelola sendiri, di mana pun AD Anda di-host: di pusat data korporasi, di AWS EC2, atau di penyedia cloud lainnya. Dengan AD yang dikelola sendiri, Anda menggunakan autentikasi NTLM untuk mengontrol autentikasi pengguna dan layanan secara langsung di instans DB RDS for SQL Server Anda tanpa menggunakan domain perantara dan forest trust. Saat pengguna mengautentikasi dengan instans DB RDS for SQL Server yang digabungkan ke domain AD yang dikelola sendiri, permintaan autentikasi diteruskan ke domain AD yang dikelola sendiri yang Anda tentukan.

Topik

Ketersediaan wilayah dan versi

Amazon RDS mendukung AD yang Dikelola Sendiri untuk SQL Server menggunakan NTLM di semua Wilayah AWS.

Persyaratan

Pastikan Anda telah memenuhi persyaratan berikut sebelum menggabungkan instans DB RDS for SQL Server ke domain AD yang dikelola sendiri.

Topik

- Konfigurasikan konektivitas jaringan Anda
- Konfigurasikan akun layanan domain AD Anda

Konfigurasikan AD on-premise Anda

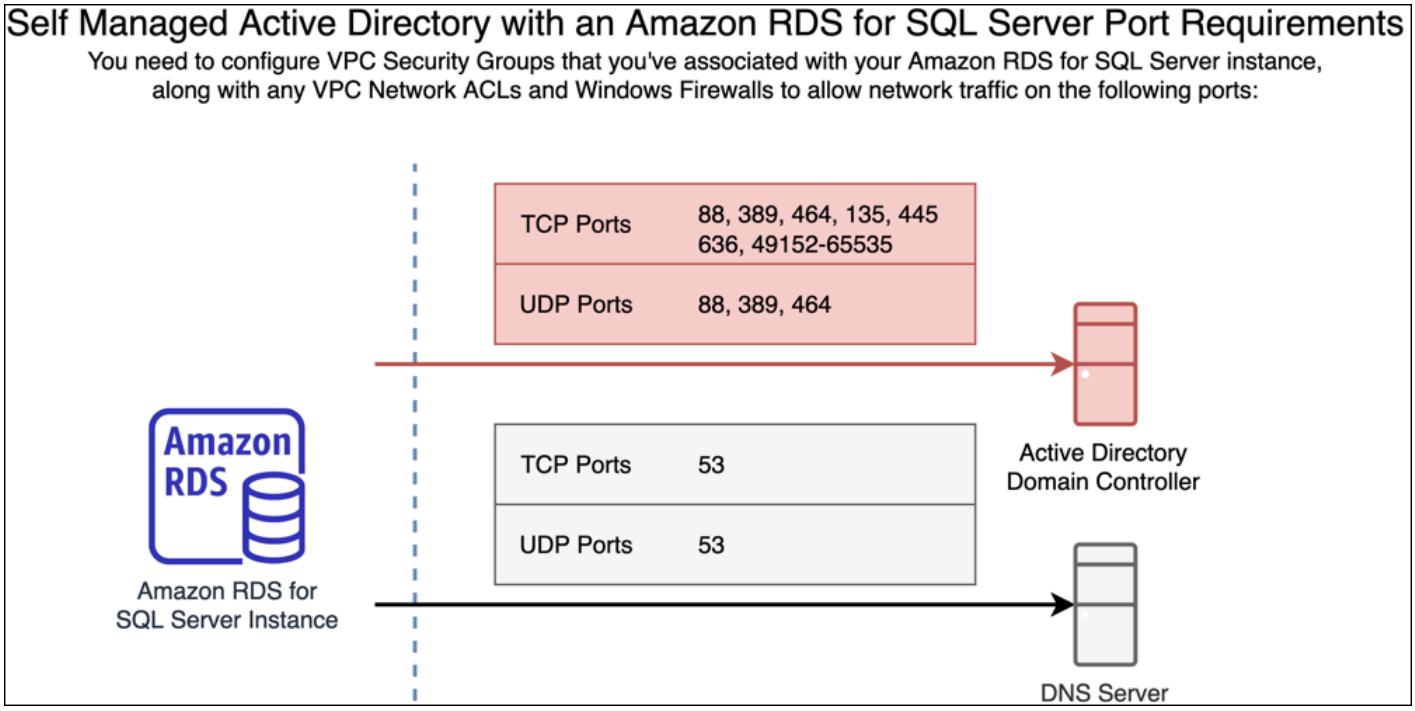
Pastikan bahwa Anda memiliki Microsoft AD on-premise atau yang dikelola sendiri lainnya tempat Anda dapat menggabungkan instans Amazon RDS for SQL Server. AD on-premise Anda harus memiliki konfigurasi berikut:

- Jika Anda telah menentukan situs Directory Active, pastikan subnet di VPC yang terkait dengan instans DB RDS for SQL Server Anda ditentukan di situs Directory Active Anda. Konfirmasikan bahwa tidak ada konflik antara subnet di VPC Anda dan subnet di situs AD Anda yang lain.
- Pengontrol domain AD Anda memiliki tingkat fungsional domain Windows Server 2008 R2 atau lebih tinggi.
- Nama domain AD Anda tidak dapat menggunakan format Domain Label Tunggal (SLD). RDS for SQL Server tidak mendukung domain SLD.
- Nama domain yang sepenuhnya memenuhi syarat (FQDN) dan unit organisasi (OU) untuk AD Anda tidak dapat melebihi 64 karakter.

Konfigurasikan konektivitas jaringan Anda

Pastikan bahwa Anda telah memenuhi konfigurasi jaringan berikut:

- Konektivitas yang dikonfigurasi antara Amazon VPC tempat Anda ingin membuat instans DB RDS for SQL Server dan Directory Active yang dikelola sendiri. Anda dapat mengatur konektivitas menggunakan AWS Direct Connect, AWS VPN, peering VPC, atau AWS Transit Gateway.
- Untuk grup keamanan VPC, grup keamanan default untuk Amazon VPC default Anda sudah ditambahkan ke instans DB RDS for SQL Server Anda di konsol. Pastikan bahwa grup keamanan dan ACL jaringan VPC untuk subnet tempat Anda membuat instans DB RDS for SQL Server Anda mengizinkan lalu lintas pada port dan dengan arah yang ditunjukkan dalam diagram berikut.



Tabel berikut mengidentifikasi peran masing-masing port.

Protokol	Port	Peran
TCP/UDP	53	Sistem Nama Domain (DNS)
TCP/UDP	88	Autentikasi Kerberos
TCP/UDP	464	Ubah/Atur kata sandi
TCP/UDP	389	Protokol Akses Direktori Ringan (LDAP)
TCP	135	Lingkungan Komputasi Terdistribusi/Pemeta Titik Akhir (DCE/EPMAP)
TCP	445	Pembagian file SMB Layanan Direktori

Protokol	Port	Peran
TCP	636	Protokol Akses Direktori Ringan melalui TLS/SSL (LDAPS)
TCP	49152 - 65535	Port efemeral untuk RPC

- Umumnya, server DNS domain terletak di pengontrol domain AD. Anda tidak perlu mengonfigurasi opsi DHCP VPC yang diatur untuk menggunakan fitur ini. Untuk informasi selengkapnya, lihat Set opsi DHCP dalam Panduan Pengguna Amazon VPC.

> **Important**
>
> Jika Anda menggunakan ACL jaringan VPC, Anda juga harus mengizinkan lalu lintas keluar pada port dinamis (49152-65535) dari instans DB RDS for SQL Server Anda. Pastikan bahwa aturan lalu lintas ini juga dicerminkan pada firewall yang berlaku untuk masing-masing pengontrol domain AD, server DNS, dan instans DB RDS for SQL Server.
> Meskipun grup keamanan Amazon VPC mengharuskan port terbuka hanya pada arah ketika lalu lintas jaringan diinisiasi, sebagian besar firewall Windows dan ACL jaringan VPC mengharuskan port terbuka pada kedua arah.

Konfigurasikan akun layanan domain AD Anda

Pastikan bahwa Anda telah memenuhi persyaratan berikut untuk akun layanan domain AD:

- Pastikan bahwa Anda memiliki akun layanan di domain AD yang dikelola sendiri dengan izin delegasi untuk menggabungkan komputer ke domain. Akun layanan domain adalah akun pengguna di AD yang dikelola sendiri yang telah mendapatkan delegasi izin untuk melakukan tugas tertentu.
- Akun layanan domain perlu mendapatkan delegasi izin berikut di Unit Organisasi (OU) tempat Anda menggabungkan instans DB RDS for SQL Server Anda:
  - Kemampuan tervalidasi untuk menulis ke nama host DNS
  - Kemampuan tervalidasi untuk menulis ke nama prinsipal layanan
  - Membuat dan menghapus objek komputer

Hal ini merepresentasikan serangkaian izin minimum yang diperlukan untuk menggabungkan objek komputer ke Directory Active yang dikelola sendiri milik Anda. Untuk informasi selengkapnya, lihat Errors when attempting to join computers to a domain dalam dokumentasi Microsoft Windows Server.

> **Important**
>
> Jangan memindahkan objek komputer yang dibuat RDS for SQL Server di Unit Organisasi setelah instans DB Anda dibuat. Memindahkan objek terkait akan menyebabkan instans DB RDS for SQL Server Anda salah konfigurasi. Jika Anda perlu memindahkan objek komputer yang dibuat oleh Amazon RDS, gunakan operasi ModifyDBInstance API RDS untuk memodifikasi parameter domain dengan lokasi objek komputer yang diinginkan.

Batasan

Batasan berikut berlaku untuk AD yang Dikelola Sendiri untuk SQL Server.

- NTLM adalah satu-satunya jenis autentikasi yang didukung. Autentikasi Kerberos tidak didukung. Jika Anda perlu menggunakan autentikasi Kerberos, Anda dapat menggunakan AWS Managed AD, bukan AD yang dikelola sendiri.
- Layanan Microsoft Distributed Transaction Coordinator (MSDTC) tidak didukung karena memerlukan autentikasi Kerberos.
- Instans DB RDS for SQL Server Anda tidak menggunakan server Protokol Waktu Jaringan (NTP) dari domain AD yang dikelola sendiri. Instans ini menggunakan layanan AWS NTP sebagai gantinya.
- Server tertaut SQL Server harus menggunakan autentikasi SQL untuk terhubung ke RDS lain untuk instans DB SQL Server yang digabungkan ke domain AD yang dikelola sendiri.
- Pengaturan Objek Kebijakan Grup (GPO) Microsoft dari domain AD yang dikelola sendiri tidak diterapkan ke instans DB RDS for SQL Server.

Gambaran umum pengaturan Directory Active yang Dikelola Sendiri

Untuk mengatur AD yang dikelola sendiri untuk instans DB RDS for SQL Server, lakukan langkah-langkah berikut, yang dijelaskan secara lebih terperinci dalam Menyiapkan Directory Active yang Dikelola Sendiri:

Di domain AD Anda:

- Buat Unit Organisasi (OU).
- Buat pengguna domain AD.
- Delegasikan kontrol ke pengguna domain AD.

Dari AWS Management Console atau API:

- Buat kunci AWS KMS.
- Buat rahasia menggunakan AWS Secrets Manager.
- Buat atau modifikasi instans DB RDS for SQL Server dan gabungkan ke domain AD yang dikelola sendiri.

Menyiapkan Directory Active yang Dikelola Sendiri

Untuk mengatur AD yang Dikelola Sendiri, lakukan langkah-langkah berikut.

Topik

Langkah 1: Buat Unit Organisasi di AD Anda

> **Important**
>
> Sebaiknya buat kredensial OU dan layanan khusus yang ditujukan untuk OU tersebut untuk akun AWS apa pun yang memiliki instans DB RDS for SQL Server yang digabungkan dengan domain AD yang dikelola sendiri. Dengan mengkhususkan kredensial OU dan layanan, Anda dapat menghindari izin yang bertentangan dan mengikuti prinsip hak akses paling rendah.

Untuk membuat OU di AD Anda

1. Hubungkan ke domain AD Anda sebagai administrator domain.
2. Buka Active Directory Users and Computers dan pilih domain tempat Anda ingin membuat OU Anda.
3. Klik kanan domain dan pilih New, lalu Organizational Unit.
4. Masukkan nama untuk OU.
5. Biarkan kotak tetap dicentang untuk Protect container from accidental deletion.
6. Klik OK. OU baru Anda akan muncul di bawah domain Anda.

Langkah 2: Buat pengguna domain AD di AD Anda

Kredensial pengguna domain akan digunakan untuk rahasia di AWS Secrets Manager.

Untuk membuat pengguna domain AD di AD Anda

1. Buka Active Directory Users and Computers dan pilih domain dan OU tempat Anda ingin membuat pengguna Anda.
2. Klik kanan objek Users dan pilih New, lalu User.
3. Masukkan nama depan, nama belakang, dan nama login untuk pengguna. Klik Next.
4. Masukkan kata sandi untuk pengguna. Jangan pilih "User must change password at next login". Jangan pilih "Account is disabled". Klik Next.
5. Klik OK. Pengguna baru Anda akan muncul di bawah domain Anda.

Langkah 3: Delegasikan kontrol ke pengguna AD

Untuk mendelegasikan kontrol ke pengguna domain AD di domain Anda

1. Buka snap-in MMC Active Directory Users and Computers dan pilih domain tempat Anda ingin membuat pengguna Anda.
2. Klik kanan OU yang Anda buat sebelumnya dan pilih Delegate Control.
3. Pada bagian Delegation of Control Wizard, klik Next.
4. Pada bagian Users or Groups, klik Add.
5. Pada bagian Select Users, Computers, or Groups, masukkan pengguna AD yang Anda buat lalu klik Check Names. Jika pemeriksaan pengguna AD Anda berhasil, klik OK.
6. Pada bagian Users or Groups, konfirmasikan bahwa pengguna AD Anda telah ditambahkan lalu klik Next.
7. Pada bagian Tasks to Delegate, pilih Create a custom task to delegate lalu klik Next.
8. Pada bagian Active Directory Object Type:
    a. Pilih Only the following objects in the folder.
    b. Pilih Computer Objects.
    c. Pilih Create selected objects in this folder.
    d. Pilih Delete selected objects in this folder lalu klik Next.
9. Pada bagian Permissions:
    a. Tetap pilih General.
    b. Pilih Validated write to DNS host name.
    c. Pilih Validated write to service principal name lalu klik Next.
10. Untuk Completing the Delegation of Control Wizard, tinjau dan konfirmasikan pengaturan Anda lalu klik Finish.

Langkah 4: Buat kunci AWS KMS

Kunci KMS digunakan untuk mengenkripsi rahasia AWS Anda.

Untuk membuat kunci AWS KMS

> **Note**
>
> Untuk Kunci Enkripsi, jangan gunakan kunci KMS default AWS. Pastikan untuk membuat kunci AWS KMS di akun AWS yang sama yang berisi instans DB RDS for SQL Server yang ingin Anda gabungkan ke AD yang dikelola sendiri.

1. Di konsol AWS KMS, pilih Buat kunci.
2. Untuk Tipe Kunci, pilih Simetris.
3. Untuk Penggunaan Kunci, pilih Enkripsi dan dekripsi.
4. Untuk Opsi lanjutan:
   a. Untuk Asal materi kunci, pilih KMS.
   b. Untuk Regionalitas, pilih Kunci Wilayah Tunggal lalu klik Berikutnya.
5. Untuk Alias, berikan nama untuk kunci KMS.
6. (Opsional) Untuk Deskripsi, berikan deskripsi kunci KMS.
7. (Opsional) Untuk Tag, berikan tag kunci KMS lalu klik Berikutnya.
8. Untuk Administrator kunci, berikan nama pengguna IAM dan pilih nama pengguna tersebut.
9. Untuk Penghapusan kunci, biarkan kotak dipilih untuk Izinkan administrator kunci untuk menghapus kunci ini lalu klik Berikutnya.
10. Untuk Pengguna kunci, berikan pengguna IAM yang sama dari langkah sebelumnya dan pilih nama pengguna tersebut. Klik Berikutnya.
11. Tinjau konfigurasi tersebut.
12. Untuk Kebijakan kunci, sertakan yang berikut pada Pernyataan kebijakan:

```
{
    "Sid": "Allow use of the KMS key on behalf of RDS",
    "Effect": "Allow",
    "Principal": {
        "Service": [
            "rds.amazonaws.com"
        ]
    },
    "Action": "kms:Decrypt",
    "Resource": "*"
```

```
}
```

13. Klik Selesai.

Langkah 5: Buat rahasia AWS

Untuk membuat rahasia

> **Note**
>
> Pastikan untuk membuat rahasia di akun AWS yang sama yang berisi instans DB RDS for SQL Server yang ingin Anda gabungkan ke AD yang dikelola sendiri.

1. Di AWS Secrets Manager, pilih Simpan rahasia baru.
2. Untuk Tipe rahasia, pilih Tipe rahasia lainnya.
3. Untuk Pasangan kunci/nilai, tambahkan dua kunci Anda:
   a. Untuk kunci pertama, masukkan `CUSTOMER_MANAGED_ACTIVE_DIRECTORY_USERNAME`.
   b. Untuk nilai kunci pertama, masukkan nama pengguna AD yang Anda buat di domain Anda pada langkah sebelumnya.
   c. Untuk kunci kedua, masukkan `CUSTOMER_MANAGED_ACTIVE_DIRECTORY_PASSWORD`.
   d. Untuk nilai kunci kedua, masukkan kata sandi yang Anda buat untuk pengguna AD di domain Anda.
4. Untuk Kunci enkripsi, masukkan kunci KMS yang Anda buat pada langkah sebelumnya lalu klik Berikutnya.
5. Untuk Nama rahasia, masukkan nama deskriptif yang akan membantu Anda menemukan rahasia Anda nanti.
6. (Opsional) Untuk Deskripsi, masukkan deskripsi untuk nama rahasia.
7. Untuk Izin sumber daya, klik Edit.
8. Tambahkan kebijakan berikut ke kebijakan izin:

> **Note**
>
> Kami menyarankan Anda menggunakan kondisi `aws:sourceAccount` dan `aws:sourceArn` dalam kebijakan untuk menghindari masalah confused deputy.

Gunakan Akun AWS untuk `aws:sourceAccount` dan ARN instans DB RDS for SQL Server untuk `aws:sourceArn`. Untuk informasi selengkapnya, lihat Pencegahan masalah confused deputy lintas layanan.

```
{
    "Version": "2012-10-17",
    "Statement":
    [
        {
            "Effect": "Allow",
            "Principal":
            {
                "Service": "rds.amazonaws.com"
            },
            "Action": "secretsmanager:GetSecretValue",
            "Resource": "*",
            "Condition":
            {
                "StringEquals":
                {
                    "aws:sourceAccount": "123456789012"
                },
                "ArnLike":
                {
                    "aws:sourceArn": "arn:aws:rds:us-west-2:123456789012:db:*"
                }
            }
        }
    ]
}
```

9. Klik Simpan lalu klik Berikutnya.
10. Untuk Konfigurasikan pengaturan rotasi, pertahankan nilai default dan pilih Berikutnya.
11. Tinjau pengaturan untuk rahasia lalu klik Simpan.
12. Pilih rahasia yang Anda buat dan salin nilai ARN Rahasia. Nilai ini akan digunakan pada langkah berikutnya untuk mengatur Directory Active yang dikelola sendiri.

Langkah 6: Buat atau ubah instans DB SQL Server

Anda dapat menggunakan konsol, CLI, atau API RDS untuk SQL Server dengan domain AD yang dikelola sendiri. Anda dapat melakukannya dengan salah satu cara berikut:

- Buat instance SQL Server DB baru menggunakan konsol, perintah create-db-instanceCLI, atau operasi CreateDBInstance RDS API.

  Untuk petunjuk, lihat Membuat instans DB Amazon RDS.
- Ubah instance SQL Server DB yang ada menggunakan konsol, perintah modify-db-instanceCLI, atau operasi ModifyDBInstance RDS API.

  Untuk petunjuk, lihat Memodifikasi instans DB Amazon RDS.
- Kembalikan instance SQL Server DB dari snapshot DB menggunakan konsol, perintah CLI restore-db-instance-from-db-snapshot, atau operasi RestoreDB DBSnapshot RDS API. InstanceFrom

  Untuk petunjuk, lihat Memulihkan dari snapshot DB.
- Kembalikan instance SQL Server DB ke point-in-time menggunakan konsol, perintah restore-db-instance-to- point-in-time CLI, atau operasi API RDS InstanceToPointInTimeRestoreDB.

  Untuk petunjuk, lihat Memulihkan instans DB dengan waktu yang ditentukan.

Saat Anda menggunakan AWS CLI, parameter berikut diperlukan untuk instans DB agar dapat menggunakan domain Directory Active yang dikelola sendiri yang sudah Anda buat:

- Untuk parameter `--domain-fqdn`, gunakan nama domain yang sepenuhnya memenuhi syarat (FQDN) dari Directory Active Anda yang dikelola sendiri.
- Untuk parameter `--domain-ou`, gunakan OU yang Anda buat di AD yang dikelola sendiri.
- Untuk parameter `--domain-auth-secret-arn`, gunakan nilai ARN Rahasia yang Anda buat pada langkah sebelumnya.
- Untuk parameter `--domain-dns-ips`, gunakan alamat IPv4 primer dan sekunder dari server DNS untuk AD yang dikelola sendiri. Jika Anda tidak memiliki alamat IP server DNS sekunder, masukkan alamat IP primer dua kali.

Contoh perintah CLI berikut menunjukkan cara membuat, memodifikasi, dan menghapus instans DB RDS for SQL Server dengan domain AD yang dikelola sendiri.

> **Important**
>
> Jika Anda memodifikasi instans DB untuk menggabungkannya ke atau menghapusnya dari domain AD yang dikelola sendiri, boot ulang instans DB tersebut diperlukan agar modifikasi diterapkan. Anda dapat memilih untuk segera menerapkan perubahan atau menunggu hingga periode pemeliharaan berikutnya. Memilih opsi Terapkan Segera akan menyebabkan waktu henti untuk instans DB AZ Tunggal. Instans DB Multi-AZ akan melakukan failover sebelum menyelesaikan boot ulang. Untuk informasi selengkapnya, lihat Menggunakan pengaturan Terapkan Segera.

Perintah CLI berikut membuat RDS baru untuk instans DB SQL Server dan menggabungkannya ke domain AD yang dikelola sendiri.

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-instance \
    --db-instance-identifier my-DB-instance \
    --db-instance-class db.m5.xlarge \
    --allocated-storage 50 \
    --engine sqlserver-se \
    --engine-version 15.00.4043.16.v1 \
    --license-model license-included \
    --master-username my-master-username \
    --master-user-password my-master-password \
    --domain-fqdn my_AD_domain.my_AD.my_domain \
    --domain-ou OU=my-AD-test-OU,DC=my-AD-test,DC=my-AD,DC=my-domain \
    --domain-auth-secret-arn "arn:aws:secretsmanager:region:account-number:secret:my-
AD-test-secret-123456" \
    --domain-dns-ips "10.11.12.13" "10.11.12.14"
```

Untuk Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier my-DB-instance ^
    --db-instance-class db.m5.xlarge ^
    --allocated-storage 50 ^
    --engine sqlserver-se ^
    --engine-version 15.00.4043.16.v1 ^
    --license-model license-included ^
    --master-username my-master-username ^
```

```
    --master-user-password my-master-password ^
    --domain-fqdn my-AD-test.my-AD.mydomain ^
    --domain-ou OU=my-AD-test-OU,DC=my-AD-test,DC=my-AD,DC=my-domain ^
    --domain-auth-secret-arn "arn:aws:secretsmanager:region:account-number:secret:my-
AD-test-secret-123456" \ ^
    --domain-dns-ips "10.11.12.13" "10.11.12.14"
```

Perintah CLI berikut memodifikasi RDS yang ada untuk instans DB SQL Server untuk menggunakan domain Active Directory yang dikelola sendiri.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-DB-instance \
    --domain-fqdn my_AD_domain.my_AD.my_domain \
    --domain-ou OU=my-AD-test-OU,DC=my-AD-test,DC=my-AD,DC=my-domain \
    --domain-auth-secret-arn "arn:aws:secretsmanager:region:account-number:secret:my-
AD-test-secret-123456" \
    --domain-dns-ips "10.11.12.13" "10.11.12.14"
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-DBinstance ^
    --domain-fqdn my_AD_domain.my_AD.my_domain ^
    --domain-ou OU=my-AD-test-OU,DC=my-AD-test,DC=my-AD,DC=my-domain ^
    --domain-auth-secret-arn "arn:aws:secretsmanager:region:account-number:secret:my-
AD-test-secret-123456" ^
    --domain-dns-ips "10.11.12.13" "10.11.12.14"
```

Perintah CLI berikut menghapus DB RDS for SQL Server instans dari domain Active Directory yang dikelola sendiri.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-DB-instance \
    --disable-domain
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-DB-instance ^
    --disable-domain
```

Langkah 7: Buat login SQL Server Autentikasi Windows

Gunakan kredensial pengguna master Amazon RDS untuk terhubung ke instans basis data SQL Server sebagaimana Anda lakukan dengan instans DB lain. Karena instans DB digabungkan ke domain AD yang dikelola sendiri, Anda dapat menyediakan login dan pengguna SQL Server. Anda melakukannya dari utilitas pengguna dan grup AD di domain AD yang dikelola sendiri. Izin basis data dikelola melalui izin SQL Server standar yang diberikan dan dicabut ke login Windows ini.

Agar pengguna AD yang dikelola sendiri dapat mengautentikasi dengan SQL Server, login Windows SQL Server harus ada untuk pengguna AD yang dikelola sendiri tersebut atau grup Directory Active yang dikelola sendiri tempat pengguna tersebut menjadi anggota. Kontrol akses fine-grained akan ditangani melalui pemberian dan pencabutan izin pada login SQL Server ini. Pengguna AD yang dikelola sendiri yang tidak memiliki login SQL Server atau yang tidak termasuk dalam grup AD yang dikelola sendiri dengan login tersebut tidak akan dapat mengakses instans DB SQL Server.

Izin ALTER ANY LOGIN diperlukan untuk membuat login SQL Server AD yang dikelola sendiri. Jika Anda belum membuat login apa pun dengan izin ini, hubungkan sebagai pengguna master instans DB menggunakan Autentikasi SQL Server dan buat login SQL Server AD yang dikelola sendiri dalam konteks pengguna master.

Anda dapat menjalankan perintah bahasa definisi data (DDL) berikut untuk membuat login SQL Server untuk pengguna atau grup AD yang dikelola sendiri.

> **Note**
>
> Tentukan pengguna dan grup yang menggunakan nama login pra-Windows 2000 dalam format *my_AD_domain**my_AD_domain_user*. Anda tidak dapat menggunakan nama prinsipal pengguna (UPN) dalam format *my_AD_domain_user*@*my_AD_domain*.

```
USE [master]
GO
CREATE LOGIN [my_AD_domain\my_AD_domain_user] FROM WINDOWS WITH DEFAULT_DATABASE =
 [master], DEFAULT_LANGUAGE = [us_english];
```

```
GO
```

Untuk informasi selengkapnya, lihat CREATE LOGIN (Transact-SQL) dalam dokumentasi Microsoft Developer Network.

Pengguna (baik manusia maupun aplikasi) dari domain Anda kini dapat terhubung ke instans RDS for SQL Server dari mesin klien yang tergabung dengan domain AD yang dikelola sendiri menggunakan autentikasi Windows.

Mengelola instans DB dalam Domain Directory Active yang dikelola sendiri

Anda dapat menggunakan konsol AWS CLI, atau API Amazon RDS untuk mengelola instans DB Anda dan relasinya dengan domain AD yang dikelola sendiri. Misalnya, Anda dapat memindahkan instans DB ke dalam, ke luar dari, atau antar-domain.

Misalnya, menggunakan API Amazon RDS, Anda dapat melakukan hal berikut:

- Untuk mencoba kembali bergabung dengan domain yang dikelola sendiri untuk keanggotaan yang gagal, gunakan operasi API ModifyDBInstance dan tentukan serangkaian parameter yang sama:
  - `--domain-fqdn`
  - `--domain-dns-ips`
  - `--domain-ou`
  - `--domain-auth-secret-arn`
- Untuk menghapus instans DB dari domain yang dikelola sendiri, gunakan operasi API `ModifyDBInstance` dan tentukan `--disable-domain` untuk parameter domain.
- Untuk memindahkan instans DB dari satu domain yang dikelola sendiri ke yang lain, gunakan operasi API `ModifyDBInstance` dan tentukan parameter domain untuk domain baru:
  - `--domain-fqdn`
  - `--domain-dns-ips`
  - `--domain-ou`
  - `--domain-auth-secret-arn`
- Untuk menampilkan daftar keanggotaan domain AD yang dikelola sendiri untuk setiap instans DB, gunakan operasi API DescribeDBInstances.

Memahami keanggotaan Domain Directory Active yang dikelola sendiri

Setelah Anda membuat atau memodifikasi instans DB, instans ini akan menjadi anggota domain AD yang dikelola sendiri. Konsol AWS menunjukkan status keanggotaan domain Directory Active yang dikelola sendiri untuk instans DB. Status instans DB dapat berupa salah satu dari daftar berikut:

- joined – Instans adalah anggota domain AD.
- joining – Instans sedang dalam proses untuk menjadi anggota domain.
- pending-join – Keanggotaan instans tertunda.
- pending-maintenance-join— AWS akan mencoba menjadikan instance sebagai anggota domain AD selama jendela pemeliharaan terjadwal berikutnya.
- pending-removal – Penghapusan instans dari domain tertunda.
- pending-maintenance-removal— AWS akan mencoba menghapus instance dari domain AD selama jendela pemeliharaan terjadwal berikutnya.
- failed – Masalah konfigurasi telah mencegah instans bergabung dengan domain AD. Periksa dan perbaiki konfigurasi Anda sebelum menerbitkan ulang perintah modifikasi instans.
- removing – Instans sedang dalam proses untuk dihapus dari domain AD.

Permintaan untuk menjadi anggota domain AD yang dikelola sendiri dapat gagal karena masalah konektivitas jaringan. Misalnya, Anda dapat membuat instans DB atau memodifikasi instans yang sudah ada dan mengalami kegagalan saat mencoba menjadikan instans DB sebagai anggota suatu domain AD yang dikelola sendiri. Dalam hal ini, terbitkan ulang perintah untuk membuat atau memodifikasi instans DB atau modifikasi instans yang baru dibuat untuk digabungkan ke domain AD yang dikelola sendiri.

Pemecahan masalah Directory Active yang dikelola sendiri

Berikut ini adalah masalah yang mungkin Anda hadapi saat menyiapkan atau memodifikasi AD yang dikelola sendiri.

Kode Kesalahan	Deskripsi	Penyebab umum	Saran pemecahan masalah
Error 2 / 0x2	Sistem tidak dapat	Format atau lokasi untuk Unit Organisasi (OU) yang ditentukan dengan	Tinjau parameter – `domain-ou` . Pastikan akun layanan domain

Kode Kesalahan	Deskripsi	Penyebab umum	Saran pemecahan masalah
	menemukan file yang ditentukan.	parameter –domain-ou tidak valid. Akun layanan domain yang ditentukan melalui AWS Secrets Manager tidak memiliki izin yang diperlukan untuk bergabung dengan OU.	memiliki izin yang benar ke OU. Untuk informasi selengkapnya, lihat Konfigurasikan akun layanan domain AD Anda.
Error 5 / 0x5	Akses ditolak.	Izin yang salah dikonfigurasi untuk akun layanan domain, atau akun komputer sudah ada di domain.	Tinjau izin akun layanan domain di domain, dan verifikasi bahwa akun komputer RDS tidak diduplikasi dalam domain. Anda dapat memverifikasi nama akun komputer RDS dengan menjalankan `SELECT @@SERVERNAME` di instans DB RDS for SQL Server Anda. Jika Anda menggunakan Multi-AZ, coba boot ulang dengan failover lalu verifikasi akun komputer RDS tersebut lagi. Untuk informasi selengkapnya, lihat Mem-boot ulang instans DB.
Error 87 / 0x57	Parameter salah.	Akun layanan domain yang ditentukan melalui AWS Secrets Manager tidak memiliki izin yang benar. Profil pengguna juga mungkin rusak.	Tinjau persyaratan untuk akun layanan domain. Untuk informasi selengkapnya, lihat Konfigurasikan akun layanan domain AD Anda.

Kode Kesalahan	Deskripsi	Penyebab umum	Saran pemecahan masalah
Error 234 / 0xEA	Unit Organisasi (OU) yang ditentukan tidak ada.	OU yang ditentukan dengan parameter `–domain-ou` tidak ada di AD yang dikelola sendiri.	Tinjau parameter `–domain-ou` dan pastikan OU yang ditentukan ada di AD yang dikelola sendiri.
Error 1326 / 0x52E	Nama pengguna atau kata sandi salah.	Kredensial akun layanan domain yang disediakan di AWS Secrets Manager berisi nama pengguna yang tidak dikenal atau kata sandi yang buruk. Akun domain mungkin juga dinonaktifkan di AD yang dikelola sendiri.	Pastikan kredensial yang disediakan di AWS Secrets Manager sudah benar dan akun domain diaktifkan di Active Directory yang dikelola sendiri.
Error 1355 / 0x54B	Domain yang ditentukan tidak ada atau tidak dapat dihubungi.	Domain sedang nonaktif, set IP DNS yang ditentukan tidak dapat dijangkau, atau FQDN yang ditentukan tidak dapat dijangkau.	Tinjau parameter `–domain-dns-ips` dan `–domain-fqdn` untuk memastikannya sudah benar. Tinjau konfigurasi jaringan instans DB RDS for SQL Server Anda dan pastikan AD yang dikelola sendiri dapat dijangkau. Untuk informasi selengkapnya, lihat Konfigurasikan konektivitas jaringan Anda.

Kode Kesalahan	Deskripsi	Penyebab umum	Saran pemecahan masalah
Kesalahan 1722/0x6BA	Server RPC tidak tersedia.	Ada masalah saat menjangkau layanan RPC domain AD Anda. Hal ini mungkin merupakan masalah layanan atau jaringan.	Validasikan bahwa layanan RPC berjalan pada pengontrol domain Anda dan port TCP `135` dan `49152-65535` dapat dijangkau di domain Anda dari instans DB RDS for SQL Server Anda.
Error 2224 / 0x8B0	Akun pengguna sudah ada.	Akun komputer yang mencoba agar ditambahkan ke AD yang dikelola sendiri sudah ada.	Identifikasi akun komputer dengan menjalankan `SELECT @@SERVERNAME` di instans DB SQL RDS for Server Anda lalu hapus dengan hati-hati dari AD yang dikelola sendiri.
Error 2242 / 0x8c2	Kata sandi pengguna ini telah kedaluwarsa.	Kata sandi untuk akun layanan domain yang ditentukan melalui AWS Secrets Manager telah kedaluwarsa.	Perbarui kata sandi untuk akun layanan domain yang digunakan untuk menggabungkan instans DB RDS for SQL Server Anda ke AD yang dikelola sendiri.

Memulihkan instans DB SQL Server lalu menambahkannya ke domain Directory Active yang dikelola sendiri

Anda dapat memulihkan snapshot DB atau melakukan point-in-time pemulihan (PITR) untuk instance SQL Server DB dan kemudian menambahkannya ke domain Active Directory yang dikelola sendiri. Setelah instans DB dipulihkan, modifikasi instans ini menggunakan proses yang dijelaskan dalam

Langkah 6: Buat atau ubah instans DB SQL Server untuk menambahkan instans DB ke domain AD yang dikelola sendiri.

Menggunakan AWS Managed Active Directory dengan RDS for SQL Server

Sekarang Anda dapat menggunakan AWS Managed Microsoft AD untuk mengautentikasi pengguna dengan Autentikasi Windows saat mereka terhubung ke instans DB RDS for SQL Server. Instans DB dapat digunakan dengan AWS Directory Service for Microsoft Active Directory, yang juga disebut AWS Managed Microsoft AD, untuk mengaktifkan Autentikasi Windows. Ketika pengguna mengautentikasi dengan instans DB SQL Server yang tergabung ke domain tepercaya, permintaan autentikasi diteruskan ke direktori domain yang Anda buat dengan AWS Directory Service.

Ketersediaan wilayah dan versi

Amazon RDS mendukung penggunaan hanya AWS Managed Microsoft AD untuk Autentikasi Windows. RDS tidak mendukung penggunaan AD Connector. Untuk informasi selengkapnya tentang IAM, lihat hal berikut:

- Kebijakan kompatibilitas aplikasi untuk AWS Managed Microsoft AD
- Kebijakan kompatibilitas aplikasi untuk AD Connector

Untuk informasi tentang ketersediaan versi dan Wilayah, lihat Autentikasi Kerberos dengan RDS for SQL Server.

Gambaran umum pengaturan autentikasi Windows

Amazon RDS menggunakan mode campuran untuk Autentikasi Windows. Pendekatan ini berarti pengguna master (nama dan kata sandi yang digunakan untuk membuat instans DB SQL Server) akan menggunakan Autentikasi SQL. Karena akun pengguna master adalah kredensial istimewa, Anda harus membatasi akses ke akun ini.

Untuk mendapatkan Autentikasi Windows menggunakan Microsoft Active Directory on-premise atau yang di-host sendiri, buat sebuah forest trust. Trust dapat bersifat satu atau dua arah. Untuk informasi selengkapnya tentang menyiapkan kepercayaan forest menggunakanAWS Directory Service, lihat Kapan membuat relasi kepercayaan dalam Panduan Administrasi AWS Directory Service.

Untuk mengatur autentikasi Windows untuk instans DB SQL Server, lakukan langkah-langkah berikut, yang dijelaskan secara lebih terperinci dalam Mengatur Autentikasi Windows untuk instans DB SQL Server:

1. Gunakan AWS Managed Microsoft AD, baik dari AWS Management Console atau API AWS Directory Service, untuk membuat direktori AWS Managed Microsoft AD.

2. Jika Anda menggunakan AWS CLI atau API Amazon RDS untuk membuat instans DB SQL Server Anda, buat peran AWS Identity and Access Management (IAM). Peran ini menggunakan kebijakan IAM terkelola `AmazonRDSDirectoryServiceAccess` dan memungkinkan Amazon RDS melakukan panggilan ke direktori Anda. Jika Anda menggunakan konsol untuk membuat instans DB SQL Server, AWS akan membuat peran IAM untuk Anda.

   Agar peran ini mengizinkan akses, titik akhir AWS Security Token Service (AWS STS) harus diaktifkan di Wilayah AWS untuk akun AWS Anda. Titik akhir AWS STS aktif secara default di semua Wilayah AWS, dan Anda dapat menggunakannya tanpa tindakan lebih lanjut. Untuk informasi selengkapnya, lihat Mengelola AWS STS di Wilayah AWS dalam Panduan Pengguna IAM.

3. Buat dan konfigurasikan pengguna dan grup dalam direktori AWS Managed Microsoft AD menggunakan alat Microsoft Directory Active. Untuk informasi selengkapnya tentang membuat pengguna dan grup di Active Directory Anda, lihat Kelola pengguna dan grup di AWS Managed Microsoft AD dalam Panduan Administrasi AWS Directory Service.

4. Jika Anda ingin menempatkan direktori dan instans DB di VPC yang berbeda, aktifkan lalu lintas antar-VPC.

5. Gunakan Amazon RDS untuk membuat instans DB SQL Server baru dari konsol, AWS CLI, atau API Amazon RDS. Dalam permintaan pembuatan, Anda perlu menyediakan pengidentifikasi domain (pengidentifikasi "`d-*`") yang dihasilkan saat Anda membuat direktori dan nama peran yang Anda buat. Anda juga dapat memodifikasi instans DB SQL Server yang sudah ada untuk menggunakan Autentikasi Windows dengan mengatur domain dan parameter peran IAM untuk instans DB.

6. Gunakan kredensial pengguna master Amazon RDS untuk terhubung ke instans basis data SQL Server sebagaimana Anda lakukan dengan instans DB lain. Karena instans DB digabungkan ke domain AWS Managed Microsoft AD, Anda dapat menyediakan login dan pengguna SQL Server dari pengguna dan grup Directory Active dalam domainnya. (Hal ini dikenal sebagai login “Windows” SQL Server.) Izin basis data dikelola melalui izin SQL Server standar yang diberikan dan dicabut ke login Windows ini.

Membuat titik akhir untuk autentikasi Kerberos

Autentikasi berbasis Kerberos mengharuskan bahwa titik akhir terdiri dari nama host yang ditentukan pelanggan, titik, lalu nama domain yang sepenuhnya memenuhi syarat (FQDN). Misalnya, berikut ini adalah contoh titik akhir yang mungkin Anda gunakan dengan autentikasi berbasis Kerberos. Dalam

contoh ini, nama host instans DB SQL Server adalah `ad-test` dan nama domainnya adalah `corp-ad.company.com`.

```
ad-test.corp-ad.company.com
```

Jika Anda ingin memastikan koneksi Anda menggunakan Kerberos, jalankan kueri berikut:

```
SELECT net_transport, auth_scheme
  FROM sys.dm_exec_connections
 WHERE session_id = @@SPID;
```

Mengatur Autentikasi Windows untuk instans DB SQL Server

Anda menggunakan AWS Directory Service for Microsoft Active Directory, yang juga disebut AWS Managed Microsoft AD, untuk mengatur Autentikasi Windows untuk instans DB SQL Server. Untuk mengatur Autentikasi Windows, lakukan langkah-langkah berikut.

Langkah 1: Buat direktori menggunakan AWS Directory Service for Microsoft Active Directory

AWS Directory Service membuat Microsoft Directory Active yang dikelola sepenuhnya di AWS Cloud. Saat Anda membuat direktori AWS Managed Microsoft AD, AWS Directory Service membuat dua pengontrol domain dan server Layanan Nama Domain (DNS) atas nama Anda. Server direktori dibuat dalam dua subnet di dua Zona Ketersediaan yang berbeda dalam sebuah VPC. Redundansi ini membantu memastikan bahwa direktori Anda tetap dapat diakses bahkan jika terjadi kegagalan.

Saat Anda membuat direktori AWS Managed Microsoft AD, AWS Directory Service melakukan tugas berikut ini atas nama Anda:

- Mengatur Microsoft Active Directory dengan VPC.
- Membuat akun administrator direktori dengan Admin nama pengguna dan kata sandi yang ditentukan. Anda menggunakan akun ini untuk mengelola direktori Anda.

  > **Note**
  >
  > Pastikan Anda menyimpan kata sandi ini. AWS Directory Service tidak menyimpan kata sandi ini, dan Anda tidak dapat mengambil atau mengatur ulang kata sandi ini.

- Membuat grup keamanan untuk pengontrol direktori.

Saat Anda meluncurkan sebuah AWS Directory Service for Microsoft Active Directory, AWS membuat Unit Organisasi (OU) yang berisi semua objek direktori Anda. OU ini, yang memiliki nama NetBIOS yang Anda ketik saat membuat direktori Anda, terletak di root domain. Root domain dimiliki dan dikelola oleh AWS.

Akun admin yang dibuat dengan direktori AWS Managed Microsoft AD Anda memiliki izin untuk aktivitas administratif paling umum untuk OU Anda:

- Membuat, memperbarui, atau menghapus pengguna, grup, dan komputer.
- Menambahkan sumber daya ke domain Anda seperti server file atau cetak, lalu memberikan izin untuk sumber daya tersebut ke pengguna dan grup di OU Anda.
- Membuat OU dan kontainer tambahan.
- Mendelegasikan otoritas.
- Membuat dan menautkan kebijakan grup.
- Memulihkan objek yang dihapus dari Keranjang Sampah Directory Active.
- Jalankan PowerShell modul AD dan DNS Windows pada Layanan Web Direktori Aktif.

Akun admin juga memiliki hak untuk melakukan aktivitas di seluruh domain berikut:

- Mengelola konfigurasi DNS (menambahkan, menghapus, atau memperbarui catatan, zona, dan penerus).
- Melihat log peristiwa DNS.
- Melihat log peristiwa keamanan.

Untuk membuat direktori dengan AWS Managed Microsoft AD

1. Di panel navigasi konsol AWS Directory Service, pilih Direktori, lalu pilih Siapkan direktori.
2. Pilih AWS Managed Microsoft AD. Ini adalah satu-satunya opsi yang saat ini didukung untuk digunakan dengan Amazon RDS.
3. Pilih Selanjutnya.
4. Di halaman Masukkan informasi direktori, berikan informasi berikut:

   Edisi

   Pilih edisi sesuai kebutuhan Anda.

Nama DNS direktori

Nama yang sepenuhnya memenuhi syarat untuk direktori, seperti `corp.example.com`. Nama yang lebih panjang dari 47 karakter tidak didukung oleh SQL Server.

Nama NetBIOS direktori

Nama pendek opsional untuk direktori, seperti `CORP`.

Deskripsi direktori

Deskripsi opsional untuk direktori.

Kata sandi admin

Kata sandi administrator direktori. Proses pembuatan direktori akan membuat akun administrator dengan nama pengguna Admin dan kata sandi ini.

Kata sandi administrator direktori tidak dapat menyertakan kata `admin`. Kata sandi peka huruf besar/kecil dan harus memiliki panjang 8–64 karakter. Kata sandi juga harus berisi minimal satu karakter dalam tiga dari empat kategori berikut:

- Huruf kecil (a-z)
- Huruf besar (A-Z)
- Angka (0-9)
- Karakter non-alfanumerik (~!@#$%^&*_-+=`|\(){}[]:;'"<>,.?/)

Konfirmasi kata sandi

Ketik ulang kata sandi administrator.

5. Pilih Selanjutnya.
6. Di halaman Pilih VPC dan subnet, berikan informasi berikut:

VPC

Pilih VPC untuk direktori.

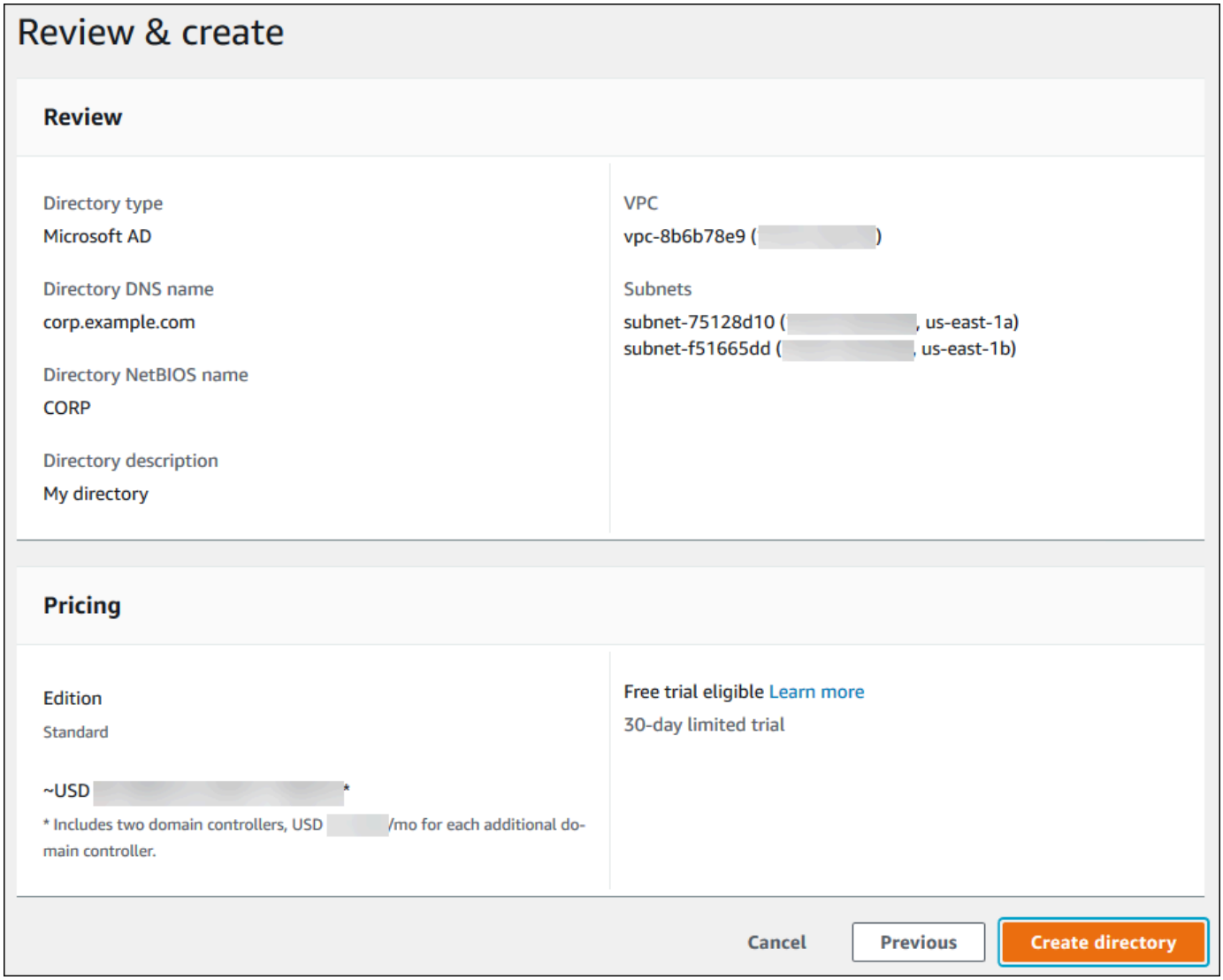
> **Note**
>
> Anda dapat menempatkan direktori dan instans DB di VPC yang berbeda, tetapi jika Anda melakukannya, pastikan untuk mengaktifkan lalu lintas antar-VPC. Untuk

informasi selengkapnya, lihat Langkah 4: Aktifkan lalu lintas antar-VPC antara direktori dan instans DB.

Subnet

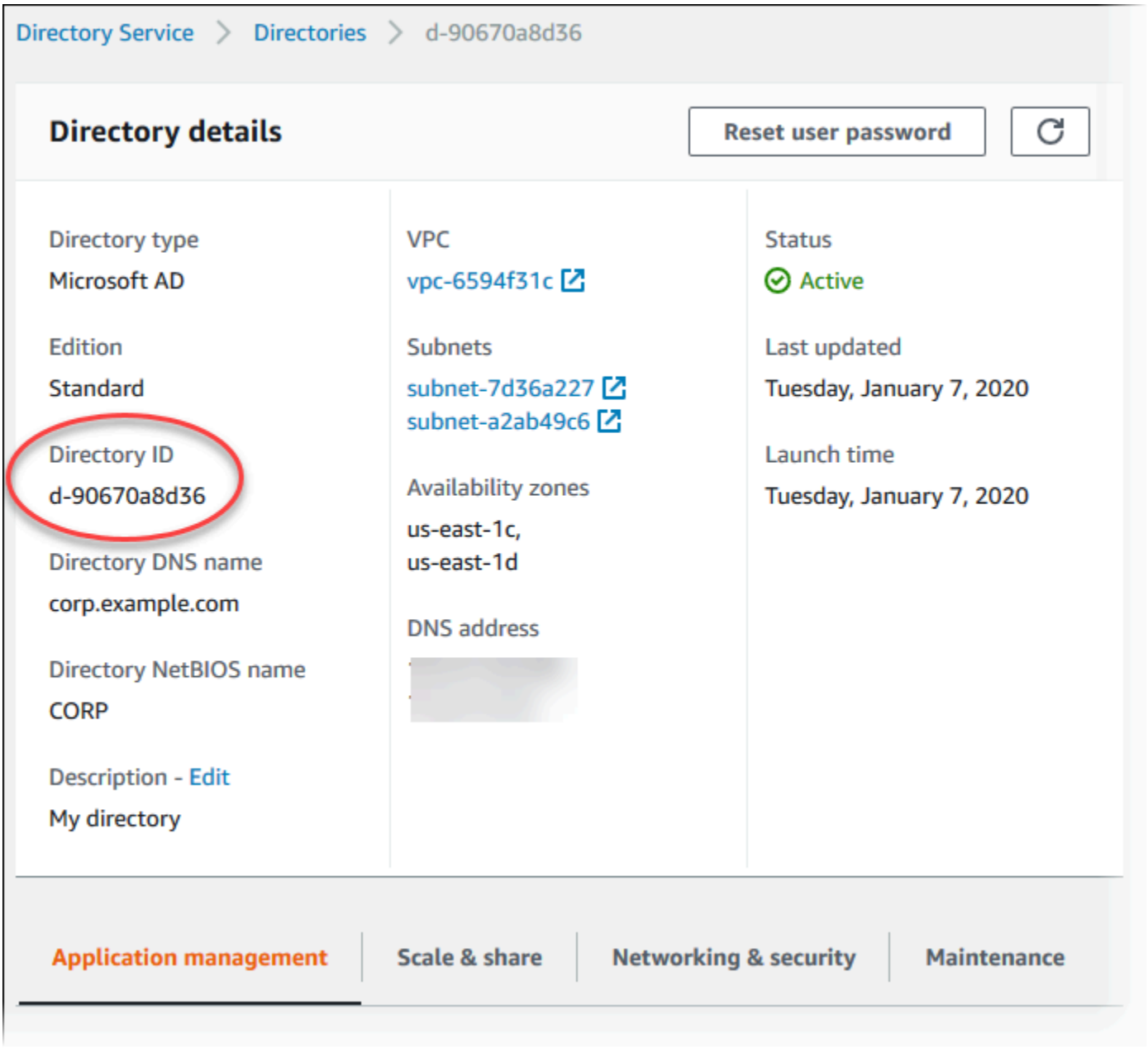
Pilih subnet untuk server direktori. Kedua subnet harus berada di Zona Ketersediaan yang berbeda.

7. Pilih Berikutnya.
8. Tinjau informasi direktori. Jika perubahan diperlukan, pilih Sebelumnya. Jika informasi sudah benar, pilih Buat direktori.



Pembuatan direktori memerlukan waktu beberapa menit. Setelah berhasil dibuat, nilai Status berubah menjadi Aktif.

Untuk melihat informasi tentang direktori Anda, pilih ID direktori di daftar direktori. Catat ID Direktori. Anda memerlukan nilai ini saat membuat atau memodifikasi instans DB SQL Server Anda.



Langkah 2: Buat peran IAM yang akan digunakan oleh Amazon RDS

Jika Anda menggunakan konsol untuk membuat instans DB SQL Server, Anda dapat melewati langkah ini. Jika Anda menggunakan CLI atau API RDS untuk membuat instans DB SQL Server Anda, Anda harus membuat peran IAM yang menggunakan kebijakan IAM terkelola

`AmazonRDSDirectoryServiceAccess`. Peran ini memungkinkan Amazon RDS melakukan panggilan ke AWS Directory Service untuk Anda.

Jika Anda menggunakan kebijakan kustom untuk bergabung dengan domain, alih-alih menggunakan kebijakan `AmazonRDSDirectoryServiceAccess` yang dikelola AWS, pastikan Anda mengizinkan tindakan `ds:GetAuthorizedApplicationDetails`. Persyaratan ini berlaku mulai Juli 2019 karena perubahan di API AWS Directory Service.

Kebijakan IAM berikut, `AmazonRDSDirectoryServiceAccess`, menyediakan akses ke AWS Directory Service.

Example Kebijakan IAM untuk menyediakan akses ke AWS Directory Service

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
            "ds:DescribeDirectories",
            "ds:AuthorizeApplication",
            "ds:UnauthorizeApplication",
            "ds:GetAuthorizedApplicationDetails"
        ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Sebaiknya gunakan kunci konteks kondisi global `aws:SourceArn` dan `aws:SourceAccount` dalam relasi kepercayaan berbasis sumber daya untuk membatasi izin layanan ke sumber daya tertentu. Ini adalah cara paling efektif untuk melindungi dari masalah confused deputy.

Anda dapat menggunakan kedua kunci konteks kondisi global dan memiliki nilai `aws:SourceArn` yang berisi ID akun. Dalam hal ini, nilai `aws:SourceAccount` dan akun dalam nilai `aws:SourceArn` harus menggunakan ID akun yang sama ketika digunakan dalam pernyataan yang sama.

- Gunakan `aws:SourceArn` jika Anda ingin akses lintas layanan untuk satu sumber daya.
- Gunakan `aws:SourceAccount` jika Anda ingin mengizinkan sumber daya apa pun di akun tersebut dikaitkan dengan penggunaan lintas layanan.

Dalam relasi kepercayaan, pastikan untuk menggunakan kunci konteks kondisi global `aws:SourceArn` dengan Amazon Resource Name (ARN) dari sumber daya yang mengakses peran. Untuk Autentikasi Windows, pastikan untuk menyertakan instans DB, seperti yang ditunjukkan pada contoh berikut.

Example relasi kepercayaan dengan kunci konteks kondisi global untuk Autentikasi Windows

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "rds.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceArn": [
                        "arn:aws:rds:Region:my_account_ID:db:db_instance_identifier"
                    ]
                }
            }
        }
    ]
}
```

Buat peran IAM menggunakan kebijakan IAM ini dan relasi kepercayaan. Untuk informasi selengkapnya tentang pembuatan peran IAM, lihat Membuat kebijakan yang dikelola pelanggan dalam Panduan Pengguna IAM.

Langkah 3: Buat dan konfigurasikan pengguna dan grup

Anda dapat membuat pengguna dan grup dengan alat Active Directory Users and Computers. Alat ini merupakan salah satu alat Active Directory Domain Services dan Active Directory Lightweight Directory Services. Pengguna merepresentasikan orang individu atau entitas yang memiliki akses ke direktori Anda. Grup sangat berguna untuk memberikan atau menolak hak akses ke grup pengguna, daripada harus menerapkan hak akses tersebut ke setiap pengguna.

Untuk membuat pengguna dan grup di direktori AWS Directory Service, Anda harus terhubung ke instans EC2 Windows yang merupakan anggota dari direktori AWS Directory Service. Anda juga

harus masuk sebagai pengguna yang memiliki hak akses untuk membuat pengguna dan grup. Untuk informasi selengkapnya, lihat Tambahkan pengguna dan grup (AD Sederhana dan AWS Managed Microsoft AD) dalam Panduan Administrasi AWS Directory Service.

Langkah 4: Aktifkan lalu lintas antar-VPC antara direktori dan instans DB

Jika Anda ingin menempatkan direktori dan instans DB dalam VPC yang sama, lewati langkah ini dan lanjutkan ke Langkah 5: Buat atau ubah instans DB SQL Server.

Jika Anda ingin menempatkan direktori dan instans DB di VPC yang berbeda, konfigurasikan lalu lintas antar-VPC menggunakan peering VPC atau AWS Transit Gateway.

Prosedur berikut mengaktifkan lalu lintas antar-VPC menggunakan peering VPC. Ikuti petunjuk di Apa yang dimaksud dengan peering VPC? dalam Panduan Peering Amazon Virtual Private Cloud.

Untuk mengaktifkan lalu lintas VPC menggunakan peering VPC

1. Siapkan aturan perutean VPC yang sesuai untuk memastikan lalu lintas jaringan dapat berjalan dua arah.
2. Pastikan grup keamanan instans DB dapat menerima lalu lintas masuk dari grup keamanan direktori.
3. Pastikan tidak ada aturan daftar kontrol akses (ACL) jaringan yang memblokir lalu lintas.

Jika akun AWS lain memiliki direktori, Anda harus berbagi direktori.

Untuk berbagi direktori antara akun AWS

1. Mulai berbagi direktori dengan akun AWS tempat instans DB akan dibuat dengan mengikuti petunjuk dalam Tutorial: Berbagi direktori AWS Managed Microsoft AD untuk gabungan domain EC2 tanpa hambatan dalam Panduan Administrasi AWS Directory Service.
2. Masuk ke konsol AWS Directory Service menggunakan akun untuk instans DB, dan pastikan bahwa domain memiliki status SHARED sebelum melanjutkan.
3. Saat masuk ke konsol AWS Directory Service menggunakan akun untuk instans DB, catat nilai ID Direktori. Anda perlu menggunakan ID direktori ini untuk menggabungkan instans DB ke domain.

Langkah 5: Buat atau ubah instans DB SQL Server

Buat atau ubah instans DB SQL Server untuk digunakan dengan direktori Anda. Anda dapat menggunakan konsol, CLI, atau API RDS untuk mengaitkan suatu instans DB dengan direktori. Anda dapat menyesuaikan waktu ini dengan cara berikut:

- Buat instance SQL Server DB baru menggunakan konsol, perintah create-db-instanceCLI, atau operasi CreateDBInstance RDS API.

  Untuk petunjuk, lihat Membuat instans DB Amazon RDS.
- Ubah instance SQL Server DB yang ada menggunakan konsol, perintah modify-db-instanceCLI, atau operasi ModifyDBInstance RDS API.

  Untuk petunjuk, lihat Memodifikasi instans DB Amazon RDS.
- Kembalikan instance SQL Server DB dari snapshot DB menggunakan konsol, perintah CLI restore-db-instance-from-db-snapshot, atau operasi RestoreDB DBSnapshot RDS API. InstanceFrom

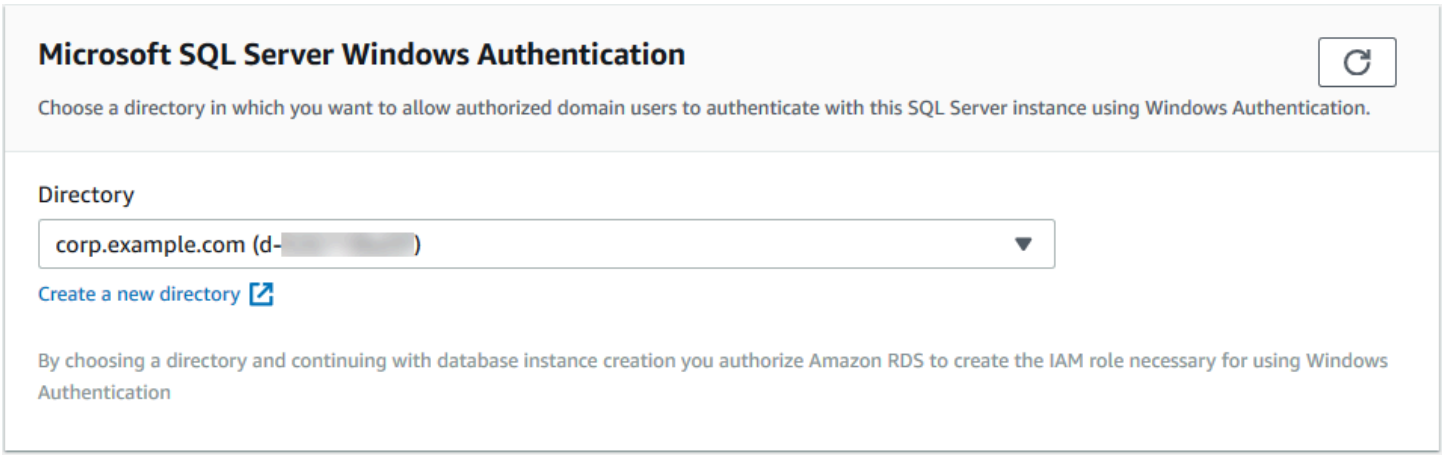
  Untuk petunjuk, lihat Memulihkan dari snapshot DB.
- Kembalikan instance SQL Server DB ke point-in-time menggunakan konsol, perintah restore-db-instance-to- point-in-time CLI, atau operasi API RDS InstanceToPointInTimeRestoreDB.

  Untuk instruksi, lihat Memulihkan instans DB dengan waktu yang ditentukan.

Autentikasi Windows hanya mendukung instans DB SQL Server di VPC.

Agar instans DB dapat menggunakan direktori domain yang Anda buat, hal berikut diperlukan:

- Untuk Direktori, Anda harus memilih pengidentifikasi domain (`d-`*`ID`*) yang dibuat saat Anda membuat direktori.
- Pastikan bahwa grup keamanan VPC memiliki aturan keluar yang memungkinkan instans DB berkomunikasi dengan direktori.



Saat Anda menggunakan AWS CLI, parameter berikut diperlukan agar instans DB dapat menggunakan direktori yang Anda buat:

- Untuk parameter `--domain`, gunakan pengidentifikasi domain (`d-`*`ID`*) yang dihasilkan saat Anda membuat direktori.
- Untuk parameter `--domain-iam-role-name`, gunakan peran yang Anda buat yang menggunakan kebijakan IAM terkelola `AmazonRDSDirectoryServiceAccess`.

Misalnya, perintah CLI berikut memodifikasi instans DB untuk menggunakan direktori.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --domain d-ID \
    --domain-iam-role-name role-name
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --domain d-ID ^
    --domain-iam-role-name role-name
```

> **Important**
>
> Jika Anda memodifikasi instans DB untuk mengaktifkan autentikasi Kerberos, boot ulang instans DB tersebut setelah membuat perubahan.

Langkah 6: Buat login SQL Server Autentikasi Windows

Gunakan kredensial pengguna master Amazon RDS untuk terhubung ke instans basis data SQL Server sebagaimana Anda lakukan dengan instans DB lain. Karena instans DB digabungkan ke domain AWS Managed Microsoft AD, Anda dapat menyediakan login dan pengguna SQL Server. Anda melakukannya dari pengguna dan grup Active Directory di domain Anda. Izin basis data dikelola melalui izin SQL Server standar yang diberikan dan dicabut ke login Windows ini.

Agar pengguna Active Directory dapat mengautentikasi dengan SQL Server, login Windows SQL Server harus ada untuk pengguna tersebut atau grup tempat pengguna tersebut menjadi anggota. Kontrol akses fine-grained akan ditangani melalui pemberian dan pencabutan izin pada login SQL Server ini. Pengguna yang tidak memiliki login SQL Server atau yang tidak termasuk dalam grup dengan login tersebut tidak akan dapat mengakses instans DB SQL Server.

Izin ALTER ANY LOGIN diperlukan untuk membuat login Active Directory SQL Server. Jika Anda belum membuat login apa pun dengan izin ini, hubungkan sebagai pengguna master instans DB menggunakan Autentikasi SQL Server.

Jalankan perintah bahasa definisi data (DDL) berikut untuk membuat login SQL Server untuk pengguna atau grup Directory Active.

> **Note**
>
> Tentukan pengguna dan grup yang menggunakan nama login pra-Windows 2000 dalam format *domainName**login_name*. Anda tidak dapat menggunakan nama prinsipal pengguna (UPN) dalam format *login_name*@*DomainName*.

```
USE [master]
GO
CREATE LOGIN [mydomain\myuser] FROM WINDOWS WITH DEFAULT_DATABASE = [master],
 DEFAULT_LANGUAGE = [us_english];
GO
```

Untuk informasi selengkapnya, lihat CREATE LOGIN (Transact-SQL) dalam dokumentasi Microsoft Developer Network.

Pengguna (baik manusia maupun aplikasi) dari domain Anda kini dapat terhubung ke instans RDS for SQL Server dari mesin klien yang tergabung dengan domain menggunakan autentikasi Windows.

Mengelola instans DB di Domain

Anda dapat menggunakan konsol, AWS CLI, atau API Amazon RDS untuk mengelola instans DB Anda dan relasinya dengan domain Anda. Misalnya, Anda dapat memindahkan instans DB ke dalam, ke luar dari, atau antar-domain.

Misalnya, menggunakan API Amazon RDS, Anda dapat melakukan hal berikut:

- Untuk mencoba kembali bergabung dengan domain untuk keanggotaan yang gagal, gunakan operasi API ModifyDBInstance dan tentukan ID direktori keanggotaan saat ini.
- Untuk memperbarui nama peran IAM untuk keanggotaan, gunakan operasi API `ModifyDBInstance` dan tentukan ID direktori keanggotaan saat ini dan peran IAM baru.
- Untuk menghapus instans DB dari domain, gunakan operasi API `ModifyDBInstance` dan tentukan `none` sebagai parameter domain.
- Untuk memindahkan instans DB dari satu domain ke domain lain, gunakan operasi API `ModifyDBInstance` dan tentukan pengidentifikasi domain baru sebagai parameter domain.
- Untuk menampilkan daftar keanggotaan untuk setiap instans DB, gunakan operasi API DescribeDBInstances.

Memahami keanggotaan Domain

Setelah Anda membuat atau memodifikasi instans DB Anda, instans ini akan menjadi anggota domain. Konsol AWS menunjukkan status keanggotaan domain untuk instans DB. Status instans DB dapat berupa salah satu dari daftar berikut:

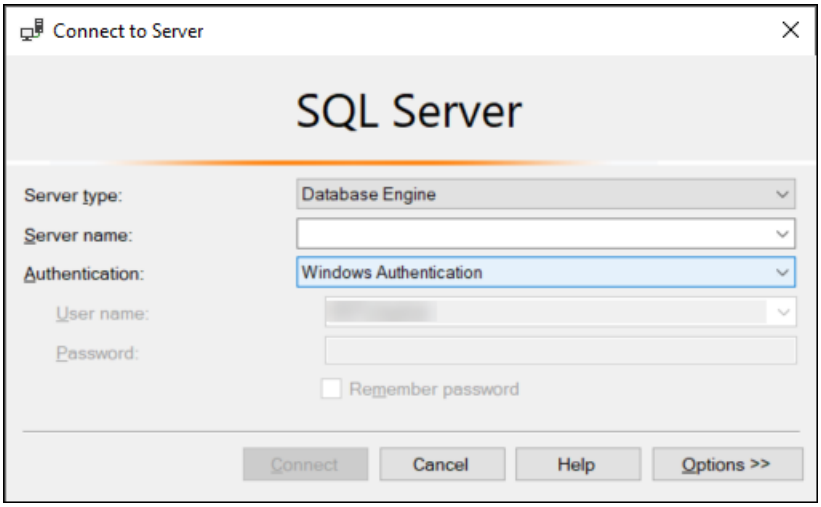
- joined – Instans adalah anggota domain.
- joining – Instans sedang dalam proses untuk menjadi anggota domain.
- pending-join – Keanggotaan instans tertunda.
- pending-maintenance-join— AWS akan mencoba menjadikan instance sebagai anggota domain selama jendela pemeliharaan terjadwal berikutnya.
- pending-removal – Penghapusan instans dari domain tertunda.

- pending-maintenance-removal— AWS akan mencoba untuk menghapus instance dari domain selama jendela pemeliharaan terjadwal berikutnya.
- failed – Masalah konfigurasi telah mencegah instans bergabung dengan domain. Periksa dan perbaiki konfigurasi Anda sebelum menerbitkan ulang perintah modifikasi instans.
- removing – Instans sedang dalam proses untuk dihapus dari domain.

Permintaan untuk menjadi anggota domain dapat gagal karena masalah konektivitas jaringan atau peran IAM yang salah. Misalnya, Anda dapat membuat instans DB atau memodifikasi instans yang sudah ada dan mengalami kegagalan saat mencoba menjadikan instans DB sebagai anggota suatu domain. Dalam hal ini, terbitkan ulang perintah untuk membuat atau memodifikasi instans DB atau modifikasi instans yang baru dibuat untuk digabungkan ke domain.

Menghubungkan ke SQL Server dengan autentikasi Windows

Untuk terhubung ke SQL Server dengan Autentikasi Windows, Anda harus masuk ke komputer yang tergabung dengan domain sebagai pengguna domain. Setelah meluncurkan SQL Server Management Studio, pilih Autentikasi Windows sebagai jenis autentikasi, seperti yang ditunjukkan berikut ini.



Memulihkan instans DB SQL Server lalu menambahkannya ke domain

Anda dapat mengembalikan snapshot DB atau melakukan point-in-time pemulihan (PITR) untuk instance SQL Server DB dan kemudian menambahkannya ke domain. Setelah instans DB dipulihkan, modifikasi instans ini menggunakan proses yang dijelaskan dalam Langkah 5: Buat atau ubah instans DB SQL Server untuk menambahkan instans DB ke domain.

Memperbarui aplikasi untuk terhubung ke instans DB Microsoft SQL Server menggunakan sertifikat SSL/TLS baru

Sejak 13 Januari 2023, Amazon RDS telah menerbitkan serifikat Otoritas Sertifikat (CA) baru untuk terhubung ke instans DB RDS menggunakan Lapisan Soket Aman atau Keamanan Lapisan Pengangkutan (SSL/TLS). Setelah itu, Anda dapat menemukan informasi tentang pembaruan aplikasi untuk menggunakan sertifikat baru.

Topik ini dapat membantu Anda menentukan apakah aplikasi klien menggunakan SSL/TLS untuk terhubung ke instans DB Anda. Jika demikian, Anda dapat memeriksa lebih lanjut apakah aplikasi tersebut memerlukan verifikasi sertifikat untuk terhubung.

> **Note**
>
> Beberapa aplikasi dikonfigurasi untuk terhubung ke instans DB SQL Server hanya jika aplikasi tersebut berhasil memverifikasi sertifikat di server.
> Untuk aplikasi tersebut, Anda harus memperbarui penyimpanan kepercayaan aplikasi klien untuk menyertakan sertifikat CA baru.

Setelah Anda memperbarui sertifikat CA di penyimpanan kepercayaan aplikasi klien, Anda dapat merotasi sertifikat di instans DB Anda. Sebaiknya Anda menguji prosedur ini di lingkungan pengembangan dan penahapan sebelum menerapkannya di lingkungan produksi Anda.

Untuk informasi selengkapnya tentang rotasi sertifikat, lihat Merotasi sertifikat SSL/TLS. Untuk informasi selengkapnya tentang cara mengunduh sertifikat, lihat . Untuk informasi tetang menggunakan SSL/TLS dengan instans DB Microsoft SQL Server, lihat Menggunakan SSL dengan instans DB Microsoft SQL Server.

Topik

- Menentukan apakah ada aplikasi yang terhubung ke instans DB Microsoft SQL Server Anda menggunakan SSL
- Menentukan apakah klien memerlukan verifikasi sertifikat agar dapat terhubung
- Memperbarui penyimpanan kepercayaan aplikasi Anda

Menentukan apakah ada aplikasi yang terhubung ke instans DB Microsoft SQL Server Anda menggunakan SSL

Periksa konfigurasi instans DB untuk nilai parameter `rds.force_ssl`. Secara default, parameter `rds.force_ssl` diatur ke 0 (nonaktif). Jika parameter `rds.force_ssl` diatur ke 1 (aktif), klien harus menggunakan SSL/TLS untuk koneksi. Lihat informasi lebih lanjut tentang grup parameter di Bekerja dengan grup parameter.

Jalankan kueri berikut untuk mendapatkan opsi enkripsi saat ini untuk semua koneksi terbuka ke instans DB. Kolom `ENCRYPT_OPTION` pengembalian `TRUE` jika koneksi dienkripsi.

```
select SESSION_ID,
    ENCRYPT_OPTION,
    NET_TRANSPORT,
    AUTH_SCHEME
    from SYS.DM_EXEC_CONNECTIONS
```

Kueri ini hanya menunjukkan koneksi saat ini. Itu tidak menunjukkan apakah aplikasi yang telah terhubung dan terputus di masa lalu telah menggunakan SSL.

Menentukan apakah klien memerlukan verifikasi sertifikat agar dapat terhubung

Anda dapat memeriksa apakah jenis klien yang berbeda memerlukan verifikasi sertifikat untuk terhubung.

> **Note**
>
> Jika Anda menggunakan konektor selain yang tercantum di atas, lihat dokumentasi konektor spesifik untuk informasi tentang cara menerapkan koneksi terenkripsi. Untuk informasi selengkapnya, lihat Modul koneksi untuk basis data Micro SQL di dokumentasi Micro SQL Server.

SQL Server Management Studio

Periksa apakah enkripsi diberlakukan untuk koneksi SQL Server Management Studio:

1. Luncurkan SQL Server Management Studio.
2. Untuk Terhubung ke server, masukkan informasi server, nama pengguna login, dan kata sandi.
3. Pilih Opsi.
4. Centang jika Enkripsikan koneksi dipilih di halaman koneksi.

Untuk informasi selengkapnya tentang SQL Server Management Studio, lihat Menggunakan SQL Server Management Studio.

Sqlcmd

Contoh berikut dengan klien `sqlcmd` menunjukkan cara memeriksa koneksi SQL Server skrip untuk menentukan apakah koneksi yang berhasil memerlukan sertifikat yang valid. Untuk informasi selengkapnya, lihat Menghubungkan dengan sqlcmd di dokumentasi Microsoft SQL Server.

Saat menggunakan `sqlcmd`, koneksi SSL memerlukan verifikasi terhadap sertifikat server jika Anda menggunakan perintah `-N` untuk mengenkripsi koneksi, seperti pada contoh berikut.

```
$ sqlcmd -N -S dbinstance.rds.amazon.com -d ExampleDB
```

> **Note**
>
> Jika `sqlcmd` dipanggil dengan opsi `-C`, itu menyetujui sertifikat server, meskipun tidak sesuai dengan penyimpanan kepercayaan klien.

ADO.NET

Dalam contoh berikut, aplikasi terhubung menggunakan SSL, dan sertifikat server harus diverifikasi.

```
using SQLC = Microsoft.Data.SqlClient;

...

    static public void Main()
```

```
    {
        using (var connection = new SQLC.SqlConnection(
            "Server=tcp:dbinstance.rds.amazon.com;" +
            "Database=ExampleDB;User ID=LOGIN_NAME;" +
            "Password=YOUR_PASSWORD;" +
            "Encrypt=True;TrustServerCertificate=False;"
            ))
        {
            connection.Open();
            ...
        }
```

Java

Dalam contoh berikut, aplikasi terhubung menggunakan SSL, dan sertifikat server harus diverifikasi.

```
String connectionUrl =
    "jdbc:sqlserver://dbinstance.rds.amazon.com;" +
    "databaseName=ExampleDB;integratedSecurity=true;" +
    "encrypt=true;trustServerCertificate=false";
```

Untuk mengaktifkan enkripsi SSL bagi klien yang terhubung menggunakan JDBC, Anda mungkin perlu menambahkan sertifikat Amazon RDS ke penyimpanan sertifikat Java CA. Untuk instruksi, lihat Mengonfigurasi klien untuk enkripsi di dokumentasi Micro SQL Server. Anda juga dapat memberikan nama file sertifikat CA tepercaya secara langsung melalui aplikasi `trustStore=`*`path-to-certificate-trust-store-file`* dengan string koneksi.

> **Note**
>
> Jika Anda menggunakan `TrustServerCertificate=true` (atau yang setara) dalam string koneksi, proses koneksi melewati validasi rantai persetujuan. Dalam hal ini, aplikasi akan terkoneksi meskipun sertifikat tidak dapat diverifikasi. Menggunakan `TrustServerCertificate=false` memberlakukan validasi sertifikat dan merupakan praktik terbaik.

Memperbarui penyimpanan kepercayaan aplikasi Anda

Anda dapat memperbarui toko kepercayaan untuk aplikasi yang menggunakan Micro SQL Server. Untuk petunjuknya, lihat Mengenkripsi koneksi spesifik. Lihat juga Mengonfigurasi klien untuk enkripsi di dokumentasi Microsoft SQL Server.

Jika Anda menggunakan sistem operasi selain Micro Windows, lihat dokumentasi distribusi perangkat lunak untuk implementasi SSL/TLS untuk informasi tentang penambahan sertifikat root CA baru. Misalnya, OpenSSL dan GnuTLS merupakan pilihan populer. Gunakan metode implementasi untuk menambahkan kepercayaan pada sertifikat root CA RDS. Micro menyediakan instruksi untuk mengonfigurasi sertifikat pada beberapa sistem.

Untuk informasi tentang cara mengunduh sertifikat root, lihat .

Untuk contoh skrip yang mengimpor sertifikat, lihat Contoh skrip untuk mengimpor sertifikat ke trust store Anda.

> **Note**
>
> Saat memperbarui penyimpanan kepercayaan, Anda dapat mempertahankan sertifikat lama selain menambahkan sertifikat baru.

Meng-upgrade mesin DB Microsoft SQL Server

Ketika Amazon RDS mendukung versi baru mesin basis data, Anda dapat meng-upgrade instans DB Anda ke versi baru. Ada dua jenis upgrade untuk instans DB SQL Server: upgrade versi mayor dan versi minor.

Upgrade versi mayor dapat berisi perubahan basis data yang tidak memiliki kompatibilitas mundur dengan aplikasi yang ada. Oleh karena itu, Anda harus melakukan upgrade versi mayor untuk instans DB Anda secara manual. Anda dapat memulai upgrade versi mayor dengan mengubah instans DB Anda. Namun, sebelum melakukan upgrade versi mayor, kami sarankan agar Anda menguji upgrade tersebut dengan mengikuti langkah-langkah yang dijelaskan dalam Menguji upgrade.

Sebaliknya, upgrade versi minor hanya menyertakan perubahan yang kompatibel dengan aplikasi yang ada. Anda dapat memulai upgrade versi minor secara manual dengan memodifikasi instans DB Anda.

Alternatifnya, Anda dapat mengaktifkan opsi Peningkatan versi minor otomatis saat membuat atau memodifikasi instans DB. Tindakan ini akan membuat instans DB Anda secara otomatis di-upgrade setelah pengujian Amazon RDS dan menyetujui versi baru. Anda dapat mengonfirmasi apakah upgrade versi minor akan bersifat otomatis dengan menggunakan perintah `describe-db-engine-versions` AWS CLI. Sebagai contoh:

```
aws rds describe-db-engine-versions --engine sqlserver-se --engine-version
 14.00.3281.6.v1
```

Dalam contoh berikut, perintah CLI menampilkan respons yang menampilkan `AutoUpgrade` true, sehingga menunjukkan bahwa upgrade bersifat otomatis.

```
...

"ValidUpgradeTarget": [
    {
        "Engine": "sqlserver-se",
        "EngineVersion": "14.00.3281.6.v1",
        "Description": "SQL Server 2017 14.00.3281.6.v1",
        "AutoUpgrade": true,
        "IsMajorVersionUpgrade": false
    }
```

```
...
```

Untuk informasi selengkapnya tentang cara melakukan upgrade, lihat Meng-upgrade instans DB SQL Server. Untuk informasi tentang versi SQL Server yang tersedia di Amazon RDS, lihat Amazon RDS for Microsoft SQL Server.

Topik

- Gambaran umum upgrade
- Upgrade versi mayor
- Pertimbangan Multi-AZ dan optimisasi dalam memori
- Pertimbangan replika baca
- Pertimbangan grup opsi
- Pertimbangan grup parameter
- Menguji upgrade
- Meng-upgrade instans DB SQL Server
- Meng-upgrade instans DB yang ditiadakan sebelum dukungan berakhir

Gambaran umum upgrade

Amazon RDS mengambil dua snapshot DB selama proses upgrade. Snapshot DB pertama adalah snapshot dari instans DB sebelum perubahan upgrade dibuat. Snapshot DB kedua diambil setelah upgrade selesai.

> **Note**
>
> Amazon RDS hanya mengambil snapshot DB jika Anda telah mengatur periode retensi cadangan untuk instans DB Anda ke angka yang lebih besar dari 0. Untuk mengubah periode retensi cadangan Anda, lihat Memodifikasi instans DB Amazon RDS.

Setelah upgrade selesai, Anda tidak dapat kembali ke versi mesin basis data sebelumnya. Jika Anda ingin kembali ke versi yang lebih lama, pulihkan snapshot DB yang diambil sebelum upgrade untuk membuat instans DB baru.

Selama upgrade SQL Server versi minor atau mayor, metrik Ruang Penyimpanan Kosong dan Kedalaman Antrean Disk akan menampilkan `-1`. Setelah upgrade selesai, kedua metrik ini akan kembali normal.

Upgrade versi mayor

Amazon RDS saat ini mendukung upgrade versi mayor berikut ke instans DB Microsoft SQL Server.

Anda dapat meng-upgrade instans DB Anda yang ada ke SQL Server 2017 atau 2019 dari versi apa pun kecuali SQL Server 2008. Untuk meng-upgrade dari SQL Server 2008, upgrade ke salah satu versi lainnya terlebih dahulu.

Versi saat ini	Versi upgrade yang didukung
SQL Server 2019	SQL Server 2022
SQL Server 2017	SQL Server 2022 SQL Server 2019
SQL Server 2016	SQL Server 2022 SQL Server 2019 SQL Server 2017
SQL Server 2014	SQL Server 2022 SQL Server 2019 SQL Server 2017 SQL Server 2016
SQL Server 2012 (akhir dukungan)	SQL Server 2022 SQL Server 2019 SQL Server 2017 SQL Server 2016

Versi saat ini	Versi upgrade yang didukung
	SQL Server 2014
SQL Server 2008 R2 (akhir dukungan)	SQL Server 2016 SQL Server 2014 SQL Server 2012

Anda dapat menggunakan kueri AWS CLI, seperti contoh berikut, untuk menemukan upgrade yang tersedia untuk versi mesin basis data tertentu.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds describe-db-engine-versions \
    --engine sqlserver-se \
    --engine-version 14.00.3281.6.v1 \
    --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" \
    --output table
```

Untuk Windows:

```
aws rds describe-db-engine-versions ^
    --engine sqlserver-se ^
    --engine-version 14.00.3281.6.v1 ^
    --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" ^
    --output table
```

Output ini menunjukkan bahwa Anda dapat meng-upgrade versi 14.00.3281.6 ke versi SQL Server 2017 atau 2019 terbaru yang tersedia.

```
-------------------------
|DescribeDBEngineVersions|
+------------------------+
|      EngineVersion     |
+------------------------+
|  14.00.3294.2.v1       |
```

```
|  14.00.3356.20.v1      |
|  14.00.3381.3.v1       |
|  14.00.3401.7.v1       |
|  14.00.3421.10.v1      |
|  14.00.3451.2.v1       |
|  15.00.4043.16.v1      |
|  15.00.4073.23.v1      |
|  15.00.4153.1.v1       |
|  15.00.4198.2.v1       |
|  15.00.4236.7.v1       |
+------------------------+
```

Tingkat kompatibilitas basis data

Anda dapat menggunakan basis data Microsoft SQL Server untuk menyesuaikan beberapa perilaku basis data guna meniru versi SQL Server yang lebih lama. Untuk informasi selengkapnya, lihat Compatibility level dalam dokumentasi Microsoft.

Saat Anda meng-upgrade instans DB, semua basis data yang ada tetap memiliki tingkat kompatibilitas aslinya. Misalnya, jika Anda meng-upgrade dari SQL Server 2014 ke SQL Server 2016, semua basis data yang ada memiliki tingkat kompatibilitas 120. Setiap basis data baru yang dibuat setelah upgrade memiliki tingkat kompatibilitas 130.

Anda dapat mengubah tingkat kompatibilitas basis data dengan menggunakan perintah ALTER DATABASE. Misalnya, untuk mengubah basis data yang bernama `customeracct` agar kompatibel dengan SQL Server 2014, berikan perintah berikut:

```
ALTER DATABASE customeracct SET COMPATIBILITY_LEVEL = 120
```

Pertimbangan Multi-AZ dan optimisasi dalam memori

Amazon RDS mendukung deployment Multi-AZ untuk instans DB yang menjalankan Microsoft SQL Server menggunakan Pencerminan Basis Data (DBM) atau Grup Ketersediaan (AG) Selalu Aktif SQL Server. Untuk informasi selengkapnya, lihat Deployment Multi-AZ untuk Amazon RDS for Microsoft SQL Server.

Jika instans DB Anda berada dalam deployment Multi-AZ, instans primer dan siaga akan di-upgrade. Amazon RDS melakukan upgrade bergulir. Anda akan mengalami pemadaman hanya selama durasi failover.

SQL Server 2014 hingga Enterprise Edition 2019 mendukung optimisasi dalam memori.

Pertimbangan replika baca

Selama upgrade versi basis data, Amazon RDS meng-upgrade semua replika baca Anda bersama instans DB primer. Amazon RDS tidak mendukung upgrade versi basis data pada replika baca secara terpisah. Untuk informasi selengkapnya tentang replika baca, lihat Menggunakan replika baca untuk Microsoft SQL Server di Amazon RDS.

Saat Anda melakukan upgrade versi basis data instans DB primer, semua replika baca juga di-upgrade secara otomatis. Amazon RDS akan meng-upgrade semua replika baca secara bersamaan sebelum meng-upgrade instans DB primer. Replika baca mungkin tidak tersedia sampai upgrade versi basis data pada instans DB primer selesai.

Pertimbangan grup opsi

Jika instans DB Anda menggunakan grup opsi kustom, dalam beberapa kasus, Amazon RDS tidak dapat secara otomatis menetapkan grup opsi baru untuk DB Anda. Misalnya, saat Anda meng-upgrade ke versi mayor baru, Anda harus menentukan grup opsi baru. Kami sarankan agar Anda membuat grup opsi baru, dan menambahkan opsi yang sama ke grup opsi kustom yang ada.

Untuk informasi selengkapnya, lihat Membuat grup opsi atau Menyalin grup opsi.

Pertimbangan grup parameter

Jika instans DB Anda menggunakan grup parameter DB kustom:

- Amazon RDS secara otomatis mem-boot ulang instans DB setelah upgrade.
- Dalam beberapa kasus, RDS tidak dapat secara otomatis menetapkan grup parameter baru ke instans DB Anda.

  Misalnya, saat Anda meng-upgrade ke versi mayor baru, Anda harus menentukan grup parameter baru. Kami menyarankan agar Anda membuat grup parameter baru, dan mengonfigurasi parameter seperti dalam grup parameter kustom yang ada.

Untuk informasi selengkapnya, lihat Membuat grup parameter DB atau Menyalin grup parameter DB.

Menguji upgrade

Sebelum melakukan upgrade versi mayor pada instans DB, Anda harus menguji kompatibilitas basis data Anda secara menyeluruh, dan semua aplikasi yang mengakses basis data, dengan versi baru. Kami menyarankan agar Anda menggunakan prosedur berikut.

Untuk menguji upgrade versi mayor

1. Tinjau Upgrade SQL Server dalam dokumentasi Microsoft untuk versi baru mesin basis data untuk melihat apakah ada masalah kompatibilitas yang mungkin memengaruhi basis data atau aplikasi Anda.
2. Jika instans DB Anda menggunakan grup opsi kustom, buat grup opsi baru yang kompatibel dengan versi baru yang menjadi target upgrade Anda. Untuk informasi selengkapnya, lihat Pertimbangan grup opsi.
3. Jika instans DB Anda menggunakan grup parameter kustom, buat grup parameter baru yang kompatibel dengan versi baru yang menjadi target upgrade Anda. Untuk informasi selengkapnya, lihat Pertimbangan grup parameter.
4. Buat snapshot DB dari instans DB yang akan di-upgrade. Untuk informasi selengkapnya, lihat Membuat snapshot DB untuk instans DB Single-AZ.
5. Pulihkan snapshot DB untuk membuat instans DB uji baru. Untuk informasi selengkapnya, lihat Memulihkan dari snapshot DB.
6. Modifikasi instans DB uji baru ini untuk di-upgrade ke versi baru menggunakan salah satu metode berikut:
   - Konsol
   - AWS CLI
   - API RDS
7. Evaluasi penyimpanan yang digunakan oleh instans yang di-upgrade untuk menentukan apakah upgrade memerlukan penyimpanan tambahan.
8. Jalankan pengujian jaminan kualitas terhadap instans DB yang di-upgrade sebanyak yang diperlukan untuk memastikan bahwa basis data dan aplikasi Anda berfungsi baik dengan versi baru. Terapkan setiap pengujian baru yang diperlukan untuk mengevaluasi dampak dari masalah kompatibilitas yang Anda identifikasi pada langkah 1. Uji semua prosedur tersimpan dan fungsi. Arahkan versi pengujian aplikasi Anda ke instans DB yang di-upgrade.

9. Jika semua pengujian berhasil, maka lakukan upgrade pada instans DB produksi Anda. Kami menyarankan agar Anda tidak mengizinkan operasi tulis ke instans DB hingga Anda mengonfirmasi bahwa semuanya berfungsi dengan benar.

Meng-upgrade instans DB SQL Server

Untuk informasi tentang upgrade instans DB SQL Server secara manual atau otomatis, lihat hal berikut:

- Meng-upgrade versi mesin instans DB
- Praktik terbaik untuk meng-upgrade SQL Server 2008 R2 ke SQL Server 2016 di Amazon RDS for SQL Server

> **Important**
>
> Jika Anda memiliki snapshot yang dienkripsi menggunakan AWS KMS, kami menyarankan agar Anda memulai upgrade sebelum dukungan berakhir.

Meng-upgrade instans DB yang ditiadakan sebelum dukungan berakhir

Setelah sebuah versi mayor ditiadakan, Anda tidak dapat menginstal versi tersebut di instans DB baru. RDS akan mencoba meng-upgrade semua instans DB yang ada secara otomatis.

Jika Anda perlu memulihkan instans DB yang tidak digunakan lagi, Anda dapat melakukan point-in-time pemulihan (PITR) atau memulihkan snapshot. Tindakan ini dapat memberikan akses sementara ke instans DB yang menggunakan versi yang sudah ditiadakan. Namun, setelah sebuah versi mayor sepenuhnya ditiadakan, instans DB ini juga akan secara otomatis di-upgrade ke versi yang didukung.

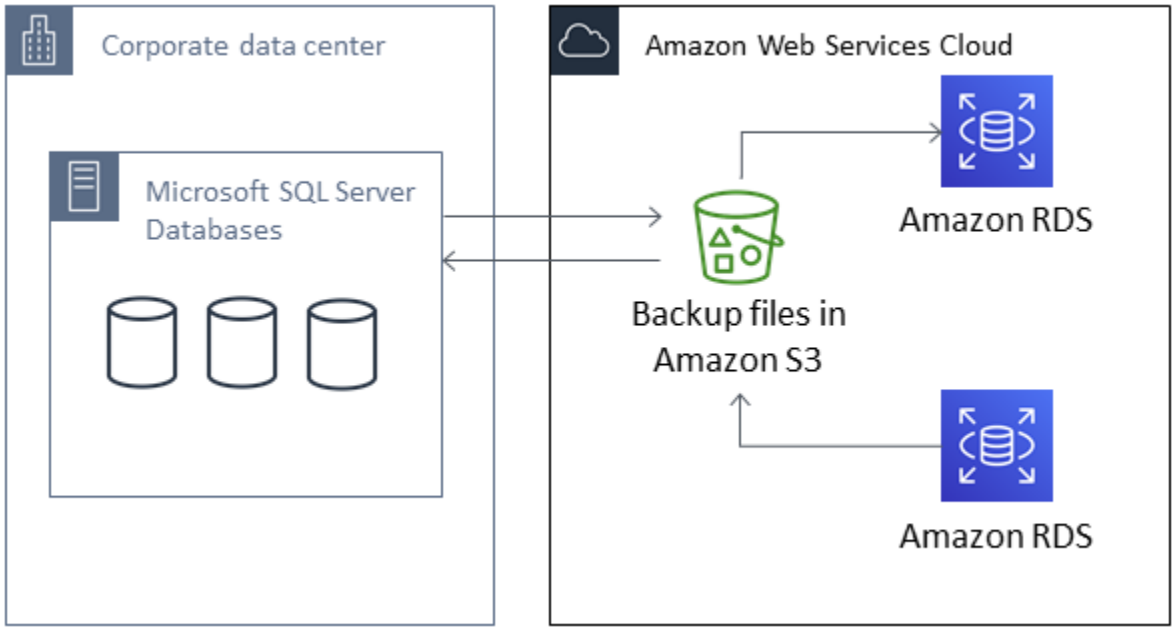
Mengimpor dan mengekspor basis data SQL Server menggunakan pencadangan dan pemulihan native

Amazon RDS mendukung pencadangan dan pemulihan native untuk basis data Microsoft SQL Server menggunakan file cadangan penuh (file .bak). Saat Anda menggunakan RDS, Anda mengakses file yang disimpan di Amazon S3 bukannya menggunakan sistem file lokal di server basis data.

Misalnya, Anda dapat membuat backup penuh dari server lokal Anda, menyimpannya di S3, lalu memulihkannya ke instans DB Amazon RDS yang ada. Anda juga dapat membuat backup dari RDS, menyimpannya di S3, lalu memulihkannya kemana pun Anda inginkan.

Pencadangan dan pemulihan native tersedia di semua AWS Wilayah untuk instans DB Multi-AZ dan Single-AZ, termasuk instans DB Multi-AZ dengan replika baca. Pencadangan dan pemulihan native tersedia untuk semua edisi Microsoft SQL Server yang didukung di Amazon RDS.

Diagram berikut menunjukkan skenario yang didukung.



Menggunakan file .bak native untuk mencadangkan dan memulihkan basis data biasanya merupakan cara tercepat untuk mencadangkan dan memulihkan basis data. Ada banyak keuntungan tambahan jika menggunakan pencadangan dan pemulihan native. Misalnya, Anda dapat melakukan hal berikut:

- Memigrasi basis data ke atau dari Amazon RDS.
- Memigrasi basis data antara beberapa instans DB RDS untuk SQL Server.

- Memigrasi data, skema, prosedur tersimpan, pemicu, dan kode basis data lainnya di dalam file .bak.
- Mencadangkan dan memulihkan basis data tunggal, bukan seluruh instans DB.
- Membuat salinan basis data untuk pengembangan, pengujian, pelatihan, dan demonstrasi.
- Menyimpan dan mentransfer file backup dengan Amazon S3, untuk perlindungan tambahan bagi pemulihan bencana.
- Buat backup native dari basis data yang telah mengaktifkan Enkripsi Data Transparan (TDE), dan pulihkan backup tersebut ke basis data on-premise. Untuk informasi selengkapnya, lihat Dukungan untuk Enkripsi Data Transparan di SQL Server.
- Kembalikan cadangan native basis data on-premise yang mengaktifkan TDE ke RDS untuk instans SQL Server DB. Untuk informasi selengkapnya, lihat Dukungan untuk Enkripsi Data Transparan di SQL Server.

Daftar Isi

Batasan dan rekomendasi

Berikut ini adalah beberapa keterbatasan dalam menggunakan pencadangan dan pemulihan native:

- Anda tidak dapat mencadangkan ke, atau memulihkan dari, bucket Amazon S3 di Wilayah AWS yang berbeda dari instans DB Amazon RDS Anda.
- Anda tidak dapat memulihkan basis data dengan nama yang sama seperti basis data yang sudah ada. Nama basis data bersifat unik.
- Kami sangat menyarankan agar Anda tidak memulihkan cadangan dari satu zona waktu ke zona waktu yang berbeda. Jika Anda memulihkan cadangan dari satu zona waktu ke zona waktu yang berbeda, Anda harus mengaudit kueri dan aplikasi Anda untuk mengetahui efek dari perubahan zona waktu.
- Amazon S3 memiliki batas ukuran 5 TB per file. Untuk pencadangan native pada basis data yang lebih besar, Anda dapat menggunakan pencadangan multi-file.

- Ukuran basis data maksimum yang dapat dicadangkan ke S3 bergantung pada memori, CPU, I/O, dan sumber daya jaringan yang tersedia pada instans DB. Semakin besar basis data, semakin banyak memori yang dikonsumsi agen pencadangan. Pengujian kami menunjukkan bahwa Anda dapat membuat backup terkompresi dari basis data berukuran 16-TB pada jenis instans generasi terbaru kami dari ukuran instans `2xlarge` dan lebih besar, dengan sumber daya sistem yang memadai.
- Anda tidak dapat mencadangkan atau memulihkan lebih dari 10 file backup sekaligus.
- Pencadangan diferensial didasarkan pada backup penuh yang terakhir. Agar pencadangan diferensial berhasil, Anda tidak dapat mengambil snapshot antara backup penuh terakhir dan pencadangan diferensial. Jika Anda menginginkan pencadangan diferensial, tetapi ada snapshot manual atau otomatis, maka lakukan backup penuh lainnya sebelum melakukan pencadangan diferensial.
- Pemulihan diferensial dan log diferensial tidak didukung untuk basis data dengan file yang file_guid (pengidentifikasi unik) diatur ke `NULL`.
- Anda dapat menjalankan hingga dua tugas pencadangan atau pemulihan pada saat bersamaan.
- Anda tidak dapat melakukan pencadangan log native dari SQL Server di Amazon RDS.
- RDS mendukung pemulihan native basis data hingga 16 TB. Pemulihan native untuk basis data di Edisi Ekspres SQL Server dibatasi hingga 10 GB.
- Anda tidak dapat melakukan pencadangan native selama waktu pemeliharaan, atau saat Amazon RDS sedang dalam proses mengambil snapshot basis data. Jika tugas pencadangan native tumpang tindih dengan jendela pencadangan harian RDS, tugas pencadangan native akan dibatalkan.
- Pada instans DB Multi-AZ, Anda hanya dapat melakukan pemulihan native di basis data yang dicadangkan di model pemulihan penuh.
- Memulihkan dari pencadangan diferensial pada instans Multi-AZ tidak didukung.
- Memanggil prosedur RDS untuk pencadangan dan pemulihan native dalam transaksi tidak didukung.
- Gunakan enkripsi simetris AWS KMS key untuk mengenkripsi backup Anda. Amazon RDS tidak mendukung kunci KMS asimetris. Untuk informasi selengkapnya, lihat Membuat kunci enkripsi simetris KMS di Panduan Pengembang AWS Key Management Service.
- File pencadangan native dienkripsi dengan kunci KMS yang ditentukan menggunakan mode kripto "Hanya Enkripsi". Saat Anda memulihkan file backup terenkripsi, ingatlah bahwa file tersebut dienkripsi dengan mode kripto "Encryption-Only".
- Anda tidak dapat memulihkan basis data yang berisi kelompok file FILESTREAM.

Jika basis data Anda dapat offline saat file backup dibuat, disalin, dan dipulihkan, kami menyarankan Anda untuk menggunakan pencadangan dan pemulihan native untuk memigrasinya ke RDS.
Jika basis data on-premise Anda tidak dapat offline, kami sarankan untuk menggunakan AWS Database Migration Service untuk memigrasikan basis data Anda ke Amazon RDS. Untuk informasi selengkapnya, lihat Apa yang dimaksud dengan AWS Database Migration Service?

Pencadangan dan pemulihan native tidak dimaksudkan untuk menggantikan kemampuan pemulihan data dari fitur salinan snapshot lintas-wilayah. Kami menyarankan agar Anda menggunakan salinan snapshot untuk menyalin snapshot basis data Anda ke Wilayah AWS lainnya untuk pemulihan bencana lintas wilayah di Amazon RDS. Untuk informasi selengkapnya, lihat Menyalin snapshot DB.

Menyiapkan pencadangan dan pemulihan native

Untuk mengatur pencadangan dan pemulihan native, Anda memerlukan tiga komponen:

1. Bucket Amazon S3 untuk menyimpan file cadangan Anda.

   Anda harus memiliki bucket S3 yang akan digunakan untuk file backup kemudian mengunggah backup yang ingin Anda migrasikan ke RDS. Jika sudah memiliki bucket Amazon S3, Anda dapat menggunakannya. Jika tidak, Anda dapat membuat bucket. Atau, Anda dapat memilih untuk dibuatkan bucket baru untuk Anda saat menambahkan opsi `SQLSERVER_BACKUP_RESTORE` dengan menggunakan AWS Management Console.

   Untuk informasi tentang penggunaan S3, lihat Panduan Pengguna Amazon Simple Storage Service
2. Peran (IAM) AWS Identity and Access Management untuk mengakses bucket.

   Jika sudah memiliki peran IAM, Anda dapat menggunakannya. Anda dapat memilih untuk dibuatkan IAM role baru untuk Anda ketika menambahkan opsi `SQLSERVER_BACKUP_RESTORE` dengan menggunakan AWS Management Console. Atau, Anda dapat membuatnya secara manual.

   Jika Anda ingin membuat peran IAM baru secara manual, gunakan pendekatan yang dibahas di bagian berikutnya. Lakukan hal yang sama jika Anda ingin melampirkan hubungan kepercayaan dan kebijakan perizinan pada peran IAM yang sudah ada.
3. Opsi `SQLSERVER_BACKUP_RESTORE` ditambahkan ke grup opsi di instans DB Anda.

   Untuk mengaktifkan pencadangan dan pemulihan native pada instans DB Anda, tambahkan opsi `SQLSERVER_BACKUP_RESTORE` untuk kelompok opsi pada instans DB Anda. Untuk informasi dan

instruksi selengkapnya, silakan lihat Dukungan untuk pencadangan dan pemulihan native di SQL Server.

Membuat peran IAM secara manual untuk pencadangan dan pemulihan native

Jika Anda ingin membuat peran IAM baru secara manual untuk digunakan dengan pencadangan dan pemulihan native, Anda dapat melakukannya. Dalam hal ini, Anda membuat peran untuk mendelegasikan izin dari layanan Amazon RDS ke bucket Amazon S3. Saat Anda membuat peran IAM, Anda melampirkan hubungan kepercayaan dan kebijakan perizinan. Hubungan kepercayaan memungkinkan RDS mengambil peran ini. Kebijakan izin menentukan tindakan yang dapat dilakukan peran ini. Untuk informasi selengkapnya tentang pembuatan peran, lihat Membuat peran untuk mendelegasikan izin ke layanan AWS.

Untuk fitur pencadangan dan pemulihan native, gunakan hubungan kepercayaan dan kebijakan perizinan yang serupa dengan contoh dalam bagian ini. Dalam contoh berikut, kami menggunakan namapengguna utama layanan `rds.amazonaws.com` sebagai alias untuk semua akun layanan. Dalam contoh lain, kami menentukan Amazon Resource Name (ARN) untuk mengidentifikasi akun, pengguna, atau peran lain yang kami berikan akses ke dalam kebijakan kepercayaan.

Sebaiknya gunakan kunci konteks kondisi global `aws:SourceArn` dan `aws:SourceAccount` dalam hubungan kepercayaan berbasis sumber daya untuk membatasi izin layanan ke sumber daya tertentu. Ini adalah perlindungan paling efektif dari masalah deputi yang membingungkan.

Anda dapat menggunakan kedua kunci konteks kondisi global dan menetapkan nilai `aws:SourceArn` yang berisi ID akun. Dalam hal ini, nilai `aws:SourceAccount` dan akun dalam nilai `aws:SourceArn` harus menggunakan ID akun yang sama ketika digunakan dalam pernyataan yang sama.

- Gunakan `aws:SourceArn` jika Anda menginginkan akses lintas layanan untuk satu sumber daya.
- Gunakan `aws:SourceAccount` jika Anda ingin mengizinkan pengaitan sumber daya apa pun di akun tersebut dengan penggunaan lintas layanan.

Dalam hubungan kepercayaan, pastikan untuk menggunakan kunci konteks kondisi global `aws:SourceArn` dengan ARN penuh dari sumber daya yang mengakses peran. Untuk pencadangan dan pemulihan native, pastikan untuk menyertakan grup opsi DB dan instans DB, seperti yang ditunjukkan dalam contoh berikut.

Example hubungan kepercayaan dengan kunci konteks kondisi global untuk pencadangan dan pemulihan native

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "rds.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceArn": [
                        "arn:aws:rds:Region:my_account_ID:db:db_instance_identifier",
                        "arn:aws:rds:Region:my_account_ID:og:option_group_name"
                    ]
                }
            }
        }
    ]
}
```

Contoh berikut menggunakan ARN untuk menentukan sumber daya. Untuk informasi cara menggunakan ARN selengkapnya, lihat Amazon resource name (ARN).

Example kebijakan perizinan untuk pencadangan dan pemulihan native tanpa didukung enkripsi

```
{
    "Version": "2012-10-17",
    "Statement":
    [
        {
        "Effect": "Allow",
        "Action":
            [
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
        "Resource": "arn:aws:s3:::bucket_name"
        },
        {
```

```
        "Effect": "Allow",
        "Action":
            [
                "s3:GetObjectAttributes",
                "s3:GetObject",
                "s3:PutObject",
                "s3:ListMultipartUploadParts",
                "s3:AbortMultipartUpload"
            ],
        "Resource": "arn:aws:s3:::bucket_name/*"
        }
    ]
}
```

Example kebijakan perizinan untuk pencadangan dan pemulihan native dengan didukung enkripsi

Jika Anda ingin mengenkripsi file backup, sertakan kunci enkripsi dalam kebijakan perizinan Anda. Untuk informasi kunci enkripsi selengkapnya, lihat Memulai di Panduan Developer AWS Key Management Service.

> **Note**
>
> Anda harus menggunakan kunci KMS enkripsi simetris untuk mengenkripsi cadangan Anda. Amazon RDS tidak mendukung kunci KMS asimetris. Untuk informasi selengkapnya, lihat Membuat kunci enkripsi simetris KMS di Panduan Pengembang AWS Key Management Service.
> Peran IAM juga harus menjadi pengguna kunci dan administrator kunci untuk kunci KMS, yaitu, harus disebutkan dalam kebijakan kunci. Untuk informasi selengkapnya, lihat Membuat kunci enkripsi simetris KMS di Panduan Pengembang AWS Key Management Service.

```
{
    "Version": "2012-10-17",
    "Statement":
    [
        {
        "Effect": "Allow",
        "Action":
            [
                "kms:DescribeKey",
                "kms:GenerateDataKey",
```

```
                "kms:Encrypt",
                "kms:Decrypt"
            ],
        "Resource": "arn:aws:kms:region:account-id:key/key-id"
        },
        {
        "Effect": "Allow",
        "Action":
            [
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
        "Resource": "arn:aws:s3:::bucket_name"
        },
        {
        "Effect": "Allow",
        "Action":
            [
                "s3:GetObjectAttributes",
                "s3:GetObject",
                "s3:PutObject",
                "s3:ListMultipartUploadParts",
                "s3:AbortMultipartUpload"
            ],
        "Resource": "arn:aws:s3:::bucket_name/*"
        }
    ]
}
```

Menggunakan pencadangan dan pemulihan native

Setelah Anda mengaktifkan dan mengkonfigurasi pencadangan dan pemulihan native, Anda dapat mulai menggunakannya. Pertama, Anda melakukan koneksi ke basis data Microsoft SQL Server, lalu Anda memanggil prosedur tersimpan Amazon RDS untuk melakukan pekerjaan tersebut. Untuk petunjuk tentang melakukan koneksi ke basis data Anda, lihat Menghubungkan ke instans DB yang menjalankan mesin basis data Microsoft SQL Server.

Beberapa prosedur yang disimpan mewajibkan Anda untuk memberikan Amazon Resource Name (ARN) ke bucket dan file Amazon S3 Anda. Format untuk ARN Anda adalah `arn:aws:s3:::bucket_name/file_name.extension`. Amazon S3 tidak memerlukan nomor akun atau Wilayah AWS di ARN.

Jika Anda juga memberikan kunci KMS opsional, format untuk ARN kuncinya adalah `arn:aws:kms:region:account-id:key/key-id`. Untuk informasi selengkapnya, lihat Amazon Resource Name (ARN) (ARN) dan namespace layanan AWS. Anda harus menggunakan kunci KMS enkripsi simetris untuk mengenkripsi cadangan Anda. Amazon RDS tidak mendukung kunci KMS asimetris. Untuk informasi selengkapnya, lihat Membuat kunci enkripsi simetris KMS di Panduan Pengembang AWS Key Management Service.

> **Note**
>
> Baik ketika Anda menggunakan kunci KMS atau tidak, tugas pencadangan dan pemulihan native mengaktifkan enkripsi 256-bit Advanced Encryption Standard (AES) sisi server secara default untuk file yang diunggah ke S3.

Untuk petunjuk bagaimana cara memanggil setiap prosedur tersimpan, lihat topik berikut:

- Membuat backup basis data
- Memulihkan basis data
- Memulihkan log
- Menyelesaikan pemulihan basis data
- Bekerja dengan basis data yang dipulihkan secara parsial
- Membatalkan tugas
- Melacak status tugas

Membuat backup basis data

Untuk membuat backup basis data Anda, gunakan prosedur `rds_backup_database` yang disimpan.

> **Note**
>
> Anda tidak dapat membuat backup basis data selama waktu pemeliharaan, atau saat Amazon RDS mengambil snapshot.

Penggunaan

```
exec msdb.dbo.rds_backup_database
 @source_db_name='database_name',
 @s3_arn_to_backup_to='arn:aws:s3:::bucket_name/file_name.extension',
 [@kms_master_key_arn='arn:aws:kms:region:account-id:key/key-id'],
 [@overwrite_s3_backup_file=0|1],
 [@type='DIFFERENTIAL|FULL'],
 [@number_of_files=n];
```

Parameter berikut diperlukan:

- `@source_db_name` – Nama basis data untuk dicadangkan.
- `@s3_arn_to_backup_to` – ARN yang menunjukkan bucket Amazon S3 yang akan digunakan untuk pencadangan, ditambah nama file backup.

  File dapat memiliki ekstensi apa saja, tetapi `.bak` biasanya digunakan.

Parameter berikut ini bersifat opsional:

- `@kms_master_key_arn`— ARN untuk kunci KMS enkripsi simetris untuk digunakan untuk mengenkripsi item.
  - Anda tidak dapat menggunakan kunci enkripsi default. Jika Anda menggunakan kunci default, basis data tidak akan dicadangkan.
  - Jika Anda tidak menentukan pengidentifikasi kunci KMS, file backup tidak akan dienkripsi. Untuk informasi selengkapnya, lihat Mengenkripsi sumber daya Amazon RDS.
  - Saat Anda menentukan kunci KMS, enkripsi sisi klien digunakan.

  - Amazon RDS tidak mendukung kunci KMS asimetris. Untuk informasi selengkapnya, lihat Membuat kunci enkripsi simetris KMS di Panduan Pengembang AWS Key Management Service.
- `@overwrite_s3_backup_file` – Nilai yang menunjukkan apakah file backup yang sudah ada akan ditimpa.
  - `0` – Tidak menimpa file yang ada. Nilai ini adalah default.

    Mengatur `@overwrite_s3_backup_file` ke 0 akan menghasilkan kesalahan jika file sudah ada.
  - `1` – Menimpa file yang sudah ada dengan nama yang ditentukan, meskipun itu bukan file cadangan.
- `@type` – Jenis pencadangan.
  - `DIFFERENTIAL` – Membuat pencadangan diferensial.
  - `FULL` – Membuat backup penuh. Nilai ini adalah default.

  Pencadangan diferensial didasarkan pada backup penuh yang terakhir. Agar pencadangan diferensial berhasil, Anda tidak dapat mengambil snapshot antara backup penuh terakhir dan pencadangan diferensial. Jika Anda menginginkan pencadangan diferensial, tetapi ada snapshot, lakukan pencadangan penuh lainnya sebelum melanjutkan dengan pencadangan diferensial.

  Anda dapat mencari backup penuh atau snapshot terakhir menggunakan contoh kueri SQL berikut:

```
select top 1
database_name
,  backup_start_date
,  backup_finish_date
from    msdb.dbo.backupset
where   database_name='mydatabase'
and     type = 'D'
order by backup_start_date desc;
```

- `@number_of_files` – Jumlah file yang menjadi tempat di mana backup akan terbagi (terpotong). Jumlah maksimum adalah 10.
  - Pencadangan multi-file didukung untuk backup penuh dan pencadangan diferensial.
  - Jika Anda memasukkan nilai 1 atau menghilangkan parameter, akan dibuat file backup tunggal.

  Berikan awalan yang sama-sama dimiliki beberapa file tersebut, lalu akhiri dengan tanda bintang (*). Tanda bintang dapat diletakkan dimana saja di bagian *`file_name`* dari ARN S3. Tanda

bintang diganti dengan serangkaian string alfanumerik dalam file yang dihasilkan, dimulai dengan `1-of-number_of_files`.

Misalnya, jika nama file pada S3 ARN adalah `backup*.bak` dan Anda mengatur `@number_of_files=4`, file backup yang dihasilkan adalah `backup1-of-4.bak`, `backup2-of-4.bak`, `backup3-of-4.bak`, dan `backup4-of-4.bak`.

- Jika salah satu nama file sudah ada, dan `@overwrite_s3_backup_file` adalah 0, akan muncul kesalahan.
- Pencadangan multi-file hanya dapat memiliki satu tanda bintang di bagian *file_name* dari ARN S3.
- Backup file tunggal dapat memiliki berapa saja tanda bintang dalam bagian *file_name* dari ARN S3. Tanda bintang tidak dihapus dari nama file yang dihasilkan.

Contoh

Example dari pencadangan diferensial

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup1.bak',
@overwrite_s3_backup_file=1,
@type='DIFFERENTIAL';
```

Example dari backup penuh dengan enkripsi

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup1.bak',
@kms_master_key_arn='arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE',
@overwrite_s3_backup_file=1,
@type='FULL';
```

Example dari pencadangan multi-file

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup*.bak',
@number_of_files=4;
```

Example dari pencadangan diferensial multi-file

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup*.bak',
@type='DIFFERENTIAL',
@number_of_files=4;
```

Example dari pencadangan multi-file dengan enkripsi

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup*.bak',
@kms_master_key_arn='arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE',
@number_of_files=4;
```

Example dari pencadangan multi-file dengan penimpaan S3

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup*.bak',
@overwrite_s3_backup_file=1,
@number_of_files=4;
```

Example dari pencadangan file tunggal dengan parameter @number_of_files

Contoh ini menghasilkan file backup dengan nama `backup*.bak`.

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup*.bak',
@number_of_files=1;
```

Memulihkan basis data

Untuk memulihkan basis data Anda, panggil prosedur tersimpan `rds_restore_database`. Amazon RDS membuat snapshot awal basis data setelah tugas pemulihan selesai dan basis data terbuka.

Penggunaan

```
exec msdb.dbo.rds_restore_database
```

```
@restore_db_name='database_name',
@s3_arn_to_restore_from='arn:aws:s3:::bucket_name/file_name.extension',
@with_norecovery=0|1,
[@kms_master_key_arn='arn:aws:kms:region:account-id:key/key-id'],
[@type='DIFFERENTIAL|FULL'];
```

Parameter berikut diperlukan:

- `@restore_db_name` – Nama basis data yang akan dipulihkan. Nama basis data bersifat unik. Anda tidak dapat memulihkan basis data dengan nama yang sama seperti basis data yang sudah ada.
- `@s3_arn_to_restore_from` – ARN yang menunjukkan prefiks Amazon S3 dan nama file backup yang digunakan untuk memulihkan basis data.
  - Untuk pencadangan file tunggal, berikan seluruh nama file.
  - Berikan awalan yang sama-sama dimiliki beberapa file tersebut, lalu akhiri dengan tanda bintang (*).
  - Jika `@s3_arn_to_restore_from` kosong, pesan kesalahan berikut dikembalikan: Awalan ARN S3 tidak boleh kosong.

Parameter berikut ini diperlukan untuk pemulihan diferensial, tetapi opsional untuk pemulihan penuh:

- `@with_norecovery` – Klausul pemulihan yang digunakan untuk operasi pemulihan.
  - Atur ke `0` untuk memulihkan dengan RECOVERY. Dalam hal ini, basis data sedang online setelah pemulihan.
  - Atur ke `1` untuk memulihkan dengan NORECOVERY. Dalam hal ini, basis data tetap dalam status RESTORING setelah tugas pemulihan selesai. Dengan pendekatan ini, Anda dapat melakukan pemulihan diferensial berikutnya.
  - Untuk pemulihan DIFFERENTIAL, tentukan `0` atau `1`.
  - Untuk pemulihan `FULL`, nilai ini default menjadi `0`.

Parameter berikut ini bersifat opsional:

- `@kms_master_key_arn` — Jika Anda mengenkripsi file backup, kunci KMS digunakan untuk mendekripsi file.

  Saat Anda menentukan kunci KMS, enkripsi sisi klien digunakan.

- `@type` – Jenis pemulihan. Nilai yang valid adalah `DIFFERENTIAL` dan `FULL`. Nilai defaultnya adalah `FULL`.

> **Note**
>
> Untuk pemulihan diferensial, basis data harus berada dalam status RESTORING atau harus ada tugas yang memulihkan dengan NORECOVERY.
> Anda tidak dapat memulihkan pencadangan diferensial di masa mendatang saat basis data online.
> Anda tidak dapat mengirimkan tugas pemulihan untuk basis data yang sudah memiliki tugas pemulihan tertunda dengan RECOVERY.
> Pemulihan penuh dengan NORECOVERY dan pemulihan diferensial tidak didukung pada instans Multi-AZ.
> Memulihkan basis data di instans Multi-Z dengan replika baca itu serupa dengan memulihkan basis data di instans Multi-AZ. Anda tidak perlu mengambil tindakan tambahan apa pun untuk memulihkan basis data di replika.

Contoh

Example dari pemulihan file tunggal

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak';
```

Example dari pemulihan multi-file

Untuk menghindari kesalahan saat memulihkan beberapa file, pastikan semua file backup memiliki awalan yang sama, dan tidak ada file lain yang menggunakan awalan tersebut.

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup*';
```

Example dari pemulihan basis data penuh dengan RECOVERY

Tiga contoh berikut ini melakukan tugas yang sama, pemulihan penuh dengan RECOVERY.

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak';
```

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
[@type='DIFFERENTIAL|FULL'];
```

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
@type='FULL',
@with_norecovery=0;
```

Example dari pemulihan basis data penuh dengan enkripsi

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
@kms_master_key_arn='arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE';
```

Example dari pemulihan basis data penuh dengan NORECOVERY

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
@type='FULL',
@with_norecovery=1;
```

Example dari pemulihan diferensial dengan NORECOVERY

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
@type='DIFFERENTIAL',
@with_norecovery=1;
```

Example dari pemulihan diferensial dengan RECOVERY

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
@type='DIFFERENTIAL',
@with_norecovery=0;
```

Memulihkan log

Untuk memulihkan basis data Anda, panggil prosedur `rds_restore_log` yang disimpan.

Penggunaan

```
exec msdb.dbo.rds_restore_log
 @restore_db_name='database_name',
 @s3_arn_to_restore_from='arn:aws:s3:::bucket_name/log_file_name.extension',
 [@kms_master_key_arn='arn:aws:kms:region:account-id:key/key-id'],
 [@with_norecovery=0|1],
 [@stopat='datetime'];
```

Parameter berikut diperlukan:

- `@restore_db_name` – Nama basis data yang lognya akan dipulihkan.
- `@s3_arn_to_restore_from` – ARN menunjukkan awalan Amazon S3 dan nama file log yang digunakan untuk memulihkan log. File dapat memiliki ekstensi apa saja, tetapi `.trn` biasanya digunakan.

  Jika `@s3_arn_to_restore_from` kosong, pesan kesalahan berikut dikembalikan: Awalan ARN S3 tidak boleh kosong.

Parameter berikut ini bersifat opsional:

- `@kms_master_key_arn` — Jika Anda mengenkripsi log, kunci KMS digunakan untuk mendekripsi log.
- `@with_norecovery` – Klausul pemulihan yang digunakan untuk operasi pemulihan. Nilai ini default menjadi 1.

  - Atur ke `0` untuk memulihkan dengan RECOVERY. Dalam hal ini, basis data sedang online setelah pemulihan. Anda tidak dapat memulihkan backup log lebih lanjut saat basis data sedang online.
  - Atur ke `1` untuk memulihkan dengan NORECOVERY. Dalam hal ini, basis data tetap dalam status RESTORING setelah tugas pemulihan selesai. Dengan pendekatan ini, Anda dapat melakukan pemulihan log berikutnya.
- `@stopat` – Nilai yang menentukan bahwa basis data dipulihkan ke statusnya pada tanggal dan waktu yang ditentukan (dalam format tanggal). Hanya catatan log transaksi yang ditulis sebelum tanggal dan waktu yang ditentukan diterapkan ke basis data.

  Jika parameter ini tidak ditentukan (NULL), log lengkap dipulihkan.

> **Note**
>
> Untuk pemulihan log, basis data harus dalam keadaan memulihkan atau tugas yang memulihkan dengan NORECOVERY harus sudah ada.
> Anda tidak dapat memulihkan backup log saat basis data sedang online.
> Anda tidak dapat mengirimkan tugas pemulihan di basis data yang sudah memiliki tugas pemulihan tertunda dengan RECOVERY.
> Pemulihan log tidak didukung pada instans Multi-AZ.

Contoh

Example dari pemulihan log

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn';
```

Example dari pemulihan log dengan enkripsi

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn',
@kms_master_key_arn='arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE';
```

Example dari pemulihan log dengan NORECOVERY

Dua contoh berikut melakukan tugas yang sama, pemulihan log dengan NORECOVERY.

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn',
@with_norecovery=1;
```

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn';
```

Example dari pemulihan log dengan RECOVERY

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn',
@with_norecovery=0;
```

Example dari pemulihan log dengan klausul STOPAT

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn',
@with_norecovery=0,
@stopat='2019-12-01 03:57:09';
```

Menyelesaikan pemulihan basis data

Jika tugas pemulihan terakhir pada basis data dilakukan menggunakan `@with_norecovery=1`, basis data sekarang dalam status RESTORING. Buka basis data ini untuk operasi normal dengan menggunakan prosedur tersimpan `rds_finish_restore`.

Penggunaan

```
exec msdb.dbo.rds_finish_restore @db_name='database_name';
```

> **Note**
>
> Untuk menggunakan pendekatan ini, basis data harus berada dalam status RESTORING tanpa ada tugas pemulihan yang tertunda.
> Prosedur `rds_finish_restore` tidak didukung dalam instans Multi-AZ.
> Untuk menyelesaikan pemulihan basis data, gunakan master login. Atau gunakan login pengguna terbaru yang memulihkan basis data atau log dengan NORECOVERY.

Bekerja dengan basis data yang dipulihkan secara parsial

Menghilangkan basis data yang dipulihkan parsial

Untuk melepaskan basis data yang dipulihkan parsial (dibiarkan dalam status RESTORING), gunakan prosedur tersimpan `rds_drop_database`.

```
exec msdb.dbo.rds_drop_database @db_name='database_name';
```

> **Note**
>
> Anda tidak dapat mengirimkan permintaan DROP basis data untuk basis data yang sudah memiliki tugas pemulihan tertunda atau sudah menyelesaikan tugas pemulihan.
> Untuk melepaskan basis data, gunakan master login. Atau gunakan login pengguna terbaru yang memulihkan basis data atau log dengan NORECOVERY.

Pemulihan snapshot dan perilaku pemulihan titik waktu untuk pemulihan basis data parsial

basis data yang dipulihkan parsial dalam instans sumber (dibiarkan dalam status RESTORING) akan dilepaskan dari target instans selama pemulihan snapshot dan pemulihan titik waktu.

Membatalkan tugas

Untuk membatalkan tugas pencadangan atau pemulihan, panggil prosedur yang tersimpan `rds_cancel_task`.

> **Note**
>
> Anda tidak dapat membatalkan tugas FINISH_RESTORE.

Penggunaan

```
exec msdb.dbo.rds_cancel_task @task_id=ID_number;
```

Parameter berikut diperlukan:

- `@task_id` – ID tugas yang dibatalkan. Anda bisa mendapatkan ID tugas dengan memanggil `rds_task_status`.

Melacak status tugas

Untuk melacak status tugas pencadangan dan pemulihan Anda, panggil prosedur tersimpan `rds_task_status`. Jika Anda tidak memberikan parameter apa pun, prosedur yang tersimpan akan memperlihatkan status semua tugas. Status tugas diperbarui kira-kira setiap dua menit sekali. Riwayat tugas disimpan selama 36 hari.

Penggunaan

```
exec msdb.dbo.rds_task_status
 [@db_name='database_name'],
 [@task_id=ID_number];
```

Parameter berikut ini bersifat opsional:

- `@db_name` – Nama basis data untuk menunjukkan status tugas.
- `@task_id` – ID tugas untuk menunjukkan status tugas.

Contoh

Example untuk menyusun daftar status untuk tugas tertentu

```
exec msdb.dbo.rds_task_status @task_id=5;
```

Example untuk menyusun daftar status untuk basis data dan tugas tertentu

```
exec msdb.dbo.rds_task_status
@db_name='my_database',
@task_id=5;
```

Example untuk menyusun daftar semua tugas dan statusnya di basis data tertentu

```
exec msdb.dbo.rds_task_status @db_name='my_database';
```

Example untuk menyusun daftar semua tugas dan statusnya di instans saat ini

```
exec msdb.dbo.rds_task_status;
```

Respons

Prosedur tersimpan `rds_task_status` memunculkan kolom berikut.

Kolom	Deskripsi
`task_id`	ID tugas.

`task_type`	Jenis tugas tergantung pada parameter input, sebagai berikut:
	• Untuk tugas pencadangan:
	  • BACKUP_DB – Pencadangan basis data secara penuh
	  • BACKUP_DB_DIFFERENTIAL – Pencadangan basis data secara diferensial
	• Untuk tugas pemulihan:
	  • RESTORE_DB – Pemulihan basis data secara penuh dengan RECOVERY
	  • RESTORE_DB_NORECOVERY – Pemulihan basis data secara penuh dengan NORECOVERY
	  • RESTORE_DB_DIFFERENTIAL – Pemulihan basis data secara diferensial dengan RECOVERY
	  • RESTORE_DB_DIFFERENTIAL_NORECOVERY – Pemulihan basis data secara diferensial dengan NORECOVERY

Kolom	Deskripsi
	• RESTORE_DB_LOG – Pemulihan log dengan RECOVERY • RESTORE_DB_LOG_NORECOVERY – Pemulihan log dengan NORECOVERY • Untuk tugas yang menyelesaikan pemulihan: • FINISH_RESTORE – Selesaikan pemulihan dan buka basis data Amazon RDS membuat snapshot awal basis data setelah basis data terbuka saat menyelesaikan tugas pemulihan berikut: • RESTORE_DB • RESTORE_DB_DIFFERENTIAL • RESTORE_DB_LOG • FINISH_RESTORE
`database_name`	Nama basis data yang terkait dengan tugas.
`% complete`	Kemajuan tugas sebagai nilai persen.
`duration (mins)`	Durasi yang dihabiskan untuk tugas, dalam menit.

Kolom	Deskripsi
`lifecycle`	Status tugas. Status yang mungkin adalah sebagai berikut: • `CREATED` – Segera setelah Anda memanggil `rds_backup_database` atau `rds_restore_database` , tugas dibuat dan status diatur menjadi `CREATED`. • `IN_PROGRESS` – Setelah tugas pencadangan atau pemulihan dimulai, statusnya diatur menjadi `IN_PROGRESS` . Diperlukan hingga 5 menit agar status berubah dari `CREATED` ke `IN_PROGRESS` . • `SUCCESS` – Setelah tugas pencadangan atau pemulihan selesai, statusnya diatur menjadi `SUCCESS`. • `ERROR` – Setelah tugas pencadangan atau pemulihan gagal, statusnya diatur menjadi `ERROR`. Untuk informasi selengkapnya tentang kesalahan, lihat kolom `task_info` . • `CANCEL_REQUESTED` – Segera setelah Anda memanggil `rds_cancel_task` , status tugas diatur menjadi `CANCEL_REQUESTED` . • `CANCELLED` – Setelah tugas berhasil dibatalkan, statusnya diubah menjadi `CANCELLED` .
`task_info`	Informasi tambahan tentang tugas. Jika terjadi kesalahan saat membuat backup atau memulihkan basis data, kolom ini berisi informasi tentang kesalahan tersebut. Untuk daftar kesalahan yang mungkin terjadi, dan strategi mitigasinya, lihat Pemecahan Masalah.
`last_updated`	Tanggal dan waktu status tugas terakhir diperbarui. Status akan diperbarui setelah setiap 5 persen kemajuan.
`created_at`	Tanggal dan waktu tugas dibuat.
`S3_object_arn`	ARN menunjukkan awalan Amazon S3 dan nama file yang sedang dicadangkan atau dipulihkan.

Kolom	Deskripsi
`overwrite_s3_backup_file`	Nilai dari parameter `@overwrite_s3_backup_file` yang ditentukan saat memanggil tugas pencadangan. Untuk informasi selengkapnya, lihat Membuat backup basis data.
`KMS_master_key_arn`	ARN untuk kunci KMS yang digunakan untuk enkripsi (untuk pencadangan) dan dekripsi (untuk pemulihan).
`filepath`	Tidak berlaku untuk tugas pencadangan native dan pemulihan native.
`overwrite_file`	Tidak berlaku untuk tugas pencadangan native dan pemulihan native.

Mengompresi file backup

Untuk menghemat ruang di bucket Amazon S3, Anda dapat mengompresi file backup. Untuk informasi selengkapnya tentang pengompresan file backup, lihat Kompresi backup dalam dokumentasi Microsoft.

Pengompresan file backup didukung untuk edisi basis data berikut:

- Edisi Perusahaan Microsoft SQL Server
- Edisi Standar Microsoft SQL Server

Untuk mengaktifkan kompresi untuk file backup Anda, jalankan kode berikut:

```
exec rdsadmin..rds_set_configuration 'S3 backup compression', 'true';
```

Untuk mengaktifkan kompresi untuk file backup Anda, jalankan kode berikut:

```
exec rdsadmin..rds_set_configuration 'S3 backup compression', 'false';
```

Pemecahan Masalah

Berikut adalah masalah yang mungkin Anda hadapi saat menggunakan pencadangan native dan pemulihan native.

Masalah	Saran pemecahan masalah
Opsi pencadangan/pemulihan basis data belum diaktifkan atau sedang dalam proses diaktifkan. Silakan di coba lagi nanti.	Pastikan Anda telah menambahkan opsi `SQLSERVER_BACKUP_RESTORE` ke grup opsi DB yang terkait dengan instans DB Anda. Untuk informasi selengkapnya, lihat Menambahkan opsi pencadangan dan pemulihan native.
Akses Ditolak	Proses pencadangan atau pemulihan tidak dapat mengakses file backup. Hal ini biasanya disebabkan oleh masalah seperti berikut: • Mengacu pada bucket yang salah. Mengacu pada bucket yang menggunakan format yang salah. Mengacu pada nama file tanpa menggunakan ARN. • Izin yang salah pada file bucket. Misalnya, jika dibuat oleh akun lain yang mencoba mengaksesnya sekarang, tambahkan izin yang benar. • Kebijakan IAM yang tidak benar atau tidak lengkap. Peran IAM Anda harus mencakup semua elemen yang diperlukan, termasuk, misalnya, versi yang benar. Semua ini disoroti dalam Mengimpor dan mengekspor basis data SQL Server menggunakan pencadangan dan pemulihan native.
BACKUP basis data WITH COMPRESSION tidak didukung pada Edisi <edition_name>	Mengompresi file backup Anda hanya didukung untuk Edisi Perusahaan dan Edisi Standar Microsoft SQL Server. Untuk informasi selengkapnya, lihat Mengompresi file backup.
Kunci <ARN> tidak ada	Anda mencoba memulihkan backup terenkripsi, tetapi tidak menyediakan kunci enkripsi yang valid. Periksa kunci enkripsi Anda dan coba lagi. Untuk informasi selengkapnya, lihat Memulihkan basis data.

Masalah	Saran pemecahan masalah
Silakan menerbitkan ulang tugas dengan properti ketik dan timpa yang benar	Jika Anda mencoba membuat backup basis data dan memberikan nama file yang sudah ada, tetapi mengatur properti overwrite ke false, operasi penyimpanan gagal. Untuk memperbaiki kesalahan ini, berikan nama file yang belum ada, atau atur properti overwrite ke true. Untuk informasi selengkapnya, lihat Membuat backup basis data. Mungkin saja Anda ingin memulihkan basis data Anda, tetapi memanggil prosedur tersimpan `rds_backup_database` secara tidak sengaja. Dalam hal ini, panggil prosedur tersimpan `rds_restore_database` . Untuk informasi selengkapnya, lihat Memulihkan basis data. Jika Anda ingin memulihkan basis data dan memanggil prosedur tersimpan `rds_restore_database` , pastikan bahwa Anda memberikan nama file backup yang valid. Untuk informasi selengkapnya, lihat Menggunakan pencadangan dan pemulihan native.
Silakan tentukan bucket yang berada di wilayah yang sama dengan instans RDS	Anda tidak dapat mencadangkan ke, atau memulihkan dari, bucket Amazon S3 di Wilayah AWS yang berbeda dari instans DB Amazon RDS Anda. Anda dapat menggunakan replikasi Amazon S3 untuk menyalin file backup ke Wilayah AWS yang benar. Untuk informasi lebih lanjut, lihat Replikasi Lintas Wilayah dalam dokumentasi Amazon S3.
Bucket yang ditentukan tidak ada	Pastikan bahwa Anda telah memberikan ARN yang benar untuk bucket dan file Anda, dalam format yang benar. Untuk informasi selengkapnya, lihat Menggunakan pencadangan dan pemulihan native.

Masalah	Saran pemecahan masalah
Pengguna <ARN> tidak berwenang untuk melakukan <kms action> pada sumber daya <ARN>	Anda meminta operasi terenkripsi, tetapi tidak memberikan izin AWS KMS yang benar. Verifikasi bahwa Anda memiliki izin yang benar, atau tambahkan. Untuk informasi selengkapnya, lihat Menyiapkan pencadangan dan pemulihan native.
Tugas Pemulihan tidak dapat memulihkan dari lebih dari 10 file backup. Kurangi jumlah file yang dicocokkan dan coba lagi.	Kurangi jumlah file yang ingin Anda gunakan untuk pemulihan. Anda dapat menjadikan setiap file lebih besar kalau perlu.
Basis data '*database_name* ' sudah ada. Dua basis data yang berbeda besar huruf atau aksen tidak diperbolehkan. Pilih nama basis data yang berbeda.	Anda tidak dapat memulihkan basis data dengan nama yang sama seperti basis data yang sudah ada. Nama basis data bersifat unik.

Mengimpor dan mengekspor data SQL Server menggunakan metode lain

Selanjutnya, Anda dapat menemukan informasi tentang penggunaan snapshot untuk mengimpor data Microsoft SQL Server Anda ke Amazon RDS. Anda juga dapat menemukan informasi tentang penggunaan snapshot untuk mengekspor data Anda dari instans DB RDS yang menjalankan SQL Server.

Jika skenario Anda mendukung, lebih mudah untuk memindahkan data ke dan dari Amazon RDS dengan menggunakan fungsi pencadangan native dan pemulihan native. Untuk informasi selengkapnya, lihat Mengimpor dan mengekspor basis data SQL Server menggunakan pencadangan dan pemulihan native.

> **Note**
>
> Amazon RDS for Microsoft SQL Server tidak mendukung impor data ke basis data `msdb`.

Mengimpor data ke RDS for SQL Server dengan menggunakan snapshot

Untuk mengimpor data ke dalam instans DB SQL Server dengan menggunakan snapshot

1. Buat instans DB. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.
2. Hentikan aplikasi dari mengakses instans DB tujuan.

   Jika Anda mencegah akses ke instans DB saat mengimpor data, transfer data akan lebih cepat. Selain itu, Anda tidak perlu khawatir tentang konflik ketika data sedang dimuat jika aplikasi lain tidak dapat menulis ke instans DB pada waktu yang sama. Jika ada yang salah dan Anda harus kembali ke snapshot basis data sebelumnya, satu-satunya perubahan yang hilang adalah data hasil impor. Anda dapat mengimpor data ini kembali setelah Anda menyelesaikan masalah.

   Untuk informasi tentang mengendalikan akses ke instans DB Anda, lihat Mengontrol akses dengan grup keamanan.
3. Buat snapshot basis data target.

   Jika basis data target sudah terisi data, kami menyarankan agar Anda mengambil snapshot basis data sebelum Anda melakukan impor data. Jika ada yang salah dalam impor data atau Anda ingin membuang perubahan, Anda dapat memulihkan basis data ke status sebelumnya dengan menggunakan snapshot. Untuk informasi tentang snapshot basis data, lihat Membuat snapshot DB untuk instans DB Single-AZ.

> **Note**
>
> Saat Anda mengambil snapshot basis data, operasi I/O ke basis data ditangguhkan untuk beberapa saat (milidetik) saat pencadangan sedang berlangsung.

4. Nonaktifkan pencadangan otomatis pada basis data target.

   Menonaktifkan pencadangan otomatis pada instans DB target akan meningkatkan kinerja saat Anda mengimpor data Anda karena Amazon RDS tidak mencatat transaksi ketika pencadangan otomatis dinonaktifkan. Namun, ada beberapa hal yang perlu dipertimbangkan. Pencadangan otomatis diperlukan untuk melakukan pemulihan titik waktu. Dengan demikian, Anda tidak dapat memulihkan basis data ke titik waktu tertentu saat Anda melakukan impor data. Selain itu, semua pencadangan otomatis yang dibuat di instans DB akan dihapus kecuali jika Anda memilih untuk menyimpannya.

   Memilih untuk menyimpan pencadangan otomatis dapat membantu melindungi Anda dari menghapus data secara tidak sengaja. Amazon RDS juga menyimpan properti instans basis data bersamaan dengan setiap pencadangan otomatis agar mudah dipulihkan. Dengan opsi ini, Anda dapat memulihkan instans basis data yang telah dihapus ke titik tertentu dalam periode retensi pencadangan bahkan setelah menghapusnya. Pencadangan otomatis secara otomatis dihapus di akhir waktu backup yang ditentukan, seperti halnya untuk instans basis data aktif.

   Anda juga dapat menggunakan snapshot sebelumnya untuk memulihkan basis data, dan semua snapshot yang Anda ambil tetap tersedia. Untuk mengetahui informasi tentang pencadangan otomatis, lihat Pengantar cadangan.

5. Nonaktifkan batasan kunci asing, jika berlaku.

   Jika Anda perlu menonaktifkan batasan kunci asing, Anda dapat melakukannya dengan skrip berikut.

```
--Disable foreign keys on all tables
    DECLARE @table_name SYSNAME;
    DECLARE @cmd NVARCHAR(MAX);
    DECLARE table_cursor CURSOR FOR SELECT name FROM sys.tables;

    OPEN table_cursor;
    FETCH NEXT FROM table_cursor INTO @table_name;

    WHILE @@FETCH_STATUS = 0 BEGIN
```

```
      SELECT @cmd = 'ALTER TABLE '+QUOTENAME(@table_name)+' NOCHECK CONSTRAINT
 ALL';
      EXEC (@cmd);
      FETCH NEXT FROM table_cursor INTO @table_name;
    END

    CLOSE table_cursor;
    DEALLOCATE table_cursor;

    GO
```

6. Lepaskan indeks, jika berlaku.
7. Nonaktifkan pemicu, jika berlaku.

   Jika Anda perlu menonaktifkan pemicu, Anda dapat melakukannya dengan skrip berikut.

```
--Disable triggers on all tables
    DECLARE @enable BIT = 0;
    DECLARE @trigger SYSNAME;
    DECLARE @table SYSNAME;
    DECLARE @cmd NVARCHAR(MAX);
    DECLARE trigger_cursor CURSOR FOR SELECT trigger_object.name trigger_name,
     table_object.name table_name
    FROM sysobjects trigger_object
    JOIN sysobjects table_object ON trigger_object.parent_obj = table_object.id
    WHERE trigger_object.type = 'TR';

    OPEN trigger_cursor;
    FETCH NEXT FROM trigger_cursor INTO @trigger, @table;

    WHILE @@FETCH_STATUS = 0 BEGIN
      IF @enable = 1
         SET @cmd = 'ENABLE ';
      ELSE
         SET @cmd = 'DISABLE ';

      SET @cmd = @cmd + ' TRIGGER dbo.'+QUOTENAME(@trigger)+' ON
 dbo.'+QUOTENAME(@table)+' ';
      EXEC (@cmd);
      FETCH NEXT FROM trigger_cursor INTO @trigger, @table;
    END

    CLOSE trigger_cursor;
```

```
    DEALLOCATE trigger_cursor;

    GO
```

8. Kueri sumber instans SQL Server untuk setiap login yang ingin Anda impor ke instans DB tujuan.

   SQL Server menyimpan login dan kata sandi di basis data `master`. Karena Amazon RDS tidak memberikan akses ke basis data `master`, Anda tidak dapat langsung melakukan impor login dan kata sandi ke instans DB tujuan Anda. Sebaliknya, Anda harus kueri basis data `master` pada sumber instans SQL Server untuk membuat file data definition language (DDL). File ini harus mencakup semua login dan kata sandi yang ingin Anda tambahkan ke instans DB tujuan. File ini juga harus menyertakan keanggotaan peran dan izin yang ingin Anda transfer.

   Untuk informasi tentang membuat kueri basis data `master`, lihat Cara mentransfer login and kata sandi antar instans SQL Server 2005 dan SQL Server 2008 di Microsoft Knowledge Base.

   Output dari skrip tersebut adalah skrip lain yang dapat Anda jalankan di instans DB tujuan. Script dalam artikel Knowledge Base memiliki kode berikut:

   ```
   p.type IN
   ```

   Setiap tempat `p.type` muncul, gunakan kode berikut ini:

   ```
   p.type = 'S'
   ```

9. Impor data menggunakan metode di Mengimpor data.
10. Berikan akses aplikasi ke instans DB target.

    Setelah impor data selesai, Anda dapat memberikan akses ke instans DB pada aplikasi yang Anda blokir selama impor. Untuk informasi tentang mengendalikan akses ke instans DB Anda, lihat Mengontrol akses dengan grup keamanan.

11. Aktifkan pencadangan otomatis pada instans DB target.

    Untuk mengetahui informasi tentang pencadangan otomatis, lihat Pengantar cadangan.

12. Aktifkan batasan kunci asing.

    Jika Anda menonaktifkan batasan kunci asing sebelumnya, sekarang Anda dapat mengaktifkannya dengan skrip berikut.

```
--Enable foreign keys on all tables
    DECLARE @table_name SYSNAME;
    DECLARE @cmd NVARCHAR(MAX);
    DECLARE table_cursor CURSOR FOR SELECT name FROM sys.tables;

    OPEN table_cursor;
    FETCH NEXT FROM table_cursor INTO @table_name;

    WHILE @@FETCH_STATUS = 0 BEGIN
      SELECT @cmd = 'ALTER TABLE '+QUOTENAME(@table_name)+' CHECK CONSTRAINT ALL';
      EXEC (@cmd);
      FETCH NEXT FROM table_cursor INTO @table_name;
    END

    CLOSE table_cursor;
    DEALLOCATE table_cursor;
```

13. Aktifkan indeks, jika berlaku.
14. Aktifkan pemicu, jika berlaku.

    Jika Anda menonaktifkan pemicu sebelumnya, sekarang Anda dapat mengaktifkannya dengan skrip berikut.

```
--Enable triggers on all tables
    DECLARE @enable BIT = 1;
    DECLARE @trigger SYSNAME;
    DECLARE @table SYSNAME;
    DECLARE @cmd NVARCHAR(MAX);
    DECLARE trigger_cursor CURSOR FOR SELECT trigger_object.name trigger_name,
     table_object.name table_name
    FROM sysobjects trigger_object
    JOIN sysobjects table_object ON trigger_object.parent_obj = table_object.id
    WHERE trigger_object.type = 'TR';

    OPEN trigger_cursor;
    FETCH NEXT FROM trigger_cursor INTO @trigger, @table;

    WHILE @@FETCH_STATUS = 0 BEGIN
      IF @enable = 1
         SET @cmd = 'ENABLE ';
      ELSE
         SET @cmd = 'DISABLE ';
```

```
     SET @cmd = @cmd + ' TRIGGER dbo.'+QUOTENAME(@trigger)+' ON
dbo.'+QUOTENAME(@table)+' ';
     EXEC (@cmd);
     FETCH NEXT FROM trigger_cursor INTO @trigger, @table;
   END

   CLOSE trigger_cursor;
   DEALLOCATE trigger_cursor;
```

Mengimpor data

Microsoft SQL Server Management Studio adalah klien SQL Server grafis yang disertakan dalam semua edisi Microsoft SQL Server kecuali Edisi Ekspres. SQL Server Management Studio Express tersedia dari Microsoft sebagai unduhan gratis. Untuk menemukan unduhan ini, lihat situs web Microsoft.

> **Note**
>
> SQL Server Management Studio hanya tersedia sebagai aplikasi berbasis Windows.

SQL Server Management Studio mencakup alat-alat berikut, yang berguna dalam melakukan impor data ke instans DB SQL Server:

- Wizard Membuat dan Menerbitkan Skrip
- Wizard Impor dan Ekspor
- Penyalinan massal

Wizard Membuat dan Menerbitkan Skrip

Wizard Membuat dan Menerbitkan Skrip membuat skrip yang berisi skema basis data, data itu sendiri, atau keduanya. Anda dapat membuat skrip untuk basis data dalam deployment SQL Server lokal Anda. Kemudian, Anda dapat menjalankan skrip untuk mentransfer informasi yang dimuat ke instans DB Amazon RDS.

> **Note**
>
> Untuk basis data sebesar 1 GiB atau lebih besar, lebih efisien jika hanya membuat skrip skema basis data. Anda kemudian menggunakan Wizard Impor dan Ekspor atau fitur penyalinan massal dari SQL Server untuk mentransfer data.

Untuk informasi terperinci tentang Wizard Membuat dan Menerbitkan Skrip, lihat Dokumentasi Microsoft SQL Server.

Di dalam wizard, perhatikan dengan cermat opsi lanjutan pada halaman Tetapkan Opsi Pembuatan Skrip untuk memastikan bahwa semua yang Anda inginkan dalam skrip Anda sudah dipilih. Misalnya, secara default, pemicu basis data tidak disertakan dalam skrip.

Saat skrip dibuat dan disimpan, Anda dapat menggunakan SQL Server Management Studio untuk membuat koneksi ke instans DB Anda lalu menjalankan skrip tersebut.

Wizard Impor dan Ekspor

Wizard Impor dan Ekspor membuat paket Layanan Integrasi khusus, yang dapat Anda gunakan untuk menyalin data dari basis data SQL Server lokal Anda ke instans DB tujuan. Wizard dapat menyaring tabel mana dan bahkan tupel mana dalam tabel yang disalin ke instans DB tujuan.

> **Note**
>
> Wizard Impor dan Ekspor berfungsi dengan baik untuk set data besar, tapi mungkin bukan cara tercepat untuk melakukan ekspor data jarak jauh dari deployment lokal Anda. Untuk cara yang lebih cepat, pertimbangkan fitur penyalinan massal SQL Server.

Untuk informasi terperinci tentang Wizard Impor dan Ekspor, lihat Dokumentasi Microsoft SQL Server.

Dalam wizard, di halaman Pilih Tujuan, lakukan hal berikut:

- Untuk Nama Server, ketik nama titik akhir untuk instans DB Anda.
- Untuk mode autentikasi server, pilih Gunakan Autentikasi SQL Server.
- Untuk Nama pengguna dan Kata Sandi, ketik kredensial untuk pengguna utama yang Anda buat untuk instans DB.

Penyalinan massal

Fitur penyalinan massal SQL Server adalah cara efisien menyalin data dari basis data sumber ke instans DB Anda. Penyalinan massal menuliskan data yang Anda tentukan ke file data, seperti file ASCII. Anda kemudian dapat menjalankan penyalinan massal untuk menulis isi file ke instans DB tujuan.

Bagian ini menggunakan utilitas bcp, yang sudah ada dalam semua edisi SQL Server. Untuk informasi terperinci tentang operasi impor dan ekspor massal, lihat dokumentasi Microsoft SQL Server.

> **Note**
>
> Sebelum Anda menggunakan penyalinan massal, Anda harus mengimpor skema basis data Anda terlebih dahulu ke instans DB tujuan. Wizard Membuat dan Menerbitkan Skrip, yang dijelaskan sebelumnya dalam topik ini, adalah alat yang sangat baik untuk tujuan ini.

Perintah berikut terkoneksi ke instans SQL Server lokal. Perintah tersebut membuat file tab-delimited dari tabel yang ditentukan dalam direktori root C:\ dari deployment SQL Server Anda. Tabel ditentukan oleh nama yang sesuai syarat, dan file teks memiliki nama yang sama dengan tabel yang sedang disalin.

```
bcp dbname.schema_name.table_name out C:\table_name.txt -n -S localhost -U username -
P password -b 10000
```

Kode sebelumnya mencakup opsi berikut:

- `-n` menentukan bahwa penyalinan massal menggunakan tipe data native dari data yang akan disalin.
- `-S` menentukan instans SQL Server yang terkoneksi dengan utilitas bcp.
- `-U` menentukan nama pengguna akun untuk login ke instans SQL Server.
- `-P` menentukan kata sandi untuk pengguna yang ditentukan oleh `-U`.
- `-b` menentukan jumlah baris per batch dari data yang diimpor.

> **Note**
>
> Mungkin ada parameter lain yang penting untuk situasi impor Anda. Misalnya, Anda mungkin memerlukan parameter `-E` yang berkaitan dengan nilai identitas. Untuk informasi lebih lanjut; lihat deskripsi lengkap sintaks baris perintah untuk utilitas bcp di Dokumentasi Microsoft SQL Server.

Misalnya, anggaplah basis data bernama `store` yang menggunakan skema default, dbo, berisi tabel bernama `customers`. Akun pengguna `admin`, dengan kata sandi `insecure`, menyalin 10.000 baris tabel `customers` ke sebuah file bernama `customers.txt`.

```
bcp store.dbo.customers out C:\customers.txt -n -S localhost -U admin -P insecure -b
 10000
```

Setelah Anda membuat file data, Anda dapat mengunggah data ke instans DB Anda dengan menggunakan perintah serupa. Sebelumnya, buat basis data dan skema pada instans DB target. Lalu gunakan argumen `in` untuk menentukan file input, bukan `out` untuk menentukan file output. Dibanding menggunakan localhost untuk menentukan instans SQL Server lokal, tentukan endpoint instans DB Anda. Jika Anda menggunakan port selain 1433, sebutkan juga. Nama pengguna dan kata sandi adalah pengguna utama dan kata sandi untuk instans DB Anda. Sintaksnya adalah sebagai berikut.

```
bcp dbname.schema_name.table_name
     in C:\table_name.txt -n -S endpoint,port -U master_user_name -
P master_user_password -b 10000
```

Untuk melanjutkan contoh sebelumnya, anggaplah bahwa nama pengguna master adalah `admin`, dan kata sandinya adalah `insecure`. Titik akhir untuk instans DB adalah `rds.ckz2kqd4qsn1.us-east-1.rds.amazonaws.com`, dan Anda menggunakan port 4080. Perintah yang digunakan adalah sebagai berikut.

```
bcp store.dbo.customers in C:\customers.txt -n -S rds.ckz2kqd4qsn1.us-
east-1.rds.amazonaws.com,4080 -U admin -P insecure -b 10000
```

> **Note**
>
> Tentukan kata sandi selain prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

Mengekspor data dari RDS for SQL Server

Anda dapat memilih salah satu opsi berikut untuk mengekspor data dari instans DB RDS untuk SQL Server:

- Pencadangan native untuk basis data menggunakan file backup penuh (.bak) – Menggunakan .bak file untuk melakukan backup basis data sangatlah optimal, dan biasanya merupakan cara tercepat untuk mengekspor data. Untuk informasi selengkapnya, lihat Mengimpor dan mengekspor basis data SQL Server menggunakan pencadangan dan pemulihan native.
- Wizard Impor dan Ekspor pada SQL Server – Untuk informasi lebih lanjut, lihat Wizard Impor dan Ekspor pada SQL Server.
- Wizard Membuat dan Menerbitkan Skrip dan utilitas bcp SQL Server – Untuk informasi lebih lanjut, lihat Wizard Membuat dan Menerbitkan Skrip dan utilitas bcp pada SQL Server.

Wizard Impor dan Ekspor pada SQL Server

Anda dapat menggunakan Wizard Impor and Ekspor SQL Server untuk menyalin satu atau lebih tabel, tampilan, atau kueri dari instans DB RDS untuk SQL Server Anda ke penyimpanan data lain. Pilihan ini baik dilakukan jika penyimpanan data target bukan SQL Server. Untuk informasi selengkapnya, lihat Wizard Impor dan Ekspor SQL Server di dokumentasi SQL Server.

Wizard Impor dan Ekspor pada SQL Server tersedia sebagai bagian dari Microsoft SQL Server Management Studio. Klien SQL Server grafis ini tersedia dalam semua edisi Microsoft SQL Server kecuali Edisi Ekspres. SQL Server Management Studio hanya tersedia sebagai aplikasi berbasis Windows. SQL Server Management Studio Express tersedia dari Microsoft sebagai unduhan gratis. Untuk menemukan unduhan ini, lihat situs web Microsoft.

Untuk menggunakan Wizard Impor dan Ekspor pada SQL Server untuk ekspor data

1. Dalam SQL Server Management Studio, hubungkan ke instans DB RDS untuk SQL Server Anda. Untuk detail tentang cara melakukannya, lihat Menghubungkan ke instans DB yang menjalankan mesin basis data Microsoft SQL Server.

2. Dalam Penjelajah objek, perluas Basis data, buka menu konteks (klik kanan) untuk basis data sumber, pilih Tugas, lalu pilih Ekspor Data. Wizard akan muncul.
3. Di halaman Pilih Sumber Data, lakukan hal berikut:
   a. Untuk Sumber Data, pilih **`SQL Server Native Client 11.0`**.
   b. Verifikasi bahwa kotak Nama server menampilkan titik akhir instans DB RDS untuk SQL Server Anda.
   c. Pilih Gunakan Autentikasi SQL Server. Untuk Nama pengguna dan Kata sandi, ketik nama pengguna master dan kata sandi instans DB Anda.
   d. Verifikasi bahwa kotak Basis data menampilkan basis data tempat Anda ingin melakukan ekspor data.
   e. Pilih Selanjutnya.
4. Pada halaman Pilih tujuan, lakukan hal berikut ini:
   a. Untuk Tujuan, pilih **`SQL Server Native Client 11.0`**.

      > **Note**
      >
      > Sumber data target lainnya tersedia. Ini termasuk penyedia data .NET Framework, penyedia OLE DB, penyedia SQL Server Native Client, penyedia ADO.NET, Microsoft Office Excel, Microsoft Office Access, dan sumber Flat File. Jika Anda memilih untuk menargetkan salah satu sumber data ini, lewati sisa langkah 4. Untuk detail tentang informasi koneksi yang harus diberikan selanjutnya, lihat Pilih tujuan dalam dokumentasi SQL Server.

   b. Untuk Nama server, ketik nama server dari instans DB SQL Server target.
   c. Pilih jenis autentikasi yang sesuai. Ketik nama pengguna dan kata sandi jika perlu.
   d. Untuk Basis data, pilih nama basis data target, atau pilih Baru untuk membuat basis data baru untuk diisi data yang di-ekspor.

      Jika Anda memilih Baru, lihat Buat basis data dalam dokumentasi SQL Server untuk perincian tentang informasi basis data yang akan diberikan.
   e. Pilih Selanjutnya.
5. Di halaman Salinan atau Kueri Tabel, pilih Salin data dari satu atau lebih tabel atau tampilan atau Tulis kueri untuk menentukan data yang akan ditransfer. Pilih Selanjutnya.

6. Jika Anda memilih Tulis kueri untuk menentukan data yang akan ditransfer, Anda akan melihat halaman Berikan Kueri Sumber. Ketik atau tempel dalam kueri SQL, lalu pilih Parse untuk memverifikasinya. Setelah kueri memvalidasi, pilih Lanjut.
7. Di halaman Pilih Tabel dan Tampilan Sumber, lakukan hal berikut:
    a. Pilih tabel dan tampilan yang ingin Anda ekspor, atau verifikasi bahwa kueri yang Anda berikan sudah dipilih.
    b. Pilih Ubah Pemetaan dan tentukan basis data dan informasi pemetaan kolom. Untuk informasi lebih lanjut, lihat Pemetaan kolom dalam dokumentasi SQL Server.
    c. (Opsional) Untuk melihat pratinjau data yang akan diekspor, pilih tabel, tampilan, atau kueri, kemudian pilih Pratinjau.
    d. Pilih Selanjutnya.
8. Di halaman Jalankan Paket, pastikan bahwa Jalankan Segera dipilih. Pilih Selanjutnya.
9. Di halaman Selesaikan Wizard, pastikan bahwa detail ekspor data sudah seperti yang Anda harapkan. Pilih Selesai.
10. Di halaman Eksekusi berhasil, pilih Tutup.

Wizard Membuat dan Menerbitkan Skrip dan utilitas bcp pada SQL Server

Anda dapat menggunakan Wizard Membuat dan Menerbitkan Skrip SQL Server untuk membuat skrip untuk seluruh basis data atau hanya objek yang dipilih. Anda dapat menjalankan skrip ini di instans DB SQL Server target untuk membuat ulang objek yang terskrip. Anda kemudian dapat menggunakan utilitas bcp untuk melakukan bulk ekspor data untuk objek yang dipilih ke instans DB target. Pilihan ini baik dilakukan jika Anda ingin memindahkan seluruh basis data (termasuk objek selain tabel) atau data dalam jumlah besar antara dua instans DB SQL Server. Untuk penjelasan lengkap tentang sintaks baris perintah bcp, lihat utilitas bcp di dokumentasi Microsoft SQL Server.

Wizard Membuat dan Menerbitkan Skrip pada SQL Server tersedia sebagai bagian dari Microsoft SQL Server Management Studio. Klien SQL Server grafis ini tersedia dalam semua edisi Microsoft SQL Server kecuali Edisi Ekspres. SQL Server Management Studio hanya tersedia sebagai aplikasi berbasis Windows. SQL Server Management Studio Express tersedia dari Microsoft sebagai unduhan gratis.

Untuk menggunakan Wizard Membuat dan Menerbitkan Skrip serta utilitas bcp SQL Server untuk mengekspor data

1. Dalam SQL Server Management Studio, hubungkan ke instans DB RDS untuk SQL Server Anda. Untuk detail tentang cara melakukannya, lihat Menghubungkan ke instans DB yang menjalankan mesin basis data Microsoft SQL Server.
2. Pada Penjelajah Objek, perluas simpul Basis data dan pilih basis data yang ingin Anda skripkan.
3. Ikuti petunjuk di Wizard Membuat dan Menerbitkan Skrip dalam dokumentasi SQL Server untuk membuat file skrip.
4. Di SQL Server Management Studio, hubungkan ke instans DB SQL Server Anda.
5. Dengan instans DB SQL Server target yang dipilih di Penjelajah Objek, pilih Buka pada menu File, pilih File, lalu buka file skrip.
6. Jika Anda telah membuat skrip seluruh basis data, tinjau pernyataan CREATE basis data dalam skrip. Pastikan basis data dibuat di lokasi dan dengan parameter yang Anda inginkan. Untuk informasi selengkapnya, lihat MEMBUAT BASIS DATA di dokumentasi SQL Server.
7. Jika Anda membuat pengguna basis data dalam skrip, periksalah untuk melihat apakah login server ada pada instans DB target untuk pengguna tersebut. Jika tidak, buat login untuk pengguna tersebut; jika tidak, perintah terskrip untuk membuat basis data akan gagal. Untuk informasi selengkapnya, lihat Membuat login di dokumentasi SQL Server.
8. Pilih !Execute pada menu SQL Editor untuk menjalankan file skrip dan membuat objek basis data. Saat skrip selesai, verifikasi bahwa semua objek basis data yang ada sesuai dengan yang diharapkan.
9. Gunakan utilitas bcp untuk mengekspor data dari instans DB RDS untuk SQL Server ke dalam file. Buka prompt perintah dan ketik perintah berikut.

```
bcp database_name.schema_name.table_name out data_file -n -S aws_rds_sql_endpoint -
U username -P password
```

   Kode sebelumnya mencakup opsi berikut:

   - table_name adalah nama dari salah satu tabel yang telah Anda buat ulang di basis data target dan sekarang ingin diisi dengan data.
   - data_file adalah jalur lengkap dan nama file data yang akan dibuat.
   - `-n` menentukan bahwa penyalinan massal menggunakan tipe data native dari data yang akan disalin.

- `-S` menentukan instans DB SQL Server untuk ekspor.
- `-U` menentukan nama pengguna yang digunakan saat terkoneksi ke instans DB SQL Server.
- `-P` menentukan kata sandi untuk pengguna yang ditentukan oleh `-U`.

Bagian berikut menunjukkan contoh perintah.

```
bcp world.dbo.city out C:\Users\JohnDoe\city.dat -n -S sql-jdoe.1234abcd.us-
west-2.rds.amazonaws.com,1433 -U JohnDoe -P ClearTextPassword
```

Ulangi langkah ini hingga Anda memiliki file data untuk semua tabel yang ingin Anda ekspor.

10. Siapkan instans DB target Anda untuk impor data secara massal dengan mengikuti petunjuk di Panduan dasar untuk mengimpor data secara massal dalam dokumentasi SQL Server.
11. Tentukan metode impor massal yang akan digunakan setelah mempertimbangkan kinerja dan kekhawatiran lain yang dibahas dalam Tentang operasi impor massal dan ekspor massal dalam dokumentasi SQL Server.
12. Impor data secara massal dari file data yang Anda buat dengan menggunakan utilitas bcp. Untuk melakukannya, ikuti petunjuk di impor dan ekspor data massal dengan menggunakan utilitas bcp atau impor data massal dengan menggunakan BULK INSERT atau OPENROWSET(BULK...) di dokumentasi SQL Server, tergantung pada apa yang Anda putuskan di langkah 11.

Menggunakan replika baca untuk Microsoft SQL Server di Amazon RDS

Anda biasanya menggunakan replika baca untuk mengonfigurasi replikasi antara instans DB Amazon RDS. Untuk informasi umum tentang replika baca, lihat Menggunakan replika baca instans DB.

Di bagian ini, Anda dapat menemukan informasi spesifik tentang replika baca di Amazon RDS for SQL Server.

Topik

Mengonfigurasi replika baca untuk SQL Server

Sebelum instans DB dapat berfungsi sebagai instans sumber untuk replikasi, Anda harus mengaktifkan pencadangan otomatis pada instans DB sumber. Untuk melakukannya, atur periode retensi cadangan ke nilai selain 0. Instans DB sumber harus merupakan deployment Multi-AZ dengan Always On Availability Groups (AGs). Mengatur jenis deployment ini juga memberlakukan pencadangan otomatis yang diaktifkan.

Membuat replika baca SQL Server tidak memerlukan pemadaman listrik untuk instans DB primer. Amazon RDS menetapkan parameter dan izin yang diperlukan untuk instans DB sumber dan replika baca tanpa gangguan layanan. Snapshot diambil dari instans DB sumber dan menjadi replika baca. Tidak terjadi penghentian saat Anda menghapus replika baca.

Anda dapat membuat hingga 15 replika baca dari satu instans DB sumber. Agar replikasi dapat beroperasi secara efektif, kami menyarankan Anda mengonfigurasi setiap replika baca dengan jumlah sumber daya komputasi dan penyimpanan yang sama dengan instans DB sumber. Jika Anda menskalakan instans DB sumber, replika baca juga perlu diskalakan.

Versi mesin SQL Server DB dari instans DB sumber dan semua replika baca harus sama. Amazon RDS meningkatkan primer segera setelah meningkatkan replika baca, terlepas dari jendela

pemeliharaan. Untuk informasi selengkapnya tentang meningkatkan versi mesin DB, lihat Meng-upgrade mesin DB Microsoft SQL Server.

Agar replika baca dapat menerima dan menerapkan perubahan dari sumber, harus memiliki sumber daya komputasi dan penyimpanan yang memadai. Jika replika baca mencapai kapasitas komputasi, jaringan, atau sumber daya penyimpanan, replika baca akan berhenti menerima atau menerapkan perubahan dari sumbernya. Anda dapat memodifikasi sumber daya penyimpanan dan CPU dari replika baca secara terpisah dari sumbernya dan replika baca lainnya.

Batasan replika baca dengan SQL Server

Batasan berikut berlaku untuk replika baca SQL Server di Amazon RDS:

- Replika baca hanya tersedia pada mesin SQL Server Edisi Perusahaan (EE).
- Replika baca tersedia untuk SQL Server versi 2016–2022.
- Instans DB sumber yang akan direplikasi haruslah deployment Multi-AZ dengan Always On AG.
- Anda dapat membuat hingga 15 replika baca dari satu instans DB sumber. Replikasi mungkin tertinggal ketika instans DB sumber Anda memiliki lebih dari 5 replika baca.
- Replika baca hanya tersedia untuk instans DB yang berjalan di kelas instans DB dengan empat vCPU atau lebih.
- Berikut ini tidak didukung di Amazon RDS for SQL Server:
  - Penyimpanan cadangan replika baca
  - oint-in-time Pemulihan P dari replika baca
  - Tangkapan replika baca secara manual
  - Replika baca Multi-AZ
  - Membuat replika baca replika
  - Sinkronisasi login pengguna untuk membaca replika
- Amazon RDS for SQL Server tidak melakukan intervensi untuk mengurangi keterlambatan replika yang tinggi antara instans DB sumber dan replika baca-nya. Pastikan instans DB sumber dan replika pembacaannya berukuran benar, dalam hal daya komputasi dan penyimpanan, agar sesuai dengan beban operasionalnya.

Pertimbangan opsi untuk RDS untuk replika SQL Server

Sebelum Anda membuat replika RDS untuk SQL Server, pertimbangkan persyaratan, batasan, dan rekomendasi berikut:

- Jika replika SQL Server Anda berada di Wilayah yang sama dengan lokasi instans DB sumbernya, pastikan replika tersebut berasal dari grup opsi yang sama dengan lokasi instans DB sumber. Perubahan pada grup opsi sumber atau keanggotaan grup opsi sumber menyebar ke replika. Perubahan ini diterapkan ke replika segera setelah diterapkan ke instans DB sumber, terlepas dari masa pemeliharaan replika.

  Untuk informasi selengkapnya tentang grup opsi, lihat Menggunakan grup opsi.
- Saat Anda membuat replika lintas Wilayah SQL, Amazon RDS membuat grup opsi khusus untuk replika tersebut.

  Anda tidak dapat menghapus replika lintas Wilayah SQL dari grup opsi khususnya. Tidak ada instans DB lain yang dapat menggunakan grup opsi khusus untuk replika lintas Wilayah SQL Server.

  Opsi berikut adalah opsi yang direplikasi. Untuk menambahkan opsi replikasi ke replika lintas Wilayah SQL Server, tambahkan opsi tersebut ke grup opsi instans DB sumber. Opsi ini juga diinstal pada semua replika instans DB sumber.

  - `TDE`

  Opsi berikut adalah opsi yang direplikasi. Anda dapat menambahkan atau menghapus opsi non-replikasi dari grup opsi khusus.

  - `MSDTC`
  - `SQLSERVER_AUDIT`
  - Untuk mengaktifkan opsi `SQLSERVER_AUDIT` pada replika baca lintas wilayah, tambahkan opsi `SQLSERVER_AUDIT` pada grup opsi khusus pada replika baca lintas wilayah dan grup opsi instans sumber. Dengan menambahkan opsi `SQLSERVER_AUDIT` pada instans sumber replika baca Lintas wilayah SQL Server, Anda dapat membuat Objek Audit Tingkat Server dan Spesifikasi Audit Tingkat Server pada setiap replika baca lintas wilayah dari instans sumber. Untuk mengizinkan akses replika baca lintas wilayah untuk mengunggah log audit yang telah selesai ke bucket Amazon S3, tambahkan opsi `SQLSERVER_AUDIT` ke grup opsi khusus dan konfigurasikan pengaturan opsi. Bucket Amazon S3 yang Anda gunakan sebagai target untuk berkas audit harus berada di Wilayah yang sama dengan replika baca lintas Wilayah. Anda

dapat mengubah pengaturan opsi `SQLSERVER_AUDIT` untuk setiap replika baca lintas wilayah secara independen sehingga masing-masing dapat mengakses bucket Amazon S3 di Wilayah masing-masing.

Pilihan berikut tidak didukung untuk replika baca lintas Wilayah.

- `SSRS`
- `SSAS`
- `SSIS`

Opsi berikut didukung sebagian untuk replika baca lintas Wilayah.

- `SQLSERVER_BACKUP_RESTORE`
- Instans DB sumber dari replika SQL Server Lintas-wilayah dapat memiliki opsi `SQLSERVER_BACKUP_RESTORE`, tetapi Anda tidak dapat melakukan pemulihan native pada instans DB sumber sampai Anda menghapus semua replika Lintas wilayahnya. Setiap tugas pemulihan native yang ada akan dibatalkan selama pembuatan replika Lintas wilayah. Anda tidak dapat menambahkan opsi `SQLSERVER_BACKUP_RESTORE` ke grup opsi khusus.

  Untuk informasi lain tentang backup dan pemulihan native, lihat Mengimpor dan mengekspor basis data SQL Server menggunakan pencadangan dan pemulihan native

Saat Anda mempromosikan replika baca lintas Wilayah SQL, replika yang dipromosikan memiliki perilaku yang sama seperti instans DB SQL Server lainnya, termasuk manajemen opsinya. Untuk informasi selengkapnya tentang grup opsi, lihat Menggunakan grup opsi.

Menyelaraskan pengguna dan objek basis data dengan replika baca SQL Server

Setiap login, peran server kustom, pekerjaan agen SQL, atau objek tingkat server lainnya yang ada di instans DB utama pada saat membuat replika baca diharapkan hadir dalam replika baca yang baru dibuat. Namun, objek tingkat server apa pun yang dibuat dalam instans DB utama setelah pembuatan replika baca tidak akan direplikasi secara otomatis, dan Anda harus membuatnya secara manual di replika baca.

Pengguna basis data secara otomatis direplikasi dari instans DB primer ke replika baca. Karena basis data replika baca dalam mode hanya-baca, pengidentifikasi keamanan (SID) dari pengguna basis data tidak dapat diperbarui dalam basis data. Oleh karena itu, saat membuat login SQL di replika baca, penting untuk memastikan bahwa SID dari login itu cocok dengan SID dari login SQL yang

sesuai di instans DB utama. Jika Anda tidak menyinkronkan SID dari login SQL, mereka tidak akan dapat mengakses basis data dalam replika baca. Windows Active Directory (AD) Authenticated Login tidak mengalami masalah ini karena SQL Server memperoleh SID dari Active Directory.

Untuk menyinkronkan login SQL dari instans DB primer ke replika baca

1. Hubungkan ke instans DB utama.
2. Buat login SQL baru di instans DB primer.

```
USE [master]
GO
CREATE LOGIN TestLogin1
WITH PASSWORD = 'REPLACE WITH PASSWORD';
```

> **Note**
>
> Tentukan kata sandi selain prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

3. Buat pengguna basis data baru untuk login SQL di basis data.

```
USE [REPLACE WITH YOUR DB NAME]
GO
CREATE USER TestLogin1 FOR LOGIN TestLogin1;
GO
```

4. Periksa SID dari login SQL yang baru dibuat di instans DB primer.

```
SELECT name, sid FROM sys.server_principals WHERE name =  TestLogin1;
```

5. Hubungkan ke replika baca. Buat login SQL baru.

```
CREATE LOGIN TestLogin1 WITH PASSWORD = 'REPLACE WITH PASSWORD', SID=[REPLACE WITH
 sid FROM STEP #4];
```

Sebagai alternatif, jika Anda memiliki akses ke basis data replika baca, Anda dapat memperbaiki pengguna yatim piatu sebagai berikut:

1. Hubungkan ke replika baca.

2. Identifikasi pengguna yatim piatu dalam basis data.

```
USE [REPLACE WITH YOUR DB NAME]
GO
EXEC sp_change_users_login 'Report';
GO
```

3. Buat login SQL baru untuk pengguna basis data yatim piatu.

```
CREATE LOGIN TestLogin1 WITH PASSWORD = 'REPLACE WITH PASSWORD', SID=[REPLACE WITH
 sid FROM STEP #2];
```

Contoh:

```
CREATE LOGIN TestLogin1 WITH PASSWORD = 'TestPa$$word#1',
 SID=[0x1A2B3C4D5E6F7G8H9I0J1K2L3M4N5O6P];
```

> **Note**
>
> Tentukan kata sandi selain prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

Pemecahan Masalah batasan replika baca SQL Server

Anda dapat memantau kelambatan replikasi di Amazon CloudWatch dengan melihat metrik Amazon RDS. `ReplicaLag` Untuk informasi tentang waktu keterlambatan replika, lihat Memantau replikasi baca.

Jika lag replikasi terlalu panjang, Anda dapat menggunakan kueri berikut untuk mendapatkan informasi tentang keterlambatan.

```
SELECT AR.replica_server_name
     , DB_NAME (ARS.database_id) 'database_name'
     , AR.availability_mode_desc
     , ARS.synchronization_health_desc
     , ARS.last_hardened_lsn
     , ARS.last_redone_lsn
     , ARS.secondary_lag_seconds
FROM sys.dm_hadr_database_replica_states ARS
```

```
INNER JOIN sys.availability_replicas AR ON ARS.replica_id = AR.replica_id
--WHERE DB_NAME(ARS.database_id) = 'database_name'
ORDER BY AR.replica_server_name;
```

Deployment Multi-AZ untuk Amazon RDS for Microsoft SQL Server

Deployment Multi-AZ memberikan ketersediaan, durabilitas data, dan toleransi kesalahan yang lebih tinggi untuk instans basis data. Jika terjadi pemeliharaan basis data yang direncanakan atau gangguan layanan yang tidak direncanakan, Amazon RDS secara otomatis gagal ke instans DB up-to-date sekunder. Fungsi ini memungkinkan operasi basis data berlanjut dengan cepat tanpa gangguan manual. Instans primer dan siaga menggunakan titik akhir yang sama, yang alamat jaringan fisiknya beralih ke replika sekunder sebagai bagian dari proses failover. Anda tidak perlu mengonfigurasi ulang aplikasi Anda saat terjadi failover.

Amazon RDS mendukung deployment Multi-AZ untuk Microsoft SQL Server menggunakan Pencerminan Basis Data (DBM) atau Grup Ketersediaan (AG) Selalu Aktif SQL Server. Amazon RDS memantau dan menjaga kondisi deployment Multi-AZ Anda. Jika terjadi masalah, RDS secara otomatis memperbaiki instans DB yang tidak berkondisi baik, menetapkan ulang sinkronisasi, dan memulai failover. Failover hanya terjadi jika replika siaga dan primer disinkronkan sepenuhnya. Anda tidak perlu mengatur apa pun.

Saat Anda menyiapkan Multi-AZ SQL Server, RDS secara otomatis mengonfigurasi semua basis data pada instans untuk menggunakan DBM atau AG. Amazon RDS menangani instans DB primer, saksi, dan sekunder untuk Anda. Karena konfigurasinya otomatis, RDS memilih DBM atau AG Selalu Aktif berdasarkan versi SQL Server yang Anda deploy.

Amazon RDS mendukung Multi-AZ dengan AG Selalu Aktif untuk versi dan edisi SQL Server berikut:

- SQL Server 2022:
  - Standard Edition
  - Enterprise Edition
- SQL Server 2019:
  - Standard Edition 15.00.4073.23 dan lebih tinggi
  - Enterprise Edition
- SQL Server 2017:
  - Standard Edition 14.00.3401.7 dan lebih tinggi
  - Enterprise Edition 14.00.3049.1 dan lebih baru
- SQL Server 2016: Enterprise Edition 13.00.5216.0 dan lebih tinggi

Amazon RDS mendukung Multi-AZ dengan DBM untuk versi dan edisi SQL Server berikut, kecuali untuk versi yang disebutkan sebelumnya:

- SQL Server 2019: Standard Edition 15.00.4043.16
- SQL Server 2017: Standard Edition dan Enterprise Edition
- SQL Server 2016: Standard Edition dan Enterprise Edition
- SQL Server 2014: Standard Edition dan Enterprise Edition

Anda dapat menggunakan kueri SQL berikut untuk menentukan apakah instans DB SQL Server Anda adalah AZ Tunggal, Multi-AZ dengan DBM, atau Multi-AZ dengan AG Selalu Aktif:

```
SELECT CASE WHEN dm.mirroring_state_desc IS NOT NULL THEN 'Multi-AZ (Mirroring)'
    WHEN dhdrs.group_database_id IS NOT NULL THEN 'Multi-AZ (AlwaysOn)'
    ELSE 'Single-AZ'
    END 'high_availability'
FROM sys.databases sd
LEFT JOIN sys.database_mirroring dm ON sd.database_id = dm.database_id
LEFT JOIN sys.dm_hadr_database_replica_states dhdrs ON sd.database_id =
 dhdrs.database_id AND dhdrs.is_local = 1
WHERE DB_NAME(sd.database_id) = 'rdsadmin';
```

Output-nya seperti berikut:

```
high_availability
Multi-AZ (AlwaysOn)
```

Menambahkan Multi-AZ ke instans DB Microsoft SQL Server

Saat Anda membuat instance SQL Server DB baru menggunakan AWS Management Console, Anda dapat menambahkan Multi-AZ dengan Database Mirroring (DBM) atau Always On AG. Anda melakukannya dengan memilih Ya (Pencerminan/Selalu Aktif) dari Deployment Multi-AZ. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.

Saat Anda mengubah instans DB SQL Server yang ada menggunakan konsol, Anda dapat menambahkan Multi-AZ dengan DBM atau AG dengan memilih Ya (Pencerminan/Selalu Aktif) dari daftar Deployment Multi-AZ di halaman Modifikasi instans DB. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

> **Note**
>
> Jika instans DB Anda menjalankan Pencerminan Basis Data (DBM)—bukan Grup Ketersediaan (AG) Selalu Aktif—Anda mungkin perlu menonaktifkan optimisasi dalam memori sebelum menambahkan Multi-AZ. Nonaktifkan optimisasi dalam memori sebelum menambahkan Multi-AZ jika instans DB Anda menjalankan SQL Server 2014, 2016, atau 2017 Enterprise Edition dan telah mengaktifkan optimisasi dalam memori.
> Jika instans DB Anda sedang menjalankan AG, langkah ini tidak diperlukan.

Menghapus Multi-AZ dari instans DB Microsoft SQL Server

Saat Anda memodifikasi instans SQL Server DB yang ada menggunakan AWS Management Console, Anda dapat menghapus Multi-AZ dengan DBM atau AG. Anda dapat melakukannya dengan memilih Tidak (Pencerminan/Selalu Aktif) dari Deployment Multi-AZ di halaman Modifikasi instans DB. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Batasan, catatan dan rekomendasi deployment Multi-AZ Microsoft SQL Server

Berikut ini adalah beberapa batasan saat melakukan deployment Multi-AZ di instans DB SQL Server:

- Multi-AZ Lintas Wilayah tidak didukung.
- Menghentikan instans DB RDS for SQL Server dalam deployment Multi-AZ tidak didukung.
- Anda tidak dapat mengonfigurasi instans DB sekunder untuk menerima aktivitas baca basis data.
- Multi-AZ dengan Grup Ketersediaan (AG) Selalu Aktif mendukung optimisasi dalam memori.
- Multi-AZ dengan Grup Ketersediaan (AG) Selalu Aktif tidak mendukung autentikasi Kerberos untuk pendengar grup ketersediaan. Hal ini karena pendengar tidak memiliki Nama Prinsipal Layanan (SPN).
- Anda tidak dapat mengubah nama basis data di instans DB SQL Server dalam deployment Multi-AZ SQL Server. Jika Anda perlu mengubah nama basis data pada instans tersebut, pertama nonaktifkan Multi-AZ untuk instans DB, kemudian ubah nama basis datanya. Terakhir, aktifkan kembali Multi-AZ untuk instans DB.
- Anda hanya dapat memulihkan instans DB Multi-AZ yang dicadangkan menggunakan model pemulihan penuh.

- Deployment Multi-AZ memiliki batas 10.000 pekerjaan SQL Server Agent.

  Jika Anda membutuhkan batas yang lebih tinggi, minta kenaikan dengan menghubungi AWS Support. Buka halaman Pusat AWS Support, masuk jika perlu, dan pilih Buat kasus. Pilih Peningkatan batas layanan. Lengkapi dan kirimkan formulir.

Berikut ini adalah beberapa catatan tentang penggunaan deployment Multi-AZ di instans DB SQL Server:

- Amazon RDS mengekspos titik akhir pendengar grup ketersediaan AG Selalu Aktif. Titik akhir terlihat di konsol, dan ditampilkan oleh operasi API `DescribeDBInstances` sebagai entri di bidang titik akhir.
- Amazon RDS mendukung failover multisubnet grup ketersediaan.
- Untuk menggunakan Multi-AZ SQL Server dengan instans DB SQL Server dalam cloud privat virtual (VPC), pertama, buat grup subnet DB yang memiliki subnet di setidaknya dua Zona Ketersediaan yang berbeda. Kemudian, tetapkan grup subnet DB menjadi replika primer instans DB SQL Server.
- Ketika instans DB diubah menjadi deployment Multi-AZ, selama modifikasi, statusnya adalah mengubah. Amazon RDS membuat replika siaga dan membuat cadangan instans DB primer. Setelah proses tersebut selesai, status instans DB primer menjadi tersedia.
- Deployment Multi-AZ mempertahankan semua basis data pada simpul yang sama. Jika terjadi failover pada basis data di host primer, semua basis data SQL Server Anda akan melakukan failover sebagai satu unit atomis ke host siaga. Amazon RDS menyediakan host baru yang berkondisi baik dan mengganti host yang tidak berkondisi baik.
- Multi-AZ dengan DBM atau AG mendukung replika siaga tunggal.
- Pengguna, login, dan izin akan direplikasi secara otomatis untuk Anda pada instans sekunder. Anda tidak perlu membuat ulang hal ini. Peran server yang ditentukan pengguna hanya direplikasi dalam instans DB yang menggunakan AG Selalu Aktif untuk deployment Multi-AZ.
- Dalam penerapan Multi-AZ, RDS untuk SQL Server membuat login SQL Server untuk memungkinkan Always On AG atau Pencerminan Database. RDS membuat login dengan pola berikut,, `db_<dbiResourceId>_node1_login`db_<dbiResourceId>_node2_login, dan. `db_<dbiResourceId>_witness_login`
- RDS untuk SQL Server membuat login SQL Server untuk memungkinkan akses membaca replika. RDS membuat login dengan pola berikut,`db_<readreplica_dbiResourceId>_node_login`.

- Dalam deployment Multi-AZ, pekerjaan SQL Server Agent direplikasi dari host utama ke host sekunder saat fitur replikasi pekerjaan diaktifkan. Untuk informasi selengkapnya, lihat Mengaktifkan replikasi pekerjaan SQL Server Agent.
- Anda mungkin menemukan adanya peningkatan latensi dibandingkan dengan deployment instans DB standar (dalam satu Zona Ketersediaan) karena replikasi data yang sinkron.
- Waktu failover dipengaruhi oleh waktu yang diperlukan untuk menyelesaikan proses pemulihan. Transaksi besar akan meningkatkan waktu failover.
- Dalam deployment Multi-AZ SQL Server, boot ulang dengan failover akan mem-boot ulang hanya instans DB primer. Setelah failover, instans DB primer akan menjadi instans DB sekunder baru. Parameter mungkin tidak diperbarui untuk instans Multi-AZ. Untuk mem-boot ulang tanpa failover, instans DB primer dan sekunder, serta parameter diperbarui setelah boot ulang. Jika instans DB tidak responsif, kami sarankan boot ulang tanpa failover.

Berikut ini adalah beberapa rekomendasi penggunaan deployment Multi-AZ di instans DB Microsoft SQL Server:

- Untuk basis data yang digunakan dalam produksi atau praproduksi, kami merekomendasikan opsi berikut:
  - Deployment Multi-AZ untuk ketersediaan tinggi
  - "IOPS yang Tersedia" untuk performa yang cepat dan konsisten
  - "Memori dioptimalkan" alih-alih "Tujuan umum"
- Anda tidak dapat memilih Zona Ketersediaan (AZ) untuk instans sekunder, jadi ketika Anda melakukan deployment host aplikasi, pertimbangkan hal ini. Basis data Anda mungkin akan melakukan failover ke AZ lain, dan host aplikasi mungkin tidak berada di AZ yang sama dengan basis data. Untuk alasan ini, kami menyarankan Anda menyeimbangkan host aplikasi Anda di semua AZ di AWS Wilayah tertentu.
- Untuk performa terbaik, jangan aktifkan Pencerminan Basis Data atau AG Selalu Aktif selama operasi pemuatan data besar. Jika Anda ingin pemuatan data secepat mungkin, selesaikan pemuatan data sebelum mengonversi instans DB ke deployment Multi-AZ.
- Aplikasi yang mengakses basis data SQL Server harus memiliki penanganan pengecualian yang mengambil kesalahan koneksi. Sampel kode berikut menunjukkan try/catch block yang mengambil kesalahan komunikasi. Dalam contoh ini, pernyataan `break` keluar dari loop `while` jika koneksi berhasil, tetapi mencoba ulang hingga 10 kali jika pengecualian ditampilkan.

```
int RetryMaxAttempts = 10;
```

```
int RetryIntervalPeriodInSeconds = 1;
int iRetryCount = 0;
while (iRetryCount < RetryMaxAttempts)
{
   using (SqlConnection connection = new SqlConnection(DatabaseConnString))
   {
      using (SqlCommand command = connection.CreateCommand())
      {
         command.CommandText = "INSERT INTO SOME_TABLE VALUES ('SomeValue');";
         try
         {
            connection.Open();
            command.ExecuteNonQuery();
            break;
         }
         catch (Exception ex)
         {
            Logger(ex.Message);
            iRetryCount++;
         }
         finally {
            connection.Close();
         }
      }
   }
   Thread.Sleep(RetryIntervalPeriodInSeconds * 1000);
}
```

- Jangan gunakan perintah `Set Partner Off` saat menggunakan instans Multi-AZ. Misalnya, jangan melakukan hal berikut.

```
--Don't do this
ALTER DATABASE db1 SET PARTNER off
```

- Jangan tetapkan mode pemulihan ke `simple`. Misalnya, jangan melakukan hal berikut.

```
--Don't do this
ALTER DATABASE db1 SET RECOVERY simple
```

- Jangan gunakan parameter `DEFAULT_DATABASE` saat membuat login baru pada instans DB Multi-AZ karena pengaturan ini tidak dapat diterapkan pada instans cerminan siaga. Misalnya, jangan melakukan hal berikut.

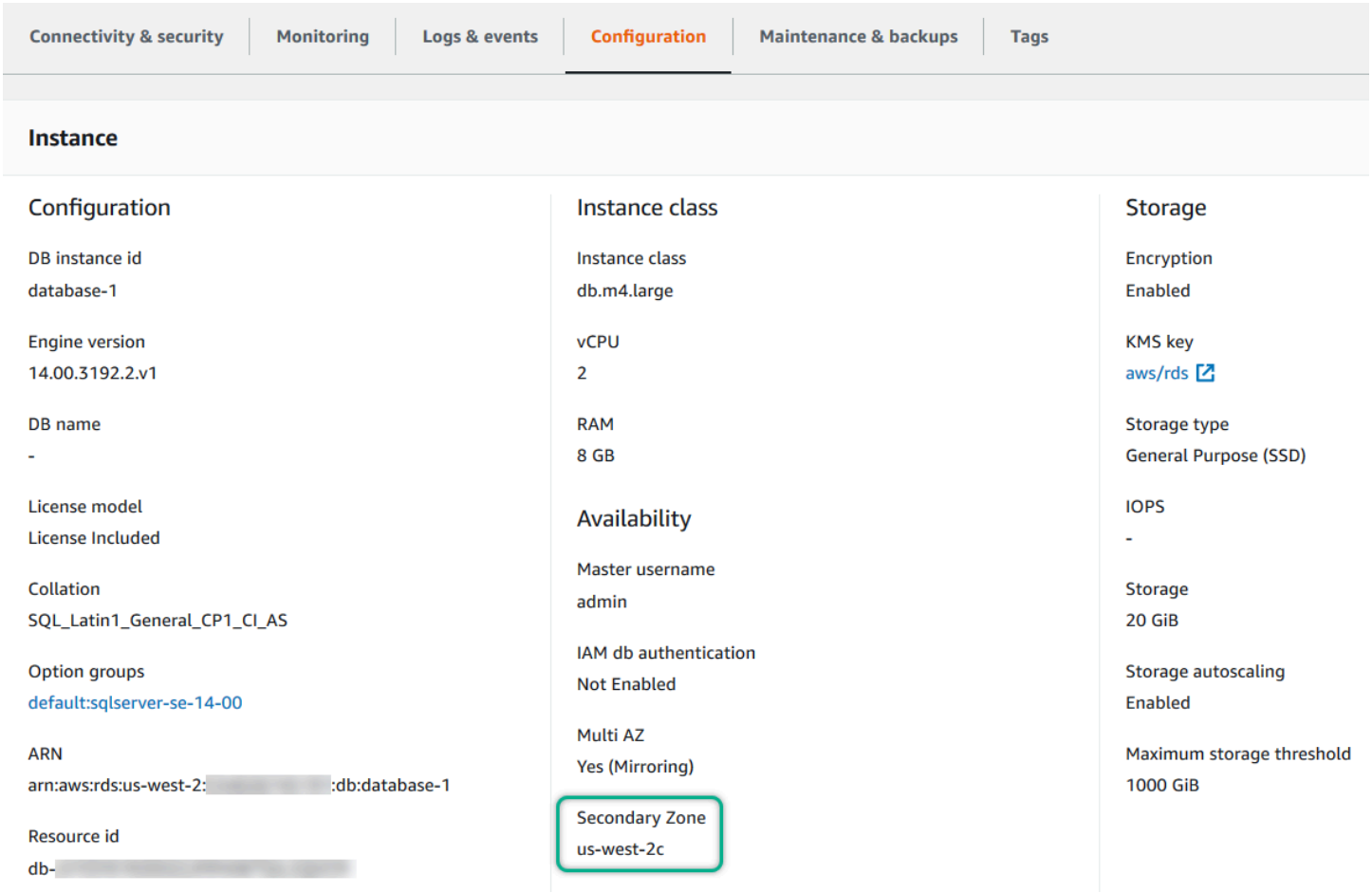
```
--Don't do this
CREATE LOGIN [test_dba] WITH PASSWORD=foo, DEFAULT_DATABASE=[db2]
```

Selain itu, jangan melakukan hal berikut.

```
--Don't do this
ALTER LOGIN [test_dba] SET DEFAULT_DATABASE=[db3]
```

Menentukan lokasi instans sekunder

Anda dapat menentukan lokasi replika sekunder dengan menggunakan AWS Management Console. Anda harus mengetahui lokasi instans sekunder jika Anda menyiapkan instans DB primer Anda dalam VPC.



Anda juga dapat melihat Availability Zone sekunder menggunakan AWS CLI perintah `describe-db-instances` atau operasi `DescribeDBInstances` RDS API. Output ini menunjukkan AZ sekunder tempat instans cerminan siaga berada.

Bermigrasi dari Pencerminan Basis Data ke Grup Ketersediaan Selalu Aktif

Di Microsoft SQL Server Enterprise Edition versi 14.00.3049.1, Grup Ketersediaan (AG) Selalu Aktif diaktifkan secara default.

Untuk bermigrasi dari Pencerminan Basis Data (DBM) ke AG, pertama-tama periksa versi Anda. Jika Anda menggunakan instans DB dengan versi sebelum Enterprise Edition 13.00.5216.0, ubah instans tersebut agar di-patch ke versi 13.00.5216.0 atau yang lebih baru. Jika Anda menggunakan instans DB dengan versi sebelum Enterprise Edition 14.00.3049.1, ubah instans tersebut agar di-patch ke versi 14.00.3049.1 atau yang lebih baru.

Jika Anda ingin meningkatkan instans DB yang dicerminkan agar menggunakan AG, jalankan peningkatan terlebih dahulu, ubah instans tersebut untuk menghapus Multi-AZ, lalu ubah kembali untuk menambahkan Multi-AZ. Tindakan ini akan mengonversi instans Anda untuk menggunakan AG Selalu Aktif.

Fitur tambahan untuk Microsoft SQL di Amazon RDS

Pada bagian berikut, Anda dapat menemukan informasi tentang penambahan instans Amazon RDS yang menjalankan mesin DB Microsoft SQL Server.

Topik

Menggunakan SSL dengan instans DB Microsoft SQL Server

Anda dapat menggunakan Secure Sockets Layer (SSL) untuk mengenkripsi koneksi antara aplikasi klien Anda dan instans DB Amazon RDS yang menjalankan Microsoft SQL Server. Dukungan SSL tersedia di semua wilayah AWS untuk semua edisi SQL Server yang didukung.

Saat Anda membuat instans DB SQL Server, Amazon RDS membuat sertifikat SSL untuk instans DB tersebut. Sertifikat SSL menyertakan titik akhir instans DB sebagai Nama Umum (CN) untuk sertifikat SSL guna melindungi dari serangan spoofing.

Ada 2 cara menggunakan SSL untuk terhubung ke instans DB SQL Server Anda:

- Paksa SSL untuk semua koneksi — ini terjadi secara transparan pada klien, dan klien tidak perlu melakukan pekerjaan apa pun untuk menggunakan SSL.
- Enkripsi koneksi tertentu — ini menyiapkan koneksi SSL dari komputer klien tertentu, dan Anda harus melakukan pekerjaan pada klien untuk mengenkripsi koneksi.

Untuk informasi tentang dukungan Transport Layer Security (TLS) untuk SQL Server, lihat Dukungan TLS 1.2 untuk Microsoft SQL Server.

Memaksa koneksi ke instans DB Anda untuk menggunakan SSL

Anda dapat memaksa semua koneksi ke instans DB Anda untuk menggunakan SSL. Jika Anda memaksa koneksi untuk menggunakan SSL, hal ini terjadi secara transparan pada klien, dan klien tidak perlu melakukan tindakan apa pun untuk menggunakan SSL.

Jika Anda ingin memaksa SSL, gunakan parameter `rds.force_ssl`. Secara default, parameter `rds.force_ssl` diatur ke `0 (off)`. Atur parameter `rds.force_ssl` ke `1 (on)` untuk memaksa koneksi untuk menggunakan SSL. Parameter `rds.force_ssl` bersifat statis, sehingga setelah Anda mengubah nilai, Anda harus me-reboot instans DB Anda agar perubahan dapat berlaku.

Anda dapat memaksa semua koneksi ke instans DB Anda untuk menggunakan SSL

1. Tentukan grup parameter yang dilampirkan ke instans DB Anda:
    a. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
    b. Di sudut kanan atas konsol Amazon RDS, pilih Wilayah AWS instans DB Anda.

c. Di panel navigasi, pilih Basis data, lalu pilih nama instans DB Anda untuk memperlihatkan detailnya.

d. Pilih tab Konfigurasi. Temukan Grup parameter di bagian tersebut.

2. Jika perlu, buat grup parameter baru. Jika instans DB Anda menggunakan grup parameter default, Anda harus membuat grup parameter baru. Jika instans DB Anda menggunakan grup parameter nondefault, Anda dapat memilih untuk mengedit grup parameter yang ada atau untuk membuat grup parameter baru. Jika Anda mengedit grup parameter yang sudah ada, perubahan akan memengaruhi semua instans DB yang menggunakan grup parameter tersebut.

   Untuk membuat grup parameter baru, ikuti petunjuk di Membuat grup parameter DB.

3. Mengedit grup parameter baru atau yang sudah ada untuk mengatur parameter `rds.force_ssl` ke `true`. Untuk mengedit grup parameter, ikuti petunjuk di Memodifikasi parameter dalam grup parameter DB.

4. Jika Anda membuat grup parameter baru, ubah instans DB Anda untuk melampirkan grup parameter baru. Ubah pengaturan Grup Parameter DB dari instans DB. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

5. Reboot instans DB Anda. Untuk informasi selengkapnya, lihat Mem-boot ulang instans DB.

Mengenkripsi koneksi spesifik

Anda dapat memaksakan semua koneksi ke instans DB Anda untuk menggunakan SSL, atau Anda dapat mengenkripsi koneksi dari komputer klien tertentu saja. Untuk menggunakan SSL dari klien tertentu, Anda harus mendapatkan sertifikat untuk komputer klien, mengimpor sertifikat di komputer klien, kemudian mengenkripsi koneksi dari komputer klien.

> **Note**
>
> Semua instans SQL Server yang dibuat setelah 5 Agustus 2014, menggunakantitik akhir instans DB dalam bidang Common Name (CN) pada sertifikat SSL. Sebelum 5 Agustus 2014, verifikasi sertifikat SSL tidak tersedia untuk instans SQL Server berbasis VPC. Jika Anda memiliki instans DB SQL Server berbasis VPC yang dibuat sebelum 5 Agustus 2014, dan Anda ingin menggunakan verifikasi sertifikat SSL dan memastikan bahwa titik akhir instans disertakan sebagai CN untuk sertifikat SSL untuk instans DB tersebut, maka ubah nama instans tersebut. Saat Anda mengubah nama pada suatu instans DB, sertifikat baru akan diterapkan dan instans akan direboot untuk mengaktifkan sertifikat baru.

Mendapatkan sertifikat untuk komputer klien

Untuk mengenkripsi koneksi dari komputer klien ke instansDB Amazon RDS yang menjalankan Microsoft SQL Server, Anda memerlukan sertifikat di komputer klien Anda.

Untuk mendapatkan sertifikat tersebut, unduh sertifikat ke komputer klien Anda. Anda dapat mengunduh sertifikat root yang berfungsi untuk semua wilayah. Anda juga dapat mengunduh paket sertifikat yang berisi sertifikat root yang lama dan baru. Selain itu, Anda juga dapat mengunduh sertifikat menengah khusus wilayah. Untuk informasi selengkapnya tentang cara mengunduh sertifikat, lihat .

Setelah Anda mengunduh sertifikat yang sesuai, impor sertifikat ke sistem operasi Microsoft Windows dengan mengikuti prosedur di bagian berikut.

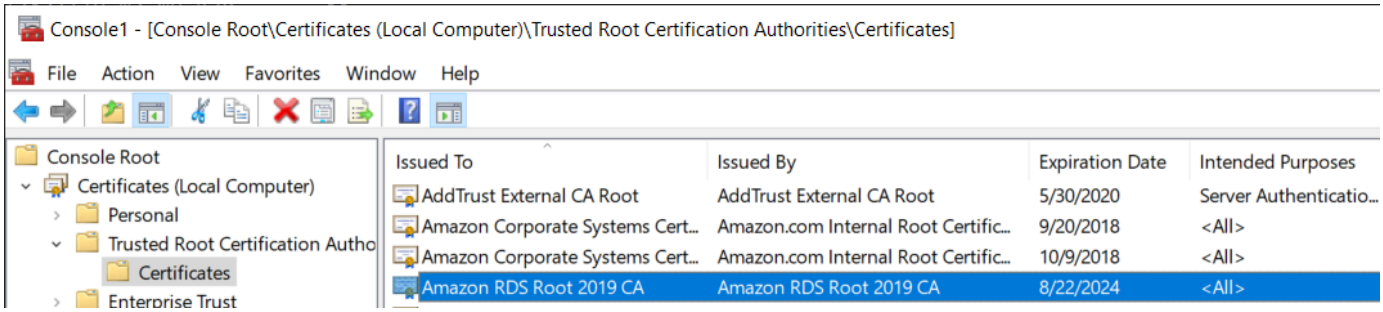
Mengimpor sertifikat untuk komputer klien

Anda dapat menggunakan prosedur berikut untuk mengimpor sertifikat Anda ke dalam sistem operasi Microsoft Windows di komputer klien Anda.

Impor sertifikat ke dalam sistem operasi Windows Anda:

1. Pada menu Mulai, ketik **Run** di kotak pencarian dan tekan Masuk.
2. Pada kotak Buka , ketik **MMC** lalu pilih OK.
3. Pada konsol MMC, pada menu File, pilih Tambah/Hapus Snap-in.
4. Pada kotak dialog Tambah atau Hapus Snap-in, untuk Snap-in yang tersedia, pilih **Certificates**, lalu pilih Tambahkan.
5. Pada kotak dialog Sertifikat snap-in, pilih Akun komputer, lalu pilih Selanjutnya.
6. Pada kotak dialog Pilih komputer, pilih Selesai.
7. Pada kotak dialog Tambah atau Hapus Snap-in, pilih OK.
8. Pada konsol MMC, perluas Sertifikat, buka menu konteks (klik kanan) untuk Otorisasi Sertifikasi Root Tepercaya, pilih Semua Tugas, lalu pilih Impor.
9. Di halaman pertama Certificate Import Wizard, pilih Selanjutnya.
10. Di halaman kedua Certificate Import Wizard, pilih Jelajahi. Pada jendela jelajah, ubah jenis file ke Semua file (*.*) karena .pem bukanlah esktensi sertifikat standar. Cari file .pem yang Anda unduh sebelumnya.
11. Pilih Buka untuk memilih file sertifikat, lalu pilih Selanjutnya.

12. Di halaman ketiga Certificate Import Wizard, pilih Selanjutnya.
13. Di halaman keempat Certificate Import Wizard, pilih Selanjutnya. Kotak dialog muncul yang menunjukkan bahwa impor berhasil.
14. Pada konsol MMC, perluas Sertifikat, perluas Otorisasi Sertifikasi Root Tepercaya, lalu pilih Sertifikat. Cari sertifikat untuk mengonfirmasi bahwa sertifikat itu ada, seperti yang ditunjukkan di sini.



Mengenkripsi koneksi ke instans DB Amazon RDS yang menjalankan Microsoft SQL Server

Setelah Anda mengimpor sertifikat ke komputer klien, Anda dapat mengenkripsi koneksi dari komputer klien ke instans DB Amazon RDS yang menjalankan Microsoft SQL Server.

Untuk SQL Server Management Studio, gunakan prosedur berikut. Untuk informasi selengkapnya tentang SQL Server Management Studio, lihat Menggunakan SQL Server Management Studio.

Untuk mengenkripsi koneksi dari SQL Server Management Studio

1. Luncurkan SQL Server Management Studio.
2. Untuk Terhubung ke server, ketikkan informasi server, nama pengguna login, dan kata sandi.
3. Pilih Opsi.
4. Pilih Enkripsikan koneksi.
5. Pilih Hubungkan.
6. Konfirmasikan bahwa koneksi Anda dienkripsi dengan menjalankan kueri berikut. Verifikasi bahwa kueri kembali `true` untuk `encrypt_option`.

```
select ENCRYPT_OPTION from SYS.DM_EXEC_CONNECTIONS where SESSION_ID = @@SPID
```

Untuk klien SQL lain, gunakan prosedur berikut.

Untuk mengenkripsi koneksi dari klien SQL lainnya

1. Tambahkan `encrypt=true` ke string koneksi Anda. String ini mungkin tersedia sebagai opsi, atau sebagai properti di halaman koneksi di alat GUI.

   > **Note**
   >
   > Untuk mengaktifkan enkripsi SSL untuk klien yang terhubung menggunakan JDBC, Anda mungkin perlu menambahkan sertifikat SQL Amazon RDS ke penyimpanan sertifikat Java CA (cacerts). Anda dapat melakukan ini dengan menggunakan utilitas alat kunci.

2. Konfirmasikan bahwa koneksi Anda dienkripsi dengan menjalankan kueri berikut. Verifikasi bahwa kueri kembali `true` untuk `encrypt_option`.

```
select ENCRYPT_OPTION from SYS.DM_EXEC_CONNECTIONS where SESSION_ID = @@SPID
```

Mengonfigurasi protokol keamanan dan cipher

Anda dapat mengaktifkan dan menonaktifkan protokol keamanan dan cipher tertentu menggunakan parameter DB. Parameter keamanan yang dapat Anda konfigurasi (kecuali untuk TLS versi 1.2) ditampilkan dalam tabel berikut.

Untuk parameter selain `rds.fips`, nilai dari `default` berarti bahwa nilai default sistem operasi digunakan, baik itu `enabled` atau `disabled`.

> **Note**
>
> Anda tidak dapat menonaktifkan TLS 1.2, karena Amazon RDS menggunakannya secara internal.

Parameter DB	Nilai yang diizinkan (default dalam huruf tebal)	Deskripsi
rds.tls10	default, diaktifkan, dinonaktifkan	TLS 1.0.
rds.tls11	default, diaktifkan, dinonaktifkan	TLS 1.1.
rds.tls12	default	TLS 1.2. Anda tidak dapat mengubah nilai ini.
rds.fips	0, 1	Saat Anda menetapkan parameter ke 1, RDS memaksa penggunaan modul yang sesuai dengan Standar Pemrosesan Informasi Federal (FIPS) 140-2. Untuk informasi selengkapnya, lihat Menggunakan SQL Server 2016 dalam mode yang sesuai dengan FIPS 140-2 dalam dokumentasi Microsoft.

Parameter DB	Nilai yang diizinkan (default dalam huruf tebal)	Deskripsi
		Note Anda harus reboot instans DB setelah modifikasi untuk merasakan dampaknya.
rds.rc4	default, diaktifkan, dinonaktifkan	RC4 stream cipher.
rds.diffie-hellman	default, diaktifkan, dinonaktifkan	Enkripsi pertukaran kunci Diffie-Hellman.
rds. diffie-hellman-min-key-bit-panjang	default, 1024, 2048, 4096	Panjang bit minimum untuk kunci Diffie-Hellman.
rds.curve25519	default, diaktifkan, dinonaktifkan	Enkripsi cipher kurva-elips Curve25519. Parameter ini tidak mendukung semua versi mesin.
rds.3des168	default, diaktifkan, dinonaktifkan	Cipher enkripsi Standar Enkripsi Data Tiga Kali Lipat (DES) memiliki panjang kunci 168-bit.

Note

Untuk informasi selengkapnya tentang nilai default untuk protokol dan sandi keamanan SQL Server, lihat Protokol dalam TLS/SSL (Schannel SSP) dan Cipher Suite dalam TLS/SSL (Schannel SSP) di dokumentasi Microsoft.

Gunakan proses berikut untuk mengonfigurasi protokol keamanan dan cipher:

1. Buat grup parameter DB kustom.
2. Ubah parameter di grup parameter.
3. Kaitkan grup parameter DB dengan instans DB Anda.

Untuk informasi selengkapnya tentang grup parameter DB, lihat Bekerja dengan grup parameter.

Membuat grup parameter yang terkait dengan keamanan

Buat grup parameter untuk parameter terkait keamanan Anda yang sesuai dengan edisi SQL Server dan versi instans DB Anda.

Konsol

Prosedur berikut membuat grup parameter untuk Edisi Standar SQL Server 2016.

Untuk membuat grup parameter

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup parameter.
3. Pilih Buat grup parameter.
4. Di panel Buat grup parameter, lakukan hal berikut:
   a. Untuk Rangkaian grup parameter, pilih sqlserver-se-13.0.
   b. Untuk Nama grup, masukkan pengidentifikasi grup parameter, seperti **`sqlserver-ciphers-se-13`**.
   c. Untuk Deskripsi, masukkan **`Parameter group for security protocols and ciphers`**.
5. Pilih Buat.

CLI

Prosedur berikut membuat grup parameter untuk Edisi Standar SQL Server 2016.

Untuk membuat grup parameter

- Jalankan salah satu perintah berikut ini.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds create-db-parameter-group \
    --db-parameter-group-name sqlserver-ciphers-se-13 \
    --db-parameter-group-family "sqlserver-se-13.0" \
    --description "Parameter group for security protocols and ciphers"
```

  Untuk Windows:

```
aws rds create-db-parameter-group ^
    --db-parameter-group-name sqlserver-ciphers-se-13 ^
    --db-parameter-group-family "sqlserver-se-13.0" ^
    --description "Parameter group for security protocols and ciphers"
```

Mengubah parameter yang terkait dengan keamanan

Ubah parameter yang terkait dengan keamanan di grup parameter yang sesuai dengan edisi SQL Server dan versi instans DB Anda.

Konsol

Prosedur berikut akan mengubah grup parameter yang telah Anda buat untuk Edisi Standar SQL Server 2016. Contoh ini menonaktifkan TLS versi 1.0.

Untuk mengubah grup parameter

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup parameter.
3. Pilih grup parameter, seperti sqlserver-ciphers-se-13.
4. Di bagian Parameter, filter daftar parameter untuk **rds**.
5. Pilih Edit parameter.
6. Pilih rds.tls10.

7. Untuk Nilai, pilih dinonaktifkan.
8. Pilih Smpan Perubahan.

CLI

Prosedur berikut akan mengubah grup parameter yang telah Anda buat untuk Edisi Standar SQL Server 2016. Contoh ini menonaktifkan TLS versi 1.0.

Untuk mengubah grup parameter

- Jalankan salah satu perintah berikut ini.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name sqlserver-ciphers-se-13 \
    --parameters
 "ParameterName='rds.tls10',ParameterValue='disabled',ApplyMethod=pending-reboot"
```

  Untuk Windows:

```
aws rds modify-db-parameter-group ^
    --db-parameter-group-name sqlserver-ciphers-se-13 ^
    --parameters
 "ParameterName='rds.tls10',ParameterValue='disabled',ApplyMethod=pending-reboot"
```

Mengaitkan grup parameter terkait keamanan dengan instans DB Anda

Untuk mengaitkan grup parameter dengan instans DB Anda, gunakan AWS Management Console atau AWS CLI.

Konsol

Anda dapat mengaitkan grup parameter dengan instans DB baru atau yang sudah ada:

- Untuk instans DB baru, kaitkan saat Anda meluncurkan instans. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.

- Untuk instans DB yang sudah ada, kaitkan dengan mengubah instans. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

CLI

Anda dapat mengaitkan grup parameter dengan instans DB baru atau yang sudah ada.

Untuk membuat instans DB dengan grup parameter

- Tentukan jenis mesin DB dan versi utama yang sama seperti yang Anda gunakan saat membuat grup parameter.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --db-instance-class db.m5.2xlarge \
    --engine sqlserver-se \
    --engine-version 13.00.5426.0.v1 \
    --allocated-storage 100 \
    --master-user-password secret123 \
    --master-username admin \
    --storage-type gp2 \
    --license-model li \
    --db-parameter-group-name sqlserver-ciphers-se-13
```

  Untuk Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --db-instance-class db.m5.2xlarge ^
    --engine sqlserver-se ^
    --engine-version 13.00.5426.0.v1 ^
    --allocated-storage 100 ^
    --master-user-password secret123 ^
    --master-username admin ^
    --storage-type gp2 ^
    --license-model li ^
    --db-parameter-group-name sqlserver-ciphers-se-13
```

> **Note**
>
> Tentukan kata sandi selain prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

Untuk mengubah instans DB serta mengaitkan grup parameter

- Jalankan salah satu perintah berikut ini.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --db-parameter-group-name sqlserver-ciphers-se-13 \
    --apply-immediately
```

  Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --db-parameter-group-name sqlserver-ciphers-se-13 ^
    --apply-immediately
```

Mengintegrasikan instans DB Amazon RDS for SQL Server dengan Amazon S3

Anda dapat mentransfer file antara instans DB yang menjalankan Amazon RDS for SQL Server dan bucket Amazon S3. Dengan melakukannya, Anda dapat menggunakan fitur Amazon S3 dengan SQL Server seperti BULK INSERT. Misalnya, Anda dapat mengunduh .csv, .xml, .txt, dan file lain dari Amazon S3 ke host instans DB dan mengimpor data dari `D:\S3\` ke dalam basis data. Semua file disimpan di `D:\S3\` pada instans DB.

Batasan berikut berlaku:

- File yang ada di folder `D:\S3` akan dihapus pada replika siaga setelah failover pada instans Multi-AZ. Untuk informasi selengkapnya, lihat Batasan Multi-AZ untuk integrasi S3.
- Instans DB dan bucket S3 harus berada di Wilayah AWS yang sama.
- Jika Anda menjalankan lebih dari satu tugas integrasi S3 sekaligus, tugas akan berjalan secara berurutan, bukan secara paralel.

> **Note**
>
> Tugas integrasi S3 akan berada di antrean yang sama dengan tugas pencadangan dan pemulihan native. Maksimal, Anda hanya dapat memiliki dua tugas yang berlangsung dalam antrean ini kapan saja. Oleh karena itu, dua tugas pencadangan dan pemulihan native yang berjalan akan memblokir tugas integrasi S3.

- Anda harus mengaktifkan ulang fitur integrasi S3 pada instans yang dipulihkan. Integrasi S3 tidak akan disebarkan dari instans sumber ke instans yang dipulihkan. File dalam `D:\S3` dihapus pada instans yang dipulihkan.
- Pengunduhan ke instans DB dibatasi hingga 100 file. Dengan kata lain, tidak boleh ada lebih dari 100 file di `D:\S3\`.
- Hanya file tanpa ekstensi file atau dengan ekstensi file berikut yang dapat diunduh: .abf, .asdatabase, .bcp, .configsettings, .csv, .dat, .deploymentoptions, .deploymenttargets, .fmt, . dan .xmla.
- Bucket S3 harus memiliki pemilik yang sama dengan peran AWS Identity and Access Management (IAM) terkait. Oleh karena itu, integrasi S3 lintas akun tidak didukung.
- bucket S3 tidak dapat dibuka untuk publik.
- Ukuran file untuk pengunggahan dari RDS ke S3 dibatasi 50 GB per file.

- Ukuran file untuk unduhan dari S3 ke RDS dibatasi sebesar ukuran maksimum yang didukung oleh S3.

Topik

- Prasyarat untuk mengintegrasikan RDS for SQL Server dengan S3
- Mengaktifkan integrasi RDS for SQL Server dengan S3
- Mentransfer file antara RDS for SQL Server dan Amazon S3
- Menampilkan daftar file di instans DB RDS
- Menghapus file di instans DB RDS
- Memantau status tugas transfer file
- Membatalkan tugas
- Batasan Multi-AZ untuk integrasi S3
- Menonaktifkan integrasi RDS for SQL Server dengan S3

Untuk informasi selengkapnya tentang menggunakan file di Amazon S3, lihat Mulai menggunakan Amazon Simple Storage Service.

Prasyarat untuk mengintegrasikan RDS for SQL Server dengan S3

Sebelum Anda memulai, temukan atau buat bucket S3 yang ingin Anda gunakan. Selain itu, tambahkan izin sehingga RDS instans DB dapat mengakses bucket S3. Untuk mengonfigurasi akses ini, Anda perlu membuat kebijakan IAM dan peran IAM.

Konsol

Untuk membuat kebijakan IAM bagi Aurora untuk mengakses Amazon S3

1. Di Konsol Manajemen IAM, pilih Kebijakan di panel navigasi.
2. Buat kebijakan baru, dan gunakan Editor visual untuk langkah-langkah berikut.
3. Untuk Layanan, masukkan **S3** lalu pilih layanan S3.
4. Untuk Tindakan, pilih yang berikut ini untuk memberikan akses yang diperlukan oleh instans DB Anda:
   - `ListAllMyBuckets` – diperlukan.
   - `ListBucket` – diperlukan.

- `GetBucketACL` – diperlukan.
- `GetBucketLocation` – diperlukan.
- `GetObject` – diperlukan untuk mengunduh file dari S3 ke `D:\S3\`
- `PutObject` – diperlukan untuk mengunggah file dari `D:\S3\` ke S3
- `ListMultipartUploadParts` – diperlukan untuk mengunggah file dari `D:\S3\` ke S3
- `AbortMultipartUpload` – diperlukan untuk mengunggah file dari `D:\S3\` ke S3

5. Untuk Sumber Daya, opsi yang ditampilkan bergantung pada tindakan mana yang Anda pilih di langkah sebelumnya. Anda mungkin melihat opsi untuk bucket, objek, atau keduanya. Untuk setiap hal berikut ini, tambahkan Amazon Resource Name (ARN) yang sesuai.

   Untuk bucket, tambahkan ARN untuk bucket yang ingin Anda gunakan. Misalnya, jika bucket Anda diberi nama `example-bucket`, atur ARN ke `arn:aws:s3:::example-bucket`.

   Untuk objek, masukkan ARN untuk bucket lalu pilih salah satu hal berikut:

   - Untuk memberikan akses ke semua file dalam bucket tertentu, pilih Semua untuk Nama bucket dan Nama objek.
   - Untuk memberikan akses ke file atau folder tertentu dalam bucket, sediakan ARN untuk bucket dan objek spesifik yang Anda ingin agar diakses oleh SQL Server.

6. Ikuti petunjuk dalam konsol hingga Anda selesai membuat kebijakan.

   Hal di atas adalah panduan singkat untuk membuat kebijakan. Untuk informasi selengkapnya tentang pembuatan peran IAM, lihat Membuat kebijakan IAM dalam Panduan Pengguna IAM.

Untuk membuat peran IAM yang menggunakan kebijakan IAM dari prosedur sebelumnya

1. Di Konsol Manajemen IAM, pilih Peran di panel navigasi.
2. Buat peran IAM baru, dan pilih opsi berikut saat muncul di konsol:

   - Layanan AWS
   - RDS
   - RDS – Tambahkan Peran ke Basis Data

   Lalu, pilih Next:Permissions di bagian bawah.

3. Untuk Lampirkan kebijakan izin, masukkan nama kebijakan IAM yang telah Anda buat sebelumnya. Lalu, pilih kebijakan dari daftar.
4. Ikuti petunjuk dalam konsol hingga Anda selesai membuat peran.

   Hal di atas adalah panduan singkat untuk membuat peran. Jika Anda menginginkan petunjuk yang lebih terperinci tentang pembuatan peran, lihat Peran IAM dalam Panduan Pengguna IAM.

AWS CLI

Untuk memberi Amazon RDS akses ke bucket Amazon S3, gunakan proses berikut:

1. Buat kebijakan IAM yang memberi Amazon RDS akses ke bucket S3.
2. Buat peran IAM yang dapat digunakan Amazon RDS atas nama Anda untuk mengakses bucket S3 Anda.

   Untuk informasi selengkapnya, lihat Membuat peran untuk mendelegasikan izin ke pengguna IAM dalam Panduan Pengguna IAM.
3. Lampirkan kebijakan IAM yang Anda buat ke peran IAM yang Anda buat.

Untuk membuat kebijakan IAM

Sertakan tindakan yang sesuai untuk memberikan akses yang diperlukan oleh instans DB Anda:

- `ListAllMyBuckets` – diperlukan.
- `ListBucket` – diperlukan.
- `GetBucketACL` – diperlukan.
- `GetBucketLocation` – diperlukan.
- `GetObject` – diperlukan untuk mengunduh file dari S3 ke `D:\S3\`
- `PutObject` – diperlukan untuk mengunggah file dari `D:\S3\` ke S3
- `ListMultipartUploadParts` – diperlukan untuk mengunggah file dari `D:\S3\` ke S3
- `AbortMultipartUpload` – diperlukan untuk mengunggah file dari `D:\S3\` ke S3

1. Perintah AWS CLI berikut membuat kebijakan IAM yang bernama `rds-s3-integration-policy` dengan opsi ini. Kebijakan ini memberikan akses ke bucket bernama `bucket_name`.

Example

Untuk Linux, macOS, atau Unix:

```
aws iam create-policy \
  --policy-name rds-s3-integration-policy \
  --policy-document '{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Action": "s3:ListAllMyBuckets",
                "Resource": "*"
            },
            {
                "Effect": "Allow",
                "Action": [
                    "s3:ListBucket",
                    "s3:GetBucketACL",
                    "s3:GetBucketLocation"
                ],
                "Resource": "arn:aws:s3:::bucket_name"
            },
            {
                "Effect": "Allow",
                "Action": [
                    "s3:GetObject",
                    "s3:PutObject",
                    "s3:ListMultipartUploadParts",
                    "s3:AbortMultipartUpload"
                ],
                "Resource": "arn:aws:s3:::bucket_name/key_prefix/*"
            }
        ]
    }'
```

Untuk Windows:

Pastikan untuk mengubah akhiran baris ke akhiran baris yang didukung oleh antarmuka Anda (^, bukan \). Selain itu, di Windows, Anda harus meng-escape semua tanda kutip ganda dengan

\. Agar tidak perlu meng-escape kutipan dalam JSON, Anda dapat menyimpannya ke file dan meneruskannya sebagai parameter.

Pertama, buat file `policy.json` dengan kebijakan izin berikut:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:ListAllMyBuckets",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:GetBucketACL",
                "s3:GetBucketLocation"
            ],
            "Resource": "arn:aws:s3:::bucket_name"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:ListMultipartUploadParts",
                "s3:AbortMultipartUpload"
            ],
            "Resource": "arn:aws:s3:::bucket_name/key_prefix/*"
        }
    ]
}
```

Lalu, gunakan perintah berikut ini untuk membuat kebijakan:

```
aws iam create-policy ^
     --policy-name rds-s3-integration-policy ^
     --policy-document file://file_path/assume_role_policy.json
```

2. Setelah kebijakan dibuat, catat Amazon Resource Name (ARN) kebijakan tersebut. Anda memerlukan ARN ini untuk langkah berikutnya.

Untuk membuat peran IAM

- Perintah AWS CLI berikut membuat peran IAM `rds-s3-integration-role` untuk tujuan ini.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws iam create-role \
    --role-name rds-s3-integration-role \
    --assume-role-policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
             "Service": "rds.amazonaws.com"
           },
          "Action": "sts:AssumeRole"
        }
      ]
    }'
```

  Untuk Windows:

  Pastikan untuk mengubah akhiran baris ke akhiran baris yang didukung oleh antarmuka Anda (^, bukan \). Selain itu, di Windows, Anda harus meng-escape semua tanda kutip ganda dengan \. Agar tidak perlu meng-escape kutipan dalam JSON, Anda dapat menyimpannya ke file dan meneruskannya sebagai parameter.

  Pertama, buat file `assume_role_policy.json` dengan kebijakan berikut:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
```

```
                "Service": [
                    "rds.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Lalu gunakan perintah berikut ini untuk membuat peran IAM:

```
aws iam create-role ^
     --role-name rds-s3-integration-role ^
     --assume-role-policy-document file://file_path/assume_role_policy.json
```

Example menggunakan kunci konteks kondisi global untuk membuat peran IAM

Sebaiknya gunakan kunci konteks kondisi global `aws:SourceArn` dan `aws:SourceAccount` dalam relasi kepercayaan berbasis sumber daya untuk membatasi izin layanan ke sumber daya tertentu. Ini adalah cara paling efektif untuk melindungi dari masalah "confused deputy".

Anda dapat menggunakan kedua kunci konteks kondisi global dan memiliki nilai `aws:SourceArn` yang berisi ID akun. Dalam hal ini, nilai `aws:SourceAccount` dan akun dalam nilai `aws:SourceArn` harus menggunakan ID akun yang sama ketika digunakan dalam pernyataan kebijakan yang sama.

- Gunakan `aws:SourceArn` jika Anda ingin akses lintas layanan untuk satu sumber daya.
- Gunakan `aws:SourceAccount` jika Anda ingin mengizinkan sumber daya apa pun di akun tersebut dikaitkan dengan penggunaan lintas layanan.

Dalam kebijakan, pastikan untuk menggunakan kunci konteks kondisi global `aws:SourceArn` dengan Amazon Resource Name (ARN) lengkap dari sumber daya yang mengakses peran. Untuk integrasi S3, pastikan untuk menyertakan ARN instans DB, seperti yang ditunjukkan dalam contoh berikut.

Untuk Linux, macOS, atau Unix:

```
aws iam create-role \
    --role-name rds-s3-integration-role \
```

```
    --assume-role-policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
             "Service": "rds.amazonaws.com"
           },
          "Action": "sts:AssumeRole",
                "Condition": {
                    "StringEquals": {

 "aws:SourceArn":"arn:aws:rds:Region:my_account_ID:db:db_instance_identifier"
                    }
                }
        }
      ]
    }'
```

Untuk Windows:

Tambahkan kunci konteks kondisi global ke `assume_role_policy.json`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "rds.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {

 "aws:SourceArn":"arn:aws:rds:Region:my_account_ID:db:db_instance_identifier"
                }
            }
        }
    ]
}
```

Untuk melampirkan kebijakan IAM untuk peran IAM

- Perintah AWS CLI berikut melampirkan kebijakan pada peran bernama `rds-s3-integration-role`. Ganti *`your-policy-arn`* dengan ARN kebijakan yang Anda catat di langkah sebelumnya.

  Example

  Untuk Linux, macOS, atau Unix:

  ```
  aws iam attach-role-policy \
      --policy-arn your-policy-arn \
      --role-name rds-s3-integration-role
  ```

  Untuk Windows:

  ```
  aws iam attach-role-policy ^
      --policy-arn your-policy-arn ^
      --role-name rds-s3-integration-role
  ```

Mengaktifkan integrasi RDS for SQL Server dengan S3

Pada bagian berikut, Anda dapat menemukan cara mengaktifkan integrasi Amazon S3 dengan Amazon RDS for SQL Server. Untuk menggunakan integrasi S3, instans DB Anda harus dikaitkan dengan peran IAM yang sebelumnya Anda buat sebelum menggunakan parameter nama fitur `S3_INTEGRATION`.

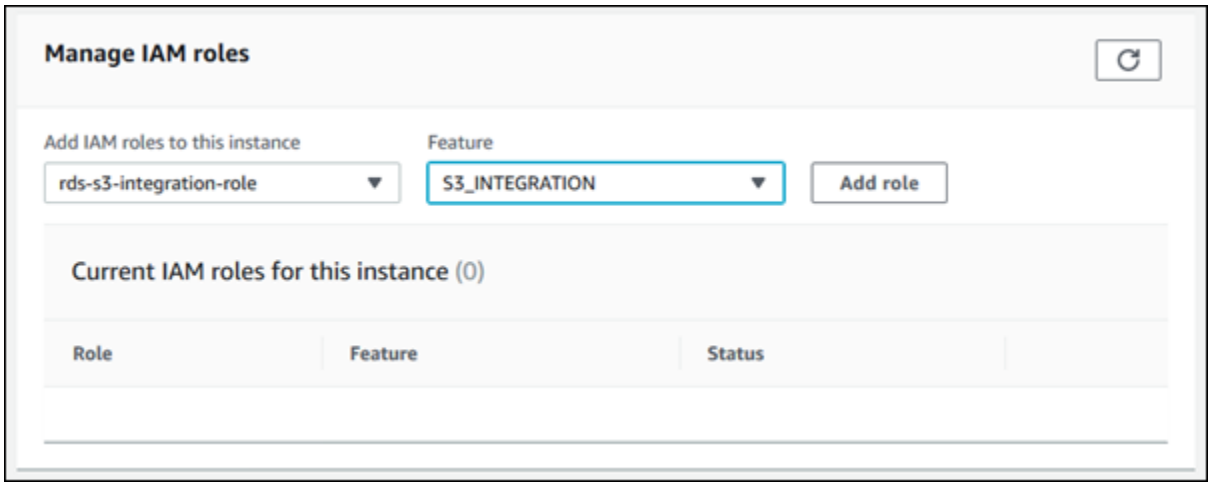
> **Note**
>
> Untuk menambahkan peran IAM ke instans DB, status instans DB harus tersedia.

Konsol

Untuk mengaitkan peran IAM Anda dengan instans DB Anda

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Pilih nama instans DB RDS for SQL Server untuk menampilkan detailnya.

3. Pada tab Konektivitas & keamanan, di Kelola peran IAM, pilih peran IAM yang akan ditambahkan pada bagian Tambahkan peran IAM ke instans ini.
4. Untuk Fitur, pilih S3_INTEGRATION.



5. Pilih Tambahkan peran.

AWS CLI

Untuk menambahkan peran IAM ke instans DB RDS for SQL Server

- Perintah AWS CLI berikut menambahkan peran IAM Anda ke instans DB RDS for SQL Server bernama *mydbinstance*.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds add-role-to-db-instance \
    --db-instance-identifier mydbinstance \
    --feature-name S3_INTEGRATION \
    --role-arn your-role-arn
```

  Untuk Windows:

```
aws rds add-role-to-db-instance ^
    --db-instance-identifier mydbinstance ^
    --feature-name S3_INTEGRATION ^
    --role-arn your-role-arn
```

Ganti *your-role-arn* dengan peran ARN yang Anda catat di langkah sebelumnya. `S3_INTEGRATION` harus ditentukan untuk opsi `--feature-name`.

Mentransfer file antara RDS for SQL Server dan Amazon S3

Anda dapat menggunakan prosedur tersimpan Amazon RDS untuk mengunduh dan mengunggah file antara Amazon S3 dan instans DB RDS Anda. Anda juga dapat menggunakan prosedur tersimpan Amazon RDS untuk menampilkan daftar dan menghapus file di instans RDS.

File yang Anda unduh dari dan unggah ke S3 disimpan di folder `D:\S3`. Ini adalah satu-satunya folder yang dapat Anda gunakan untuk mengakses file Anda. Anda dapat menyusun file menjadi subfolder, yang dibuat untuk Anda saat Anda menyertakan folder tujuan selama pengunduhan.

Beberapa prosedur tersimpan mengharuskan Anda memberikan Amazon Resource Name (ARN) ke bucket dan file S3 Anda. Format untuk ARN Anda adalah `arn:aws:s3:::bucket_name/file_name`. Amazon S3 tidak memerlukan nomor akun atau Wilayah AWS di ARN.

Tugas integrasi S3 berjalan secara berurutan dan berada di antrean yang sama dengan tugas pencadangan dan pemulihan native. Maksimal, Anda hanya dapat memiliki dua tugas yang berlangsung dalam antrean ini kapan saja. Mungkin perlu waktu hingga lima menit hingga tugas mulai diproses.

Mengunduh file dari bucket Amazon S3 ke instans DB SQL Server

Untuk mengunduh file dari bucket S3 ke instans DB RDS for SQL Server, gunakan prosedur tersimpan Amazon RDS `msdb.dbo.rds_download_from_s3` dengan parameter berikut ini.

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
`@s3_arn_of_file`	NVARCHAR	–	Wajib	ARN S3 dari file yang akan diunduh, misalnya: `arn:aws:s3:::bucket_name/mydata.csv`
`@rds_file_path`	NVARCHAR	–	Opsional	Jalur file untuk instans RDS. Jika tidak ditentukan, jalur file-nya adalah

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
				`D:\S3\<filename in s3>`. RDS mendukung jalur absolut dan jalur relatif. Jika Anda ingin membuat subfolder, sertakan dalam jalur file.
`@overwrite_file`	INT	0	Opsional	Timpa file yang ada: 0 = Jangan timpa 1 = Timpa

Anda dapat mengunduh file tanpa ekstensi file dan file dengan ekstensi file berikut: .bcp, .csv, .dat, .fmt, .info, .lst, .tbl, .txt, dan .xml.

> **Note**
>
> File dengan ekstensi file .ispac dapat diunduh ketika SQL Server Integration Services diaktifkan. Untuk informasi selengkapnya tentang pengaktifan SSIS, lihat SQL Server Integration Services.
> File dengan ekstensi file berikut dapat diunduh ketika SQL Server Analysis Services diaktifkan: .abf, .asdatabase, .configsettings, .deploymentoptions, .deploymenttargets, dan .xmla. Untuk informasi selengkapnya tentang pengaktifan SSAS, lihat SQL Server Analysis Services.

Contoh berikut menunjukkan prosedur tersimpan untuk mengunduh file dari S3.

```
exec msdb.dbo.rds_download_from_s3
     @s3_arn_of_file='arn:aws:s3:::bucket_name/bulk_data.csv',
     @rds_file_path='D:\S3\seed_data\data.csv',
     @overwrite_file=1;
```

Contoh operasi `rds_download_from_s3` membuat folder yang diberi nama `seed_data` di `D:\S3\` jika folder belum ada. Kemudian, contoh tersebut mengunduh file sumber `bulk_data.csv` dari S3

ke file baru bernama `data.csv` di instans DB. Jika file sebelumnya ada, file ini akan ditimpa karena parameter `@overwrite_file` diatur ke 1.

Mengunggah file dari instans DB SQL Server ke bucket Amazon S3

Untuk mengunggah file dari instans DB RDS for SQL Server ke bucket S3, gunakan prosedur tersimpan Amazon RDS `msdb.dbo.rds_upload_to_s3` dengan parameter berikut ini.

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
`@s3_arn_of_file`	NVARCHAR	–	Wajib	ARN S3 dari file yang akan dibuat di S3, misalnya: `arn:aws:s3:::bucket_name/mydata.csv`
`@rds_file_path`	NVARCHAR	–	Wajib	Jalur file yang akan diunggah ke S3. Jalur absolut dan relatif didukung.
`@overwrite_file`	INT	–	Opsional	Timpa file yang ada: 0 = Jangan timpa 1 = Timpa

Contoh berikut mengunggah file bernama `data.csv` dari lokasi yang ditentukan di `D:\S3\seed_data\` ke file `new_data.csv` dalam bucket S3 yang ditentukan berdasarkan ARN.

```
exec msdb.dbo.rds_upload_to_s3
  @rds_file_path='D:\S3\seed_data\data.csv',
  @s3_arn_of_file='arn:aws:s3:::bucket_name/new_data.csv',
  @overwrite_file=1;
```

Jika file sebelumnya ada di S3, file ini akan ditimpa karena parameter @overwrite_file diatur ke 1.

Menampilkan daftar file di instans DB RDS

Untuk menampilkan daftar file yang tersedia di instans DB, gunakan prosedur tersimpan dan fungsi. Pertama, jalankan prosedur tersimpan berikut untuk mengumpulkan detail file dari file dalam `D:\S3\`.

```
exec msdb.dbo.rds_gather_file_details;
```

Prosedur tersimpan menampilkan ID tugas. Seperti tugas lain, prosedur tersimpan ini berjalan secara asinkron. Segera setelah status tugas menjadi `SUCCESS`, Anda dapat menggunakan ID tugas dalam fungsi `rds_fn_list_file_details` untuk menampilkan daftar file dan direktori yang ada di D:\S3\, seperti yang ditunjukkan berikut ini.

```
SELECT * FROM msdb.dbo.rds_fn_list_file_details(TASK_ID);
```

Fungsi `rds_fn_list_file_details` menampilkan tabel dengan kolom berikut.

Parameter output	Deskripsi
`filepath`	Jalur file absolut (misalnya, `D:\S3\mydata.csv`)
`size_in_bytes`	Ukuran file (dalam byte)
`last_modified_utc`	Tanggal dan waktu modifikasi terakhir dalam format UTC
`is_directory`	Opsi yang menunjukkan apakah item merupakan direktori (`true`/`false`)

Menghapus file di instans DB RDS

Untuk menghapus file yang tersedia pada instans DB, gunakan prosedur tersimpan Amazon RDS `msdb.dbo.rds_delete_from_filesystem` dengan parameter berikut ini.

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
`@rds_file_path`	NVARCHAR	–	Wajib	Jalur file dari file yang akan dihapus. Jalur

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
				absolut dan relatif didukung.
`@force_delete`	INT	0	Opsional	Untuk menghapus direktori, tanda ini harus disertakan dan diatur ke 1. 1 = menghapus direktori Parameter ini diabaikan jika Anda menghapus file.

Untuk menghapus direktori, `@rds_file_path` harus diakhiri dengan garis miring terbalik (\) dan `@force_delete` harus diatur ke 1.

Contoh berikut menghapus file `D:\S3\delete_me.txt`.

```
exec msdb.dbo.rds_delete_from_filesystem
    @rds_file_path='D:\S3\delete_me.txt';
```

Contoh berikut menghapus direktori `D:\S3\example_folder\`.

```
exec msdb.dbo.rds_delete_from_filesystem
    @rds_file_path='D:\S3\example_folder\',
    @force_delete=1;
```

Memantau status tugas transfer file

Untuk melacak status tugas integrasi S3, panggil fungsi `rds_fn_task_status`. Fungsi ini membutuhkan dua parameter. Parameter pertama harus selalu `NULL` karena tidak berlaku pada integrasi S3. Parameter kedua dapat berisi ID tugas.

Untuk melihat daftar semua tugas, tetapkan parameter pertama untuk `NULL` dan parameter kedua untuk `0`, seperti yang ditunjukkan dalam contoh berikut.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,0);
```

Untuk melihat tugas tertentu, atur parameter pertama ke NULL dan parameter kedua ke ID tugas, seperti yang ditunjukkan dalam contoh berikut.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,42);
```

Fungsi rds_fn_task_status akan menampilkan informasi berikut.

Parameter output	Deskripsi
task_id	ID tugas.
task_type	Untuk integrasi S3, tugas dapat memiliki jenis tugas berikut: • DOWNLOAD_FROM_S3 • UPLOAD_TO_S3 • LIST_FILES_ON_DISK • DELETE_FILES_ON_DISK
database_name	Tidak berlaku untuk tugas integrasi S3.
% complete	Progres tugas sebagai persentase.
duration(mins)	Durasi yang dihabiskan untuk tugas, dalam menit.
lifecycle	Status tugas. Status yang mungkin adalah: • CREATED – Setelah Anda memanggil salah satu prosedur tersimpan integrasi S3, sebuah tugas akan dibuat dan statusnya diatur ke CREATED. • IN_PROGRESS – Setelah tugas dimulai, statusnya diatur ke IN_PROGRESS . Proses ini dapat memakan waktu sampai lima menit hingga status berubah dari CREATED menjadi IN_PROGRESS .

Parameter output	Deskripsi
	• SUCCESS – Setelah tugas selesai, statusnya akan diatur ke SUCCESS. • ERROR – Jika tugas gagal, statusnya akan diatur ke ERROR. Untuk informasi selengkapnya mengenai kesalahan, lihat kolom task_info . • CANCEL_REQUESTED – Setelah Anda memanggil rds_cancel_task , status tugas akan diatur ke CANCEL_REQUESTED . • CANCELLED – Setelah tugas berhasil dibatalkan, statusnya akan diatur ke CANCELLED .
task_info	Informasi tambahan mengenai tugas. Jika terjadi kesalahan selama pemrosesan, kolom ini berisi informasi tentang kesalahan tersebut.
last_updated	Tanggal dan waktu status tugas terakhir diperbarui.
created_at	Tanggal dan waktu tugas dibuat.
S3_object_arn	ARN dari objek S3 tempat file diunduh atau diunggah.
overwrite_S3_backup_file	Tidak berlaku untuk tugas integrasi S3.
KMS_master_key_arn	Tidak berlaku untuk tugas integrasi S3.
filepath	Jalur file pada instans DB RDS.
overwrite_file	Opsi yang menunjukkan apakah file yang sudah ada akan ditimpa.
task_metadata	Tidak berlaku untuk tugas integrasi S3.

Membatalkan tugas

Untuk membatalkan tugas integrasi S3, gunakan prosedur tersimpan `msdb.dbo.rds_cancel_task` dengan parameter `task_id`. Tugas hapus dan tugas tampilkan daftar yang sedang berlangsung tidak dapat dibatalkan. Contoh berikut menunjukkan permintaan untuk membatalkan tugas.

```
exec msdb.dbo.rds_cancel_task @task_id = 1234;
```

Untuk mendapatkan gambaran umum tentang semua tugas dan ID tugasnya, gunakan fungsi `rds_fn_task_status` sebagaimana dijelaskan dalam Memantau status tugas transfer file.

Batasan Multi-AZ untuk integrasi S3

Pada instans Multi-AZ, file dalam folder `D:\S3` dihapus pada replika siaga setelah failover. Failover dapat direncanakan, misalnya, selama modifikasi instans DB seperti mengubah kelas instans atau meng-upgrade versi mesin. Atau, failover bisa jadi tidak terencana, selama pemadaman replika primer.

> **Note**
>
> Kami tidak menyarankan penggunaan `D:\S3` untuk penyimpanan file. Praktik terbaiknya adalah mengunggah file yang dibuat ke Amazon S3 agar durabel, dan mengunduh file tersebut saat Anda perlu mengimpor data.

Untuk menentukan waktu failover terakhir, Anda dapat menggunakan prosedur tersimpan `msdb.dbo.rds_failover_time`. Untuk informasi selengkapnya, lihat Menentukan waktu failover terakhir.

Example tidak ada failover terbaru

Contoh ini menampilkan output saat tidak ada failover terbaru di log kesalahan. Tidak terjadi failover sejak 2020-04-29 23:59:00.01.

Oleh karena itu, semua file yang diunduh setelah waktu tersebut yang belum dihapus menggunakan prosedur tersimpan `rds_delete_from_filesystem` masih dapat diakses di host saat ini. File yang diunduh sebelum waktu tersebut mungkin juga tersedia.

errorlog_available_from	recent_failover_time
29-04-2020 23:59:00.0100000	kosong

Example failover terbaru

Contoh ini menampilkan output saat tidak ada failover di log kesalahan. Failover terbaru adalah pada 2020-05-05 18:57:51.89.

Semua file yang diunduh setelah waktu tersebut yang belum dihapus menggunakan prosedur tersimpan `rds_delete_from_filesystem` masih dapat diakses di host saat ini.

errorlog_available_from	recent_failover_time
29-04-2020 23:59:00.0100000	2020-05-05 18:57:51.8900000

Menonaktifkan integrasi RDS for SQL Server dengan S3

Di bagian berikut ini, Anda dapat menemukan cara mengaktifkan integrasi Amazon S3 dengan Amazon RDS for SQL Server. File dalam `D:\S3\` tidak dihapus saat menonaktifkan integrasi S3.

> **Note**
>
> Untuk menghapus peran IAM dari instans DB, status instans DB harus `available`.

Konsol

Untuk membatalkan pengaitan peran IAM Anda dari instans DB Anda

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Pilih nama instans DB RDS for SQL Server untuk menampilkan detailnya.
3. Pada tab Konektivitas & keamanan, di bagian Kelola peran IAM, pilih peran IAM yang akan dihapus.

4. Pilih Hapus.

AWS CLI

Untuk menghapus peran IAM dari instans DB RDS for SQL Server

- Perintah AWS CLI berikut menghapus peran IAM dari instans DB RDS for SQL Server yang bernama *mydbinstance*.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds remove-role-from-db-instance \
    --db-instance-identifier mydbinstance \
    --feature-name S3_INTEGRATION \
    --role-arn your-role-arn
```

  Untuk Windows:

```
aws rds remove-role-from-db-instance ^
    --db-instance-identifier mydbinstance ^
    --feature-name S3_INTEGRATION ^
    --role-arn your-role-arn
```

  Ganti *your-role-arn* dengan ARN peran IAM yang sesuai untuk opsi `--feature-name`.

Menggunakan Database Mail di Amazon RDS for SQL Server

Anda dapat menggunakan Database Mail untuk mengirim pesan email ke pengguna dari instans basis data Amazon RDS di SQL Server. Pesan dapat berisi file dan hasil kueri. Database Mail mencakup komponen berikut:

- Objek konfigurasi dan keamanan – Objek-objek ini membuat profil dan akun, dan disimpan di basis data `msdb`.
- Objek pesan – Objek-objek ini mencakup sp_send_dbmail prosedur tersimpan yang digunakan untuk mengirim pesan, dan struktur data yang menyimpan informasi tentang pesan. Semuanya disimpan di basis data `msdb`.
- Pembuatan log dan audit objek – Database Mail menulis informasi log ke database `msdb` dan log event aplikasi Microsoft Windows.
- Database Mail dapat executable – `DatabaseMail.exe` membaca dari antrean di basis data `msdb` dan mengirim pesan email.

RDS mendukung Database Mail untuk semua versi SQL Server di Edisi Web, Standar, dan Perusahaan.

Batasan

Batasan berikut berlaku untuk menggunakan Database Mail di instans DB SQL Server Anda:

- Pesan basis data tidak mendukung Edisi Ekspres SQL Server.
- Mengubah parameter konfigurasi Database Mail tidak didukung. Untuk melihat nilai preset (default), gunakan prosedur tersimpan sysmail_help_configure_sp.
- Lampiran file tidak sepenuhnya didukung. Untuk informasi selengkapnya, lihat Bekerja dengan lampiran file.
- Ukuran lampiran file maksimum adalah 1 MB.
- Database Mail memerlukan konfigurasi tambahan pada intans DB Multi-AZ. Untuk informasi selengkapnya, lihat Pertimbangan untuk deployment multi-AZ.
- Mengonfigurasikan Agen SQL Server untuk mengirim pesan email ke operator yang sudah ditentukan tidak didukung.

Mengaktifkan Database Mail

Gunakan proses berikut untuk mengaktifkan Database Mail untuk instans DB Anda:

1. Buat grup parameter baru.
2. Ubah grup parameter untuk mengatur parameter `database mail xps` ke 1.
3. Hubungkan grup parameter baru dengan instans DB.

Membuat grup parameter untuk Database Mail

Buat grup parameter untuk parameter `database mail xps` yang sesuai dengan edisi SQL Server dan versi instans DB Anda.

> **Note**
>
> Anda juga dapat mengubah grup parameter yang ada. Ikuti prosedur di Mengubah grup parameter untuk Database Mail.

Konsol

Contoh berikut akan membuat grup parameter untuk SQL Server Standard Edition 2016.

Untuk membuat grup parameter

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup parameter.
3. Pilih Buat grup parameter.
4. Di panel Buat grup parameter, lakukan hal berikut:
   a. Untuk Rangkaian grup parameter, pilih sqlserver-se-13.0.
   b. Untuk Nama grup, masukkan pengidentifikasi grup parameter, seperti **`dbmail-sqlserver-se-13`**.
   c. Untuk Deskripsi, masukkan **`Database Mail XPs`**.
5. Pilih Buat.

CLI

Contoh berikut akan membuat grup parameter untuk SQL Server Standard Edition 2016.

Untuk membuat grup parameter

- Gunakan salah satu perintah berikut ini.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds create-db-parameter-group \
    --db-parameter-group-name dbmail-sqlserver-se-13 \
    --db-parameter-group-family "sqlserver-se-13.0" \
    --description "Database Mail XPs"
```

  Untuk Windows:

```
aws rds create-db-parameter-group ^
    --db-parameter-group-name dbmail-sqlserver-se-13 ^
    --db-parameter-group-family "sqlserver-se-13.0" ^
    --description "Database Mail XPs"
```

Mengubah grup parameter untuk Database Mail

Ubah parameter `database mail xps` di grup parameter yang sesuai dengan edisi SQL Server dan versi instans DB Anda.

Untuk mengaktifkan Database Mail, atur parameter `database mail xps` ke 1.

Konsol

Contoh berikut akan mengubah grup parameter yang telah Anda buat untuk SQL Server Standard Edition 2016.

Untuk mengubah grup parameter

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup parameter.

3. Pilih grup parameter, seperti dbmail-sqlserver-se-13.
4. Di bagian Parameter, filter daftar parameter untuk **`mail`**.
5. Pilih Database Mail xps.
6. Pilih Edit parameter.
7. Masukkan **`1`**.
8. Pilih Smpan Perubahan.

CLI

Contoh berikut akan mengubah grup parameter yang telah Anda buat untuk SQL Server Standard Edition 2016.

Untuk mengubah grup parameter

- Gunakan salah satu perintah berikut ini.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name dbmail-sqlserver-se-13 \
    --parameters "ParameterName='database mail
 xps',ParameterValue=1,ApplyMethod=immediate"
```

  Untuk Windows:

```
aws rds modify-db-parameter-group ^
    --db-parameter-group-name dbmail-sqlserver-se-13 ^
    --parameters "ParameterName='database mail
 xps',ParameterValue=1,ApplyMethod=immediate"
```

Kaitkan grup parameter dengan instans DB

Anda dapat menggunakan AWS Management Console atau AWS CLI untuk mengaitkan grup parameter Database Mail dengan instans DB.

Konsol

Anda dapat mengaitkan grup parameter Database Mail dengan instans DB baru atau yang sudah ada.

- Untuk instans DB baru, kaitkan saat Anda meluncurkan instans. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.
- Untuk instans DB yang sudah ada, kaitkan dengan mengubah instans. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

CLI

Anda dapat mengaitkan grup parameter Database Mail dengan instans DB baru atau yang sudah ada.

Untuk membuat instans DB dengan grup parameter Database Mail

- Tentukan jenis mesin DB dan versi utama yang sama seperti yang Anda gunakan saat membuat grup parameter.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --db-instance-class db.m5.2xlarge \
    --engine sqlserver-se \
    --engine-version 13.00.5426.0.v1 \
    --allocated-storage 100 \
    --manage-master-user-password \
    --master-username admin \
    --storage-type gp2 \
    --license-model li
    --db-parameter-group-name dbmail-sqlserver-se-13
```

  Untuk Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --db-instance-class db.m5.2xlarge ^
```

```
    --engine sqlserver-se ^
    --engine-version 13.00.5426.0.v1 ^
    --allocated-storage 100 ^
    --manage-master-user-password ^
    --master-username admin ^
    --storage-type gp2 ^
    --license-model li ^
    --db-parameter-group-name dbmail-sqlserver-se-13
```

Untuk mengubah instans DB dan mengaitkan grup parameter Database Mail

- Gunakan salah satu perintah berikut ini.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --db-parameter-group-name dbmail-sqlserver-se-13 \
    --apply-immediately
```

  Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --db-parameter-group-name dbmail-sqlserver-se-13 ^
    --apply-immediately
```

Mengonfigurasi Database Mail

Anda melakukan tugas berikut untuk mengonfigurasikan Database Mail:

1. Buat profil Database Mail.
2. Buat akun Database Mail.
3. Tambahkan akun Database Mail ke profil Database Mail.
4. Tambahkan pengguna ke profil Database Mail.

> **Note**
>
> Untuk mengonfigurasi Database Mail, pastikan bahwa Anda memiliki izin `execute` pada prosedur yang disimpan dalam database `msdb`.

Membuat profil Database Mail

Untuk membuat profil Database Mail, Anda menggunakan prosedur tersimpan sysmail_add_profile_sp. Contoh berikut membuat profil dengan nama `Notifications`.

Untuk membuat profil

- Gunakan pernyataan SQL berikut.

```
USE msdb
GO

EXECUTE msdb.dbo.sysmail_add_profile_sp
    @profile_name         = 'Notifications',
    @description          = 'Profile used for sending outgoing notifications using
 Amazon SES.';
GO
```

Membuat akun Database Mail

Untuk membuat akun Database Mail, Anda menggunakan prosedur tersimpan sysmail_add_account_sp. Contoh berikut membuat akun dengan nama `SES` di RDS untuk instans DB SQL Server di VPC pribadi, menggunakan Layanan Email Sederhana Amazon.

Menggunakan Amazon SES memerlukan parameter berikut:

- `@email_address`— Identitas terverifikasi Amazon SES. Untuk informasi selengkapnya, lihat Identitas terverifikasi di Amazon SES.
- `@mailserver_name`— Titik akhir SMTP Amazon SES. Untuk informasi selengkapnya, lihat Menghubungkan ke titik akhir Amazon SES SMTP.
- `@username`— Nama pengguna SMTP Amazon SES. Untuk informasi selengkapnya, lihat Mendapatkan kredensial Amazon SES SMTP.

Jangan gunakan nama pengguna AWS Identity and Access Management.

- `@password`— Kata sandi Amazon SES SMTP. Untuk informasi selengkapnya, lihat Mendapatkan kredensial Amazon SES SMTP.

Untuk membuat akun

- Gunakan pernyataan SQL berikut.

```
USE msdb
GO

EXECUTE msdb.dbo.sysmail_add_account_sp
    @account_name        = 'SES',
    @description         = 'Mail account for sending outgoing notifications.',
    @email_address       = 'nobody@example.com',
    @display_name        = 'Automated Mailer',
    @mailserver_name     = 'vpce-0a1b2c3d4e5f-01234567.email-smtp.us-
west-2.vpce.amazonaws.com',
    @port                = 587,
    @enable_ssl          = 1,
    @username            = 'Smtp_Username',
    @password            = 'Smtp_Password';
GO
```

> **Note**
>
> Tentukan kredensial selain prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

Menambahkan akun Database Mail ke profil Database Mail

Untuk menambahkan akun Database Mail ke profil Database Mail, gunakan prosedur tersimpan sysmail_add_profileaccount_sp. Contoh berikut menambahkan akun `SES` ke profil `Notifications`.

Menambahkan akun ke profil

- Gunakan pernyataan SQL berikut.

```
USE msdb
GO

EXECUTE msdb.dbo.sysmail_add_profileaccount_sp
    @profile_name        = 'Notifications',
    @account_name        = 'SES',
    @sequence_number     = 1;
GO
```

Menambahkan pengguna ke profil Database Mail

Untuk memberikan izin bagipengguna utama basis data `msdb` untuk menggunakan profil Database Mail, Anda menggunakan prosedur sysmail_add_principalprofile_sp tersimpan. Pengguna utama adalah entitas yang dapat meminta sumber daya SQL Server. Pengguna utama basis data harus dipetakan ke pengguna autentikasi SQL Server, pengguna Autentikasi Windows, atau grup Autentikasi Windows.

Contoh berikut memberikan akses publik ke profil `Notifications`.

Untuk menambahkan pengguna ke profil

- Gunakan pernyataan SQL berikut.

```
USE msdb
GO

EXECUTE msdb.dbo.sysmail_add_principalprofile_sp
    @profile_name       = 'Notifications',
    @principal_name     = 'public',
    @is_default         = 1;
GO
```

Prosedur dan fungsi yang disimpan Amazon RDS untuk Database Mail

Microsoft menyediakan prosedur tersimpan untuk menggunakan Database Mail, seperti membuat, mencantumkan, memperbarui, dan menghapus akun dan profil. Selain itu, RDS menyediakan prosedur dan fungsi yang disimpan untuk Database Mail yang ditampilkan dalam tabel berikut.

Prosedur/Fungsi	Deskripsi
rds_fn_sysmail_allitems	Menampilkan pesan terkirim, termasuk yang dikirim oleh pengguna lain.
rds_fn_sysmail_event_log	Menampilkan acara, termasuk acara untuk pesan yang dikirimkan oleh pengguna lain.
rds_fn_sysmail_mailattachments	Menampilkan lampiran terkirim, termasuk yang dikirim oleh pengguna lain.
rds_sysmail_control	Memulai dan menghentikan antrian email (DatabaseMailprocess.exe).
rds_sysmail_delete_mailitems_sp	Menghapus pesan email yang dikirim oleh semua pengguna dari tabel internal Database Mail.

Mengirim pesan email menggunakan Database Mail

Anda menggunakan prosedur sp_send_dbmail tersimpan untuk mengirim pesan email menggunakan Database Mail.

Penggunaan

```
EXEC msdb.dbo.sp_send_dbmail
@profile_name = 'profile_name',
@recipients = 'recipient1@example.com[; recipient2; ... recipientn]',
@subject = 'subject',
@body = 'message_body',
[@body_format = 'HTML'],
[@file_attachments = 'file_path1; file_path2; ... file_pathn'],
[@query = 'SQL_query'],
[@attach_query_result_as_file = 0|1]';
```

Parameter berikut diperlukan:

- `@profile_name` – Nama profil Database Mail yang akan digunakan untuk mengirim pesan.
- `@recipients` – Daftar alamat email tujuan pengiriman pesan yang dipisahkan titik koma.
- `@subject` – Subjek pesan.

- `@body` – Isi pesan. Anda juga dapat menggunakan variabel yang dinyatakan sebagai isi.

Parameter berikut ini bersifat opsional:

- `@body_format` – Parameter ini digunakan dengan variabel yang dinyatakan untuk mengirim email dalam format HTML.
- `@file_attachments` – Daftar lampiran pesan yang disusun dengan titik koma. Path file harus berupa path absolut.
- `@query` – Kueri SQL untuk dijalankan. Hasil kueri dapat dilampirkan sebagai file atau disertakan dalam isi pesan.
- `@attach_query_result_as_file` – Apakah melampirkan hasil kueri sebagai file. Atur ke 0 untuk tidak, 1 untuk ya. Defaultnya adalah 60.

Contoh-contoh

Contoh-contoh berikut ini mendemonstrasikan cara mengirim pesan email.

Example mengirimkan pesan ke satu penerima

```
USE msdb
GO

EXEC msdb.dbo.sp_send_dbmail
     @profile_name       = 'Notifications',
     @recipients         = 'nobody@example.com',
     @subject            = 'Automated DBMail message - 1',
     @body               = 'Database Mail configuration was successful.';
GO
```

Example mengirimkan pesan ke beberapa penerima

```
USE msdb
GO

EXEC msdb.dbo.sp_send_dbmail
     @profile_name       = 'Notifications',
     @recipients         = 'recipient1@example.com;recipient2@example.com',
     @subject            = 'Automated DBMail message - 2',
     @body               = 'This is a message.';
```

```
GO
```

Example mengirimkan hasil kueri SQL sebagai lampiran file

```
USE msdb
GO

EXEC msdb.dbo.sp_send_dbmail
     @profile_name       = 'Notifications',
     @recipients         = 'nobody@example.com',
     @subject            = 'Test SQL query',
     @body               = 'This is a SQL query test.',
     @query              = 'SELECT * FROM abc.dbo.test',
     @attach_query_result_as_file = 1;
GO
```

Example mengirimkan pesan dalam format HTML

```
USE msdb
GO

DECLARE @HTML_Body as NVARCHAR(500) = 'Hi, <h4> Heading </h4> </br> See the report. <b>
 Regards </b>';

EXEC msdb.dbo.sp_send_dbmail
     @profile_name       = 'Notifications',
     @recipients         = 'nobody@example.com',
     @subject            = 'Test HTML message',
     @body               = @HTML_Body,
     @body_format        = 'HTML';
GO
```

Example mengirimkan pesan menggunakan pemicu saat event tertentu terjadi di basis data

```
USE AdventureWorks2017
GO
IF OBJECT_ID ('Production.iProductNotification', 'TR') IS NOT NULL
DROP TRIGGER Purchasing.iProductNotification
GO

CREATE TRIGGER iProductNotification ON Production.Product
   FOR INSERT
```

```
   AS
   DECLARE @ProductInformation nvarchar(255);
   SELECT
   @ProductInformation = 'A new product, ' + Name + ', is now available for $' +
 CAST(StandardCost AS nvarchar(20)) + '!'
   FROM INSERTED i;

EXEC msdb.dbo.sp_send_dbmail
     @profile_name       = 'Notifications',
     @recipients         = 'nobody@example.com',
     @subject            = 'New product information',
     @body               = @ProductInformation;
GO
```

Melihat pesan, log, dan lampiran

Anda menggunakan prosedur yang disimpan RDS untuk melihat pesan, log event, dan lampiran.

Untuk melihat semua pesan email

- Gunakan kueri SQL berikut.

```
SELECT * FROM msdb.dbo.rds_fn_sysmail_allitems(); --WHERE sent_status='sent' or
 'failed' or 'unsent'
```

Untuk melihat semua log event email

- Gunakan kueri SQL berikut.

```
SELECT * FROM msdb.dbo.rds_fn_sysmail_event_log();
```

Untuk melihat semua lampiran email

- Gunakan kueri SQL berikut.

```
SELECT * FROM msdb.dbo.rds_fn_sysmail_mailattachments();
```

Menghapus pesan

Anda menggunakan prosedur `rds_sysmail_delete_mailitems_sp` yang disimpan untuk menghapus pesan.

> **Note**
>
> RDS secara otomatis menghapus item tabel mail saat data riwayat DBMail mencapai ukuran 1 GB, dengan jangka waktu penyimpanan setidaknya 24 jam.
> Jika Anda ingin menyimpan mail untuk periode yang lebih lama, Anda dapat mengarsipkannya. Untuk informasi selengkapnya, lihat Buat pekerjaan SQL Server Agent untuk mengarsipkan pesan Database Mail dan log event di dokumentasi Microsoft.

Untuk menghapus semua pesan email

- Gunakan pernyataan SQL berikut.

```
DECLARE @GETDATE datetime
SET @GETDATE = GETDATE();
EXECUTE msdb.dbo.rds_sysmail_delete_mailitems_sp @sent_before = @GETDATE;
GO
```

Untuk menghapus semua pesan email dengan status tertentu

- Gunakan laporan SQL berikut untuk menghapus semua pesan yang gagal.

```
DECLARE @GETDATE datetime
SET @GETDATE = GETDATE();
EXECUTE msdb.dbo.rds_sysmail_delete_mailitems_sp @sent_status = 'failed';
GO
```

Memulai antrean email

Anda menggunakan prosedur `rds_sysmail_control` yang disimpan untuk memulai proses Database Mail.

> **Note**
>
> Mengaktifkan Database Mail secara otomatis memulai antrean email.

Untuk memulai antrean email

- Gunakan pernyataan SQL berikut.

```
EXECUTE msdb.dbo.rds_sysmail_control start;
GO
```

Menghentikan antrean email

Anda menggunakan prosedur `rds_sysmail_control` yang disimpan untuk memulai menghentikan proses Database Mail.

Untuk menghentikan antrean email

- Gunakan pernyataan SQL berikut.

```
EXECUTE msdb.dbo.rds_sysmail_control stop;
GO
```

Bekerja dengan lampiran file

Ekstensi lampiran file berikut tidak didukung di pesan Database Mail messages dari RDS di SQL Server: .ade, .adp, .apk, .appx, .appxbundle, .bat, .bak, .cab, .chm, .cmd, .com, .cpl, .dll, .dmg, .exe, .hta, .inf and .wsh.

Database Mail menggunakan konteks keamanan Microsoft Windows dari pengguna saat ini untuk mengontrol akses ke file. Pengguna yang login dengan Autentikasi SQL Server tidak dapat melampirkan file menggunakan parameter `@file_attachments` dengan prosedur `sp_send_dbmail` yang disimpan. Windows tidak mengizinkan SQL Server untuk memberikan kredensial dari komputer jarak jauh ke komputer jarak jauh lainnya. Oleh karena itu, Database Mail tidak dapat melampirkan file dari jaringan bersama saat perintah dijalankan dari komputer selain komputer yang menjalankan SQL Server.

Namun, Anda dapat menggunakan pekerjaan SQL Server Agent untuk melampirkan file. Untuk informasi sselengkapnya tentang SQL Server Agent, lihat Menggunakan SQL Server Agent dan SQL Server Agent di dokumentasi Microsoft.

Pertimbangan untuk deployment multi-AZ

Ketika Anda mengonfigurasikan Database Mail di instans DB Multi-AZ DB, konfigurasi tidak secara otomatis diperbanyak ke sekunder. Kami merekomendasikan untuk mengonversi instans Multi-AZ ke instans Single-AZ, mengonfigurasi Database Mail, kemudian mengonversikan kembali instans DB ke Multi-AZ. Kemudian, kedua node primer dan sekunder memiliki konfigurasi Database Mail.

Jika Anda membuat replika baca dari instans Multi-AZ yang memiliki Database Mail yang dikonfigurasi, replika tersebut akan mewarisi konfigurasi tersebut, namun tanpa kata sandi ke server SMTP. Perbarui akun Database Mail dengan kata sandi.

Amazon RDS for SQL Server mendukung penyimpanan instans lokal untuk basis data tempdb

Penyimpanan instans menyediakan penyimpanan tingkat blok sementara untuk instans DB Anda. Penyimpanan ini terletak pada disk yang secara fisik terpasang pada komputer host. Disk ini memiliki penyimpanan instans Non-Volatile Memory Express (NVMe) yang didasarkan pada solid-state drive (SSD). Penyimpanan tersebut dioptimalkan untuk latensi rendah, performa I/O acak sangat tinggi, dan throughput baca berurutan.

Dengan menempatkan file data `tempdb` dan file log `tempdb` di penyimpanan instans, Anda dapat mencapai latensi baca dan tulis yang lebih rendah dibandingkan dengan penyimpanan standar berdasarkan Amazon EBS.

> **Note**
>
> File log dari file dan basis data untuk basis data SQL Server tidak ditempatkan di penyimpanan instans.

Mengaktifkan penyimpanan instans

Ketika RDS menyediakan instans DB dengan salah satu kelas instans berikut, basis data `tempdb` secara otomatis ditempatkan ke penyimpanan instans:

- db.m5d
- db.r5d
- db.x2iedn

Untuk mengaktifkan penyimpanan instans, lakukan salah satu hal berikut:

- Buat instans DB SQL Server menggunakan salah satu dari jenis instans ini. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.
- Mengubah instans DB SQL Server yang ada untuk menggunakan salah satunya. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Penyimpanan instans tersedia di semua Wilayah AWS di mana satu atau beberapa jenis instans ini didukung. Untuk informasi selengkapnya tentang daftar kelas instans `db.m5d` dan `db.r5d`, lihat

Kelas instans DB . Untuk informasi selengkapnya tentang kelas instans yang didukung oleh Amazon RDS for SQL Server, lihat Dukungan kelas instans DB untuk Microsoft SQL Server.

Pertimbangan lokasi dan ukuran file

Pada instans tanpa penyimpanan instans, RDS menyimpan data `tempdb` dan file log di direktori `D:\rdsdbdata\DATA`. Kedua file mulai dari 8 MB secara default.

Pada instans dengan penyimpanan instans, RDS menyimpan data `tempdb` dan file log di direktori `T:\rdsdbdata\DATA`.

Saat `tempdb` hanya memiliki satu file data (`tempdb.mdf`) dan satu file log (`templog.ldf`), `templog.ldf` dimulai pada 8 MB secara default dan `tempdb.mdf` dimulai pada 80% atau lebih dari kapasitas penyimpanan instans. Dua puluh persen dari kapasitas penyimpanan atau 200 GB, mana pun yang kurang, tetap bebas untuk memulai. Beberapa file data `tempdb` membagi ruang disk 80% secara merata, sementara file log selalu memiliki ukuran awal 8-MB.

Misalnya, jika Anda mengubah kelas instans DB Anda dari `db.m5.2xlarge` ke `db.m5d.2xlarge`, ukuran file data `tempdb` meningkat dari 8 MB masing-masing hingga 234 GB secara total.

> **Note**
>
> Selain data `tempdb` dan file log di penyimpanan instans (`T:\rdsdbdata\DATA`), Anda masih dapat membuat data dan file log `tempdb` ekstra pada volume data (`D:\rdsdbdata\DATA`). File-file tersebut selalu memiliki ukuran awal 8 MB.

Pertimbangan backup

Anda mungkin perlu untuk mempertahankan backup untuk waktu yang lama, menimbulkan biaya dari waktu ke waktu. Blok data dan log `tempdb` dapat berubah sangat sering tergantung pada beban kerja. Hal ini dapat sangat meningkatkan ukuran snapshot DB.

Saat `tempdb` ada pada penyimpanan instans, snapshot tidak mencakup file sementara. Ini berarti bahwa ukuran snapshot lebih kecil dan mengkonsumsi lebih sedikit alokasi backup gratis dibandingkan dengan penyimpanan EBS saja.

Kesalahan disk penuh

Jika Anda menggunakan semua ruang yang tersedia di penyimpanan instans, Anda mungkin menerima kesalahan seperti berikut:

- Log transaksi untuk basis data 'tempdb' penuh karena 'ACTIVE_TRANSACTION'.
- Tidak dapat mengalokasikan ruang untuk objek 'dbo.SORT sementara menjalankan penyimpanan: 140738941419520' dalam basis data 'tempdb' karena filegroup 'PRIMARY' penuh. Membuat ruang disk dengan menghapus file yang tidak diperlukan, membuang objek dalam filegroup, menambahkan file tambahan ke filegroup, atau menyalakan autogrowth untuk file yang ada di filegroup.

Anda dapat melakukan satu atau beberapa hal berikut ketika penyimpanan instans penuh:

- Menyesuaikan beban kerja Anda atau cara Anda menggunakan `tempdb`.
- Menaikkan skala untuk menggunakan kelas instans DB dengan penyimpanan NVMe lebih banyak.
- Berhenti menggunakan penyimpanan instans, dan menggunakan kelas instans dengan hanya penyimpanan EBS.
- Gunakan mode campuran dengan menambahkan data sekunder atau file log untuk `tempdb` pada volume EBS.

Menghapus penyimpanan instans

Untuk menghapus penyimpanan instans, ubah instans DB SQL Server Anda untuk menggunakan jenis instans yang tidak mendukung penyimpanan instans, seperti db.m5, db.r5, atau db.x1e.

> **Note**
>
> Ketika Anda menghapus penyimpanan instans, file-file sementara dipindahkan ke direktori `D:\rdsdbdata\DATA` dan dikurangi ukurannya ke 8 MB.

Menggunakan kejadian diperpanjang dengan server Amazon RDS for Microsoft SQL Server

Anda dapat menggunakan kejadian diperpanjang di Microsoft SQL Server untuk menangkap informasi debugging dan pemecahan masalah untuk Amazon RDS for SQL Server. Peristiwa diperpanjang menggantikan SQL Trace dan Server Profiler, yang tidak digunakan lagi oleh Microsoft. Peristiwa diperpanjang mirip dengan jejak profiler tetapi dengan kontrol yang lebih rinci pada kejadian yang dilacak. Peristiwa diperpanjang didukung untuk SQL Server versi 2014 dan kemudian di Amazon RDS. Untuk informasi selengkapnya, lihat Gambaran umum kejadian diperpanjang di dokumentasi Microsoft.

Peristiwa diperpanjang diaktifkan secara otomatis untuk pengguna dengan hak pengguna master di Amazon RDS for SQL Server.

Topik

Batasan dan rekomendasi

Saat menggunakan kejadian diperpanjang di RDS untuk SQL Server, batasan berikut berlaku:

- Peristiwa diperpanjang didukung hanya untuk Edisi Perusahaan dan Standar.
- Anda tidak dapat mengubah sesi kejadian diperpanjang default.
- Pastikan untuk mengatur mode partisi memori sesi ke `NONE`.
- Mode retensi kejadian sesi dapat berupa `ALLOW_SINGLE_EVENT_LOSS` atau `ALLOW_MULTIPLE_EVENT_LOSS`.
- Target Event Tracing for Windows (ETW) tidak didukung.
- Pastikan bahwa target file berada di direktori `D:\rdsdbdata\log`.
- Untuk target pencocokan pasangan, atur properti `respond_to_memory_pressure` ke 1.
- Memori target ring buffer tidak boleh lebih besar dari 4 MB.
- Tindakan berikut tidak didukung:

- `debug_break`
- `create_dump_all_threads`
- `create_dump_single_threads`

- Peristiwa `rpc_completed` didukung pada versi berikut dan yang lebih baru: 15.0.4083.2, 14.0.3370.1, 13.0.5865.1, 12.0.6433.1, 11.0.7507.2.

Mengkonfigurasi kejadian diperpanjang di RDS untuk SQL Server

Pada RDS untuk SQL Server, Anda dapat mengkonfigurasi nilai-nilai parameter tertentu dari sesi kejadian diperpanjang. Tabel berikut menjelaskan parameter yang dapat dikonfigurasi ini.

Nama parameter	Deskripsi
`xe_session_max_memory`	Tentukan jumlah maksimum memori untuk dialokasikan u dengan pengaturan `max_memory` sesi kejadian.
`xe_session_max_event_size`	Menentukan ukuran memori maksimum yang diperbolehk pengaturan `max_event_size` sesi kejadian.
`xe_session_max_dispatch_latency`	Tentukan jumlah waktu kejadian di-buffer dalam memori s Nilai ini sesuai dengan pengaturan `max_dispatch_lat`
`xe_file_target_size`	Tentukan ukuran maksimum target file. Nilai ini sesuai de
`xe_file_retention`	Menentukan waktu retensi dalam hari untuk file yang diha

> **Note**
>
> Pengaturan `xe_file_retention` ke nol menyebabkan file .xel dihapus secara otomatis setelah kunci pada file-file ini dirilis oleh SQL Server. Kunci dilepaskan setiap kali file .xel mencapai batas ukuran yang ditetapkan dalam `xe_file_target_size`.

Anda dapat menggunakan prosedur tersimpan `rdsadmin.dbo.rds_show_configuration` untuk menunjukkan nilai-nilai parameter saat ini. Misalnya, gunakan pernyataan SQL berikut untuk melihat pengaturan `xe_session_max_memory` saat ini.

```
exec rdsadmin..rds_show_configuration 'xe_session_max_memory'
```

Anda dapat menggunakan prosedur `rdsadmin.dbo.rds_set_configuration` yang disimpan untuk memodifikasinya. Misalnya, gunakan pernyataan SQL berikut untuk mengatur `xe_session_max_memory` ke 4 MB.

```
exec rdsadmin..rds_set_configuration 'xe_session_max_memory', 4
```

Pertimbangan untuk deployment multi-AZ

Ketika Anda membuat sesi kejadian diperpanjang pada instans DB primer, sesi tersebut tidak menyebar ke replika siaga. Anda dapat melakukan failover dan membuat sesi kejadian diperpanjang pada instans DB primer yang baru. Atau Anda dapat menghapus dan kemudian menambahkan kembali konfigurasi multi-AZ untuk menyebarkan sesi kejadian diperpanjang ke replika siaga. RDS menghentikan semua sesi kejadian diperpanjang non default pada replika siaga, sehingga sesi ini tidak mengkonsumsi sumber daya pada replika siaga. Karena ini, setelah replika siaga menjadi instans DB primer, pastikan untuk secara manual memulai sesi kejadian diperpanjang pada instans primer baru.

> **Note**
>
> Pendekatan ini berlaku untuk Always On Availability Group dan Database Mirroring.

Anda juga dapat menggunakan pekerjaan SQL Server Agent untuk melacak replika siaga dan memulai sesi jika replika siaga menjadi replika utama. Mislanya, gunakan kueri berikut dalam langkah pekerjaan SQL Server Agent untuk memulai ulang sesi kejadian pada instans DB primer.

```
BEGIN
    IF (DATABASEPROPERTYEX('rdsadmin','Updateability')='READ_WRITE'
    AND DATABASEPROPERTYEX('rdsadmin','status')='ONLINE'
    AND (DATABASEPROPERTYEX('rdsadmin','Collation') IS NOT NULL OR
 DATABASEPROPERTYEX('rdsadmin','IsAutoClose')=1)
    )
    BEGIN
        IF NOT EXISTS (SELECT 1 FROM sys.dm_xe_sessions WHERE name='xe1')
            ALTER EVENT SESSION xe1 ON SERVER STATE=START
        IF NOT EXISTS (SELECT 1 FROM sys.dm_xe_sessions WHERE name='xe2')
            ALTER EVENT SESSION xe2 ON SERVER STATE=START
```

```
    END
END
```

Kueri ini memulai ulang sesi kejadian `xe1` dan `xe2` pada instans DB primer jika sesi ini berada dalam keadaan berhenti. Anda juga dapat menambahkan jadwal dengan interval yang nyaman untuk kueri ini.

Melakukan kueri file kejadian diperpanjang

Anda dapat menggunakan SQL Server Management Studio atau fungsi `sys.fn_xe_file_target_read_file` untuk melihat data dari kejadian diperpanjang yang menggunakan target file. Untuk informasi selengkapnya tentang fungsi ini, lihat sys.fn_xe_file_target_read_file (Transact-SQL) di dokumentasi Microsoft.

Target file kejadian diperpanjang hanya dapat menulis file ke direktori `D:\rdsdbdata\log` pada RDS untuk SQL Server.

Misalnya, gunakan kueri SQL berikut untuk menyusun daftar dari isi semua file sesi kejadian diperpanjang yang namanya dimulai dengan `xe`.

```
SELECT * FROM sys.fn_xe_file_target_read_file('d:\rdsdbdata\log\xe*', null,null,null);
```

Akses ke cadangan log transaksi dengan RDS for SQL Server

Dengan akses ke cadangan log transaksi untuk RDS for SQL Server, Anda dapat menampilkan daftar file cadangan log transaksi untuk basis data dan menyalinnya ke bucket Amazon S3 target. Dengan menyalin cadangan log transaksi di bucket Amazon S3, Anda dapat menggunakannya dalam kombinasi dengan cadangan basis data lengkap dan diferensial untuk melakukan pemulihan basis data titik waktu. Anda menggunakan prosedur tersimpan RDS untuk mengatur akses ke cadangan log transaksi, menampilkan daftar cadangan log transaksi yang tersedia, dan menyalinnya ke bucket Amazon S3 Anda.

Akses ke cadangan log transaksi memberikan kemampuan dan manfaat berikut:

- Tampilkan daftar dan lihat metadata cadangan log transaksi yang tersedia untuk basis data pada instans DB RDS for SQL Server.
- Salin cadangan log transaksi yang tersedia dari RDS for SQL Server ke bucket Amazon S3 target.
- Lakukan pemulihan basis data titik waktu tanpa perlu memulihkan seluruh instans DB. Untuk informasi selengkapnya tentang memulihkan instans DB ke suatu titik waktu, lihat Memulihkan instans DB dengan waktu yang ditentukan.

Ketersediaan dan dukungan

Akses ke cadangan log transaksi didukung di semua Wilayah AWS. Akses ke cadangan log transaksi tersedia untuk semua edisi dan versi Microsoft SQL Server yang didukung di Amazon RDS.

Persyaratan

Persyaratan berikut harus dipenuhi sebelum mengaktifkan akses ke cadangan log transaksi:

- Pencadangan otomatis harus diaktifkan pada instans DB dan retensi cadangan harus diatur ke nilai satu hari atau lebih. Untuk informasi selengkapnya tentang mengaktifkan pencadangan otomatis dan mengonfigurasi kebijakan retensi, lihat Mengaktifkan pencadangan otomatis.
- Bucket Amazon S3 harus ada di akun dan Wilayah yang sama dengan instans DB sumber. Sebelum mengaktifkan akses ke cadangan log transaksi, pilih bucket Amazon S3 yang sudah ada atau buat bucket baru yang akan digunakan untuk file cadangan log transaksi Anda.
- Kebijakan izin bucket Amazon S3 harus dikonfigurasi sebagai berikut untuk memungkinkan Amazon RDS menyalin file log transaksi ke dalamnya:

  1. Atur properti kepemilikan akun objek pada bucket ke Pemilik Bucket Pilihan.

2. Tambahkan kebijakan berikut. Tidak akan ada kebijakan secara default, jadi gunakan Daftar Kontrol Akses (ACL) bucket untuk mengedit kebijakan bucket dan menambahkannya.

   Contoh berikut menggunakan ARN untuk menentukan sumber daya. Sebaiknya gunakan kunci konteks kondisi global `SourceArn` dan `SourceAccount` dalam relasi kepercayaan berbasis sumber daya untuk membatasi izin layanan ke sumber daya tertentu. Untuk informasi selengkapnya tentang menggunakan ARN, lihat Amazon Resource Name (ARN) dan Bekerja dengan Amazon Resource Name (ARN) di Amazon RDS.

   Example kebijakan izin Amazon S3 untuk akses ke cadangan log transaksi

```
    {
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Only allow writes to my bucket with bucket owner full control",
            "Effect": "Allow",
            "Principal": {
                "Service": "backups.rds.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::{customer_bucket}/{customer_path}/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:sourceAccount": "{customer_account}",
                    "aws:sourceArn": "{db_instance_arn}"
                }
            }
        }
    ]
}
```

- Peran AWS Identity and Access Management (IAM) untuk mengakses bucket Amazon S3. Jika Anda sudah memiliki peran IAM, Anda dapat menggunakannya. Anda dapat memilih untuk dibuatkan peran IAM baru ketika menambahkan opsi `SQLSERVER_BACKUP_RESTORE` dengan menggunakan AWS Management Console. Alternatifnya, Anda dapat membuatnya secara manual. Untuk informasi selengkapnya tentang membuat dan mengonfigurasi peran IAM dengan

SQLSERVER_BACKUP_RESTORE, lihat Membuat peran IAM secara manual untuk pencadangan dan pemulihan native.

- Opsi SQLSERVER_BACKUP_RESTORE harus ditambahkan ke grup opsi pada instans DB Anda. Untuk informasi selengkapnya tentang menambahkan opsi SQLSERVER_BACKUP_RESTORE, lihat Dukungan untuk pencadangan dan pemulihan native di SQL Server.

  > **Note**
  >
  > Jika instans DB Anda memiliki enkripsi penyimpanan aktif, tindakan dan kunci AWS KMS (KMS) harus disediakan dalam peran IAM yang disediakan dalam grup opsi pencadangan dan pemulihan native.

  Secara opsional, jika Anda bermaksud menggunakan prosedur tersimpan `rds_restore_log` untuk melakukan pemulihan basis data titik waktu, sebaiknya gunakan jalur Amazon S3 yang sama untuk grup opsi pencadangan dan pemulihan native dan akses ke cadangan log transaksi. Metode ini memastikan bahwa ketika Amazon RDS mengambil peran dari grup opsi untuk melakukan fungsi pemulihan log, layanan ini memiliki akses untuk mengambil cadangan log transaksi dari jalur Amazon S3 yang sama.
- Jika instans DB dienkripsi, terlepas dari jenis enkripsi (kunci yang dikelola AWS atau kunci yang dikelola Pelanggan), Anda harus memberikan kunci KMS yang dikelola Pelanggan dalam peran IAM dan dalam prosedur tersimpan `rds_tlog_backup_copy_to_S3`.

Batasan dan rekomendasi

Akses ke cadangan log transaksi memiliki batasan dan rekomendasi sebagai berikut:

- Anda dapat menampilkan daftar dan menyalin hingga tujuh hari terakhir cadangan log transaksi untuk instans DB apa pun yang memiliki retensi cadangan yang dikonfigurasi antara satu hingga 35 hari.
- Bucket Amazon S3 yang digunakan untuk akses ke cadangan log transaksi harus ada di akun dan Wilayah yang sama dengan instans DB sumber. Penyalinan lintas akun dan lintas Wilayah tidak didukung.
- Hanya satu bucket Amazon S3 yang dapat dikonfigurasi sebagai target untuk menyalin cadangan log transaksi. Anda dapat memilih bucket Amazon S3 target baru dengan prosedur tersimpan

`rds_tlog_copy_setup`. Untuk informasi selengkapnya tentang memilih bucket Amazon S3 baru, lihat Mengatur akses ke cadangan log transaksi.

- Anda tidak dapat menentukan kunci KMS saat menggunakan prosedur tersimpan `rds_tlog_backup_copy_to_S3` jika instans RDS Anda tidak diaktifkan untuk enkripsi penyimpanan.
- Penyalinan multi-akun tidak didukung. Peran IAM yang digunakan untuk menyalin hanya akan mengizinkan akses tulis ke bucket Amazon S3 dalam akun pemilik instans DB.
- Hanya dua tugas konkuren dari jenis apa pun dapat dijalankan pada instans DB RDS for SQL Server.
- Hanya satu tugas penyalinan yang dapat dijalankan untuk satu basis data pada waktu tertentu. Jika Anda ingin menyalin cadangan log transaksi untuk beberapa basis data pada instans DB, gunakan tugas penyalinan terpisah untuk setiap basis data.
- Jika Anda menyalin cadangan log transaksi yang sudah ada dengan nama yang sama di bucket Amazon S3, cadangan log transaksi yang ada akan ditimpa.
- Anda hanya dapat menjalankan prosedur tersimpan yang disediakan dengan akses ke cadangan log transaksi pada instans DB primer. Anda tidak dapat menjalankan prosedur tersimpan ini pada replika baca RDS for SQL Server atau pada instans sekunder dari klaster DB Multi-AZ.
- Jika DB RDS for SQL Server instans di-boot ulang saat prosedur tersimpan `rds_tlog_backup_copy_to_S3` sedang berjalan, tugas ini akan secara otomatis dimulai ulang dari awal ketika instans DB kembali online. Cadangan log transaksi apa pun yang telah disalin ke bucket Amazon S3 saat tugas berjalan sebelum boot ulang akan ditimpa.
- Basis data sistem Microsoft SQL Server dan basis data `RDSAdmin` tidak dapat dikonfigurasi untuk akses ke cadangan log transaksi.
- Penyalinan ke bucket yang dienkripsi oleh SSE-KMS tidak didukung.

Mengatur akses ke cadangan log transaksi

Untuk mengatur akses ke cadangan log transaksi, lengkapi daftar persyaratan di bagian Persyaratan, lalu jalankan prosedur tersimpan `rds_tlog_copy_setup`. Prosedur ini akan mengaktifkan akses ke fitur cadangan log transaksi di tingkat instans DB. Anda tidak perlu menjalankannya untuk setiap basis data individual pada instans DB.

> **Important**
>
> Pengguna basis data harus diberi peran `db_owner` dalam SQL Server pada setiap basis data untuk mengonfigurasi dan menggunakan akses ke fitur cadangan log transaksi.

Example penggunaan:

```
exec msdb.dbo.rds_tlog_copy_setup
@target_s3_arn='arn:aws:s3:::mybucket/myfolder';
```

Parameter berikut diperlukan:

- `@target_s3_arn` – ARN bucket Amazon S3 target untuk menyalin file cadangan log transaksi.

Example mengatur bucket target Amazon S3:

```
exec msdb.dbo.rds_tlog_copy_setup @target_s3_arn='arn:aws:s3:::accesstlogs-
testbucket/mytestdb1';
```

Untuk memvalidasi konfigurasi, panggil prosedur tersimpan `rds_show_configuration`.

Example memvalidasi konfigurasi:

```
exec rdsadmin.dbo.rds_show_configuration @name='target_s3_arn_for_tlog_copy';
```

Untuk mengubah akses ke cadangan log transaksi agar mengarah ke bucket Amazon S3 yang berbeda, Anda dapat melihat nilai bucket Amazon S3 saat ini dan menjalankan kembali prosedur tersimpan `rds_tlog_copy_setup` menggunakan nilai baru untuk `@target_s3_arn`.

Example melihat bucket Amazon S3 yang ada yang dikonfigurasi untuk akses ke cadangan log transaksi

```
exec rdsadmin.dbo.rds_show_configuration @name='target_s3_arn_for_tlog_copy';
```

Example memperbarui ke bucket Amazon S3 target baru

```
exec msdb.dbo.rds_tlog_copy_setup
 @target_s3_arn='arn:aws:s3:::mynewbucket/mynewfolder';
```

Menampilkan daftar cadangan log transaksi yang tersedia

Dengan RDS for SQL Server, basis data yang dikonfigurasi untuk menggunakan model pemulihan penuh dan retensi cadangan instans DB yang diatur ke satu atau beberapa hari akan memiliki cadangan log transaksi yang diaktifkan secara otomatis. Dengan mengaktifkan akses ke cadangan log transaksi, cadangan log transaksi tersebut dari rentang waktu hingga tujuh hari akan tersedia untuk Anda salin ke bucket Amazon S3 Anda.

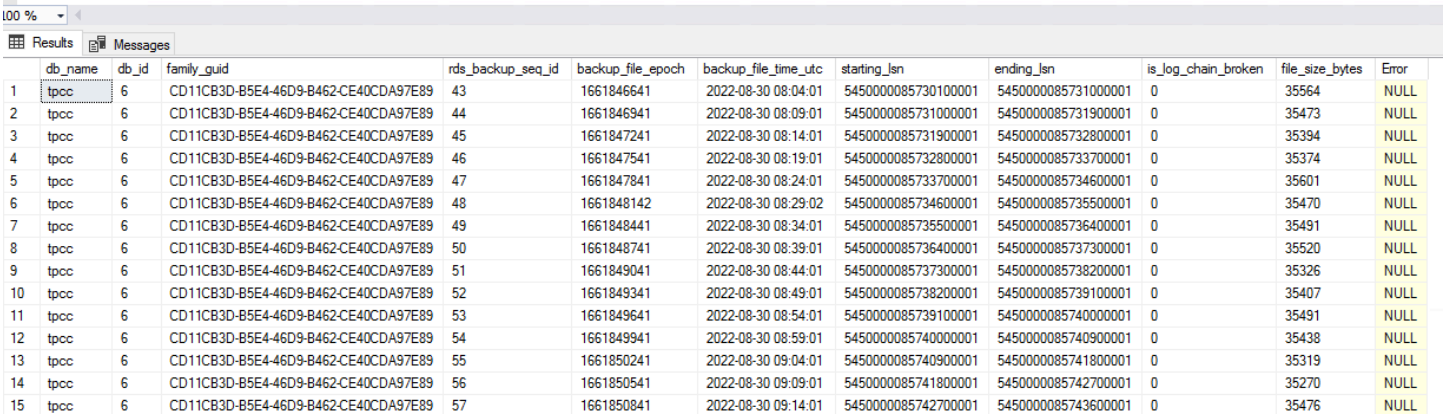
Setelah Anda mengaktifkan akses ke cadangan log transaksi, Anda dapat mulai menggunakannya untuk menampilkan daftar dan menyalin file cadangan log transaksi yang tersedia.

Menampilkan daftar cadangan log transaksi

Untuk menampilkan daftar semua cadangan log transaksi yang tersedia untuk basis data individual, panggil fungsi `rds_fn_list_tlog_backup_metadata`. Anda dapat menggunakan klausa `ORDER BY` atau `WHERE` saat memanggil fungsi.

Example menampilkan daftar dan memfilter file cadangan log transaksi yang tersedia

```
SELECT * from msdb.dbo.rds_fn_list_tlog_backup_metadata('mydatabasename');
SELECT * from msdb.dbo.rds_fn_list_tlog_backup_metadata('mydatabasename') WHERE
 rds_backup_seq_id = 3507;
SELECT * from msdb.dbo.rds_fn_list_tlog_backup_metadata('mydatabasename') WHERE
 backup_file_time_utc > '2022-09-15 20:44:01' ORDER BY backup_file_time_utc DESC;
```



100 %

Results Messages

	db_name	db_id	family_guid	rds_backup_seq_id	backup_file_epoch	backup_file_time_utc	starting_lsn	ending_lsn	is_log_chain_broken	file_size_bytes	Error
1	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	43	1661846641	2022-08-30 08:04:01	5450000085730100001	5450000085731000001	0	35564	NULL
2	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	44	1661846941	2022-08-30 08:09:01	5450000085731000001	5450000085731900001	0	35473	NULL
3	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	45	1661847241	2022-08-30 08:14:01	5450000085731900001	5450000085732800001	0	35394	NULL
4	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	46	1661847541	2022-08-30 08:19:01	5450000085732800001	5450000085733700001	0	35374	NULL
5	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	47	1661847841	2022-08-30 08:24:01	5450000085733700001	5450000085734600001	0	35601	NULL
6	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	48	1661848142	2022-08-30 08:29:02	5450000085734600001	5450000085735500001	0	35470	NULL
7	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	49	1661848441	2022-08-30 08:34:01	5450000085735500001	5450000085736400001	0	35491	NULL
8	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	50	1661848741	2022-08-30 08:39:01	5450000085736400001	5450000085737300001	0	35520	NULL
9	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	51	1661849041	2022-08-30 08:44:01	5450000085737300001	5450000085738200001	0	35326	NULL
10	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	52	1661849341	2022-08-30 08:49:01	5450000085738200001	5450000085739100001	0	35407	NULL
11	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	53	1661849641	2022-08-30 08:54:01	5450000085739100001	5450000085740000001	0	35491	NULL
12	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	54	1661849941	2022-08-30 08:59:01	5450000085740000001	5450000085740900001	0	35438	NULL
13	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	55	1661850241	2022-08-30 09:04:01	5450000085740900001	5450000085741800001	0	35319	NULL
14	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	56	1661850541	2022-08-30 09:09:01	5450000085741800001	5450000085742700001	0	35270	NULL
15	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	57	1661850841	2022-08-30 09:14:01	5450000085742700001	5450000085743600001	0	35476	NULL

Fungsi `rds_fn_list_tlog_backup_metadata` ini menampilkan output berikut:

Nama kolom	Jenis data	Deskripsi
`db_name`	sysname	Nama basis data yang disediakan untuk menampilkan daftar cadangan log transaksinya.
`db_id`	int	Pengidentifikasi basis data internal untuk parameter input `db_name`.
`family_guid`	uniqueidentifier	ID unik basis data asli saat pembuatan. Nilai ini tetap sama ketika basis data dipulihkan, bahkan ke nama basis data yang berbeda.
`rds_backup_seq_id`	int	ID yang digunakan RDS secara internal untuk mempertahankan nomor urutan untuk setiap file cadangan log transaksi.
`backup_file_epoch`	bigint	Waktu epoch saat file cadangan transaksi dihasilkan.
`backup_file_time_utc`	datetime	Nilai konversi waktu UTC untuk nilai `backup_file_epoch` .
`starting_lsn`	numeric(25,0)	Nomor urutan log dari catatan log pertama atau tertua untuk file cadangan log transaksi.
`ending_lsn`	numeric(25,0)	Nomor urutan log dari catatan log terakhir atau berikutnya untuk file cadangan log transaksi.

Nama kolom	Jenis data	Deskripsi
`is_log_chain_broken`	bit	Nilai boolean yang menunjukkan apakah rantai log terputus antara file cadangan log transaksi saat ini dan file cadangan log transaksi sebelumnya.
`file_size_bytes`	bigint	Ukuran cadangan transaksional diatur dalam byte.
`Error`	varchar(4000)	Pesan kesalahan jika fungsi `rds_fn_list_tlog_backup_metadata` mengeluarkan pengecualian. NULL jika tidak ada pengecualian.

Menyalin cadangan log transaksi

Untuk menyalin serangkaian cadangan log transaksi yang tersedia untuk basis data individual ke bucket Amazon S3 Anda, panggil prosedur tersimpan `rds_tlog_backup_copy_to_S3`. Prosedur tersimpan `rds_tlog_backup_copy_to_S3` akan memulai tugas baru untuk menyalin cadangan log transaksi.

> **Note**
>
> Prosedur tersimpan `rds_tlog_backup_copy_to_S3` akan menyalin cadangan log transaksi tanpa memvalidasi berdasarkan atribut `is_log_chain_broken`. Untuk alasan ini, Anda harus secara manual mengonfirmasi rantai log yang tidak terputus sebelum menjalankan prosedur tersimpan `rds_tlog_backup_copy_to_S3`. Untuk penjelasan selengkapnya, lihat Memvalidasi rantai log cadangan log transaksi.

Example penggunaan prosedur tersimpan **`rds_tlog_backup_copy_to_S3`**

```
exec msdb.dbo.rds_tlog_backup_copy_to_S3
 @db_name='mydatabasename',
 [@kms_key_arn='arn:aws:kms:region:account-id:key/key-id'],
 [@backup_file_start_time='2022-09-01 01:00:15'],
 [@backup_file_end_time='2022-09-01 21:30:45'],
 [@starting_lsn=149000000112100001],
 [@ending_lsn=149000000120400001],
 [@rds_backup_starting_seq_id=5],
```

```
 [@rds_backup_ending_seq_id=10];
```

Parameter input berikut tersedia:

Parameter	Deskripsi
`@db_name`	Nama basis data yang cadangan log transaksinya akan disalin
`@kms_key_arn`	ARN kunci KMS yang digunakan untuk mengenkripsi instans DB yang terenkripsi penyimpanannya.
`@backup_file_start_time`	Stempel waktu UTC seperti yang disediakan dari kolom `[backup_file_time_utc]` pada fungsi `rds_fn_list_tlog_backup_metadata` .
`@backup_file_end_time`	Stempel waktu UTC seperti yang disediakan dari kolom `[backup_file_time_utc]` pada fungsi `rds_fn_list_tlog_backup_metadata` .
`@starting_lsn`	Nomor urutan log (LSN) seperti yang disediakan dari kolom `[starting_lsn]` pada fungsi `rds_fn_list_tlog_backup_metadata` .
`@ending_lsn`	Nomor urutan log (LSN) seperti yang disediakan dari kolom `[ending_lsn]` pada fungsi `rds_fn_list_tlog_backup_metadata` .
`@rds_backup_starting_seq_id`	Urutan ID seperti yang disediakan dari kolom `[rds_backup_seq_id]` pada fungsi `rds_fn_list_tlog_backup_metadata` .
`@rds_backup_ending_seq_id`	Urutan ID seperti yang disediakan dari kolom `[rds_backup_seq_id]` pada fungsi `rds_fn_list_tlog_backup_metadata` .

Anda dapat menentukan satu set parameter waktu, LSN, atau ID urutan. Hanya satu set parameter yang diperlukan.

Anda juga dapat menentukan hanya satu parameter di salah satu set. Misalnya, dengan memberikan nilai hanya untuk parameter `backup_file_end_time`, semua file cadangan log transaksi yang tersedia sebelum waktu tersebut dalam batas tujuh hari akan disalin ke bucket Amazon S3 Anda.

Berikut ini adalah kombinasi parameter input yang valid untuk prosedur tersimpan `rds_tlog_backup_copy_to_S3`.

Parameter yang disediakan	Hasil yang diharapkan
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name = 'testdb1', @backup_file_start_time='2022-08-23 00:00:00', @backup_file_end_time='2022-08-30 00:00:00';`	Menyalin cadangan log transaksi dari tujuh hari terakhir serta dalam rentang `backup_file_start_time` dan `backup_file_end_time` yang disediakan. Dalam contoh ini, prosedur tersimpan akan menyalin cadangan log transaksi yang dihasilkan antara '2022-08-23 00:00:00' dan '2022-08-30 00:00:00'.
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name = 'testdb1',`	Menyalin cadangan log transaksi dari tujuh hari terakhir dan mulai dari `backup_file_start_time` yang disediaka

Parameter yang disediakan	Hasil yang diharapkan
`@backup_file_start_time='2022-08-23 00:00:00';`	n. Dalam contoh ini, prosedur tersimpan akan menyalin cadangan log transaksi dari '2022-08-23 00:00:00' hingga cadangan log transaksi terbaru.
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name = 'testdb1', @backup_file_end_time='2022-08-30 00:00:00';`	Menyalin cadangan log transaksi dari tujuh hari terakhir hingga `backup_file_end_time` yang disediaka n. Dalam contoh ini, prosedur tersimpan akan menyalin cadangan log transaksi dari '2022-08-23 00:00:00' hingga '2022-08-30 00:00:00'.

Parameter yang disediakan	Hasil yang diharapkan	
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name='testdb1', @starting_lsn =14900000 00040007, @ending_lsn = 149000000 0050009;`	Menyalin cadangan log transaksi yang tersedia dari tujuh hari terakhir dan berada dalam rentang `starting_lsn` dan `ending_lsn` yang disediakan. Dalam contoh ini, prosedur tersimpan akan menyalin cadangan log transaksi dari tujuh hari terakhir dengan rentang LSN antara 1490000000040007 dan 149000000 0050009.	

Parameter yang disediakan	Hasil yang diharapkan	
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name='testdb1', @starting_lsn =14900000 00040007;`	Menyalin cadangan log transaksi yang tersedia dari tujuh hari terakhir, mulai dari `starting_lsn` yang disediakan. Dalam contoh ini, prosedur tersimpan akan menyalin cadangan log transaksi dari LSN 1490000000040007 hingga cadangan log transaksi terbaru.	
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name='testdb1', @ending_lsn =14900000 00050009;`	Menyalin cadangan log transaksi yang tersedia dari tujuh hari terakhir, hingga `ending_lsn` yang disediakan. Dalam contoh ini, prosedur tersimpan akan menyalin cadangan log transaksi mulai dari tujuh hari terakhir hingga lsn 149000000 0050009.	

Parameter yang disediakan	Hasil yang diharapkan	
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name='testdb1', @rds_backup_starting_seq_id= 2000, @rds_backup_ending_seq_id= 5000;`	Menyalin cadangan log transaksi yang tersedia dari tujuh hari terakhir, dan berada dalam rentang `rds_backup_starting_seq_id` dan `rds_backup_ending_seq_id` yang disediakan. Dalam contoh ini, prosedur tersimpan akan menyalin cadangan log transaksi mulai dari tujuh hari terakhir dan dalam rentang id urutan cadangan rds yang disediakan, mulai dari seq_id 2000 hingga seq_id 5000.	

Parameter yang disediakan	Hasil yang diharapkan
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name='testdb1', @rds_backup_starting_seq_id= 2000;`	Menyalin cadangan log transaksi yang tersedia dari tujuh hari terakhir, mulai dari `rds_backup_starting_seq_id` yang disediakan. Dalam contoh ini, prosedur tersimpan akan menyalin cadangan log transaksi mulai dari seq_id 2000, hingga cadangan log transaksi terbaru.
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name='testdb1', @rds_backup_ending_seq_id= 5000;`	Menyalin cadangan log transaksi yang tersedia dari tujuh hari terakhir, hingga `rds_backup_ending_seq_id` yang disediakan. Dalam contoh ini, prosedur tersimpan akan menyalin cadangan log transaksi mulai dari tujuh hari terakhir, hingga seq_id 5000.

Parameter yang disediakan	Hasil yang diharapkan	
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name='testdb1', @rds_backup_starting_seq_id= 2000; @rds_backup_ending_seq_id= 2000;`	Menyalin cadangan log transaksi tunggal dengan `rds_backup_starting_seq_id` yang disediakan, jika tersedia dalam tujuh hari terakhir. Dalam contoh ini, prosedur tersimpan akan menyalin cadangan log transaksi tunggal yang memiliki seq_id 2000, jika ada dalam tujuh hari terakhir.	

Memvalidasi rantai log cadangan log transaksi

Basis data yang dikonfigurasi untuk akses ke cadangan log transaksi harus memiliki retensi cadangan otomatis yang aktif. Retensi cadangan otomatis mengatur basis data pada instans DB ke model pemulihan `FULL`. Untuk mendukung pemulihan titik waktu untuk basis data, hindari mengubah model pemulihan basis data, yang dapat mengakibatkan rantai log terputus. Kami menyarankan agar basis data tetap diatur ke model pemulihan `FULL`.

Untuk memvalidasi rantai log secara manual sebelum menyalin cadangan log transaksi, panggil fungsi `rds_fn_list_tlog_backup_metadata` dan tinjau nilai di kolom `is_log_chain_broken`. Nilai “1" menunjukkan rantai log terputus antara cadangan log saat ini dan cadangan log sebelumnya.

Contoh berikut menunjukkan rantai log yang terputus dalam output dari prosedur tersimpan `rds_fn_list_tlog_backup_metadata`.

rds_sequence_id	first_lsn	last_lsn	is_log_chain_broken
43	90023	90457	0
44	90457	90985	0
45	90987	92034	1

Dalam rantai log normal, nilai nomor urutan log (LSN) untuk first_lsn untuk rds_sequence_id yang diberikan harus cocok dengan nilai last_lsn di rds_sequence_id sebelumnya. Pada gambar, rds_sequence_id 45 memiliki nilai first_lsn 90987, yang tidak cocok dengan nilai last_lsn 90985 untuk rds_sequence_id 44 sebelumnya.

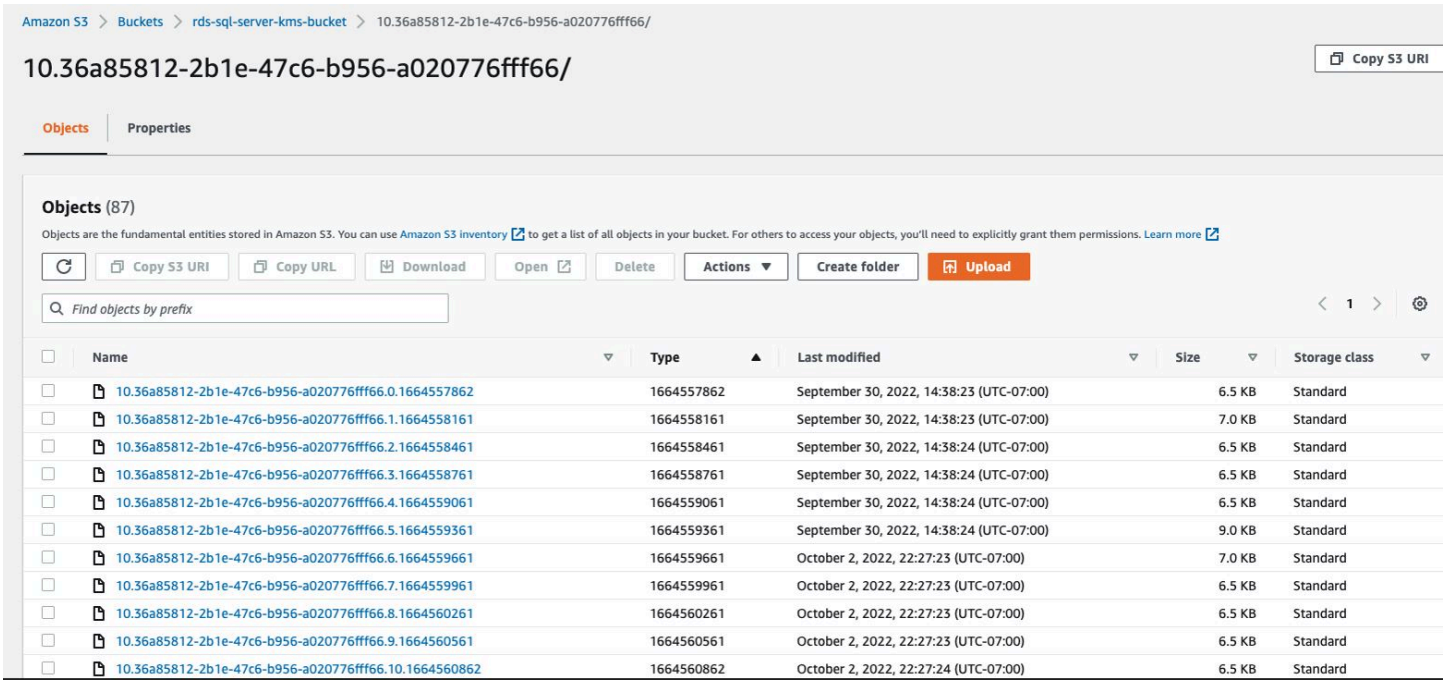
Untuk informasi selengkapnya tentang arsitektur log transaksi SQL Server dan nomor urutan log, lihat Transaction Log Logical Architecture dalam dokumentasi Microsoft SQL Server.

Struktur folder dan file bucket Amazon S3

Cadangan log transaksi memiliki struktur standar dan konvensi penamaan berikut dalam bucket Amazon S3:

- Folder baru dibuat dalam jalur `target_s3_arn` untuk setiap basis data dengan struktur penamaan `{db_id}.{family_guid}`.
- Di dalam folder, cadangan log transaksi memiliki struktur nama file `{db_id}.{family_guid}.{rds_backup_seq_id}.{backup_file_epoch}`.
- Anda dapat melihat detail `family_guid,db_id,rds_backup_seq_id and backup_file_epoch` dengan fungsi `rds_fn_list_tlog_backup_metadata`.

Contoh berikut menunjukkan folder dan struktur file dari satu set cadangan log transaksi dalam bucket Amazon S3.



Melacak status tugas

Untuk melacak status tugas penyalinan Anda, panggil prosedur tersimpan `rds_task_status`. Jika Anda tidak memberikan parameter apa pun, prosedur tersimpan akan menampilkan status semua tugas.

Example penggunaan:

```
exec msdb.dbo.rds_task_status
  @db_name='database_name',
  @task_id=ID_number;
```

Parameter berikut ini bersifat opsional:

- `@db_name` – Nama basis data yang status tugasnya akan ditampilkan.
- `@task_id` – ID tugas yang status tugasnya akan ditampilkan.

Example menampilkan daftar status untuk ID tugas tertentu:

```
exec msdb.dbo.rds_task_status @task_id=5;
```

Example menampilkan daftar status untuk basis data dan tugas tertentu:

```
exec msdb.dbo.rds_task_status@db_name='my_database',@task_id=5;
```

Example menampilkan daftar semua tugas dan statusnya untuk basis data tertentu:

```
exec msdb.dbo.rds_task_status @db_name='my_database';
```

Example menampilkan daftar semua tugas dan statusnya di instans DB saat ini:

```
exec msdb.dbo.rds_task_status;
```

Membatalkan tugas

Untuk membatalkan tugas yang berjalan, panggil prosedur tersimpan `rds_cancel_task`.

Example penggunaan:

```
exec msdb.dbo.rds_cancel_task @task_id=ID_number;
```

Parameter berikut diperlukan:

- `@task_id` – ID tugas yang akan dibatalkan. Anda dapat melihat ID tugas dengan memanggil prosedur tersimpan `rds_task_status`.

Untuk informasi selengkapnya tentang melihat dan membatalkan tugas yang berjalan, lihat Mengimpor dan mengekspor basis data SQL Server menggunakan pencadangan dan pemulihan native.

Pemecahan masalah akses ke cadangan log transaksi

Berikut ini adalah masalah yang mungkin Anda temui saat menggunakan prosedur tersimpan untuk akses ke cadangan log transaksi.

Prosedur Tersimpan	Pesan Kesalahan	Masalah	Saran pemecahan masalah
rds_tlog_copy_setup	Pencadangan dinonaktifkan pada instans DB ini. Aktifkan pencadangan instans DB dengan retensi setidaknya “1" dan coba lagi.	Pencadangan otomatis tidak diaktifkan untuk instans DB.	Retensi cadangan instans DB harus diaktifkan dengan retensi setidaknya satu hari. Untuk informasi selengkapnya tentang mengaktifkan cadangan otomatis dan mengonfigurasi retensi cadangan, lihat Periode retensi cadangan.
rds_tlog_copy_setup	Kesalahan saat menjalankan prosedur tersimpan rds_tlog_copy_setup. Hubungkan kembali ke titik akhir RDS dan coba lagi.	Terjadi kesalahan internal.	Hubungkan kembali ke titik akhir RDS dan jalankan prosedur tersimpan `rds_tlog_copy_setup` lagi.
rds_tlog_copy_setup	Menjalankan prosedur tersimpan rds_tlog_backup_copy_setup di dalam transaksi	Prosedur tersimpan dicoba dalam transaksi menggunakan `BEGIN` dan `END`.	Hindari menggunakan `BEGIN` dan `END` saat menjalankan prosedur tersimpan `rds_tlog_copy_setup` .

Prosedur Tersimpan	Pesan Kesalahan	Masalah	Saran pemecahan masalah
	tidak didukung. Verifikasi bahwa sesi tidak memiliki transaksi terbuka dan coba lagi.		
rds_tlog_copy_setup	Nama bucket S3 untuk parameter input `@target_s3_arn` harus berisi setidaknya satu karakter selain spasi.	Nilai yang salah diberikan untuk parameter input `@target_s3_arn` .	Pastikan parameter input `@target_s3_arn` menentukan ARN bucket Amazon S3 lengkap.

Prosedur Tersimpan	Pesan Kesalahan	Masalah	Saran pemecahan masalah
rds_tlog_copy_setup	Opsi SQLSERVER_BACKUP_RESTORE tidak diaktifkan atau sedang dalam proses diaktifkan. Aktifkan opsi atau coba lagi nanti.	Opsi SQLSERVER_BACKUP_RESTORE tidak diaktifkan pada instans DB atau baru saja diaktifkan dan menunggu aktivasi internal.	Aktifkan opsi SQLSERVER_BACKUP_RESTORE seperti yang ditentukan di bagian Persyaratan. Tunggu beberapa menit dan jalankan prosedur tersimpan rds_tlog_copy_setup lagi.
rds_tlog_copy_setup	ARN S3 target untuk parameter input @target_s3_arn tidak boleh kosong atau null.	Nilai NULL diberikan untuk parameter input @target_s3_arn, atau nilainya tidak diberikan.	Pastikan parameter input @target_s3_arn menentukan ARN bucket Amazon S3 lengkap.

Prosedur Tersimpan	Pesan Kesalahan	Masalah	Saran pemecahan masalah
rds_tlog_copy_setup	ARN S3 target untuk parameter input `@target_s3_arn` harus diawali dengan arn:aws.	Parameter input `@target_s3_arn` diberikan tanpa `arn:aws` di bagian depan.	Pastikan parameter input `@target_s3_arn` menentukan ARN bucket Amazon S3 lengkap.
rds_tlog_copy_setup	ARN S3 target sudah diatur ke nilai yang diberikan.	Prosedur tersimpan `rds_tlog_copy_setup` sebelumnya berjalan dan dikonfigurasi dengan ARN bucket Amazon S3.	Untuk mengubah nilai bucket Amazon S3 untuk akses ke cadangan log transaksi, berikan `target S3 ARN` yang berbeda.

Prosedur Tersimpan	Pesan Kesalahan	Masalah	Saran pemecahan masalah
rds_tlog_copy_setup	Tidak dapat menghasilkan kredensial untuk mengaktifkan Akses ke Cadangan Log Transaksi. Konfirmasikan ARN jalur S3 yang diberikan dengan `rds_tlog_copy_setup` , dan coba lagi nanti.	Ada kesalahan yang tidak ditentukan saat menghasilkan kredensial untuk mengaktifkan akses ke cadangan log transaksi.	Tinjau konfigurasi pengaturan Anda dan coba lagi.

Prosedur Tersimpan	Pesan Kesalahan	Masalah	Saran pemecahan masalah
rds_tlog_copy_setup	Anda tidak dapat menjalankan prosedur tersimpan rds_tlog_copy_setup saat ada tugas yang tertunda. Tunggu tugas tertunda selesai dan coba lagi.	Hanya dua tugas yang dapat dijalankan kapan saja. Ada tugas tertunda yang menunggu penyelesaian.	Lihat tugas yang tertunda dan tunggu tugas tersebut selesai. Untuk informasi selengkapnya tentang pemantauan status tugas, lihat Melacak status tugas.
rds_tlog_backup_copy_to_S3	Tugas penyalinan file cadangan T-log telah dikeluarkan untuk basis data: %s dengan Id tugas: %d, coba lagi nanti.	Hanya satu tugas penyalinan yang dapat dijalankan kapan saja untuk basis data tertentu. Ada tugas penyalinan tertunda yang menunggu penyelesaian.	Lihat tugas yang tertunda dan tunggu tugas tersebut selesai. Untuk informasi selengkapnya tentang pemantauan status tugas, lihat Melacak status tugas.

Prosedur Tersimpan	Pesan Kesalahan	Masalah	Saran pemecahan masalah
rds_tlog_backup_copy_to_S3	Setidaknya satu dari tiga set parameter ini harus disediakan. SET-1:(@backup_file_start_time, @backup_file_end_time) | SET-2:(@starting_lsn, @ending_lsn) | SET-3:(@rds_backup_starting_seq_id, @rds_backup_ending_seq_id)	Tidak ada satu pun dari tiga set parameter yang disediakan, atau set parameter yang disediakan tidak memiliki parameter yang diperlukan.	Anda dapat menentukan parameter waktu, lsn, atau ID urutan. Satu set dari tiga set parameter ini diperlukan. Untuk informasi selengkapnya tentang parameter yang diperlukan, lihat Menyalin cadangan log transaksi.

Prosedur Tersimpan	Pesan Kesalahan	Masalah	Saran pemecahan masalah
rds_tlog_backup_copy_to_S3	Pencadangan dinonaktifkan pada instans Anda. Aktifkan pencadangan dan coba lagi dalam beberapa waktu.	Pencadangan otomatis tidak diaktifkan untuk instans DB.	Untuk informasi selengkapnya tentang mengaktifkan cadangan otomatis dan mengonfigurasi retensi cadangan, lihat Periode retensi cadangan.
rds_tlog_backup_copy_to_S3	Tidak dapat menemukan basis data yang diberikan %s.	Nilai yang diberikan untuk parameter input `@db_name` tidak cocok dengan nama basis data pada instans DB.	Gunakan nama basis data yang benar. Untuk menampilkan daftar semua basis data berdasarkan nama, jalankan `SELECT * from sys.databases`
rds_tlog_backup_copy_to_S3	Tidak dapat menjalankan prosedur tersimpan rds_tlog_backup_copy_to_S3 untuk database sistem SQL Server atau database rdsadmin.	Nilai yang disediakan untuk parameter input `@db_name` cocok dengan nama basis data sistem SQL Server atau basis data RDSAdmin.	Basis data berikut tidak diizinkan untuk digunakan dengan akses ke cadangan log transaksi: `master, model, msdb, tempdb, RDSAdmin.`

Prosedur Tersimpan	Pesan Kesalahan	Masalah	Saran pemecahan masalah
rds_tlog_backup_copy_to_S3	Nama basis data untuk parameter input @db_name tidak boleh kosong atau null.	Nilai yang diberikan untuk parameter input `@db_name` kosong atau `NULL`.	Gunakan nama basis data yang benar. Untuk menampilkan daftar semua basis data berdasarkan nama, jalankan `SELECT * from sys.databases`
rds_tlog_backup_copy_to_S3	Periode retensi cadangan instans DB harus diatur ke setidaknya 1 untuk menjalankan prosedur tersimpan rds_tlog_backup_copy_setup.	Pencadangan otomatis tidak diaktifkan untuk instans DB.	Untuk informasi selengkapnya tentang mengaktifkan cadangan otomatis dan mengonfigurasi retensi cadangan, lihat Periode retensi cadangan.

Prosedur Tersimpan	Pesan Kesalahan	Masalah	Saran pemecahan masalah
rds_tlog_backup_copy_to_S3	Kesalahan saat menjalankan prosedur tersimpan rds_tlog_backup_copy_to_S3. Hubungkan kembali ke titik akhir RDS dan coba lagi.	Terjadi kesalahan internal.	Hubungkan kembali ke titik akhir RDS dan jalankan prosedur tersimpan `rds_tlog_backup_copy_to_S3` lagi.

Prosedur Tersimpan	Pesan Kesalahan	Masalah	Saran pemecahan masalah
rds_tlog_backup_copy_to_S3	Hanya satu dari tiga set parameter ini yang dapat disediakan. SET-1:(@backup_file_start_time, @backup_file_end_time) | SET-2:(@starting_lsn, @ending_lsn) | SET-3:(@rds_backup_starting_seq_id, @rds_backup_ending_seq_id)	Beberapa set parameter disediakan.	Anda dapat menentukan parameter waktu, lsn, atau ID urutan. Satu set dari tiga set parameter ini diperlukan. Untuk informasi selengkapnya tentang parameter yang diperlukan, lihat Menyalin cadangan log transaksi.

Prosedur Tersimpan	Pesan Kesalahan	Masalah	Saran pemecahan masalah
rds_tlog_backup_copy_to_S3	Menjalankan prosedur tersimpan rds_tlog_backup_copy_to_S3 di dalam transaksi tidak didukung. Verifikasi bahwa sesi tidak memiliki transaksi terbuka dan coba lagi.	Prosedur tersimpan dicoba dalam transaksi menggunakan `BEGIN` dan `END`.	Hindari menggunakan `BEGIN` dan `END` saat menjalankan prosedur tersimpan `rds_tlog_backup_copy_to_S3` .

Prosedur Tersimpan	Pesan Kesalahan	Masalah	Saran pemecahan masalah
rds_tlog_backup_copy_to_S3	Parameter yang disediakan berada di luar periode retensi log cadangan transaksi. Untuk menampilkan daftar file cadangan log transaksi yang tersedia, jalankan fungsi rds_fn_list_tlog_backup_metadata.	Tidak ada cadangan log transaksional yang tersedia untuk parameter input yang disediakan yang sesuai dalam periode retensi salinan.	Coba lagi dengan set parameter yang valid. Untuk informasi selengkapnya tentang parameter yang diperlukan, lihat Menyalin cadangan log transaksi.

Prosedur Tersimpan	Pesan Kesalahan	Masalah	Saran pemecahan masalah
rds_tlog_backup_copy_to_S3	Ada kesalahan izin dalam memproses permintaan. Pastikan bucket berada di Akun dan Wilayah yang sama dengan Instans DB, dan konfirmasikan izin kebijakan bucket S3 berdasarkan templat dalam dokumentasi publik.	Ada masalah yang terdeteksi pada bucket S3 yang disediakan atau izin kebijakannya.	Konfirmasikan pengaturan Anda untuk akses ke cadangan log transaksi sudah benar. Untuk informasi selengkapnya tentang persyaratan penyiapan bucket S3 Anda, lihat Persyaratan.

Prosedur Tersimpan	Pesan Kesalahan	Masalah	Saran pemecahan masalah
rds_tlog_backup_copy_to_S3	Menjalankan prosedur tersimpan `rds_tlog_backup_copy_to_S3` pada instans replika baca RDS tidak diizinkan.	Prosedur tersimpan dicoba pada instans replika baca RDS.	Hubungkan ke instans DB primer RDS untuk menjalankan prosedur tersimpan `rds_tlog_backup_copy_to_S3` .
rds_tlog_backup_copy_to_S3	LSN untuk parameter input `@starting_lsn` harus kurang dari `@ending_lsn` .	Nilai yang diberikan untuk parameter input `@starting_lsn` lebih besar dari nilai yang diberikan untuk parameter input `@ending_lsn` .	Pastikan nilai yang diberikan untuk parameter input `@starting_lsn` kurang dari nilai yang diberikan untuk parameter input `@ending_lsn` .

Prosedur Tersimpan	Pesan Kesalahan	Masalah	Saran pemecahan masalah
rds_tlog_backup_copy_to_S3	Prosedur tersimpan `rds_tlog_backup_copy_to_S3` hanya dapat dilakukan oleh anggota peran `db_owner` dalam basis data sumber.	Peran `db_owner` belum diberikan untuk akun yang mencoba menjalankan prosedur tersimpan `rds_tlog_backup_copy_to_S3` pada `db_name` yang disediakan.	Pastikan akun yang menjalankan prosedur tersimpan diizinkan dengan peran `db_owner` untuk `db_name` yang disediakan.
rds_tlog_backup_copy_to_S3	ID urutan untuk parameter input `@rds_backup_starting_seq_id` harus kurang dari atau sama dengan`@rds_up_ending_seq_id` .	Nilai yang diberikan untuk parameter input `@rds_backup_starting_seq_id` lebih besar dari nilai yang diberikan untuk parameter input `@rds_backup_ending_seq_id` .	Pastikan nilai yang diberikan untuk parameter input `@rds_backup_starting_seq_id` kurang dari nilai yang diberikan untuk parameter input `@rds_backup_ending_seq_id` .

Prosedur Tersimpan	Pesan Kesalahan	Masalah	Saran pemecahan masalah
rds_tlog_backup_copy_to_S3	Opsi SQLSERVER_BACKUP_RESTORE tidak diaktifkan atau sedang dalam proses diaktifkan. Aktifkan opsi atau coba lagi nanti.	Opsi SQLSERVER_BACKUP_RESTORE tidak diaktifkan pada instans DB atau baru saja diaktifkan dan menunggu aktivasi internal.	Aktifkan opsi SQLSERVER_BACKUP_RESTORE seperti yang ditentukan di bagian Persyaratan. Tunggu beberapa menit dan jalankan prosedur tersimpan rds_tlog_backup_copy_to_S3 lagi.
rds_tlog_backup_copy_to_S3	Waktu mulai untuk parameter input @backup_file_start_time harus kurang dari @backup_file_end_time.	Nilai yang diberikan untuk parameter input @backup_file_start_time lebih besar dari nilai yang diberikan untuk parameter input @backup_file_end_time.	Pastikan nilai yang diberikan untuk parameter input @backup_file_start_time kurang dari nilai yang diberikan untuk parameter input @backup_file_end_time.

Prosedur Tersimpan	Pesan Kesalahan	Masalah	Saran pemecahan masalah
rds_tlog_backup_copy_to_S3	Kami tidak dapat memproses permintaan karena tidak ada akses. Periksa pengaturan dan izin Anda untuk fitur tersebut.	Mungkin ada masalah dengan izin bucket Amazon S3, atau bucket Amazon S3 yang disediakan ada di akun atau Wilayah lain.	Pastikan izin kebijakan bucket Amazon S3 diizinkan untuk mengizinkan akses RDS. Pastikan bucket Amazon S3 berada di akun dan Wilayah yang sama dengan instans DB.
rds_tlog_backup_copy_to_S3	Anda tidak dapat memberikan ARN Kunci KMS sebagai parameter input ke prosedur tersimpan untuk instans yang tidak dienkripsi penyimpannya.	Ketika enkripsi penyimpanan tidak diaktifkan pada instans DB, parameter input `@kms_key_arn` tidak boleh disediakan.	Jangan berikan parameter input untuk `@kms_key_arn` .

Prosedur Tersimpan	Pesan Kesalahan	Masalah	Saran pemecahan masalah
rds_tlog_backup_copy_to_S3	Anda harus memberikan ARN Kunci KMS sebagai parameter input ke prosedur tersimpan untuk instans yang terenkripsi penyimpanannya.	Ketika enkripsi penyimpanan diaktifkan pada instans DB, parameter input `@kms_key_arn` harus disediakan.	Berikan parameter input untuk `@kms_key_arn` menggunakan nilai yang cocok dengan ARN bucket Amazon S3 yang akan digunakan untuk cadangan log transaksi.
rds_tlog_backup_copy_to_S3	Anda harus menjalankan prosedur tersimpan `rds_tlog_copy_setup` dan mengatur `@target_s3_arn`, sebelum menjalankan prosedur tersimpan `rds_tlog_backup_copy_to_S3`.	Akses ke prosedur penyiapan cadangan log transaksi tidak selesai sebelum mencoba menjalankan prosedur tersimpan `rds_tlog_backup_copy_to_S3`.	Jalankan prosedur tersimpan `rds_tlog_copy_setup` sebelum menjalankan prosedur tersimpan `rds_tlog_backup_copy_to_S3`. Untuk informasi selengkapnya tentang menjalankan prosedur penyiapan akses ke cadangan log transaksi, lihat Mengatur akses ke cadangan log transaksi.

Opsi untuk mesin basis data Microsoft SQL Server

Di bagian ini, Anda dapat menemukan deskripsi untuk opsi yang tersedia untuk instans Amazon RDS yang menjalankan mesin DB Microsoft SQL Server. Untuk mengaktifkan opsi ini, Anda dapat menambahkannya ke grup opsi, lalu mengaitkan grup opsi dengan instans DB Anda. Untuk informasi selengkapnya, lihat Menggunakan grup opsi.

Jika Anda mencari fitur opsional yang tidak ditambahkan melalui grup opsi RDS (seperti SSL, Microsoft Windows Authentication, dan integrasi Amazon S3), lihat Fitur tambahan untuk Microsoft SQL di Amazon RDS.

Amazon RDS mendukung opsi berikut untuk instans DB Microsoft SQL Server.

Opsi	ID Opsi	Edisi mesin
Server Tertaut dengan Oracle OLEDB	`OLEDB_ORACLE`	SQL Server Enterprise Edition SQL Server Standard Edition
Pencadangan dan pemulihan native	`SQLSERVER_BACKUP_RESTORE`	SQL Server Enterprise Edition SQL Server Standard Edition SQL Server Web Edition SQL Server Express Edition
Enkripsi Data Transparan	`TRANSPARENT_DATA_ENCRYPTION` (konsol RDS)	SQL Server 2014–2022 Enterprise Edition SQL Server 2022 Standard Edition

Opsi	ID Opsi	Edisi mesin
	TDE (AWS CLI dan API RDS)	
SQL Server Audit	SQLSERVER_AUDIT	Di RDS, mulai SQL Server 2014, semua edisi SQL Server mendukung audit tingkat server, dan Enterprise Edition juga mendukung audit tingkat basis data. Mulai SQL Server SQL Server 2016 (13.x) SP1, semua edisi mendukung audit tingkat server dan tingkat basis data. Untuk informasi selengkapnya, lihat SQL Server Audit (mesin basis data) di dokumentasi SQL Server.
SQL Server Analysis Services	SSAS	SQL Server Enterprise Edition SQL Server Standard Edition

Opsi	ID Opsi	Edisi mesin
SQL Server Integration Services	SSIS	SQL Server Enterprise Edition SQL Server Standard Edition
SQL Server Reporting Services	SSRS	SQL Server Enterprise Edition SQL Server Standard Edition
Microsoft Distributed Transaction Coordinator	MSDTC	Di RDS, mulai SQL Server 2014, semua edisi SQL Server mendukung transaksi terdistribusi.

Membuat daftar opsi yang tersedia untuk versi dan edisi SQL Server

Anda dapat menggunakan perintah `describe-option-group-options` AWS CLI untuk membuat daftar opsi yang tersedia untuk versi dan edisi SQL Server, dan pengaturan untuk opsi tersebut.

Contoh berikut menunjukkan opsi dan pengaturan opsi untuk SQL Server 2019 Enterprise Edition. Opsi `--engine-name` ini diperlukan.

```
aws rds describe-option-group-options --engine-name sqlserver-ee --major-engine-version
 15.00
```

Outputnya seperti berikut:

```
{
    "OptionGroupOptions": [
        {
            "Name": "MSDTC",
            "Description": "Microsoft Distributed Transaction Coordinator",
            "EngineName": "sqlserver-ee",
```

```
            "MajorEngineVersion": "15.00",
            "MinimumRequiredMinorEngineVersion": "4043.16.v1",
            "PortRequired": true,
            "DefaultPort": 5000,
            "OptionsDependedOn": [],
            "OptionsConflictsWith": [],
            "Persistent": false,
            "Permanent": false,
            "RequiresAutoMinorEngineVersionUpgrade": false,
            "VpcOnly": false,
            "OptionGroupOptionSettings": [
                {
                    "SettingName": "ENABLE_SNA_LU",
                    "SettingDescription": "Enable support for SNA LU protocol",
                    "DefaultValue": "true",
                    "ApplyType": "DYNAMIC",
                    "AllowedValues": "true,false",
                    "IsModifiable": true,
                    "IsRequired": false,
                    "MinimumEngineVersionPerAllowedValue": []
                },
        ...

        {
            "Name": "TDE",
            "Description": "SQL Server - Transparent Data Encryption",
            "EngineName": "sqlserver-ee",
            "MajorEngineVersion": "15.00",
            "MinimumRequiredMinorEngineVersion": "4043.16.v1",
            "PortRequired": false,
            "OptionsDependedOn": [],
            "OptionsConflictsWith": [],
            "Persistent": true,
            "Permanent": false,
            "RequiresAutoMinorEngineVersionUpgrade": false,
            "VpcOnly": false,
            "OptionGroupOptionSettings": []
        }
    ]
}
```

Dukungan untuk Server Tertaut dengan Oracle OLEDB di Amazon RDS for SQL Server

Server tertaut dengan Oracle Provider for OLEDB di RDS for SQL Server memungkinkan Anda mengakses sumber data eksternal di basis data Oracle. Anda dapat membaca data dari sumber data Oracle jarak jauh dan menjalankan perintah di server basis data Oracle jarak jauh di luar instans DB RDS for SQL Server Anda. Dengan menggunakan server tertaut dengan Oracle OLEDB, Anda dapat:

- Mengakses langsung sumber data selain SQL Server
- Mengueri beragam sumber data Oracle dengan kueri yang sama tanpa perlu memindahkan data
- Mengeluarkan kueri, pembaruan, perintah, dan transaksi terdistribusi pada sumber data di seluruh ekosistem perusahaan
- Mengintegrasikan koneksi ke basis data Oracle dari dalam Microsoft Business Intelligence Suite (SSIS, SSRS, SSAS)
- Memigrasi dari basis data Oracle ke RDS for SQL Server

Anda dapat mengaktifkan satu atau beberapa server tertaut untuk Oracle di RDS yang sudah ada atau baru untuk instans DB SQL Server. Kemudian, Anda dapat mengintegrasikan sumber data Oracle eksternal dengan instans DB.

Daftar Isi

Versi dan Wilayah yang didukung

RDS for SQL Server mendukung server tertaut dengan Oracle OLEDB di semua Wilayah untuk SQL Server Standard dan Enterprise Editions di versi berikut:

- SQL Server 2022, semua versi
- SQL Server 2019, semua versi
- SQL Server 2017, semua versi

Server tertaut dengan Oracle OLEDB didukung untuk versi Oracle Database berikut:

- Oracle Database 21c, semua versi
- Oracle Database 19c, semua versi
- Oracle Database 18c, semua versi

Batasan dan rekomendasi

Perlu diingat bahwa batasan dan rekomendasi berikut yang berlaku untuk server tertaut dengan Oracle OLEDB:

- Mengizinkan lalu lintas jaringan dengan menambahkan port TCP yang berlaku dalam grup keamanan untuk setiap instans DB RDS for SQL Server. Misalnya, jika Anda mengonfigurasi server tertaut antara instans DB EC2 Oracle dan instans DB RDS for SQL Server, Anda harus mengizinkan lalu lintas dari alamat IP instans DB EC2 Oracle. Anda juga harus mengizinkan lalu lintas pada port yang digunakan SQL Server untuk memproses komunikasi basis data. Untuk informasi selengkapnya mengenai Grup Keamanan, lihat Mengontrol akses dengan grup keamanan.
- Lakukan boot ulang pada instans DB RDS for SQL Server setelah mengaktifkan, menonaktifkan, atau memodifikasi opsi `OLEDB_ORACLE` di grup opsi Anda. Status grup opsi menampilkan `pending_reboot` untuk peristiwa ini dan diperlukan.
- Hanya autentikasi sederhana yang didukung dengan nama pengguna dan kata sandi untuk sumber data Oracle.
- Open Database Connectivity (ODBC) tidak didukung. Hanya driver OLEDB versi terbaru yang didukung.
- Transaksi terdistribusi (XA) didukung. Untuk mengaktifkan transaksi terdistribusi, aktifkan opsi `MSDTC` Grup Opsi untuk instans DB Anda dan pastikan transaksi XA diaktifkan. Untuk informasi

selengkapnya, lihat Dukungan untuk Microsoft Distributed Transaction Coordinator di RDS for SQL Server.

- Membuat nama sumber data (DSN) untuk digunakan sebagai pintasan string koneksi tidak didukung.
- Penelusuran driver OLEDB tidak didukung. Anda dapat menggunakan SQL Server Extended Events untuk melacak peristiwa OLEDB. Untuk informasi lebih lanjut, lihat Mengatur Peristiwa yang Diperluas di RDS for SQL Server.
- Akses ke folder katalog untuk server tertaut Oracle tidak didukung menggunakan SQL Server Management Studio (SSMS).

Mengaktifkan server tertaut menggunakan Oracle

Aktifkan server tertaut menggunakan Oracle dengan menambahkan opsi `OLEDB_ORACLE` ke instans DB RDS for SQL Server Anda. Gunakan proses berikut:

1. Buat grup opsi baru, atau pilih grup opsi yang sudah ada.
2. Tambahkan opsi `OLEDB_ORACLE` untuk grup opsi.
3. Pilih versi driver OLEDB yang akan digunakan.
4. Kaitkan grup opsi dengan instans DB.
5. Boot ulang instans DB.

Membuat grup opsi untuk OLEDB_ORACLE

Untuk dapat bekerja dengan server tertaut menggunakan Oracle, buat grup opsi atau ubah grup opsi yang sesuai dengan edisi SQL Server dan versi instans DB yang akan Anda gunakan. Untuk menyelesaikan prosedur ini, gunakan AWS Management Console atau AWS CLI.

Konsol

Prosedur berikut akan membuat grup opsi untuk SQL Server Standard Edition 2019.

Untuk membuat grup opsi

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup opsi.

3. Pilih Buat grup.
4. Di jendela Buat grup opsi, lakukan hal berikut:
   a. Untuk Nama, ketikkan nama untuk grup opsi yang unik dalam akun AWS Anda, seperti **`oracle-oledb-se-2019`**. Nama tersebut hanya boleh berisi huruf, angka, dan tanda hubung.
   b. Untuk Deskripsi, masukkan deskripsi singkat grup opsi, seperti **`OLEDB_ORACLE option group for SQL Server SE 2019`**. Deskripsi digunakan untuk tampilan.
   c. Untuk Mesin, pilih sqlserver-se.
   d. Untuk Versi mesin utama, pilih 15.00.
5. Pilih Buat.

CLI

Prosedur berikut akan membuat grup opsi untuk SQL Server Standard Edition 2019.

Untuk membuat grup opsi

- Gunakan salah satu perintah berikut ini.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds create-option-group \
    --option-group-name oracle-oledb-se-2019 \
    --engine-name sqlserver-se \
    --major-engine-version 15.00 \
    --option-group-description "OLEDB_ORACLE option group for SQL Server SE 2019"
```

  Untuk Windows:

```
aws rds create-option-group ^
    --option-group-name oracle-oledb-se-2019 ^
    --engine-name sqlserver-se ^
    --major-engine-version 15.00 ^
    --option-group-description "OLEDB_ORACLE option group for SQL Server SE 2019"
```

Menambahkan opsi **OLEDB_ORACLE** ke grup opsi

Selanjutnya, gunakan AWS Management Console atau AWS CLI untuk menambahkan opsi `OLEDB_ORACLE` ke grup opsi Anda.

Konsol

Untuk menambahkan opsi OLEDB_ORACLE

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup opsi.
3. Pilih grup opsi yang baru saja Anda buat, yaitu oracle-oledb-se-2019 dalam contoh ini.
4. Pilih Tambah opsi.
5. Di bagian Detail opsi, pilih OLEDB_ORACLE untuk Nama opsi.
6. Di bagian Penjadwalan, pilih apakah akan menambahkan opsi langsung atau pada masa pemeliharaan berikutnya.
7. Pilih Tambah opsi.

CLI

Untuk menambahkan opsi OLEDB_ORACLE

- Tambahkan opsi `OLEDB_ORACLE` untuk grup opsi.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds add-option-to-option-group \
    --option-group-name oracle-oledb-se-2019 \
    --options OptionName=OLEDB_ORACLE \
    --apply-immediately
```

  Untuk Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name oracle-oledb-se-2019 ^
    --options OptionName=OLEDB_ORACLE ^
```

```
    --apply-immediately
```

Mengaitkan grup opsi dengan instans DB

Untuk mengaitkan grup opsi `OLEDB_ORACLE` dan grup parameter dengan instans DB Anda, gunakan AWS Management Console atau AWS CLI

Konsol

Agar dapat menyelesaikan pengaktifan server tertaut untuk Oracle, kaitkan grup opsi `OLEDB_ORACLE` Anda dengan instans DB baru atau yang sudah ada:

- Untuk instans DB baru, kaitkan saat Anda meluncurkan instans. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.
- Untuk instans DB yang sudah ada, kaitkan dengan memodifikasi instans. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

CLI

Anda dapat mengaitkan grup opsi `OLEDB_ORACLE` dan grup parameter dengan instans DB baru atau yang sudah ada.

Untuk membuat instans dengan grup opsi **`OLEDB_ORACLE`** dan grup parameter

- Tentukan tipe mesin DB yang sama dan versi utama yang Anda gunakan saat membuat grup opsi.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mytestsqlserveroracleoledbinstance \
    --db-instance-class db.m5.2xlarge \
    --engine sqlserver-se \
    --engine-version 15.0.4236.7.v1 \
    --allocated-storage 100 \
    --manage-master-user-password \
    --master-username admin \
    --storage-type gp2 \
```

```
    --license-model li \
    --domain-iam-role-name my-directory-iam-role \
    --domain my-domain-id \
    --option-group-name oracle-oledb-se-2019 \
    --db-parameter-group-name my-parameter-group-name
```

Untuk Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mytestsqlserveroracleoledbinstance ^
    --db-instance-class db.m5.2xlarge ^
    --engine sqlserver-se ^
    --engine-version 15.0.4236.7.v1 ^
    --allocated-storage 100 ^
    --manage-master-user-password ^
    --master-username admin ^
    --storage-type gp2 ^
    --license-model li ^
    --domain-iam-role-name my-directory-iam-role ^
    --domain my-domain-id ^
    --option-group-name oracle-oledb-se-2019 ^
    --db-parameter-group-name my-parameter-group-name
```

Untuk memodifikasi instans dan mengaitkan grup opsi **`OLEDB_ORACLE`**

- Gunakan salah satu perintah berikut ini.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mytestsqlserveroracleoledbinstance \
    --option-group-name oracle-oledb-se-2019 \
    --db-parameter-group-name my-parameter-group-name \
    --apply-immediately
```

  Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mytestsqlserveroracleoledbinstance ^
```

```
    --option-group-name oracle-oledb-se-2019 ^
    --db-parameter-group-name my-parameter-group-name ^
    --apply-immediately
```

Memodifikasi properti penyedia OLEDB

Anda dapat melihat dan mengubah properti penyedia OLEDB. Hanya pengguna `master` yang dapat melakukan tugas ini. Semua server tertaut untuk Oracle yang dibuat di instans DB menggunakan properti yang sama dari penyedia OLEDB tersebut. Panggil prosedur tersimpan `sp_MSset_oledb_prop` untuk mengubah properti penyedia OLEDB.

Untuk mengubah properti penyedia OLEDB

```
USE [master]
GO
EXEC sp_MSset_oledb_prop N'OraOLEDB.Oracle', N'AllowInProcess', 1
EXEC sp_MSset_oledb_prop N'OraOLEDB.Oracle', N'DynamicParameters', 0
GO
```

Properti berikut dapat dimodifikasi:

Nama properti	Nilai yang Direkomendasikan (1 = Aktif, 0 = Mati)	Deskripsi
`Dynamic parameter`	1	Mengizinkan placeholder SQL (diwakili oleh '?') dalam kueri parameter.
`Nested queries`	1	Mengizinkan pernyataan `SELECT` bertingkat dalam klausa `FROM`, seperti subkueri.
`Level zero only`	0	Hanya antarmuka OLEDB tingkat dasar yang dipanggil untuk penyedia.
`Allow inprocess`	1	Jika diaktifkan, Microsoft SQL Server memungkin kan penyedia untuk dipakai sebagai server dalam proses. Atur properti ini ke 1 untuk menggunakan server tertaut Oracle.

Nama properti	Nilai yang Direkomendasikan (1 = Aktif, 0 = Mati)	Deskripsi
`Non transacted updates`	0	Jika bukan nol, SQL Server memungkinkan pembaruan.
`Index as access path`	Salah	Jika bukan nol, SQL Server mencoba menggunak an indeks penyedia untuk mengambil data.
`Disallow adhoc access`	Salah	Jika diatur, SQL Server tidak mengizinkan menjalankan kueri pass-through terhadap penyedia OLEDB. Meskipun opsi ini dapat dicentang, terkadang ini merupakan pilihan tepat untuk menjalankan kueri pass-through.
`Supports LIKE operator`	1	Menunjukkan bahwa penyedia mendukung kueri menggunakan kata kunci LIKE.

Memodifikasi properti driver OLEDB

Anda dapat melihat dan mengubah properti driver OLEDB saat membuat server tertaut untuk Oracle. Hanya pengguna `master` yang dapat melakukan tugas ini. Properti driver menentukan cara driver OLEDB menangani data saat menggunakan sumber data Oracle jarak jauh. Properti driver khusus untuk setiap server tertaut Oracle yang dibuat di instans DB. Panggil prosedur tersimpan `master.dbo.sp_addlinkedserver` untuk mengubah properti driver OLEDB.

Contoh: Untuk membuat server tertaut dan mengubah properti driver `FetchSize` OLEDB

```
EXEC master.dbo.sp_addlinkedserver
@server = N'Oracle_link2',
@srvproduct=N'Oracle',
@provider=N'OraOLEDB.Oracle',
@datasrc=N'my-oracle-test.cnetsipka.us-west-2.rds.amazonaws.com:1521/ORCL',
@provstr='FetchSize=200'
GO
```

```
EXEC master.dbo.sp_addlinkedsrvlogin
@rmtsrvname=N'Oracle_link2',
@useself=N'False',
@locallogin=NULL,
@rmtuser=N'master',
@rmtpassword='Test#1234'
GO
```

> **Note**
>
> Tetapkan kata sandi selain prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

Menonaktifkan server tertaut menggunakan Oracle

Untuk menonaktifkan server tertaut menggunakan Oracle, hapus opsi `OLEDB_ORACLE` dari grup opsi.

> **Important**
>
> Menghapus opsi tidak akan menghapus konfigurasi server tertaut yang sudah ada di instans DB. Anda harus melepaskannya secara manual untuk dapat menghapusnya dari instans DB. Anda dapat mengaktifkan kembali opsi `OLEDB_ORACLE` setelah dihapus agar bisa menggunakan kembali konfigurasi server tertaut yang sebelumnya dikonfigurasi di instans DB.

Konsol

Prosedur berikut akan menghapus opsi `OLEDB_ORACLE`.

Untuk menghapus opsi OLEDB_ORACLE dari grup opsi

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup opsi.
3. Pilih grup opsi dengan opsi `OLEDB_ORACLE` (`oracle-oledb-se-2019` dalam contoh sebelumnya).
4. Pilih Hapus opsi.
5. Di bagian Opsi penghapusan, pilih OLEDB_ORACLE untuk Opsi yang akan dihapus.

6. Di bagian Langsung terapkan, pilih Ya untuk segera menghapus opsi, atau Tidak untuk menghapusnya selama masa pemeliharaan berikutnya.
7. Pilih Hapus.

CLI

Prosedur berikut akan menghapus opsi `OLEDB_ORACLE`.

Untuk menghapus opsi OLEDB_ORACLE dari grup opsi

- Gunakan salah satu perintah berikut ini.

  Example

  Untuk Linux, macOS, atau Unix:

  ```
  aws rds remove-option-from-option-group \
      --option-group-name oracle-oledb-se-2019 \
      --options OLEDB_ORACLE \
      --apply-immediately
  ```

  Untuk Windows:

  ```
  aws rds remove-option-from-option-group ^
      --option-group-name oracle-oledb-se-2019 ^
      --options OLEDB_ORACLE ^
      --apply-immediately
  ```

Dukungan untuk pencadangan dan pemulihan native di SQL Server

Dengan menggunakan pencadangan dan pemulihan native untuk basis data SQL Server, Anda dapat membuat cadangan diferensial atau penuh dari basis data on-premise dan menyimpan file cadangan di Amazon S3. Kemudian Anda dapat memulihkan ke instans DB Amazon RDS yang sudah ada dan menjalankan SQL Server. Anda juga dapat mencadangkan basis data RDS for SQL Server, menyimpannya di Amazon S3, dan memulihkannya di lokasi lain. Selain itu, Anda juga dapat memulihkan cadangan ke server on-premise, atau instans DB Amazon RDS lain yang menjalankan SQL Server. Untuk informasi lebih lanjut, lihat Mengimpor dan mengekspor basis data SQL Server menggunakan pencadangan dan pemulihan native.

Amazon RDS mendukung pencadangan dan pemulihan native untuk basis data Microsoft SQL Server menggunakan file cadangan diferensial dan penuh (file .bak).

Menambahkan opsi pencadangan dan pemulihan native

Proses umum untuk menambahkan opsi pencadangan dan pemulihan native ke instans DB adalah sebagai berikut:

1. Buat grup opsi baru, atau salin atau ubah grup opsi yang sudah ada.
2. Tambahkan opsi `SQLSERVER_BACKUP_RESTORE` untuk grup opsi.
3. Mengaitkan peran AWS Identity and Access Management (IAM) dengan opsi. Peran IAM harus memiliki akses ke bucket S3 agar dapat menyimpan cadangan basis data.

   Artinya, opsi tersebut harus memiliki pengaturan Amazon Resource Name (ARN) yang valid dalam format `arn:aws:iam::`*`account-id`*`:role/`*`role-name`*. Untuk informasi selengkapnya, lihat Amazon Resource Name (ARN) di Referensi Umum AWS.

   Peran IAM juga harus memiliki hubungan kepercayaan dan kebijakan izin yang diberikan. Hubungan kepercayaan memungkinkan RDS untuk mengambil peran, dan kebijakan izin menentukan tindakan yang dapat dilakukan peran tersebut. Untuk informasi selengkapnya, lihat Membuat peran IAM secara manual untuk pencadangan dan pemulihan native.
4. Kaitkan grup opsi dengan instans DB.

Setelah menambahkan opsi pencadangan dan pemulihan native, Anda tidak perlu memulai ulang instans DB. Begitu grup opsi aktif, Anda dapat langsung mulai mencadangkan dan memulihkan.

Konsol

Untuk menambah opsi pencadangan dan pemulihan native

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup opsi.
3. Buat grup opsi baru atau menggunakan grup opsi yang sudah ada. Untuk informasi cara membuat grup opsi DB kustom, lihat Membuat grup opsi.

   Untuk menggunakan grup opsi yang sudah ada, lanjutkan ke langkah berikutnya.
4. Tambahkan opsi SQLSERVER_BACKUP_RESTORE ke grup opsi. Untuk informasi cara menambahkan aturan selengkapnya, lihat Menambahkan opsi ke grup opsi.
5. Lakukan salah satu dari cara berikut:

   - Untuk menggunakan peran IAM dan pengaturan Amazon S3 yang sudah ada, pilih peran IAM yang sudah ada untuk Peran IAM. Jika Anda menggunakan peran IAM yang sudah ada, RDS akan menggunakan pengaturan Amazon S3 yang dikonfigurasi untuk peran ini.
   - Untuk membuat peran baru dan mengonfigurasi pengaturan Amazon S3, lakukan hal berikut:
     1. Untuk Peran IAM, pilih Buat Peran Baru.
     2. Untuk Bucket S3, pilih bucket S3 dari daftar.
     3. Untuk awalan S3 (opsional), tentukan awalan yang akan digunakan untuk file yang disimpan di bucket Amazon S3.

        Awalan ini dapat menyertakan jalur file, tetapi tidak harus. Jika Anda memberikan awalan, RDS akan menambahkan awalan tersebut ke semua file cadangan. Kemudian RDS akan menggunakan awalan selama pemulihan untuk mengidentifikasi file terkait dan mengabaikan file yang tidak relevan. Misalnya, Anda mungkin menggunakan bucket S3 untuk tujuan selain dari menyimpan file cadangan. Dalam kasus ini, Anda dapat menggunakan awalan agar RDS melakukan pencadangan dan pemulihan native hanya pada folder tertentu beserta subfoldernya.

        Jika Anda membiarkan awalan kosong, RDS tidak akan menggunakan awalan untuk mengidentifikasi file cadangan atau file yang akan dipulihkan. Akibatnya, selama pemulihan banyak file, RDS akan mencoba memulihkan setiap file di setiap folder bucket S3.
     4. Pilih kotak centang Aktifkan enkripsi untuk mengenkripsi file cadangan. Biarkan kotak centang dihapus (default) agar file cadangan tidak terenkripsi.

Jika Anda memilih Aktifkan enkripsi, pilih kunci enkripsi untuk AWS KMS key. Untuk informasi kunci enkripsi selengkapnya, lihat Memulai di Panduan Developer.AWS Key Management Service

6. Pilih Tambah opsi.
7. Terapkan grup opsi ke instans DB baru atau yang sudah ada:

   - Untuk instans DB baru, terapkan grup opsi saat Anda meluncurkan instans. Untuk informasi lebih lanjut, lihat Membuat instans DB Amazon RDS.
   - Untuk instans DB yang sudah ada, terapkan grup opsi dengan memodifikasi instans dan menambahkan grup opsi baru. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

CLI

Prosedur ini akan membuat asumsi berikut:

- Anda menambahkan opsi SQLSERVER_BACKUP_RESTORE ke grup opsi yang sudah ada. Untuk informasi cara menambahkan aturan selengkapnya, lihat Menambahkan opsi ke grup opsi.
- Anda mengaitkan opsi dengan peran IAM yang sudah ada dan memiliki akses ke bucket S3 untuk menyimpan cadangan.
- Anda menerapkan grup opsi ke instans DB yang sudah ada. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Untuk menambah opsi pencadangan dan pemulihan native

1. Tambahkan opsi `SQLSERVER_BACKUP_RESTORE` ke grup opsi.

   Example

   Untuk Linux, macOS, atau Unix:

```
aws rds add-option-to-option-group \
 --apply-immediately \
 --option-group-name mybackupgroup \
 --options "OptionName=SQLSERVER_BACKUP_RESTORE, \
   OptionSettings=[{Name=IAM_ROLE_ARN,Value=arn:aws:iam::account-id:role/role-
name}]"
```

Untuk Windows:

```
aws rds add-option-to-option-group ^
 --option-group-name mybackupgroup ^
 --options "[{\"OptionName\": \"SQLSERVER_BACKUP_RESTORE\", ^
 \"OptionSettings\": [{\"Name\": \"IAM_ROLE_ARN\", ^
 \"Value\": \"arn:aws:iam::account-id:role/role-name"}]}]" ^
 --apply-immediately
```

> **Note**
>
> Saat menggunakan command prompt Windows, Anda harus meng-escape tanda kutip ganda (") dalam kode JSON dengan memberikan garis miring terbalik (\) di depannya.

2. Terapkan grup opsi ke instans DB.

   Example

   Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
 --db-instance-identifier mydbinstance \
 --option-group-name mybackupgroup \
 --apply-immediately
```

   Untuk Windows:

```
aws rds modify-db-instance ^
 --db-instance-identifier mydbinstance ^
 --option-group-name mybackupgroup ^
 --apply-immediately
```

Mengubah pengaturan opsi pencadangan dan pemulihan native

Setelah mengaktifkan opsi pencadangan dan pemulihan native, Anda dapat mengubah pengaturan untuk opsi tersebut. Untuk informasi cara mengubah pengaturan opsi selengkapnya, lihat Memodifikasi pengaturan opsi.

Menghapus opsi pencadangan dan pemulihan native

Anda dapat menonaktifkan pencadangan dan pemulihan native dengan menghapus opsi dari instans DB Anda. Setelah menghapus opsi pencadangan dan pemulihan native, Anda tidak perlu memulai ulang instans DB.

Untuk menghapus opsi pencadangan dan pemulihan native dari instans DB, lakukan salah satu langkah berikut:

- Hapus opsi dari grup opsi miliknya. Perubahan ini akan memengaruhi semua instans DB yang menggunakan grup opsi. Untuk informasi lebih lanjut, lihat Menghapus opsi dari grup opsi.
- Ubah instans DB lalu tentukan grup opsi lain yang tidak menyertakan opsi pencadangan dan pemulihan native. Perubahan ini akan memengaruhi satu instans DB. Anda dapat menentukan grup opsi default (kosong) atau grup opsi kustom lain. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Dukungan untuk Enkripsi Data Transparan di SQL Server

Amazon RDS mendukung penggunaan Enkripsi Data Transparan (TDE) untuk mengenkripsi data tersimpan di instans DB Anda yang menjalankan Microsoft SQL Server. TDE mengenkripsi data secara otomatis sebelum ditulis ke penyimpanan, dan mendekripsi data secara otomatis saat data dibaca dari penyimpanan.

Amazon RDS mendukung TDE untuk versi dan edisi SQL Server berikut:

- SQL Server 2022 Standard dan Enterprise Edition
- SQL Server 2019 Standard dan Enterprise Edition
- SQL Server 2017 Enterprise Edition
- SQL Server 2016 Enterprise Edition
- SQL Server 2014 Enterprise Edition

Enkripsi Data Transparan untuk SQL Server menyediakan manajemen kunci enkripsi dengan menggunakan arsitektur kunci dua tingkat. Sertifikat, yang dihasilkan dari kunci utama basis data, digunakan untuk melindungi kunci enkripsi data. Kunci enkripsi basis data melakukan enkripsi dan dekripsi data yang sebenarnya pada basis data pengguna. Amazon RDS mencadangkan dan mengelola kunci utama basis data dan sertifikat TDE.

Enkripsi Data transparan digunakan dalam skenario ketika Anda perlu mengenkripsi data sensitif. Misalnya, saat Anda harus memberikan file dan cadangan data kepada pihak ketiga, atau mengatasi masalah kepatuhan peraturan terkait keamanan. Anda tidak dapat mengenkripsi basis data sistem untuk SQL Server, seperti basis data `model` atau `master`.

Pembahasan mendetail tentang Enkripsi Data Transparan berada di luar cakupan panduan ini, tetapi pastikan bahwa Anda memahami kekuatan dan kelemahan keamanan dari setiap algoritma dan kunci enkripsi. Untuk informasi tentang Enkripsi Data Transparan untuk SQL Server, lihat Enkripsi Data Transparan (TDE) di dokumentasi Microsoft.

Topik

- Mengaktifkan TDE untuk RDS for SQL Server
- Mengenkripsi data di RDS for SQL Server
- Mencadangkan dan memulihkan sertifikat TDE pada RDS for SQL Server
- Mencadangkan dan memulihkan sertifikat TDE untuk basis data on-premise

- Menonaktifkan TDE untuk RDS for SQL Server

Mengaktifkan TDE untuk RDS for SQL Server

Untuk mengaktifkan Enkripsi Data Transparan untuk instans DB RDS for SQL Server, tentukan opsi TDE dalam grup opsi RDS yang dikaitkan dengan instans DB tersebut:

1. Tentukan apakah instans DB Anda sudah dikaitkan dengan grup opsi yang memiliki opsi TDE. Untuk melihat grup opsi yang terkait dengan instans DB, gunakan konsol RDS, describe-db-instanceAWS CLIperintah, atau operasi API DescribedBInstances.
2. Jika instans DB tidak dikaitkan dengan grup opsi yang TDE-nya diaktifkan, Anda memiliki dua pilihan. Anda dapat membuat grup opsi dan menambahkan opsi TDE, atau menyesuaikan grup opsi yang sudah dikaitkan untuk menambahkannya.

   > **Note**
   >
   > Dalam konsol RDS, opsi ini bernama `TRANSPARENT_DATA_ENCRYPTION`. Di AWS CLI dan API RDS, namanya adalah `TDE`.

   Untuk informasi tentang membuat atau menyesuaikan grup opsi, lihat Menggunakan grup opsi. Untuk informasi tentang menambahkan opsi ke grup opsi, lihat Menambahkan opsi ke grup opsi.
3. Kaitkan instans DB dengan grup opsi yang memiliki opsi TDE. Untuk informasi tentang mengaitkan instans DB dengan grup opsi, lihat Memodifikasi instans DB Amazon RDS.

Pertimbangan grup opsi

Opsi TDE adalah opsi yang persisten. Anda tidak dapat menghapusnya dari grup opsi kecuali semua instans DB dan cadangan tidak lagi terkait dengan grup opsi. Setelah Anda menambahkan opsi TDE ke grup opsi, grup opsi hanya dapat dikaitkan dengan instans DB yang menggunakan TDE. Untuk informasi selengkapnya tentang opsi yang persisten dalam grup opsi, lihat Gambaran umum grup opsi.

Karena opsi TDE adalah opsi yang persisten, Anda dapat mengalami konflik antara grup opsi dan instans DB yang terkait. Anda dapat mengalami konflik dalam situasi berikut:

- Grup opsi saat ini memiliki opsi TDE, dan Anda menggantinya dengan grup opsi yang tidak memiliki opsi TDE.

- Anda memulihkan dari snapshot DB ke instans DB baru yang tidak memiliki grup opsi yang berisi opsi TDE. Untuk informasi selengkapnya tentang skenario ini, lihat Pertimbangan grup opsi.

Pertimbangan performa SQL Server

Penggunaan Enkripsi Data Transparan dapat memengaruhi performa instans DB SQL Server.

Performa basis data yang tidak terenkripsi juga dapat terdegradasi jika basis data tersebut berada di instans DB yang memiliki setidaknya satu basis data terenkripsi. Oleh karena itu, sebaiknya Anda menyimpan basis data terenkripsi dan tidak terenkripsi di instans DB terpisah.

Mengenkripsi data di RDS for SQL Server

Jika opsi TDE ditambahkan ke grup opsi, Amazon RDS menghasilkan sertifikat yang digunakan dalam proses enkripsi. Anda kemudian dapat menggunakan sertifikat tersebut untuk menjalankan pernyataan SQL yang mengenkripsi data dalam basis data pada instans DB.

Contoh berikut menggunakan sertifikat yang dihasilkan RDS, bernama `RDSTDECertificateName`, untuk mengenkripsi basis data yang disebut `myDatabase`.

```
---------- Turning on TDE -------------

-- Find an RDS TDE certificate to use
USE [master]
GO
SELECT name FROM sys.certificates WHERE name LIKE 'RDSTDECertificate%'
GO

USE [myDatabase]
GO
-- Create a database encryption key (DEK) using one of the certificates from the
 previous step
CREATE DATABASE ENCRYPTION KEY WITH ALGORITHM = AES_256
ENCRYPTION BY SERVER CERTIFICATE [RDSTDECertificateName]
GO

-- Turn on encryption for the database
ALTER DATABASE [myDatabase] SET ENCRYPTION ON
GO

-- Verify that the database is encrypted
```

```
USE [master]
GO
SELECT name FROM sys.databases WHERE is_encrypted = 1
GO
SELECT db_name(database_id) as DatabaseName, * FROM sys.dm_database_encryption_keys
GO
```

Waktu yang diperlukan untuk mengenkripsi basis data SQL Server menggunakan TDE bergantung pada beberapa faktor. Faktor tersebut mencakup ukuran instans DB, apakah instans menggunakan penyimpanan IOPS yang Tersedia, jumlah data, dan faktor lainnya.

Mencadangkan dan memulihkan sertifikat TDE pada RDS for SQL Server

RDS for SQL Server menyediakan prosedur tersimpan untuk mencadangkan, memulihkan, dan menghapus sertifikat TDE. RDS for SQL Server juga menyediakan fungsi untuk melihat sertifikat TDE pengguna yang dipulihkan.

Sertifikat TDE pengguna digunakan untuk memulihkan basis data ke RDS for SQL Server yang on-premise dan TDE-nya diaktifkan. Sertifikat ini memiliki awalan `UserTDECertificate_`. Setelah memulihkan basis data, dan sebelum membuatnya tersedia untuk digunakan, RDS menyesuaikan basis data yang TDE-nya diaktifkan untuk menggunakan sertifikat TDE yang dihasilkan RDS. Sertifikat ini memiliki awalan `RDSTDECertificate`.

Sertifikat TDE pengguna tetap berada di instans DB RDS for SQL Server, kecuali jika Anda menghapusnya menggunakan prosedur tersimpan `rds_drop_tde_certificate`. Untuk informasi selengkapnya, lihat Menghapus sertifikat TDE yang dipulihkan.

Anda dapat menggunakan sertifikat TDE pengguna untuk memulihkan basis data lain dari instans DB sumber. Basis data untuk memulihkan harus menggunakan sertifikat TDE yang sama dan TDE-nya diaktifkan. Anda tidak perlu mengimpor (memulihkan) sertifikat yang sama lagi.

Topik

- Prasyarat
- Batasan
- Mencadangkan sertifikat TDE
- Memulihkan sertifikat TDE
- Melihat sertifikat TDE yang dipulihkan
- Menghapus sertifikat TDE yang dipulihkan

Prasyarat

Sebelum Anda dapat mencadangkan atau memulihkan sertifikat TDE pada RDS for SQL Server, pastikan untuk melakukan tugas-tugas berikut. Tiga yang pertama dijelaskan dalam Menyiapkan pencadangan dan pemulihan native.

1. Buat bucket Amazon S3 untuk menyimpan file yang akan dicadangkan dan dipulihkan.

   Sebaiknya Anda menggunakan bucket terpisah untuk pencadangan basis data dan untuk pencadangan sertifikat TDE.
2. Buat peran IAM untuk mencadangkan dan memulihkan file.

   Peran IAM harus berupa pengguna dan administrator untuk AWS KMS key.

   Selain izin yang diperlukan untuk pencadangan dan pemulihan asli SQL Server, peran IAM juga memerlukan izin berikut:

   - `s3:GetBucketACL`, `s3:GetBucketLocation`, dan `s3:ListBucket` pada sumber daya bucket S3
   - `s3:ListAllMyBuckets` di sumber daya `*`
3. Tambahkan opsi `SQLSERVER_BACKUP_RESTORE` ke grup opsi di instans DB Anda.

   Ini merupakan tambahan dari opsi `TRANSPARENT_DATA_ENCRYPTION` (`TDE`).
4. Pastikan Anda memiliki kunci KMS enkripsi simetris. Anda memiliki opsi berikut:
   - Anda dapat menggunakan kunci KMS yang sudah ada di akun Anda. Tidak ada tindakan lebih lanjut yang diperlukan.
   - Jika Anda tidak memiliki kunci KMS enkripsi simetris yang ada di akun Anda, buat kunci KMS dengan mengikuti petunjuk di Membuat kunci di Panduan Developer AWS Key Management Service.
5. Aktifkan integrasi Amazon S3 untuk mentransfer file antara instans DB dan Amazon S3.

   Untuk informasi selengkapnya tentang mengaktifkan integrasi Amazon S3, lihat. Mengintegrasikan instans DB Amazon RDS for SQL Server dengan Amazon S3

Batasan

Penggunaan prosedur tersimpan untuk mencadangkan dan memulihkan sertifikat TDE memiliki batasan sebagai berikut:

- Opsi SQLSERVER_BACKUP_RESTORE dan TRANSPARENT_DATA_ENCRYPTION (TDE) harus ditambahkan ke grup opsi yang Anda kaitkan dengan instans DB.
- Pencadangan dan pemulihan sertifikat TDE tidak didukung pada instans DB Multi-AZ.
- Membatalkan pencadangan dan pemulihan sertifikat TDE tidak didukung.
- Anda tidak dapat menggunakan sertifikat TDE pengguna untuk enkripsi TDE basis data lain di instans DB RDS for SQL Server Anda. Anda dapat menggunakannya hanya untuk memulihkan basis data lain dari instans DB sumber yang TDE-nya diaktifkan dan yang menggunakan sertifikat TDE yang sama.
- Anda hanya dapat menghapus sertifikat TDE pengguna.
- Jumlah maksimum sertifikat TDE pengguna yang didukung pada RDS adalah 10. Jika jumlahnya melebihi 10, hapus sertifikat TDE yang tidak digunakan dan coba lagi.
- Nama sertifikat harus diisi dan tidak boleh kosong.
- Saat memulihkan sertifikat, nama sertifikat tidak dapat menyertakan kata kunci RDSTDECERTIFICATE, dan harus dimulai dengan awalan UserTDECertificate_.
- Parameter @certificate_name hanya dapat menyertakan karakter berikut: a-z, 0-9, @, $, #, dan garis bawah (_).
- Ekstensi file untuk @certificate_file_s3_arn harus .cer (peka huruf besar kecil).
- Ekstensi file untuk @private_key_file_s3_arn harus .pvk (peka huruf besar kecil).
- Metadata S3 untuk file kunci privat harus menyertakan tag x-amz-meta-rds-tde-pwd. Untuk informasi selengkapnya, lihat Mencadangkan dan memulihkan sertifikat TDE untuk basis data on-premise.

Mencadangkan sertifikat TDE

Untuk mencadangkan sertifikat TDE, gunakan prosedur tersimpan rds_backup_tde_certificate. Ini memiliki sintaks berikut.

```
EXECUTE msdb.dbo.rds_backup_tde_certificate
    @certificate_name='UserTDECertificate_certificate_name |
 RDSTDECertificatetimestamp',
    @certificate_file_s3_arn='arn:aws:s3:::bucket_name/certificate_file_name.cer',
    @private_key_file_s3_arn='arn:aws:s3:::bucket_name/key_file_name.pvk',
    @kms_password_key_arn='arn:aws:kms:region:account-id:key/key-id',
    [@overwrite_s3_files=0|1];
```

Parameter berikut diperlukan:

- `@certificate_name` – Nama sertifikat TDE yang akan dicadangkan.
- `@certificate_file_s3_arn` – Amazon Resource Name (ARN) tujuan untuk file cadangan sertifikat di Amazon S3.
- `@private_key_file_s3_arn` – S3 ARN tujuan file kunci privat yang mengamankan sertifikat TDE.
- `@kms_password_key_arn` – ARN kunci KMS simetris yang digunakan untuk mengenkripsi kata sandi kunci privat.

Parameter berikut bersifat opsional:

- `@overwrite_s3_files` – Menunjukkan apakah akan menimpa file kunci privat dan sertifikat yang ada di S3:
  - `0` – Tidak menimpa file yang ada. Nilai ini adalah default.

    Mengatur `@overwrite_s3_files` ke 0 akan menghasilkan kesalahan jika file sudah ada.
  - `1` – Menimpa file yang sudah ada dengan nama yang ditentukan, meskipun itu bukan file cadangan.

Example pencadangan sertifikat TDE

```
EXECUTE msdb.dbo.rds_backup_tde_certificate
    @certificate_name='RDSTDECertificate20211115T185333',
    @certificate_file_s3_arn='arn:aws:s3:::TDE_certs/mycertfile.cer',
    @private_key_file_s3_arn='arn:aws:s3:::TDE_certs/mykeyfile.pvk',
    @kms_password_key_arn='arn:aws:kms:us-
west-2:123456789012:key/AKIAIOSFODNN7EXAMPLE',
    @overwrite_s3_files=1;
```

Memulihkan sertifikat TDE

Anda dapat menggunakan prosedur tersimpan `rds_restore_tde_certificate` untuk memulihkan (mengimpor) sertifikat TDE pengguna. Ini memiliki sintaks berikut.

```
EXECUTE msdb.dbo.rds_restore_tde_certificate
    @certificate_name='UserTDECertificate_certificate_name',
    @certificate_file_s3_arn='arn:aws:s3:::bucket_name/certificate_file_name.cer',
```

```
    @private_key_file_s3_arn='arn:aws:s3:::bucket_name/key_file_name.pvk',
    @kms_password_key_arn='arn:aws:kms:region:account-id:key/key-id';
```

Parameter berikut diperlukan:

- `@certificate_name` – Nama sertifikat TDE yang akan dipulihkan. Nama harus dimulai dengan awalan `UserTDECertificate_`.
- `@certificate_file_s3_arn` – S3 ARN file cadangan yang digunakan untuk memulihkan sertifikat TDE.
- `@private_key_file_s3_arn` – S3 ARN file cadangan kunci privat untuk sertifikat TDE yang akan dipulihkan.
- `@kms_password_key_arn` – ARN kunci KMS simetris yang digunakan untuk mengenkripsi kata sandi kunci privat.

Example pemulihan sertifikat TDE

```
EXECUTE msdb.dbo.rds_restore_tde_certificate
    @certificate_name='UserTDECertificate_myTDEcertificate',
    @certificate_file_s3_arn='arn:aws:s3:::TDE_certs/mycertfile.cer',
    @private_key_file_s3_arn='arn:aws:s3:::TDE_certs/mykeyfile.pvk',
    @kms_password_key_arn='arn:aws:kms:us-
west-2:123456789012:key/AKIAIOSFODNN7EXAMPLE';
```

Melihat sertifikat TDE yang dipulihkan

Anda dapat menggunakan fungsi `rds_fn_list_user_tde_certificates` untuk melihat sertifikat TDE pengguna yang dipulihkan (diimpor). Ini memiliki sintaks berikut.

```
SELECT * FROM msdb.dbo.rds_fn_list_user_tde_certificates();
```

Outputnya seperti berikut. Tidak semua kolom ditampilkan di sini.

```
name   certif princi pvt_ke issuer cert_s thumbp subjec start_ expiry pvt_key_l
       te_id  _id    ncrypt me     al_num t      t      e      te     ast_backu
                     _type_                                          p_date
                     c
```

UserTDrtific_tde_c	343	1	ENCRYP_BY_MAR_KEY	AnyCopy Shippi	793e57a369fd1d9e472c32671d9ccaaf	0x6BB2341103 80B FE1BA2C69509 5B5	AnyCopy Shippi	2022-05 19:49:000000	2023-05 19:49:000000	NULL

Menghapus sertifikat TDE yang dipulihkan

Untuk menghapus sertifikat TDE pengguna yang dipulihkan (diimpor) yang tidak Anda gunakan, gunakan prosedur tersimpan `rds_drop_tde_certificate`. Ini memiliki sintaks berikut.

```
EXECUTE msdb.dbo.rds_drop_tde_certificate
 @certificate_name='UserTDECertificate_certificate_name';
```

Parameter berikut diperlukan:

- `@certificate_name` – Nama sertifikat TDE yang akan dihapus.

Anda hanya dapat menghapus sertifikat TDE yang dipulihkan (diimpor). Anda tidak dapat menghapus sertifikat yang dihasilkan RDS.

Example penghapusan sertifikat TDE

```
EXECUTE msdb.dbo.rds_drop_tde_certificate
 @certificate_name='UserTDECertificate_myTDEcertificate';
```

Mencadangkan dan memulihkan sertifikat TDE untuk basis data on-premise

Anda dapat mencadangkan sertifikat TDE untuk basis data on-premise, lalu memulihkannya ke RDS for SQL Server. Anda juga dapat memulihkan sertifikat TDE RDS for SQL Server ke instans DB on-premise.

Prosedur berikut mencadangkan sertifikat TDE dan kunci privat. Kunci privat dienkripsi menggunakan kunci data yang dihasilkan dari kunci KMS enkripsi simetris Anda.

Untuk mencadangkan sertifikat TDE on-premise

1. Hasilkan kunci data menggunakan AWS CLI generate-data-keyperintah.

   ```
   aws kms generate-data-key \
       --key-id my_KMS_key_ID \
       --key-spec AES_256
   ```

   Outputnya seperti berikut.

   ```
   {
   "CiphertextBlob": "AQIDAHimL2NEoAlOY6Bn7LJfnxi/OZe9kTQo/
   XQXduug1rmerwGiL7g5ux4av9GfZLxYTDATAAAAfjB8BgkqhkiG9w0B
   BwagbzBtAgEAMGgGCSqGSIb3DQEHATAeBglghkgBZQMEAS4wEQQMyCxLMi7GRZgKqD65AgEQgDtjvZLJo2cQ31Vetng
   2RezQy3sAS6ZHrCjfnfn0c65bFdhsXxjSMnudIY7AKw==",
   "Plaintext": "U/fpGtmzGCYBi8A2+0/9qcRQRK2zmG/aOn939ZnKi/0=",
   "KeyId": "arn:aws:kms:us-west-2:123456789012:key/1234abcd-00ee-99ff-88dd-
   aa11bb22cc33"
   }
   ```

   Anda dapat menggunakan output teks biasa pada langkah berikutnya sebagai kata sandi kunci privat.

2. Cadangkan sertifikat TDE Anda seperti yang ditunjukkan dalam contoh berikut.

   ```
   BACKUP CERTIFICATE myOnPremTDEcertificate TO FILE = 'D:\tde-cert-backup.cer'
   WITH PRIVATE KEY (
   FILE = 'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\cert-
   backup-key.pvk',
   ENCRYPTION BY PASSWORD = 'U/fpGtmzGCYBi8A2+0/9qcRQRK2zmG/aOn939ZnKi/0=');
   ```

3. Simpan file cadangan sertifikat ke bucket sertifikat Amazon S3.

4. Simpan file cadangan kunci privat ke bucket sertifikat S3 Anda, dengan tag berikut di metadata file:

   - Kunci – `x-amz-meta-rds-tde-pwd`
   - Nilai – Nilai `CiphertextBlob` dari menghasilkan kunci data, seperti pada contoh berikut.

   ```
   AQIDAHimL2NEoAlOY6Bn7LJfnxi/OZe9kTQo/
   XQXduug1rmerwGiL7g5ux4av9GfZLxYTDATAAAAfjB8BgkqhkiG9w0B
   BwagbzBtAgEAMGgGCSqGSIb3DQEHATAeBglghkgBZQMEAS4wEQQMyCxLMi7GRZgKqD65AgEQgDtjvZLJo2cQ31Vet
   2RezQy3sAS6ZHrCjfnfn0c65bFdhsXxjSMnudIY7AKw==
   ```

Prosedur berikut memulihkan sertifikat TDE RDS for SQL Server ke instans DB on-premise. Anda menyalin dan memulihkan sertifikat TDE pada instans DB tujuan menggunakan cadangan sertifikat, file kunci privat yang sesuai, dan kunci data. Sertifikat yang dipulihkan dienkripsi oleh kunci utama basis data server baru.

Untuk memulihkan sertifikat TDE

1. Salin file cadangan sertifikat TDE dan file kunci privat dari Amazon S3 ke instans tujuan. Untuk informasi selengkapnya tentang menyalin file dari Amazon S3, lihat Mentransfer file antara RDS for SQL Server dan Amazon S3.
2. Gunakan kunci KMS Anda untuk mendekripsi teks sandi output guna mengambil teks biasa dari kunci data. Teks sandi berada di metadata S3 file cadangan kunci privat.

   ```
   aws kms decrypt \
       --key-id my_KMS_key_ID \
       --ciphertext-blob fileb://exampleCiphertextFile | base64 -d \
       --output text \
       --query Plaintext
   ```

   Anda dapat menggunakan output teks biasa pada langkah berikutnya sebagai kata sandi kunci privat.
3. Gunakan perintah SQL berikut untuk memulihkan sertifikat TDE Anda.

   ```
   CREATE CERTIFICATE myOnPremTDEcertificate FROM FILE='D:\tde-cert-backup.cer'
   WITH PRIVATE KEY (FILE = N'D:\tde-cert-key.pvk',
   DECRYPTION BY PASSWORD = 'plain_text_output');
   ```

Untuk informasi lebih lanjut tentang dekripsi KMS, lihat mendekripsi di bagian KMS untuk Referensi Perintah AWS CLI.

Setelah sertifikat TDE dipulihkan pada instans DB tujuan, Anda dapat memulihkan basis data yang terenkripsi dengan sertifikat tersebut.

> **Note**
>
> Anda dapat menggunakan sertifikat TDE yang sama untuk mengenkripsi beberapa basis data SQL Server pada instans DB sumber. Untuk memigrasikan beberapa basis data ke instans tujuan, salin sertifikat TDE yang terkait dengan basis data tersebut ke instans tujuan cukup satu kali.

Menonaktifkan TDE untuk RDS for SQL Server

Untuk menonaktifkan TDE untuk instans DB RDS for SQL Server, pertama-tama pastikan bahwa tidak ada objek terenkripsi yang tersisa di instans DB. Untuk melakukannya, dekripsi atau hapus objek. Jika ada objek terenkripsi di instans DB, Anda tidak dapat menonaktifkan TDE untuk instans DB. Saat Anda menggunakan konsol untuk menghapus opsi TDE dari grup opsi, konsol menunjukkan bahwa TDE sedang dalam proses. Selain itu, peristiwa kesalahan terjadi jika grup opsi dikaitkan dengan instans DB atau snapshot DB terenkripsi.

Contoh berikut menghapus enkripsi TDE dari basis data yang bernama `customerDatabase`.

```
------------- Removing TDE ---------------

USE [customerDatabase]
GO

-- Turn off encryption of the database
ALTER DATABASE [customerDatabase]
SET ENCRYPTION OFF
GO

-- Wait until the encryption state of the database becomes 1. The state is 5
 (Decryption in progress) for a while
SELECT db_name(database_id) as DatabaseName, * FROM sys.dm_database_encryption_keys
GO

-- Drop the DEK used for encryption
```

```
DROP DATABASE ENCRYPTION KEY
GO

-- Alter to SIMPLE Recovery mode so that your encrypted log gets truncated
USE [master]
GO
ALTER DATABASE [customerDatabase] SET RECOVERY SIMPLE
GO
```

Saat semua objek didekripsi, Anda memiliki dua opsi:

1. Anda dapat menyesuaikan instans DB untuk dikaitkan dengan grup opsi tanpa opsi TDE.
2. Anda dapat menghapus opsi TDE dari grup opsi.

SQL Server Audit

Di Amazon RDS, Anda dapat mengaudit basis data Microsoft SQL Server menggunakan mekanisme SQL Server Audit bawaan. Anda dapat membuat audit dan spesifikasi audit dengan cara yang sama seperti Anda membuatnya untuk server basis data on-premise.

RDS mengunggah log audit yang telah selesai ke bucket S3 menggunakan peran IAM yang Anda berikan. Jika Anda mengaktifkan retensi, RDS akan menyimpan log audit di instans DB selama periode waktu yang dikonfigurasi.

Untuk informasi selengkapnya, lihat SQL Server Audit (database engine) dalam dokumentasi Microsoft SQL Server.

SQL Server Audit dengan Aliran Aktivitas Basis Data (DAS)

Anda dapat menggunakan Aliran Aktivitas Basis Data untuk RDS agar dapat mengintegrasikan peristiwa SQL Server Audit dengan alat pemantauan aktivitas basis data dari Imperva, McAfee, dan IBM. Untuk informasi cara mengaudit dengan Aliran Aktivitas Basis Data untuk RDS SQL Server selengkapnya, lihat Pengauditan di Microsoft SQL Server

Topik

- Dukungan untuk SQL Server Audit
- Menambahkan SQL Server Audit ke opsi instans DB
- Menggunakan SQL Server Audit
- Melihat log audit
- Menggunakan SQL Server Audit dengan instans Multi-AZ
- Mengonfigurasi bucket S3
- Membuat peran IAM secara manual untuk SQL Server Audit

Dukungan untuk SQL Server Audit

Di Amazon RDS, dimulai dengan SQL Server 2014, semua edisi SQL Server mendukung audit tingkat server, dan edisi Enterprise juga mendukung audit tingkat basis data. Dimulai dengan SQL Server 2016 (13.x) SP1, semua edisi mendukung audit tingkat server dan tingkat basis data. Untuk informasi selengkapnya, lihat SQL Server Audit (database engine) dalam dokumentasi SQL Server.

RDS mendukung konfigurasi pengaturan opsi berikut untuk SQL Server Audit.

Pengaturan opsi	Nilai valid	Deskripsi
`IAM_ROLE_ARN`	Amazon Resource Name (ARN) yang valid dalam format `arn:aws:iam:: account-id :role/role-name` .	ARN dari peran IAM yang memberikan akses ke bucket S3 tempat Anda ingin menyimpan log audit. Untuk informasi selengkapnya, lihat Amazon Resource Name (ARN) di Referensi Umum AWS.
`S3_BUCKET_ARN`	ARN yang valid dalam format `arn:aws:s3:::bucket-name` atau `arn:aws:s3:::bucket-name/key-prefix`	ARN untuk bucket S3 tempat Anda ingin menyimpan log audit.
`ENABLE_COMPRESSION`	`true` atau `false`	Mengontrol kompresi log audit. Secara default, kompresi diaktifkan (ditetapkan ke `true`).
`RETENTION_TIME`	`0` untuk `840`	Waktu retensi (dalam jam) catatan SQL Server Audit disimpan di instans RDS Anda. Secara default, retensi dinonaktifkan.

RDS mendukung SQL Server Audit di semua Wilayah AWS kecuali Timur Tengah (Bahrain).

Menambahkan SQL Server Audit ke opsi instans DB

Mengaktifkan SQL Server Audit memerlukan dua langkah: mengaktifkan opsi di instans DB, dan mengaktifkan fitur di dalam SQL Server. Proses untuk menambahkan opsi SQL Server Audit ke instans DB adalah sebagai berikut:

1. Buat grup opsi baru, atau salin atau ubah grup opsi yang sudah ada.

2. Tambahkan dan konfigurasikan semua opsi yang diperlukan.
3. Kaitkan grup opsi dengan instans DB.

Setelah menambahkan opsi SQL Server Audit, Anda tidak perlu memulai ulang instans DB. Begitu grup opsi aktif, Anda dapat membuat audit dan menyimpan log audit di bucket S3.

Untuk menambah dan mengonfigurasi SQL Server Audit pada grup opsi instans DB

1. Pilih salah satu cara berikut:
   - Gunakan grup opsi yang sudah ada.
   - Buat grup opsi DB kustom dan gunakan grup opsi tersebut. Untuk informasi selengkapnya, lihat Membuat grup opsi.
2. Tambahkan opsi SQLSERVER_AUDIT ke grup opsi, lalu konfigurasikan pengaturan opsi. Untuk informasi cara menambahkan aturan selengkapnya, lihat Menambahkan opsi ke grup opsi.
   - Untuk peran IAM, Anda dapat memilih peran tersebut jika sudah memiliki peran IAM dengan kebijakan yang diperlukan. Untuk membuat peran IAM baru, pilih Buat Peran Baru. Untuk informasi tentang kebijakan yang diperlukan, lihat Membuat peran IAM secara manual untuk SQL Server Audit.
   - Untuk Pilih tujuan S3, pilih opsi ini jika Anda sudah memiliki bucket S3 yang ingin Anda gunakan. Untuk membuat bucket S3, pilih Buat Bucket S3 Baru.
   - Untuk Mengaktifkan Kompresi, biarkan opsi ini dipilih untuk mengompresi file audit. Kompresi diaktifkan secara default. Untuk menonaktifkan kompresi, hapus Aktifkan Kompresi.
   - Untuk Retensi log audit, pilih opsi ini untuk menyimpan catatan audit di instans DB. Tentukan waktu retensi dalam jam. Waktu retensi maksimum adalah 35 hari.
3. Terapkan grup opsi ke instans DB baru atau yang sudah ada. Pilih salah satu cara berikut:
   - Jika Anda membuat instans DB baru, terapkan grup opsi ketika Anda meluncurkan instans.
   - Di instans DB yang sudah ada, terapkan grup opsi dengan mengubah instans lalu memberikan grup opsi baru. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Memodifikasi opsi SQL Server Audit

Setelah mengaktifkan opsi SQL Server Audit, Anda dapat mengubah pengaturan. Untuk informasi cara mengubah pengaturan opsi, lihat Memodifikasi pengaturan opsi.

Menghapus SQL Server Audit dari opsi instans DB

Anda dapat mematikan fitur SQL Server Audit dengan menonaktifkan audit lalu menghapus opsi.

Untuk menghapus audit

1. Nonaktifkan semua pengaturan audit di dalam SQL Server. Untuk mempelajari tempat audit dijalankan, buat kueri pada tampilan katalog keamanan SQL Server. Untuk informasi selengkapnya, lihat Security catalog views dalam dokumentasi Microsoft SQL Server.
2. Hapus opsi SQL Server Audit dari instans DB. Pilih salah satu cara berikut:
   - Hapus opsi SQL Server Audit dari grup opsi yang digunakan instans DB. Perubahan ini akan memengaruhi semua instans DB yang menggunakan kelompok opsi yang sama. Untuk informasi selengkapnya, lihat Menghapus opsi dari grup opsi.
   - Ubah instans DB, lalu pilih grup opsi tanpa opsi SQL Server Audit. Perubahan ini hanya memengaruhi instans DB yang Anda ubah. Anda dapat menentukan grup opsi default (kosong) atau grup opsi kustom lain. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.
3. Setelah menghapus opsi SQL Server Audit dari instans DB, Anda tidak perlu memulai ulang proses tersebut. Hapus file audit yang tidak diperlukan dari bucket S3 Anda.

Menggunakan SQL Server Audit

Anda dapat mengendalikan audit server, spesifikasi audit server, dan spesifikasi audit basis data dengan cara yang sama seperti Anda mengontrolnya untuk server basis data on-premise.

Membuat audit

Anda membuat audit server dengan cara yang sama seperti Anda membuatnya untuk server basis data on-premise. Untuk informasi cara membuat audit server, lihat CREATE SERVER AUDIT dalam dokumentasi Microsoft SQL Server.

Untuk menghindari kesalahan, patuhi batasan berikut:

- Jangan melebihi jumlah maksimum audit server yang didukung per instans, yaitu 50.
- Perintahkan SQL Server untuk menulis data ke file biner.
- Jangan gunakan `RDS_` sebagai awalan dalam nama audit server.
- Untuk `FILEPATH`, tentukan `D:\rdsdbdata\SQLAudit`.

- Untuk `MAXSIZE`, tentukan ukuran antara 2 MB dan 50 MB.
- Jangan mengonfigurasi `MAX_ROLLOVER_FILES` atau `MAX_FILES`.
- Jangan mengonfigurasi SQL Server untuk mematikan instans DB jika gagal menulis catatan audit.

Membuat spesifikasi audit

Anda membuat spesifikasi audit server dan spesifikasi audit basis data dengan cara yang sama seperti Anda membuatnya untuk server basis data on-premise. Untuk informasi tentang membuat spesifikasi audit, lihat CREATE SERVER AUDIT SPECIFICATION dan CREATE DATABASE AUDIT SPECIFICATION dalam dokumentasi Microsoft SQL Server.

Untuk menghindari kesalahan, jangan gunakan `RDS_` sebagai awalan dalam nama spesifikasi audit basis data atau spesifikasi audit server.

Melihat log audit

Log audit Anda disimpan di `D:\rdsdbdata\SQLAudit`.

Setelah SQL Server selesai menulis ke file log audit—saat file mencapai batas ukurannya—Amazon RDS akan mengunggah file ke bucket S3. Jika penyimpanan diaktifkan, Amazon RDS akan memindahkan file ke folder penyimpanan: `D:\rdsdbdata\SQLAudit\transmitted`.

Untuk informasi cara mengonfigurasi retensi, lihat Menambahkan SQL Server Audit ke opsi instans DB.

Catatan audit disimpan di instans DB hingga file log audit diunggah. Anda dapat melihat catatan audit dengan menjalankan perintah berikut.

```
SELECT   *
 FROM     msdb.dbo.rds_fn_get_audit_file
             ('D:\rdsdbdata\SQLAudit\*.sqlaudit'
             , default
             , default )
```

Anda dapat menggunakan perintah yang sama untuk melihat catatan audit di folder penyimpanan Anda dengan mengubah filter ke `D:\rdsdbdata\SQLAudit\transmitted\*.sqlaudit`.

```
SELECT   *
 FROM     msdb.dbo.rds_fn_get_audit_file
             ('D:\rdsdbdata\SQLAudit\transmitted\*.sqlaudit'
```

```
            , default
            , default )
```

Menggunakan SQL Server Audit dengan instans Multi-AZ

Untuk instans Multi-AZ, proses mengirim file log audit ke Amazon S3 serupa dengan proses untuk instans Single-AZ. Namun, ada beberapa perbedaan penting:

- Objek spesifikasi audit basis data direplikasi ke semua simpul.
- Audit server dan spesifikasi audit server tidak direplikasi ke simpul sekunder. Sebaliknya, Anda harus membuat atau memodifikasinya secara manual.

Untuk merekam audit server atau spesifikasi audit server dari kedua simpul:

1. Buat audit server atau spesifikasi audit server di simpul primer.
2. Lakukan failover ke simpul sekunder lalu buat audit server atau spesifikasi audit server dengan nama dan GUID yang sama di simpul sekunder. Gunakan parameter `AUDIT_GUID` untuk menentukan GUID.

Mengonfigurasi bucket S3

file log audit secara otomatis diunggah dari instans DB ke bucket S3 Anda. Pembatasan berikut berlaku untuk bucket S3 yang Anda gunakan sebagai target untuk file audit:

- Bucket S3 harus berada di Wilayah AWS yang sama dengan instans DB.
- Bucket S3 tidak boleh dibuka untuk umum.
- Bucket S3 tidak dapat menggunakan Kunci Objek S3.
- Pemilik bucket juga harus menjadi pemilik peran IAM.

Kunci target yang digunakan untuk menyimpan data mematuhi skema penamaan ini: `bucket-name/key-prefix/instance-name/audit-name/node_file-name.ext`

> **Note**
>
> Anda menetapkan nama bucket dan nilai awalan kunci dengan setelan opsi (`S3_BUCKET_ARN`).

Skema tersebut terdiri atas elemen-elemen berikut:

- **`bucket-name`** – Nama bucket S3 Anda.
- **`key-prefix`** – Awalan kunci kustom yang ingin Anda gunakan untuk log audit.
- **`instance-name`** – Nama instans Amazon RDS Anda.
- **`audit-name`** – Nama audit.
- **`node`** – Pengidentifikasi simpul yang merupakan sumber log audit (`node1` atau `node2`). Ada satu simpul untuk instans Single-AZ dan dua simpul replikasi untuk instans Multi-AZ. Ini bukan simpul primer dan sekunder karena peran primer dan sekunder berubah seiring waktu. Sebaliknya, pengidentifikasi simpul merupakan label sederhana.
  - **`node1`** – simpul replikasi pertama (Single-AZ hanya memiliki satu simpul).
  - **`node2`** – simpul replikasi kedua (Multi-AZ memiliki dua simpul).
- **`file-name`** – Nama file target. Nama file diambil apa adanya dari SQL Server.
- **`ext`** – Ekstensi file (`zip` atau `sqlaudit`):
  - **`zip`** – Jika kompresi diaktifkan (default).
  - **`sqlaudit`** – Jika kompresi dinonaktifkan.

Membuat peran IAM secara manual untuk SQL Server Audit

Biasanya, saat Anda membuat opsi baru, AWS Management Console akan membuat peran IAM dan kebijakan kepercayaan IAM untuk Anda. Namun, Anda dapat secara manual membuat peran IAM baru untuk digunakan dengan SQL Server Audit sehingga Anda dapat menyesuaikannya dengan persyaratan tambahan yang mungkin Anda miliki. Untuk melakukan ini, Anda membuat peran IAM dan mendelegasikan izin sehingga layanan Amazon RDS dapat menggunakan bucket Amazon S3. Saat Anda membuat peran IAM ini, Anda dapat memberikan kebijakan kepercayaan dan izin. Dengan adanya kebijakan kepercayaan, Amazon RDS dapat mengambil peran ini. Kebijakan izin menentukan tindakan yang dapat dilakukan peran ini. Untuk informasi selengkapnya, lihat Membuat peran untuk mendelegasikan izin ke layanan AWS di Panduan Pengguna AWS Identity and Access Management.

Anda dapat menggunakan contoh di bagian ini untuk membuat kebijakan hubungan kepercayaan dan izin yang Anda butuhkan.

Contoh berikut menunjukkan hubungan kepercayaan untuk SQL Server Audit. Contoh tersebut menggunakan pengguna utama layanan `rds.amazonaws.com` untuk memungkinkan RDS menulis ke bucket S3. Pengguna utama layanan adalah pengidentifikasi yang digunakan untuk memberikan

izin layanan. Setiap kali Anda mengizinkan akses ke `rds.amazonaws.com` dengan cara ini, Anda mengizinkan RDS untuk melakukan tindakan atas nama Anda. Untuk informasi pengguna utama layanan selengkapnya, lihat Elemen kebijakan AWS JSON: Pengguna Utama.

Example hubungan kepercayaan untuk SQL Server Audit

```
{
     "Version": "2012-10-17",
     "Statement": [
         {
             "Effect": "Allow",
             "Principal": {
                 "Service": "rds.amazonaws.com"
             },
             "Action": "sts:AssumeRole"
         }
     ]
 }
```

Sebaiknya gunakan kunci konteks kondisi global `aws:SourceArn` dan `aws:SourceAccount` dalam hubungan kepercayaan berbasis sumber daya untuk membatasi izin layanan ke sumber daya tertentu. Ini adalah cara paling efektif untuk melindungi dari masalah confused deputy.

Anda dapat menggunakan kedua kunci konteks kondisi global dan memiliki nilai `aws:SourceArn` yang berisi ID akun. Dalam hal ini, nilai `aws:SourceAccount` dan akun dalam nilai `aws:SourceArn` harus menggunakan ID akun yang sama ketika digunakan dalam pernyataan yang sama.

- Gunakan `aws:SourceArn` jika Anda ingin akses lintas layanan untuk satu sumber daya.
- Gunakan `aws:SourceAccount` jika Anda ingin mengizinkan sumber daya apa pun di akun tersebut dikaitkan dengan penggunaan lintas layanan.

Dalam hubungan kepercayaan, pastikan Anda menggunakan kunci konteks kondisi global `aws:SourceArn` dengan Amazon Resource Name (ARN) dari sumber daya yang mengakses peran tersebut. Untuk SQL Server Audit, pastikan Anda menyertakan grup opsi DB dan instans DB, seperti yang ditunjukkan pada contoh berikut.

Example hubungan kepercayaan dengan kunci konteks kondisi global untuk SQL Server Audit

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "rds.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceArn": [
                        "arn:aws:rds:Region:my_account_ID:db:db_instance_identifier",
                        "arn:aws:rds:Region:my_account_ID:og:option_group_name"
                    ]
                }
            }
        }
    ]
}
```

Dalam contoh kebijakan izin untuk SQL Server Audit berikut, kami menentukan ARN untuk bucket Amazon S3. Anda dapat menggunakan ARN untuk mengidentifikasi akun, pengguna, atau peran tertentu yang ingin diberi akses. Untuk informasi cara menggunakan ARN selengkapnya, lihat Amazon resource name (ARN).

Example kebijakan izin untuk SQL Server Audit

```
{
     "Version": "2012-10-17",
     "Statement": [
         {
             "Effect": "Allow",
             "Action": "s3:ListAllMyBuckets",
             "Resource": "*"
         },
         {
             "Effect": "Allow",
             "Action": [
                 "s3:ListBucket",
                 "s3:GetBucketACL",
                 "s3:GetBucketLocation"
             ],
             "Resource": "arn:aws:s3:::bucket_name"
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:ListMultipartUploadParts",
                "s3:AbortMultipartUpload"
            ],
            "Resource": "arn:aws:s3:::bucket_name/key_prefix/*"
        }
    ]
}
```

> **Note**
>
> Tindakan `s3:ListAllMyBuckets` diperlukan untuk memverifikasi bahwa akun AWS yang sama memiliki bucket S3 dan instans DB SQL Server. Tindakan tersebut mencantumkan nama bucket di akun.
> Namesapce bucket S3 bersifat global. Jika Anda tidak sengaja menghapus bucket, pengguna lain dapat membuat bucket dengan nama yang sama di akun lain. Kemudian, data SQL Server Audit akan ditulis ke bucket baru.

Dukungan untuk SQL Server Analysis Services di Amazon RDS for SQL Server

Microsoft SQL Server Analysis Services (SSAS) adalah bagian dari rangkaian Microsoft Business Intelligence (MSBI). SSAS merupakan pemrosesan analitik online (OLAP) dan alat penambangan data yang diinstal dalam SQL Server. Anda menggunakan SSAS untuk menganalisis data agar membantu membuat keputusan bisnis. SSAS berbeda dari basis data relasional SQL Server karena SSAS dioptimalkan untuk kueri dan penghitungan yang umum di lingkungan kecerdasan bisnis.

Anda dapat mengaktifkan SSAS pada instans DB yang sudah ada atau yang baru. Layanan ini diinstal pada instans DB yang sama dengan mesin basis data Anda. Untuk informasi selengkapnya tentang SSAS, lihat Microsoft Analysis services documentation.

Amazon RDS mendukung SSRS untuk SQL Server Standard dan Enterprise Edition pada versi berikut:

- Mode tabel:
  - SQL Server 2019, versi 15.00.4043.16.v1 dan yang lebih baru
  - SQL Server 2017, versi 14.00.3223.3.v1 dan yang lebih baru
  - SQL Server 2016, versi 13.00.5426.0.v1 dan yang lebih baru
- Mode multidimensi:
  - SQL Server 2017, versi 14.00.3381.3.v1 dan yang lebih baru
  - SQL Server 2016, versi 13.00.5882.1.v1 dan yang lebih baru

Daftar Isi

- Menggunakan SSAS di Amazon RDS
  - Menyiapkan pengguna yang diautentikasi Windows untuk SSAS
  - Menambahkan pengguna domain sebagai administrator basis data
  - Membuat proksi SSIS
  - Menjadwalkan pemrosesan basis data SSAS menggunakan SQL Server Agent
  - Merilis akses SSAS dari proksi
- Membuat cadangan basis data SSAS
- Memulihkan basis data SSAS
  - Memulihkan instans DB dengan waktu yang ditentukan
- Mengubah mode SSAS
- Mematikan SSAS
- Memecahkan masalah SSAS

Batasan

Batasan berikut berlaku untuk penggunaan SSAS di RDS for SQL Server:

- RDS for SQL Server mendukung menjalankan SSAS di mode Tabular atau Multidimensional. Untuk informasi selengkapnya, lihat Comparing tabular and multidimensional solutions dalam dokumentasi Microsoft.
- Anda hanya dapat menggunakan satu mode SSAS dalam satu waktu. Sebelum mengubah mode, pastikan untuk menghapus semua basis data SSAS.

  Untuk informasi selengkapnya, lihat Mengubah mode SSAS.
- Mode multidimensi tidak didukung pada SQL Server 2019.
- Instans Multi-AZ tidak didukung.
- Instans harus menggunakan Active Directory yang dikelola sendiri atau AWS Directory Service for Microsoft Active Directory untuk autentikasi SSAS. Untuk informasi selengkapnya, lihat Menggunakan Active Directory dengan RDS for SQL Server.
- Pengguna tidak diberi akses administrator server SSAS, tetapi mereka dapat diberikan akses administrator tingkat basis data.
- Satu-satunya port yang didukung untuk mengakses SSAS adalah 2383.

- Anda tidak dapat melakukan deployment pada proyek secara langsung. Kami menyediakan prosedur RDS yang disimpan untuk melakukan hal ini. Untuk informasi selengkapnya, lihat Men-deploy proyek SSAS di Amazon RDS.
- Pemrosesan selama deployment tidak didukung.
- Gunakan file .xmla untuk deployment tidak didukung.
- File input proyek SSAS dan file output cadangan basis data hanya dapat berada di folder `D:\S3` pada instans DB.

Mengaktifkan SSAS

Gunakan proses berikut untuk mengaktifkan SSAS untuk instans DB Anda:

1. Buat grup opsi baru, atau pilih grup opsi yang sudah ada.
2. Tambahkan opsi `SSAS` untuk grup opsi.
3. Kaitkan grup opsi dengan instans DB.
4. Izinkan akses masuk ke grup keamanan cloud privat virtual (VPC) untuk port pendengar SSAS.
5. Aktifkan Integrasi Amazon S3.

Membuat grup opsi untuk SSAS

Gunakan AWS Management Console atau AWS CLI untuk membuat grup opsi yang sesuai dengan mesin SQL Server dan versi instans DB yang Anda rencanakan untuk digunakan.

> **Note**
>
> Anda juga dapat menggunakan grup opsi yang ada jika grup opsi tersebut untuk mesin dan versi SQL Server yang benar.

Konsol

Prosedur berikut membuat grup opsi untuk SQL Server Standard Edition 2017.

Untuk membuat grup opsi

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.

2. Di panel navigasi, pilih Grup opsi.
3. Pilih Buat grup.
4. Di panel Buat grup opsi, lakukan hal berikut:
   a. Untuk Nama, ketikkan nama untuk grup opsi yang unik dalam akun AWS Anda, seperti **`ssas-se-2017`**. Nama tersebut hanya boleh berisi huruf, angka, dan tanda hubung.
   b. Untuk Deskripsi, masukkan deskripsi singkat grup opsi, seperti **`SSAS option group for SQL Server SE 2017`**. Deskripsi digunakan untuk tampilan.
   c. Untuk Mesin, pilih sqlserver-se.
   d. Untuk Versi mesin utama, pilih 14.00.
5. Pilih Buat.

CLI

Contoh CLI berikut akan membuat grup opsi untuk SQL Server Standard Edition 2017.

Untuk membuat grup opsi

- Gunakan salah satu perintah berikut ini.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds create-option-group \
    --option-group-name ssas-se-2017 \
    --engine-name sqlserver-se \
    --major-engine-version 14.00 \
    --option-group-description "SSAS option group for SQL Server SE 2017"
```

  Untuk Windows:

```
aws rds create-option-group ^
    --option-group-name ssas-se-2017 ^
    --engine-name sqlserver-se ^
    --major-engine-version 14.00 ^
    --option-group-description "SSAS option group for SQL Server SE 2017"
```

Menambahkan opsi SSAS ke grup opsi

Selanjutnya, gunakan AWS Management Console atau AWS CLI untuk menambahkan opsi SSAS ke grup opsi.

Konsol

Untuk menambahkan opsi SSAS

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup opsi.
3. Pilih grup opsi yang baru saja Anda buat.
4. Pilih Tambah opsi.
5. Pada Detail opsi, pilih SSAS untuk Nama opsi.
6. Pada Pengaturan opsi, lakukan berikut ini:
   a. Untuk memori Max, masukkan nilai dalam kisaran 10-80.

      Memori maks menentukan ambang batas atas tempat SSAS mulai melepaskan memori secara lebih agresif untuk memberi ruang bagi permintaan yang sedang berjalan, dan juga permintaan prioritas tinggi baru. Jumlah tersebut adalah persentase dari total memori instans DB. Nilai yang diizinkan adalah 10–80, dan default-nya adalah 45.
   b. Untuk Mode, pilih mode server SSAS, Tabel atau Multidimensi.

      Jika Anda tidak melihat pengaturan opsi Mode, itu berarti mode Multidimensi tidak didukung di Wilayah AWS Anda. Untuk informasi selengkapnya, lihat Batasan.

      Tabel adalah default.
   c. Untuk Grup keamanan, pilih grup keamanan VPC yang akan dikaitkan dengan opsi.

   > **Note**
   >
   > Port untuk mengakses SSAS, 2383, telah terisi.

7. Di bagian Penjadwalan, pilih apakah akan menambahkan opsi langsung atau pada masa pemeliharaan berikutnya.
8. Pilih Tambah opsi.

CLI

Untuk menambahkan opsi SSAS

1. Buat file JSON, misalnya `ssas-option.json`, dengan parameter berikut:

   - `OptionGroupName` – Nama grup opsi yang Anda buat atau pilih sebelumnya (`ssas-se-2017` dalam contoh berikut).
   - `Port` – Port yang Anda gunakan untuk mengakses SSAS. Satu-satunya port yang didukung adalah 2383.
   - `VpcSecurityGroupMemberships` – Keanggotaan grup keamanan VPC untuk instans DB RDS Anda.
   - `MAX_MEMORY` – Ambang batas atas di mana SSAS mulai melepaskan memori secara lebih agresif untuk memberi ruang bagi permintaan yang sedang berjalan, dan juga permintaan prioritas tinggi baru. Jumlah tersebut adalah persentase dari total memori instans DB. Nilai yang diizinkan adalah 10–80, dan default-nya adalah 45.
   - `MODE` – Mode server SSAS, salah satu `Tabular` atau `Multidimensional`. `Tabular` adalah default.

     Jika Anda menerima kesalahan bahwa pengaturan opsi `MODE` itu tidak valid, berarti mode Multidimensi tidak didukung di Wilayah AWS Anda. Untuk informasi selengkapnya, lihat Batasan.

   Berikut ini adalah contoh file JSON dengan pengaturan opsi SSAS.

   ```
   {
   "OptionGroupName": "ssas-se-2017",
   "OptionsToInclude": [
    {
    "OptionName": "SSAS",
    "Port": 2383,
    "VpcSecurityGroupMemberships": ["sg-0abcdef123"],
    "OptionSettings": [{"Name":"MAX_MEMORY","Value":"60"},
   {"Name":"MODE","Value":"Multidimensional"}]
    }],
   "ApplyImmediately": true
   }
   ```

2. Tambahkan opsi SSAS ke grup opsi.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds add-option-to-option-group \
    --cli-input-json file://ssas-option.json \
    --apply-immediately
```

Untuk Windows:

```
aws rds add-option-to-option-group ^
    --cli-input-json file://ssas-option.json ^
    --apply-immediately
```

Mengaitkan grup opsi dengan instans DB

Anda dapat menggunakan konsol atau CLI untuk mengaitkan grup opsi Anda dengan instans DB Anda.

Konsol

Kaitkan grup opsi dengan instans DB baru atau yang sudah ada:

- Untuk instans DB baru, kaitkan grup opsi dengan instans DB saat Anda meluncurkan instans tersebut. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.
- Untuk instans DB yang ada, ubah instans tersebut dan kaitkan grup opsi baru dengannya. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

> **Note**
>
> Jika Anda menggunakan instans yang ada, instans tersebut harus sudah memiliki domain Active Directory dan peran AWS Identity and Access Management (IAM) yang terkait dengannya. Jika Anda membuat instans baru, tentukan domain Active Directory dan peran IAM yang ada. Untuk informasi selengkapnya, lihat Menggunakan Active Directory dengan RDS for SQL Server.

CLI

Anda dapat mengaitkan grup opsi Anda dengan instans DB baru atau yang sudah ada.

> **Note**
>
> Jika Anda menggunakan instans yang ada, instans tersebut harus sudah memiliki domain Active Directory dan peran IAM yang terkait dengannya. Jika Anda membuat instans baru, tentukan domain Active Directory dan peran IAM yang ada. Untuk informasi selengkapnya, lihat Menggunakan Active Directory dengan RDS for SQL Server.

Membuat instans DB yang menggunakan grup opsi Anda

- Tentukan tipe mesin DB yang sama dan versi utama yang Anda gunakan saat membuat grup opsi.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds create-db-instance \
    --db-instance-identifier myssasinstance \
    --db-instance-class db.m5.2xlarge \
    --engine sqlserver-se \
    --engine-version 14.00.3223.3.v1 \
    --allocated-storage 100 \
    --manage-master-user-password \
    --master-username admin \
    --storage-type gp2 \
    --license-model li \
    --domain-iam-role-name my-directory-iam-role \
    --domain my-domain-id \
    --option-group-name ssas-se-2017
```

  Untuk Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier myssasinstance ^
    --db-instance-class db.m5.2xlarge ^
    --engine sqlserver-se ^
```

```
    --engine-version 14.00.3223.3.v1 ^
    --allocated-storage 100 ^
    --manage-master-user-password ^
    --master-username admin ^
    --storage-type gp2 ^
    --license-model li ^
    --domain-iam-role-name my-directory-iam-role ^
    --domain my-domain-id ^
    --option-group-name ssas-se-2017
```

Memodifikasi instans DB dan mengaitkan grup opsi

- Gunakan salah satu perintah berikut.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier myssasinstance \
    --option-group-name ssas-se-2017 \
    --apply-immediately
```

  Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier myssasinstance ^
    --option-group-name ssas-se-2017 ^
    --apply-immediately
```

Mengizinkan akses masuk ke grup keamanan VPC Anda

Buat aturan masuk untuk port pendengar SSAS yang ditentukan di grup keamanan VPC yang terkait dengan instans DB Anda. Untuk informasi selengkapnya tentang menyiapkan grup keamanan, lihat Memberikan akses ke instans DB di VPC Anda dengan membuat grup keamanan.

Mengaktifkan integrasi Amazon S3

Untuk mengunduh file konfigurasi model ke host Anda untuk deployment, gunakan integrasi Amazon S3. Untuk informasi selengkapnya, lihat Mengintegrasikan instans DB Amazon RDS for SQL Server dengan Amazon S3.

Men-deploy proyek SSAS di Amazon RDS

Pada RDS, Anda tidak dapat men-deploy proyek SSAS secara langsung menggunakan prosedur SQL Server Management Studio (SSMS). Untuk men-deploy proyek, gunakan prosedur RDS yang tersimpan.

> **Note**
>
> Gunakan file .xmla untuk deployment tidak didukung.

Sebelum Anda men-deploy proyek, pastikan hal-hal berikut ini:

- Integrasi Amazon S3 diaktifkan. Untuk informasi selengkapnya, lihat Mengintegrasikan instans DB Amazon RDS for SQL Server dengan Amazon S3.
- Pengaturan konfigurasi `Processing Option` diatur ke `Do Not Process`. Pengaturan ini berarti tidak ada pemrosesan yang terjadi setelah deployment.
- Anda memiliki file *`myssasproject`*`.asdatabase` dan *`myssasproject`*`.deploymentoptions`. Semuanya dibuat secara otomatis saat Anda membangun proyek SSAS.

Men-deploy proyek SSAS di RDS

1. Unduh file `.asdatabase` (model SAS) dari bucket S3 Anda ke instans DB Anda, seperti yang ditunjukkan dalam contoh berikut. Untuk informasi lebih lanjut tentang parameter unduhan, lihat Mengunduh file dari bucket Amazon S3 ke instans DB SQL Server.

   ```
   exec msdb.dbo.rds_download_from_s3
   @s3_arn_of_file='arn:aws:s3:::bucket_name/myssasproject.asdatabase',
   [@rds_file_path='D:\S3\myssasproject.asdatabase'],
   [@overwrite_file=1];
   ```

2. Unduh file `.deploymentoptions` dari bucket S3 Anda ke instans DB Anda.

```
exec msdb.dbo.rds_download_from_s3
@s3_arn_of_file='arn:aws:s3:::bucket_name/myssasproject.deploymentoptions',
[@rds_file_path='D:\S3\myssasproject.deploymentoptions'],
[@overwrite_file=1];
```

3. Deploy proyek.

```
exec msdb.dbo.rds_msbi_task
@task_type='SSAS_DEPLOY_PROJECT',
@file_path='D:\S3\myssasproject.asdatabase';
```

Memantau status tugas deployment

Untuk melacak status tugas deployment (atau download) Anda, panggil fungsi `rds_fn_task_status`. Pelacakan status membutuhkan dua parameter. Parameter pertama harus selalu `NULL` karena tidak diterpkan ke SSAS. Parameter kedua menerima ID tugas.

Untuk melihat daftar semua tugas, tetapkan parameter pertama untuk `NULL` dan parameter kedua untuk `0`, seperti yang ditunjukkan dalam contoh berikut.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,0);
```

Untuk melihat tugas tertentu, atur parameter pertama ke `NULL` dan parameter kedua ke ID tugas, seperti yang ditunjukkan dalam contoh berikut.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,42);
```

Fungsi `rds_fn_task_status` akan menampilkan informasi berikut.

Parameter output	Deskripsi
`task_id`	ID tugas.
`task_type`	Untuk SSRS, tugas dapat memiliki jenis berikut: • `SSAS_DEPLOY_PROJECT` • `SSAS_ADD_DB_ADMIN_MEMBER`

Parameter output	Deskripsi
	• SSAS_BACKUP_DB • SSAS_RESTORE_DB
database_name	Tidak berlaku untuk tugas SSAS.
% complete	Kemajuan tugas sebagai persentase.
duration (mins)	Durasi yang dihabiskan untuk tugas, dalam menit.

Parameter output	Deskripsi
`lifecycle`	Status tugas. Status yang mungkin adalah: • `CREATED` – Setelah Anda memanggil salah satu prosedur tersimpan SSRS, tugas dibuat dan statusnya akan diatur ke `CREATED`. • `IN_PROGRESS` – Setelah tugas dimulai, statusnya akan diatur ke `IN_PROGRESS`. Proses ini dapat memakan waktu sampai lima menit hingga status berubah dari `CREATED` menjadi `IN_PROGRESS`. • `SUCCESS` – Setelah tugas selesai, statusnya akan diatur ke `SUCCESS`. • `ERROR` – Jika tugas gagal, statusnya akan diatur ke `ERROR`. Untuk informasi selengkapnya mengenai kesalahan, lihat kolom `task_info`. • `CANCEL_REQUESTED` – Setelah Anda memanggil `rds_cancel_task`, status tugas akan diatur ke `CANCEL_REQUESTED`. • `CANCELLED` – Setelah tugas berhasil dibatalkan, statusnya akan diatur ke `CANCELLED`.

Parameter output	Deskripsi
`task_info`	Informasi tambahan mengenai tugas. Jika terjadi kesalahan selama pemrosesan, kolom ini akan memuat informasi tentang kesalahan tersebut. Untuk informasi selengkapnya, lihat Memecahkan masalah SSAS.
`last_updated`	Tanggal dan waktu status tugas terakhir diperbarui.
`created_at`	Tanggal dan waktu tugas dibuat.
`S3_object_arn`	Tidak berlaku untuk tugas SSAS.
`overwrite_S3_backup_file`	Tidak berlaku untuk tugas SSAS.
`KMS_master_key_arn`	Tidak berlaku untuk tugas SSAS.
`filepath`	Tidak berlaku untuk tugas SSAS.
`overwrite_file`	Tidak berlaku untuk tugas SSAS.
`task_metadata`	Metadata yang terkait dengan tugas SSAS.

Menggunakan SSAS di Amazon RDS

Setelah men-deploy proyek SSAS, Anda dapat memproses basis data OLAP secara langsung di SSMS.

Menggunakan SSAS pada RDS

1. Pada SSMS, hubungkan ke SSAS menggunakan nama pengguna dan kata sandi untuk domain Active Directory.
2. Perluas Basis data. Basis data SSAS yang baru di-deploy akan muncul.

3. Temukan string koneksi, dan perbarui nama pengguna dan kata sandi untuk memberikan akses ke basis data SQL sumber. Hal ini diperlukan untuk memproses objek SSAS.

   a. Untuk mode Tabel, lakukan hal berikut:

      1. Perluas tab Koneksi.
      2. Buka menu konteks (klik kanan) untuk , lalu pilih Properti.
      3. Perbarui nama pengguna dan kata sandi dalam string koneksi.

   b. Untuk mode Multidimensi, lakukan hal berikut:

      1. Perluas tab Sumber Data.
      2. Buka menu konteks (klik kanan) untuk objek sumber data, lalu pilih Properti.
      3. Perbarui nama pengguna dan kata sandi dalam string koneksi.

4. Buka menu konteks (klik kanan) untuk basis data SSAS yang Anda buat lalu pilih Memproses Basis Data.

   Bergantung pada ukuran data input, operasi pemrosesan akan memakan waktu beberapa menit untuk selesai.

Topik

- Menyiapkan pengguna yang diautentikasi Windows untuk SSAS
- Menambahkan pengguna domain sebagai administrator basis data
- Membuat proksi SSIS
- Menjadwalkan pemrosesan basis data SSAS menggunakan SQL Server Agent
- Merilis akses SSAS dari proksi

Menyiapkan pengguna yang diautentikasi Windows untuk SSAS

Pengguna utama dapat (kadang dipanggil pengguna master) menggunakan contoh kode berikut untuk menyiapkan login yang diautentikasi Windows dan memberikan izin prosedur yang diperlukan. Tindakan ini akan memberikan izin kepada pengguna domain untuk menjalankan tugas pelanggan SSAS, menggunakan prosedur transfer file S3, membuat kredensial, dan bekerja dengan proksi SQL Server Agent. Untuk informasi lebih lanjut, lihat Credentials (database engine) dan Create a SQL Server Agent proxy dalam dokumentasi Microsoft.

Anda dapat memberikan beberapa atau semua izin berikut sesuai kebutuhan kepada pengguna yang diautentikasi Windows.

Example

```
-- Create a server-level domain user login, if it doesn't already exist
USE [master]
GO
CREATE LOGIN [mydomain\user_name] FROM WINDOWS
GO

-- Create domain user, if it doesn't already exist
USE [msdb]
GO
CREATE USER [mydomain\user_name] FOR LOGIN [mydomain\user_name]
GO

-- Grant necessary privileges to the domain user
USE [master]
GO
GRANT ALTER ANY CREDENTIAL TO [mydomain\user_name]
GO

USE [msdb]
GO
GRANT EXEC ON msdb.dbo.rds_msbi_task TO [mydomain\user_name] with grant option
GRANT SELECT ON msdb.dbo.rds_fn_task_status TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_task_status TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_cancel_task TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_download_from_s3 TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_upload_to_s3 TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_delete_from_filesystem TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.rds_gather_file_details TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.sp_add_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_update_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_grant_login_to_proxy TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.sp_revoke_login_from_proxy TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.sp_delete_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_enum_login_for_proxy to [mydomain\user_name] with grant
 option
```

```
GRANT EXEC ON msdb.dbo.sp_enum_proxy_for_subsystem TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.rds_sqlagent_proxy TO [mydomain\user_name] with grant option
ALTER ROLE [SQLAgentUserRole] ADD MEMBER [mydomain\user_name]
GO
```

Menambahkan pengguna domain sebagai administrator basis data

Anda dapat menambahkan pengguna domain sebagai administrator basis data SSAS dengan cara berikut:

- Administrator basis data dapat menggunakan SSMS untuk membuat peran dengan hak istimewa `admin`, lalu tambahkan pengguna ke peran tersebut.
- Anda dapat menggunakan prosedur tersimpan berikut.

  ```
  exec msdb.dbo.rds_msbi_task
  @task_type='SSAS_ADD_DB_ADMIN_MEMBER',
  @database_name='myssasdb',
  @ssas_role_name='exampleRole',
  @ssas_role_member='domain_name\domain_user_name';
  ```

  Parameter berikut yang diperlukan:

  - `@task_type` – Jenis tugas MSBI, dalam kasus ini `SSAS_ADD_DB_ADMIN_MEMBER`.
  - `@database_name` – Nama basis data SSAS yang Anda berikan hak istimewa administrator.
  - `@ssas_role_name` – Nama peran administrator basis data SSAS. Jika peran tersebut belum ada, ini akan dibuat.
  - `@ssas_role_member` – Pengguna basis data SSAS yang Anda tambahkan ke peran administrator.

Membuat proksi SSIS

Untuk dapat menjadwalkan pemrosesan basis data SSAS menggunakan SQL Server Agent, buat kredenal SSAS dan proksi SSAS. Jalankan prosedur ini sebagai pengguna yang diautentikasi Windows.

Membuat kredensial SSAS

- Buat kredensial untuk proksi. Untuk melakukannya, Anda dapat menggunakan SSMS atau laporan SQL berikut.

```
USE [master]
GO
CREATE CREDENTIAL [SSAS_Credential] WITH IDENTITY = N'mydomain\user_name', SECRET =
 N'mysecret'
GO
```

> **Note**
>
> `IDENTITY` harus merupakan kredensial login yang diautentikasi domain. Ganti *`mysecret`* dengan kata sandi untuk kredensial login yang diautentikasi domain.

Membuat proksi SSAS

1. Gunakan laporan SQL berikut untuk membuat proksi.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_add_proxy
 @proxy_name=N'SSAS_Proxy',@credential_name=N'SSAS_Credential',@description=N''
GO
```

2. Gunakan laporan SQL berikut untuk memberikan akses ke proksi kepada pengguna lain.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_grant_login_to_proxy
 @proxy_name=N'SSAS_Proxy',@login_name=N'mydomain\user_name'
GO
```

3. Gunakan laporan SQL berikut untuk memberikan akses subsistem SSAS ke proksi.

```
USE [msdb]
GO
EXEC msdb.dbo.rds_sqlagent_proxy
 @task_type='GRANT_SUBSYSTEM_ACCESS',@proxy_name='SSAS_Proxy',@proxy_subsystem='SSAS'
GO
```

Untuk melihat proksi dan izin pada proksi

1. Gunakan laporan SQL berikut untuk melihat penerima izin dari proksi.

```
USE [msdb]
GO
EXEC sp_help_proxy
GO
```

2. Gunakan laporan SQL berikut untuk melihat izin subsistem.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_enum_proxy_for_subsystem
GO
```

Menjadwalkan pemrosesan basis data SSAS menggunakan SQL Server Agent

Setelah membuat kredensial dan proksi serta memberikan akses SSAS ke proksi tersebut, Anda dapat membuat pekerjaan SQL Server Agent untuk menjadwalkan pemrosesan basis data SSAS.

Menjadwalkan pemrosesan basis data SSAS

- Gunakan SSMS atau T-SQL untuk membuat pekerjaan SQL Server Agent. Contoh berikut menggunakan T-SQL. Anda dapat mengkonfigurasi jadwal pekerjaannya lebih lanjut melalui SSMS atau T-SQL.
  - Parameter `@command` menguraikan perintah XML for Analysis (XMLA) yang akan dijalankan oleh pekerjaan SQL Server Agent. Contoh ini mengkonfigurasi pemrosesan basis data Multidimensi SSAS.
  - Parameter `@server` menguraikan nama server SSAS target dari pekerjaan SQL Server Agent.

    Untuk memanggil layanan SSAS dalam instans DB RDS yang sama di mana pekerjaan SQL Server Agent berada, gunakan `localhost:2383`.

    Untuk memanggil layanan SSAS dari luar instans DB RDS, gunakan titik akhir RDS. Anda juga dapat menggunakan titik akhir Kerberos Active Directory (AD) (*`your-DB-instance-name`*`.`*`your-AD-domain-name`*) jika instans DB RDS digabungkan dengan domain yang

sama. Untuk instans DB eksternal, pastikan Anda mengonfigurasi grup keamanan VPC dengan benar yang terkait dengan instans RDS DB untuk koneksi yang aman.

Anda dapat mengedit kueri lebih lanjut untuk mendukung berbagai operasi XMLA. Lakukan pengeditan dengan langsung memodifikasi kueri T-SQL atau menggunakan UI SSMS setelah pembuatan pekerjaan SQL Server Agent.

```
USE [msdb]
GO
DECLARE @jobId BINARY(16)
EXEC msdb.dbo.sp_add_job @job_name=N'SSAS_Job',
    @enabled=1,
    @notify_level_eventlog=0,
    @notify_level_email=0,
    @notify_level_netsend=0,
    @notify_level_page=0,
    @delete_level=0,
    @category_name=N'[Uncategorized (Local)]',
    @job_id = @jobId OUTPUT
GO
EXEC msdb.dbo.sp_add_jobserver
    @job_name=N'SSAS_Job',
    @server_name = N'(local)'
GO
EXEC msdb.dbo.sp_add_jobstep @job_name=N'SSAS_Job',
 @step_name=N'Process_SSAS_Object',
    @step_id=1,
    @cmdexec_success_code=0,
    @on_success_action=1,
    @on_success_step_id=0,
    @on_fail_action=2,
    @on_fail_step_id=0,
    @retry_attempts=0,
    @retry_interval=0,
    @os_run_priority=0, @subsystem=N'ANALYSISCOMMAND',
    @command=N'<Batch xmlns="http://schemas.microsoft.com/analysisservices/2003/
engine">
        <Parallel>
            <Process xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
                xmlns:ddl2="http://schemas.microsoft.com/analysisservices/2003/
engine/2" xmlns:ddl2_2="http://schemas.microsoft.com/analysisservices/2003/
engine/2/2"
                xmlns:ddl100_100="http://schemas.microsoft.com/
analysisservices/2008/engine/100/100" xmlns:ddl200="http://schemas.microsoft.com/
analysisservices/2010/engine/200"
                xmlns:ddl200_200="http://schemas.microsoft.com/
analysisservices/2010/engine/200/200" xmlns:ddl300="http://schemas.microsoft.com/
analysisservices/2011/engine/300"
                xmlns:ddl300_300="http://schemas.microsoft.com/
analysisservices/2011/engine/300/300" xmlns:ddl400="http://schemas.microsoft.com/
analysisservices/2012/engine/400"
                xmlns:ddl400_400="http://schemas.microsoft.com/
analysisservices/2012/engine/400/400" xmlns:ddl500="http://schemas.microsoft.com/
analysisservices/2013/engine/500"
                xmlns:ddl500_500="http://schemas.microsoft.com/
analysisservices/2013/engine/500/500">
                <Object>
                    <DatabaseID>Your_SSAS_Database_ID</DatabaseID>
                </Object>
                <Type>ProcessFull</Type>
                <WriteBackTableCreation>UseExisting</WriteBackTableCreation>
            </Process>
        </Parallel>
    </Batch>',
    @server=N'localhost:2383',
    @database_name=N'master',
    @flags=0,
    @proxy_name=N'SSAS_Proxy'
GO
```

Merilis akses SSAS dari proksi

Anda dapat mencabut akses ke subsistem SSAS dan menghapus proksi SSAS menggunakan prosedur tersimpan berikut.

Untuk mencabut akses dan menghapus proksi

1. Mencabut akses subsistem.

```
USE [msdb]
GO
```

```
EXEC msdb.dbo.rds_sqlagent_proxy
 @task_type='REVOKE_SUBSYSTEM_ACCESS',@proxy_name='SSAS_Proxy',@proxy_subsystem='SSAS'
GO
```

2. Mencabut izin pada proksi.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_revoke_login_from_proxy
 @proxy_name=N'SSAS_Proxy',@name=N'mydomain\user_name'
GO
```

3. Hapus proksi.

```
USE [msdb]
GO
EXEC dbo.sp_delete_proxy @proxy_name = N'SSAS_Proxy'
GO
```

Membuat cadangan basis data SSAS

Anda dapat membuat file cadangan basis data SSAS hanya di folder `D:\S3` pada instans DB. Untuk memindahkan file cadangan ke bucket S3, gunakan Amazon S3.

Anda dapat membuat cadangan basis data SSAS sebagai berikut:

- Pengguna domain dengan peran `admin` untuk basis data tertentu dapat menggunakan SSMS untuk mencadangkan basis data ke folder `D:\S3`.

  Untuk informasi selengkapnya, lihat Menambahkan pengguna domain sebagai administrator basis data.
- Anda dapat menggunakan prosedur tersimpan berikut. Prosedur yang disimpan ini tidak mendukung enkripsi.

```
exec msdb.dbo.rds_msbi_task
@task_type='SSAS_BACKUP_DB',
@database_name='myssasdb',
@file_path='D:\S3\ssas_db_backup.abf',
[@ssas_apply_compression=1],
[@ssas_overwrite_file=1];
```

Parameter berikut yang diperlukan:

- `@task_type` – Jenis tugas MSBI, dalam kasus ini `SSAS_BACKUP_DB`.
- `@database_name` – Nama basis data SSAS yang Anda cadangkan.
- `@file_path` – Jalur untuk file cadangan SSAS. Ekstensi `.abf` diperlukan.

Parameter berikut ini bersifat opsional:

- `@ssas_apply_compression` – Apakah akan menerapkan kompresi cadangan SSAS atau tidak. Nilai yang valid adalah 1 (Ya) dan 0 (Tidak).
- `@ssas_overwrite_file` – Apakah harus menimpa file cadangan SSAS atuu tidak. Nilai yang valid adalah 1 (Ya) dan 0 (Tidak).

Memulihkan basis data SSAS

Gunakan prosedur tersimpan berikut untuk memulihkan basis data SSAS dari cadangan.

Anda tidak dapat memulihkan basis data jika ada basis data SSAS dengan nama yang sama. Prosedur tersimpan untuk memulihkan tidak mendukung file cadangan terenkripsi.

```
exec msdb.dbo.rds_msbi_task
@task_type='SSAS_RESTORE_DB',
@database_name='mynewssasdb',
@file_path='D:\S3\ssas_db_backup.abf';
```

Parameter berikut yang diperlukan:

- `@task_type` – Jenis tugas MSBI, dalam kasus ini `SSAS_RESTORE_DB`.
- `@database_name` – Nama basis data SSAS baru yang sedang Anda pulihkan.
- `@file_path` – Jalur menuju file cadangan SSAS.

Memulihkan instans DB dengan waktu yang ditentukan

P oint-in-time recovery (PITR) tidak berlaku untuk database SSAS. Jika Anda melakukan PITR, hanya data SSAS di snapshot terakhir sebelum waktu yang diminta yang tersedia di instans yang dipulihkan.

Untuk memiliki database up-to-date SSAS pada instans DB yang dipulihkan

1. Cadangkan basis data SSAS Anda ke folder `D:\S3` di instans sumber.
2. Pindahkan file cadangan ke bucket S3.
3. Transfer file cadangan dari bucket S3 ke folder `D:\S3` di instans yang dipulihkan.
4. Jalankan prosedur tersimpan untuk memulihkan basis data SSAS ke instans yang dipulihkan.

   Anda juga dapat memproses ulang proyek SSAS untuk memulihkan basis data.

Mengubah mode SSAS

Anda dapat mengubah mode tempat SSAS berjalan, baik Tabular maupun Multidimensional. Untuk mengubah mode, gunakan AWS Management Console atau AWS CLI untuk mengubah pengaturan opsi di opsi SSAS.

> **Important**
>
> Anda hanya dapat menggunakan satu mode SSAS dalam satu waktu. Pastikan Anda telah menghapus semua basis data SSAS sebelum mengubah mode, atau Anda akan menerima pesan kesalahan.

Konsol

Prosedur konsol Amazon RDS berikut akan mengubah mode SSAS ke Tabular dan menetapkan parameter `MAX_MEMORY` menjadi 70 persen.

Untuk mengubah opsi SSAS

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup opsi.
3. Pilih grup opsi dengan opsi `SSAS` yang ingin Anda modifikasi (`ssas-se-2017` dalam contoh sebelumnya).
4. Pilih Opsi ubah.
5. Ubah pengaturan opsi:

a. Untuk Memori maks., masukkan **70**.

b. Untuk Mode, pilih Tabular.

6. Pilih Opsi ubah.

AWS CLI

Contoh AWS CLI berikut akan mengubah mode SSAS ke Tabular dan menetapkan parameter MAX_MEMORY menjadi 70 persen.

Agar perintah CLI berfungsi, pastikan Anda menyertakan semua parameter yang diperlukan, meskipun Anda tidak mengubahnya.

Untuk mengubah opsi SSAS

- Gunakan salah satu perintah berikut ini.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds add-option-to-option-group \
    --option-group-name ssas-se-2017 \
    --options
 "OptionName=SSAS,VpcSecurityGroupMemberships=sg-12345e67,OptionSettings=[{Name=MAX_MEMORY,
{Name=MODE,Value=Tabular}]" \
    --apply-immediately
```

  Untuk Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name ssas-se-2017 ^
    --options
 OptionName=SSAS,VpcSecurityGroupMemberships=sg-12345e67,OptionSettings=[{Name=MAX_MEMORY,V
{Name=MODE,Value=Tabular}] ^
    --apply-immediately
```

Mematikan SSAS

Untuk menonaktfikan SSAS, hapus SSAS opsi dari grup opsi.

> **Important**
>
> Sebelum Anda menghapus opsi SSAS, hapus basis data SSAS Anda.
> Kami sangat menyarankan agar Anda mencadangkan basis data SSAS sebelum menghapusnya dan menghapus opsi SSAS.

Konsol

Untuk menghapus opsi SSAS dari grup opsi

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup opsi.
3. Pilih grup opsi dengan opsi SSAS yang ingin Anda hapus (`ssas-se-2017` dalam contoh sebelumnya).
4. Pilih Hapus opsi.
5. Pada Hapus opsi, pilih SSAS untuk Opsi yang akan dihapus.
6. Di bagian Langsung terapkan, pilih Ya untuk segera menghapus opsi, atau Tidak untuk menghapusnya pada masa pemeliharaan berikutnya.
7. Pilih Hapus.

AWS CLI

Untuk menghapus opsi SSAS dari grup opsi

- Gunakan salah satu perintah berikut ini.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds remove-option-from-option-group \
    --option-group-name ssas-se-2017 \
    --options SSAS \
    --apply-immediately
```

  Untuk Windows:

```
aws rds remove-option-from-option-group ^
    --option-group-name ssas-se-2017 ^
    --options SSAS ^
    --apply-immediately
```

Memecahkan masalah SSAS

Anda mungkin mengalami masalah berikut saat menggunakan SSAS.

Isu	Tipe	Saran pemecahan masalah
Tidak dapat mengonfigurasi opsi SSAS. Mode SSAS yang diminta adalah *new_mode*, tetapi instans DB saat ini memiliki *jumlah* basis data *current_mode* . Hapus basis data yang ada sebelum beralih ke mode *new_mode*. Untuk memperoleh kembali akses ke mode *current_mode* agar dapat menghapus data, perbarui grup opsi DB saat ini, atau tambahkan grup opsi baru dengan %s sebagai nilai pengaturan opsi MODE untuk opsi SSAS.	Peristiwa RDS	Anda tidak dapat mengubah mode SSAS jika Anda masih memiliki basis data SSAS yang menggunakan mode saat ini. Hapus basis data SSAS, lalu coba lagi.
Tidak dapat menghapus opsi SSAS karena ada *sejumlah* basis data *mode* yang ada. Opsi SSAS tidak dapat dihapus sampai semua basis data SSAS dihapus. Tambahkan opsi SSAS lagi, hapus semua basis data SSAS, lalu coba lagi.	Peristiwa RDS	Anda tidak dapat menonaktifkan SSAS jika Anda masih memiliki basis data SSAS. Hapus basis data SSAS, lalu coba lagi.
Opsi SSAS tidak diaktifkan atau sedang dalam proses diaktifkan. Coba lagi nanti.	Prosedur tersimpan RDS	Anda tidak dapat menjalankan prosedur tersimpan SSAS saat opsi dinonaktifkan, atau saat sedang dihidupkan.

Isu	Tipe	Saran pemecahan masalah
Opsi SSAS tidak dikonfigurasi dengan benar. Pastikan status keanggotaan grup opsi adalah "sinkron", lalu tinjau log peristiwa RDS untuk mengetahui pesan kesalahan konfigurasi SSAS yang relevan. Setelah penyelidikan ini, coba lagi. Jika kesalahan terus terjadi, hubungi Dukungan AWS.	Prosedur tersimpan RDS	Anda tidak dapat menjalankan prosedur tersimpan SSAS ketika keanggotaan grup opsi Anda tidak dalam status `in-sync`. Ini akan mengakibatkan opsi SSAS berada dalam status konfigurasi yang salah. Jika status keanggotaan grup opsi Anda berubah menjadi `failed` karena modifikasi opsi SSAS, ada dua kemungkinan alasan: 1. Opsi SSAS telah dihapus tanpa basis data SSAS dihapus. 2. Mode SSAS diperbarui dari Tabular ke Multidimensional, atau dari Multidimensional ke Tabular, tanpa menghapus basis data SSAS yang sudah ada. Konfigurasikan ulang opsi SSAS karena RDS hanya mengizinkan satu mode SSAS pada satu waktu, dan tidak mendukung penghapusan opsi SSAS dengan basis data SSAS yang sudah ada. Periksa log peristiwa RDS untuk mengetahui kesalahan konfigurasi instans SSAS Anda, dan selesaikan masalah yang sesuai.

Isu	Tipe	Saran pemecahan masalah
Deployment gagal. Perubahan hanya dapat di-deploy di server yang berjalan dalam mode *deployment_file_mode* . Mode server saat ini adalah *current_mode* .	Prosedur tersimpan RDS	Anda tidak dapat melakukan deployment basis data Tabular ke server Multidimensi, atau basis data Multidimensi ke server Tabular. Pastikan Anda menggunakan file dengan mode yang benar, dan verifikasi bahwa pengaturan opsi `MODE` telah diatur ke nilai yang sesuai.
Pemulihan gagal. File cadangan hanya dapat dipulihkan di server yang berjalan dalam mode *restore_file_mode* . Mode server saat ini adalah *current_mode* .	Prosedur tersimpan RDS	Anda tidak dapat mengembalikan basis data Tabular ke server Multidimensi, atau basis data Multidimensi ke server Tabular. Pastikan Anda menggunakan file dengan mode yang benar, dan verifikasi bahwa pengaturan opsi `MODE` telah diatur ke nilai yang sesuai.
Pemulihan gagal. File cadangan dan versi instans RDS DB tidak kompatibel.	Prosedur tersimpan RDS	Anda tidak dapat memulihkan basis data SSAS menggunakan versi yang tidak kompatibel dengan versi instans SQL Server. Untuk informasi selengkapnya, lihat Compatibility levels for tabular models dan Compatibility level of a multidimensional database dalam dokumentasi Microsoft.

Isu	Tipe	Saran pemecahan masalah
Pemulihan gagal. File cadangan yang ditentukan dalam operasi pemulihan rusak atau bukan file cadangan SSAS. Pastikan @rds_file_path diformat dengan benar.	Prosedur tersimpan RDS	Anda tidak dapat memulihkan basis data SSAS dengan file yang rusak. Pastikan file tersebut tidak rusak atau rusak. Kesalahan ini juga dapat muncul ketika `@rds_file_path` tidak diformat dengan benar (misalnya, ia memiliki garis miring terbalik ganda seperti pada `D:\S3\\incorrect_format.abf`).
Pemulihan gagal. Nama basis data yang dipulihkan tidak dapat berisi kata-kata yang dicadangkan atau karakter yang tidak valid: . , ; ' ` : / \\ * | ? \" & % $! + = ( ) [ ] { } < >, atau lebih panjang dari 100 karakter.	Prosedur tersimpan RDS	Nama basis data yang dipulihkan tidak dapat berisi kata atau karakter yang dicadangkan yang tidak valid, atau lebih dari 100 karakter. Untuk konvensi penamaan objek SSAS, lihat Object naming rules dalam dokumentasi Microsoft.
Nama peran yang tidak valid diberikan. Nama peran tidak boleh berisi string yang dipesan.	Prosedur tersimpan RDS	Nama peran tidak boleh berisi string yang dipesan. Untuk konvensi penamaan objek SSAS, lihat Object naming rules dalam dokumentasi Microsoft.
Nama peran yang tidak valid diberikan. Nama peran tidak dapat berisi salah satu karakter cadangan berikut: . , ; ' ` : / \\ * | ? \" & % $! + = ( ) [ ] { } < >	Prosedur tersimpan RDS	Nama peran tidak boleh berisi karakter yang dipesan. Untuk konvensi penamaan objek SSAS, lihat Object naming rules dalam dokumentasi Microsoft.

Dukungan untuk SQL Server Integration Services di Amazon RDS for SQL Server

Microsoft SQL Server Integration Services (SSIS) adalah komponen yang dapat Anda gunakan untuk melakukan berbagai tugas migrasi data. SSIS merupakan platform untuk integrasi data dan aplikasi alur kerja. SSIS dilengkapi dengan alat data warehousing yang digunakan untuk ekstraksi, transformasi, dan pemuatan (ETL) data. Anda juga dapat menggunakan alat ini untuk mengotomatiskan pemeliharaan basis data SQL Server dan pembaruan ke data kubus multidimensi.

Proyek SSIS diatur ke dalam paket-paket yang disimpan sebagai file .dtsx berbasis XML. Paket dapat berisi aliran kontrol dan aliran data. Anda menggunakan aliran data untuk mewakili operasi ETL. Setelah deployment, paket akan disimpan di SQL Server dalam basis data SSISDB. SSISDB adalah basis data pemrosesan transaksi online (OLTP) dalam mode pemulihan penuh.

Amazon RDS for SQL Server mendukung pengoperasian SSIS secara langsung di instans DB RDS. Anda dapat mengaktifkan SSIS di instans DB yang sudah ada atau yang baru. SSIS diinstal di instans DB yang sama seperti mesin basis data Anda.

RDS mendukung SSIS untuk SQL Server Standard dan Enterprise Editions pada versi berikut:

- SQL Server 2022, semua versi
- SQL Server 2019, versi 15.00.4043.16.v1 dan yang lebih baru
- SQL Server 2017, versi 14.00.3223.3.v1 dan yang lebih baru
- SQL Server 2016, versi 13.00.5426.0.v1 dan yang lebih baru

Daftar Isi

  - Menyiapkan pengguna yang diautentikasi Windows untuk SSIS
- Melakukan deployment satu proyek SSIS
- Memantau status tugas deployment
- Menggunakan SSIS
  - Mengatur pengelola koneksi basis data untuk proyek SSIS
  - Membuat proksi SSIS
  - Menjadwalkan paket SSIS menggunakan SQL Server Agent
  - Mencabut akses SSIS dari proksi
- Menonaktifkan SSIS
- Menurunkan basis data SSISDB

Batasan dan rekomendasi

Batasan dan rekomendasi berikut ini berlaku untuk menjalankan SSIS di RDS for SQL Server:

- Instans DB harus memiliki grup parameter terkait dengan parameter `clr enabled` yang diatur ke 1. Untuk informasi selengkapnya, lihat Memodifikasi parameter untuk SSIS.

  > **Note**
  >
  > Jika Anda mengaktifkan parameter `clr enabled` di SQL Server 2017 atau 2019, Anda tidak dapat menggunakan runtime bahasa umum (CLR) di instans DB Anda. Untuk informasi selengkapnya, lihat Fitur yang tidak didukung dan fitur dengan dukungan terbatas.

- Tugas alur kontrol berikut didukung:
  - Tugas Analysis Services Execute DDL
  - Tugas Analysis Services Processing
  - Tugas Bulk Insert
  - Tugas Check Database Integrity
  - Tugas Data Flow
  - Tugas Data Mining Query
  - Tugas Data Profiling
  - Tugas Execute Package

  - Tugas Execute SQL Server Agent Job
  - Tugas Execute SQL
  - Tugas Execute T-SQL Statement
  - Tugas Notify Operator
  - Tugas Rebuild Index
  - Tugas Reorganize Index
  - Tugas Shrink Database
  - Tugas Transfer Database
  - Tugas Transfer Jobs
  - Tugas Transfer Logins
  - Tugas Transfer SQL Server Objects
  - Tugas Update Statistics
- Hanya deployment proyek yang didukung.
- Menjalankan paket SSIS menggunakan SQL Server Agent didukung.
- Catatan log SSIS dapat dimasukkan hanya ke basis data yang dibuat pengguna.
- Hanya gunakan folder `D:\S3` untuk bekerja di file. File yang ditempatkan di direktori lain dihapus. Cara mengetahui beberapa detail lokasi file lainnya:
  - Tempatkan file input dan output proyek SSIS dalam folder `D:\S3`.
  - Untuk Tugas Data Flow, ubah lokasi untuk `BLOBTempStoragePath` dan `BufferTempStoragePath` ke file di dalam folder `D:\S3`. Jalur file harus dimulai dengan `D:\S3\`.
  - Pastikan semua parameter, variabel, dan ekspresi yang digunakan untuk koneksi file mengarah ke folder `D:\S3`.
  - Di instans Multi-AZ, file yang dibuat oleh SSIS dalam folder `D:\S3` akan dihapus setelah failover. Untuk informasi selengkapnya, lihat Batasan Multi-AZ untuk integrasi S3.
  - Unggah file yang dibuat oleh SSIS dalam folder `D:\S3` ke bucket Amazon S3 Anda agar dapat bertahan lama.
- Transformasi Kolom Impor dan Kolom Ekspor serta komponen Skrip pada Tugas Data Flow tidak didukung.
- Anda tidak dapat mengaktifkan pembuangan saat menjalankan paket SSIS, dan juga tidak dapat menambahkan tap data pada paket SSIS.
- Fitur SSIS Scale Out tidak didukung.

- Anda tidak dapat melakukan deployment pada proyek secara langsung. Kami menyediakan prosedur RDS yang disimpan untuk melakukan hal ini. Untuk informasi selengkapnya, lihat Melakukan deployment satu proyek SSIS.
- Menyusun file proyek SSIS (.ispac) dengan mode perlindungan `DoNotSavePasswords` untuk deployment pada RDS.
- SSIS tidak didukung di instans Always On dengan replika baca.
- Anda tidak dapat mencadangkan basis data SSISDB yang terkait dengan opsi `SSIS`.
- Mengimpor dan memulihkan basis data SSISDB dari instans SSIS lain tidak didukung.
- Anda dapat terhubung ke instans DB SQL Server lainnya atau ke sumber data Oracle. Menghubungkan ke mesin basis data lain, seperti MySQL atau PostgreSQL, tidak didukung untuk SSIS di RDS for SQL Server. Untuk informasi selengkapnya mengenai cara menghubungkan ke sumber data Oracle, lihat Server Tertaut dengan Oracle OLEDB.

Mengaktifkan SSIS

Anda dapat mengaktifkan SSIS dengan menambahkan opsi SSIS ke instans DB Anda. Gunakan proses berikut:

1. Buat grup opsi baru, atau pilih grup opsi yang sudah ada.
2. Tambahkan opsi `SSIS` untuk grup opsi.
3. Buat grup parameter baru, atau pilih grup parameter yang sudah ada.
4. Ubah grup parameter untuk mengatur parameter `clr enabled` ke 1.
5. Kaitkan grup opsi dan grup parameter dengan instans DB.
6. Aktifkan integrasi Amazon S3.

> **Note**
>
> Jika basis data dengan nama SSISDB atau kredensial login SSIS yang dicadangkan sudah ada di instans DB, Anda tidak dapat mengaktifkan SSIS di instans.

Membuat grup opsi untuk SSIS

Untuk dapat bekerja dengan SSIS, buat grup opsi atau ubah grup opsi yang sesuai dengan edisi SQL Server dan versi instans DB yang akan Anda gunakan. Untuk melakukannya, gunakan AWS Management Console atau AWS CLI.

Konsol

Prosedur berikut akan membuat grup opsi untuk SQL Server Standard Edition 2016.

Untuk membuat grup opsi

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup opsi.
3. Pilih Buat grup.
4. Di jendela Buat grup opsi, lakukan hal berikut:
   a. Untuk Nama, ketikkan nama untuk grup opsi yang unik dalam akun AWS Anda, seperti **`ssis-se-2016`**. Nama tersebut hanya boleh berisi huruf, angka, dan tanda hubung.
   b. Untuk Deskripsi, masukkan deskripsi singkat grup opsi, seperti **`SSIS option group for SQL Server SE 2016`**. Deskripsi digunakan untuk tampilan.
   c. Untuk Mesin, pilih sqlserver-se.
   d. Untuk Versi mesin utama, pilih 13.00.
5. Pilih Buat.

CLI

Prosedur berikut akan membuat grup opsi untuk SQL Server Standard Edition 2016.

Untuk membuat grup opsi

- Gunakan salah satu perintah berikut ini.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds create-option-group \
```

```
    --option-group-name ssis-se-2016 \
    --engine-name sqlserver-se \
    --major-engine-version 13.00 \
    --option-group-description "SSIS option group for SQL Server SE 2016"
```

Untuk Windows:

```
aws rds create-option-group ^
    --option-group-name ssis-se-2016 ^
    --engine-name sqlserver-se ^
    --major-engine-version 13.00 ^
    --option-group-description "SSIS option group for SQL Server SE 2016"
```

Menambahkan opsi SSIS ke grup opsi

Selanjutnya, gunakan AWS Management Console atau AWS CLI untuk menambahkan opsi `SSIS` ke grup opsi Anda.

Konsol

Untuk menambahkan opsi SSIS

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup opsi.
3. Pilih grup opsi yang baru saja Anda buat, ssis-se-2016 dalam contoh ini.
4. Pilih Tambah opsi.
5. Di bagian Detail opsi, pilih SSIS untuk Nama opsi.
6. Di bagian Penjadwalan, pilih apakah akan menambahkan opsi langsung atau pada masa pemeliharaan berikutnya.
7. Pilih Tambah opsi.

CLI

Untuk menambahkan opsi SSIS

- Tambahkan opsi `SSIS` untuk grup opsi.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds add-option-to-option-group \
    --option-group-name ssis-se-2016 \
    --options OptionName=SSIS \
    --apply-immediately
```

Untuk Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name ssis-se-2016 ^
    --options OptionName=SSIS ^
    --apply-immediately
```

Membuat grup parameter untuk SSIS

Membuat atau mengubah grup parameter untuk parameter `clr enabled` yang sesuai dengan edisi SQL Server dan versi instans DB yang akan Anda gunakan untuk SSIS.

Konsol

Prosedur berikut membuat grup parameter untuk Edisi Standar SQL Server 2016.

Untuk membuat grup parameter

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup parameter.
3. Pilih Buat grup parameter.
4. Di panel Buat grup parameter, lakukan hal berikut:
   a. Untuk Rangkaian grup parameter, pilih sqlserver-se-13.0.
   b. Untuk Nama grup, masukkan pengidentifikasi grup parameter, seperti **`ssis-sqlserver-se-13`**.
   c. Untuk Deskripsi, masukkan **`clr enabled parameter group`**.
5. Pilih Buat.

CLI

Prosedur berikut membuat grup parameter untuk Edisi Standar SQL Server 2016.

Untuk membuat grup parameter

- Jalankan salah satu perintah berikut ini.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds create-db-parameter-group \
    --db-parameter-group-name ssis-sqlserver-se-13 \
    --db-parameter-group-family "sqlserver-se-13.0" \
    --description "clr enabled parameter group"
```

  Untuk Windows:

```
aws rds create-db-parameter-group ^
    --db-parameter-group-name ssis-sqlserver-se-13 ^
    --db-parameter-group-family "sqlserver-se-13.0" ^
    --description "clr enabled parameter group"
```

Memodifikasi parameter untuk SSIS

Ubah parameter `clr enabled` di grup parameter yang sesuai dengan edisi SQL Server dan versi instans DB Anda. Untuk SSIS, atur parameter `clr enabled` ke 1.

Konsol

Prosedur berikut akan mengubah grup parameter yang telah Anda buat untuk SQL Server Standard Edition 2016.

Untuk mengubah grup parameter

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup parameter.
3. Pilih grup parameter, seperti ssis-sqlserver-se-13.

4. Di bagian Parameter, filter daftar parameter untuk **`clr`**.
5. Pilih clr diaktifkan.
6. Pilih Edit parameter.
7. Dari Nilai, pilih 1.
8. Pilih Simpan perubahan.

CLI

Prosedur berikut akan mengubah grup parameter yang telah Anda buat untuk SQL Server Standard Edition 2016.

Untuk mengubah grup parameter

- Jalankan salah satu perintah berikut ini.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name ssis-sqlserver-se-13 \
    --parameters "ParameterName='clr
 enabled',ParameterValue=1,ApplyMethod=immediate"
```

  Untuk Windows:

```
aws rds modify-db-parameter-group ^
    --db-parameter-group-name ssis-sqlserver-se-13 ^
    --parameters "ParameterName='clr
 enabled',ParameterValue=1,ApplyMethod=immediate"
```

Mengaitkan grup opsi dan grup parameter dengan instans DB

Untuk mengaitkan grup opsi SSIS dan grup parameter dengan instans DB Anda, gunakan AWS Management Console atau AWS CLI

> **Note**
>
> Jika Anda menggunakan instans yang ada, instans tersebut harus sudah memiliki domain Active Directory dan peran AWS Identity and Access Management (IAM) yang terkait dengannya. Jika Anda membuat instans baru, tentukan domain Active Directory dan peran IAM yang ada. Untuk informasi selengkapnya, lihat Menggunakan Active Directory dengan RDS for SQL Server.

Konsol

Untuk menyelesaikan pengaktifan SSIS, kaitkan grup opsi dan grup parameter SSIS Anda dengan instans DB baru atau yang sudah ada:

- Untuk instans DB baru, kaitkan saat Anda meluncurkan instans. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.
- Untuk instans DB yang sudah ada, kaitkan dengan memodifikasi instans. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

CLI

Anda dapat mengaitkan grup opsi SSIS dan grup parameter dengan instans DB baru atau yang sudah ada.

Untuk membuat instans dengan grup opsi dan grup parameter SSIS

- Tentukan jenis mesin DB dan versi utama yang sama seperti yang Anda gunakan saat membuat grup opsi.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds create-db-instance \
    --db-instance-identifier myssisinstance \
    --db-instance-class db.m5.2xlarge \
    --engine sqlserver-se \
    --engine-version 13.00.5426.0.v1 \
    --allocated-storage 100 \
    --manage-master-user-password \
```

```
    --master-username admin \
    --storage-type gp2 \
    --license-model li \
    --domain-iam-role-name my-directory-iam-role \
    --domain my-domain-id \
    --option-group-name ssis-se-2016 \
    --db-parameter-group-name ssis-sqlserver-se-13
```

Untuk Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier myssisinstance ^
    --db-instance-class db.m5.2xlarge ^
    --engine sqlserver-se ^
    --engine-version 13.00.5426.0.v1 ^
    --allocated-storage 100 ^
    --manage-master-user-password ^
    --master-username admin ^
    --storage-type gp2 ^
    --license-model li ^
    --domain-iam-role-name my-directory-iam-role ^
    --domain my-domain-id ^
    --option-group-name ssis-se-2016 ^
    --db-parameter-group-name ssis-sqlserver-se-13
```

Untuk mengubah instans serta mengaitkan grup opsi dan grup parameter SSIS

- Gunakan salah satu perintah berikut ini.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier myssisinstance \
    --option-group-name ssis-se-2016 \
    --db-parameter-group-name ssis-sqlserver-se-13 \
    --apply-immediately
```

  Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier myssisinstance ^
    --option-group-name ssis-se-2016 ^
    --db-parameter-group-name ssis-sqlserver-se-13 ^
    --apply-immediately
```

Mengaktifkan integrasi S3

Untuk mengunduh file proyek SSIS (.ispac) ke host Anda untuk deployment, gunakan integrasi file S3. Untuk informasi selengkapnya, lihat Mengintegrasikan instans DB Amazon RDS for SQL Server dengan Amazon S3.

Izin administratif untuk SSISDB

Ketika instans dibuat atau dimodifikasi menggunakan opsi SSIS, hasilnya adalah basis data SSISDB dengan peran ssis_admin dan ssis_logreader yang diberikan kepada pengguna utama. Pengguna utama memiliki hak istimewa berikut dalam SSISDB:

- mengubah peran ssis_admin
- mengganti peran ssis_logreader
- mengubah pengguna mana pun

Karena pengguna utama adalah pengguna autentikasi SQL, maka Anda tidak dapat menggunakan pengguna utama untuk menjalankan paket SSIS. Pengguna utama dapat menggunakan hak istimewa ini untuk membuat pengguna SSISDB baru dan menambahkannya ke peran ssis_admin dan ssis_logreader. Tindakan ini berguna untuk memberikan akses ke pengguna domain Anda agar dapat menggunakan SSIS.

Menyiapkan pengguna yang diautentikasi Windows untuk SSIS

Pengguna utama dapat menggunakan contoh kode berikut untuk menyiapkan kredensial login yang diautentikasi Windows dalam SSISDB dan memberikan izin prosedur yang diperlukan. Tindakan ini akan memberikan izin kepada pengguna domain untuk melakukan deployment dan menjalankan paket SSIS, menggunakan prosedur transfer file S3, membuat kredensial, dan bekerja dengan proksi SQL Server Agent. Untuk informasi lebih lanjut, lihat Credentials (database engine) dan Create a SQL Server Agent proxy dalam dokumentasi Microsoft.

> **Note**
>
> Anda dapat memberikan beberapa atau semua izin berikut sesuai kebutuhan kepada pengguna yang diautentikasi Windows.

Example

```
-- Create a server-level SQL login for the domain user, if it doesn't already exist
USE [master]
GO
CREATE LOGIN [mydomain\user_name] FROM WINDOWS
GO

-- Create a database-level account for the domain user, if it doesn't already exist

USE [SSISDB]
GO
CREATE USER [mydomain\user_name] FOR LOGIN [mydomain\user_name]

-- Add SSIS role membership to the domain user
ALTER ROLE [ssis_admin] ADD MEMBER [mydomain\user_name]
ALTER ROLE [ssis_logreader] ADD MEMBER [mydomain\user_name]
GO

-- Add MSDB role membership to the domain user
USE [msdb]
GO
CREATE USER [mydomain\user_name] FOR LOGIN [mydomain\user_name]

-- Grant MSDB stored procedure privileges to the domain user
GRANT EXEC ON msdb.dbo.rds_msbi_task TO [mydomain\user_name] with grant option
GRANT SELECT ON msdb.dbo.rds_fn_task_status TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_task_status TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_cancel_task TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_download_from_s3 TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_upload_to_s3 TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_delete_from_filesystem TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.rds_gather_file_details TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.sp_add_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_update_proxy TO [mydomain\user_name] with grant option
```

```
GRANT EXEC ON msdb.dbo.sp_grant_login_to_proxy TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.sp_revoke_login_from_proxy TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.sp_delete_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_enum_login_for_proxy to [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.sp_enum_proxy_for_subsystem TO [mydomain\user_name]  with grant
 option
GRANT EXEC ON msdb.dbo.rds_sqlagent_proxy TO [mydomain\user_name] WITH GRANT OPTION


-- Add the SQLAgentUserRole privilege to the domain user
USE [msdb]
GO
ALTER ROLE [SQLAgentUserRole] ADD MEMBER [mydomain\user_name]
GO

-- Grant the ALTER ANY CREDENTIAL privilege to the domain user
USE [master]
GO
GRANT ALTER ANY CREDENTIAL TO [mydomain\user_name]
GO
```

Melakukan deployment satu proyek SSIS

Pada RDS, Anda tidak dapat melakukan deployment proyek SSIS secara langsung menggunakan prosedur SQL Server Management Studio (SSMS) atau SSIS. Untuk mengunduh file proyek dari Amazon S3 lalu melakukan deployment, gunakan prosedur tersimpan RDS.

Untuk menjalankan prosedur tersimpan, lakukan login sebagai pengguna yang Anda berikan izin agar dapat menjalankan prosedur tersimpan. Untuk informasi lebih lanjut, lihat Menyiapkan pengguna yang diautentikasi Windows untuk SSIS.

Untuk melakukan deployment proyek SSIS

1. Unduh file proyek (.ispac).

   ```
   exec msdb.dbo.rds_download_from_s3
   @s3_arn_of_file='arn:aws:s3:::bucket_name/ssisproject.ispac',
   [@rds_file_path='D:\S3\ssisproject.ispac'],
   [@overwrite_file=1];
   ```

2. Kirim tugas deployment dan pastikan beberapa hal berikut:

   - Folder ini ada di katalog SSIS.
   - Nama proyek sesuai dengan nama proyek yang Anda gunakan saat mengembangkan proyek SSIS.

```
exec msdb.dbo.rds_msbi_task
@task_type='SSIS_DEPLOY_PROJECT',
@folder_name='DEMO',
@project_name='ssisproject',
@file_path='D:\S3\ssisproject.ispac';
```

Memantau status tugas deployment

Untuk melacak status tugas deployment Anda, panggil fungsi `rds_fn_task_status`. Pelacakan status membutuhkan dua parameter. Parameter pertama harus selalu `NULL` karena tidak berlaku untuk SSIS. Parameter kedua menerima ID tugas.

Untuk melihat daftar semua tugas, tetapkan parameter pertama untuk `NULL` dan parameter kedua untuk `0`, seperti yang ditunjukkan dalam contoh berikut.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,0);
```

Untuk melihat tugas tertentu, atur parameter pertama ke `NULL` dan parameter kedua ke ID tugas, seperti yang ditunjukkan dalam contoh berikut.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,42);
```

Fungsi `rds_fn_task_status` akan menampilkan informasi berikut.

Parameter output	Deskripsi
`task_id`	ID tugas.
`task_type`	`SSIS_DEPLOY_PROJECT`
`database_name`	Tidak berlaku untuk tugas SSIS.

Parameter output	Deskripsi
`% complete`	Kemajuan tugas sebagai persentase.
`duration (mins)`	Durasi yang dihabiskan untuk tugas, dalam menit.
`lifecycle`	Status tugas. Status yang mungkin adalah: • `CREATED` – Setelah Anda memanggil prosedur tersimpan `msdb.dbo.rds_msbi_task` , maka tugas akan dibuat, dan statusnya akan diatur ke `CREATED`. • `IN_PROGRESS` – Setelah tugas dimulai, statusnya akan diatur ke `IN_PROGRESS` . Proses ini dapat memakan waktu sampai lima menit hingga status berubah dari `CREATED` menjadi `IN_PROGRESS` . • `SUCCESS` – Setelah tugas selesai, statusnya akan diatur ke `SUCCESS`. • `ERROR` – Jika tugas gagal, statusnya akan diatur ke `ERROR`. Untuk informasi selengkapnya mengenai kesalahan, lihat kolom `task_info` . • `CANCEL_REQUESTED` – Setelah Anda memanggil `rds_cancel_task` , status tugas akan diatur ke `CANCEL_REQUESTED` . • `CANCELLED` – Setelah tugas berhasil dibatalkan, statusnya akan diatur ke `CANCELLED` .

Parameter output	Deskripsi
`task_info`	Informasi tambahan mengenai tugas. Jika terjadi kesalahan selama pemrosesan, kolom ini berisi informasi tentang kesalahan tersebut.
`last_updated`	Tanggal dan waktu status tugas terakhir diperbarui.
`created_at`	Tanggal dan waktu tugas dibuat.
`S3_object_arn`	Tidak berlaku untuk tugas SSIS.
`overwrite_S3_backup_file`	Tidak berlaku untuk tugas SSIS.
`KMS_master_key_arn`	Tidak berlaku untuk tugas SSIS.
`filepath`	Tidak berlaku untuk tugas SSIS.
`overwrite_file`	Tidak berlaku untuk tugas SSIS.
`task_metadata`	Metadata yang terkait dengan tugas SSIS.

Menggunakan SSIS

Setelah melakukan deployment proyek SSIS ke dalam katalog SSIS, Anda dapat menjalankan paket secara langsung dari SSMS atau menjadwalkannya menggunakan SQL Server Agent. Anda harus menggunakan kredensial login yang diautentikasi Windows untuk menjalankan paket SSIS. Untuk informasi selengkapnya, lihat Menyiapkan pengguna yang diautentikasi Windows untuk SSIS.

Topik

- Mengatur pengelola koneksi basis data untuk proyek SSIS
- Membuat proksi SSIS
- Menjadwalkan paket SSIS menggunakan SQL Server Agent
- Mencabut akses SSIS dari proksi

Mengatur pengelola koneksi basis data untuk proyek SSIS

Saat menggunakan pengelola koneksi, Anda dapat menggunakan jenis autentikasi ini:

- Untuk koneksi basis data lokal yang menggunakan AWS Managed Active Directory, Anda dapat menggunakan autentikasi SQL atau autentikasi Windows. Untuk autentikasi Windows, gunakan *DB_instance_name*.*fully_qualified_domain_name* sebagai nama server dari string koneksi.

  Contohnya adalah `myssisinstance.corp-ad.example.com`, dengan `myssisinstance` adalah nama instans DB dan `corp-ad.example.com` adalah nama domain yang sepenuhnya memenuhi syarat.
- Untuk koneksi jarak jauh, selalu gunakan autentikasi SQL.
- Untuk koneksi basis data lokal yang menggunakan Active Directory yang dikelola sendiri, Anda dapat menggunakan autentikasi SQL atau autentikasi Windows. Untuk autentikasi Windows, gunakan `.` atau *LocalHost* sebagai nama server dari string koneksi.

Membuat proksi SSIS

Untuk dapat menjadwalkan paket SSIS menggunakan Agen SQL Server, buat kredensial SSIS dan proksi SSIS. Jalankan prosedur ini sebagai pengguna yang diautentikasi Windows.

Untuk membuat kredensial SSIS

- Buat kredensial untuk proksi. Untuk melakukannya, Anda dapat menggunakan SSMS atau laporan SQL berikut.

```
USE [master]
GO
CREATE CREDENTIAL [SSIS_Credential] WITH IDENTITY = N'mydomain\user_name', SECRET =
 N'mysecret'
GO
```

> **Note**
>
> `IDENTITY` harus merupakan kredensial login yang diautentikasi domain. Ganti *mysecret* dengan kata sandi untuk kredensial login yang diautentikasi domain.

Setiap kali host utama SSISDB diubah, ganti kredensial proksi SSIS agar host baru dapat mengaksesnya.

Untuk membuat proksi SSIS

1. Gunakan pernyataan SQL berikut untuk membuat proksi.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_add_proxy
 @proxy_name=N'SSIS_Proxy',@credential_name=N'SSIS_Credential',@description=N''
GO
```

2. Gunakan pernyataan SQL berikut untuk memberikan akses ke proksi kepada pengguna lain.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_grant_login_to_proxy
 @proxy_name=N'SSIS_Proxy',@login_name=N'mydomain\user_name'
GO
```

3. Gunakan pernyataan SQL berikut untuk memberikan akses subsistem SSIS ke proksi.

```
USE [msdb]
GO
EXEC msdb.dbo.rds_sqlagent_proxy
 @task_type='GRANT_SUBSYSTEM_ACCESS',@proxy_name='SSIS_Proxy',@proxy_subsystem='SSIS'
GO
```

Untuk melihat proksi dan izin pada proksi

1. Gunakan laporan SQL berikut untuk melihat penerima izin dari proksi.

```
USE [msdb]
GO
EXEC sp_help_proxy
GO
```

2. Gunakan laporan SQL berikut untuk melihat izin subsistem.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_enum_proxy_for_subsystem
GO
```

Menjadwalkan paket SSIS menggunakan SQL Server Agent

Setelah membuat kredensial dan proksi serta memberikan akses SSIS ke proksi tersebut, Anda dapat membuat pekerjaan SQL Server Agent untuk menjadwalkan paket SSIS.

Untuk menjadwalkan paket SSIS

- Anda dapat menggunakan SSMS atau T-SQL untuk membuat pekerjaan SQL Server Agent. Contoh berikut menggunakan T-SQL.

```
USE [msdb]
GO
DECLARE @jobId BINARY(16)
EXEC msdb.dbo.sp_add_job @job_name=N'MYSSISJob',
@enabled=1,
@notify_level_eventlog=0,
@notify_level_email=2,
@notify_level_page=2,
@delete_level=0,
@category_name=N'[Uncategorized (Local)]',
@job_id = @jobId OUTPUT
GO
EXEC msdb.dbo.sp_add_jobserver @job_name=N'MYSSISJob',@server_name=N'(local)'
GO
EXEC msdb.dbo.sp_add_jobstep
 @job_name=N'MYSSISJob',@step_name=N'ExecuteSSISPackage',
@step_id=1,
@cmdexec_success_code=0,
@on_success_action=1,
@on_fail_action=2,
@retry_attempts=0,
@retry_interval=0,
@os_run_priority=0,
@subsystem=N'SSIS',
@command=N'/ISSERVER "\"\SSISDB\MySSISFolder\MySSISProject\MySSISPackage.dtsx\"" /
SERVER "\"my-rds-ssis-instance.corp-ad.company.com/\""
```

```
/Par "\"$ServerOption::LOGGING_LEVEL(Int16)\"";1 /Par
 "\"$ServerOption::SYNCHRONIZED(Boolean)\"";True /CALLERINFO SQLAGENT /REPORTING
 E',
@database_name=N'master',
@flags=0,
@proxy_name=N'SSIS_Proxy'
GO
```

Mencabut akses SSIS dari proksi

Anda dapat mencabut akses ke subsistem SSIS dan menghapus proksi SSIS menggunakan prosedur tersimpan berikut.

Untuk mencabut akses dan menghapus proksi

1. Mencabut akses subsistem.

```
USE [msdb]
GO
EXEC msdb.dbo.rds_sqlagent_proxy
 @task_type='REVOKE_SUBSYSTEM_ACCESS',@proxy_name='SSIS_Proxy',@proxy_subsystem='SSIS'
GO
```

2. Mencabut izin pada proksi.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_revoke_login_from_proxy
 @proxy_name=N'SSIS_Proxy',@name=N'mydomain\user_name'
GO
```

3. Hapus proksi.

```
USE [msdb]
GO
EXEC dbo.sp_delete_proxy @proxy_name = N'SSIS_Proxy'
GO
```

Menonaktifkan SSIS

Untuk menonaktifkan SSIS, hapus opsi `SSIS` dari grup opsinya.

> **Important**
>
> Menghapus opsi tidak akan menghapus basis data SSISDB sehingga Anda dapat menghapus opsi dengan aman tanpa harus kehilangan proyek SSIS.
> Anda dapat mengaktifkan kembali opsi `SSIS` setelah penghapusan agar dapat menggunakan kembali proyek SSIS yang sebelumnya di-deploy ke katalog SSIS.

Konsol

Prosedur berikut akan menghapus opsi `SSIS`.

Untuk menghapus opsi SSIS dari grup opsi

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup opsi.
3. Pilih grup opsi dengan opsi `SSIS` (`ssis-se-2016` dalam contoh sebelumnya).
4. Pilih Hapus opsi.
5. Di bagian Hapus opsi, pilih SSIS untuk Opsi yang akan dihapus.
6. Di bagian Langsung terapkan, pilih Ya untuk segera menghapus opsi, atau Tidak untuk menghapusnya pada masa pemeliharaan berikutnya.
7. Pilih Hapus.

CLI

Prosedur berikut akan menghapus opsi `SSIS`.

Untuk menghapus opsi SSIS dari grup opsi

- Gunakan salah satu perintah berikut ini.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds remove-option-from-option-group \
    --option-group-name ssis-se-2016 \
    --options SSIS \
    --apply-immediately
```

Untuk Windows:

```
aws rds remove-option-from-option-group ^
    --option-group-name ssis-se-2016 ^
    --options SSIS ^
    --apply-immediately
```

Menurunkan basis data SSISDB

Setelah menghapus opsi SSIS, basis data SSISDB tidak akan dihapus. Untuk melepaskan basis data SSISDB, gunakan prosedur tersimpan `rds_drop_ssis_database` setelah menghapus opsi SSIS.

Untuk melepaskan basis data SSIS

- Gunakan prosedur tersimpan berikut.

```
USE [msdb]
GO
EXEC dbo.rds_drop_ssis_database
GO
```

Setelah melepaskan basis data SSISDB, Anda akan mendapatkan katalog SSISDB baru jika mengaktifkan kembali opsi SSIS.

Dukungan untuk SQL Server Reporting Services di Amazon RDS for SQL Server

Microsoft SQL Server Reporting Services (SSRS) adalah aplikasi berbasis server yang digunakan untuk pembuatan dan distribusi laporan. Ini adalah bagian dari rangkaian layanan SQL Server yang juga mencakup SQL Server Analysis Services (SSAS) dan SQL Server Integration Services (SSIS). SSRS adalah layanan yang dibangun di atas SQL Server. Anda dapat menggunakannya untuk mengumpulkan data dari berbagai sumber data dan menyajikannya dengan cara yang mudah dimengerti dan siap untuk dianalisis.

Amazon RDS for SQL Server mendukung eksekusi SSRS secara langsung pada instans DB RDS. Anda dapat menggunakan SSRS dengan instans DB yang sudah ada atau yang baru.

RDS mendukung SSRS untuk SQL Server Edisi Standard dan Enterprise pada versi berikut:

- SQL Server 2022, semua versi
- SQL Server 2019, versi 15.00.4043.16.v1 dan yang lebih baru
- SQL Server 2017, versi 14.00.3223.3.v1 dan yang lebih baru
- SQL Server 2016, versi 13.00.5820.21.v1 dan yang lebih baru

Daftar Isi

- Men-deploy laporan ke SSRS
- Mengonfigurasi sumber data laporan
- Menggunakan Email SSRS untuk mengirim laporan
- Mencabut izin tingkat sistem
- Memantau status tugas
- Menonaktifkan SSRS
- Menghapus basis data SSRS

Batasan dan rekomendasi

Batasan dan rekomendasi berikut ini berlaku untuk menjalankan SSRS di RDS for SQL Server:

- Anda tidak dapat menggunakan SSRS pada instans DB yang telah membaca replika.
- Instans harus menggunakan Active Directory yang dikelola sendiri atau AWS Directory Service for Microsoft Active Directory untuk autentikasi server web dan portal web SSRS. Untuk informasi selengkapnya, lihat Menggunakan Active Directory dengan RDS for SQL Server.
- Anda tidak dapat mencadangkan basis data server pelaporan yang dibuat dengan opsi SSRS.
- Mengimpor dan memulihkan basis data server laporan dari instans SSRS lainnya tidak didukung.

  Pastikan untuk menggunakan basis data yang dibuat saat opsi SSRS ditambahkan ke instans DB RDS. Untuk informasi selengkapnya, lihat Laporkan basis data server.
- Anda tidak dapat mengonfigurasi SSRS untuk mendengarkan pada port SSL default (443). Nilai yang diizinkan adalah 1150–49511, kecuali 1234, 1434, 3260, 3343, 3389, dan 47001.
- Langganan melalui berbagi berkas Microsoft Windows tidak didukung.
- Menggunakan Reporting Services Configuration Manager tidak didukung.
- Membuat dan memodifikasi peran tidak didukung.
- Memodifikasi properti server laporan tidak didukung.
- Administrator sistem dan peran pengguna sistem tidak diizinkan.
- Anda tidak dapat mengedit penetapan peran tingkat sistem melalui portal web.

Mengaktifkan SSRS

Gunakan proses berikut untuk mengaktifkan SSRS untuk instans DB Anda:

1. Buat grup opsi baru, atau pilih grup opsi yang sudah ada.
2. Tambahkan opsi SSRS untuk grup opsi.
3. Kaitkan grup opsi dengan instans DB.
4. Izinkan akses masuk ke grup keamanan cloud privat virtual (VPC) untuk port pendengar SSRS.

Membuat grup opsi untuk SSRS

Untuk bekerja dengan SSRS, buat grup opsi yang sesuai dengan versi dan mesin SQL Server untuk instans DB yang ingin Anda gunakan. Untuk melakukannya, gunakan AWS Management Console atau AWS CLI.

> **Note**
>
> Anda juga dapat menggunakan grup opsi yang sudah ada jika itu untuk mesin dan versi SQL Server yang benar.

Konsol

Prosedur berikut membuat grup opsi untuk SQL Server Standard Edition 2017.

Untuk membuat grup opsi

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup opsi.
3. Pilih Buat grup.
4. Di panel Create option group (Buat grup opsi), lakukan hal berikut:
   a. Untuk Nama, masukkan nama yang unik untuk grup opsi dalam Akun AWS Anda, seperti **`ssrs-se-2017`**. Nama tersebut hanya boleh berisi huruf, angka, dan tanda hubung.
   b. Untuk Deskripsi, masukkan deskripsi singkat grup opsi, seperti **`SSRS option group for SQL Server SE 2017`**. Deskripsi digunakan untuk tampilan.
   c. Untuk Mesin, pilih sqlserver-se.
   d. Untuk Versi mesin utama, pilih 14.00.
5. Pilih Buat.

CLI

Prosedur berikut membuat grup opsi untuk SQL Server Standard Edition 2017.

Untuk membuat grup opsi

- Gunakan salah satu perintah berikut ini.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds create-option-group \
    --option-group-name ssrs-se-2017 \
    --engine-name sqlserver-se \
    --major-engine-version 14.00 \
    --option-group-description "SSRS option group for SQL Server SE 2017"
```

Untuk Windows:

```
aws rds create-option-group ^
    --option-group-name ssrs-se-2017 ^
    --engine-name sqlserver-se ^
    --major-engine-version 14.00 ^
    --option-group-description "SSRS option group for SQL Server SE 2017"
```

Menambahkan opsi SSRS ke grup opsi Anda

Selanjutnya, gunakan AWS Management Console atau AWS CLI untuk menambahkan opsi SSRS ke grup opsi Anda.

Konsol

Untuk menambahkan opsi SSRS

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup opsi.
3. Pilih grup opsi yang baru saja Anda buat, lalu pilih Tambahkan opsi.
4. Pada Detail opsi, pilih SSRS untuk Nama opsi.

5. Pada Pengaturan opsi, lakukan berikut ini:

   a. Masukkan port untuk layanan SSRS untuk mendengarkan. Default-nya adalah 8443. Untuk daftar nilai yang diizinkan, lihat Batasan dan rekomendasi.

   b. Masukkan nilai untuk Memori maks.

      Memori maks menentukan ambang atas yang di atasnya tidak ada permintaan alokasi memori baru yang diberikan untuk melaporkan aplikasi server. Jumlah tersebut adalah persentase dari total memori instans DB. Nilai yang diperbolehkan adalah 10-80.

   c. Untuk Grup keamanan, pilih grup keamanan VPC untuk dikaitkan dengan opsi. Gunakan grup keamanan yang sama yang terkait dengan instans DB Anda.

6. Untuk menggunakan Email SSRS untuk mengirim laporan, centang Konfigurasikan opsi pengiriman email di bagian Pengiriman email dalam layanan pelaporan, lalu lakukan berikut ini:

   a. Untuk alamat email Pengirim, masukkan alamat email yang akan digunakan di bidang Dari pada pesan yang dikirim oleh Email SSRS.

      Tentukan akun pengguna yang memiliki izin untuk mengirim email dari server SMTP.

   b. Untuk server SMTP, tentukan gateway atau server SMTP yang akan digunakan.

      Ini dapat berupa alamat IP, nama NetBIOS komputer di intranet perusahaan Anda, atau nama domain yang sepenuhnya memenuhi syarat.

   c. Untuk port SMTP, masukkan port yang akan digunakan untuk menghubungkan ke server email. Default-nya adalah 25.

   d. Untuk menggunakan autentikasi:

      i. Centang kotak Gunakan autentikasi.

      ii. Untuk Secret Amazon Resource Name (ARN) , masukkan AWS Secrets Manager ARN untuk kredensial pengguna.

         Gunakan format berikut:

         arn:aws:secretsmanager:*Region*:*AccountId*:secret:*SecretName*-*6RandomChara*

         Contoh:

         arn:aws:secretsmanager:*us-west-2*:*123456789012*:secret:*MySecret-a1b2c3*

Untuk informasi lebih lanjut tentang pembuatan rahasia, lihat Menggunakan Email SSRS untuk mengirim laporan.

e. Pilih Gunakan Secure Sockets Layer (SSL) untuk mengenkripsi pesan email menggunakan SSL.

7. Pada Scheduling, tentukan apakah akan menambahkan opsi segera atau pada periode pemeliharaan berikutnya.
8. Pilih Tambah opsi.

CLI

Untuk menambahkan opsi SSRS

1. Buat file JSON, misalnya `ssrs-option.json`.

   a. Tetapkan parameter wajib berikut:

   - `OptionGroupName` – Nama grup opsi yang Anda buat atau pilih sebelumnya (`ssrs-se-2017` dalam contoh berikut).
   - `Port` – Port untuk layanan SSRS untuk didengarkan. Default-nya adalah 8443. Untuk daftar nilai yang diizinkan, lihat Batasan dan rekomendasi.
   - `VpcSecurityGroupMemberships` – Keanggotaan grup keamanan VPC untuk instans DB RDS Anda.
   - `MAX_MEMORY` – Ambang batas atas yang di atasnya tidak ada permintaan alokasi memori baru yang diberikan untuk melaporkan aplikasi server. Jumlah tersebut adalah persentase dari total memori instans DB. Nilai yang diperbolehkan adalah 10-80.

   b. (Opsional) Tetapkan parameter berikut untuk menggunakan SSRS Email:

   - `SMTP_ENABLE_EMAIL` – Tetapkan menjadi `true` untuk menggunakan Email SSRS. Default-nya adalah `false`.
   - `SMTP_SENDER_EMAIL_ADDRESS` – Alamat email yang akan digunakan di bidang Dari pada pesan yang dikirim oleh Email SSRS. Tentukan akun pengguna yang memiliki izin untuk mengirim email dari server SMTP.
   - `SMTP_SERVER` – Gateway atau server SMTP yang akan digunakan. Ini dapat berupa alamat IP, nama NetBIOS komputer di intranet perusahaan Anda, atau nama domain yang sepenuhnya memenuhi syarat.

- SMTP_PORT – Port yang digunakan untuk menghubungkan ke server email. Default-nya adalah 25.
- SMTP_USE_SSL – Tetapkan menjadi `true` untuk mengenkripsi pesan email menggunakan SSL. Default-nya adalah `true`.
- SMTP_EMAIL_CREDENTIALS_SECRET_ARN – Secrets Manager ARN yang memegang kredensial pengguna. Gunakan format berikut:

  **arn:aws:secretsmanager:*Region*:*AccountId*:secret:*SecretName*-*6RandomCharact*

  Untuk informasi lebih lanjut tentang pembuatan rahasia, lihat Menggunakan Email SSRS untuk mengirim laporan.
- SMTP_USE_ANONYMOUS_AUTHENTICATION – Tetapkan menjadi `true` dan jangan sertakan SMTP_EMAIL_CREDENTIALS_SECRET_ARN jika Anda tidak ingin menggunakan autentikasi.

  Default-nya adalah `false` jika SMTP_ENABLE_EMAIL adalah `true`.

Contoh berikut mencakup parameter Email SSRS, menggunakan ARN rahasia.

```
{
"OptionGroupName": "ssrs-se-2017",
"OptionsToInclude": [
 {
 "OptionName": "SSRS",
 "Port": 8443,
 "VpcSecurityGroupMemberships": ["sg-0abcdef123"],
 "OptionSettings": [
            {"Name": "MAX_MEMORY","Value": "60"},
            {"Name": "SMTP_ENABLE_EMAIL","Value": "true"}
            {"Name": "SMTP_SENDER_EMAIL_ADDRESS","Value": "nobody@example.com"},
            {"Name": "SMTP_SERVER","Value": "email-smtp.us-west-2.amazonaws.com"},
            {"Name": "SMTP_PORT","Value": "25"},
            {"Name": "SMTP_USE_SSL","Value": "true"},
            {"Name": "SMTP_EMAIL_CREDENTIALS_SECRET_ARN","Value":
 "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret-a1b2c3"}
            ]
 }],
"ApplyImmediately": true
}
```

2. Tambahkan opsi SSRS untuk grup opsi.

   Example

   Untuk Linux, macOS, atau Unix:

```
aws rds add-option-to-option-group \
    --cli-input-json file://ssrs-option.json \
    --apply-immediately
```

   Untuk Windows:

```
aws rds add-option-to-option-group ^
    --cli-input-json file://ssrs-option.json ^
    --apply-immediately
```

Mengaitkan grup opsi Anda dengan instans DB Anda

Gunakan AWS Management Console atau AWS CLI untuk mengaitkan grup opsi Anda dengan instans DB Anda.

Jika Anda menggunakan instans DB yang ada, instans tersebut harus memiliki domain Active Directory dan peran AWS Identity and Access Management (IAM) yang terkait dengannya. Jika Anda membuat instans baru, tentukan domain Active Directory dan peran IAM yang ada. Untuk informasi selengkapnya, lihat Menggunakan Active Directory dengan RDS for SQL Server.

Konsol

Anda dapat mengaitkan grup opsi dengan instans DB baru atau yang sudah ada:

- Untuk instans DB baru, kaitkan grup opsi saat Anda meluncurkan instans tersebut. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.
- Untuk instans DB yang ada, ubah instans tersebut dan kaitkan grup opsi baru. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

CLI

Anda dapat mengaitkan grup opsi dengan instans DB baru atau yang sudah ada.

Untuk membuat instans DB yang menggunakan grup opsi Anda

- Tentukan jenis mesin DB dan versi utama yang sama seperti yang Anda gunakan saat membuat grup opsi.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds create-db-instance \
    --db-instance-identifier myssrsinstance \
    --db-instance-class db.m5.2xlarge \
    --engine sqlserver-se \
    --engine-version 14.00.3223.3.v1 \
    --allocated-storage 100 \
    --manage-master-user-password  \
    --master-username admin \
    --storage-type gp2 \
    --license-model li \
    --domain-iam-role-name my-directory-iam-role \
    --domain my-domain-id \
    --option-group-name ssrs-se-2017
```

  Untuk Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier myssrsinstance ^
    --db-instance-class db.m5.2xlarge ^
    --engine sqlserver-se ^
    --engine-version 14.00.3223.3.v1 ^
    --allocated-storage 100 ^
    --manage-master-user-password ^
    --master-username admin ^
    --storage-type gp2 ^
    --license-model li ^
    --domain-iam-role-name my-directory-iam-role ^
    --domain my-domain-id ^
    --option-group-name ssrs-se-2017
```

Untuk mengubah instans DB untuk menggunakan grup opsi Anda

- Jalankan salah satu perintah berikut ini.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier myssrsinstance \
    --option-group-name ssrs-se-2017 \
    --apply-immediately
```

  Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier myssrsinstance ^
    --option-group-name ssrs-se-2017 ^
    --apply-immediately
```

Mengizinkan akses masuk ke grup keamanan VPC Anda

Untuk mengizinkan akses masuk ke grup keamanan VPC yang terkait dengan instans DB Anda, buat aturan masuk untuk port pendengar SSRS yang ditentukan. Untuk informasi selengkapnya tentang menyiapkan grup keamanan, lihat Memberikan akses ke instans DB di VPC Anda dengan membuat grup keamanan.

Laporkan basis data server

Jika instans DB Anda dikaitkan dengan opsi SSRS, dua basis data baru akan dibuat di instans DB Anda:

- `rdsadmin_ReportServer`
- `rdsadmin_ReportServerTempDB`

Database ini bertindak sebagai database ReportServer dan ReportServerTemp DB. SSRS menyimpan datanya dalam ReportServer database dan menyimpan datanya di database ReportServerTemp DB. Untuk informasi selengkapnya, lihat Basis Data Server Laporan dalam dokumentasi Microsoft.

RDS memiliki dan mengelola basis data ini, sehingga operasi basis data seperti ALTER dan DROP tidak diizinkan. Akses tidak diizinkan pada basis data `rdsadmin_ReportServerTempDB`. Namun, Anda dapat melakukan operasi baca di basis data `rdsadmin_ReportServer`.

File log SSRS

Anda dapat membuat daftar, melihat, dan mengunduh file log SSRS. File log SSRS mengikuti konvensi penamaan ReportServerService _ *timestamp* .log. Log server laporan ini berada di direktori `D:\rdsdbdata\Log\SSRS`. (Direktori `D:\rdsdbdata\Log` ini juga merupakan direktori induk untuk log kesalahan dan log SQL Server Agent). Untuk informasi selengkapnya, lihat Melihat dan mencantumkan file log basis data.

Untuk instans SSRS yang ada, layanan SSRS mungkin harus dimulai ulang untuk mengakses log server laporan. Anda dapat memulai ulang layanan tersebut dengan memperbarui opsi `SSRS`.

Untuk informasi selengkapnya, lihat Bekerja dengan log Microsoft SQL Server.

Mengakses portal web SSRS

Gunakan proses berikut untuk mengakses portal web SSRS:

1. Mengaktifkan Secure Sockets Layer (SSL).
2. Berikan akses kepada pengguna domain.
3. Akses portal web menggunakan browser dan kredensial pengguna domain.

Menggunakan SSL pada RDS

SSRS menggunakan protokol SSL HTTPS untuk koneksinya. Untuk menggunakan protokol ini, impor sertifikat SSL ke sistem operasi Microsoft Windows di komputer klien Anda.

Untuk informasi selengkapnya tentang sertifikat SSL, lihat . Untuk informasi selengkapnya tentang penggunaan SSL dengan SQL Server, lihat Menggunakan SSL dengan instans DB Microsoft SQL Server.

Memberikan akses ke pengguna domain

Dalam aktivasi SSRS baru, tidak ada penetapan peran dalam SSRS. Untuk memberikan akses kepada pengguna domain atau grup pengguna ke portal web, RDS menyediakan prosedur tersimpan.

Untuk memberikan akses kepada pengguna domain di portal web

- Gunakan prosedur tersimpan berikut.

```
exec msdb.dbo.rds_msbi_task
@task_type='SSRS_GRANT_PORTAL_PERMISSION',
@ssrs_group_or_username=N'AD_domain\user';
```

Pengguna domain atau grup pengguna diberi peran sistem `RDS_SSRS_ROLE`. Peran ini diberi tugas tingkat sistem berikut:

- Menjalankan laporan
- Mengelola pekerjaan
- Mengelola jadwal bersama
- Melihat jadwal bersama

Peran tingkat item `Content Manager` di folder root juga diberikan.

Mengakses portal web

Setelah tugas `SSRS_GRANT_PORTAL_PERMISSION` berhasil diselesaikan, Anda memiliki akses ke portal menggunakan browser web. URL portal web memiliki format berikut.

```
https://rds_endpoint:port/Reports
```

Dalam format ini, hal-hal berikut berlaku:

- *`rds_endpoint`* – Titik akhir untuk instans DB RDS yang Anda gunakan dengan SSRS.

  Titik akhir tersebut dapat ditemukan di tab Konektivitas & keamanan untuk instans DB Anda. Untuk informasi selengkapnya, lihat Menghubungkan ke instans DB yang menjalankan mesin basis data Microsoft SQL Server.
- *`port`* – Port pendengar untuk SSRS yang Anda tetapkan di opsi `SSRS`.

Untuk mengakses portal web

1. Masukkan URL portal web di browser Anda.

```
https://myssrsinstance.cg034itsfake.us-east-1.rds.amazonaws.com:8443/Reports
```

2. Masuk dengan kredensial untuk pengguna domain yang Anda beri akses dengan tugas `SSRS_GRANT_PORTAL_PERMISSION`.

Men-deploy laporan ke SSRS

Setelah memiliki akses ke portal web, Anda dapat men-deploy laporan ke portal tersebut. Anda dapat menggunakan fitur Unggah di portal web untuk mengunggah laporan, atau men-deploy langsung dari SQL Server data tools (SSDT). Saat men-deploy dari SSDT, pastikan hal berikut:

- Pengguna yang meluncurkan SSDT memiliki akses ke portal web SSRS.
- Nilai `TargetServerURL` dalam properti proyek SSRS diatur ke titik akhir HTTPS instans DB RDS yang diakhiri dengan `ReportServer`, misalnya:

```
https://myssrsinstance.cg034itsfake.us-east-1.rds.amazonaws.com:8443/ReportServer
```

Mengonfigurasi sumber data laporan

Setelah men-deploy laporan ke SSRS, Anda harus mengonfigurasi sumber data laporan. Saat mengonfigurasi sumber data laporan, pastikan hal berikut:

- Untuk instans DB RDS for SQL Server yang digabungkan ke AWS Directory Service for Microsoft Active Directory, gunakan nama domain yang sepenuhnya memenuhi syarat (FQDN) sebagai nama sumber data dari string koneksi. Contohnya adalah *`myssrsinstance.corp-ad.example.com`*, dengan *`myssrsinstance`* adalah nama instans DB dan *`corp-ad.example.com`* adalah nama domain yang sepenuhnya memenuhi syarat.
- Untuk instans DB RDS for SQL Server yang digabungkan ke Active Directory yang dikelola sendiri, gunakan `.`, atau *`LocalHost`* sebagai nama sumber data string koneksi.

Menggunakan Email SSRS untuk mengirim laporan

SSRS menyertakan ekstensi Email SSRS, yang dapat Anda gunakan untuk mengirim laporan kepada pengguna.

Untuk mengonfigurasi Email SSRS, gunakan pengaturan opsi SSRS. Untuk informasi selengkapnya, lihat Menambahkan opsi SSRS ke grup opsi Anda.

Setelah mengonfigurasi Email SSRS, Anda dapat berlangganan laporan di server laporan. Untuk informasi selengkapnya, lihat Pengiriman email di Layanan Pelaporan di dokumentasi Microsoft.

Integrasi dengan AWS Secrets Manager diperlukan agar Email SSRS berfungsi pada RDS. Untuk mengintegrasikan dengan Secrets Manager, Anda harus membuat rahasia.

> **Note**
>
> Jika mengubah rahasianya nanti, Anda juga harus memperbarui opsi SSRS di grup opsi.

Untuk membuat rahasia untuk Email SSRS

1. Ikuti langkah-langkah di Membuat rahasia di Panduan Pengguna AWS Secrets Manager.
   a. Untuk Pilih jenis rahasia, pilih Jenis rahasia lainnya.
   b. Untuk pasangan kunci/nilai, masukkan berikut ini:
      - **SMTP_USERNAME** – Masukkan pengguna dengan izin untuk mengirim email dari server SMTP.
      - **SMTP_PASSWORD** – Masukkan kata sandi untuk pengguna SMTP.
   c. Untuk kunci Enkripsi, jangan gunakan default AWS KMS key. Gunakan kunci Anda sendiri, atau buat yang baru.

      Kebijakan kunci KMS harus mengizinkan tindakan `kms:Decrypt`, misalnya:

```
{
    "Sid": "Allow use of the key",
    "Effect": "Allow",
    "Principal": {
        "Service": [
            "rds.amazonaws.com"
        ]
    },
    "Action": [
        "kms:Decrypt"
    ],
```

```
    "Resource": "*"
}
```

2. Ikuti langkah-langkah di Melampirkan kebijakan izin ke rahasia di Panduan Pengguna AWS Secrets Manager. Kebijakan izin memberikan tindakan `secretsmanager:GetSecretValue` kepada prinsipal layanan `rds.amazonaws.com`.

   Sebaiknya Anda menggunakan ketentuan `aws:sourceAccount` dan `aws:sourceArn` dalam kebijakan untuk menghindari masalah wakil yang membingungkan. Gunakan Akun AWS untuk `aws:sourceAccount` dan ARN grup opsi untuk `aws:sourceArn`. Untuk informasi selengkapnya, lihat Pencegahan masalah confused deputy lintas layanan.

   Contoh berikut menunjukkan kebijakan izin.

```
{
  "Version" : "2012-10-17",
  "Statement" : [ {
    "Effect" : "Allow",
    "Principal" : {
      "Service" : "rds.amazonaws.com"
    },
    "Action" : "secretsmanager:GetSecretValue",
    "Resource" : "*",
    "Condition" : {
      "StringEquals" : {
        "aws:sourceAccount" : "123456789012"
      },
      "ArnLike" : {
        "aws:sourceArn" : "arn:aws:rds:us-west-2:123456789012:og:ssrs-se-2017"
      }
    }
  } ]
}
```

   Untuk contoh lainnya, lihat Contoh kebijakan izin untuk Secrets Manager AWS di Panduan Pengguna AWS Secrets Manager.

Mencabut izin tingkat sistem

Peran sistem `RDS_SSRS_ROLE` tidak memiliki izin yang memadai untuk menghapus penetapan peran tingkat sistem. Untuk menghapus pengguna atau grup pengguna dari `RDS_SSRS_ROLE`,

gunakan prosedur tersimpan yang sama dengan yang Anda gunakan untuk memberikan peran, tetapi gunakan jenis tugas `SSRS_REVOKE_PORTAL_PERMISSION`.

Untuk mencabut akses dari pengguna domain untuk portal web

- Gunakan prosedur tersimpan berikut.

```
exec msdb.dbo.rds_msbi_task
@task_type='SSRS_REVOKE_PORTAL_PERMISSION',
@ssrs_group_or_username=N'AD_domain\user';
```

Tindakan ini akan menghapus pengguna dari peran sistem `RDS_SSRS_ROLE`. Tindakan ini juga menghapus pengguna dari peran tingkat item `Content Manager` jika pengguna memilikinya.

Memantau status tugas

Untuk melacak status penetapan atau pencabutan tugas Anda, panggil fungsi `rds_fn_task_status`. Membutuhkan dua parameter. Parameter pertama harus selalu `NULL` karena tidak berlaku untuk SSRS. Parameter kedua menerima ID tugas.

Untuk melihat daftar semua tugas, tetapkan parameter pertama untuk `NULL` dan parameter kedua untuk `0`, seperti yang ditunjukkan dalam contoh berikut.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,0);
```

Untuk melihat tugas tertentu, atur parameter pertama ke `NULL` dan parameter kedua ke ID tugas, seperti yang ditunjukkan dalam contoh berikut.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,42);
```

Fungsi `rds_fn_task_status` akan menampilkan informasi berikut.

Parameter output	Deskripsi
`task_id`	ID tugas.
`task_type`	Untuk SSRS, tugas dapat memiliki jenis berikut: • `SSRS_GRANT_PORTAL_PERMISSION`

Parameter output	Deskripsi
	• `SSRS_REVOKE_PORTAL_PERMISSION`
`database_name`	Tidak berlaku untuk tugas SSRS.
`% complete`	Persentase kemajuan tugas.
`duration (mins)`	Durasi yang dihabiskan untuk tugas, dalam menit.

Parameter output	Deskripsi
`lifecycle`	Status tugas. Status yang mungkin adalah: • `CREATED` – Setelah Anda memanggil salah satu prosedur tersimpan SSRS, tugas dibuat dan status diubah menjadi `CREATED`. • `IN_PROGRESS` – Setelah tugas dimulai, statusnya diubah menjadi `IN_PROGRESS`. Proses ini dapat memakan waktu sampai lima menit hingga status berubah dari `CREATED` menjadi `IN_PROGRESS`. • `SUCCESS` – Setelah tugas selesai, statusnya akan diatur ke `SUCCESS`. • `ERROR` – Jika tugas gagal, statusnya akan diatur ke `ERROR`. Untuk informasi selengkapnya tentang kesalahan, lihat kolom `task_info`. • `CANCEL_REQUESTED` – Setelah Anda memanggil prosedur tersimpan `rds_cancel_task`, status tugasnya diubah menjadi `CANCEL_REQUESTED`. • `CANCELLED` – Setelah tugas berhasil dibatalkan, statusnya diubah menjadi `CANCELLED`.
`task_info`	Informasi tambahan mengenai tugas. Jika terjadi kesalahan selama pemrosesan, kolom ini berisi informasi tentang kesalahan tersebut.
`last_updated`	Tanggal dan waktu status tugas terakhir diperbarui.

Parameter output	Deskripsi
`created_at`	Tanggal dan waktu tugas dibuat.
`S3_object_arn`	Tidak berlaku untuk tugas SSRS.
`overwrite_S3_backup_file`	Tidak berlaku untuk tugas SSRS.
`KMS_master_key_arn`	Tidak berlaku untuk tugas SSRS.
`filepath`	Tidak berlaku untuk tugas SSRS.
`overwrite_file`	Tidak berlaku untuk tugas SSRS.
`task_metadata`	Metadata yang terkait dengan tugas SSRS.

Menonaktifkan SSRS

Untuk menonaktifkan SSRS, hapus opsi `SSRS` dari grup opsi. Menghapus opsi tersebut tidak akan menghapus basis data SSRS. Untuk informasi selengkapnya, lihat Menghapus basis data SSRS.

Anda dapat mengaktifkan SSRS lagi dengan menambahkan opsi `SSRS` kembali. Jika Anda juga telah menghapus basis data SSRS, membaca opsi pada instans DB yang sama akan membuat basis data server laporan baru.

Konsol

Untuk menghapus opsi SSRS dari grup opsi

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup opsi.
3. Pilih grup opsi dengan opsi `SSRS` (`ssrs-se-2017` dalam contoh sebelumnya).
4. Pilih Hapus opsi.
5. Pada Opsi penghapusan, pilih SSRS untuk Opsi yang akan dihapus.
6. Pada Langsung terapkan, pilih Ya untuk segera menghapus opsi, atau Tidak untuk menghapusnya di periode pemeliharaan berikutnya.
7. Pilih Hapus.

CLI

Untuk menghapus opsi SSRS dari grup opsi

- Jalankan salah satu perintah berikut ini.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds remove-option-from-option-group \
    --option-group-name ssrs-se-2017 \
    --options SSRS \
    --apply-immediately
```

  Untuk Windows:

```
aws rds remove-option-from-option-group ^
    --option-group-name ssrs-se-2017 ^
    --options SSRS ^
    --apply-immediately
```

Menghapus basis data SSRS

Menghapus opsi SSRS tidak akan menghapus basis data server laporan. Untuk menghapusnya, gunakan prosedur tersimpan berikut.

Untuk menghapus basis data server laporan, pastikan untuk menghapus opsi SSRS terlebih dahulu.

Untuk menghapus basis data SSRS

- Gunakan prosedur tersimpan berikut.

```
exec msdb.dbo.rds_drop_ssrs_databases
```

Dukungan untuk Microsoft Distributed Transaction Coordinator di RDS for SQL Server

Transaksi terdistribusi adalah transaksi basis data yang melibatkan dua atau beberapa host jaringan. Amazon RDS for SQL Server mendukung transaksi terdistribusi di antara host, dengan satu host dapat menjadi salah satu opsi berikut ini:

- Instans DB RDS for SQL Server
- Host SQL Server on-premise
- Host Amazon EC2 dengan SQL Server diinstal
- Semua host EC2 atau instans DB RDS lainnya dengan mesin basis data yang mendukung transaksi terdistribusi

Di RDS, dimulai dengan SQL Server 2012 (versi 11.00.5058.0.v1 dan yang lebih baru), semua edisi RDS for SQL Server mendukung transaksi terdistribusi. Dukungan disediakan menggunakan Microsoft Distributed Transaction Coordinator (MSDTC). Untuk informasi mendalam tentang MSDTC, lihat Distributed Transaction Coordinator dalam dokumentasi Microsoft.

Daftar Isi

Batasan

Batasan berikut berlaku untuk penggunaan MSDTC di RDS for SQL Server:

- MSDTC tidak didukung pada instans yang menggunakan SQL Server Database Mirroring. Untuk informasi selengkapnya, lihat Transactions - availability groups and database mirroring.
- Parameter `in-doubt xact resolution` harus diatur ke 1 atau 2. Untuk informasi selengkapnya, lihat Memodifikasi parameter untuk MSDTC.
- MSDTC mengharuskan semua host yang berpartisipasi dalam transaksi terdistribusi dapat diselesaikan menggunakan nama host mereka. RDS akan otomatis mempertahankan fungsionalitas ini untuk instans yang bergabung dengan domain. Namun, untuk instans mandiri, pastikan Anda mengonfigurasi server DNS secara manual.
- Transaksi XA Java Database Connectivity (JDBC) didukung untuk SQL Server 2017 versi 14.00.3223.3 dan yang lebih tinggi, serta SQL Server 2019.
- Transaksi terdistribusi yang bergantung pada pustaka tautan dinamis (DLL) klien pada instans RDS tidak didukung.
- Penggunaan pustaka tautan dinamis XA kustom tidak didukung.

Mengaktifkan MSDTC

Gunakan proses berikut untuk mengaktifkan MSDTC untuk instans DB Anda:

1. Buat grup opsi baru, atau pilih grup opsi yang sudah ada.
2. Tambahkan opsi `MSDTC` untuk grup opsi.
3. Buat grup parameter baru, atau pilih grup parameter yang sudah ada.
4. Ubah grup parameter untuk menyetel parameter `in-doubt xact resolution` ke 1 atau 2.
5. Kaitkan grup opsi dan grup parameter dengan instans DB.

Membuat grup opsi untuk MSDTC

Gunakan AWS Management Console atau AWS CLI untuk membuat grup opsi yang sesuai dengan mesin SQL Server dan versi instans DB Anda.

> **Note**
>
> Anda juga dapat menggunakan grup opsi yang ada jika grup opsi tersebut ditujukan mesin dan versi SQL Server yang benar.

Konsol

Prosedur berikut akan membuat grup opsi untuk SQL Server Standard Edition 2016.

Untuk membuat grup opsi

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup opsi.
3. Pilih Buat grup.
4. Di panel Buat grup opsi, lakukan hal berikut:
   a. Untuk Nama, ketikkan nama untuk grup opsi yang unik dalam akun AWS Anda, seperti **`msdtc-se-2016`**. Nama tersebut hanya boleh berisi huruf, angka, dan tanda hubung.
   b. Untuk Deskripsi, masukkan deskripsi singkat grup opsi, seperti **`MSDTC option group for SQL Server SE 2016`**. Deskripsi digunakan untuk tampilan.
   c. Untuk Mesin, pilih sqlserver-se.
   d. Untuk Versi mesin utama, pilih 13.00.
5. Pilih Buat.

CLI

Contoh berikut akan membuat grup opsi untuk SQL Server Standard Edition 2016.

Untuk membuat grup opsi

- Gunakan salah satu perintah berikut ini.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds create-option-group \
    --option-group-name msdtc-se-2016 \
    --engine-name sqlserver-se \
    --major-engine-version 13.00 \
    --option-group-description "MSDTC option group for SQL Server SE 2016"
```

Untuk Windows:

```
aws rds create-option-group ^
    --option-group-name msdtc-se-2016 ^
    --engine-name sqlserver-se ^
    --major-engine-version 13.00 ^
    --option-group-description "MSDTC option group for SQL Server SE 2016"
```

Menambahkan opsi MSDTC ke grup opsi

Selanjutnya, gunakan AWS Management Console atau AWS CLI untuk menambahkan opsi `MSDTC` ke grup opsi.

Pengaturan opsi berikut diperlukan:

- Port – Port yang Anda gunakan untuk mengakses MSDTC. Nilai yang diizinkan adalah 1150–49151 kecuali untuk 1234, 1434, 3260, 3343, 3389, dan 47001. Nilai default-nya adalah 5000.

  Pastikan port yang ingin Anda gunakan diaktifkan dalam aturan firewall Anda. Selain itu, pastikan jika diperlukan bahwa port ini diaktifkan dalam aturan masuk dan keluar untuk grup keamanan yang terkait dengan instans DB Anda. Untuk informasi selengkapnya, lihat Tidak dapat terhubung ke instans DB Amazon RDS.
- Grup keamanan – Keanggotaan grup keamanan VPC untuk instans DB RDS Anda.
- Jenis autentikasi – Mode autentikasi antara host. Jenis autentikasi berikut ini didukung:
  - Timbal Balik – Instans RDS diautentikasi satu sama lain menggunakan autentikasi terintegrasi. Jika opsi ini dipilih, semua instans yang terkait dengan grup opsi ini harus bergabung dengan domain.
  - Tidak Ada – Tidak ada autentikasi yang dilakukan antara host. Kami tidak menyarankan penggunaan mode ini di lingkungan produksi.
- Ukuran log transaksi – Ukuran log transaksi MSDTC. Nilai yang diizinkan adalah 4–1024 MB. Ukuran default-nya adalah 4 MB.

Pengaturan opsi berikut ini bersifat opsional:

- Aktifkan koneksi masuk – Apakah akan mengizinkan koneksi MSDTC masuk ke instans yang terkait dengan grup opsi ini atau tidak.
- Aktifkan koneksi keluar – Apakah akan mengizinkan koneksi MSDTC keluar dari instans yang terkait dengan grup opsi ini atau tidak.
- Aktifkan XA – Apakah akan mengizinkan transaksi XA atau tidak. Untuk informasi selengkapnya tentang protokol XA, lihat spesifikasi XA.
- Aktifkan SNA LU – Apakah akan mengizinkan protokol SNA LU digunakan untuk transaksi terdistribusi atau tidak. Untuk informasi selengkapnya tentang dukungan protokol SNA LU, lihat Managing IBM CICS LU 6.2 transactions dalam dokumentasi Microsoft.

Konsol

Untuk menambahkan opsi MSDTC

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup opsi.
3. Pilih grup opsi yang baru saja Anda buat.
4. Pilih Tambah opsi.
5. Di bagian Detail opsi, pilih MSDTC untuk Nama opsi.
6. Di bagian Pengaturan opsi:
    a. Untuk Port, masukkan nomor port untuk mengakses MSDTC. Default-nya adalah 5000.
    b. Untuk Grup keamanan, pilih grup keamanan VPC yang akan dikaitkan dengan opsi.
    c. Untuk Jenis autentikasi, pilih Timbal Balik atau Tidak Ada.
    d. Untuk Ukuran log transaksi, masukkan nilai dari 4–1024. Default-nya adalah 4.
7. Di bagian Konfigurasi tambahan, lakukan tindakan berikut:
    a. Untuk Koneksi, sebagaimana dibutuhkan pilih Aktifkan koneksi masuk dan Aktifkan koneksi keluar.
    b. Untuk Protokol yang diizinkan, sebagaimana dibutuhkan pilih Aktifkan XA dan Aktifkan SNA LU.

8. Di bagian Penjadwalan, pilih apakah akan menambahkan opsi langsung atau pada masa pemeliharaan berikutnya.
9. Pilih Tambah opsi.

   Boot ulang tidak diperlukan untuk menambahkan opsi ini.

CLI

Untuk menambahkan opsi MSDTC

1. Buat file JSON, misalnya `msdtc-option.json`, dengan parameter yang diperlukan berikut ini.

```
{
"OptionGroupName":"msdtc-se-2016",
"OptionsToInclude": [
 {
 "OptionName":"MSDTC",
 "Port":5000,
 "VpcSecurityGroupMemberships":["sg-0abcdef123"],
 "OptionSettings":[{"Name":"AUTHENTICATION","Value":"MUTUAL"},
{"Name":"TRANSACTION_LOG_SIZE","Value":"4"}]
 }],
"ApplyImmediately": true
}
```

2. Tambahkan opsi `MSDTC` untuk grup opsi.

   Example

   Untuk Linux, macOS, atau Unix:

```
aws rds add-option-to-option-group \
    --cli-input-json file://msdtc-option.json \
    --apply-immediately
```

   Untuk Windows:

```
aws rds add-option-to-option-group ^
    --cli-input-json file://msdtc-option.json ^
    --apply-immediately
```

Tidak diperlukan boot ulang.

Membuat grup parameter untuk MSDTC

Buat atau ubah grup parameter untuk parameter `in-doubt xact resolution` yang sesuai dengan edisi SQL Server dan versi instans DB Anda.

Konsol

Contoh berikut akan membuat grup parameter untuk SQL Server Standard Edition 2016.

Untuk membuat grup parameter

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup parameter.
3. Pilih Buat grup parameter.
4. Di panel Buat grup parameter, lakukan hal berikut:
   a. Untuk Rangkaian grup parameter, pilih sqlserver-se-13.0.
   b. Untuk Nama grup, masukkan pengidentifikasi grup parameter, seperti **`msdtc-sqlserver-se-13`**.
   c. Untuk Deskripsi, masukkan **`in-doubt xact resolution`**.
5. Pilih Buat.

CLI

Contoh berikut akan membuat grup parameter untuk SQL Server Standard Edition 2016.

Untuk membuat grup parameter

- Gunakan salah satu perintah berikut ini.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds create-db-parameter-group \
```

```
    --db-parameter-group-name msdtc-sqlserver-se-13 \
    --db-parameter-group-family "sqlserver-se-13.0" \
    --description "in-doubt xact resolution"
```

Untuk Windows:

```
aws rds create-db-parameter-group ^
    --db-parameter-group-name msdtc-sqlserver-se-13 ^
    --db-parameter-group-family "sqlserver-se-13.0" ^
    --description "in-doubt xact resolution"
```

Memodifikasi parameter untuk MSDTC

Ubah parameter `in-doubt xact resolution` di grup parameter yang sesuai dengan edisi SQL Server dan versi instans DB Anda.

Untuk MSDTC, atur parameter `in-doubt xact resolution` untuk salah satu dari opsi berikut:

- 1 – `Presume commit`. Setiap transaksi in-doubt MSDTC dianggap telah dilakukan.
- 2 – `Presume abort`. Setiap transaksi in-doubt MSDTC dianggap telah dihentikan.

Untuk informasi selengkapnya, lihat in-doubt xact resolution server configuration option dalam dokumentasi Microsoft.

Konsol

Contoh berikut akan mengubah grup parameter yang telah Anda buat untuk SQL Server Standard Edition 2016.

Untuk mengubah grup parameter

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup parameter.
3. Pilih grup parameter, seperti msdtc-sqlserver-se-13.
4. Di bagian Parameter, filter daftar parameter untuk **`xact`**.
5. Pilih in-doubt xact resolution.
6. Pilih Edit parameter.

7. Masukkan **1** atau **2**.
8. Pilih Smpan Perubahan.

CLI

Contoh berikut akan mengubah grup parameter yang telah Anda buat untuk SQL Server Standard Edition 2016.

Untuk mengubah grup parameter

- Gunakan salah satu perintah berikut ini.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name msdtc-sqlserver-se-13 \
    --parameters "ParameterName='in-doubt xact
 resolution',ParameterValue=1,ApplyMethod=immediate"
```

  Untuk Windows:

```
aws rds modify-db-parameter-group ^
    --db-parameter-group-name msdtc-sqlserver-se-13 ^
    --parameters "ParameterName='in-doubt xact
 resolution',ParameterValue=1,ApplyMethod=immediate"
```

Mengaitkan grup opsi dan grup parameter dengan instans DB

Anda dapat menggunakan AWS Management Console atau AWS CLI untuk mengaitkan grup opsi dan grup parameter MSDTC dengan instans DB.

Konsol

Anda dapat mengaitkan grup opsi MSDTC dan grup parameter dengan instans DB baru atau yang sudah ada.

- Untuk instans DB baru, kaitkan saat Anda meluncurkan instans. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.

- Untuk instans DB yang sudah ada, kaitkan dengan memodifikasi instans. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

> **Note**
>
> Jika Anda menggunakan instans DB yang digabungkan oleh domain, instans tersebut harus sudah memiliki domain Active Directory dan peran AWS Identity and Access Management (IAM) yang terkait dengannya. Jika Anda membuat instans baru yang digabungkan oleh domain, tentukan domain Active Directory dan peran IAM yang sudah ada. Untuk informasi selengkapnya, lihat Menggunakan AWS Managed Active Directory dengan RDS for SQL Server.

CLI

Anda dapat mengaitkan grup opsi MSDTC dan grup parameter dengan instans DB baru atau yang sudah ada.

> **Note**
>
> Jika Anda menggunakan instans DB yang sudah ada dan digabungkan oleh domain, instans tersebut harus sudah memiliki domain Active Directory dan peran IAM yang terkait dengannya. Jika Anda membuat instans baru yang digabungkan oleh domain, tentukan domain Active Directory dan peran IAM yang sudah ada. Untuk informasi selengkapnya, lihat Menggunakan AWS Managed Active Directory dengan RDS for SQL Server.

Untuk membuat instans DB dengan grup opsi dan grup parameter MSDTC

- Tentukan jenis mesin DB dan versi utama yang sama seperti yang Anda gunakan saat membuat grup opsi.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --db-instance-class db.m5.2xlarge \
    --engine sqlserver-se \
```

```
    --engine-version 13.00.5426.0.v1 \
    --allocated-storage 100 \
    --manage-master-user-password \
    --master-username admin \
    --storage-type gp2 \
    --license-model li \
    --domain-iam-role-name my-directory-iam-role \
    --domain my-domain-id \
    --option-group-name msdtc-se-2016 \
    --db-parameter-group-name msdtc-sqlserver-se-13
```

Untuk Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --db-instance-class db.m5.2xlarge ^
    --engine sqlserver-se ^
    --engine-version 13.00.5426.0.v1 ^
    --allocated-storage 100 ^
    --manage-master-user-password ^
    --master-username admin ^
    --storage-type gp2 ^
    --license-model li ^
    --domain-iam-role-name my-directory-iam-role ^
    --domain my-domain-id ^
    --option-group-name msdtc-se-2016 ^
    --db-parameter-group-name msdtc-sqlserver-se-13
```

Untuk mengubah instans DB serta mengaitkan grup opsi dan grup parameter MSDTC

- Gunakan salah satu perintah berikut ini.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --option-group-name msdtc-se-2016 \
    --db-parameter-group-name msdtc-sqlserver-se-13 \
    --apply-immediately
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --option-group-name msdtc-se-2016 ^
    --db-parameter-group-name msdtc-sqlserver-se-13 ^
    --apply-immediately
```

Menggunakan transaksi terdistribusi

Di Amazon RDS for SQL Server, Anda dapat menjalankan transaksi terdistribusi dengan cara yang sama seperti transaksi terdistribusi yang berjalan on-premise:

- Menggunakan transaksi yang dapat dipromosikan .NET framework `System.Transactions`, yang mengoptimalkan transaksi terdistribusi dengan menunda pembuatannya hingga diperlukan.

  Dalam hal ini, promosi bersifat otomatis dan tidak mengharuskan Anda untuk melakukan intervensi apa pun. Jika hanya ada satu pengelola sumber daya dalam transaksi, tidak ada promosi yang dilakukan. Untuk informasi selengkapnya tentang cakupan transaksi implisit, lihat Implementing an implicit transaction using transaction scope dalam dokumentasi Microsoft.

  Transaksi yang dapat dipromosikan didukung dengan implementasi .NET ini:

  - Dimulai dengan ADO.NET 2.0, `System.Data.SqlClient` mendukung transaksi yang dapat dipromosikan dengan SQL Server. Untuk informasi selengkapnya, lihat System.Transactions integration with SQL Server dalam dokumentasi Microsoft.
  - ODP.NET mendukung `System.Transactions`. Transaksi lokal dibuat untuk koneksi pertama yang dibuka dalam cakupan `TransactionsScope` ke Oracle Database 11g rilis 1 (versi 11.1) dan yang lebih baru. Ketika koneksi kedua dibuka, transaksi ini akan otomatis dipromosikan menjadi transaksi terdistribusi. Untuk informasi dukungan transaksi terdistribusi di ODP.NET selengkapnya, lihat Microsoft Distributed Transaction Coordinator integration dalam dokumentasi Microsoft.
- Menggunakan pernyataan `BEGIN DISTRIBUTED TRANSACTION`. Untuk informasi selengkapnya, lihat BEGIN DISTRIBUTED TRANSACTION (Transact-SQL) dalam dokumentasi Microsoft.

Menggunakan transaksi XA

Mulai dari RDS for SQL Server 2017 versi 14.00.3223.3, Anda dapat mengontrol transaksi terdistribusi menggunakan JDBC. Saat Anda mengatur pengaturan opsi `Enable XA` ke `true` dalam opsi `MSDTC`, RDS akan otomatis mengaktifkan transaksi JDBC dan memberikan peran `SqlJDBCXAUser` tersebut kepada pengguna `guest`. Tindakan ini akan memungkinkan pelaksanaan transaksi terdistribusi melalui JDBC. Untuk informasi selengkapnya, termasuk contoh kode, lihat Understanding XA transactions dalam dokumentasi Microsoft.

Menggunakan pelacakan transaksi

RDS mendukung pengendalian jejak transaksi MSDTC dan mengunduhnya dari instans DB RDS untuk memecahkan masalah. Anda dapat mengontrol sesi pelacakan transaksi dengan menjalankan prosedur RDS berikut.

```
exec msdb.dbo.rds_msdtc_transaction_tracing 'trace_action',
[@traceall='0|1'],
[@traceaborted='0|1'],
[@tracelong='0|1'];
```

Parameter berikut diperlukan:

- `trace_action` – Tindakan pelacakan. Tindakan tersebut dapat berupa `START`, `STOP`, atau `STATUS`.

Parameter berikut ini bersifat opsional:

- `@traceall` – Atur ke 1 untuk melacak semua transaksi terdistribusi. Default-nya adalah 0.
- `@traceaborted` – Atur ke 1 untuk melacak transaksi terdistribusi yang dibatalkan. Default-nya adalah 0.
- `@tracelong` – Atur ke 1 untuk melacak transaksi terdistribusi yang berlangsung lama. Default-nya adalah 0.

Example tindakan pelacakan START

Untuk memulai sesi pelacakan transaksi baru, jalankan pernyataan contoh berikut.

```
exec msdb.dbo.rds_msdtc_transaction_tracing 'START',
```

```
@traceall='0',
@traceaborted='1',
@tracelong='1';
```

> **Note**
>
> Hanya satu sesi pelacakan transaksi yang dapat aktif pada satu waktu. Jika perintah sesi pelacakan START dikeluarkan saat sesi pelacakan aktif, kesalahan akan dikembalikan dan sesi pelacakan aktif tetap tidak berubah.

Example tindakan pelacakan STOP

Untuk menghentikan sesi pelacakan transaksi, jalankan pernyataan berikut.

```
exec msdb.dbo.rds_msdtc_transaction_tracing 'STOP'
```

Pernyataan ini akan menghentikan sesi pelacakan transaksi aktif dan menyimpan data jejak transaksi ke direktori log di instans DB RDS. Baris pertama output berisi hasil keseluruhan, dan baris berikutnya menunjukkan detail operasi.

Berikut ini adalah contoh penghentian sesi pelacakan yang berhasil.

```
OK: Trace session has been successfully stopped.
Setting log file to: D:\rdsdbdata\MSDTC\Trace\dtctrace.log
Examining D:\rdsdbdata\MSDTC\Trace\msdtctr.mof for message formats,  8 found.
Searching for TMF files on path: (null)
Logfile D:\rdsdbdata\MSDTC\Trace\dtctrace.log:
 OS version    10.0.14393  (Currently running on 6.2.9200)
 Start Time    <timestamp>
 End Time      <timestamp>
 Timezone is   @tzres.dll,-932 (Bias is 0mins)
 BufferSize            16384 B
 Maximum File Size     10 MB
 Buffers  Written      Not set (Logger may not have been stopped).
 Logger Mode Settings (11000002) ( circular paged
 ProcessorCount         1
Processing completed   Buffers: 1, Events: 3, EventsLost: 0 :: Format Errors: 0,
 Unknowns: 3
Event traces dumped to d:\rdsdbdata\Log\msdtc_<timestamp>.log
```

Anda dapat menggunakan informasi terperinci untuk mencari nama file log yang dihasilkan. Untuk informasi selengkapnya tentang pengunduhan file log dari instans DB RDS, lihat Memantau file log Amazon RDS.

Log sesi pelacakan tetap berada di instans selama 35 hari. Semua log sesi pelacakan yang lebih lama akan otomatis dihapus.

Example tindakan pelacakan STATUS

Untuk melacak status sesi pelacakan transaksi, jalankan pernyataan berikut.

```
exec msdb.dbo.rds_msdtc_transaction_tracing 'STATUS'
```

Pernyataan ini menghasilkan informasi berikut sebagai baris kumpulan hasil yang terpisah.

```
OK
SessionStatus: <Started|Stopped>
TraceAll: <True|False>
TraceAborted: <True|False>
TraceLongLived: <True|False>
```

Baris pertama menunjukkan hasil keseluruhan dari operasi: `OK` atau `ERROR` beserta detail, jika ada. Baris berikutnya menunjukkan detail status sesi pelacakan:

- `SessionStatus` dapat berupa salah satu status berikut:
  - `Started` jika sesi pelacakan berjalan.
  - `Stopped` jika tidak ada sesi pelacakan yang berjalan.
- Tanda sesi pelacakan bisa bernilai `True` atau `False` bergantung pada pengaturannya dalam perintah `START`.

Memodifikasi opsi MSDTC

Setelah mengaktifkan opsi `MSDTC`, Anda dapat mengubah pengaturannya. Untuk informasi cara mengubah pengaturan opsi, lihat Memodifikasi pengaturan opsi.

> **Note**
>
> Beberapa perubahan pada pengaturan opsi MSDTC mengharuskan layanan MSDTC dimulai ulang. Persyaratan ini dapat memengaruhi transaksi terdistribusi yang berjalan.

Menonaktifkan MSDTC

Untuk menonaktifkan MSDTC, hapus opsi `MSDTC` dari grup opsinya.

Konsol

Untuk menghapus opsi MSDTC dari grup opsinya

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup opsi.
3. Pilih grup opsi dengan opsi `MSDTC` (`msdtc-se-2016` dalam contoh sebelumnya).
4. Pilih Hapus opsi.
5. Di bagian Hapus opsi, pilih MSDTC untuk Opsi yang akan dihapus.
6. Di bagian Langsung terapkan, pilih Ya untuk segera menghapus opsi, atau Tidak untuk menghapusnya pada masa pemeliharaan berikutnya.
7. Pilih Hapus.

CLI

Untuk menghapus opsi MSDTC dari grup opsinya

- Gunakan salah satu perintah berikut ini.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds remove-option-from-option-group \
    --option-group-name msdtc-se-2016 \
    --options MSDTC \
    --apply-immediately
```

  Untuk Windows:

```
aws rds remove-option-from-option-group ^
    --option-group-name msdtc-se-2016 ^
    --options MSDTC ^
```

```
    --apply-immediately
```

Memecahkan Masalah MSDTC untuk RDS for SQL Server

Dalam beberapa kasus, Anda mungkin mengalami masalah saat membuat koneksi antara MSDTC yang berjalan di komputer klien dan layanan MSDTC yang berjalan pada instans DB RDS for SQL Server. Jika demikian, pastikan beberapa hal berikut:

- Aturan masuk untuk grup keamanan yang terkait dengan instans DB telah dikonfigurasi dengan benar. Untuk informasi selengkapnya, lihat Tidak dapat terhubung ke instans DB Amazon RDS.
- Komputer klien Anda telah dikonfigurasi dengan benar.
- Aturan firewall MSDTC pada komputer klien Anda telah diaktifkan.

Untuk mengonfigurasi komputer klien

1. Buka Component Services.

   Bisa juga melalui Server Manager, pilih Tools, lalu pilih Component Services.
2. Perluas Component Services, perluas Computers, perluas My Computer, lalu perluas Distributed Transaction Coordinator.
3. Buka menu konteks (klik kanan) untuk Local DTC lalu pilih Properties.
4. Pilih tab Security.
5. Pilih semua dari opsi berikut:

   - Network DTC Access
   - Allow Inbound
   - Allow Outbound
6. Pastikan bahwa mode autentikasi yang benar telah dipilih:

   - Mutual Authentication Required – Mesin klien digabungkan ke domain yang sama dengan simpul lain yang berpartisipasi dalam transaksi terdistribusi, atau ada hubungan kepercayaan yang dikonfigurasi antar-domain.
   - No Authentication Required – Semua kasus lainnya.
7. Pilih OK untuk menyimpan perubahan Anda.
8. Jika diminta untuk memulai ulang layanan, pilih Yes.

Untuk mengaktifkan aturan firewall MSDTC

1. Buka Windows Firewall, lalu pilih Advanced settings.

   Bisa juga melalui Server Manager, pilih Tools, lalu pilih Windows Firewall with Advanced Security.

   > **Note**
   >
   > Bergantung pada sistem operasi Anda, Windows Firewall mungkin disebut Windows Defender Firewall.

2. Pilih Inbound Rules di panel kiri.
3. Aktifkan aturan firewall berikut, jika belum diaktifkan:

   - Distributed Transaction Coordinator (RPC)
   - Distributed Transaction Coordinator (RPC)-EPMAP
   - Distributed Transaction Coordinator (TCP-In)
4. Tutup Windows Firewall.

Tugas DBA umum untuk Microsoft SQL Server

Bagian ini menjelaskan implementasi spesifik Amazon RDS dari beberapa tugas DBA umum untuk instans DB yang menjalankan mesin basis data Microsoft SQL Server. Agar dapat memberikan pengalaman layanan terkelola, Amazon RDS tidak memberikan akses shell ke instans DB, serta membatasi akses ke prosedur dan tabel sistem tertentu yang memerlukan hak istimewa lanjutan.

> **Note**
>
> Saat bekerja dengan instans DB SQL Server, Anda dapat menjalankan skrip untuk memodifikasi basis data yang baru dibuat, tetapi Anda tidak dapat memodifikasi basis data [model], basis data yang digunakan sebagai model untuk basis data baru.

Topik

Mengakses basis data tempdb pada instans DB Microsoft SQL Server di Amazon RDS

Anda dapat mengakses basis data `tempdb` pada instans DB Microsoft SQL Server di Amazon RDS. Anda dapat menjalankan kode di `tempdb` dengan menggunakan Transact-SQL melalui Microsoft SQL Server Management Studio (SSMS), atau aplikasi klien SQL standar lainnya. Untuk informasi selengkapnya tentang terhubung ke instans DB Anda, lihat Menghubungkan ke instans DB yang menjalankan mesin basis data Microsoft SQL Server.

Pengguna master untuk instans DB Anda diberi akses `CONTROL` ke `tempdb` sehingga pengguna ini dapat mengubah opsi basis data `tempdb`. Pengguna utama bukan pemilik basis data untuk basis data `tempdb`. Jika perlu, pengguna induk dapat memberikan akses `CONTROL` kepada pengguna lain sehingga mereka juga dapat mengubah opsi basis data `tempdb`.

> **Note**
>
> Anda tidak dapat menjalankan Perintah Konsol basis data (DBCC) pada basis data `tempdb`.

Memodifikasi opsi basis data tempdb

Anda dapat memodifikasi opsi basis data pada basis data `tempdb` di instans DB Amazon RDS Anda. Untuk informasi selengkapnya tentang opsi mana yang dapat dimodifikasi, lihat basis data tempdb di dokumentasi Microsoft.

Opsi basis data seperti opsi ukuran file maksimum persisten setelah Anda memulai ulang instans DB Anda. Anda dapat memodifikasi opsi basis data untuk mengoptimalkan performa ketika mengimpor data, dan untuk mencegah kehabisan penyimpanan.

Mengoptimalkan performa ketika mengimpor data

Untuk mengoptimalkan performa saat mengimpor sejumlah besar data ke dalam instans DB Anda, atur properti `SIZE` dan `FILEGROWTH` dari basis data tempdb ke jumlah besar. Untuk informasi selengkapnya tentang cara mengoptimalkan `tempdb`, lihat Mengoptimalkan performa tempdb di dokumentasi Microsoft.

Contoh berikut menunjukkan pengaturan ukuran menjadi 100 GB dan pertumbuhan file menjadi 10 persen.

```
alter database[tempdb] modify file (NAME = N'templog', SIZE=100GB, FILEGROWTH = 10%)
```

Mencegah masalah penyimpanan

Untuk mencegah basis data `tempdb` menggunakan semua ruang disk yang tersedia, atur properti `MAXSIZE`. Contoh berikut ini menunjukkan cara mengatur properti ke 2048 MB.

```
alter database [tempdb] modify file (NAME = N'templog', MAXSIZE = 2048MB)
```

Mengurangi basis data tempdb

Ada dua cara untuk mengurangi basis data `tempdb` di instans DB Amazon RDS Anda. Anda dapat menggunakan prosedur `rds_shrink_tempdbfile`, atau Anda dapat mengatur properti `SIZE`,

Menggunakan prosedur rds_shrink_tempdbfile

Anda dapat menggunakan prosedur Amazon RDS `msdb.dbo.rds_shrink_tempdbfile` untuk mengurangi basis data `tempdb`. Anda hanya dapat memanggil `rds_shrink_tempdbfile` jika Anda memiliki akses `CONTROL` ke `tempdb`. Ketika Anda memanggil `rds_shrink_tempdbfile`, tidak ada waktu henti untuk instans DB Anda.

Prosedur `rds_shrink_tempdbfile` memiliki parameter berikut.

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
`@temp_filename`	SYSNAME	—	diperlukan	Nama logis dari file yang akan dikurangi.
`@target_size`	int	null	opsional	Ukuran baru untuk file, dalam megabyte.

Contoh berikut ini akan mengambil nama file untuk basis data `tempdb`.

```
use tempdb;
GO

select name, * from sys.sysfiles;
```

```
GO
```

Contoh berikut mengurangi file basis data `tempdb` bernama `test_file`, dan meminta ukuran baru sebesar `10` megabyte:

```
exec msdb.dbo.rds_shrink_tempdbfile @temp_filename = N'test_file', @target_size = 10;
```

Mengatur properti SIZE

Anda juga dapat mengurangi basis data `tempdb` dengan mengatur properti `SIZE`, lalu memulai ulang instans DB Anda. Untuk informasi selengkapnya tentang cara memulai ulang instans DB, lihat Mem-boot ulang instans DB.

Contoh berikut ini menunjukkan cara mengatur properti `SIZE` ke 1024 MB.

```
alter database [tempdb] modify file (NAME = N'templog', SIZE = 1024MB)
```

Konfigurasi TempDB untuk penerapan Multi-AZ

Jika instans RDS untuk SQL Server DB Anda berada dalam Penyebaran Multi-AZ menggunakan Pencerminan Database (DBM) atau Grup Ketersediaan Selalu Aktif (AG), ingatlah pertimbangan berikut untuk menggunakan database. `tempdb`

Anda tidak dapat mereplikasi `tempdb` data dari instans DB utama Anda ke instans DB sekunder Anda. Ketika Anda gagal ke instance DB sekunder, `tempdb` pada instance DB sekunder itu akan kosong.

Anda dapat menyinkronkan konfigurasi opsi `tempdb` database, termasuk ukuran file dan pengaturan pertumbuhan otomatis, dari instans DB utama Anda ke instans DB sekunder Anda. Sinkronisasi `tempDB` konfigurasi didukung pada semua RDS untuk versi SQL Server. Anda dapat mengaktifkan sinkronisasi otomatis `tempdb` konfigurasi dengan menggunakan prosedur tersimpan berikut:

```
EXECUTE msdb.dbo.rds_set_system_database_sync_objects @object_types = 'TempDbFile';
```

> **Important**
>
> Sebelum menggunakan prosedur `rds_set_system_database_sync_objects` tersimpan, pastikan Anda telah mengatur `tempdb` konfigurasi pilihan Anda pada instans DB utama Anda, bukan pada instans DB sekunder Anda. Jika Anda membuat perubahan

konfigurasi pada instans DB sekunder Anda, `tempdb` konfigurasi pilihan Anda dapat dihapus ketika Anda mengaktifkan sinkronisasi otomatis.

Anda dapat menggunakan fungsi berikut untuk mengonfirmasi apakah sinkronisasi otomatis `tempdb` konfigurasi diaktifkan:

```
SELECT * from msdb.dbo.rds_fn_get_system_database_sync_objects();
```

Ketika sinkronisasi otomatis `tempdb` konfigurasi dihidupkan, akan ada nilai pengembalian untuk `object_class` bidang tersebut. Ketika dimatikan, tidak ada nilai yang dikembalikan.

Anda dapat menggunakan fungsi berikut untuk menemukan objek terakhir kali disinkronkan, dalam waktu UTC:

```
SELECT * from msdb.dbo.rds_fn_server_object_last_sync_time();
```

Misalnya, jika Anda memodifikasi `tempdb` konfigurasi pada 01:00 dan kemudian menjalankan `rds_fn_server_object_last_sync_time` fungsi, nilai yang dikembalikan untuk `last_sync_time` harus setelah 01:00, menunjukkan bahwa sinkronisasi otomatis terjadi.

Jika Anda juga menggunakan replikasi pekerjaan SQL Server Agent, Anda dapat mengaktifkan replikasi untuk pekerjaan Agen SQL dan `tempdb` konfigurasi dengan menyediakannya dalam parameter: `@object_type`

```
EXECUTE msdb.dbo.rds_set_system_database_sync_objects @object_types =
 'SQLAgentJob,TempDbFile';
```

Untuk informasi selengkapnya tentang replikasi pekerjaan SQL Server Agent, lihat. Mengaktifkan replikasi pekerjaan SQL Server Agent

Sebagai alternatif untuk menggunakan prosedur `rds_set_system_database_sync_objects` tersimpan untuk memastikan bahwa perubahan `tempdb` konfigurasi disinkronkan secara otomatis, Anda dapat menggunakan salah satu metode manual berikut:

> **Note**
>
> Kami merekomendasikan untuk mengaktifkan sinkronisasi otomatis `tempdb` konfigurasi dengan menggunakan prosedur yang `rds_set_system_database_sync_objects`

disimpan. Menggunakan sinkronisasi otomatis mencegah kebutuhan untuk melakukan tugas-tugas manual ini setiap kali Anda mengubah `tempdb` konfigurasi Anda.

- Pertama, ubah instans DB Anda dan matikan Multi-AZ, kemudian modifikasi tempdb, dan terakhir aktifkan kembali Multi-AZ. Metode ini tidak melibatkan waktu henti apa pun.

  Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.
- Perubahan pertama `tempdb` dalam instans primer asli, kemudian gagal secara manual, dan terakhir memodifikasi `tempdb` di instans utama baru. Metode ini melibatkan waktu henti.

  Untuk informasi selengkapnya, lihat Mem-boot ulang instans DB.

Menganalisis beban kerja basis data di instans DB Amazon RDS for SQL Server dengan basis data Engine Tuning Advisor

Basis data Engine Tuning Advisor adalah aplikasi klien yang disediakan oleh Microsoft yang menganalisis beban kerja basis data dan merekomendasikan serangkaian indeks optimal untuk basis data Microsoft SQL Server berdasarkan jenis kueri yang Anda jalankan. Seperti SQL Server Management Studio, Anda menjalankan Tuning Advisor dari komputer klien yang terhubung ke instans DB Amazon RDS yang menjalankan SQL Server. Komputer klien dapat berupa komputer lokal yang Anda jalankan di lokasi on-premise Anda sendiri atau dapat berupa instans Windows Amazon EC2 yang berjalan di wilayah yang sama dengan instans DB Amazon RDS Anda.

Bagian ini menunjukkan cara mengambil beban kerja Tuning Advisor yang akan dianalisis. Ini adalah proses yang dipilih untuk mengambil beban kerja karena Amazon RDS membatasi akses host ke instans SQL Server. Untuk informasi selengkapnya, lihat Database Engine Tuning Advisor dalam dokumentasi Microsoft.

Untuk menggunakan Tuning Advisor, Anda harus menyediakan beban kerja ke advisor tersebut. Beban kerja adalah sekumpulan pernyataan Transact-SQL yang dijalankan di basis data atau basis data yang ingin Anda selaraskan. Basis data Engine Tuning Advisor menggunakan file pelacakan, tabel pelacakan, skrip Transact-SQL, atau file XML sebagai input beban kerja saat mengatur basis data. Ketika bekerja dengan Amazon RDS, beban kerja dapat berupa file di komputer klien atau tabel basis data di DB Amazon RDS for SQL Server yang dapat diakses oleh komputer klien. File atau tabel harus berisi kueri dari basis data yang ingin disetel dalam format yang sesuai untuk diputar ulang.

Untuk mengoptimalkan Tuning Advisor, beban kerja harus serealistis mungkin. Anda dapat membuat file atau tabel beban kerja dengan melakukan pelacakan terhadap instans DB Anda. Saat pelacakan berjalan, Anda dapat menyimulasikan beban di instans DB atau menjalankan aplikasi dengan beban normal.

Ada dua jenis pelacakan: sisi klien dan sisi server. Pelacakan sisi klien lebih mudah diatur dan Anda dapat menonton pelacakan peristiwa yang direkam secara waktu nyata dalam SQL Server Profiler. Pelacakan sisi server lebih rumit untuk disiapkan dan memerlukan beberapa skrip Transact-SQL. Selain itu, karena pelacakan ditulis ke file di instans DB Amazon RDS, ruang penyimpanan akan digunakan oleh pelacakan. Ini penting untuk melacak berapa banyak ruang penyimpanan yang digunakan oleh pelacakan sisi server yang berjalan karena instans DB dapat memasuki status penuh penyimpanan dan tidak akan lagi tersedia jika ruang penyimpanan habis.

Untuk pelacakan sisi klien, ketika jumlah data pelacakan yang cukup telah diambil di SQL Server Profiler, Anda kemudian dapat membuat file beban kerja dengan menyimpan pelacakan ke file di komputer lokal atau di tabel basis data di instans DB yang tersedia untuk komputer klien. Kekurangan utama dari penggunaan pelacakan sisi klien adalah bahwa pelacakan mungkin tidak mengambil semua kueri saat berada dalam beban berat. Hal ini dapat melemahkan efektivitas analisis yang dilakukan oleh basis data Engine Tuning Advisor. Jika Anda perlu menjalankan pelacakan dalam beban berat dan ingin memastikan bahwa pelacakan dapat menangkap setiap kueri selama sesi pelacakan, Anda harus menggunakan pelacakan sisi server.

Untuk pelacakan sisi server, Anda harus mendapatkan file pelacakan di instans DB ke dalam file beban kerja yang sesuai atau Anda dapat menyimpan pelacakan ke tabel di instans DB setelah pelacakan selesai. Anda dapat menggunakan SQL Server Profiler untuk menyimpan pelacakan ke file di komputer lokal atau membuat Tuning Advisor membaca dari tabel pelacakan di instans DB.

Menjalankan pelacakan sisi klien di instans DB SQL Server

Untuk menjalankan pelacakan sisi klien di instans DB SQL Server

1. Mulai SQL Server Profiler. SQL Server Profiler diinstal di folder Performance Tools dari folder instans SQL Server. Anda harus memuat atau menentukan templat definisi pelacakan untuk memulai pelacakan sisi klien.
2. Di menu SQL Server Profiler File, pilih Pelacakan Baru. Di kotak dialog Hubungkan ke Server, masukkan titik akhir instans DB, port, nama pengguna utama, dan kata sandi basis data yang ingin Anda lacak.
3. Di kotak dialog Properti Pelacakan, masukkan nama pelacakan dan pilih templat definisi pelacakan. Templat default, TSQL_Replay, dikirimkan bersama aplikasi. Anda dapat mengedit

templat ini untuk menentukan pelacakan Anda. Edit peristiwa dan informasi peristiwa di bagian tab Pemilihan Peristiwa dari kotak dialog Properti Pelacakan.

Untuk informasi templat definisi pelacakan dan cara menggunakan SQL Server Profiler untuk menentukan pelacakan sisi klien selengkapnya, lihat Database Engine Tuning Advisor dalam dokumentasi Microsoft.

4. Mulai pelacakan sisi klien dan perhatikan kueri SQL secara waktu nyata saat berjalan di instans DB Anda.
5. Pilih Hentikan Pelacakan dari menu File setelah Anda menyelesaikan pelacakan. Simpan hasil sebagai file atau sebagai tabel pelacakan di instans DB.

Menjalankan pelacakan sisi server di instans DB SQL Server

Menulis skrip untuk membuat pelacakan sisi server bisa menjadi rumit dan tidak dicakup dalam dokumen ini. Bagian ini berisi contoh skrip yang dapat Anda gunakan sebagai contoh. Seperti halnya pelacakan sisi klien, tujuan skrip ini adalah untuk membuat file beban kerja atau tabel pelacakan yang dapat Anda buka menggunakan basis data Engine Tuning Advisor.

Berikut ini adalah contoh skrip singkat yang memulai pelacakan sisi server dan mengambil detail ke file beban kerja. Pelacakan pada mulanya menyimpan ke file RDSTrace.trc di direktori D:\RDSDBDATA\Log dan bergulir setiap 100 MB sehingga file pelacakan berikutnya diberi nama RDSTrace_1.trc, RDSTrace_2.trc, dst.

```
DECLARE @file_name NVARCHAR(245) = 'D:\RDSDBDATA\Log\RDSTrace';
DECLARE @max_file_size BIGINT = 100;
DECLARE @on BIT = 1
DECLARE @rc INT
DECLARE @traceid INT

EXEC @rc = sp_trace_create @traceid OUTPUT, 2, @file_name, @max_file_size
IF (@rc = 0) BEGIN
   EXEC sp_trace_setevent @traceid, 10, 1, @on
   EXEC sp_trace_setevent @traceid, 10, 2, @on
   EXEC sp_trace_setevent @traceid, 10, 3, @on
 . . .
   EXEC sp_trace_setfilter @traceid, 10, 0, 7, N'SQL Profiler'
   EXEC sp_trace_setstatus @traceid, 1
   END
```

Contoh berikut adalah skrip yang menghentikan pelacakan. Perhatikan bahwa pelacakan yang dibuat oleh skrip sebelumnya terus berjalan hingga Anda secara eksplisit menghentikan pelacakan atau proses tidak berjalan di ruang disk.

```
DECLARE @traceid INT
SELECT @traceid = traceid FROM ::fn_trace_getinfo(default)
WHERE property = 5 AND value = 1 AND traceid <> 1

IF @traceid IS NOT NULL BEGIN
   EXEC sp_trace_setstatus @traceid, 0
   EXEC sp_trace_setstatus @traceid, 2
END
```

Anda dapat menyimpan hasil pelacakan sisi server ke tabel basis data dan menggunakan tabel basis data sebagai beban kerja untuk Tuning Advisor menggunakan fungsi fn_trace_gettable. Perintah berikut memuat hasil dari semua file bernama RDSTrace.trc di direktori D:\rdsdbdata\Log, termasuk semua file rollover, seperti RDSTrace_1.trc, ke dalam tabel bernama RDSTrace dalam basis data saat ini.

```
SELECT * INTO RDSTrace
FROM fn_trace_gettable('D:\rdsdbdata\Log\RDSTrace.trc', default);
```

Untuk menyimpan file rollover tertentu ke tabel, misalnya file RDSTrace_1.trc, tentukan nama file rollover dan pengganti 1, bukan default, sebagai parameter terakhir untuk fn_trace_gettable.

```
SELECT * INTO RDSTrace_1
FROM fn_trace_gettable('D:\rdsdbdata\Log\RDSTrace_1.trc', 1);
```

Menjalankan Tuning Advisor dengan pelacakan

Setelah Anda membuat pelacakan, baik sebagai file lokal maupun sebagai tabel basis data, Anda kemudian dapat menjalankan Tuning Advisor pada instans DB Anda. Menggunakan Tuning Advisor dengan Amazon RDS adalah proses yang sama seperti ketika bekerja dengan instans SQL Server jarak jauh mandiri. Anda dapat menggunakan Tuning Advisor UI di mesin klien atau menggunakan utilitas dta.exe dari baris perintah. Dalam kedua kasus, Anda harus terhubung ke instans DB Amazon RDS menggunakan titik akhir instans DB serta memberikan nama pengguna utama dan kata sandi pengguna utama Anda saat menggunakan Tuning Advisor.

Contoh kode berikut menunjukkan penggunaan utilitas baris perintah dta.exe di instans DB Amazon RDS dengan titik akhir **`dta.cnazcmklsdei.us-east-1.rds.amazonaws.com`**. Contohnya

mencakup nama pengguna utama **admin** dan kata sandi pengguna utama **test**, basis data contoh yang akan disetel diberi nama mesin bernama **C:\RDSTrace.trc**. Kode baris perintah contoh juga menentukan sesi pelacakan bernama **RDSTrace1** dan menentukan file output ke mesin lokal bernama **RDSTrace.sql** untuk skrip output SQL, **RDSTrace.txt** untuk file hasil, dan **RDSTrace.xml** untuk file XML analisis. Terdapat pula tabel kesalahan yang ditentukan di basis data RDSDTA bernama **RDSTraceErrors**.

```
dta -S dta.cnazcmklsdei.us-east-1.rds.amazonaws.com -U admin -P test -D RDSDTA -
if C:\RDSTrace.trc -s RDSTrace1 -of C:\ RDSTrace.sql -or C:\ RDSTrace.txt -ox C:\
 RDSTrace.xml -e RDSDTA.dbo.RDSTraceErrors
```

Berikut adalah contoh kode baris perintah yang sama kecuali beban kerja input merupakan tabel di instans Amazon RDS jarak jauh yang disebut **RDSTrace** dan berada di basis data **RDSDTA**.

```
dta -S dta.cnazcmklsdei.us-east-1.rds.amazonaws.com -U admin -P test -D RDSDTA -it
 RDSDTA.dbo.RDSTrace -s RDSTrace1 -of C:\ RDSTrace.sql -or C:\ RDSTrace.txt -ox C:\
 RDSTrace.xml -e RDSDTA.dbo.RDSTraceErrors
```

Untuk daftar lengkap parameter baris perintah utilitas dta, lihat dta Utility dalam dokumentasi Microsoft.

Mengubah akun **db_owner** menjadi **rdsa** untuk basis data Anda

Saat Anda membuat atau memulihkan basis data di instans DB RDS for SQL Server, Amazon RDS akan mengatur pemilik basis data `rdsa`. Jika Anda memiliki deployment Multi-AZ menggunakan SQL Server Database Mirroring (DBM) atau Always On Availability Groups (AG), Amazon RDS akan menetapkan pemilik basis data di instans DB sekunder ke `NT AUTHORITY\SYSTEM`. Pemilik basis data sekunder tidak dapat diubah sampai instans DB sekunder dinaikkan ke peran utama. Dalam kebanyakan kasus, menetapkan pemilik basis data ke `NT AUTHORITY\SYSTEM` tidak akan menimbulkan masalah saat menjalankan kueri, tetapi dapat menimbulkan kesalahan saat menjalankan prosedur tersimpan sistem seperti `sys.sp_updatestats` yang memerlukan izin lebih tinggi untuk dijalankan.

Anda dapat menggunakan kueri berikut untuk mengidentifikasi pemilik basis data yang dimiliki oleh `NT AUTHORITY\SYSTEM`:

```
SELECT name FROM sys.databases WHERE SUSER_SNAME(owner_sid) = 'NT AUTHORITY\SYSTEM';
```

Anda dapat menggunakan prosedur tersimpan Amazon RDS `rds_changedbowner_to_rdsa` untuk mengubah pemilik basis data `rdsa`. Basis data berikut tidak diizinkan untuk digunakan dengan `rds_changedbowner_to_rdsa`: `master, model, msdb, rdsadmin, rdsadmin_ReportServer, rdsadmin_ReportServerTempDB, SSISDB`.

Untuk mengubah pemilik database`rdsa`, panggil prosedur yang `rds_changedbowner_to_rdsa` disimpan dan berikan nama database.

Example Penggunaan:

```
exec msdb.dbo.rds_changedbowner_to_rdsa 'TestDB1';
```

Parameter berikut diperlukan:

- `@db_name` – Nama basis data untuk mengubah pemilik basis data menjadi `rdsa`.

Kolasi dan set karakter untuk Microsoft SQL Server

SQL Server mendukung kolasi di berbagai tingkat. Anda mengatur kolasi server default saat membuat instans DB. Anda dapat menimpa kolasi di tingkat basis data, tabel, atau kolom.

Topik

- Kolasi tingkat server untuk Microsoft SQL Server
- Kolasi tingkat basis data untuk Microsoft SQL Server

Kolasi tingkat server untuk Microsoft SQL Server

Saat membuat instans DB Microsoft SQL Server, Anda dapat mengatur kolasi server yang ingin digunakan. Jika Anda tidak memilih kolasi yang berbeda, kolasi tingkat server akan di-default-kan ke SQL_Latin1_General_CP1_CI_AS. Kolasi server diterapkan secara default untuk semua basis data dan objek basis data.

> **Note**
>
> Anda tidak dapat mengubah kolasi saat memulihkan dari snapshot DB.

Saat ini, Amazon RDS mendukung kolasi server berikut:

Kolasi	Deskripsi
Arabic_CI_AS	Bahasa Arab, tidak peka huruf besar-kecil, peka aksen, tidak peka jenis kana, tidak peka lebar
Chinese_PRC_BIN2	Bahasa Mandarin RRT, urutan titik kode biner
Chinese_PRC_CI_AS	Bahasa Mandarin-RRT, peka huruf besar-kec il, peka aksen, tidak peka jenis kana, tidak peka lebar
Chinese_Taiwan_Stroke_CI_AS	Bahasa Mandarin-Taiwan-Gurat, peka huruf besar-kecil, peka aksen, tidak peka jenis kana, tidak peka lebar
Denmark_Norwegian_CI_AS	Bahasa Denmark-Norwegia, tidak peka huruf besar-kecil, peka aksen, tidak peka jenis kana, tidak peka lebar
Finnish_Swedish_CI_AS	Bahasa Finlandia, Swedia, dan Swedia (Finlandia), peka huruf besar-kecil, peka aksen, tidak peka jenis kana, tidak peka lebar
French_CI_AS	Bahasa Prancis, tidak peka huruf besar-kecil, peka aksen, tidak peka jenis kana, tidak peka lebar
Hebrew_BIN	Bahasa Ibrani, urutan biner
Hebrew_CI_AS	Bahasa Ibrani, tidak peka huruf besar-kecil, peka aksen, tidak peka jenis kana, tidak peka lebar
Japanese_BIN	Bahasa Jepang, urutan biner
Japanese_CI_AS	Bahasa Jepang, tidak peka huruf besar-kecil, peka aksen, tidak peka jenis kana, tidak peka lebar

Kolasi	Deskripsi
Japanese_CS_AS	Bahasa Jepang, peka huruf besar-kecil, peka aksen, tidak peka jenis kana, tidak peka lebar
Japanese_XJIS_140_CI_AS	Bahasa Jepang, tidak peka huruf besar-kecil, peka aksen, tidak peka jenis kana, tidak peka lebar, tidak peka variasi
Japanese_XJIS_140_CI_AS_KS_VSS	Bahasa Jepang, tidak peka huruf besar-kecil, peka aksen, peka jenis kana, tidak peka lebar, tidak peka pelengkap, peka variasi
Japanese_XJIS_140_CI_AS_VSS	Bahasa Jepang, tidak peka huruf besar-kecil, peka aksen, tidak peka jenis kana, tidak peka lebar, tidak peka pelengkap, peka variasi
Japanese_XJIS_140_CS_AS_KS_WS	Bahasa Jepang, peka huruf besar-kecil, peka aksen, peka jenis kana, peka lebar, tidak peka variasi
Korean_Wansung_CI_AS	Bahasa Korea-Wansung, tidak peka huruf besar-kecil, peka aksen, tidak peka jenis kana, tidak peka lebar
Latin1_General_100_BIN	Bahasa Latin1-Umum-100, urutan biner
Latin1_General_100_BIN2	Bahasa Latin1-Umum-100, urutan titik kode biner
Latin1_General_100_BIN2_UTF8	Bahasa Latin1-Umum-100, urutan titik kode biner, dienkodekan dengan UTF-8
Latin1_General_100_CI_AS	Bahasa Latin1-Umum-100, tidak peka huruf besar-kecil, peka aksen, tidak peka jenis kana, tidak peka lebar

Kolasi	Deskripsi
Latin1_General_100_CI_AS_SC_UTF8	Bahasa Latin1-Umum-100, peka huruf besar-kecil, peka aksen, karakter tambahan, dienkodekan dengan UTF-8
Latin1_General_BIN	Bahasa Latin1-Umum, urutan biner
Latin1_General_BIN2	Bahasa Latin1-Umum, urutan titik kode biner
Latin1_General_CI_AI	Bahasa Latin1-Umum, tidak peka huruf besar-kecil, tidak peka aksen, tidak peka jenis kana, tidak peka lebar
Latin1_General_CI_AS	Bahasa Latin1-Umum, tidak peka huruf besar-kecil, peka aksen, tidak peka jenis kana, tidak peka lebar
Latin1_General_CI_AS_KS	Bahasa Latin1-Umum, tidak peka huruf besar-kecil, peka aksen, peka jenis kana, tidak peka lebar
Latin1_General_CS_AS	Bahasa Latin1-Umum, peka huruf besar-kecil, peka aksen, tidak peka jenis kana, tidak peka lebar
Modern_Spanish_CI_AS	Bahasa Spanyol-Modern, tidak peka huruf besar-kecil, peka aksen, tidak peka jenis kana, tidak peka lebar
Polish_CI_AS	Bahasa Polandia, tidak peka huruf besar-kecil, peka aksen, tidak peka jenis kana, tidak peka lebar
SQL_1xCompat_CP850_CI_AS	Bahasa Latin1-Umum, tidak peka huruf besar-kecil, peka akses, tidak peka jenis kana tidak peka lebar untuk Unicode Data, SQL Server Sort Order 49 pada Code Page 850 untuk non-Unicode Data

Kolasi	Deskripsi
SQL_Latin1_General_CP1_CI_AI	Bahasa Latin1-Umum, tidak peka huruf besar-kecil, tidak peka akses, tidak peka jenis kana tidak peka lebar untuk Unicode Data, SQL Server Sort Order 54 pada Code Page 1252 untuk non-Unicode Data
SQL_Latin1_General_CP1_CI_AS (default)	Bahasa Latin1-Umum, tidak peka huruf besar-kecil, peka akses, tidak peka jenis kana tidak peka lebar untuk Unicode Data, SQL Server Sort Order 52 pada Code Page 1252 untuk non-Unicode Data
SQL_Latin1_General_CP1_CS_AS	Bahasa Latin1-Umum, peka huruf besar-kecil, peka akses, tidak peka jenis kana tidak peka lebar untuk Unicode Data, SQL Server Sort Order 51 pada Code Page 1252 untuk non-Unicode Data
SQL_Latin1_General_CP437_CI_AI	Bahasa Latin1-Umum, tidak peka huruf besar-kecil, tidak peka akses, tidak peka jenis kana tidak peka lebar untuk Unicode Data, SQL Server Sort Order 34 pada Code Page 437 untuk non-Unicode Data
SQL_Latin1_General_CP850_BIN	Bahasa Latin1-Umum, urutan biner untuk Unicode Data, SQL Server Sort Order 40 pada Code Page 850 untuk non-Unicode Data
SQL_Latin1_General_CP850_BIN2	Bahasa Latin1-Umum, urutan titik kode biner untuk Unicode Data, SQL Server Sort Order 40 pada Code Page 850 untuk non-Unicode Data

Kolasi	Deskripsi
SQL_Latin1_General_CP850_CI_AI	Bahasa Latin1-Umum, tidak peka huruf besar-kecil, tidak peka akses, tidak peka jenis kana tidak peka lebar untuk Unicode Data, SQL Server Sort Order 44 pada Code Page 850 untuk non-Unicode Data
SQL_Latin1_General_CP850_CI_AS	Bahasa Latin1-Umum, tidak peka huruf besar-kecil, peka akses, tidak peka jenis kana tidak peka lebar untuk Unicode Data, SQL Server Sort Order 42 pada Code Page 850 untuk non-Unicode Data
SQL_Latin1_General_CP1256_CI_AS	Bahasa Latin1-Umum, tidak peka huruf besar-kecil, peka akses, tidak peka jenis kana tidak peka lebar untuk Unicode Data, SQL Server Sort Order 146 pada Code Page 1256 untuk non-Unicode Data
Thai_CI_AS	Bahasa Thailand, tidak peka huruf besar-kecil, peka aksen, tidak peka jenis kana, tidak peka lebar
Turkish_CI_AS	Bahasa Turki, tidak peka huruf besar-kecil, peka aksen, tidak peka jenis kana, tidak peka lebar

Untuk memilih kolasi:

- Jika Anda menggunakan konsol Amazon RDS, saat membuat instans DB baru, pilih Konfigurasi tambahan, lalu masukkan kolasi di kolom Kolasi. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.
- Jika Anda menggunakan AWS CLI, gunakan opsi `--character-set-name` dengan perintah `create-db-instance` . Untuk informasi selengkapnya, lihat create-db-instance.
- Jika Anda menggunakan API Amazon RDS, gunakan parameter `CharacterSetName` dengan operasi `CreateDBInstance`. Untuk informasi selengkapnya, lihat CreateDBInstance.

Kolasi tingkat basis data untuk Microsoft SQL Server

Anda dapat mengubah kolasi default pada basis data, tabel, atau tingkat kolom dengan menimpa kolasi saat membuat basis data baru atau objek basis data. Misalnya, jika kolasi server bawaan Anda adalah SQL_Latin1_General_CP1_CI_AS, Anda dapat mengubahnya ke Mohawk_100_CI_AS untuk dukungan kolasi Mohawk. Bahkan argumen dalam kueri dapat diketik agar dapat menggunakan susunan yang berbeda jika perlu.

Misalnya, kueri berikut akan mengubah kolasi default bawaan untuk kolom AccountName menjadi Mohawk_100_CI_AS

```
CREATE TABLE [dbo].[Account]
 (
     [AccountID] [nvarchar](10) NOT NULL,
     [AccountName] [nvarchar](100) COLLATE Mohawk_100_CI_AS NOT NULL
 ) ON [PRIMARY];
```

Mesin DB Microsoft SQL Server mendukung Unicode berdasarkan jenis data NCHAR, NVARCHAR, dan NTEXT bawaan. Misalnya, jika Anda membutuhkan dukungan CJK, gunakan tipe data Unicode untuk penyimpanan karakter dan menimpa kolasi server default saat membuat basis data dan tabel Anda. Berikut adalah beberapa tautan dari Microsoft yang mencakup dukungan kolasi dan Unicode untuk SQL Server:

- Bekerja dengan kolasi
- Kolasi dan terminologi internasional
- Menggunakan kolasi SQL Server
- Pertimbangan internasional untuk aplikasi basis data dan mesin basis data

Membuat pengguna basis data

Anda dapat membuat pengguna basis data untuk instans DB Amazon RDS for Microsoft SQL Server. Gunakan aplikasi seperti SQL Server Management Suite (SSMS). Anda masuk ke instans DB sebagai pengguna utama yang dibuat saat membuat instans DB.

```
--Initially set context to master database
USE [master];
GO
```

```
--Create a server-level login named theirname with password theirpassword
CREATE LOGIN [theirname] WITH PASSWORD = 'theirpassword';
GO
--Set context to msdb database
USE [msdb];
GO
--Create a database user named theirname and link it to server-level login theirname
CREATE USER [theirname] FOR LOGIN [theirname];
GO
```

Untuk contoh menambahkan pengguna basis data ke peran, lihat Menambahkan pengguna ke peran SQL AgentUser .

> **Note**
>
> Jika mendapatkan kesalahan izin saat menambahkan pengguna, Anda dapat memulihkan hak istimewa dengan memodifikasi kata sandi pengguna utama instans DB. Untuk informasi selengkapnya, lihat Mengatur ulang kata sandi peran `db_owner`.

Menentukan model pemulihan untuk basis data Microsoft SQL Server

Di Amazon RDS, model pemulihan, periode retensi, dan status basis data ditautkan.

Penting bagi Anda untuk memahami konsekuensinya sebelum membuat perubahan pada salah satu pengaturan ini. Setiap pengaturan dapat memengaruhi pengaturan lain. Sebagai contoh:

- Jika Anda mengubah model pemulihan basis data ke SIMPLE atau BULK_LOGGED saat retensi cadangan diaktifkan, Amazon RDS akan mereset model pemulihan ke FULL dalam waktu lima menit. Perubahan ini juga akan mengakibatkan RDS mengambil snapshot instans DB.
- Jika Anda mengatur retensi cadangan ke `0` hari, RDS akan mengatur mode pemulihan ke SIMPLE.
- Jika Anda mengubah model pemulihan basis data dari SIMPLE ke opsi lain apa pun ketika penyimpanan cadangan diatur ke `0` hari, RDS akan mereset model pemulihan ke SIMPLE.

> **Important**
>
> Jangan sekali-kali mengubah model pemulihan pada instans Multi-AZ meskipun tampaknya Anda dapat melakukannya—misalnya menggunakan ALTER DATABASE. Oleh karena

itu, retensi cadangan dengan mode pemulihan FULL diperlukan untuk Multi-AZ. Jika Anda mengubah model pemulihan, RDS akan segera mengubahnya kembali menjadi FULL. Reset otomatis ini memaksa RDS untuk membangun kembali duplikasi sepenuhnya. Selama pembangunan kembali ini, ketersediaan basis data akan menurun selama 30-90 menit hingga duplikasi siap untuk failover. Instans DB juga akan mengalami penurunan performa dengan cara yang sama seperti selama konversi dari Single-AZ ke Multi-AZ. Lamanya performa terdegradasi bergantung pada ukuran penyimpanan basis data—makin besar basis data tersimpan, makin lama degradasi.

Untuk informasi model pemulihan SQL Server selengkapnya, lihat Recovery models (SQL Server) dalam dokumentasi Microsoft.

Menentukan waktu failover terakhir

Untuk menentukan waktu failover terakhir, gunakan prosedur berikut:

```
execute msdb.dbo.rds_failover_time;
```

Prosedur ini akan menampilkan informasi berikut.

Parameter output	Deskripsi
errorlog_available_from	Menampilkan waktu sejak log kesalahan tersedia di direktori log.
recent_failover_time	Menampilkan waktu failover terakhir jika tersedia dari log kesalahan. Jika tidak, `null` akan ditampilkan.

> **Note**
>
> Prosedur tersimpan akan mencari semua log kesalahan SQL Server yang tersedia di direktori log untuk mengambil waktu failover terbaru. Jika pesan failover telah ditimpa oleh SQL Server, prosedur tidak akan mengambil waktu failover.

Example tanpa failover terbaru

Contoh ini menampilkan output saat tidak ada failover terbaru di log kesalahan. Tidak terjadi failover sejak 29-04-2020 23:59:00.01.

errorlog_available_from	recent_failover_time
29-04-2020 23:59:00.0100000	kosong

Example failover terbaru

Contoh ini menampilkan output saat tidak ada failover di log kesalahan. Failover terbaru adalah pada 05-05-2020 18:57:51.89.

errorlog_available_from	recent_failover_time
29-04-2020 23:59:00.0100000	05-05-2020 18:57:51.8900000

Menonaktifkan sisipan cepat selama pemuatan massal

Mulai dari SQL Server 2016, sisipan cepat diaktifkan secara default. Sisipan cepat memanfaatkan pencatatan log minimal yang terjadi saat basis data berada dalam model pemulihan pencatatan log sederhana atau massal untuk mengoptimalkan performa sisipan. Dengan sisipan cepat, setiap batch beban massal memperoleh luasan baru yang mengabaikan pencarian alokasi untuk luasan yang ada dengan ruang kosong yang tersedia untuk mengoptimalkan performa sisipan.

Namun, dengan sisipan cepat, muatan massal dengan ukuran batch kecil dapat menyebabkan bertambahnya ruang tidak terpakai yang digunakan objek. Jika peningkatan ukuran batch tidak memungkinkan, mengaktifkan tanda pelacakan 692 dapat membantu mengurangi ruang terpesan yang tidak digunakan, tetapi dengan mengorbankan performa. Mengaktifkan tanda pelacakan ini akan menonaktifkan sisipan cepat saat memuat data secara massal ke indeks berklaster atau heap.

Anda mengaktifkan tanda pelacakan 692 sebagai parameter startup menggunakan grup parameter DB. Untuk informasi selengkapnya, lihat Bekerja dengan grup parameter.

Tanda pelacakan 692 didukung untuk Amazon RDS di SQL Server 2016 dan versi yang lebih baru. Untuk informasi tanda pelacakan selengkapnya, lihat DBCC TRACEON - trace flags dalam dokumentasi Microsoft.

Menghapus sementara basis data Microsoft SQL Server

Anda dapat menghapus sementara basis data di instans DB Amazon RDS yang menjalankan Microsoft SQL Server dalam deployment Single-AZ atau Multi-AZ. Untuk menghapus sementara basis data, gunakan perintah berikut:

```
--replace your-database-name with the name of the database you want to drop
EXECUTE msdb.dbo.rds_drop_database  N'your-database-name'
```

> **Note**
>
> Gunakan kutipan tunggal lurus dalam perintah. Kutipan cerdas akan menyebabkan kesalahan.

Setelah Anda menggunakan prosedur ini untuk menghapus sementara basis data, Amazon RDS akan menghapus sementara semua koneksi yang ada ke basis data dan menghapus riwayat pencadangan basis data.

Mengganti nama basis data Microsoft SQL Server dalam deployment Multi-AZ

Untuk mengganti nama instans basis data Microsoft SQL Server yang menggunakan Multi-AZ, gunakan prosedur berikut:

1. Pertama, nonaktifkan Multi-AZ untuk instans DB.
2. Ganti nama basis data dengan menjalankan `rdsadmin.dbo.rds_modify_db_name`.
3. Kemudian, aktifkan Multi-AZ Mirroring atau Always On Availability Groups untuk instans DB agar dapat mengembalikannya ke kondisi semula.

Untuk informasi selengkapnya, lihat Menambahkan Multi-AZ ke instans DB Microsoft SQL Server.

> **Note**
>
> Jika instans Anda tidak menggunakan Multi-AZ, Anda tidak perlu mengubah pengaturan apa pun sebelum atau setelah menjalankan `rdsadmin.dbo.rds_modify_db_name`.

Contoh: Pada contoh berikut, prosedur tersimpan `rdsadmin.dbo.rds_modify_db_name` mengganti nama basis data dari **`MOO`** menjadi **`ZAR`**. Hal ini mirip dengan menjalankan pernyataan DDL `ALTER DATABASE [MOO] MODIFY NAME = [ZAR]`.

```
EXEC rdsadmin.dbo.rds_modify_db_name N'MOO', N'ZAR'
GO
```

Mengatur ulang kata sandi peran **`db_owner`**

Jika Anda mengunci diri dari peran `db_owner` di basis data Microsoft SQL Server, Anda dapat mereset kata sandi peran `db_owner` dengan memodifikasi kata sandi utama instans DB. Dengan mengubah kata sandi utama instans DB, Anda dapat memperoleh kembali akses ke instans DB, mengakses basis data menggunakan sandi yang telah dimodifikasi untuk `db_owner`, dan memulihkan hak istimewa untuk peran `db_owner` yang mungkin secara tidak sengaja dicabut. Anda dapat mengubah kata sandi instans DB menggunakan konsol Amazon RDS, perintah AWS CLI modify-db-instance, atau menggunakan operasi ModifyDBInstance. Untuk informasi cara mengubah instans DB selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Memulihkan instans DB yang telah dihentikan lisensinya

Microsoft telah meminta beberapa pelanggan Amazon RDS yang tidak melaporkan informasi Microsoft License Mobility mereka untuk menghentikan instans DB mereka. Amazon RDS mengambil snapshot dari instans DB ini, dan Anda dapat memulihkan dari snapshot ke instans DB baru yang memiliki model Termasuk Lisensi.

Anda dapat memulihkan dari snapshot Standar Edition ke Standard Edition atau Enterprise Edition.

Anda dapat memulihkan dari snapshot Enterprise Edition ke Standard Edition atau Enterprise Edition.

Untuk memulihkan dari snapshot SQL Server setelah Amazon RDS membuat snapshot akhir instans Anda

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Snapshot.
3. Pilih snapshot instans DB SQL Server. Amazon RDS akan membuat snapshot akhir instans DB. Nama snapshot instans yang telah dihentikan ada dalam format *instance_name*`-final-snapshot`. Misalnya, jika nama instans DB Anda adalah **`mytest.cdxgahslksma.us-east-1.rds.com`**, snapshot terakhir disebut **`mytest-final-snapshot`** dan terletak di Wilayah AWS yang sama dengan instans DB asli.
4. Untuk Tindakan, pilih Pulihkan Snapshot.

   Jendela Pulihkan Instans DB akan muncul.
5. Untuk Model Lisensi, pilih termasuk lisensi.
6. Pilih mesin DB SQL Server yang ingin Anda gunakan.
7. Untuk Pengidentifikasi Instans DB, masukkan nama instans DB yang dipulihkan.
8. Pilih Pulihkan Instans DB.

Untuk informasi cara memulihkan dari snapshot selengkapnya, lihat Memulihkan dari snapshot DB.

Melakukan transisi basis data Microsoft SQL Server dari OFFLINE ke ONLINE

Anda dapat bertransisi dari basis data Microsoft SQL Server di instans DB Amazon RDS dari `OFFLINE` ke `ONLINE`.

Metode SQL Server	Metode Amazon RDS
ALTER DATABASE *db_name* SET ONLINE;	EXEC rdsadmin.dbo.rds_set_database_online *db_name*

Menggunakan pengambilan data perubahan

Amazon RDS mendukung pengambilan data perubahan (CDC) untuk instans DB yang menjalankan Microsoft SQL Server. CDC merekam perubahan yang dilakukan terhadap data di tabel. CDC akan menyimpan metadata setiap perubahan, dan Anda dapat mengaksesnya nanti. Untuk informasi cara kerja CDC selengkapnya, lihat Change data capture dalam dokumentasi Microsoft.

Sebelum menggunakan CDC dengan instans DB Amazon RDS, aktifkan CDC di basis data dengan menjalankan `msdb.dbo.rds_cdc_enable_db`. Anda harus memiliki hak istimewa pengguna utama untuk mengaktifkan CDC di instans DB Amazon RDS. Setelah CDC diaktifkan, setiap pengguna yang merupakan `db_owner` basis data tersebut dapat mengaktifkan atau menonaktifkan CDC pada tabel di basis data tersebut.

> **Important**
>
> Selama pemulihan, CDC akan dinonaktifkan. Semua metadata terkait akan otomatis dihapus dari basis data. Hal ini berlaku untuk pemulihan snapshot, pemulihan point-in-time, dan pemulihan SQL Server Native dari S3. Setelah melakukan salah satu jenis pemulihan ini, Anda dapat mengaktifkan ulang CDC dan menetapkan ulang tabel untuk melacak.

Untuk mengaktifkan CDC instans DB, jalankan prosedur tersimpan `msdb.dbo.rds_cdc_enable_db`.

```
exec msdb.dbo.rds_cdc_enable_db 'database_name'
```

Untuk menonaktifkan CDC instans DB, jalankan prosedur tersimpan `msdb.dbo.rds_cdc_disable_db`.

```
exec msdb.dbo.rds_cdc_disable_db 'database_name'
```

Topik

- Melacak tabel menggunakan pengambilan data perubahan
- Pekerjaan pengambilan data perubahan
- Pengambilan data perubahan untuk instans Multi-AZ

Melacak tabel menggunakan pengambilan data perubahan

Setelah CDC diaktifkan di basis data, Anda dapat mulai melacak tabel tertentu. Anda dapat memilih tabel yang akan dilacak dengan menjalankan sys.sp_cdc_enable_table.

```
--Begin tracking a table
exec sys.sp_cdc_enable_table
   @source_schema           = N'source_schema'
,  @source_name             = N'source_name'
,  @role_name               = N'role_name'

--The following parameters are optional:

--, @capture_instance       = 'capture_instance'
--, @supports_net_changes   = supports_net_changes
--, @index_name             = 'index_name'
--, @captured_column_list   = 'captured_column_list'
--, @filegroup_name         = 'filegroup_name'
--, @allow_partition_switch = 'allow_partition_switch'
;
```

Untuk melihat konfigurasi CDC untuk tabel Anda, jalankan sys.sp_cdc_help_change_data_capture.

```
--View CDC configuration
exec sys.sp_cdc_help_change_data_capture

--The following parameters are optional and must be used together.
--  'schema_name', 'table_name'
;
```

Untuk informasi tabel CDC, fungsi, dan prosedur tersimpan dalam dokumentasi SQL Server selengkapnya, lihat bagian berikut:

- Prosedur tersimpan pengambilan data perubahan (Transact-SQL)
- Fungsi pengambilan data perubahan (Transact-SQL)
- Tabel pengambilan data perubahan (Transact-SQL)

Pekerjaan pengambilan data perubahan

Saat Anda mengaktifkan CDC, SQL Server akan membuat pekerjaan CDC. Pemilik basis data (`db_owner`) dapat melihat, membuat, memodifikasi, dan menghapus pekerjaan CDC. Namun, akun sistem RDS tetap memilikinya. Oleh karena itu, pekerjaan tidak akan terlihat dari tampilan native, prosedur, atau di SQL Server Management Studio.

Untuk mengendalikan perilaku CDC di basis data, gunakan prosedur SQL Server native, seperti sp_cdc_enable_table dan sp_cdc_start_job. Untuk mengubah parameter pekerjaan CDC, seperti `maxtrans` dan `maxscans`, Anda dapat menggunakan sp_cdc_change_job.

Untuk mendapatkan informasi pekerjaan CDC selengkapnya, Anda dapat mengueri tampilan pengelolaan dinamis berikut:

- sys.dm_cdc_errors
- sys.dm_cdc_log_scan_sessions
- sysjobs
- sysjobhistory

Pengambilan data perubahan untuk instans Multi-AZ

Jika Anda menggunakan CDC di instans Multi-AZ, pastikan konfigurasi pekerjaan CDC duplikat cocok dengan duplikat yang ada di pengguna utama. Pekerjaan CDC dipetakan ke `database_id`. Jika ID basis data di server sekunder berbeda dengan yang ada di pengguna utama, pekerjaan tidak akan dikaitkan dengan basis data yang benar. Untuk mencoba mencegah kesalahan setelah failover, RDS akan menghapus dan membuat ulang pekerjaan di pengguna utama baru. Pekerjaan yang dibuat ulang akan menggunakan parameter yang dicatat oeh pengguna utama sebelum failover.

Meskipun proses ini berjalan cepat, pekerjaan CDC masih mungkin berjalan sebelum RDS dapat memperbaikinya. Berikut tiga cara untuk memaksa parameter agar konsisten antara replika utama dan sekunder:

- Gunakan parameter pekerjaan yang sama untuk semua basis data yang mengaktifkan CDC.
- Sebelum Anda mengubah konfigurasi pekerjaan CDC, konversi instans Multi-AZ menjadi Single-AZ.
- Transfer parameter secara manual setiap kali Anda mengubahnya di pengguna utama.

Untuk melihat dan menentukan parameter CDC yang digunakan untuk membuat ulang pekerjaan CDC setelah failover, gunakan `rds_show_configuration` dan `rds_set_configuration`.

Contoh berikut menampilkan nilai yang diatur ke `cdc_capture_maxtrans`. Untuk parameter yang diatur ke `RDS_DEFAULT`, RDS akan otomatis mengonfigurasi nilai tersebut.

```
-- Show configuration for each parameter on either primary and secondary replicas.
exec rdsadmin.dbo.rds_show_configuration 'cdc_capture_maxtrans';
```

Untuk mengatur konfigurasi pada di server sekunder, jalankan `rdsadmin.dbo.rds_set_configuration`. Prosedur ini mengatur nilai parameter untuk semua basis data di server sekunder. Pengaturan ini hanya digunakan setelah failover. Contoh berikut mengatur `maxtrans` untuk semua pekerjaan pengambilan CDC ke *`1000`*:

```
--To set values on secondary. These are used after failover.
exec rdsadmin.dbo.rds_set_configuration 'cdc_capture_maxtrans', 1000;
```

Untuk mengatur parameter pekerjaan CDC di pengguna utama, gunakan sys.sp_cdc_change_job.

Menggunakan SQL Server Agent

Dengan Amazon RDS, Anda dapat menggunakan SQL Server Agent di instans DB yang menjalankan Microsoft SQL Server Enterprise Edition, Standard Edition, atau Web Edition. SQL Server Agent adalah layanan Microsoft Windows yang menjalankan tugas administratif terjadwal, yang disebut pekerjaan. Anda dapat menggunakan SQL Server Agent untuk menjalankan pekerjaan T-SQL untuk membangun ulang indeks, menjalankan pemeriksaan kerusakan, dan mengumpulkan data agregat dalam instans DB SQL Server.

Saat Anda membuat instans DB SQL Server, pengguna utama didaftarkan dalam peran `SQLAgentUserRole`.

SQL Server Agent dapat menjalankan pekerjaan sesuai jadwal, sebagai respons terhadap kejadian tertentu, atau sesuai permintaan. Untuk informasi selengkapnya, lihat SQL Server Agent dalam dokumentasi Microsoft.

> **Note**
>
> Hindari menjadwalkan pekerjaan untuk dijalankan selama masa pemeliharaan dan pencadangan instans DB. Proses pemeliharaan dan pencadangan yang diluncurkan oleh AWS dapat mengganggu pekerjaan atau menyebabkan pekerjaan dibatalkan.

Dalam deployment Multi-AZ, pekerjaan SQL Server Agent direplikasi dari host utama ke host sekunder saat fitur replikasi pekerjaan diaktifkan. Untuk informasi selengkapnya, lihat Mengaktifkan replikasi pekerjaan SQL Server Agent.
Deployment multi-AZ memiliki batas 10.000 pekerjaan SQL Server Agent. Jika Anda membutuhkan batas yang lebih tinggi, minta peningkatan batas dengan menghubungiAWS Support . Buka halaman Pusat AWS Support, masuk jika perlu, lalu pilih Buat kasus. Pilih Peningkatan batas layanan. Lengkapi lalu kirimkan formulir.

Untuk melihat riwayat pekerjaan SQL Server Agent individu di SQL Server Management Studio (SSMS), buka Object Explorer, klik kanan pekerjaan, lalu pilih Lihat Riwayat.

Karena SQL Server Agent berjalan di host terkelola di instans DB, ada beberapa tindakan yang tidak didukung:

- Menjalankan pekerjaan replikasi dan menjalankan skrip baris perintah dengan menggunakan ActiveX, shell perintah Windows, atau Windows tidak didukung. PowerShell
- Anda tidak dapat memulai, menghentikan, atau memulai ulang SQL Server Agent secara manual.
- Pemberitahuan email melalui SQL Server Agent tidak tersedia dari instans DB.
- Peringatan dan operator SQL Server Agent tidak didukung.
- Menggunakan SQL Server Agent untuk membuat cadangan tidak didukung. Gunakan Amazon RDS untuk mencadangkan instans DB.

Mengaktifkan replikasi pekerjaan SQL Server Agent

Anda dapat mengaktifkan replikasi pekerjaan SQL Server Agent menggunakan prosedur tersimpan berikut:

```
EXECUTE msdb.dbo.rds_set_system_database_sync_objects @object_types = 'SQLAgentJob';
```

Anda dapat menjalankan prosedur tersimpan di semua versi SQL Server yang didukung oleh Amazon RDS for SQL Server. Pekerjaan dalam kategori berikut direplikasi:

- [Tidak Dikategorikan (Lokal)]
- [Tidak Dikategorikan (Multi-Server)]
- [Tidak Dikategorikan]
- Data Collector

- Database Engine Tuning Advisor
- Database Maintenance
- Full-Text

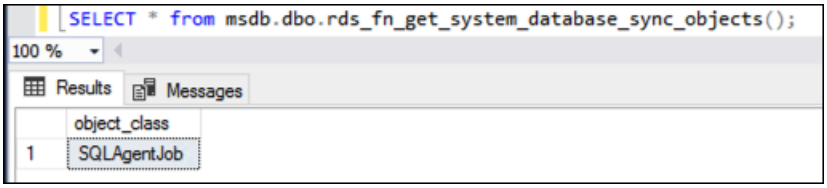
Hanya pekerjaan yang menggunakan langkah pekerjaan T-SQL yang akan direplikasi. Pekerjaan dengan tipe langkah seperti SQL Server Integration Services (SSIS), SQL Server Reporting Services (SSRS), Replikasi, dan tidak direplikasi. PowerShell Pekerjaan yang menggunakan Database Mail dan objek tingkat server tidak direplikasi.

> **Important**
>
> Host utama adalah sumber kebenaran untuk replikasi. Sebelum mengaktifkan replikasi pekerjaan, pastikan bahwa pekerjaan Agen SQL Server Anda berada di urutan pertama. Jika ini tidak dilakukan, pekerjaan Agen SQL Server Anda akan dihapus jika Anda mengaktifkan fitur saat pekerjaan yang lebih baru berada di host sekunder.

Anda dapat menggunakan fungsi berikut untuk mengonfirmasi apakah replikasi telah diaktifkan atau belum.

```
SELECT * from msdb.dbo.rds_fn_get_system_database_sync_objects();
```

Kueri T-SQL akan menampilkan hal berikut ini jika pekerjaan SQL Server Agent sedang mereplikasi. Jika pekerjaan SQL Server Agent sedang tidak mereplikasi, kueri T-SQL tidak akan menampilkan apa pun untuk `object_class`.

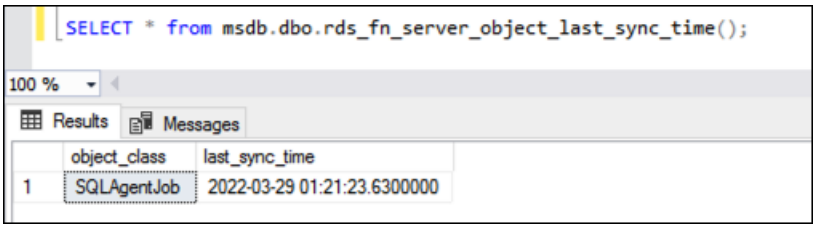


Anda dapat menggunakan fungsi berikut untuk mengetahui kapan terakhir kali objek disinkronkan dalam waktu UTC.

```
SELECT * from msdb.dbo.rds_fn_server_object_last_sync_time();
```

Misalnya, Anda memodifikasi pekerjaan SQL Server Agent pada 01:00. Anda mengharapkan waktu sinkronisasi terbaru setelah 01:00, yang menunjukkan bahwa sinkronisasi telah terjadi.

Setelah sinkronisasi, nilai yang ditampilkan untuk `date_created` dan `date_modified` pada simpul sekunder diperkirakan akan cocok.



Jika Anda juga menggunakan `tempdb` replikasi, Anda dapat mengaktifkan replikasi untuk pekerjaan Agen SQL dan `tempdb` konfigurasi dengan menyediakannya dalam parameter: `@object_type`

```
EXECUTE msdb.dbo.rds_set_system_database_sync_objects @object_types =
 'SQLAgentJob,TempDbFile';
```

Untuk informasi lebih lanjut tentang `tempdb` replikasi, lihatKonfigurasi TempDB untuk penerapan Multi-AZ.

Menambahkan pengguna ke peran SQL AgentUser

Untuk mengizinkan login atau pengguna tambahan menggunakan SQL Server Agent, masuk sebagai pengguna utama lalu lakukan hal berikut:

1. Buat login level server lainnya menggunakan perintah `CREATE LOGIN`.
2. Buat pengguna di `msdb` menggunakan perintah `CREATE USER`, lalu tautkan pengguna ini ke login yang telah Anda buat di langkah sebelumnya.
3. Tambahkan pengguna ke `SQLAgentUserRole` menggunakan prosedur `sp_addrolemember` yang disimpan sistem.

Misalnya, anggap nama pengguna utama Anda adalah **`admin`** dan Anda ingin memberikan akses ke SQL Server Agent kepada pengguna bernama **`theirname`** dengan kata sandi **`theirpassword`**. Dalam kasus ini, Anda dapat menggunakan prosedur berikut.

Untuk menambahkan pengguna ke peran SQL AgentUser

1. Masuk sebagai pengguna utama.
2. Jalankan perintah berikut:

```
--Initially set context to master database
```

```
USE [master];
GO
--Create a server-level login named theirname with password theirpassword
CREATE LOGIN [theirname] WITH PASSWORD = 'theirpassword';
GO
--Set context to msdb database
USE [msdb];
GO
--Create a database user named theirname and link it to server-level login
 theirname
CREATE USER [theirname] FOR LOGIN [theirname];
GO
--Added database user theirname in msdb to SQLAgentUserRole in msdb
EXEC sp_addrolemember [SQLAgentUserRole], [theirname];
```

Menghapus pekerjaan SQL Server Agent

Anda menggunakan prosedur tersimpan `sp_delete_job` untuk menghapus pekerjaan SQL Server Agent di Amazon RDS for Microsoft SQL Server.

Anda tidak dapat menggunakan SSMS untuk menghapus pekerjaan SQL Server Agent. Jika mencoba melakukannya, Anda akan mendapatkan pesan kesalahan yang serupa dengan pesan berikut ini:

```
The EXECUTE permission was denied on the object 'xp_regread', database
 'mssqlsystemresource', schema 'sys'.
```

Sebagai layanan terkelola, RDS dibatasi untuk tidak menjalankan prosedur yang mengakses registri Windows. Ketika Anda menggunakan SSMS untuk menghapus pekerjaan, SSMS akan mencoba menjalankan proses (`xp_regread`) yang tidak diotorisasi RDS.

> **Note**
>
> Di RDS for SQL Server, hanya anggota peran sysadmin yang diizinkan untuk memperbarui atau menghapus pekerjaan yang dimiliki oleh login lain.

Untuk menghapus pekerjaan SQL Server Agent

- Jalankan pernyataan T-SQL berikut:

```
EXEC msdb..sp_delete_job @job_name = 'job_name';
```

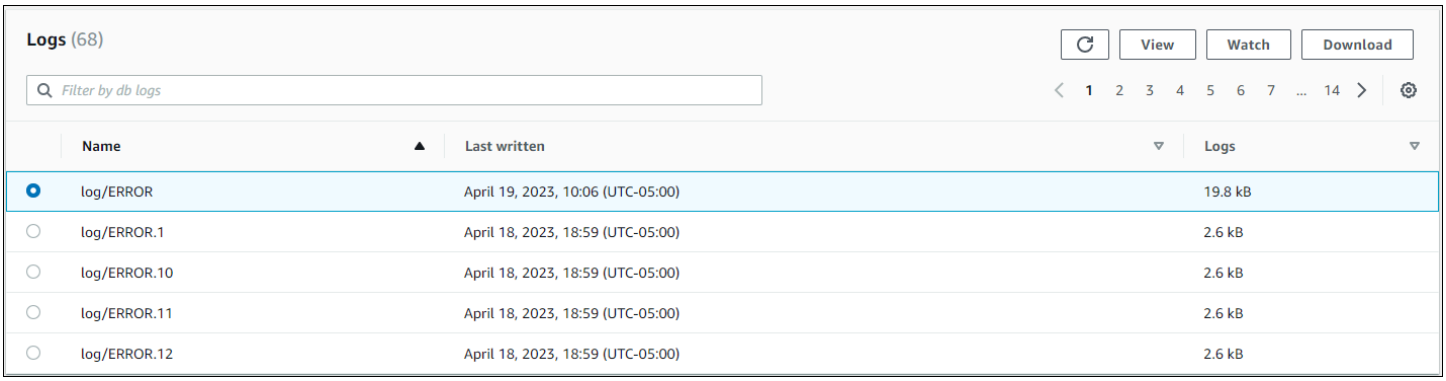
Bekerja dengan log Microsoft SQL Server

Anda dapat menggunakan konsol Amazon RDS untuk melihat, menonton, dan mengunduh log SQL Server Agent, log kesalahan Microsoft SQL Server, dan log SQL Server Reporting Services (SSRS).

Menonton file log

Jika melihat log di konsol Amazon RDS, Anda dapat melihatnya isinya sebagaimana adanya pada saat itu. Menonton log di konsol terbuka dalam keadaan dinamis sehingga Anda dapat melihat pembaruannya dalam waktu yang mendekati waktu nyata.

Hanya log terbaru yang aktif untuk ditonton. Misalnya, anggap Anda memiliki log berikut yang ditampilkan:



Hanya log/KESALAHAN, sebagai log terbaru, yang secara aktif diperbarui. Anda dapat memilih untuk menonton log lain, tetapi log lain bersifat statis dan tidak akan diperbarui.

Mengarsipkan file log

Konsol Amazon RDS menunjukkan log selama seminggu terakhir hingga hari ini. Anda dapat mengunduh dan mengarsipkan log untuk menyimpannya sebagai referensi selama waktu tersebut. Satu cara untuk mengarsipkan log adalah dengan memuatnya ke dalam bucket Amazon S3 Untuk petunjuk cara menyiapkan bucket Amazon S3 dan mengunggah file, lihat dasar-dasar Amazon S3 dalam Panduan Memulai Layanan Penyimpanan Sederhana Amazon lalu klik Mulai.

Melihat log kesalahan dan agen

Untuk melihat log kesalahan dan agen Microsoft SQL Server, gunakan prosedur `rds_read_error_log` yang disimpan oleh Amazon RDS dengan parameter berikut ini:

- **`@index`** – versi log yang akan diambil. Nilai default-nya adalah 0, yang mengambil log kesalahan saat ini. Tentukan 1 untuk mengambil log sebelumnya, tentukan 2 untuk mengambil log sebelum itu, dan selanjutnya.
- **`@type`** – jenis log yang akan diambil. Tentukan 1 untuk mengambil log kesalahan. Tentukan 2 untuk mengambil log agen.

Example

Contoh berikut meminta log kesalahan saat ini.

```
EXEC rdsadmin.dbo.rds_read_error_log @index = 0, @type = 1;
```

Untuk informasi kesalahan SQL Server selengkapnya, lihat Database engine errors dalam dokumentasi Microsoft.

Bekerja dengan file pelacakan dan dump

Bagian ini menjelaskan cara bekerja dengan file pelacakan dan file dump untuk instans DB Amazon RDS Anda yang menjalankan Microsoft SQL Server.

Membuat kueri SQL pelacakan

```
declare @rc int
declare @TraceID int
declare @maxfilesize bigint

set @maxfilesize = 5

exec @rc = sp_trace_create @TraceID output,  0, N'D:\rdsdbdata\log\rdstest',
 @maxfilesize, NULL
```

Melihat pelacakan terbuka

```
select * from ::fn_trace_getinfo(default)
```

Melihat konten pelacakan

```
select * from ::fn_trace_gettable('D:\rdsdbdata\log\rdstest.trc', default)
```

Mengatur periode retensi untuk file pelacakan dan dump

File pelacakan dan dump dapat terakumulasi dan menghabiskan ruang disk. Secara default, Amazon RDS akan menghapus file pelacakan dan dump yang telah melebihi tujuh hari.

Untuk melihat periode retensi file pelacakan dan dump saat ini, gunakan prosedur `rds_show_configuration`, sebagaimana ditunjukkan dalam contoh berikut.

```
exec rdsadmin..rds_show_configuration;
```

Untuk mengubah periode retensi file pelacakan, gunakan prosedur `rds_set_configuration` dan atur `tracefile retention` dalam menit. Contoh berikut akan mengatur periode retensi file pelacakan menjadi 24 jam.

```
exec rdsadmin..rds_set_configuration 'tracefile retention', 1440;
```

Untuk mengubah periode retensi file dump, gunakan prosedur `rds_set_configuration` dan atur `dumpfile retention` dalam menit. Contoh berikut mengatur periode retensi file dump menjadi 3 hari.

```
exec rdsadmin..rds_set_configuration 'dumpfile retention', 4320;
```

Untuk alasan keamanan, Anda tidak dapat menghapus file pelacakan atau dump tertentu di instans DB SQL Server. Untuk menghapus semua file pelacakan atau dump yang tidak digunakan, atur periode retensi file ke 0.

Amazon RDS for MySQL

Amazon RDS mendukung instans DB yang menjalankan versi MySQL berikut:

- MySQL 8.0
- MySQL 5.7

Untuk informasi selengkapnya tentang dukungan versi kecil, lihat Versi MySQL di Amazon RDS.

Untuk membuat instans DB Amazon RDS for MySQL, gunakan alat manajemen dan antarmuka Amazon RDS. Kemudian, Anda dapat melakukan hal berikut:

- Mengubah ukuran instans DB Anda
- Mengizinkan koneksi ke instance DB Anda
- Membuat dan memulihkan dari cadangan atau snapshot
- Membuat sekunder Multi-AZ
- Membuat replika baca
- Memantau performa instans DB Anda

Untuk menyimpan dan mengakses data di instans DB Anda, gunakan aplikasi dan utilitas MySQL standar.

Amazon RDS for MySQL mematuhi banyak standar industri. Misalnya, Anda dapat menggunakan basis data RDS for MySQL untuk membangun aplikasi yang mematuhi HIPAA. Anda dapat menggunakan basis data RDS untuk MySQL untuk menyimpan informasi terkait layanan kesehatan, termasuk informasi kesehatan yang dilindungi (PHI) berdasarkan Perjanjian Rekan Bisnis (BAA) dengan AWS. Amazon RDS for MySQL juga memenuhi persyaratan keamanan Federal Risk and Authorization Management Program (FedRAMP). Selain itu, Amazon RDS for MySQL telah menerima FedRAMP Joint Authorization Board (JAB) Provisional Authority to Operate (P-ATO) di FedRAMP HIGH Baseline di dalam Wilayah AWS GovCloud (US). Untuk informasi selengkapnya tentang standar kepatuhan yang didukung, lihat kepatuhan cloud AWS.

Untuk informasi tentang fitur-fitur di setiap versi MySQL, lihat Fitur utama MySQL dalam dokumentasi MySQL.

Sebelum membuat instans DB, selesaikan langkah-langkah di Menyiapkan Amazon RDS. Saat Anda membuat instans DB, pengguna master RDS mendapatkan hak istimewa DBA, dengan beberapa batasan. Gunakan akun ini untuk tugas administratif seperti membuat akun basis data tambahan.

Anda dapat membuat berikut ini:

- Instans DB
- Snapshot DB
- P oint-in-time mengembalikan
- Pencadangan otomatis
- Pencadangan manual

Anda dapat menggunakan instans DB yang menjalankan MySQL di dalam cloud privat virtual (VPC) berdasarkan Amazon VPC. Anda juga dapat menambahkan fitur ke instans DB MySQL Anda dengan mengaktifkan berbagai opsi. Amazon RDS mendukung deployment Multi-AZ untuk MySQL sebagai solusi failover dengan ketersediaan tinggi.

> **Important**
>
> Untuk memberikan pengalaman layanan terkelola, Amazon RDS tidak memberikan akses shell ke instans DB. Hal tersebut juga membatasi akses ke prosedur dan tabel sistem tertentu yang membutuhkan hak istimewa tingkat lanjut. Anda dapat mengakses basis data Anda menggunakan klien SQL standar seperti klien mysql. Namun, Anda tidak dapat mengakses host secara langsung dengan menggunakan Telnet atau Secure Shell (SSH).

Topik

- Dukungan fitur MySQL di Amazon RDS
- Versi MySQL di Amazon RDS
- Menghubungkan ke instans DB yang menjalankan mesin basis data MySQL
- Mengamankan koneksi instans DB MySQL
- Meningkatkan performa kueri RDS for MySQL dengan Amazon RDS Optimized Reads
- Meningkatkan performa penulisan dengan RDS Optimized Writes for MySQL
- Meng-upgrade mesin DB MySQL
- Meningkatkan versi mesin snapshot DB MySQL

- Impor data ke dalam instans DB MySQL
- Menggunakan replikasi MySQL di Amazon RDS
- Mengonfigurasikan klaster aktif-aktif untuk RDS for MySQL
- Mengekspor data dari instans DB MySQL dengan menggunakan replikasi
- Opsi untuk instans DB MySQL
- Parameter untuk MySQL
- Tugas umum DBA untuk instans DB MySQL
- Zona waktu lokal untuk instans DB MySQL
- Masalah umum dan batasan untuk Amazon RDS for MySQL
- RDS for MySQL

Dukungan fitur MySQL di Amazon RDS

RDS for MySQL mendukung sebagian besar fitur dan kemampuan MySQL. Beberapa fitur mungkin memiliki dukungan terbatas atau hak istimewa yang dibatasi.

Anda dapat memfilter fitur-fitur Amazon RDS baru pada halaman Apa yang Baru dengan Basis Data?. Untuk Produk, pilih Amazon RDS. Lalu, cari dengan menggunakan kata kunci seperti **MySQL 2022**.

> **Note**
>
> Berikut ini bukan daftar lengkap.

Topik

Mesin penyimpanan yang didukung untuk RDS for MySQL

Meskipun MySQL mendukung banyak mesin penyimpanan dengan berbagai kemampuan, tidak semuanya dioptimalkan untuk pemulihan kerusakan dan ketahanan data. Amazon RDS sepenuhnya mendukung mesin penyimpanan InnoDB untuk instans DB MySQL. Fitur Amazon RDS seperti pemulihan Titik Waktu dan pemulihan snapshot memerlukan mesin penyimpanan yang dapat dipulihkan dan hanya didukung untuk mesin penyimpanan InnoDB. Untuk informasi selengkapnya, lihat Dukungan memcached MySQL.

Mesin Penyimpanan Gabungan saat ini tidak didukung oleh Amazon RDS for MySQL.

Untuk skema yang dibuat pengguna, mesin penyimpanan MyISAM tidak mendukung pemulihan yang andal dan dapat menyebabkan kehilangan atau kerusakan data ketika MySQL memulai ulang setelah pemulihan, sehingga menghalangi pemulihan Titik Waktu atau pemulihan snapshot berjalan seperti yang diinginkan. Namun, jika Anda tetap memilih menggunakan MyISAM dengan Amazon RDS, snapshot dapat bermanfaat pada beberapa kondisi.

> **Note**
>
> Tabel sistem dalam skema `mysql` bisa berada dalam penyimpanan MyISAM.

Jika Anda ingin mengonversi tabel MyISAM yang ada untuk tabel InnoDB, Anda dapat menggunakan perintah `ALTER TABLE` (misalnya, `alter table TABLE_NAME engine=innodb;`). Harap diingat bahwa MyISAM dan InnoDB memiliki keunggulan dan kekurangan yang berbeda, jadi Anda harus sepenuhnya mengevaluasi dampak peralihan ini pada aplikasi Anda sebelum melakukannya.

MySQL 5.1, 5.5, dan 5.6 sudah tidak didukung di Amazon RDS. Namun, Anda dapat memulihkan snapshot MySQL 5.1, dan 5.6 yang sudah ada. Saat Anda memulihkan snapshot MySQL 5.1, 5.5, atau 5.6, instans DB secara otomatis ditingkatkan ke MySQL 5.7.

Menggunakan memcached dan opsi lain dengan MySQL di Amazon RDS

Sebagian besar mesin Amazon RDS DB mendukung grup opsi yang memungkinkan Anda memilih fitur tambahan untuk instans DB Anda. Instans DB RDS for MySQL mendukung opsi `memcached`, sebuah cache sederhana berbasis kunci. Untuk informasi selengkapnya tentang `memcached` dan opsi lainnya, lihat Opsi untuk instans DB MySQL. Untuk informasi selengkapnya tentang penggunaan grup opsi, lihat Menggunakan grup opsi.

Pemanasan cache InnoDB untuk MySQL di Amazon RDS

Pemanasan cache InnoDB dapat memberikan peningkatan kinerja untuk instans DB MySQL Anda dengan menyimpan status buffer pool saat ini ketika instans DB dimatikan, lalu memuat ulang buffer pool tersebut dari informasi tersimpan ketika instans DB dimulai. Dengan begitu, buffer pool tidak perlu melakukan “pemanasan” dari penggunaan basis data normal dan sebagai gantinya mengisi buffer pool di awal dengan halaman-halaman untuk kueri umum. File yang menyimpan informasi buffer pool yang disimpan hanya menyimpan metadata untuk halaman yang ada di dalam buffer pool, bukan halaman itu sendiri. Hasilnya, file tidak memerlukan banyak ruang penyimpanan. Ukuran filenya sekitar 0,2 persen dari ukuran cache. Misalnya, untuk cache 64 GiB, ukuran file pemanasan cache adalah 128 MiB. Untuk informasi selengkapnya tentang pemanasan cache, lihat Menyimpan dan memulihkan status buffer pool dalam dokumentasi MySQL.

Instans DB RDS for MySQL mendukung pemanasan cache InnoDB. Untuk mengaktifkan pemanasan cache InnoDB, atur parameter `innodb_buffer_pool_dump_at_shutdown` dan `innodb_buffer_pool_load_at_startup` ke 1 dalam grup parameter untuk instans DB Anda.

Mengubah nilai parameter-parameter ini di grup parameter akan memengaruhi semua instans DB MySQL yang menggunakan grup parameter tersebut. Untuk mengaktifkan pemanasan cache InnoDB bagi instans DB MySQL tertentu, Anda mungkin perlu membuat grup parameter baru untuk instans-instans tersebut. Untuk informasi tentang grup parameter, lihat Bekerja dengan grup parameter.

Pemanasan cache InnoDB terutama memberikan manfaat kinerja untuk instans DB yang menggunakan penyimpanan standar. Jika Anda menggunakan penyimpanan PIOPS, Anda biasanya tidak melihat peningkatan kinerja yang signifikan.

Important

Jika instans DB MySQL tidak mati secara normal, seperti saat failover, status buffer pool tidak akan disimpan ke disk. Dalam kasus ini, MySQL memuat file buffer pool apa pun yang tersedia saat instans DB dimulai ulang. Tidak ada kerugian yang timbul, tetapi buffer pool yang dipulihkan mungkin tidak mencerminkan status terbaru buffer pool sebelum mulai ulang. Untuk memastikan bahwa Anda memiliki status terbaru dari pool buffer yang tersedia untuk menyiapkan cache InnoDB saat startup, kami menyarankan Anda mencadangkan pool buffer secara berkala "sesuai permintaan".
Anda dapat membuat event untuk mencadangkan pool buffer secara otomatis dan pada interval rutin. Misalnya, pernyataan berikut membuat event bernama `periodic_buffer_pool_dump` yang mencadangkan pool buffer setiap jam.

```
CREATE EVENT periodic_buffer_pool_dump
ON SCHEDULE EVERY 1 HOUR
DO CALL mysql.rds_innodb_buffer_pool_dump_now();
```

Untuk informasi selengkapnya tentang peristiwa MySQL, lihat Sintaks peristiwa dalam dokumentasi MySQL.

Mencadangkan dan memuat pool buffer sesuai permintaan

Anda dapat menyimpan dan memuat cache InnoDB “sesuai permintaan”.

- Untuk mencadangkan status saat ini dari pool buffer ke disk, panggil prosedur tersimpan mysql.rds_innodb_buffer_pool_dump_now.
- Untuk memuatkan keadaan tersimpan kolam penyangga dari disk, panggil prosedur tersimpan mysql.rds_innodb_buffer_pool_load_now.

- Untuk membatalkan operasi pemuatan yang sedang berlangsung, panggil prosedur tersimpan mysql.rds_innodb_buffer_pool_load_abort.

Fitur MySQL yang tidak didukung oleh Amazon RDS

Amazon RDS saat ini tidak mendukung fitur-fitur MySQL berikut:

- Plugin Autentikasi
- Pencatatan Log Kesalahan ke Log Sistem
- Enkripsi Ruang Tabel InnoDB
- Plugin Kekuatan Kata Sandi
- Variabel sistem yang dipertahankan
- Plugin Tulis Ulang Kueri Penulis Ulang
- Replikasi semisinkron
- Ruang tabel yang dapat dipindahkan
- Plugin X

> **Note**
>
> ID transaksi global didukung untuk semua RDS for MySQL versi 5.7, dan untuk RDS for MySQL versi 8.0.26 dan 8.0 yang lebih tinggi.

Untuk memberikan pengalaman layanan terkelola, Amazon RDS tidak memberikan akses shell ke instans DB. Amazon RDS juga membatasi akses ke prosedur dan tabel sistem tertentu yang memerlukan hak istimewa tingkat lanjut. Amazon RDS mendukung akses ke basis data di instans DB dengan menggunakan aplikasi klien SQL standar. Amazon RDS tidak mengizinkan akses host langsung ke instans DB dengan menggunakan Telnet, Secure Shell (SSH), atau Windows Remote Desktop Connection. Saat membuat instans DB, Anda diberikan peran db_owner untuk semua basis data pada instans tersebut, dan Anda memiliki semua izin tingkat basis data kecuali yang digunakan untuk pencadangan. Amazon RDS mengelola pencadangan untuk Anda.

Versi MySQL di Amazon RDS

Untuk MySQL, nomor versi disusun dengan format versi = X.Y.Z. Dalam terminologi Amazon RDS, X.Y menunjukkan versi utama, dan Z adalah nomor versi kecil. Untuk implementasi Amazon RDS, perubahan versi dianggap utama jika nomor versi utamanya berubah–-misalnya, dari versi 5.7 menjadi 8.0. Perubahan versi dianggap kecil jika hanya nomor versi minor yang berubah—misalnya, dari versi 8.0.32 ke 8.0.34.

Topik

Versi kecil MySQL yang didukung di Amazon RDS

Amazon RDS saat ini mendukung versi kecil MySQL berikut.

> **Note**
>
> Tanggal yang berupa hanya bulan dan tahun merupakan perkiraan, dan akan diperbarui dengan tanggal persisnya saat diketahui.
> Amazon RDS Extended Support tidak tersedia untuk versi minor.

Versi mesin MySQL	Tanggal rilis komunitas	Tanggal rilis RDS	Tanggal akhir dukungan standar RDS
8.0			
8.0.36	16 Januari 2024	12 Februari 2024	Maret 2025

Versi mesin MySQL	Tanggal rilis komunitas	Tanggal rilis RDS	Tanggal akhir dukungan standar RDS
8.0.35	25 Oktober 2023	9 November 2023	Maret 2025
8.0.34	18 Juli 2023	9 Agustus 2023	September 2024
8.0.33	18 April 2023	15 Juni 2023	September 2024
8.0.32	17 Januari 2023	7 Februari 2023	September 2024
5.7			
5.7.44*	25 Oktober 2023	2 November 2023	29 Februari 2024

* Versi minor ini akan terus tersedia ketika versi utama ada di Amazon RDS Extended Support. Untuk informasi selengkapnya, lihat Menggunakan Dukungan Diperpanjang Amazon RDS.

Anda dapat menentukan versi MySQL mana pun yang saat ini didukung ketika membuat instans basis data baru. Anda dapat menentukan versi besar (seperti MySQL 5.7), dan versi kecil mana pun yang didukung untuk versi besar tersebut. Jika tidak ada versi yang ditentukan, Amazon RDS menetapkan default ke versi yang didukung, biasanya versi terbaru. Jika versi utama ditentukan tetapi versi kecil tidak, Amazon RDS menjadikan menetapkan default ke rilis versi utama terbaru yang telah Anda tentukan. Untuk melihat daftar versi yang didukung, serta default untuk instans DB yang baru dibuat, gunakan perintah. `describe-db-engine-versions` AWS CLI

Misalnya, untuk membuat daftar versi mesin yang didukung untuk RDS for MySQL, jalankan perintah CLI berikut:

```
aws rds describe-db-engine-versions --engine mysql --query "*[].
{Engine:Engine,EngineVersion:EngineVersion}" --output text
```

Versi MySQL default mungkin berbeda berdasarkan Wilayah AWS. Untuk membuat instans basis data dengan versi kecil tertentu, tentukan versi kecil selama pembuatan instans basis data. Anda dapat menentukan versi minor default untuk Wilayah AWS menggunakan AWS CLI perintah berikut:

```
aws rds describe-db-engine-versions --default-only --engine mysql
 --engine-version major-engine-version --region region --query "*[].
{Engine:Engine,EngineVersion:EngineVersion}" --output text
```

Ganti *major-engine-version*dengan versi mesin utama, dan ganti *wilayah* dengan Wilayah AWS. Misalnya, AWS CLI perintah berikut mengembalikan versi mesin minor MySQL default untuk versi utama 5.7 dan US West (Oregon Wilayah AWS) (us-barat-2):

```
aws rds describe-db-engine-versions --default-only --engine mysql --engine-version 5.7
 --region us-west-2 --query "*[].{Engine:Engine,EngineVersion:EngineVersion}" --output
 text
```

Dengan Amazon RDS, Anda dapat mengontrol kapan harus meningkatkan versi instans MySQL Anda ke versi utama baru yang didukung oleh Amazon RDS. Anda dapat mempertahankan kompatibilitas dengan versi MySQL tertentu, menguji versi baru dengan aplikasi Anda sebelum di-deploy di dalam produksi, dan melakukan peningkatan versi utama pada waktu-waktu yang paling pas dengan jadwal Anda.

Jika peningkatan versi kecil otomatis diaktifkan, instans DB Anda akan otomatis ditingkatkan ke versi MySQL kecil karena didukung oleh Amazon RDS. Proses patching ini terjadi selama periode pemeliharaan terjadwal Anda. Anda dapat memodifikasi instans DB untuk mengaktifkan atau menonaktifkan peningkatan versi kecil otomatis.

Jika Anda memilih untuk tidak melakukan peningkatan terjadwal otomatis, Anda dapat melakukan peningkatan manual ke rilis versi kecil yang didukung dengan mengikuti prosedur yang sama seperti untuk pembaruan versi utama. Untuk informasi, lihat Meng-upgrade versi mesin instans DB.

Amazon RDS saat ini mendukung peningkatan versi utama dari MySQL versi 5.6 menjadi versi 5.7, dan dari MySQL versi 5.7 menjadi versi 8.0. Karena peningkatan versi utama melibatkan beberapa risiko kompatibilitas, peningkatan tersebut tidak terjadi secara otomatis; Anda harus membuat permintaan untuk memodifikasi instans DB. Anda harus menguji peningkatan apa pun secara menyeluruh sebelum meningkatkan versi instans produksi Anda. Untuk informasi tentang peningkatan instans DB MySQL, lihat Meng-upgrade mesin DB MySQL.

Anda dapat menguji instans DB terhadap versi baru sebelum melakukan peningkatan versi dengan membuat snapshot DB dari instans DB yang sudah Anda miliki, memulihkan dari snapshot DB tersebut untuk membuat instans DB baru, kemudian memulai peningkatan versi untuk instans DB baru. Anda kemudian dapat melakukan eksperimen secara aman pada klon instans DB yang

ditingkatkan tersebut sebelum memutuskan apakah Anda akan meningkatkan versi instans DB asli Anda atau tidak.

Versi utama MySQL yang didukung di Amazon RDS

Versi utama RDS for MySQL tersedia di bawah dukungan standar setidaknya sampai berakhirnya siklus komunitas untuk versi komunitas yang sesuai. Anda dapat terus menjalankan versi utama melewati tanggal akhir dukungan standar RDS dengan biaya tertentu. Untuk informasi selengkapnya, lihat Menggunakan Dukungan Diperpanjang Amazon RDS dan harga Amazon RDS for MySQL.

Anda dapat menggunakan tanggal berikut untuk merencanakan siklus pengujian dan peningkatan Anda.

> **Note**
>
> Tanggal yang hanya berisi bulan dan tahun merupakan perkiraan, dan diperbarui dengan tanggal pasti saat diketahui.

Versi utama MySQL	Tanggal rilis komunitas	Tanggal rilis RDS	Tanggal akhir pemakaian komunitas	Tanggal akhir dukungan standar RDS	Tanggal dimulainya harga Dukungan yang Diperpanjang tahun 1 RDS	Tanggal mulai harga tahun ke-3 Dukungan yang Diperluas RDS	Tanggal akhir Dukungan yang Diperpanjang RDS
MySQL 8.0	19 April 2018	23 Oktober 2018	April 2026	31 Juli 2026	1 Agustus 2026	1 Agustus 2028	31 Juli 2029
MySQL 5.7*	21 Oktober 2015	22 Februari 2016	Oktober 2023	29 Februari 2024	1 Maret 2024	1 Maret 2026	28 Februari 2027

* MySQL 5.7 sekarang hanya tersedia di bawah RDS Extended Support. Untuk informasi selengkapnya, lihat Menggunakan Dukungan Diperpanjang Amazon RDS.

Menggunakan lingkungan Pratinjau Basis Data

Pada Juli 2023, Oracle mengumumkan model rilis baru untuk MySQL. Model ini mencakup dua jenis rilis: Rilis Inovasi dan rilis LTS. Amazon RDS membuat Rilis Inovasi MySQL tersedia di lingkungan Pratinjau RDS. Untuk mempelajari lebih lanjut tentang rilis Inovasi MySQL, lihat Introducing MySQL Innovation and Long-Term Support (LTS) versions.

Instans DB RDS for MySQL di lingkungan Pratinjau Basis Data secara fungsional mirip dengan instans DB RDS for MySQL lainnya. Namun, Anda tidak dapat menggunakan Lingkungan Pratinjau Basis Data untuk beban kerja produksi.

Lingkungan Pratinjau memiliki batasan berikut ini:

- Amazon RDS menghapus semua instans DB 60 hari setelah Anda membuatnya, termasuk semua cadangan dan snapshot.
- Anda hanya dapat menggunakan penyimpanan SSD Tujuan Umum dan IOPS yang Tersedia.
- Anda tidak bisa mendapatkan AWS Support bantuan dari instans DB. Sebagai gantinya, Anda dapat memposting pertanyaan Anda ke komunitas Tanya Jawab yang AWS dikelola, Re:post.AWS
- Anda tidak dapat menyalin snapshot instans DB ke lingkungan produksi.

Opsi berikut didukung oleh pratinjau.

- Anda dapat membuat instans DB menggunakan kelas instans DB db.m6i, db.r6i, db.m6g, db.m5, db.t3, db.r6g, dan db.r5. Untuk informasi selengkapnya tentang kelas instans RDS, lihat Kelas instans DB .
- Anda dapat menggunakan deployment AZ tunggal dan multi-AZ.
- Anda dapat menggunakan fungsi dump dan load MySQL standar untuk mengekspor basis data dari atau mengimpor basis data ke lingkungan Pratinjau Basis Data.

Fitur yang tidak didukung di lingkungan Pratinjau Basis Data

Fitur berikut ini tidak tersedia di lingkungan Pratinjau Basis Data:

- Salinan snapshot lintas Wilayah

- Replika baca lintas Wilayah

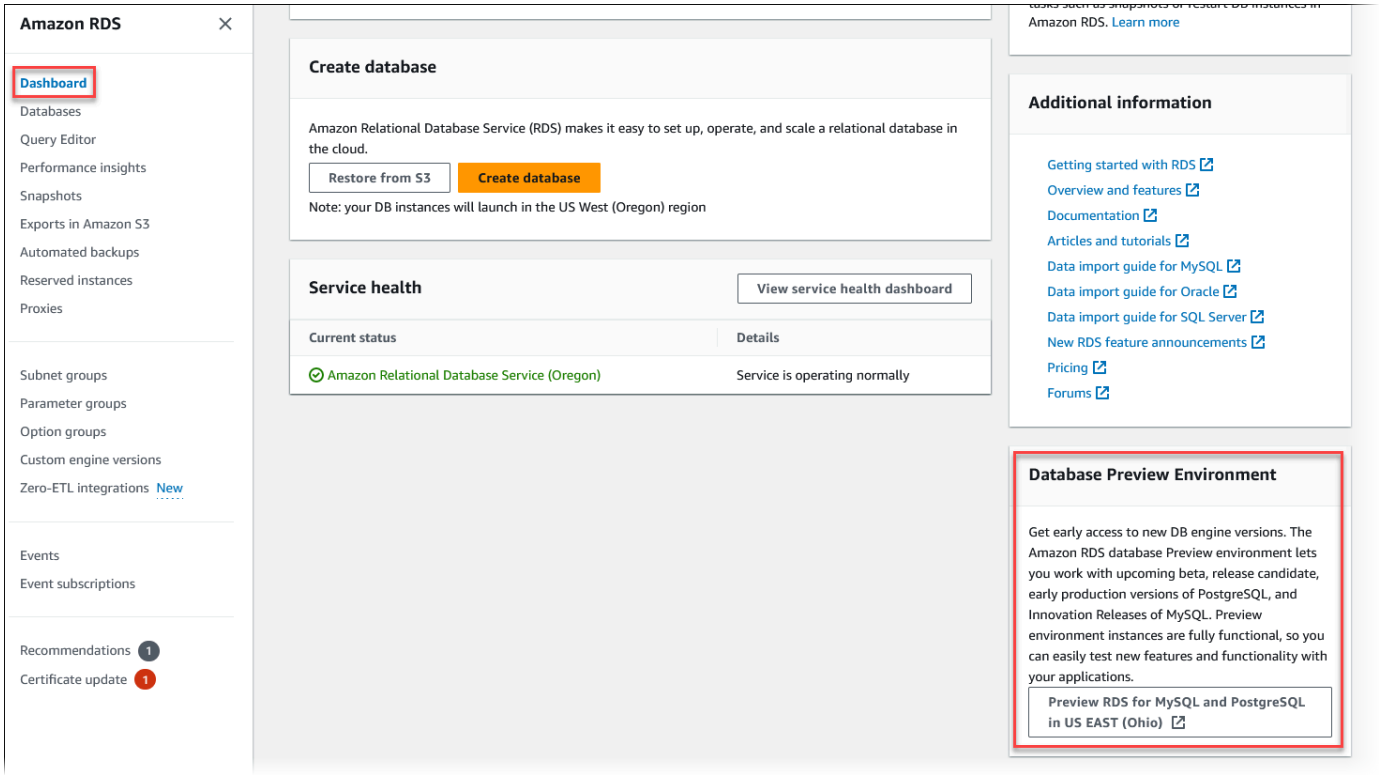
Membuat instans DB baru di Lingkungan Pratinjau Basis Data

Anda dapat membuat instans DB di lingkungan Pratinjau Database menggunakan AWS Management Console, AWS CLI, atau RDS API.

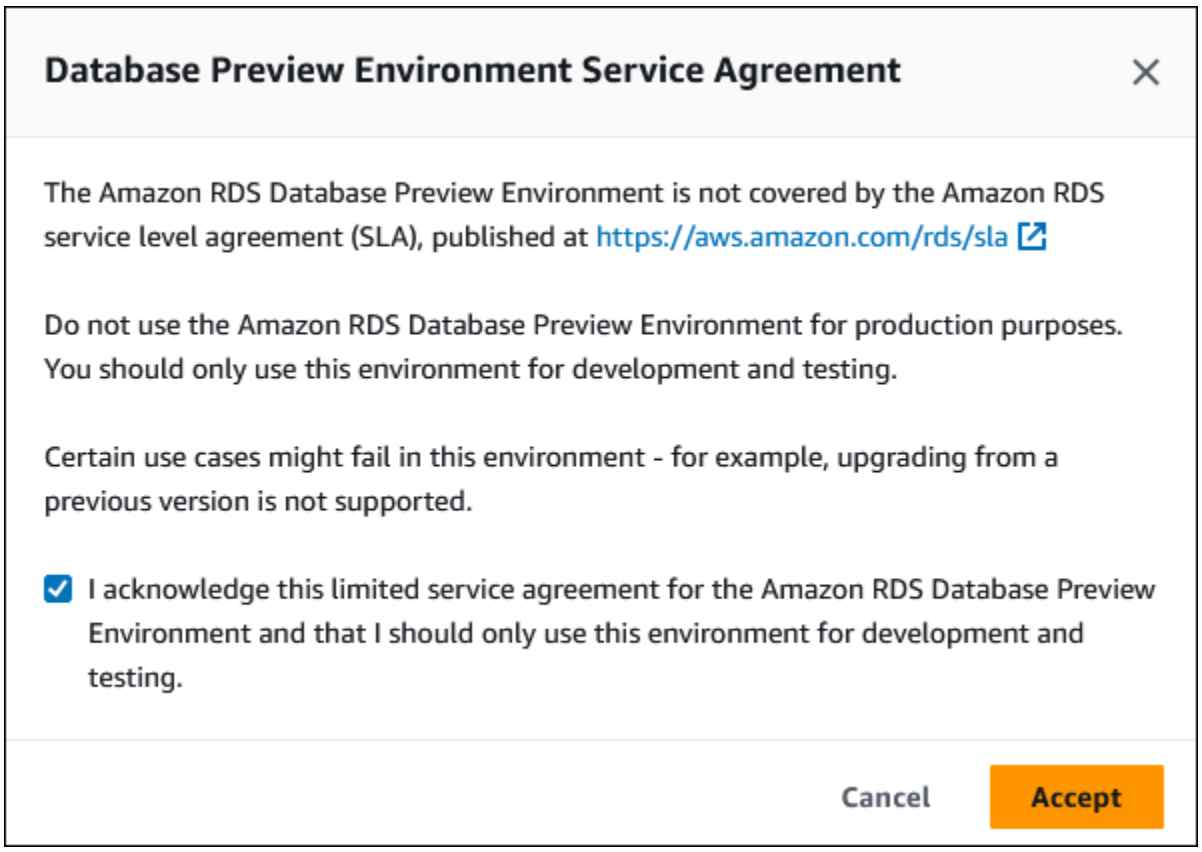
Konsol

Untuk membuat instans DB di Lingkungan Pratinjau Basis Data

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Pilih Dasbor dari panel navigasi.
3. Di halaman Dasbor, cari bagian Lingkungan Pratinjau Basis Data, seperti yang ditunjukkan pada gambar berikut.



Anda dapat langsung menuju Lingkungan pratinjau basis data. Sebelum dapat melanjutkan, Anda harus mengakui dan menerima batasan.



4. Untuk membuat instans DB RDS for MySQL, ikuti proses yang sama seperti untuk membuat instans DB Amazon RDS apa pun. Untuk informasi lebih lanjut, lihat prosedur Konsol di Membuat instans DB.

AWS CLI

Untuk membuat instance DB di lingkungan Pratinjau Database menggunakan AWS CLI, gunakan endpoint berikut.

```
rds-preview.us-east-2.amazonaws.com
```

Untuk membuat instans DB RDS for MySQL, ikuti proses yang sama seperti untuk membuat instans DB Amazon RDS apa pun. Untuk informasi lebih lanjut, lihat prosedur AWS CLI di Membuat instans DB.

API RDS

Untuk membuat instans DB di Lingkungan Pratinjau Basis Data menggunakan API RDS, gunakan titik akhir berikut.

```
rds-preview.us-east-2.amazonaws.com
```

Untuk membuat instans DB RDS for MySQL, ikuti proses yang sama seperti untuk membuat instans DB Amazon RDS apa pun. Untuk informasi lebih lanjut, lihat prosedur API RDS di Membuat instans DB.

MySQL versi 8.2 di lingkungan Pratinjau Database

MySQL versi 8.2 sekarang tersedia di lingkungan Amazon RDS Database Preview. MySQL versi 8.2 berisi beberapa perbaikan yang dijelaskan dalam Perubahan di MySQL 8.2.0.

Untuk informasi tentang Lingkungan Pratinjau Basis Data, lihat the section called “ Lingkungan Pratinjau Basis Data”. Untuk mengakses Lingkungan Pratinjau dari konsol, pilih https://console.aws.amazon.com/rds-preview/.

MySQL versi 8.1 di lingkungan Pratinjau Basis Data

MySQL versi 8.1 sekarang tersedia di lingkungan Pratinjau Basis Data Amazon RDS. MySQL versi 8.1 berisi beberapa perbaikan yang dijelaskan di Perubahan dalam MySQL 8.1.0.

Untuk informasi tentang Lingkungan Pratinjau Basis Data, lihat the section called “ Lingkungan Pratinjau Basis Data”. Untuk mengakses Lingkungan Pratinjau dari konsol, pilih https://console.aws.amazon.com/rds-preview/.

Versi-versi yang dihentikan untuk Amazon RDS for MySQL

Amazon RDS for MySQL versi 5.1, 5.5, dan 5.6 telah dihentikan.

Untuk informasi tentang kebijakan penghentian Amazon RDS untuk MySQL, lihat Pertanyaan Umum Amazon RDS.

Menghubungkan ke instans DB yang menjalankan mesin basis data MySQL

Sebelum Anda dapat terhubung ke instans DB yang menjalankan mesin basis data MySQL, Anda harus membuat instans DB. Untuk mengetahui informasinya, lihat Membuat instans DB Amazon RDS. Setelah Amazon RDS menyediakan instans DB, Anda dapat menggunakan aplikasi klien MySQL standar atau utilitas guna terhubung ke instans tersebut. Dalam string koneksi, Anda menentukan alamat DNS dari titik akhir instans DB sebagai parameter host, dan menentukan nomor port dari titik akhir instans DB sebagai parameter port.

Untuk mengautentikasi ke instans DB RDS, Anda dapat menggunakan salah satu metode autentikasi untuk MySQL dan autentikasi basis data AWS Identity and Access Management (IAM):

- Untuk mempelajari cara mengautentikasi ke MySQL menggunakan salah satu metode autentikasi untuk MySQL, lihat Metode autentikasi dalam dokumentasi MySQL.
- Untuk mempelajari cara mengautentikasi ke MySQL menggunakan autentikasi basis data IAM, lihat Autentikasi basis data IAM untuk MariaDB, MySQL, dan PostgreSQL.

Anda dapat terhubung ke instans DB MySQL dengan menggunakan alat seperti alat baris perintah MySQL. Untuk mengetahui informasi selengkapnya tentang cara menggunakan klien baris perintah MySQL, lihat mysql - alat baris perintah MySQL di dokumentasi MySQL. Satu aplikasi berbasis GUI yang dapat Anda gunakan untuk terhubung adalah MySQL Workbench. Untuk mengetahui informasi selengkapnya, lihat halaman Mengunduh MySQL Workbench. Untuk mengetahui informasi tentang cara menginstal MySQL (termasuk klien baris perintah MySQL), lihat Menginstal dan meningkatkan MySQL.

Sebagian besar distribusi Linux menyertakan klien MariaDB, bukan klien Oracle MySQL. Untuk menginstal klien baris perintah MySQL di Amazon Linux 2023, jalankan perintah berikut:

```
sudo dnf install mariadb105
```

Untuk menginstal klien baris perintah MySQL di Amazon Linux 2, jalankan perintah berikut:

```
sudo yum install mariadb
```

Untuk menginstal klien baris perintah MySQL di sebagian besar distribusi Linux berbasis DEB, jalankan perintah berikut:

```
apt-get install mariadb-client
```

Untuk memeriksa versi klien baris perintah MySQL Anda, jalankan perintah berikut:

```
mysql --version
```

Untuk membaca dokumentasi MySQL untuk versi klien Anda saat ini, jalankan perintah berikut:

```
man mysql
```

Untuk terhubung ke instans DB dari luar Amazon VPC-nya, instans DB harus dapat diakses secara publik, akses harus diberikan menggunakan aturan masuk grup keamanan instans DB, dan persyaratan lain harus dipenuhi. Untuk mengetahui informasinya, lihat Tidak dapat terhubung ke instans DB Amazon RDS.

Anda dapat menggunakan enkripsi Secure Sockets Layer (SSL) atau Keamanan Lapisan Pengangkutan (TLS) pada koneksi ke instans DB MySQL. Untuk mengetahui informasinya, lihat Menggunakan SSL/TLS dengan instans DB MySQL. Jika Anda menggunakan autentikasi basis data AWS Identity and Access Management (IAM), pastikan untuk menggunakan koneksi SSL/TLS. Untuk mengetahui informasinya, lihat Autentikasi basis data IAM untuk MariaDB, MySQL, dan PostgreSQL.

Anda juga dapat terhubung ke instans DB dari server web. Untuk mengetahui informasi selengkapnya, lihat Tutorial: Membuat server web dan instans DB Amazon RDS.

> **Note**
>
> Untuk mengetahui informasi tentang cara menghubungkan ke instans DB MariaDB, lihat Menghubungkan ke instans DB yang menjalankan mesin basis data MariaDB.

Topik

- Menemukan informasi koneksi untuk instans DB MySQL
- Menghubungkan dari klien baris perintah MySQL (tidak terenkripsi)
- Menghubungkan dari MySQL Workbench
- Menghubungkan dengan Amazon Web Services JDBC Driver for MySQL
- Memecahkan masalah koneksi ke instans DB MySQL Anda

Menemukan informasi koneksi untuk instans DB MySQL

Informasi koneksi untuk instans DB mencakup titik akhir, port, dan pengguna basis datanya yang valid, seperti pengguna utama. Sebagai contoh, anggaplah bahwa nilai titik akhir adalah `mydb.123456789012.us-east-1.rds.amazonaws.com`. Dalam hal ini, nilai port adalah `3306`, dan pengguna basis data adalah `admin`. Dengan informasi ini, Anda menentukan nilai-nilai berikut dalam string koneksi:

- Untuk host atau nama host, atau nama DNS, tentukan `mydb.123456789012.us-east-1.rds.amazonaws.com`.
- Untuk port, tentukan `3306`.
- Untuk pengguna, tentukan `admin`.

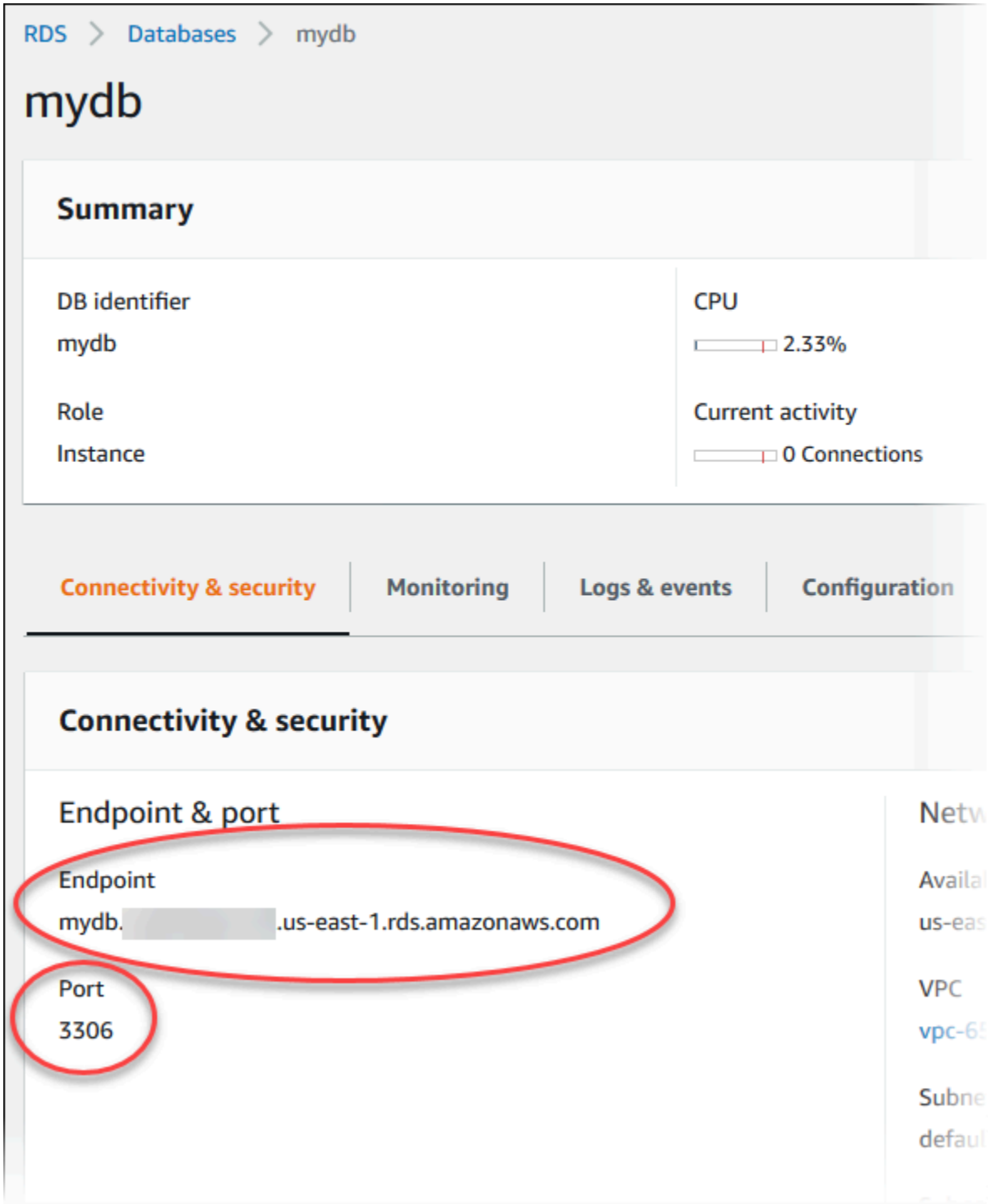
Untuk terhubung ke instans DB, gunakan klien apa saja untuk mesin DB MySQL. Misalnya, Anda dapat menggunakan klien baris perintah MySQL atau MySQL Workbench.

Untuk menemukan informasi koneksi untuk instans DB, Anda dapat menggunakan, AWS CLI describe-db-instancesperintahAWS Management Console, atau operasi Amazon RDS API DescribedBInstances untuk mencantumkan detailnya.

Konsol

Untuk menemukan informasi koneksi instans DB di AWS Management Console

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data untuk menampilkan daftar instans DB Anda.
3. Pilih nama instans DB MySQL untuk menampilkan detailnya.
4. Di tab Konektivitas & keamanan, salin titik akhir. Selain itu, catat nomor porta. Anda memerlukan titik akhir dan nomor port untuk terhubung ke instans DB.



5. Jika Anda perlu menemukan nama pengguna utama, pilih tab Konfigurasi dan lihat nilai Nama pengguna utama.

AWS CLI

Untuk menemukan informasi koneksi untuk instance MySQL DB dengan menggunakan, panggil AWS CLI perintah. describe-db-instances Dalam panggilan tersebut, buat kueri untuk ID instans DB, titik akhir, port, dan nama pengguna utama.

Untuk Linux, macOS, atau Unix:

```
aws rds describe-db-instances \
  --filters "Name=engine,Values=mysql" \
  --query "*[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername]"
```

Untuk Windows:

```
aws rds describe-db-instances ^
  --filters "Name=engine,Values=mysql" ^
  --query "*[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername]"
```

Output Anda akan terlihat seperti berikut ini.

```
[
    [
        "mydb1",
        "mydb1.123456789012.us-east-1.rds.amazonaws.com",
        3306,
        "admin"
    ],
    [
        "mydb2",
        "mydb2.123456789012.us-east-1.rds.amazonaws.com",
        3306,
        "admin"
    ]
]
```

API RDS

Untuk menemukan informasi koneksi instans DB dengan menggunakan API Amazon RDS, panggil operasi DescribeDBInstances. Dalam output, temukan nilai untuk alamat titik akhir, port titik akhir, dan nama pengguna utama.

Menghubungkan dari klien baris perintah MySQL (tidak terenkripsi)

> **Important**
>
> Hanya gunakan koneksi MySQL yang tidak terenkripsi saat klien dan server berada di VPC yang sama dan jaringan tepercaya. Untuk mengetahui informasi tentang cara menggunakan koneksi terenkripsi, lihat Menghubungkan dari klien baris perintah MySQL dengan SSL/TLS (terenkripsi).

Untuk terhubung ke instans DB menggunakan klien baris perintah MySQL, ketik perintah berikut pada prompt perintah. Untuk parameter -h, lakukan penggantian pada nama DNS (titik akhir) untuk instans DB Anda. Untuk parameter -P, lakukan penggantian pada port untuk instans DB Anda. Untuk parameter -u, lakukan penggantian nama pengguna dari pengguna basis data yang valid, seperti pengguna master. Masukkan kata sandi pengguna master saat diminta.

```
mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com -P 3306 -
u mymasteruser -p
```

Setelah memasukkan kata sandi untuk pengguna, Anda akan melihat output yang terlihat seperti berikut ini.

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9738
Server version: 8.0.28 Source distribution

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

mysql>
```

Menghubungkan dari MySQL Workbench

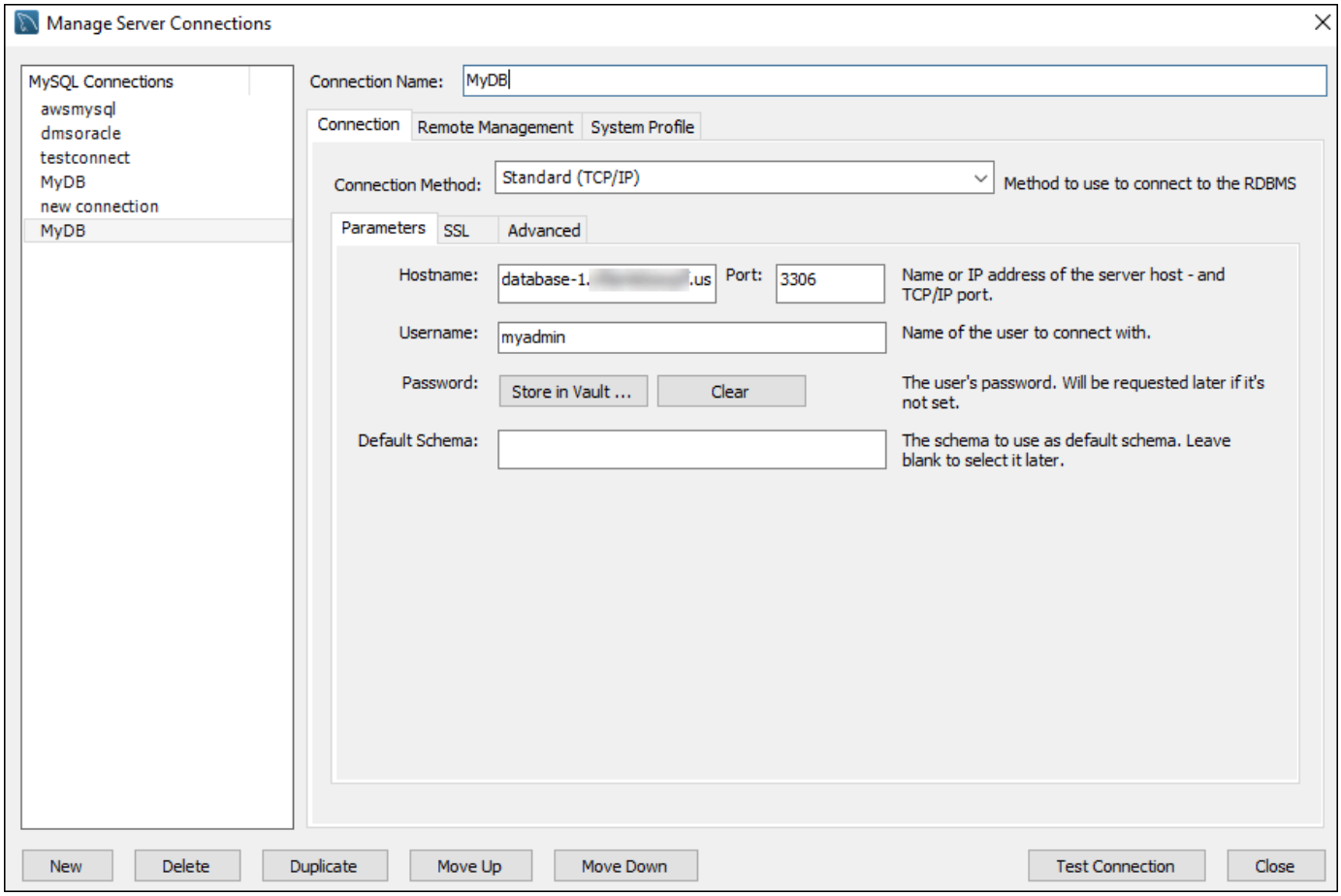
Untuk menghubungkan dari MySQL Workbench

1. Unduh dan instal MySQL Workbench di Mengunduh MySQL Workbench.
2. Buka MySQL Workbench.

3. Dari Basis Data, pilih Kelola Koneksi.
4. Di jendela Kelola Koneksi Server, pilih Baru.
5. Di jendela Hubungkan ke Basis Data, masukkan informasi berikut:

   - Koneksi Tersimpan – Masukkan nama untuk koneksi, seperti **`MyDB`**.
   - Nama Host – Masukkan titik akhir instans DB.
   - Port – Masukkan port yang digunakan oleh instans DB.
   - Nama Pengguna – Masukkan nama pengguna dari pengguna basis data yang valid, seperti pengguna master.
   - Kata Sandi – Atau, pilih Simpan di Vault lalu masukkan dan simpan kata sandi untuk pengguna.

   Jendelanya akan terlihat seperti berikut:



Anda dapat menggunakan fitur MySQL Workbench untuk menyesuaikan koneksi. Misalnya, Anda dapat menggunakan tab SSL untuk mengonfigurasi koneksi SSL/TLS. Untuk mengetahui informasi tentang cara menggunakan MySQL Workbench, lihat Dokumentasi MySQL Workbench. Mengenkripsi koneksi klien ke instans DB MySQL dengan SSL/TLS, lihat Mengenkripsi koneksi klien ke instans DB MySQL dengan SSL/TLS.

6. Atau, pilih Tes Koneksi untuk mengonfirmasi bahwa koneksi ke instans DB berhasil.
7. Pilih Tutup.
8. Dari Basis Data, pilih Hubungkan ke Basis Data.
9. Dari Koneksi Tersimpan, pilih koneksi Anda.
10. Pilih OKE.

Menghubungkan dengan Amazon Web Services JDBC Driver for MySQL

AWS JDBC Driver for MySQL adalah driver klien yang dirancang untuk RDS for MySQL. Secara default, driver memiliki pengaturan yang dioptimalkan untuk digunakan dengan RDS for MySQL.

Untuk informasi lebih lanjut tentang driver dan petunjuk lengkap untuk menggunakannya, lihat Driver AWS JDBC untuk repositori MySQL. GitHub

Driver ini kompatibel dengan driver MySQL Connector/J. Untuk menginstal atau meningkatkan konektor Anda, ganti file .jar konektor MySQL (berada di aplikasi CLASSPATH) dengan AWS JDBC Driver untuk file .jar MySQL, dan perbarui awalan URL koneksi dari `jdbc:mysql://` ke `jdbc:mysql:aws://`.

Driver JDBC AWS untuk MySQL mendukung autentikasi basis data IAM. Untuk informasi selengkapnya, lihat Autentikasi Database AWS IAM di AWS Driver JDBC untuk repositori MySQL. GitHub Untuk mengetahui informasi selengkapnya tentang autentikasi basis data IAM, lihat Autentikasi basis data IAM untuk MariaDB, MySQL, dan PostgreSQL.

Memecahkan masalah koneksi ke instans DB MySQL Anda

Dua penyebab umum kegagalan koneksi ke instans DB baru adalah:

- Instans DB dibuat menggunakan grup keamanan yang tidak mengotorisasi koneksi dari perangkat atau instans Amazon EC2 tempat aplikasi atau utilitas MySQL berjalan. Instans DB harus memiliki grup keamanan VPC yang mengotorisasi koneksi. Untuk mengetahui informasi selengkapnya, lihat Amazon VPC dan Amazon RDS.

  Anda dapat menambahkan atau mengedit aturan masuk di grup keamanan. Untuk Sumber, pilih IP Saya. Pilihan ini akan mengizinkan akses ke instans DB dari alamat IP yang terdeteksi di browser Anda.
- Instans DB dibuat menggunakan port default 3306, dan perusahaan Anda memiliki aturan firewall yang memblokir koneksi ke port tersebut dari perangkat di jaringan perusahaan Anda. Untuk memperbaiki kegagalan ini, buat ulang instans dengan port yang berbeda.

Untuk mengetahui informasi selengkapnya tentang masalah koneksi, lihat Tidak dapat terhubung ke instans DB Amazon RDS.

Mengamankan koneksi instans DB MySQL

Anda dapat mengelola keamanan instans DB MySQL Anda.

Topik

Keamanan MySQL di Amazon RDS

Keamanan untuk instans DB MySQL dikelola pada tiga tingkat:

- AWS Identity and Access Management mengontrol siapa yang dapat melakukan tindakan manajemen Amazon RDS pada instans DB. Saat Anda terhubung ke AWS menggunakan kredensial IAM, akun IAM Anda harus memiliki kebijakan IAM yang memberikan izin yang diperlukan untuk menjalankan operasi manajemen Amazon RDS. Untuk informasi selengkapnya, lihat Manajemen identitas dan akses untuk Amazon RDS.
- Saat membuat instans basis data, Anda menggunakan grup keamanan VPC untuk mengendalikan perangkat dan instans Amazon EC2 yang boleh membuka koneksi dengan titik akhir dan port instans basis data. Koneksi tersebut dapat dilakukan menggunakan Secure Sockets Layer (SSL) dan Keamanan Lapisan Pengangkutan (TLS). Selain itu, aturan firewall di perusahaan Anda dapat mengontrol apakah perangkat yang berjalan di perusahaan Anda dapat membuka koneksi ke instans DB.
- Untuk mengautentikasi login dan izin untuk instans DB MySQL, Anda dapat mengambil pendekatan berikut, atau kombinasi keduanya.

  Anda dapat menggunakan pendekatan yang sama seperti instans DB MySQL yang berdiri sendiri. Perintah seperti `CREATE USER`, `RENAME USER`, `GRANT`, `REVOKE`, dan `SET PASSWORD` berfungsi seperti halnya dalam basis data on-premise, termasuk memodifikasi langsung tabel skema basis data. Untuk informasi, lihat Kontrol akses dan manajemen akun dalam dokumentasi MySQL.

Anda juga dapat menggunakan autentikasi basis data IAM. Dengan autentikasi basis data IAM, Anda mengautentikasi instans DB Anda menggunakan pengguna IAM atau peran IAM dan token autentikasi. Token autentikasi adalah nilai unik yang dihasilkan dengan menggunakan proses penandatanganan Signature Versi 4. Dengan menggunakan autentikasi basis data IAM, Anda dapat menggunakan kredensial yang sama untuk mengontrol akses ke sumber daya AWS dan basis data Anda. Untuk informasi selengkapnya, lihat Autentikasi basis data IAM untuk MariaDB, MySQL, dan PostgreSQL.

Opsi lain adalah autentikasi Kerberos untuk RDS for MySQL. Instans DB bekerja dengan AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD) untuk mengaktifkan autentikasi Kerberos. Saat pengguna mengautentikasi instans DB MySQL yang digabungkan ke domain yang percaya, permintaan autentikasi diteruskan. Permintaan yang diteruskan diarahkan ke direktori domain yang Anda buat dengan AWS Directory Service. Untuk informasi selengkapnya, lihat Menggunakan autentikasi Kerberos untuk MySQL.

Saat Anda membuat instans DB Amazon RDS, pengguna master memiliki hak istimewa default berikut:

- `alter`
- `alter routine`
- `create`
- `create routine`
- `create temporary tables`
- `create user`
- `create view`
- `delete`
- `drop`
- `event`
- `execute`
- `grant option`
- `index`
- `insert`
- `lock tables`

- `process`
- `references`
- `replication client`
- `replication slave`
- `select`
- `show databases`
- `show view`
- `trigger`
- `update`

> **Note**
>
> Meskipun pengguna master pada instans DB memungkinkan untuk dihapus, tindakan ini tidak disarankan. Untuk membuat ulang pengguna master, gunakan operasi ModifyDBInstance API RDS atau perintah modify-db-instance AWS CLI dan tetapkan kata sandi pengguna master baru dengan parameter yang sesuai. Jika tidak ada dalam instans, pengguna master akan dibuat dengan kata sandi yang ditentukan.

Untuk menyediakan layanan pengelolaan bagi setiap instans basis data, pengguna `rdsadmin` dibuat saat instans basis data dibuat. Upaya untuk menghapus, mengganti nama, mengubah kata sandi, atau mengubah hak istimewa untuk akun `rdsadmin` akan mengakibatkan kesalahan.

Untuk memungkinkan manajemen instans DB, perintah `kill` dan `kill_query` standar telah dibatasi. Perintah Amazon RDS `rds_kill` dan `rds_kill_query` diberikan agar Anda dapat mengakhiri sesi pengguna atau kueri pada instans DB.

Menggunakan Plugin Validasi Kata Sandi untuk RDS for MySQL

MySQL menyediakan plugin `validate_password` untuk keamanan yang lebih baik. Plugin tersebut memberlakukan kebijakan kata sandi menggunakan parameter di dalam grup parameter DB untuk instans DB MySQL Anda. Plugin ini didukung untuk instans DB yang menjalankan MySQL versi 5.7 dan 8.0. Untuk informasi lebih lanjut tentang plugin `validate_password`, lihat Plugin Validasi Kata Sandi di dalam dokumentasi MySQL.

Mengaktifkan plugin validate_password untuk instans DB MySQL

1. Buat koneksi ke instans DB MySQL dan jalankan perintah berikut.

```
INSTALL PLUGIN validate_password SONAME 'validate_password.so';
```

2. Konfigurasikan parameter untuk plugin di dalam grup parameter DB yang digunakan oleh instans DB.

   Untuk informasi lebih lanjut tentang parameter, lihat Opsi dan variabel Plugin Validasi Kata Sandi di dalam dokumentasi MySQL.

   Untuk informasi selengkapnya tentang cara memodifikasi parameter instans DB, lihat Memodifikasi parameter dalam grup parameter DB.

Setelah menginstal dan mengaktifkan plugin `password_validate`, atur ulang kata sandi yang ada agar sesuai dengan kebijakan validasi baru Anda.

Amazon RDS tidak memvalidasi kata sandi. Instans DB MySQL melakukan validasi kata sandi. Jika Anda mengatur kata sandi pengguna dengan AWS Management Console, perintah `modify-db-instance` AWS CLI, atau operasi API RDS `ModifyDBInstance`, perubahan dapat berhasil meskipun kata sandi baru tersebut tidak memenuhi kebijakan kata sandi Anda. Namun, kata sandi baru hanya diatur di dalam instans DB MySQL jika memenuhi kebijakan kata sandi. Dalam kasus ini, Amazon RDS merekam peristiwa berikut.

```
"RDS-EVENT-0067" - An attempt to reset the master password for the DB instance has
 failed.
```

Untuk informasi lebih lanjut tentang peristiwa Amazon RDS, lihat Menggunakan pemberitahuan peristiwa Amazon RDS.

Mengenkripsi koneksi klien ke instans DB MySQL dengan SSL/TLS

Lapisan Soket Aman (Secure Sockets Layer, SSL) adalah protokol standar industri untuk mengamankan koneksi jaringan antara klien dan server. Setelah SSL versi 3.0, namanya diubah

menjadi Keamanan Lapisan Pengangkutan (TLS). Amazon RDS mendukung enkripsi SSL/TLS untuk instans DB MySQL. Dengan SSL/TLS, Anda dapat mengenkripsi koneksi antara klien aplikasi dan instans DB MySQL. Dukungan SSL/TLS tersedia di semua Wilayah AWS untuk MySQL.

Topik

- Menggunakan SSL/TLS dengan instans DB MySQL
- Mewajibkan SSL/TLS untuk semua koneksi dengan instans basis data MySQL
- Menghubungkan dari klien baris perintah MySQL dengan SSL/TLS (terenkripsi)

Menggunakan SSL/TLS dengan instans DB MySQL

Amazon RDS membuat sertifikat SSL/TLS dan menginstal sertifikat tersebut pada instans DB ketika Amazon RDS menyediakan instans. Sertifikat ini ditandatangani oleh otoritas sertifikat. Sertifikat SSL/TLS mencakup titik akhir instans DB sebagai Nama Umum (Common Name, CN) untuk sertifikat SSL/TLS guna menghalangi serangan spoofing.

Sertifikat SSL/TLS yang dibuat oleh Amazon RDS adalah entitas akar tepercaya dan semestinya berfungsi dalam sebagian besar kasus, tetapi mungkin gagal jika aplikasi Anda tidak menerima rantai sertifikat. Jika aplikasi Anda tidak menerima rantai sertifikat, Anda mungkin perlu menggunakan sertifikat perantara untuk terhubung dengan Wilayah AWS Anda. Misalnya, Anda harus menggunakan sertifikat perantara untuk terhubung dengan Wilayah AWS GovCloud (US) menggunakan SSL/TLS.

Untuk informasi tentang mengunduh sertifikat, lihat . Lihat informasi yang lebih lengkap tentang cara menggunakan TLS/SSL dengan MySQL di Memperbarui aplikasi untuk terhubung ke instans DB MySQL menggunakan sertifikat SSL/TLS baru.

MySQL menggunakan OpenSSL untuk koneksi yang aman. Amazon RDS for MySQL mendukung Keamanan Lapisan Pengangkutan (TLS) versi 1.0, 1.1, 1.2, dan 1.3. Dukungan TLS bergantung pada versi MySQL. Tabel berikut menunjukkan dukungan TLS untuk versi MySQL.

Versi MySQL	TLS 1.0	TLS 1.1	TLS 1.2	TLS 1.3
MySQL 8.0	Tidak didukung	Tidak didukung	Didukung	Didukung
MySQL 5.7	Didukung	Didukung	Didukung	Tidak didukung

Anda dapat meminta koneksi SSL/TLS untuk akun pengguna tertentu. Misalnya, Anda dapat menggunakan salah satu pernyataan berikut, sesuai dengan versi MySQL Anda, untuk mewajibkan koneksi SSL/TLS pada akun pengguna `encrypted_user`.

Untuk melakukannya, gunakan pernyataan berikut.

```
ALTER USER 'encrypted_user'@'%' REQUIRE SSL;
```

Untuk informasi selengkapnya tentang koneksi SSL/TLS dengan MySQL, lihat Using encrypted connections dalam dokumentasi MySQL.

Mewajibkan SSL/TLS untuk semua koneksi dengan instans basis data MySQL

Gunakan parameter `require_secure_transport` untuk mewajibkan bahwa semua koneksi pengguna dengan instans DB MySQL Anda menggunakan SSL/TLS. Secara default, parameter `require_secure_transport` diatur ke `OFF`. Anda dapat mengatur parameter `require_secure_transport` ke `ON` guna mewajibkan SSL/TLS untuk koneksi dengan instans basis data Anda.

Anda dapat mengatur nilai parameter `require_secure_transport` dengan memperbarui grup parameter basis data untuk instans basis data Anda. Anda tidak perlu melakukan reboot instans DB agar perubahan berlaku.

Saat parameter `require_secure_transport` diatur ke `ON` untuk suatu instans DB, klien basis data dapat terhubung dengannya jika dapat membentuk koneksi terenkripsi. Jika tidak, pesan kesalahan yang serupa dengan yang berikut ini ditampilkan kepada klien:

```
MySQL Error 3159 (HY000): Connections using insecure transport are prohibited while --
require_secure_transport=ON.
```

Untuk informasi tentang mengatur parameter, lihat Memodifikasi parameter dalam grup parameter DB.

Untuk mengetahui informasi selengkapnya tentang parameter `require_secure_transport`, lihat dokumentasi MySQL.

Menghubungkan dari klien baris perintah MySQL dengan SSL/TLS (terenkripsi)

Parameter-parameter program klien `mysql` sedikit berbeda jika Anda menggunakan MySQL versi 5.7, MySQL versi 8.0, atau versi MariaDB.

Untuk mengetahui versi yang Anda miliki, jalankan perintah `mysql` dengan opsi `--version`. Dalam contoh berikut, output menunjukkan bahwa program klien berasal dari MariaDB.

```
$ mysql --version
mysql  Ver 15.1 Distrib 10.5.15-MariaDB, for osx10.15 (x86_64) using readline 5.1
```

Sebagian besar distribusi Linux, seperti Amazon Linux, CentOS, SUSE, dan Debian telah mengganti MySQL dengan MariaDB, dan versi `mysql` di dalamnya adalah dari MariaDB.

Untuk terhubung dengan instans DB Anda menggunakan SSL/TLS, ikuti langkah-langkah ini:

Untuk terhubung dengan instans DB dengan SSL/TLS menggunakan klien baris perintah MySQL

1. Unduh sertifikat root yang dapat digunakan untuk semua Wilayah AWS.

   Untuk informasi tentang mengunduh sertifikat, lihat .

2. Gunakan klien baris perintah MySQL untuk terhubung dengan instans DB melalui enkripsi SSL/TLS. Untuk parameter `-h`, ganti nama (titik akhir) DNS untuk instans DB Anda. Untuk parameter `--ssl-ca`, ganti nama file sertifikat SSL/TLS. Untuk parameter `-P`, ganti port untuk instans DB Anda. Untuk parameter `-u`, ganti nama pengguna dari pengguna basis data yang valid, seperti pengguna master. Masukkan kata sandi pengguna master ketika diminta.

   Contoh berikut menunjukkan cara meluncurkan klien dengan menggunakan parameter `--ssl-ca` yang memakai klien MySQL 5.7 atau lebih tinggi:

   ```
   mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=global-bundle.pem --ssl-mode=REQUIRED -P 3306 -u myadmin -p
   ```

   Untuk mewajibkan bahwa koneksi SSL/TLS memeriksa titik akhir instans DB terhadap titik akhir dalam sertifikat SSL/TLS, masukkan perintah berikut:

   ```
   mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=global-bundle.pem --ssl-mode=VERIFY_IDENTITY -P 3306 -u myadmin -p
   ```

Contoh berikut menunjukkan cara meluncurkan klien dengan menggunakan parameter `--ssl-ca` yang memakai klien MariaDB:

```
mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=global-bundle.pem --ssl -P 3306 -u myadmin -p
```

3. Masukkan kata sandi pengguna master ketika diminta.

Anda akan melihat output mirip dengan yang berikut ini.

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9738
Server version: 8.0.28 Source distribution

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

mysql>
```

Memperbarui aplikasi untuk terhubung ke instans DB MySQL menggunakan sertifikat SSL/TLS baru

Sejak 13 Januari 2023, Amazon RDS telah menerbitkan serifikat Otoritas Sertifikat (CA) baru untuk terhubung ke instans DB RDS menggunakan Lapisan Soket Aman atau Keamanan Lapisan Pengangkutan (SSL/TLS). Setelah itu, Anda dapat menemukan informasi tentang pembaruan aplikasi untuk menggunakan sertifikat baru.

Topik ini dapat membantu Anda menentukan apakah aplikasi klien menggunakan SSL/TLS untuk terhubung ke instans DB Anda. Jika demikian, Anda dapat memeriksa lebih lanjut apakah aplikasi tersebut memerlukan verifikasi sertifikat untuk terhubung.

> **Note**
>
> Beberapa aplikasi dikonfigurasi untuk terhubung ke instans DB MySQL hanya jika aplikasi tersebut berhasil memverifikasi sertifikat pada server. Untuk aplikasi tersebut, Anda harus memperbarui penyimpanan kepercayaan aplikasi klien untuk menyertakan sertifikat CA baru. Anda dapat menentukan mode SSL berikut: `disabled`, `preferred`, dan `required`. Saat Anda menggunakan mode SSL `preferred` dan sertifikat CA tidak ada atau tidak diperbarui, koneksi kembali tidak menggunakan SSL dan terhubung tanpa enkripsi.

Karena versi terbaru menggunakan protokol OpenSSL, sertifikat server yang kedaluwarsa tidak mencegah koneksi yang berhasil kecuali mode SSL `required` ditentukan. Sebaiknya hindari mode `preferred`. Dalam mode `preferred`, jika koneksi menghadapi sertifikat yang tidak valid, koneksi berhenti menggunakan enkripsi dan melanjutkan tanpa enkripsi.

Setelah Anda memperbarui sertifikat CA di penyimpanan kepercayaan aplikasi klien, Anda dapat merotasi sertifikat di instans DB Anda. Sebaiknya Anda menguji prosedur ini di lingkungan pengembangan dan penahapan sebelum menerapkannya di lingkungan produksi Anda.

Untuk informasi selengkapnya tentang rotasi sertifikat, lihat Merotasi sertifikat SSL/TLS. Untuk informasi selengkapnya tentang cara mengunduh sertifikat, lihat . Untuk informasi tentang menggunakan SSL/TLS dengan instans DB MySQL, lihat Menggunakan SSL/TLS dengan instans DB MySQL.

Topik

- Menentukan apakah ada aplikasi yang terhubung ke instans DB MySQL menggunakan SSL
- Menentukan apakah klien memerlukan verifikasi sertifikat untuk terhubung
- Memperbarui penyimpanan kepercayaan aplikasi Anda
- Contoh kode Java untuk membangun koneksi SSL

Menentukan apakah ada aplikasi yang terhubung ke instans DB MySQL menggunakan SSL

Jika Anda menggunakan Amazon RDS for MySQL versi 5.7 atau 8.0 dan Skema Performa diaktifkan, jalankan kueri berikut untuk memeriksa apakah koneksi menggunakan SSL/TLS. Untuk informasi tentang cara mengaktifkan Skema Performa, lihat Memulai cepat Skema Performa di dokumentasi MySQL.

```
mysql> SELECT id, user, host, connection_type
       FROM performance_schema.threads pst
       INNER JOIN information_schema.processlist isp
       ON pst.processlist_id = isp.id;
```

Dalam output contoh ini, Anda dapat melihat sesi Anda sendiri (`admin`) dan aplikasi yang masuk sebagai `webapp1` menggunakan SSL.

```
+----+-----------------+------------------+-----------------+
| id | user            | host             | connection_type |
+----+-----------------+------------------+-----------------+
|  8 | admin           | 10.0.4.249:42590 | SSL/TLS         |
|  4 | event_scheduler | localhost        | NULL            |
| 10 | webapp1         | 159.28.1.1:42189 | SSL/TLS         |
+----+-----------------+------------------+-----------------+
3 rows in set (0.00 sec)
```

Menentukan apakah klien memerlukan verifikasi sertifikat untuk terhubung

Anda dapat memeriksa apakah klien JDBC dan klien MySQL memerlukan verifikasi sertifikat untuk terhubung.

JDBC

Contoh MySQL Connector/J 8.0 berikut menunjukkan satu cara untuk memeriksa properti koneksi JDBC aplikasi untuk menentukan apakah koneksi yang berhasil memerlukan sertifikat yang valid. Untuk informasi selengkapnya tentang semua opsi koneksi JDBC untuk MySQL, lihat Properti konfigurasi di dokumentasi MySQL.

Saat menggunakan MySQL Connector/J 8.0, koneksi SSL memerlukan verifikasi terhadap sertifikat CA server jika properti koneksi Anda memiliki `sslMode` yang diatur ke `VERIFY_CA` atau `VERIFY_IDENTITY`, seperti pada contoh berikut.

```
Properties properties = new Properties();
properties.setProperty("sslMode", "VERIFY_IDENTITY");
properties.put("user", DB_USER);
properties.put("password", DB_PASSWORD);
```

> **Note**
>
> Jika Anda menggunakan MySQL Java Connector v5.1.38 atau yang lebih baru, atau MySQL Java Connector v8.0.9 atau yang lebih baru untuk terhubung ke basis data Anda, meski Anda

belum mengonfigurasi aplikasi secara eksplisit untuk menggunakan SSL/TLS saat terhubung ke basis data Anda, driver klien ini akan menggunakan SSL/TLS secara default. Selain itu, saat menggunakan SSL/TLS, aplikasi akan melakukan verifikasi sertifikat parsial dan gagal terhubung jika sertifikat server basis data kedaluwarsa.

MySQL

Contoh Klien MySQL berikut menunjukkan dua cara untuk memeriksa koneksi MySQL skrip untuk menentukan apakah koneksi yang berhasil memerlukan sertifikat yang valid. Untuk informasi selengkapnya tentang semua opsi koneksi dengan Klien MySQL, lihat Konfigurasi sisi klien untuk koneksi terenkripsi di dokumentasi MySQL.

Saat menggunakan Klien MySQL 5.7 atau MySQL 8.0, koneksi SSL memerlukan verifikasi terhadap sertifikat CA server jika, untuk opsi `--ssl-mode`, Anda menentukan `VERIFY_CA` atau `VERIFY_IDENTITY`, seperti pada contoh berikut.

```
mysql -h mysql-database.rds.amazonaws.com -uadmin -ppassword --ssl-ca=/tmp/ssl-cert.pem
 --ssl-mode=VERIFY_CA
```

Saat menggunakan Klien MySQL 5.6, koneksi SSL memerlukan verifikasi terhadap sertifikat CA server jika Anda menentukan opsi `--ssl-verify-server-cert`, seperti pada contoh berikut.

```
mysql -h mysql-database.rds.amazonaws.com -uadmin -ppassword --ssl-ca=/tmp/ssl-cert.pem
 --ssl-verify-server-cert
```

Memperbarui penyimpanan kepercayaan aplikasi Anda

Untuk informasi tentang cara memperbarui penyimpanan kepercayaan untuk aplikasi MySQL, lihat Menginstal sertifikat SSL di dokumentasi MySQL.

Untuk informasi tentang cara mengunduh sertifikat root, lihat .

Untuk contoh skrip yang mengimpor sertifikat, lihat Contoh skrip untuk mengimpor sertifikat ke trust store Anda.

> **Note**
>
> Saat memperbarui penyimpanan kepercayaan, Anda dapat mempertahankan sertifikat lama selain menambahkan sertifikat baru.

Jika Anda menggunakan driver JDBC mysql dalam aplikasi, atur properti berikut dalam aplikasi.

```
System.setProperty("javax.net.ssl.trustStore", certs);
System.setProperty("javax.net.ssl.trustStorePassword", "password");
```

Saat Anda memulai aplikasi, atur properti berikut.

```
java -Djavax.net.ssl.trustStore=/path_to_truststore/MyTruststore.jks -
Djavax.net.ssl.trustStorePassword=my_truststore_password com.companyName.MyApplication
```

> **Note**
>
> Tentukan kata sandi selain prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

Contoh kode Java untuk membangun koneksi SSL

Contoh kode berikut menunjukkan cara menyiapkan koneksi SSL yang memvalidasi sertifikat server menggunakan JDBC.

```
public class MySQLSSLTest {

        private static final String DB_USER = "username";
        private static final String DB_PASSWORD = "password";
        // This key store has only the prod root ca.
        private static final String KEY_STORE_FILE_PATH = "file-path-to-keystore";
        private static final String KEY_STORE_PASS = "keystore-password";

        public static void test(String[] args) throws Exception {
```

```
            Class.forName("com.mysql.jdbc.Driver");

            System.setProperty("javax.net.ssl.trustStore", KEY_STORE_FILE_PATH);
            System.setProperty("javax.net.ssl.trustStorePassword", KEY_STORE_PASS);

            Properties properties = new Properties();
            properties.setProperty("sslMode", "VERIFY_IDENTITY");
            properties.put("user", DB_USER);
            properties.put("password", DB_PASSWORD);


            Connection connection = null;
            Statement stmt = null;
            ResultSet rs = null;
            try {
                connection =
 DriverManager.getConnection("jdbc:mysql://mydatabase.123456789012.us-
east-1.rds.amazonaws.com:3306",properties);
                stmt = connection.createStatement();
                rs=stmt.executeQuery("SELECT 1 from dual");
            } finally {
                if (rs != null) {
                    try {
                        rs.close();
                    } catch (SQLException e) {
                    }
                }
                if (stmt != null) {
                   try {
                        stmt.close();
                    } catch (SQLException e) {
                   }
                }
                if (connection != null) {
                    try {
                        connection.close();
                    } catch (SQLException e) {
                        e.printStackTrace();
                    }
                }
            }
            return;
        }
    }
```

Important

Setelah Anda menentukan bahwa koneksi database Anda menggunakan SSL/TLS dan telah memperbarui toko kepercayaan aplikasi Anda, Anda dapat memperbarui database Anda untuk menggunakan sertifikat 2048-g1. rds-ca-rsa Untuk mengetahui petunjuknya, lihat langkah 3 dalam Memperbarui sertifikat CA Anda dengan memodifikasi instans atau cluster DB.

Tentukan kata sandi selain prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

Menggunakan autentikasi Kerberos untuk MySQL

Anda kini dapat menggunakan autentikasi Kerberos untuk mengautentikasi pengguna saat dia menghubungi instans basis data MySQL. Instans basis data bekerja dengan AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD) untuk mengaktifkan autentikasi Kerberos. Saat pengguna mengautentikasi ke instans basis data MySQL yang bergabung dengan domain yang memercayai, permintaan autentikasi diteruskan. Permintaan yang diteruskan menuju direktori domain yang Anda buat dengan AWS Directory Service.

Menyimpan semua kredensial Anda di direktori yang sama dapat menghemat waktu dan tenaga Anda. Dengan pendekatan ini, Anda memiliki sebuah tempat terpusat untuk menyimpan dan mengelola kredensial bagi beberapa instans basis data. Menggunakan direktori juga dapat meningkatkan profil keamanan keseluruhan Anda.

Kawasan dan ketersediaan versi

Ketersediaan dan dukungan fitur bervariasi di antara versi-versi spesifik setiap mesin basis data, dan di antara Wilayah AWS. Lihat informasi yang lebih lengkap tentang versi dan ketersediaan Kawasan Amazon RDS dengan autentikasi Kerberos di Autentikasi Kerberos.

Ikhtisar Menyiapkan autentikasi Kerberos untuk instans basis data MySQL

Untuk menyiapkan autentikasi Kerberos bagi instans basis data MySQL, selesaikan langkah-langkah umum berikut, yang nanti akan dijelaskan lebih terperinci:

1. Gunakan AWS Managed Microsoft AD untuk membuat direktori AWS Managed Microsoft AD. Anda dapat menggunakan AWS Management Console, AWS CLI, atau AWS Directory Service

untuk membuat direktori. Lihat detail cara melakukannya di Buat direktori AWS Managed Microsoft AD dalam Panduan Administrasi AWS Directory Service.

2. Buat sebuah peran AWS Identity and Access Management (IAM) yang menggunakan kebijakan IAM terkelola `AmazonRDSDirectoryServiceAccess`. Peran ini memungkinkan Amazon RDS melakukan panggilan ke direktori Anda.

   Agar peran mengizinkan akses, titik akhir AWS Security Token Service (AWS STS) harus diaktifkan di Wilayah AWS untuk akun AWS Anda. Titik-titik akhir AWS STS aktif secara bawaan di semua Wilayah AWS, dan Anda dapat menggunakannya tanpa tindakan lebih lanjut. Lihat informasi yang lebih lengkap di Mengaktifkan dan menonaktifkan AWS STS di Wilayah AWS dalam Panduan Pengguna IAM.

3. Buat dan konfigurasikan pengguna dalam direktori AWS Managed Microsoft AD dengan menggunakan alat Microsoft Active Directory. Lihat informasi yang lebih lengkap tentang cara membuat pengguna di Active Directory di Mengelola pengguna dan grup di Microsoft AD terkelola AWS dalam Panduan Administrasi AWS Directory Service.

4. Buat atau ubah instans basis data MySQL. Jika Anda menggunakan CLI atau API RDS dalam permintaan buat, tentukan pengidentifikasi domain dengan parameter `Domain`. Gunakan pengidentifikasi `d-*` yang dihasilkan saat Anda membuat direktori Anda dan nama peran yang Anda buat.

   Jika Anda mengubah instans basis data MySQL yang ada untuk menggunakan autentikasi Kerberos, atur parameter-parameter domain dan peran IAM untuk instans basis data. Temukan lokasi instans basis data dalam VPC yang sama dengan direktori domain.

5. Gunakan kredensial pengguna master Amazon RDS untuk menghubungi instans basis data MySQL. Buat pengguna di MySQL dengan menggunakan klausul `CREATE USER IDENTIFIED WITH 'auth_pam'`. Pengguna yang Anda buat dengan cara ini dapat masuk ke instans basis data MySQL dengan menggunakan autentikasi Kerberos.

Menyiapkan autentikasi Kerberos untuk instans basis data MySQL

Anda menggunakan AWS Managed Microsoft AD untuk mengatur autentikasi Kerberos bagi instans basis data MySQL. Untuk menyiapkan autentikasi Kerberos, lakukan langkah-langkah berikut.

Langkah 1: Buat sebuah direktori dengan menggunakan AWS Managed Microsoft AD

AWS Directory Service membuat Active Directory terkelola penuh di AWS Cloud. Saat Anda membuat direktori AWS Managed Microsoft AD, AWS Directory Service membuat dua server

pengendali domain dan Sistem Nama Domain (DNS) atas nama Anda. Server direktori dibuat di subnet berbeda di VPC. Redundansi ini membantu memastikan bahwa direktori Anda tetap dapat diakses meski terjadi kegagalan.

Saat Anda membuat direktori AWS Managed Microsoft AD, AWS Directory Service melakukan tugas berikut ini atas nama Anda:

- Menyiapkan Active Directory di dalam VPC.
- Membuat akun administrator direktori dengan nama pengguna Admin dan kata sandi yang ditentukan. Anda menggunakan akun ini untuk mengelola direktori Anda.

  > **Note**
  >
  > Pastikan untuk menyimpan kata sandi ini. AWS Directory Service tidak menyimpannya. Anda dapat mengaturnya ulang, tetapi tidak dapat mengambilnya.

- Membuat grup keamanan untuk pengendali direktori.

Saat Anda meluncurkan sebuah AWS Managed Microsoft AD, AWS membuat Unit Organisasi (OU) yang berisi semua objek direktori Anda. OU ini memiliki nama NetBIOS yang Anda masukkan saat membuat direktori, dan terletak di root domain. Root domain dimiliki dan dikelola oleh AWS.

Akun Admin yang dibuat dengan direktori AWS Managed Microsoft AD Anda memiliki izin untuk melakukan berbagai tugas administratif paling umum bagi OU Anda:

- Membuat, memperbarui, atau menghapus pengguna
- Menambahkan sumber daya ke domain Anda seperti server file atau cetak, lalu menetapkan izin untuk sumber daya tersebut kepada pengguna di OU Anda
- Membuat OU dan kontainer tambahan
- Mendelegasikan kewenangan
- Memulihkan objek yang dihapus dari Keranjang Sampah Active Directory
- Jalankan PowerShell modul AD dan DNS Windows pada Layanan Web Direktori Aktif

Akun Admin juga berhak untuk melakukan aktivitas di seluruh domain berikut:

- Mengelola konfigurasi DNS (menambahkan, menghapus, atau memperbarui catatan, zona, dan penerus)

- Melihat log peristiwa DNS
- Melihat log peristiwa keamanan

Untuk membuat direktori dengan AWS Managed Microsoft AD

1. Masuk ke AWS Management Console dan buka konsol AWS Directory Service di https://console.aws.amazon.com/directoryservicev2/.
2. Di panel navigasi, pilih Direktori, lalu pilih Siapkan Direktori.
3. Pilih AWS Managed Microsoft AD. AWS Managed Microsoft AD adalah satu-satunya opsi yang saat ini dapat Anda gunakan dengan Amazon RDS.
4. Masukkan informasi berikut:

   Nama DNS Direktori

   Nama yang sepenuhnya memenuhi syarat direktori, seperti **`corp.example.com`**.

   Nama NetBIOS direktori

   Nama singkat direktori, seperti **`CORP`**.

   Deskripsi direktori

   (Opsional) Deskripsi direktori.

   Kata sandi admin

   Kata sandi administrator direktori. Proses pembuatan direktori menciptakan akun administrator dengan nama pengguna Admin dan kata sandi ini.

   Kata sandi administrator direktori dan tidak boleh menyertakan kata "admin". Kata sandi peka terhadap huruf besar/kecil dan harus terdiri dari 8-64 karakter. Kata sandi juga harus berisi setidaknya satu karakter dari tiga di antara empat kategori berikut:

   - Huruf kecil (a-z)
   - Huruf besar (A-Z)
   - Angka (0–9)
   - Karakter non-alfanumerik (~!@#$%^&*_-+=`|\(){}[]:;"'<>,.?/)

   Konfirmasi kata sandi

   Kata sandi administrator diketik ulang.

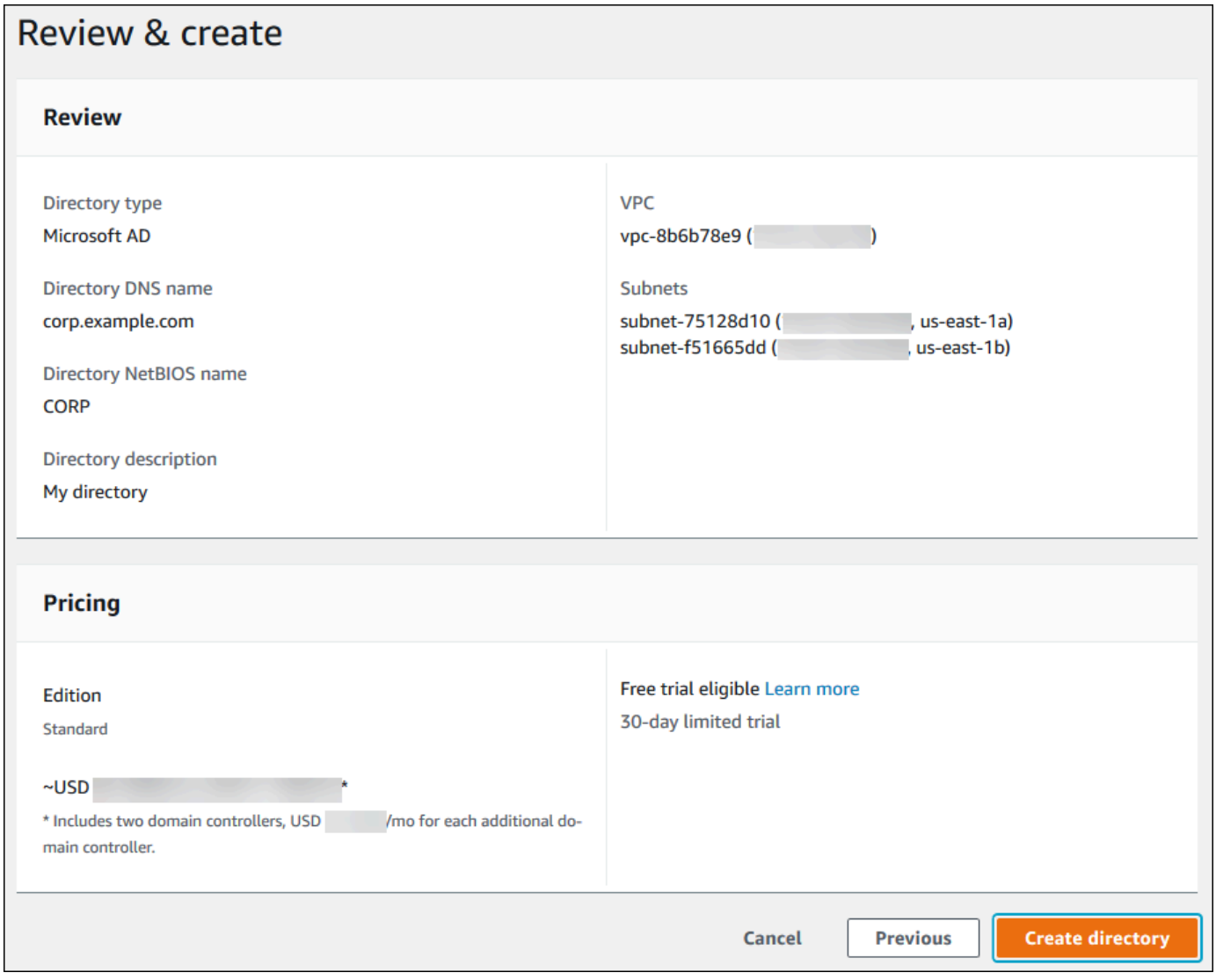
5. Pilih Berikutnya.
6. Masukkan informasi berikut di bagian Jaringan, lalu pilih Berikutnya:

   VPC

   VPC untuk direktori. Buat instans basis data MySQL dalam VPC yang sama ini.

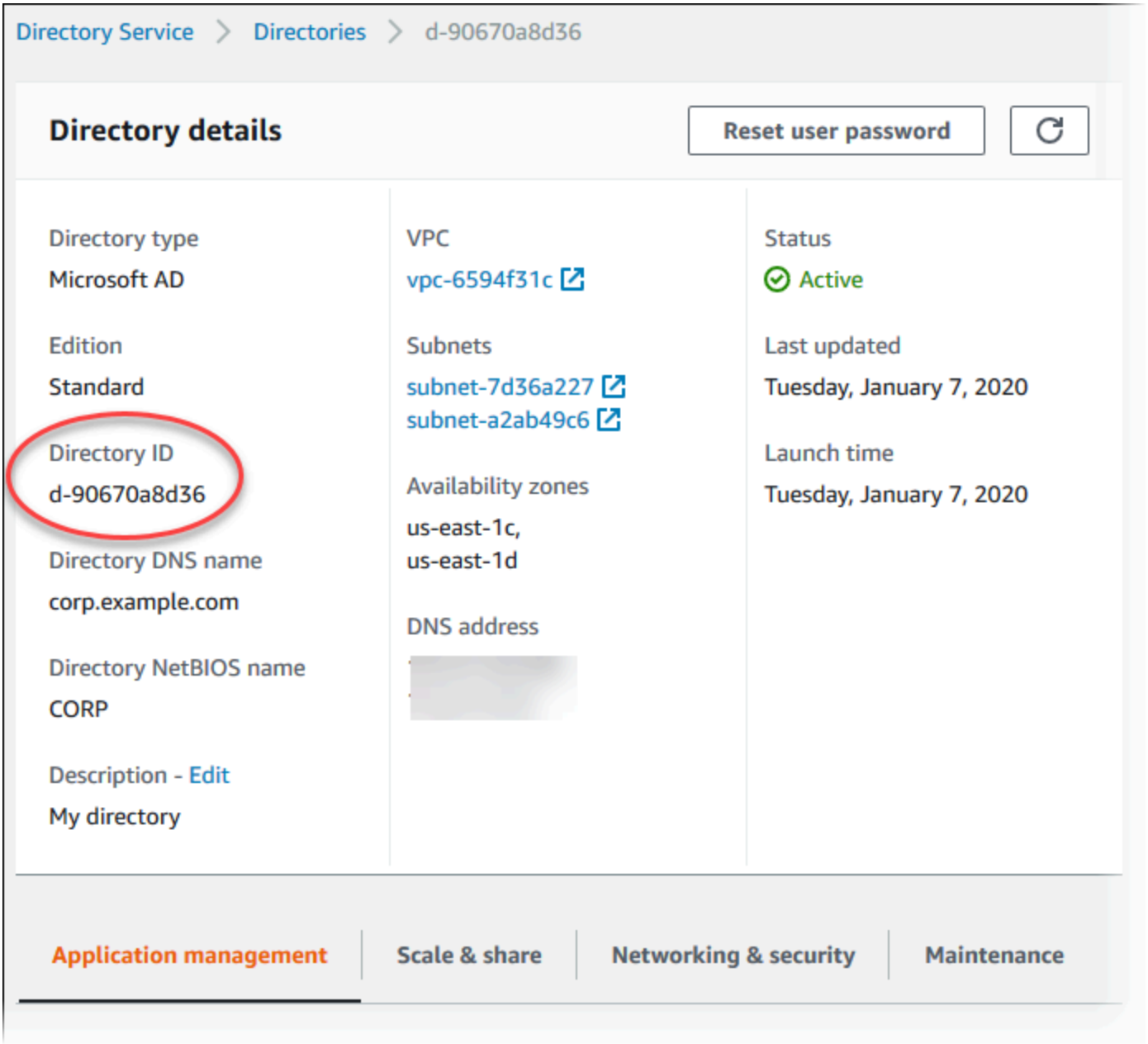
   Subnet

   Subnet untuk server direktori. Kedua subnet harus berada di Zona Ketersediaan yang berbeda.
7. Tinjau informasi direktori dan buat perubahan yang diperlukan. Jika informasi sudah benar, pilih Buat direktori.



Pembuatan direktori memerlukan waktu beberapa menit. Setelah direktori berhasil dibuat, nilai Status berubah menjadi Aktif.

Untuk melihat informasi tentang direktori Anda, pilih nama direktori di daftar direktori. Catat nilai ID Direktori karena Anda memerlukan nilai ini saat membuat atau mengubah instans basis data MySQL Anda.



Langkah 2: Buat peran IAM untuk penggunaan oleh Amazon RDS

Agar Amazon RDS memanggil AWS Directory Service untuk Anda, diperlukan peran IAM yang menggunakan kebijakan IAM terkelola `AmazonRDSDirectoryServiceAccess`. Peran ini memungkinkan Amazon RDS melakukan panggilan ke AWS Directory Service.

Ketika instans basis data dibuat dengan menggunakan AWS Management Console dan pengguna konsol memiliki izin `iam:CreateRole`, konsol akan membuat peran ini secara otomatis. Dalam hal ini, nama perannya adalah `rds-directoryservice-kerberos-access-role`. Jika tidak, Anda harus membuat peran IAM secara manual. Saat Anda membuat peran IAM ini, pilih `Directory Service`, lalu lampirkan kebijakan terkelola AWS `AmazonRDSDirectoryServiceAccess` ke peran itu.

Lihat informasi yang lebih lengkap tentang membuat peran IAM untuk sebuah layanan di Membuat peran untuk melimpahkan izin ke layanan AWS dalam Panduan Pengguna IAM.

> **Note**
>
> Peran IAM yang digunakan untuk Windows Authentication untuk RDS for SQL Server tidak dapat digunakan untuk RDS for MySQL.

Anda memiliki opsi untuk membuat kebijakan dengan izin yang diperlukan alih-alih menggunakan kebijakan IAM terkelola `AmazonRDSDirectoryServiceAccess`. Untuk melakukannya, peran IAM harus memiliki kebijakan kepercayaan IAM berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "directoryservice.rds.amazonaws.com",
          "rds.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
```

```
  ]
}
```

Peran ini juga harus memiliki kebijakan peran IAM berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ds:DescribeDirectories",
        "ds:AuthorizeApplication",
        "ds:UnauthorizeApplication",
        "ds:GetAuthorizedApplicationDetails"
      ],
    "Effect": "Allow",
    "Resource": "*"
    }
  ]
}
```

Langkah 3: Buat dan konfigurasikan pengguna

Anda dapat membuat pengguna dengan alat Active Directory Users and Computers. Alat ini adalah bagian dari alat Active Directory Domain Services dan Active Directory Lightweight Directory Services. Pengguna mewakili orang atau entitas individual yang memiliki akses ke direktori Anda.

Untuk membuat pengguna di direktori AWS Directory Service, Anda harus menghubungi instans Amazon EC2 yang berbasis Microsoft Windows. Instans ini harus anggota direktori AWS Directory Service dan masuk sebagai pengguna yang memiliki privilese membuat pengguna. Lihat informasi yang lebih lengkap di Kelola pengguna dan grup di AWS Managed Microsoft AD dalam Panduan Administrasi Layanan Direktori AWS.

Langkah 4: Buat atau ubah instans basis data MySQL

Buat atau ubah instans basis data MySQL untuk penggunaan dengan direktori Anda. Anda dapat menggunakan konsol, CLI, atau API RDS untuk mengaitkan instans basis data dengan direktori. Anda dapat menyesuaikan waktu ini dengan cara berikut:

- Buat instance MySQL DB baru menggunakan konsol, perintah create-db-instanceCLI, atau operasi CreateDBInstance RDS API.

Untuk petunjuk, lihat Membuat instans DB Amazon RDS.

- Ubah instance MySQL DB yang ada menggunakan konsol, perintah modify-db-instanceCLI, atau operasi ModifyDBInstance RDS API.

  Untuk petunjuk, lihat Memodifikasi instans DB Amazon RDS.

- Kembalikan instance MySQL DB dari snapshot DB menggunakan konsol, perintah CLI restore-db-instance-from-db-snapshot, atau operasi RestoreDB DBSnapshot RDS API. InstanceFrom

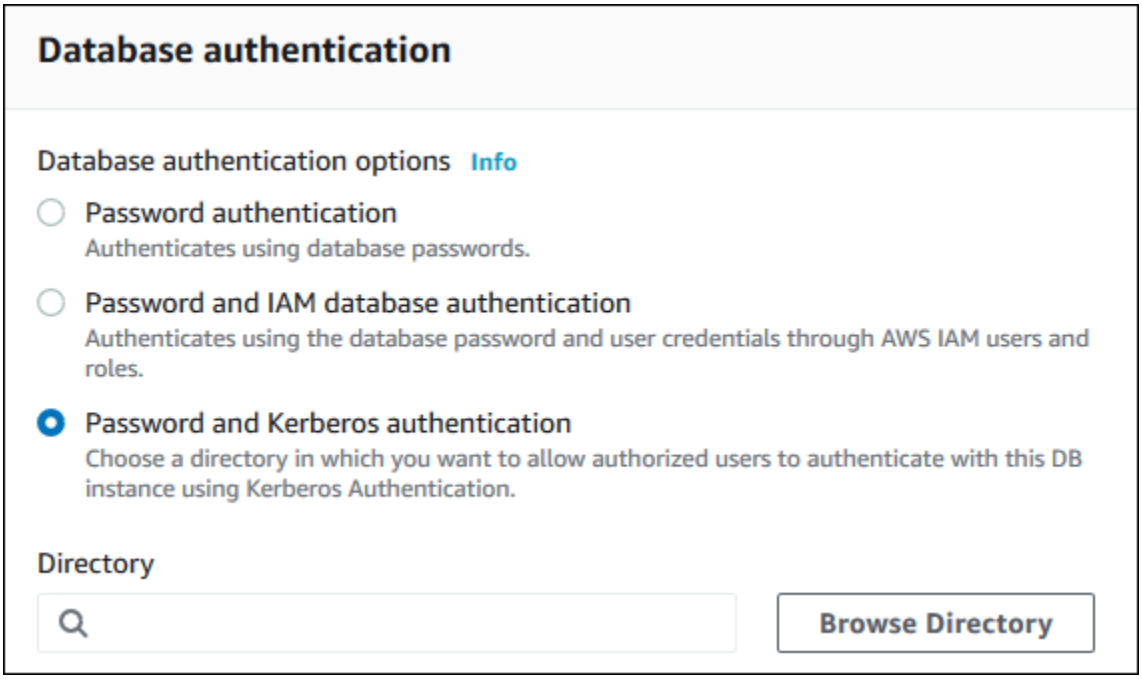
  Untuk petunjuk, lihat Memulihkan dari snapshot DB.

- Kembalikan instance MySQL DB ke point-in-time menggunakan konsol, perintah restore-db-instance-to- point-in-time CLI, atau operasi RestoreDB RDS API. InstanceToPointInTime

  Untuk petunjuk, lihat Memulihkan instans DB dengan waktu yang ditentukan.

Autentikasi Kerberos hanya didukung untuk instans basis data MySQL dalam VPC. Instans basis data dapat berada dalam VPC yang sama dengan direktori, atau dalam VPC yang berbeda. Instans basis data harus menggunakan grup keamanan yang memungkinkan data keluar dari VPC direktori sehingga instans basis data dapat berkomunikasi dengan direktori.

Saat Anda menggunakan konsol untuk membuat, memodifikasi, atau memulihkan instans DB, pilih Autentikasi kata sandi dan Kerberos di bagian Autentikasi basis data. Pilih Jelajah Direktori, lalu pilih direktori, atau pilih Buat direktori baru.



Saat Anda menggunakan AWS CLI atau API RDS, kaitkan instans basis data dengan direktori. Parameter-parameter berikut diperlukan bagi instans basis data untuk menggunakan direktori domain yang Anda buat:

- Untuk parameter `--domain`, gunakan pengidentifikasi domain (pengidentifikasi "d-*") yang dihasilkan saat Anda membuat direktori.
- Untuk parameter `--domain-iam-role-name`, gunakan peran yang Anda buat yang menggunakan kebijakan IAM terkelola `AmazonRDSDirectoryServiceAccess`.

Misalnya, perintah CLI berikut memodifikasi instans DB untuk menggunakan direktori.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --domain d-ID \
    --domain-iam-role-name role-name
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --domain d-ID ^
    --domain-iam-role-name role-name
```

> **Important**
>
> Jika Anda mengubah instans basis data untuk mengaktifkan autentikasi Kerberos, but ulang instans basis data setelah membuat perubahan.

Langkah 5: Buat login MySQL autentikasi Kerberos

Gunakan kredensial pengguna master Amazon RDS untuk menghubungi instans basis data MySQL sebagaimana Anda lakukan dengan instans basis data lain. Instans basis data digabungkan dengan domain AWS Managed Microsoft AD. Jadi, Anda dapat menyediakan login dan pengguna MySQL

dari pengguna Active Directory di domain Anda. Izin basis data dikelola melalui izin MySQL standar yang diberikan kepada dan dicabut dari login ini.

Anda dapat mengizinkan pengguna Active Directory untuk mengautentikasi dengan MySQL. Untuk melakukannya, pertama-tama gunakan kredensial pengguna master Amazon RDS untuk menghubungi instans basis data MySQL sebagaimana dengan instans basis data lain. Setelah Anda masuk, buat pengguna yang diautentikasi secara eksternal dengan PAM (Pluggable Authentication Modules) di MySQL seperti ditunjukkan berikut.

```
CREATE USER 'testuser'@'%' IDENTIFIED WITH 'auth_pam';
```

Ganti *testuser* dengan nama pengguna. Pengguna (baik orang maupun aplikasi) dari domain Anda kini dapat menghubungi instans basis data dari mesin klien yang digabungkan dengan domain dengan menggunakan autentikasi Kerberos.

> **Important**
>
> Kami sangat menyarankan agar klien menggunakan koneksi SSL/TLS saat menggunakan autentikasi PAM. Jika klien tidak menggunakan koneksi SSL/TLS, kata sandi mungkin terkirim sebagai teks jelas dalam beberapa kasus. Untuk mewajibkan koneksi terenkripsi SSL/TLS bagi pengguna AD Anda, jalankan perintah berikut:
>
> ```
> UPDATE mysql.user SET ssl_type = 'any' WHERE ssl_type = '' AND PLUGIN =
>  'auth_pam' and USER = 'testuser';
> FLUSH PRIVILEGES;
> ```
>
> Untuk informasi selengkapnya, lihat Menggunakan SSL/TLS dengan instans DB MySQL.

Mengelola instans basis data dalam domain

Anda dapat menggunakan CLI atau API RDS untuk mengelola instans basis data Anda dan hubungannya dengan Active Directory terkelola Anda. Misalnya, Anda dapat mengaitkan Active Directory untuk autentikasi Kerberos dan melepaskan Active Directory untuk menonaktifkan autentikasi Kerberos. Anda juga dapat memindahkan instans basis data agar diautentikasi secara eksternal oleh satu Active Directory ke yang lain.

Misalnya, dengan menggunakan API Amazon RDS, Anda dapat melakukan hal-hal berikut:

- Untuk mencoba lagi mengaktifkan autentikasi Kerberos pada keanggotaan yang gagal, gunakan operasi API `ModifyDBInstance` dan tentukan ID direktori keanggotaan saat ini.
- Untuk memperbarui nama peran IAM bagi keanggotaan, gunakan operasi API `ModifyDBInstance` dan tentukan ID direktori keanggotaan saat ini dan peran IAM baru.
- Untuk menonaktifkan autentikasi Kerberos pada instans basis data, gunakan operasi API `ModifyDBInstance` dan tentukan `none` sebagai parameter domain.
- Untuk memindahkan instans basis data dari satu domain ke domain lain, gunakan operasi API `ModifyDBInstance` dan tentukan pengidentifikasi domain baru sebagai parameter domain.
- Untuk memerinci keanggotaan bagi setiap instans basis data, gunakan operasi API `DescribeDBInstances`.

Memahami keanggotaan domain

Setelah Anda membuat atau mengubah instans basis data, instans akan menjadi anggota domain. Anda dapat melihat status keanggotaan domain untuk instans DB dengan menjalankan perintah describe-db-instancesCLI. Status instans DB dapat berupa salah satu dari daftar berikut:

- `kerberos-enabled` – Instans DB mengaktifkan autentikasi Kerberos.
- `enabling-kerberos` – AWS sedang dalam proses mengaktifkan autentikasi Kerberos pada instans basis data ini.
- `pending-enable-kerberos` – Pengaktifan autentikasi Kerberos tertunda pada instans basis data ini.
- `pending-maintenance-enable-kerberos` – AWS akan mencoba mengaktifkan autentikasi Kerberos pada instans basis data selama jendela pemeliharaan terjadwal berikutnya.
- `pending-disable-kerberos` – Penonaktifan autentikasi Kerberos tertunda pada instans basis data ini.
- `pending-maintenance-disable-kerberos` – AWS akan mencoba menonaktifkan autentikasi Kerberos pada instans basis data selama jendela pemeliharaan terjadwal berikutnya.
- `enable-kerberos-failed` – Sebuah masalah konfigurasi mencegah AWS dari mengaktifkan autentikasi Kerberos pada instans basis data. Periksa dan perbaiki konfigurasi Anda sebelum menerbitkan ulang perintah MODIFY instans basis data.
- `disabling-kerberos` – AWS sedang dalam proses menonaktifkan autentikasi Kerberos pada instans basis data ini.

Permintaan untuk mengaktifkan autentikasi Kerberos dapat gagal karena masalah konektivitas jaringan atau peran IAM yang salah. Misalnya, anggap bahwa Anda membuat instans basis data atau mengubah instans basis data yang sudah ada dan upaya mengaktifkan autentikasi Kerberos gagal. Jika ini terjadi, terbitkan ulang perintah pengubahan atau ubah instans basis data yang baru dibuat untuk bergabung dengan domain.

Membuat koneksi dengan MySQL lewat autentikasi Kerberos

Untuk menghubungi MySQL dengan autentikasi Kerberos, Anda harus masuk dengan menggunakan jenis autentikasi Kerberos.

Untuk membuat pengguna basis data yang dapat Anda hubungi dengan menggunakan autentikasi Kerberos, gunakan klausul `IDENTIFIED WITH` pada pernyataan `CREATE USER`. Lihat petunjuk di Langkah 5: Buat login MySQL autentikasi Kerberos.

Untuk menghindari kesalahan, gunakan klien `mysql` MariaDB. Anda dapat mengunduh perangkat lunak MariaDB di https://downloads.mariadb.org/.

Pada penggugah/prompt perintah, hubungi salah satu titik akhir yang terkait dengan instans basis data MySQL Anda. Ikuti prosedur-prosedur umum dalam Menghubungkan ke instans DB yang menjalankan mesin basis data MySQL. Saat Anda diminta untuk memasukkan kata sandi, masukkan kata sandi Kerberos yang terkait dengan nama penggunanya.

Memulihkan instans basis data MySQL dan menambahkannya ke domain

Anda dapat memulihkan snapshot DB atau menyelesaikan point-in-time pemulihan untuk instance MySQL DB dan kemudian menambahkannya ke domain. Setelah instans basis data dipulihkan, ubah instans itu dengan menggunakan proses yang dijelaskan dalam Langkah 4: Buat atau ubah instans basis data MySQL untuk menambahkannya ke domain.

Keterbatasan MySQL autentikasi Kerberos

Keterbatasan berikut berlaku bagi autentikasi Kerberos untuk MySQL:

- Hanya AWS Managed Microsoft AD yang didukung. Namun, Anda dapat menggabungkan instans basis data RDS for MySQL dengan domain Microsoft AD terkelola bersama yang dimiliki oleh beberapa akun di Wilayah AWS yang sama.
- Anda harus membut ulang instans basis data setelah mengaktifkan fitur ini.
- Panjang nama domain tidak boleh lebih dari 61 karakter.

- Anda tidak dapat mengaktifkan autentikasi Kerberos dan autentikasi IAM dengan serentak. Pilih satu metode autentikasi atau metode lain untuk instans basis data MySQL Anda.
- Jangan ubah porta instans basis data setelah mengaktifkan fitur ini.
- Jangan gunakan autentikasi Kerberos dengan replika baca.
- Jika pemutakhiran versi kecil otomatis untuk instans basis data MySQL yang menggunakan autentikasi Kerberos diaktifkan, Anda harus mematikan autentikasi Kerberos, lalu mengaktifkannya kembali setelah pemutakhiran otomatis. Lihat informasi yang lebih lengkap tentang peningkatan versi auto kecil di Upgrade versi minor otomatis untuk MySQL.
- Untuk menghapus instans basis data dengan fitur ini aktif, pertama-tama nonaktifkan fitur. Untuk melakukannya, gunakan perintah CLI `modify-db-instance` untuk instans basis data dan tentukan `none` untuk parameter `--domain`.

  Jika Anda menggunakan CLI atau API RDS untuk menghapus instans basis data dengan fitur ini aktif, akan terjadi penundaan.
- Anda tidak dapat mengatur hubungan kepercayaan rimba antara Microsoft Active Directory on-premise atau yang dihost mandiri dan AWS Managed Microsoft AD.

Meningkatkan performa kueri RDS for MySQL dengan Amazon RDS Optimized Reads

Anda dapat mempercepat pemrosesan kueri untuk RDS for MySQL dengan Amazon RDS Optimized Reads. Instans DB RDS for MySQL atau klaster DB Multi-AZ yang menggunakan RDS Optimized Reads dapat memproses kueri hingga 2x lebih cepat dibandingkan dengan instans atau klaster DB yang tidak menggunakannya.

Topik

Ikhtisar RDS Optimized Reads

Saat Anda menggunakan instans DB RDS for MySQL atau klaster DB Multi-AZ yang mengaktifkan RDS Optimized Reads, performa kueri akan lebih cepat melalui penggunaan penyimpanan instans. Penyimpanan instans menyediakan penyimpanan tingkat blok sementara untuk instans BD atau klaster DB Multi-AZ Anda. Penyimpanan terletak pada solid state drive (SSD) Non-Volatile Memory Express (NVMe) yang secara fisik terpasang ke server host. Penyimpanan ini dioptimalkan untuk latensi rendah, performa I/O acak tinggi, dan throughput baca berurutan tinggi.

RDS Optimized Reads diaktifkan secara default ketika instans DB atau klaster DB Multi-AZ menggunakan kelas instans DB dengan penyimpanan instans, seperti db.m5d atau db.m6gd. Dengan RDS Optimized Reads, beberapa objek sementara disimpan di penyimpanan instans. Objek sementara ini termasuk file sementara internal, tabel sementara internal pada disk, file peta memori, dan file cache log biner (binlog). Untuk informasi selengkapnya tentang penyimpanan instans, lihat Penyimpanan instans Amazon EC2 dalam Panduan Pengguna Amazon Elastic Compute Cloud untuk Instans Linux.

Beban kerja yang menghasilkan objek sementara di MySQL untuk pemrosesan kueri dapat memanfaatkan penyimpanan instans untuk mempercepat pemrosesan kueri. Jenis beban kerja ini mencakup kueri yang melibatkan pengurutan, agregasi hash, penggabungan dengan beban tinggi,

Ekspresi Tabel Umum (CTE), dan kueri pada kolom yang tidak diindeks. Volume penyimpanan instans ini memberikan IOPS dan performa yang lebih tinggi, terlepas dari konfigurasi penyimpanan yang digunakan untuk penyimpanan Amazon EBS secara persisten. Karena RDS Optimized Reads memindahkan beban operasi pada objek sementara ke penyimpanan instans, operasi input/output per detik (IOPS) atau throughput penyimpanan persisten (Amazon EBS) kini dapat digunakan untuk operasi pada objek persisten. Operasi ini mencakup pembacaan dan penulisan file data biasa, dan operasi mesin latar belakang, seperti flushing dan penggabungan buffer sisipan.

> **Note**
>
> Snapshot RDS manual dan otomatis hanya berisi file mesin untuk objek persisten. Objek sementara yang dibuat di penyimpanan instans tidak disertakan dalam snapshot RDS.

Kasus penggunaan RDS Optimized Reads

Jika Anda memiliki beban kerja yang sangat bergantung pada objek sementara, seperti tabel atau file internal, untuk eksekusi kueri, Anda dapat memperoleh manfaat dengan mengaktifkan RDS Optimized Reads. Kasus penggunaan berikut ini adalah kandidat untuk RDS Optimized Reads:

- Aplikasi yang menjalankan kueri analitis dengan ekspresi tabel umum (CTE) yang kompleks, tabel turunan, dan operasi pengelompokan
- Replika baca yang melayani lalu lintas baca padat dengan kueri yang tidak dioptimalkan
- Aplikasi yang menjalankan kueri pelaporan berdasarkan permintaan atau dinamis yang melibatkan operasi yang kompleks, seperti kueri dengan klausa `GROUP BY` dan `ORDER BY`
- Beban kerja yang menggunakan tabel sementara internal untuk pemrosesan kueri

  Anda dapat memantau variabel status mesin `created_tmp_disk_tables` untuk menentukan jumlah tabel sementara berbasis disk yang dibuat pada instans DB Anda.
- Aplikasi yang membuat tabel sementara dalam jumlah besar, baik secara langsung maupun dalam prosedur, untuk menyimpan hasil sementara
- Kueri basis data yang melakukan pengelompokan atau pengurutan pada kolom yang tidak diindeks

Praktik terbaik RDS Optimized Reads

Gunakan praktik terbaik RDS Optimized Reads berikut:

- Tambahkan logika coba lagi untuk kueri hanya baca jika terjadi kegagalan karena penyimpanan instans penuh selama eksekusi.
- Pantau ruang penyimpanan yang tersedia di penyimpanan instans dengan metrik CloudWatch `FreeLocalStorage`. Jika penyimpanan instans hampir penuh karena beban kerja pada instans DB, modifikasi instans DB untuk menggunakan kelas instans DB yang lebih besar.
- Jika memori instans DB atau klaster DB Multi-AZ Anda sudah memadai tetapi masih mencapai batas penyimpanan pada penyimpanan instans, tingkatkan nilai `binlog_cache_size` untuk mempertahankan entri binlog khusus sesi dalam memori. Konfigurasi ini akan mencegah penulisan entri binlog ke file cache binlog sementara pada disk.

  Parameter `binlog_cache_size` dibuat per sesi. Anda dapat mengubah nilai untuk setiap sesi baru. Pengaturan untuk parameter ini dapat meningkatkan pemanfaatan memori pada instans DB selama beban kerja mencapai puncaknya. Oleh karena itu, pertimbangkan untuk meningkatkan nilai parameter berdasarkan pola beban kerja aplikasi Anda dan memori yang tersedia pada instans DB.
- Gunakan nilai default `MIXED` untuk `binlog_format`. Tergantung ukuran transaksi, mengatur `binlog_format` ke `ROW` dapat menghasilkan file cache binlog berukuran besar pada penyimpanan instans.
- Atur parameter internal_tmp_mem_storage_engine ke `TempTable`, sesuaikan parameter temptable_max_mmap dengan ukuran penyimpanan yang tersedia pada penyimpanan instans.
- Jangan melakukan perubahan besar-besaran dalam satu transaksi. Transaksi seperti ini dapat menghasilkan file cache binlog berukuran besar pada penyimpanan instans dan dapat menyebabkan masalah ketika penyimpanan instans penuh. Pertimbangkan untuk membagi penulisan menjadi beberapa transaksi kecil guna meminimalkan penggunaan penyimpanan untuk file cache binlog.
- Gunakan nilai default `ABORT_SERVER` untuk parameter `binlog_error_action`. Hal ini dapat mencegah masalah pencatatan log biner pada instans DB yang mengaktifkan pencadangan.

Menggunakan RDS Optimized Reads

Saat Anda menyediakan instans DB RDS for MySQL dengan salah satu kelas instans DB berikut dalam deployment instans DB Satu AZ, deployment instans DB Multi-AZ, atau deployment klaster DB Multi-AZ, instans DB akan otomatis menggunakan RDS Optimized Reads.

Untuk mengaktifkan RDS Optimized Reads, lakukan salah satu tindakan berikut:

- Buat instans DB RDS for MySQL atau klaster DB Multi-AZ menggunakan salah satu kelas instans DB berikut. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.
- Modifikasi instans DB RDS for MySQL atau klaster DB Multi-AZ yang sudah ada untuk menggunakan salah satu kelas instans DB berikut. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

RDS Optimized Reads tersedia di semua Wilayah AWS RDS yang mendukung satu atau beberapa kelas instans DB dengan penyimpanan SSD NVMe lokal. Untuk informasi tentang kelas instans DB, lihat the section called “Kelas instans DB”.

Ketersediaan kelas instans DB untuk Wilayah AWS berbeda-beda. Untuk mengetahui dukungan kelas instans DB pada suatu Wilayah AWS, lihat the section called “Menentukan dukungan kelas instans DB di Wilayah AWS”.

Jika Anda tidak ingin menggunakan RDS Optimized Reads, modifikasi instans DB atau klaster DB Multi-AZ Anda agar tidak menggunakan kelas instans DB yang mendukung fitur tersebut.

Memantau instans DB yang menggunakan RDS Optimized Reads

Anda dapat memantau instans DB yang menggunakan RDS Optimized Reads dengan metrik CloudWatch berikut:

- `FreeLocalStorage`
- `ReadIOPSLocalStorage`
- `ReadLatencyLocalStorage`
- `ReadThroughputLocalStorage`
- `WriteIOPSLocalStorage`
- `WriteLatencyLocalStorage`
- `WriteThroughputLocalStorage`

Metrik ini menyediakan data tentang penyimpanan instans, IOPS, dan throughput yang tersedia. Untuk informasi selengkapnya tentang metrik ini, lihat Metrik CloudWatch tingkat instans Amazon untuk Amazon RDS.

Batasan RDS Optimized Reads

Batasan berikut berlaku untuk RDS Optimized Reads:

- RDS Optimalkan Reads didukung untuk RDS for MySQL versi 8.0.28 dan yang lebih tinggi. Untuk informasi tentang versi RDS for MySQL, lihat Versi MySQL di Amazon RDS.
- Anda tidak dapat mengubah lokasi objek sementara ke penyimpanan persisten (Amazon EBS) pada kelas instans DB yang mendukung RDS Optimized Reads.
- Saat pencatatan log biner pada instans DB diaktifkan, ukuran transaksi maksimum dibatasi oleh ukuran penyimpanan instans. Pada MySQL, setiap sesi yang membutuhkan penyimpanan lebih banyak daripada nilai `binlog_cache_size` transaksi penulisan akan berubah menjadi file cache binlog sementara, yang dibuat pada penyimpanan instans.
- Transaksi dapat gagal ketika penyimpanan instans penuh.

Meningkatkan performa penulisan dengan RDS Optimized Writes for MySQL

Anda dapat meningkatkan performa transaksi penulisan dengan RDS Optimized Writes for MySQL. Ketika basis data RDS for MySQL Anda menggunakan RDS Optimized Writes, RDS dapat mencapai throughput transaksi penulisan hingga dua kali lebih tinggi.

Topik

Ikhtisar RDS Optimized Writes

Ketika RDS Optimized Writes diaktifkan, basis data RDS for MySQL Anda hanya akan menulis satu kali saat melakukan flushing data ke penyimpanan yang kuat tanpa memerlukan buffer penulisan ganda. Basis data terus melindungi properti ACID untuk menghasilkan transaksi basis data yang andal dan meningkatkan performa.

Basis data relasional, seperti MySQL, menyediakan properti ACID seperti atomisitas, konsistensi, isolasi, dan daya tahan untuk transaksi basis data yang andal. Untuk membantu menyediakan properti ini, MySQL menggunakan area penyimpanan data yang disebut buffer penulisan ganda yang mencegah kesalahan penulisan sebagian halaman. Kesalahan ini terjadi ketika ada kegagalan perangkat keras saat basis data memperbarui halaman, seperti dalam kasus pemadaman listrik. Basis data MySQL dapat mendeteksi penulisan sebagian halaman dan memulihkan dengan salinan halaman di buffer penulisan ganda. Selain memberikan perlindungan, teknik ini juga menambah operasi penulisan. Untuk informasi selengkapnya tentang buffer penulisan ganda MySQL, lihat Doublewrite Buffer dalam dokumentasi MySQL.

Saat RDS Optimized Writes diaktifkan, basis data RDS for MySQL hanya menulis satu kali saat melakukan flushing data ke penyimpanan yang kuat tanpa menggunakan buffer penulisan ganda. RDS Optimized Writes berguna jika Anda menjalankan beban kerja penulisan berat pada basis data RDS for MySQL. Contoh basis data dengan beban kerja penulisan berat antara lain basis data yang mendukung pembayaran digital, perdagangan finansial, dan aplikasi game.

basis data ini dijalankan pada kelas instans DB yang menggunakan AWS Nitro System. Karena konfigurasi perangkat keras dalam sistem ini, basis data dapat menulis halaman 16-KiB secara langsung ke file data secara andal dan kuat dalam satu langkah. AWS Nitro System mendukung RDS Optimized Writes.

Anda dapat mengatur `rds.optimized_writes` parameter basis data baru untuk mengontrol fitur RDS Optimized Writes untuk basis data RDS for MySQL. Akses parameter ini dalam grup parameter DB untuk RDS for MySQL versi 8.0. Tetapkan parameter menggunakan nilai berikut:

- `AUTO` – Aktifkan RDS Optimized Writes jika didukung oleh basis data. Nonaktifkan RDS Optimized Writes jika tidak didukung basis data. Ini adalah pengaturan default.
- `OFF` – Nonaktifkan RDS Optimized Writes meski didukung oleh basis data.

Jika Anda sudah memiliki basis data dengan versi mesin, kelas instans DB, dan/atau format sistem file yang tidak mendukung RDS Optimized Writes, Anda bisa mengaktifkan fitur ini dengan membuat deployment blue/green. Untuk informasi selengkapnya, lihat the section called “Pengaktifan pada basis data yang sudah ada”.

Jika Anda memigrasikan basis data RDS for MySQL yang dikonfigurasi untuk menggunakan RDS Optimized Writes ke kelas instans DB yang tidak mendukung fitur tersebut, RDS secara otomatis menonaktifkan RDS Optimized Writes untuk basis data tersebut.

Saat RDS Optimized Writes dinonaktifkan, basis data akan menggunakan buffer penulisan ganda MySQL.

Untuk menentukan apakah basis data RDS for MySQL menggunakan RDS Optimized Writes, lihat nilai parameter `innodb_doublewrite` saat ini untuk basis data. Jika basis data menggunakan RDS Optimized Writes, parameter ini diatur ke `FALSE` (`0`).

Menggunakan RDS Optimized Writes

Anda dapat mengaktifkan RDS Optimized Writes saat membuat basis data RDS for MySQL dengan konsol RDS, AWS CLI, atau RDS API. RDS Optimized Writes diaktifkan secara otomatis ketika kedua kondisi berikut terpenuhi selama pembuatan basis data:

- Anda menentukan versi mesin DB dan kelas instans DB yang mendukung RDS Optimized Writes.
  - RDS Optimized Writes didukung untuk RDS for MySQL versi 8.0.30 dan yang lebih tinggi. Untuk informasi tentang versi RDS for MySQL, lihat Versi MySQL di Amazon RDS.

- RDS Optimized Writes didukung untuk basis data RDS for MySQL yang menggunakan kelas instans DB berikut:
  - db.m7g
  - db.m6g
  - db.m6gd
  - db.m6i
  - db.m5
  - db.m5d
  - db.r7g
  - db.r6g
  - db.r6gd
  - db.r6i
  - db.r5
  - db.r5b
  - db.r5d
  - db.x2idn
  - db.x2iedn

  Untuk informasi tentang kelas instans DB, lihat the section called “Kelas instans DB”.

  Ketersediaan kelas instans DB untuk Wilayah AWS berbeda-beda. Untuk mengetahui dukungan kelas instans DB pada suatu Wilayah AWS, lihat the section called “Menentukan dukungan kelas instans DB di Wilayah AWS”.

  Untuk meningkatkan basis data Anda ke kelas instans DB yang mendukung RDS Optimized Writes, Anda bisa membuat deployment blue/green. Untuk informasi selengkapnya, lihat the section called “Pengaktifan pada basis data yang sudah ada”.
- Dalam grup parameter yang terkait dengan basis data, parameter `rds.optimized_writes` diatur ke `AUTO`. Dalam grup parameter default, parameter ini selalu diatur ke `AUTO`.

Jika Anda ingin menggunakan versi mesin DB dan kelas instans DB yang mendukung RDS Optimized Writes, tetapi tidak ingin menggunakan fitur ini, tentukan grup parameter khusus saat Anda membuat basis data. Dalam grup parameter ini, atur parameter `rds.optimized_writes` ke `OFF`. Agar nantinya basis data menggunakan RDS Optimized Writes, Anda dapat mengatur parameter

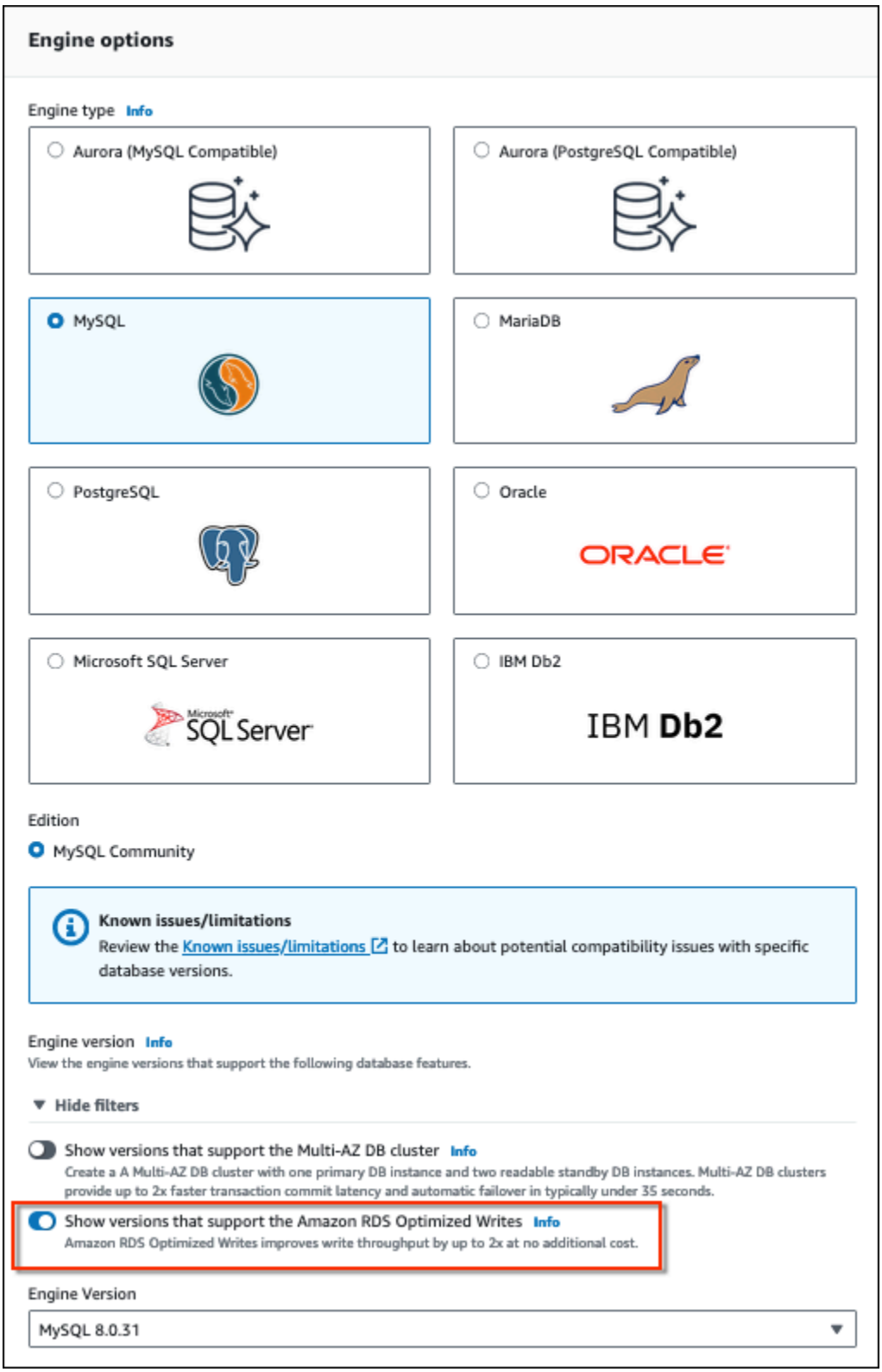
ke AUTO untuk mengaktifkannya. Untuk informasi tentang pembuatan grup parameter khusus dan pengaturan parameter, lihat Bekerja dengan grup parameter.

Untuk informasi tentang pembuatan instans DB, lihat Membuat instans DB Amazon RDS.

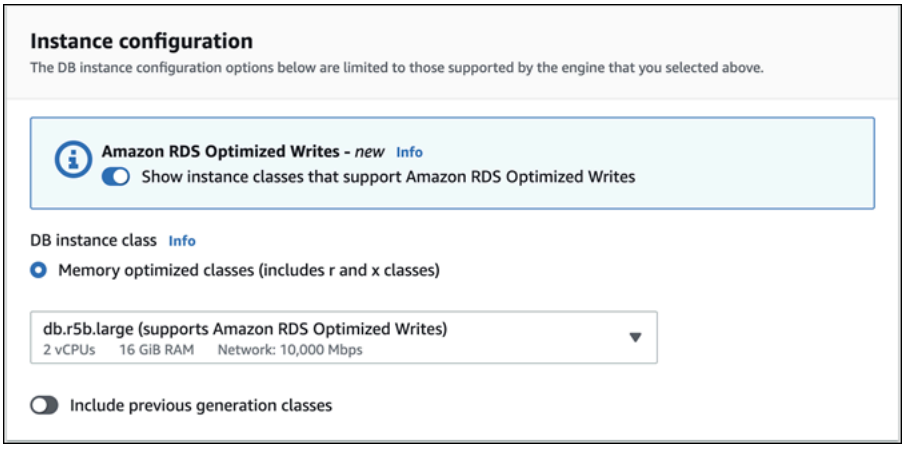
Konsol

Saat menggunakan konsol RDS untuk membuat basis data RDS for MySQL, Anda dapat memfilter versi mesin DB dan kelas instans DB yang mendukung RDS Optimized Writes. Setelah mengaktifkan filter, Anda dapat memilih versi mesin DB dan kelas instans DB yang tersedia.

Untuk memilih versi mesin DB yang mendukung RDS Optimized Writes, filter versi mesin DB RDS for MySQL yang mendukungnya di Versi mesin, lalu pilih versi.



Di bagian Konfigurasi instans, gunakan filter untuk menemukan kelas instans DB yang mendukung RDS Optimized Writes, lalu pilih kelas instans DB.



Setelah menentukan pilihan ini, Anda dapat memilih pengaturan lain sesuai kebutuhan dan menyelesaikan pembuatan basis data RDS for MySQL dengan konsol.

AWS CLI

Untuk membuat instance DB dengan menggunakanAWS CLI, gunakan create-db-instanceperintah. Pastikan nilai `--engine-version` dan `--db-instance-class` mendukung RDS Optimized Writes. Selain itu, pastikan parameter `rds.optimized_writes` untuk grup parameter yang terkait dengan instans DB telah diatur ke AUT0. Contoh ini mengaitkan grup parameter default dengan instans DB.

Example Membuat instans DB yang menggunakan RDS Optimized Writes

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --engine mysql \
    --engine-version 8.0.30 \
    --db-instance-class db.r5b.large \
    --manage-master-user-password \
    --master-username admin \
    --allocated-storage 200
```

Untuk Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --engine mysql ^
```

```
    --engine-version 8.0.30 ^
    --db-instance-class db.r5b.large ^
    --manage-master-user-password ^
    --master-username admin ^
    --allocated-storage 200
```

API RDS

Anda dapat membuat instans DB menggunakan operasi CreateDBInstance. Saat Anda menggunakan operasi ini, pastikan nilai `EngineVersion` dan `DBInstanceClass` mendukung RDS Optimized Writes. Selain itu, pastikan parameter `rds.optimized_writes` untuk grup parameter yang terkait dengan instans DB telah diatur ke `AUTO`.

Mengaktifkan RDS Optimized Writes pada basis data yang sudah ada

Untuk mengubah basis data RDS for MySQL yang sudah ada untuk mengaktifkan RDS Optimized Writes, basis data harus dibuat dengan versi mesin DB dan kelas instans DB yang didukung. Selain itu, basis data harus sudah dibuat setelah RDS Optimized Writes dirilis pada 27 Nopember 2022, karena konfigurasi sistem file yang diperlukan tidak kompatibel dengan basis data yang dibuat sebelum dirilis. Jika kondisi ini terpenuhi, Anda dapat mengaktifkan RDS Optimized Writes dengan mengatur parameter `rds.optimized_writes` ke `AUTO`.

Jika basis data Anda tidak dibuat dengan versi mesin, kelas instans, atau konfigurasi sistem file yang didukung, Anda dapat menggunakan Deployment Blue/Green RDS untuk bermigrasi ke konfigurasi yang didukung. Sambil membuat deployment blue/green, lakukan hal berikut:

- Pilih Aktifkan Optimized Writes pada basis data hijau, lalu tentukan versi mesin dan kelas instans DB yang mendukung RDS Optimized Writes. Untuk daftar versi mesin dan kelas instans yang didukung, lihat Menggunakan RDS Optimized Writes.
- Di bagian Penyimpanan, pilih Tingkatkan konfigurasi sistem file penyimpanan. Opsi ini meningkatkan basis data ke konfigurasi sistem file dasar yang kompatibel.

Saat Anda membuat deployment blue/green, jika parameter `rds.optimized_writes` diatur ke `AUTO`, RDS Optimized Writes akan secara otomatis diaktifkan pada lingkungan hijau. Anda kemudian dapat beralih antara deployment blue/green, yang mendukung lingkungan hijau sebagai lingkungan produksi yang baru.

Untuk informasi selengkapnya, lihat the section called “Membuat deployment blue/green”.

Batasan RDS Optimized Writes

Saat Anda memulihkan basis data RDS for MySQL dari snapshot, Anda hanya bisa mengaktifkan RDS Optimized Writes untuk basis data jika semua kondisi berikut terpenuhi:

- Snapshot dibuat dari basis data yang mendukung RDS Optimized Writes.
- Snapshot dibuat dari basis data yang dibuat setelah RDS Optimized Writes dirilis.
- Snapshot dikembalikan ke basis data yang mendukung RDS Optimized Writes.
- basis data yang dipulihkan berkaitan dengan grup parameter yang parameter `rds.optimized_writes`-nya diatur ke `AUTO`.

Meng-upgrade mesin DB MySQL

Ketika Amazon RDS mendukung versi baru mesin basis data, Anda dapat meng-upgrade instans DB Anda ke versi baru. Ada dua jenis upgrade untuk basis data MySQL: upgrade versi mayor dan upgrade versi minor.

Upgrade versi mayor

Upgrade versi mayor dapat berisi perubahan basis data yang tidak memiliki kompatibilitas mundur dengan aplikasi yang ada. Oleh karena itu, Anda harus melakukan upgrade versi mayor untuk instans DB Anda secara manual. Anda dapat memulai upgrade versi mayor dengan mengubah instans DB Anda. Sebelum Anda melakukan upgrade versi mayor, kami sarankan agar Anda mengikuti petunjuk dalam Upgrade versi mayor untuk MySQL.

Untuk upgrade versi mayor terhadap deployment instans DB Multi-AZ, Amazon RDS secara bersamaan meng-upgrade replika utama dan siaga. Instans DB Anda tidak akan tersedia hingga upgrade selesai. Saat ini, Amazon RDS tidak mendukung upgrade versi mayor untuk deployment klaster DB Multi-AZ.

> **Tip**
>
> Anda dapat meminimalkan waktu henti yang diperlukan untuk upgrade versi mayor dengan menggunakan deployment blue/green. Untuk informasi selengkapnya, lihat Menggunakan Deployment Blue/Green Amazon RDS untuk pembaruan basis data.

Upgrade versi minor

Upgrade versi minor hanya mencakup perubahan yang memiliki kompatibilitas mundur dengan aplikasi yang ada. Anda dapat memulai upgrade versi minor secara manual dengan memodifikasi instans DB Anda. Atau Anda dapat mengaktifkan opsi Peningkatan versi minor otomatis saat membuat atau memodifikasi instans DB. Tindakan ini akan membuat Amazon RDS secara otomatis meng-upgrade instans DB Anda setelah menguji dan menyetujui versi baru. Untuk informasi tentang melakukan upgrade, lihat Meng-upgrade versi mesin instans DB.

Saat Anda melakukan upgrade versi minor klaster DB Multi-AZ, Amazon RDS meng-upgrade instans DB pembaca satu per satu. Kemudian, salah satu instans DB pembaca beralih menjadi instans DB penulis baru. Amazon RDS kemudian meng-upgrade instans penulis lama (yang sekarang menjadi instans pembaca).

> **Note**
>
> Waktu henti untuk upgrade versi minor deployment instans DB Multi-AZ dapat berlangsung selama beberapa menit. Klaster DB Multi-AZ biasanya mengurangi waktu henti upgrade versi minor menjadi sekitar 35 detik. Saat digunakan dengan Proksi RDS, Anda dapat mengurangi waktu henti menjadi satu detik atau kurang. Untuk informasi selengkapnya, lihat Menggunakan Proksi RDS. Sebagai alternatif, Anda dapat menggunakan proxy database open source seperti ProxySQL, PgBouncer, atau Driver AWS JDBC untuk MySQL.

Jika instans DB MySQL Anda menggunakan replika baca, maka Anda harus meng-upgrade semua replika baca sebelum meng-upgrade instans sumber.

Topik

Gambaran umum upgrade

Saat Anda menggunakan AWS Management Console untuk memutakhirkan instans DB, ini menunjukkan target pemutakhiran yang valid untuk instans DB. Anda juga dapat menggunakan AWS CLI perintah berikut untuk mengidentifikasi target pemutakhiran yang valid untuk instans DB:

Untuk Linux, macOS, atau Unix:

```
aws rds describe-db-engine-versions \
  --engine mysql \
  --engine-version version-number \
```

```
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Untuk Windows:

```
aws rds describe-db-engine-versions ^
  --engine mysql ^
  --engine-version version-number ^
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Misalnya, untuk mengidentifikasi target pemutakhiran yang valid untuk instance MySQL versi 8.0.28 DB, jalankan perintah berikut: AWS CLI

Untuk Linux, macOS, atau Unix:

```
aws rds describe-db-engine-versions \
  --engine mysql \
  --engine-version 8.0.28 \
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Untuk Windows:

```
aws rds describe-db-engine-versions ^
  --engine mysql ^
  --engine-version 8.0.28 ^
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Amazon RDS mengambil dua atau lebih snapshot DB selama proses upgrade. Amazon RDS mengambil hingga dua snapshot dari instans DB sebelum melakukan perubahan upgrade. Jika upgrade tidak berfungsi untuk basis data Anda, Anda dapat memulihkan salah satu snapshot ini untuk membuat instans DB yang menjalankan versi lama. Amazon RDS mengambil snapshot lain dari instans DB saat upgrade selesai. Amazon RDS mengambil snapshot ini terlepas dari apakah AWS Backup mengelola cadangan untuk instans DB.

> **Note**
>
> Amazon RDS hanya mengambil snapshot DB jika Anda telah mengatur periode retensi cadangan untuk instans DB Anda ke angka yang lebih besar dari 0. Untuk mengubah periode retensi cadangan Anda, lihat Memodifikasi instans DB Amazon RDS.

Setelah upgrade selesai, Anda tidak dapat kembali ke versi mesin basis data sebelumnya. Jika Anda ingin kembali ke versi yang lebih lama, pulihkan snapshot DB pertama yang diambil untuk membuat instans DB baru.

Anda mengontrol waktu untuk meng-upgrade instans DB Anda ke versi baru yang didukung oleh Amazon RDS. Tingkat kontrol ini membantu Anda menjaga kompatibilitas dengan versi basis data spesifik dan menguji versi baru untuk aplikasi Anda sebelum menerapkannya dalam produksi. Saat Anda siap, Anda dapat melakukan upgrade versi pada waktu yang paling cocok dengan jadwal Anda.

Jika instans DB Anda menggunakan replikasi baca, maka Anda harus meng-upgrade semua replika baca sebelum meng-upgrade instans sumber.

Nomor versi MySQL

Urutan penomoran versi untuk RDS untuk mesin database MySQL baik dalam bentuk major.minor.patch.yyyymmdd atau major.minor.patch, misalnya, 8.0.33.R2.20231201 atau 5.7.44. Format yang digunakan tergantung pada versi mesin MySQL. Untuk informasi tentang penomoran versi RDS Extended Support, lihat. Penamaan versi Amazon RDS Extended Support

utama

Nomor versi utama adalah bilangan bulat dan bagian pecahan pertama dari nomor versi, misalnya, 8.0. Upgrade versi mayor akan meningkatkan bagian mayor dari nomor versi. Misalnya, upgrade dari 5.7 .44 ke 8.0.33 adalah upgrade versi utama, di mana 5.7 dan 8.0 adalah nomor versi utama.

kecil

Nomor versi minor adalah bagian ketiga dari nomor versi, misalnya, 33 di 8.0.33.

tambalan

Patch adalah bagian keempat dari nomor versi, misalnya, R2 di 8.0.33.R2. Versi patch RDS mencakup perbaikan bug penting yang ditambahkan ke versi minor setelah dirilis.

YYYMMDD

Tanggal adalah bagian kelima dari nomor versi, misalnya, 20231201 di 8.0.33.R2.20231201. Versi tanggal RDS adalah patch keamanan yang mencakup perbaikan keamanan penting yang ditambahkan ke versi minor setelah dirilis. Itu tidak termasuk perbaikan apa pun yang mungkin mengubah perilaku mesin.

Versi utama	Versi minor	Skema penamaan
8.0	≥ 33	Instans DB baru menggunakan major.minor.patch.YYMMDD, misalnya, 8.0.33.R2.20231201. Instans DB yang ada mungkin menggunakan major.minor.patch, misalnya, 8.0.33.R2, hingga versi mayor atau minor Anda berikutnya ditingkatkan.
	< 33	Instans DB yang ada menggunakan major.minor.patch, misalnya, 8.0.32.R2.
5.7	≥ 42	Instans DB baru menggunakan major.minor.patch.YYMMDD, misalnya, 5.7.42.R2.20231201. Instans DB yang ada mungkin menggunakan major.minor.patch, misalnya, 5.7.42.R2, hingga versi mayor atau minor Anda berikutnya ditingkatkan.

Nomor versi RDS

Nomor versi RDS menggunakan skema *major*.*minor*.*patch* atau *major*.*minor*.*patch*.*YYYYMMDD* penamaan. Versi patch RDS mencakup perbaikan bug penting yang ditambahkan ke versi minor setelah dirilis. Versi tanggal RDS (*YYMMDD*) adalah patch keamanan. Patch keamanan tidak menyertakan perbaikan apa pun yang dapat mengubah perilaku

mesin. Untuk informasi tentang penomoran versi RDS Extended Support, lihat. Penamaan versi Amazon RDS Extended Support

Untuk mengidentifikasi nomor versi Amazon RDS untuk basis data Anda, Anda harus terlebih dahulu membuat ekstensi `rds_tools` dengan menggunakan perintah berikut:

```
CREATE EXTENSION rds_tools;
```

Anda dapat mengetahui nomor versi RDS dari database RDS untuk MySQL Anda dengan kueri SQL berikut:

```
mysql> select mysql.rds_version();
```

Misalnya, query RDS untuk MySQL 8.0.34 database mengembalikan output berikut:

```
+---------------------+
| mysql.rds_version() |
+---------------------+
| 8.0.34.R2.20231201  |
+---------------------+
1 row in set (0.01 sec)
```

Upgrade versi mayor untuk MySQL

Amazon RDS mendukung upgrade di tempat berikut untuk versi mayor mesin basis data MySQL:

- MySQL 5.6 ke MySQL 5.7
- MySQL 5.7 ke MySQL 8.0

> **Note**
>
> Anda hanya dapat membuat instans DB MySQL versi 5.7 dan 8.0 dengan kelas instans DB generasi terbaru dan generasi saat ini, selain kelas instans DB db.m3 generasi sebelumnya. Dalam beberapa kasus, Anda sebaiknya meng-upgrade instans DB MySQL versi 5.6 yang berjalan pada kelas instans DB generasi sebelumnya (selain db.m3) ke instans DB MySQL versi 5.7. Dalam hal ini, pertama-tama modifikasi instans DB untuk menggunakan kelas instans DB generasi terbaru atau generasi saat ini. Setelah Anda melakukannya, Anda lalu

dapat memodifikasi instans DB untuk menggunakan mesin basis data MySQL versi 5.7. Untuk informasi kelas instans DB Amazon RDS, lihat Kelas instans DB .

Topik

- Gambaran umum upgrade versi mayor MySQL
- Upgrade ke MySQL versi 5.7 mungkin berjalan lambat
- Pra-pemeriksaan untuk upgrade dari MySQL 5.7 ke 8.0
- Rollback setelah kegagalan untuk meng-upgrade dari MySQL 5.7 ke 8.0

Gambaran umum upgrade versi mayor MySQL

Upgrade versi mayor dapat berisi perubahan basis data yang tidak memiliki kompatibilitas mundur dengan aplikasi yang ada. Akibatnya, Amazon RDS tidak menerapkan upgrade versi mayor secara otomatis; Anda harus memodifikasi instans DB secara manual. Kami menyarankan Anda untuk menguji upgrade apa pun secara menyeluruh sebelum menerapkannya ke instans produksi Anda.

Untuk melakukan upgrade versi mayor untuk instans DB MySQL versi 5.6 pada Amazon RDS ke MySQL versi 5.7 atau lebih baru, lakukan pembaruan OS yang tersedia terlebih dahulu. Setelah pembaruan OS selesai, upgrade ke masing-masing versi mayor: 5.6 ke 5.7, lalu 5.7 ke 8.0. Instans DB MySQL yang dibuat sebelum tanggal 24 April 2014 akan menunjukkan pembaruan OS yang tersedia hingga pembaruan ini diterapkan. Untuk informasi selengkapnya tentang pembaruan OS, lihat Menerapkan pembaruan untuk instans DB.

Selama upgrade versi mayor MySQL, Amazon RDS menjalankan `mysql_upgrade` biner MySQL untuk meng-upgrade tabel, jika perlu. Selain itu, Amazon RDS mengosongkan tabel `slow_log` dan `general_log` selama upgrade versi mayor. Untuk mempertahankan informasi log, simpan konten log sebelum upgrade versi mayor.

Upgrade versi mayor MySQL biasanya selesai dalam waktu sekitar 10 menit. Beberapa upgrade mungkin memakan waktu lebih lama karena ukuran kelas instans DB atau karena instans tidak mengikuti panduan operasional tertentu dalam Praktik terbaik untuk Amazon RDS. Jika Anda meng-upgrade instans DB dari konsol Amazon RDS, status instans DB menunjukkan waktu upgrade selesai. Jika Anda meng-upgrade menggunakan AWS Command Line Interface (AWS CLI), gunakan describe-db-instancesperintah dan periksa `Status` nilainya.

Upgrade ke MySQL versi 5.7 mungkin berjalan lambat

MySQL versi 5.6.4 memperkenalkan format tanggal dan waktu baru untuk kolom `datetime`, `time`, dan `timestamp` yang memungkinkan komponen pecahan dalam nilai tanggal dan waktu. Saat meng-upgrade instans DB ke MySQL versi 5.7, MySQL memaksa konversi semua jenis kolom tanggal dan waktu ke format baru.

Karena konversi ini membuat ulang tabel Anda, mungkin perlu waktu yang cukup lama untuk menyelesaikan upgrade instans DB. Konversi paksa terjadi untuk setiap instans DB yang menjalankan versi sebelum MySQL versi 5.6.4. Hal ini juga terjadi untuk setiap instans DB yang di-upgrade dari versi sebelum MySQL versi 5.6.4 ke versi selain 5.7.

Jika instans DB Anda menjalankan versi sebelum MySQL versi 5.6.4, atau di-upgrade dari versi sebelum 5.6.4, kami merekomendasikan sebuah langkah tambahan. Dalam kasus ini, kami menyarankan Anda untuk mengonversi kolom `datetime`, `time`, dan `timestamp` di basis data Anda sebelum meng-upgrade instans DB Anda ke MySQL versi 5.7. Konversi ini dapat secara signifikan mengurangi jumlah waktu yang diperlukan untuk meng-upgrade instans DB ke MySQL versi 5.7. Untuk meng-upgrade kolom tanggal dan waktu Anda ke format baru, berikan perintah `ALTER TABLE <table_name> FORCE;` untuk setiap tabel yang berisi kolom tanggal dan waktu. Karena mengubah tabel akan mengunci tabel sebagai hanya baca, kami menyarankan Anda melakukan pembaruan ini selama periode pemeliharaan.

Untuk menemukan semua tabel di basis data Anda yang memiliki `datetime`, `time`, atau `timestamp` dan membuat `ALTER TABLE <table_name> FORCE;` perintah untuk setiap tabel, gunakan kueri berikut.

```
SET show_old_temporals = ON;
   SELECT table_schema, table_name,column_name, column_type
   FROM information_schema.columns
   WHERE column_type LIKE '%/* 5.5 binary format */';
   SET show_old_temporals = OFF;
```

Pra-pemeriksaan untuk upgrade dari MySQL 5.7 ke 8.0

MySQL 8.0 menyertakan sejumlah inkompatibilitas dengan MySQL 5.7. Inkompatibilitas ini dapat menyebabkan masalah selama upgrade dari MySQL 5.7 ke MySQL 8.0. Jadi, beberapa persiapan mungkin diperlukan di basis data Anda agar upgrade berhasil. Berikut ini adalah daftar umum inkompatibilitas tersebut:

- Tidak boleh ada tabel yang menggunakan jenis atau fungsi data yang usang.

- Tidak boleh ada file orphan *.frm.
- Pemicu tidak boleh memiliki pendefinisi yang hilang atau kosong atau konteks pembuatan yang tidak valid.
- Tidak boleh ada tabel partisi yang menggunakan mesin penyimpanan yang tidak memiliki dukungan partisi native.
- Tidak boleh ada pelanggaran kata kunci atau kata yang dicadangkan. Mungkin ada beberapa kata kunci yang digunakan sistem di MySQL 8.0 yang sebelumnya tidak digunakan sistem.

  Untuk informasi selengkapnya, lihat Keywords and reserved words dalam dokumentasi MySQL.
- Tidak boleh ada tabel di basis data sistem `mysql` MySQL 5.7 yang memiliki nama yang sama dengan tabel yang digunakan oleh kamus data MySQL 8.0.
- Tidak boleh ada mode SQL usang yang ditentukan dalam pengaturan variabel sistem `sql_mode` Anda.
- Tidak boleh ada tabel atau prosedur tersimpan dengan elemen kolom individu `ENUM` atau `SET` dengan panjang melebihi 255 karakter atau 1020 byte.
- Sebelum melakukan upgrade ke MySQL 8.0.13 atau lebih tinggi, tidak boleh ada partisi tabel yang berada di ruang tabel InnoDB yang dibagikan.
- Tidak boleh ada kueri dan definisi program tersimpan dari MySQL 8.0.12 atau lebih rendah yang menggunakan pengkualifikasi `ASC` atau `DESC` untuk klausa `GROUP BY`.
- Penginstalan MySQL 5.7 Anda tidak boleh menggunakan fitur yang tidak didukung di MySQL 8.0.

  Untuk informasi selengkapnya, lihat Features removed in MySQL 8.0 dalam dokumentasi MySQL.
- Tidak boleh ada nama batasan kunci asing yang lebih panjang dari 64 karakter.
- Untuk peningkatan dukungan Unicode, pertimbangkan untuk mengonversi objek yang menggunakan charset `utf8mb3` agar menggunakan charset `utf8mb4`. Set karakter `utf8mb3` sudah tidak digunakan lagi. Selain itu, pertimbangkan untuk menggunakan `utf8mb4` untuk referensi set karakter, bukan `utf8`, karena saat ini `utf8` adalah alias untuk set karakter `utf8mb3`.

  Untuk informasi selengkapnya, lihat The utf8mb3 character set (3-byte UTF-8 unicode encoding) dalam dokumentasi MySQL.

Saat Anda memulai upgrade dari MySQL 5.7 ke 8.0, Amazon RDS menjalankan pra-pemeriksaan secara otomatis untuk mendeteksi inkompatibilitas ini. Untuk informasi tentang upgrade ke MySQL 8.0, lihat Upgrading MySQL dalam dokumentasi MySQL.

Pra-pemeriksaan ini wajib dilakukan. Anda tidak dapat memilih untuk melewatinya. Pra-pemeriksaan menyediakan manfaat berikut:

- Hal ini memungkinkan Anda menghindari waktu henti yang tidak direncanakan selama upgrade.
- Jika ada inkompatibilitas, Amazon RDS mencegah upgrade dan menyediakan log bagi Anda untuk mempelajarinya. Kemudian, Anda dapat menggunakan log ini untuk menyiapkan basis data Anda untuk upgrade ke MySQL 8.0 dengan mengurangi inkompatibilitas. Untuk informasi terperinci tentang cara mengatasi inkompatibilitas, lihat Preparing your installation for upgrade dalam dokumentasi MySQL dan Upgrading to MySQL 8.0? Inilah yang perlu Anda ketahui... di Blog MySQL Server.

Sebagian pra-pemeriksaan disertakan dengan MySQL dan sebagian lainnya dibuat secara khusus oleh tim Amazon RDS. Untuk informasi tentang pra-pemeriksaan yang disediakan oleh MySQL, lihat Upgrade checker utility.

Pra-pemeriksaan berjalan sebelum instans DB dihentikan untuk upgrade, sehingga instans tersebut tidak akan menyebabkan waktu henti ketika berjalan. Jika pra-pemeriksaan menemukan inkompatibilitas, Amazon RDS secara otomatis membatalkan upgrade sebelum instans DB dihentikan. Amazon RDS juga menghasilkan peristiwa untuk inkompatibilitas. Untuk informasi selengkapnya tentang peristiwa Amazon RDS, lihat Menggunakan pemberitahuan peristiwa Amazon RDS.

Amazon RDS mencatat informasi terperinci tentang setiap inkompatibilitas dalam file log `PrePatchCompatibility.log`. Dalam kebanyakan kasus, entri log berisi tautan ke dokumentasi MySQL untuk mengoreksi inkompatibilitas. Untuk informasi selengkapnya tentang format file log, lihat Melihat dan mencantumkan file log basis data.

Karena sifatnya, pra-pemeriksaan akan menganalisis objek di basis data Anda. Analisis ini mengakibatkan konsumsi sumber daya dan menambah waktu penyelesaian upgrade.

> **Note**
>
> Amazon RDS menjalankan semua pra-pemeriksaan ini hanya untuk upgrade dari MySQL 5.7 ke MySQL 8.0. Untuk upgrade dari MySQL 5.6 ke MySQL 5.7, pra-pemeriksaan terbatas untuk mengonfirmasi bahwa tidak ada tabel orphan dan ada ruang penyimpanan yang cukup untuk membuat ulang tabel. Pra-pemeriksaan tidak dijalankan untuk upgrade terhadap rilis yang lebih rendah daripada MySQL 5.7.

Rollback setelah kegagalan untuk meng-upgrade dari MySQL 5.7 ke 8.0

Ketika Anda meng-upgrade instans DB dari MySQL versi 5.7 ke MySQL versi 8.0, upgrade dapat gagal. Secara khusus, upgrade dapat gagal jika kamus data memiliki inkompatibilitas yang tidak terdeteksi oleh pra-pemeriksaan. Dalam kasus ini, basis data akan gagal diaktifkan dalam versi MySQL 8.0 yang baru. Pada tahap ini, Amazon RDS melakukan rollback perubahan yang dilakukan untuk upgrade. Setelah rollback, instans DB MySQL menjalankan MySQL versi 5.7. Ketika upgrade gagal dan di-rollback, Amazon RDS menghasilkan peristiwa dengan ID peristiwa RDS-EVENT-0188.

Biasanya, upgrade gagal karena ada inkompatibilitas dalam metadata antara basis data dalam instans DB Anda dan versi MySQL target. Ketika upgrade gagal, Anda dapat melihat detail tentang inkompatibilitas ini dalam file `upgradeFailure.log`. Atasi inkompatibilitas sebelum mencoba meng-upgrade kembali.

Selama percobaan upgrade yang gagal dan rollback, instans DB Anda akan diaktifkan ulang. Perubahan parameter tertunda diterapkan selama pengaktifan ulang ini dan dipersistensi setelah rollback.

Untuk informasi tentang upgrade ke MySQL 8.0, lihat topik-topik berikut dalam dokumentasi MySQL:

- Mempersiapkan Penginstalan Anda untuk Upgrade
- Meng-upgrade ke MySQL 8.0? Inilah yang perlu Anda ketahui...

> **Note**
>
> Saat ini, rollback otomatis setelah kegagalan upgrade hanya didukung untuk upgrade versi mayor MySQL 5.7 ke 8.0.

Menguji upgrade

Sebelum Anda melakukan upgrade versi mayor pada instans DB Anda, uji basis data Anda secara menyeluruh untuk kompatibilitas dengan versi baru. Selain itu, uji secara menyeluruh semua aplikasi yang mengakses basis data untuk kompatibilitas dengan versi baru. Kami menyarankan agar Anda menggunakan prosedur berikut.

Untuk menguji upgrade versi mayor

1. Tinjau dokumentasi upgrade untuk versi baru mesin basis data untuk melihat apakah ada masalah kompatibilitas yang mungkin memengaruhi basis data atau aplikasi Anda:
   - Perubahan dalam MySQL 5.6
   - Perubahan dalam MySQL 5.7
   - Perubahan dalam MySQL 8.0
2. Jika instans DB Anda adalah anggota dari grup parameter DB kustom, buat grup parameter DB baru dengan pengaturan Anda saat ini yang kompatibel dengan versi mayor yang baru. Tentukan grup parameter DB baru saat Anda meng-upgrade instans uji Anda, sehingga pengujian upgrade Anda memastikan bahwa instans ini berfungsi dengan benar. Untuk informasi selengkapnya tentang cara membuat grup parameter DB, lihat Bekerja dengan grup parameter.
3. Buat snapshot DB dari instans DB yang akan di-upgrade. Untuk informasi selengkapnya, lihat Membuat snapshot DB untuk instans DB Single-AZ.
4. Pulihkan snapshot DB untuk membuat instans DB uji baru. Untuk informasi selengkapnya, lihat Memulihkan dari snapshot DB.
5. Modifikasi instans DB uji baru ini untuk di-upgrade ke versi baru menggunakan salah satu metode yang dijelaskan sebagai berikut. Jika Anda membuat grup parameter baru di langkah 2, tentukan grup parameter tersebut.
6. Evaluasi penyimpanan yang digunakan oleh instans yang di-upgrade untuk menentukan apakah upgrade memerlukan penyimpanan tambahan.
7. Jalankan pengujian jaminan kualitas terhadap instans DB yang di-upgrade sebanyak yang diperlukan untuk memastikan bahwa basis data dan aplikasi Anda berfungsi baik dengan versi baru. Terapkan setiap pengujian baru yang diperlukan untuk mengevaluasi dampak dari masalah kompatibilitas yang Anda identifikasi dalam langkah 1. Uji semua prosedur tersimpan dan fungsi. Arahkan versi pengujian aplikasi Anda ke instans DB yang di-upgrade.
8. Jika semua pengujian berhasil, maka lakukan upgrade pada instans DB produksi Anda. Kami menyarankan agar Anda tidak mengizinkan operasi tulis ke instans DB hingga Anda mengonfirmasi bahwa semuanya berfungsi dengan benar.

Meng-upgrade instans DB MySQL

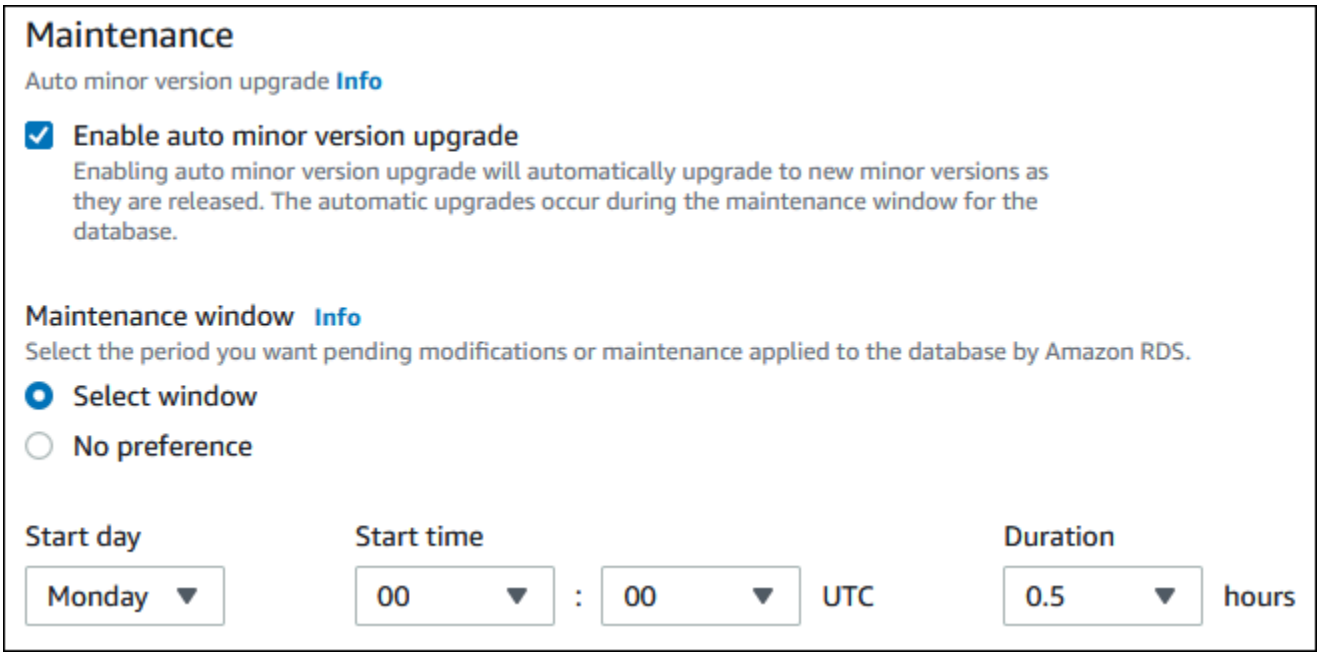
Untuk informasi tentang upgrade instans DB MySQL secara manual atau otomatis, lihat Meng-upgrade versi mesin instans DB.

Upgrade versi minor otomatis untuk MySQL

Jika Anda menentukan pengaturan berikut saat membuat atau memodifikasi instans DB, Anda dapat melakukan upgrade instans DB secara otomatis.

- Pengaturan Peningkatan versi minor otomatis diaktifkan.
- Pengaturan Periode retensi cadangan lebih besar dari 0.

Di AWS Management Console, pengaturan ini berada di bawah Konfigurasi tambahan. Gambar berikut menunjukkan pengaturan Peningkatan versi minor otomatis.



Untuk informasi selengkapnya tentang pengaturan ini, lihat Pengaturan untuk instans DB.

Untuk beberapa RDS untuk versi utama MySQL di Wilayah AWS beberapa, satu versi minor ditunjuk oleh RDS sebagai versi upgrade otomatis. Setelah versi minor diuji dan disetujui oleh Amazon RDS, upgrade versi minor terjadi secara otomatis selama periode pemeliharaan Anda. RDS tidak secara otomatis menetapkan versi minor yang lebih baru sebagai versi upgrade otomatis. Sebelum RDS menetapkan versi upgrade otomatis yang lebih baru, beberapa kriteria dipertimbangkan, seperti yang berikut ini:

- Masalah keamanan yang diketahui
- Bug dalam versi komunitas MySQL
- Stabilitas armada secara keseluruhan sejak versi minor dirilis

Anda dapat menggunakan AWS CLI perintah berikut untuk menentukan versi target pemutakhiran minor otomatis saat ini untuk versi minor MySQL tertentu secara spesifik. Wilayah AWS

Untuk Linux, macOS, atau Unix:

```
aws rds describe-db-engine-versions \
--engine mysql \
--engine-version minor-version \
--region region \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
--output text
```

Untuk Windows:

```
aws rds describe-db-engine-versions ^
--engine mysql ^
--engine-version minor-version ^
--region region ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output text
```

Misalnya, AWS CLI perintah berikut menentukan target upgrade minor otomatis untuk MySQL minor versi 8.0.11 di AS Timur (Ohio) (us-timur-2). Wilayah AWS

Untuk Linux, macOS, atau Unix:

```
aws rds describe-db-engine-versions \
--engine mysql \
--engine-version 8.0.11 \
--region us-east-2 \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
--output table
```

Untuk Windows:

```
aws rds describe-db-engine-versions ^
--engine mysql ^
--engine-version 8.0.11 ^
--region us-east-2 ^
```

```
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output table
```

Output Anda akan seperti yang berikut ini.

```
---------------------------------
|    DescribeDBEngineVersions    |
+--------------+-----------------+
|  AutoUpgrade |  EngineVersion  |
+--------------+-----------------+
|  False       |  8.0.15         |
|  False       |  8.0.16         |
|  False       |  8.0.17         |
|  False       |  8.0.19         |
|  False       |  8.0.20         |
|  False       |  8.0.21         |
|  True        |  8.0.23         |
|  False       |  8.0.25         |
+--------------+-----------------+
```

Dalam contoh ini, nilai `AutoUpgrade` adalah `True` untuk MySQL versi 8.0.23. Jadi, target upgrade minor otomatis adalah MySQL versi 8.0.23, yang disorot dalam output.

Instans DB MySQL secara otomatis di-upgrade selama periode pemeliharaan Anda jika kriteria berikut terpenuhi:

- Pengaturan Peningkatan versi minor otomatis diaktifkan.
- Pengaturan Periode retensi cadangan lebih besar dari 0.
- Instans DB menjalankan versi mesin DB minor yang lebih rendah dari versi minor upgrade otomatis saat ini.

Untuk informasi selengkapnya, lihat Meng-upgrade versi mesin minor secara otomatis.

Menggunakan replika baca untuk mengurangi waktu henti ketika meng-upgrade basis data MySQL

Dalam kebanyakan kasus, deployment blue/green adalah opsi terbaik untuk mengurangi waktu henti saat meng-upgrade instans MySQL DB. Untuk informasi selengkapnya, lihat Menggunakan Deployment Blue/Green Amazon RDS untuk pembaruan basis data.

Jika Anda tidak dapat menggunakan deployment blue/green dan instans DB MySQL saat ini sedang digunakan dengan aplikasi produksi, Anda dapat menggunakan prosedur berikut untuk meng-upgrade versi basis data untuk instans DB Anda. Prosedur ini dapat mengurangi jumlah waktu henti untuk aplikasi Anda.

Dengan menggunakan replika baca, Anda dapat melakukan sebagian besar langkah-langkah pemeliharaan terlebih dahulu dan meminimalkan perubahan yang diperlukan selama pemadaman sebenarnya. Dengan teknik ini, Anda dapat menguji dan mempersiapkan instans DB baru tanpa membuat perubahan pada instans DB Anda yang sudah ada.

Prosedur berikut menunjukkan contoh upgrade dari MySQL versi 5.7 ke MySQL versi 8.0. Anda dapat menggunakan langkah umum yang sama untuk upgrade ke versi mayor lainnya.

> **Note**
>
> Ketika Anda meng-upgrade dari MySQL versi 5.7 ke MySQL versi 8.0, selesaikan pra-pemeriksaan sebelum melakukan upgrade. Untuk informasi selengkapnya, lihat Pra-pemeriksaan untuk upgrade dari MySQL 5.7 ke 8.0.

Untuk meng-upgrade basis data MySQL saat instans DB sedang digunakan

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Buat sebuah replika baca dari instans DB MySQL 5.7 Anda. Proses ini membuat salinan yang dapat di-upgrade dari basis data Anda. Mungkin ada replika baca lainnya dari instans DB tersebut.
   a. Pada konsol, pilih Basis data, lalu pilih instans DB yang ingin Anda upgrade.
   b. Untuk Tindakan, pilih Buat replika baca.
   c. Berikan nilai untuk Pengidentifikasi instans DB untuk replika baca Anda dan pastikan bahwa Kelas instans DB dan pengaturan lainnya sudah sesuai dengan instans DB MySQL 5.7 Anda.
   d. Pilih Buat replika baca.
3. (Opsional) Ketika replika baca telah dibuat dan Status menunjukkan Tersedia, ubah replika baca menjadi deployment Multi-AZ dan aktifkan pencadangan.

Secara default, replika baca dibuat sebagai deployment AZ Tunggal dengan pencadangan dinonaktifkan. Karena replika baca ini akhirnya akan menjadi instans DB produksi, praktik terbaiknya adalah mengonfigurasi deployment Multi-AZ dan mengaktifkan pencadangan sekarang.

a. Pada konsol, pilih Basis data, lalu pilih replika baca yang baru saja Anda buat.

b. Pilih Ubah.

c. Untuk Deployment Multi-AZ, pilih Buat instans siaga.

d. Untuk Periode Retensi Cadangan, pilih nilai selain nol positif, misalnya 3 hari, lalu pilih Lanjutkan.

e. Untuk Penjadwalan modifikasi, pilih Terapkan segera.

f. Pilih Modifikasi instans DB.

4. Saat Status replika baca menunjukkan Tersedia, upgrade replika baca ke MySQL 8.0:

a. Pada konsol, pilih Basis data, lalu pilih replika baca yang baru saja Anda buat.

b. Pilih Ubah.

c. Untuk Versi mesin DB, pilih versi MySQL 8.0 sebagai target upgrade, lalu pilih Lanjutkan.

d. Untuk Penjadwalan modifikasi, pilih Terapkan segera.

e. Pilih Modifikasi instans DB untuk memulai upgrade.

5. Ketika pemutakhiran selesai dan Status menunjukkan Tersedia, verifikasi bahwa replika baca yang ditingkatkan adalah up-to-date dengan instans MySQL 5.7 DB sumber. Untuk memverifikasi, hubungkan ke replika baca dan jalankan perintah `SHOW REPLICA STATUS`. Jika `Seconds_Behind_Master` bidangnya`0`, maka replikasi adalah up-to-date.

> **Note**
>
> Versi sebelumnya dari MySQL menggunakan `SHOW SLAVE STATUS` bukan `SHOW REPLICA STATUS`. Jika Anda menggunakan MySQL versi sebelum 8.0.23, gunakan `SHOW SLAVE STATUS`.

6. (Opsional) Buat replika baca dari replika baca Anda.

Jika Anda ingin instans DB memiliki replika baca setelah dipromosikan menjadi instans DB mandiri, Anda dapat membuat replika baca sekarang.

a. Pada konsol, pilih Basis data, lalu pilih replika baca yang baru saja Anda upgrade.

b. Untuk Tindakan, pilih Buat replika baca.

c. Berikan nilai untuk Pengidentifikasi instans DB untuk replika baca Anda dan pastikan bahwa Kelas instans DB dan pengaturan lainnya sudah sesuai dengan instans DB MySQL 5.7 Anda.

d. Pilih Buat replika baca.

7. (Opsional) Konfigurasikan grup parameter kustom DB untuk replika baca.

   Jika Anda ingin instans DB menggunakan grup parameter kustom setelah dipromosikan menjadi instans DB mandiri, Anda dapat membuat grup parameter DB sekarang dan mengaitkannya dengan replika baca.

   a. Buat grup parameter DB kustom untuk MySQL 8.0. Untuk petunjuk, lihat Membuat grup parameter DB.

   b. Modifikasi parameter yang ingin Anda ubah dalam grup parameter DB yang baru saja Anda buat. Untuk petunjuk, lihat Memodifikasi parameter dalam grup parameter DB.

   c. Pada konsol, pilih Basis data, lalu pilih replika baca.

   d. Pilih Ubah.

   e. Untuk Grup parameter DB, pilih grup parameter DB MySQL 8.0 yang baru Anda buat, lalu pilih Lanjutkan.

   f. Untuk Penjadwalan modifikasi, pilih Terapkan segera.

   g. Pilih Modifikasi instans DB untuk memulai upgrade.

8. Jadikan replika baca MySQL 8.0 Anda sebagai instans DB mandiri.

   Important

   Saat Anda mempromosikan replika baca MySQL 8.0 Anda menjadi instans DB mandiri, replika baca ini bukan lagi merupakan replika dari instans DB MySQL 5.7 Anda. Kami sarankan Anda mempromosikan replika baca MySQL 8.0 selama periode pemeliharaan jika instans DB MySQL 5.7 sumber Anda berada dalam mode hanya baca dan semua operasi tulis ditangguhkan. Saat promosi selesai, Anda dapat mengarahkan operasi tulis Anda ke instans DB MySQL 8.0 yang telah di-upgrade untuk memastikan bahwa tidak ada operasi tulis yang hilang.
   Selain itu, sebelum mempromosikan replika baca MySQL 8.0, kami sarankan Anda menjalankan semua operasi bahasa definisi data (DDL) yang diperlukan di replika

baca MySQL 8.0 tersebut. Contohnya adalah membuat indeks. Pendekatan ini akan menghindari efek negatif pada performa replika baca MySQL 8.0 setelah dipromosikan. Untuk mempromosikan replika baca, gunakan prosedur berikut.

a. Pada konsol, pilih Basis data, lalu pilih replika baca yang baru saja Anda upgrade.
b. Untuk Tindakan, pilih Promosikan.
c. Pilih Ya untuk mengaktifkan pencadangan otomatis untuk instans replika baca. Untuk informasi selengkapnya, lihat Pengantar cadangan.
d. Pilih Lanjutkan.
e. Pilih Promosikan Replika Baca.

9. Sekarang Anda memiliki versi upgrade dari basis data MySQL Anda. Pada tahap ini, Anda dapat mengarahkan aplikasi Anda ke instans DB MySQL 8.0 yang baru.

Meningkatkan versi mesin snapshot DB MySQL

Dengan Amazon RDS, Anda dapat membuat volume penyimpanan snapshot DB untuk instans DB MySQL Anda. Saat Anda membuat snapshot DB, snapshot tersebut didasarkan pada versi mesin yang digunakan oleh instans DB Anda. Selain meningkatkan ke versi mesin DB dari instans DB Anda, Anda juga dapat meningkatkan versi mesin untuk snapshot DB Anda. Untuk RDS for MySQL, Anda dapat meningkatkan snapshot versi 5.7 ke versi 8.0. Anda dapat meningkatkan snapshot DB terenkripsi atau tidak terenkripsi.

Versi berikut mendukung peningkatan snapshot DB MySQL:

- Anda dapat meningkatkan dari snapshot RDS for MySQL versi 5.7.16 dan versi 5.7 yang lebih tinggi.
- Anda dapat meningkatkan ke snapshot RDS for MySQL versi 8.0.28 dan yang lebih tinggi, kecuali untuk versi 8.0.29, 8.0.30, dan 8.0.31.

  Anda tidak dapat meningkatkan versi 5.7.40, 5.7.41, dan 5.7.42 ke versi 8.0.28, tetapi Anda dapat meningkatkan versi ini ke versi 8.0.32 dan yang lebih tinggi.

Setelah memulihkan snapshot DB yang ditingkatkan ke versi mesin baru, uji untuk memastikan bahwa peningkatan berhasil dilakukan. Untuk informasi selengkapnya tentang peningkatan versi mayor, lihat the section called “Meng-upgrade mesin DB MySQL”. Untuk mempelajari cara memulihkan snapshot DB RDS, lihat the section called “Memulihkan dari snapshot DB”.

> **Note**
>
> Anda tidak dapat meningkatkan snapshot DB otomatis yang dibuat selama proses pencadangan otomatis.

Anda dapat memutakhirkan snapshot DB menggunakan AWS Management Console, AWS CLI, atau RDS API.

Konsol

Untuk meningkatkan snapshot DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.

2. Di panel navigasi, pilih Snapshot.
3. Pilih snapshot yang ingin Anda tingkatkan.
4. Untuk Tindakan, pilih Tingkatkan snapshot. Halaman Tingkatkan snapshot muncul.
5. Pilih Versi mesin baru yang akan menjadi target peningkatan.
6. Pilih Simpan perubahan untuk meningkatkan snapshot.

   Selama proses peningkatan, semua tindakan snapshot dinonaktifkan untuk snapshot DB ini. Selain itu, status snapshot DB berubah dari Tersedia menjadi Ditingkatkan, lalu berubah menjadi Aktif jika sudah selesai. Jika snapshot DB tidak dapat ditingkatkan karena masalah kerusakan snapshot, status berubah menjadi Tidak Tersedia. Anda tidak dapat memulihkan snapshot dari status ini.

   > **Note**
   >
   > Jika peningkatan snapshot DB gagal, snapshot di-rollback ke kondisi awal sesuai dengan versi asli.

AWS CLI

Untuk memutakhirkan snapshot DB ke versi mesin database baru, gunakan AWS CLI modify-db-snapshotperintah.

Opsi

- `--db-snapshot-identifier` – Pengidentifikasi snapshot DB yang akan ditingkatkan. Pengidentifikasi harus berupa Amazon Resource Name (ARN) yang unik. Untuk informasi selengkapnya, lihat Bekerja dengan Amazon Resource Name (ARN) di Amazon RDS.
- `--engine-version` – Versi mesin untuk meningkatkan snapshot DB.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-snapshot \
    --db-snapshot-identifier my_db_snapshot \
    --engine-version new_version
```

Untuk Windows:

```
aws rds modify-db-snapshot ^
    --db-snapshot-identifier my_db_snapshot ^
    --engine-version new_version
```

RDS API

Untuk meningkatkan snapshot DB ke versi mesin basis data baru, panggil operasi ModifyDBSnapshot RDS API.

Parameter-parameter

- `DBSnapshotIdentifier` – Pengidentifikasi untuk snapshot DB. Pengidentifikasi harus berupa Amazon Resource Name (ARN) yang unik. Untuk informasi selengkapnya, lihat Bekerja dengan Amazon Resource Name (ARN) di Amazon RDS.
- `EngineVersion` – Versi mesin untuk meningkatkan snapshot DB.

Impor data ke dalam instans DB MySQL

Anda dapat menggunakan beberapa teknik yang berbeda untuk mengimpor data ke dalam instans DB RDS for MySQL. Pendekatan terbaik bergantung pada sumber data, jumlah data, dan apakah impor dilakukan satu kali atau berkelanjutan. Jika Anda memigrasikan aplikasi bersama dengan data, pertimbangkan juga jumlah waktu henti yang Anda alami.

Ikhtisar

Temukan teknik untuk mengimpor data ke instans DB RDS for MySQL dalam tabel berikut.

Sumber	Jumlah data	Satu kali atau berkelanjutan	Waktu henti aplikasi	Teknik	Informasi lain
Basis data MySQL yang sudah ada di on-premise atau di Amazon EC2	Setiap	Satu kali	Beberapa	Buat cadangan dari basis data on-premise Anda, simpan di Amazon S3, lalu pulihkan file cadangan ke instans DB Amazon RDS baru yang menjalankan MySQL.	Memulihkan cadangan ke instans DB MySQL
Setiap basis data yang sudah ada	Setiap	Satu kali atau berkelanjutan	Minimal	Gunakan AWS Database Migration Service untuk memigrasikan database dengan downtime minimal dan, untuk banyak mesin DB database, lanjutkan replikasi yang sedang berlangsung.	Apa itu AWS Database Migration Service dan Panduan Pengguna

Sumber	Jumlah data	Satu kali atau berkelanjutan	Waktu henti aplikasi	Teknik	Informasi lain
					Menggunakan basis data yang kompatibel dengan MySQL sebagai target untuk AWS DMS dalam AWS Database Migration Service
Instans DB MySQL yang sudah ada	Setiap	Satu kali atau berkelanjutan	Minimal	Buat replika baca untuk replikasi berkelanjutan. Dorong replika baca untuk pembuatan satu kali instans DB baru.	Menggunakan replika baca instans DB

Sumber	Jumlah data	Satu kali atau berkelanjutan	Waktu henti aplikasi	Teknik	Informasi lain
Basis data MariaDB atau MySQL yang sudah ada	Kecil	Satu kali	Beberapa	Salin data secara langsung ke instans DB MySQL Anda dengan menggunakan utilitas baris perintah.	Mengimpor data dari MariaDB eksternal atau database MySQL ke RDS untuk MariaDB atau RDS untuk MySQL DB instance
Data tidak disimpan dalam basis data yang sudah ada	Sedang	Satu kali	Beberapa	Buat file datar dan impor mereka menggunakan pernyataan MySQL`LOAD DATA LOCAL INFILE`.	Mengimpor data dari sumber mana pun ke instans DB MySQL atau MariaDB

Sumber	Jumlah data	Satu kali atau berkelanjutan	Waktu henti aplikasi	Teknik	Informasi lain
Basis data MySQL atau MariaDB yang sudah ada di on-premise atau di Amazon EC2	Setiap	Berkelanjutan	Minimal	Konfigurasikan replikasi dengan basis data MariaDB atau MySQL yang sudah ada sebagai sumber replikasi.	Mengonfigurasi replikasi posisi file log biner dengan instans sumber eksternal Mengimpor data ke basis data Amazon RDS MariaDB atau MySQL dengan lebih sedikit waktu henti

> **Note**
>
> Basis data sistem `'mysql'` berisi informasi autentikasi dan otorisasi yang dibutuhkan untuk masuk ke instans DB Anda dan mengakses data Anda. Penurunan, pengubahan, penamaan

ulang, atau penyingkatan tabel, data, atau konten lain dari basis data `'mysql'` dalam instans DB Anda dapat mengakibatkan kesalahan dan dapat menyebabkan instans DB dan data Anda tidak dapat diakses. Jika ini terjadi, Anda dapat mengembalikan instance DB dari snapshot menggunakan AWS CLI `restore-db-instance-from-db-snapshot` perintah. Anda dapat memulihkan instans DB menggunakan AWS CLI `restore-db-instance-to-point-in-time` perintah.

Impor pertimbangan data

Di bawah ini Anda dapat menemukan informasi teknis tambahan terkait pemuatan data ke dalam MySQL. Informasi ini ditujukan bagi pengguna tingkat lanjut yang sudah memahami arsitektur server MySQL.

Log biner

Beban data menimbulkan penalti performa dan memerlukan ruang disk kosong tambahan (hingga empat kali lebih banyak) jika log biner diaktifkan dibandingkan jika pemuatan data yang sama dengan log biner yang tidak diaktifkan. Keparahan penalti performa dan jumlah ruang disk kosong yang dibutuhkan berbanding lurus dengan ukuran transaksi yang digunakan untuk memuat data.

Ukuran transaksi

Ukuran transaksi memainkan peran yang penting dalam pemuatan data MySQL. Ukuran ini memiliki pengaruh besar pada konsumsi sumber daya, pemanfaatan ruang disk, kelanjutan proses, waktu untuk pemulihan, dan format input (file rata atau SQL). Bagian ini menjelaskan bagaimana ukuran transaksi dapat memengaruhi log biner dan mengakibatkan kasus penonaktifan log biner selama pemuatan data yang besar. Sebagaimana disebutkan sebelumnya, log biner diaktifkan dan dinonaktifkan dengan mengatur periode retensi cadangan otomatis Amazon RDS. Nilai non-nol mengaktifkan log biner, dan nilai nol menonaktifkannya. Kami juga akan menjelaskan dampak transaksi yang besar pada InnoDB dan mengapa menjaga ukuran transaksi tetap kecil itu penting.

Transaksi kecil

Untuk transaksi kecil, log biner menggandakan jumlah tulis disk yang dibutuhkan untuk memuat data. Efek ini dapat secara ekstrem menurunkan performa untuk sesi basis data lain dan meningkatkan waktu yang dibutuhkan untuk memuat data. Degradasi yang dialami sebagian bergantung pada tingkat pengunggahan, aktivitas basis data lainnya yang terjadi selama pemuatan, dan kapasitas dari instans DB Amazon RDS Anda.

Log biner juga mengonsumsi ruang disk kira-kira setara dengan jumlah data yang dimuat sampai log dicadangkan dan dihapus. Untungnya, Amazon RDS meminimalkan hal ini dengan mencadangkan dan menghapus log biner secara rutin.

Transaksi besar

Transaksi besar menimbulkan penalti 3X untuk IOPS dan konsumsi disk dengan log biner diaktifkan. Tindakan ini dilakukan karena cache log biner tumpah ke disk, sehingga mengonsumsi ruang disk dan menimbulkan IO tambahan untuk setiap penulisan. Cache tidak dapat ditulis ke binlog hingga transaksi dipindahkan atau di-rollback, sehingga mengonsumsi ruang disk sebanding dengan jumlah data yang dimuat. Saat transaksi dipindahkan, cache harus disalin ke binlog, sehingga menciptakan salinan ketiga dari data pada disk.

Karenanya, harus ada ruang disk kosong setidaknya tiga kali ukuran data untuk memuat data dibandingkan jika memuat dengan log biner dinonaktifkan. Misalnya, 10 GiB data yang dimuat sebagai transaksi tunggal mengonsumsi setidaknya 30 GiB ruang disk selama pemuatan. Data akan mengonsumsi 10 GiB untuk tabel + 10 GiB untuk cache log biner + 10 GiB untuk log biner itu sendiri. File cache akan tetap berada pada disk hingga sesi yang menciptakannya berakhir atau sesi mengisi cache log biner lagi pada transaksi lain. Log biner harus tetap berada pada disk hingga pencadangan selesai, sehingga mungkin perlu waktu beberapa saat sebelum 20 GiB ekstra tersebut dibebaskan.

Jika data dimuat menggunakan LOAD DATA LOCAL INFILE, salinan data lain akan diciptakan jika basis data harus dipulihkan dari sebuah cadangan yang dibuat sebelum pemuatan. Selama pemulihan, MySQL mengekstrak data dari log biner ke dalam sebuah file rata. MySQL kemudian menjalankan LOAD DATA LOCAL INFILE, sama seperti dalam transaksi asli. Akan tetapi, saat ini file input bersifat lokal bagi server basis data. Melanjutkan contoh sebelumnya, pemulihan gagal kecuali jika ada setidaknya 40 GiB ruang disk kosong yang tersedia.

Penonaktifan log biner

Jika memungkinkan, nonaktifkan log biner selama pemuatan data besar untuk menghindari kebutuhan overhead sumber daya dan ruang disk tambahan. Dalam Amazon RDS, penonaktifan log biner sesederhana mengatur periode retensi cadangan menjadi nol. Jika Anda melakukan ini, kami menyarankan agar Anda mengambil snapshot DB dari instans basis data sesaat sebelum pemuatan. Dengan melakukan ini, Anda dapat dengan cepat dan mudah membatalkan perubahan yang dibuat selama pemuatan jika Anda membutuhkannya.

Setelah pemuatan, atur periode retensi cadangan kembali ke nilai yang sesuai (bukan nol).

Anda tidak dapat mengatur periode retensi cadangan ke nol jika instans DB adalah instans DB sumber untuk replika baca.

InnoDB

Informasi dalam bagian ini menyediakan alasan yang kuat untuk menjaga ukuran transaksi tetap kecil saat menggunakan InnoDB.

Urungkan

InnoDB menghasilkan pembatalan untuk mendukung fitur seperti rollback dan MVCC transaksi. Pembatalan disimpan dalam tablespace sistem InnoDB (biasanya ibdata1) dan dipertahankan hingga dibuang oleh utas pembersihan. Utas pembersihan tidak dapat melangkahi pembatalan transaksi aktif terlama, sehingga secara efektif diblokir hingga transaksi dipindahkan atau menyelesaikan rollback. Jika basis data sedang memproses transaksi lain selama pemuatan, pembatalan juga terakumulasi dalam tablespace sistem dan tidak dapat dibuang meskipun transaksi dipindahkan dan tidak ada transaksi lain yang harus dibatalkan untuk MVCC. Dalam situasi ini, semua transaksi (termasuk transaksi hanya baca) yang mengakses baris mana pun yang diubah oleh transaksi apa pun (bukan hanya transaksi pemuatan) akan melambat. Pelambatan terjadi karena transaksi memindai melalui pembatalan yang seharusnya telah dibersihkan jika bukan karena transaksi pemuatan yang berjalan lama.

Pembatalan disimpan dalam tablespace sistem, dan tablespace sistem tidak pernah menyusut ukurannya. Karenanya, transaksi pemuatan data yang besar dapat menyebabkan tablespace sistem menjadi cukup besar, yang mengonsumsi ruang disk yang tidak dapat Anda klaim ulang tanpa menciptakan ulang basis data dari awal.

Rollback

InnoDB dioptimalkan untuk dipindahkan. Pelaksanaan rollback pada transaksi yang besar dapat memakan waktu yang sangat lama. Dalam beberapa kasus, mungkin lebih cepat untuk melakukan point-in-time pemulihan atau mengembalikan snapshot DB.

Format data input

MySQL dapat menerima data yang masuk dalam salah satu dari dua bentuk: file rata dan SQL. Bagian ini menjelaskan beberapa keuntungan dan kerugian utama dari masing-masing faktor.

File rata

Pemuatan file rata dengan LOAD DATA LOCAL INFILE dapat menjadi metode pemuatan data yang paling cepat dan paling murah selama transaksi dijaga tetap relatif kecil. Dibandingkan dengan pemuatan data yang sama dengan SQL, file rata biasanya membutuhkan lebih sedikit lalu lintas jaringan, yang mengurangi biaya transmisi, dan memuat lebih cepat karena pengurangan overhead dalam basis data.

Satu transaksi besar

LOAD DATA LOCAL INFILE memuat keseluruhan file rata sebagai satu transaksi. Ini bukan hal yang buruk. Jika ukuran masing-masing file dapat dijaga tetap kecil, ada sejumlah keuntungan:

- Kemampuan melanjutkan – Pelacakan atas file mana yang telah dimuat menjadi mudah. Jika masalah muncul selama pemuatan, Anda dapat melanjutkan dari bagian terakhir yang tidak bermasalah dengan mudah. Beberapa data mungkin harus ditransmisikan ulang ke Amazon RDS, tetapi dengan ukuran file yang kecil, jumlah yang ditransmisikan ulang minimal.
- Pemuatan data secara paralel – Jika Anda memiliki sisa IOPS dan bandwidth jaringan untuk pemuatan file tunggal, pemuatan secara paralel dapat menghemat waktu.
- Throttling tingkat beban – Pemuatan data memiliki dampak negatif pada proses lain? Lakukan throttling beban dengan meningkatkan interval antar file.

Berhati-hatilah

Keuntungan dari LOAD DATA LOCAL INFILE berkurang dengan cepat seiring meningkatnya ukuran transaksi. Jika membagi sejumlah data yang besar menjadi data yang lebih kecil bukanlah opsi yang tepat, SQL dapat menjadi pilihan yang lebih baik.

SQL

SQL memiliki satu keuntungan utama dibandingkan file rata: kemudahannya untuk menjaga ukuran transaksi tetap kecil. Akan tetapi, SQL dapat memakan waktu yang jauh lebih lama untuk memuat dibandingkan file rata dan sulit untuk menentukan di mana pemuatan dapat dilanjutkan setelah sebuah kegagalan. Misalnya, file mysqldump tidak dapat dimulai ulang. Jika sebuah kegagalan terjadi saat memuat file mysqldump, maka file membutuhkan modifikasi atau penggantian sebelum pemuatan dapat dilanjutkan. Alternatifnya adalah pemulihan pada titik waktu sebelum pemuatan dan pemutaran ulang file setelah penyebab kegagalan diperbaiki.

Pengambilan titik pemeriksaan menggunakan snapshot Amazon RDS

Jika Anda memiliki pemuatan yang akan memakan waktu berjam-jam atau bahkan berhari-hari, pemuatan tanpa log biner bukanlah sebuah prospek yang sangat menarik kecuali Anda dapat melakukan titik pemeriksaan secara berkala. Di sinilah fitur snapshot DB Amazon RDS menjadi sangat berguna. Snapshot DB membuat salinan point-in-time konsisten dari instance database Anda yang dapat digunakan mengembalikan database ke titik waktu itu setelah crash atau kecelakaan lainnya.

Untuk menciptakan sebuah titik pemeriksaan, ambil sebuah snapshot DB saja. Snapshot DB sebelumnya yang diambil untuk titik pemeriksaan dapat dibuang tanpa memengaruhi durabilitas atau waktu pemulihan.

Snapshot juga cepat, sehingga titik pemeriksaan yang sering tidak menambah waktu pemuatan secara signifikan.

Pengurangan waktu pemuatan

Berikut ini adalah beberapa tips tambahan untuk mengurangi waktu pemuatan:

- Ciptakan semua indeks sekunder sebelum memuat. Kontra-intuitif ini untuk yang familier dengan basis data lainnya. Penambahan atau pemodifikasian sebuah indeks sekunder menyebabkan MySQL membuat tabel baru dengan perubahan indeks, menyalin data dari tabel yang sudah ada ke tabel baru, dan menghapus tabel asli.
- Muat data dalam urutan PK. Tindakan ini sangat membantu khususnya untuk tabel InnoDB, ketika waktu pemuatan dapat berkurang 75–80 persen dan ukuran file data menjadi setengahnya.
- Nonaktifkan batasan kunci asing foreign_key_checks=0. Untuk file rata yang dimuat dengan LOAD DATA LOCAL INFILE, tindakan ini dibutuhkan dalam banyak kasus. Untuk pemuatan apa pun, penonaktifan pemeriksaan FK menyediakan peningkatan performa yang signifikan. Pastikan untuk mengaktifkan batasan dan memverifikasi data setelah pemuatan.
- Muat secara paralel kecuali sudah mendekati batas sumber daya. Gunakan tabel berpartisi jika perlu.
- Gunakan sisipan multi-nilai saat memuat dengan SQL untuk meminimalkan overhead saat menjalankan pernyataan. Saat menggunakan mysqldump, tindakan ini akan dilakukan secara otomatis.
- Kurangi IO log InnoDB innodb_flush_log_at_trx_commit=0

- Jika Anda memuat data ke dalam sebuah instans DB yang tidak memiliki replika baca, atur parameter sync_binlog ke 0 saat memuat data. Saat pemuatan data selesai, atur parameter sync_binlog kembali ke 1.
- Muat data sebelum mengonversi instans DB menjadi deployment Multi-AZ. Akan tetapi, jika instans DB sudah menggunakan deployment Multi-AZ, pengalihan ke deployment Satu AZ untuk pemuatan data tidak direkomendasikan, karena hanya memberikan peningkatan marginal.

> **Note**
>
> Penggunaan innodb_flush_log_at_trx_commit=0 menyebabkan InnoDB mengalirkan log-nya setiap detik dan bukan pada setiap pemindahan. Tindakan ini memberikan keuntungan kecepatan yang signifikan, tetapi dapat menyebabkan kehilangan data selama crash. Berhati-hatilah saat menggunakannya.

Topik

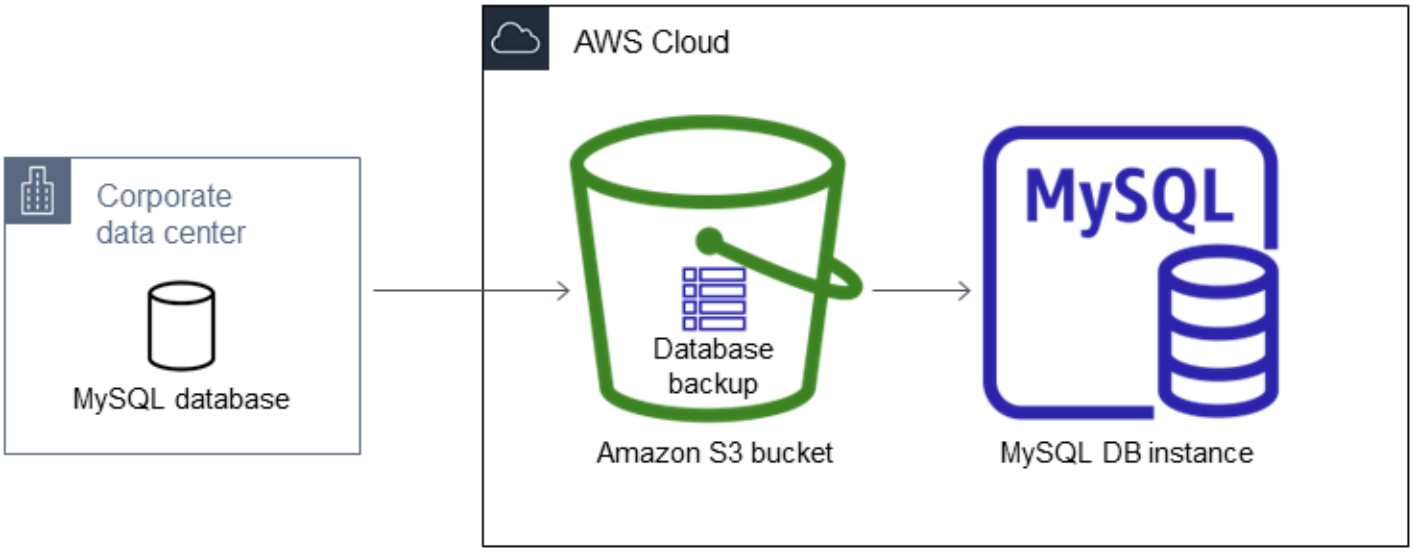
- Memulihkan cadangan ke instans DB MySQL
- Mengimpor data dari MariaDB eksternal atau database MySQL ke RDS untuk MariaDB atau RDS untuk MySQL DB instance
- Mengimpor data ke basis data Amazon RDS MariaDB atau MySQL dengan lebih sedikit waktu henti
- Mengimpor data dari sumber mana pun ke instans DB MySQL atau MariaDB

Memulihkan cadangan ke instans DB MySQL

Amazon RDS mendukung pengimporan basis data MySQL menggunakan file cadangan. Anda dapat membuat cadangan basis data Anda, menyimpannya di Amazon S3, lalu memulihkan file cadangan ke instans DB Amazon RDS baru yang menjalankan MySQL.

Skenario yang dijelaskan dalam bagian ini adalah pemulihan cadangan basis data on-premise. Anda dapat menggunakan teknik ini untuk basis data di lokasi lain, seperti layanan cloud Amazon EC2 atau non-AWS, selama basis data dapat diakses.

Anda dapat menemukan skenario yang mendukung dalam diagram berikut.



Pengimporan file cadangan dari Amazon S3 didukung oleh MySQL di semua Wilayah AWS.

Sebaiknya impor basis data Anda ke Amazon RDS menggunakan file cadangan jika basis data on-premise Anda dapat dijadikan offline selagi file cadangan diciptakan, disalin, dan dipulihkan. Jika basis data Anda tidak dapat dijadikan offline, Anda dapat menggunakan replikasi log biner (binlog) untuk memperbarui basis data Anda setelah Anda bermigrasi ke Amazon RDS melalui Amazon S3, seperti yang dijelaskan dalam topik ini. Untuk informasi selengkapnya, lihat Mengonfigurasi replikasi posisi file log biner dengan instans sumber eksternal. Anda juga dapat menggunakan AWS Database Migration Service untuk memigrasikan basis data Anda ke Amazon RDS. Untuk informasi selengkapnya, lihat Apa itu AWS Database Migration Service?

Batasan dan rekomendasi untuk pengimporan file cadangan dari Amazon S3 ke Amazon RDS

Berikut adalah beberapa batasan dan rekomendasi untuk pengimporan file cadangan dari Amazon S3:

- Anda hanya dapat mengimpor data Anda ke instans DB baru, bukan ke instans DB yang sudah ada.
- Anda harus menggunakan Percona XtraBackup untuk membuat cadangan database lokal Anda.
- Anda tidak dapat mengimpor data dari ekspor snapshot DB ke Amazon S3.
- Anda tidak dapat bermigrasi dari basis data sumber yang memiliki tabel di luar direktori data MySQL default.
- Anda harus mengimpor data Anda ke versi minor default dari versi utama MySQL Anda di Wilayah AWS Anda. Sebagai contoh, jika versi utama Anda adalah MySQL 8.0, dan versi minor default untuk Wilayah AWS Anda adalah 8.0.28, Anda harus mengimpor data Anda ke dalam instans DB MySQL versi 8.0.28. Anda dapat meningkatkan instans DB setelah mengimpor. Untuk informasi tentang penentuan versi minor default, lihat Versi MySQL di Amazon RDS.
- Migrasi mundur tidak didukung baik untuk versi utama maupun minor. Misalnya, Anda tidak dapat bermigrasi dari versi 8.0 ke versi 5.7, dan Anda tidak dapat bermigrasi dari versi 8.0.32 ke versi 8.0.31.
- Anda tidak dapat mengimpor basis data MySQL 5.5 atau 5.6.
- Anda tidak dapat mengimpor basis data MySQL on-premise dari satu versi utama ke versi utama lainnya. Misalnya, Anda tidak dapat mengimpor basis data MySQL 5.7 ke basis data 8.0 RDS for MySQL. Anda dapat meningkatkan instans DB Anda setelah selesai mengimpor.
- Anda tidak dapat memulihkan dari sebuah basis data sumber terenkripsi, tetapi Anda dapat memulihkan ke sebuah instans DB Amazon RDS terenkripsi.
- Anda tidak dapat memulihkan dari cadangan terenkripsi dalam bucket Amazon S3.
- Anda tidak dapat memulihkan dari bucket Amazon S3 di Wilayah AWS selain instans DB Amazon RDS Anda.
- Pengimporan dari Amazon S3 tidak didukung pada kelas instans DB db.t2.micro. Akan tetapi, Anda dapat memulihkan ke kelas instans DB yang berbeda, dan mengubah kelas instans DB pada lain waktu. Untuk informasi selengkapnya tentang kelas instans, lihat Spesifikasi perangkat keras kelas instans DB .

- Amazon S3 membatasi ukuran file yang diunggah ke bucket Amazon S3 hingga 5 TB. Jika file cadangan melebihi 5 TB, Anda harus membagi file cadangan tersebut ke dalam beberapa file yang lebih kecil.
- Saat Anda memulihkan basis data Anda, cadangan akan disalin dan diekstrak pada instans DB Anda. Sebaiknya sediakan ruang penyimpanan untuk instans basis data RDS for Db2 Anda dengan ukuran yang sama dengan atau lebih besar daripada jumlah ukuran cadangan ditambah ukuran basis data asli pada disk.
- Amazon RDS membatasi jumlah file yang dapat diunggah ke sebuah bucket Amazon S3 hingga 1 juta. Jika data cadangan untuk basis data Anda, termasuk semua pencadangan penuh dan inkremental, melebihi 1 juta file, gunakan sebuah file Gzip (.gz), tar (.tar.gz), atau xbstream Percona (.xbstream) untuk menyimpan file pencadangan penuh dan inkremental dalam bucket Amazon S3. Percona XtraBackup 8.0 hanya mendukung Percona xbstream untuk kompresi.
- Akun pengguna tidak diimpor secara otomatis. Simpan akun pengguna Anda dari basis data sumber dan tambahkan ke instans DB baru Anda pada lain waktu.
- Fungsi tidak diimpor secara otomatis. Simpan fungsi Anda dari basis data sumber dan tambahkan ke instans DB baru Anda pada lain waktu.
- Prosedur yang disimpan tidak diimpor secara otomatis. Simpan prosedur yang disimpan dari basis data sumber dan tambahkan ke instans DB baru Anda pada lain waktu.
- Informasi zona waktu tidak diimpor secara otomatis. Rekam informasi zona waktu untuk basis data sumber Anda, dan atur zona waktu instans DB baru Anda pada lain waktu. Untuk informasi selengkapnya, lihat Zona waktu lokal untuk instans DB MySQL.
- Parameter `innodb_data_file_path` harus dikonfigurasi dengan hanya satu file data yang menggunakan nama file data `"ibdata1:12M:autoextend"` default. Basis data yang berisi dua file data, atau memiliki file data dengan nama yang berbeda, tidak dapat dimigrasi menggunakan metode ini.

  Berikut ini adalah contoh nama file yang tidak diperbolehkan: `"innodb_data_file_path=ibdata1:50M; ibdata2:50M:autoextend"` dan `"innodb_data_file_path=ibdata01:50M:autoextend"`.
- Ukuran maksimum basis data yang dipulihkan adalah ukuran basis data maksimum yang didukung dikurangi ukuran cadangan. Jadi, jika ukuran basis data maksimum yang didukung adalah 64 TiB, dan ukuran cadangan adalah 30 TiB, ukuran maksimal basis data yang dipulihkan adalah 34 TiB, seperti dalam contoh berikut:

  ```
  64 TiB - 30 TiB = 34 TiB
  ```

Untuk informasi tentang ukuran basis data maksimum yang didukung oleh Amazon RDS for MySQL, lihat Penyimpanan SSD Tujuan Umum dan Penyimpanan SSD IOPS yang Tersedia.

Ikhtisar pengaturan untuk mengimpor file cadangan dari Amazon S3 ke Amazon RDS

Berikut adalah beberapa komponen yang harus Anda atur untuk mengimpor file cadangan dari Amazon S3 ke Amazon RDS:

- Bucket Amazon S3 untuk menyimpan file cadangan Anda.
- Sebuah cadangan dari basis data on-premise Anda yang diciptakan oleh Percona XtraBackup.
- Peran AWS Identity and Access Management (IAM) untuk mengizinkan Amazon RDS mengakses bucket.

Anda dapat menggunakan bucket Amazon S3 yang sudah Anda miliki. Jika Anda belum memiliki bucket Amazon S3, silakan buat baru. Jika Anda ingin membuat bucket baru, lihat Membuat bucket.

Gunakan XtraBackup alat Percona untuk membuat cadangan Anda. Untuk informasi selengkapnya, lihat Membuat cadangan basis data.

Jika sudah memiliki peran IAM, Anda dapat menggunakannya. Jika Anda belum memiliki peran IAM, silakan buat baru secara manual. Atau, Anda dapat meminta wizard untuk membuatkan peran IAM baru di akun Anda saat memulihkan basis data menggunakan AWS Management Console. Jika Anda ingin membuat peran IAM baru secara manual, atau menyematkan kebijakan kepercayaan dan izin ke sebuah peran IAM yang sudah ada, lihat Penciptaan sebuah peran IAM secara manual. Jika Anda ingin dibuatkan peran IAM baru, ikuti prosedurnya di Konsol.

Membuat cadangan basis data

Gunakan XtraBackup perangkat lunak Percona untuk membuat cadangan Anda. Kami menyarankan Anda menggunakan versi terbaru XtraBackup Percona. Anda dapat menginstal Percona XtraBackup dari Unduh Percona. XtraBackup

> **Warning**
>
> Saat membuat cadangan basis data, XtraBackup mungkin menyimpan kredensil dalam file xtrabackup_info. Periksa file tersebut untuk memastikan pengaturan `tool_command` di dalamnya tidak berisi informasi sensitif.

> **Note**
>
> Untuk migrasi MySQL 8.0, Anda harus menggunakan Percona 8.0. XtraBackup Percona XtraBackup 8.0.12 dan versi yang lebih tinggi mendukung migrasi semua versi MySQL. Jika Anda bermigrasi ke RDS untuk MySQL 8.0.20 atau lebih tinggi, Anda harus menggunakan Percona 8.0.12 atau lebih tinggi. XtraBackup
> Untuk migrasi MySQL 5.7, Anda juga dapat menggunakan Percona 2.4. XtraBackup Untuk migrasi versi MySQL sebelumnya, Anda juga dapat menggunakan XtraBackup Percona 2.3 atau 2.4.

Anda dapat membuat cadangan lengkap file database MySQL Anda menggunakan Percona. XtraBackup Atau, jika Anda sudah menggunakan Percona XtraBackup untuk mencadangkan file database MySQL Anda, Anda dapat mengunggah direktori dan file cadangan lengkap dan tambahan yang ada.

Untuk informasi lebih lanjut tentang mencadangkan database Anda dengan Percona XtraBackup, lihat Percona XtraBackup - dokumentasi dan Biner xtrabackup di situs web Percona.

Membuat cadangan lengkap dengan Percona XtraBackup

Untuk membuat cadangan lengkap file database MySQL Anda yang dapat dipulihkan dari Amazon S3, gunakan utilitas XtraBackup `xtrabackup` Percona () untuk mencadangkan database Anda.

Misalnya, perintah berikut akan membuat sebuah cadangan basis data MySQL dan menyimpan file-nya dalam folder `/on-premises/s3-restore/backup`.

```
xtrabackup --backup --user=<myuser> --password=<password> --target-dir=</on-premises/
s3-restore/backup>
```

Jika Anda ingin mengompres cadangan ke dalam satu file (yang nantinya dapat dipecah, jika perlu), Anda dapat menyimpan cadangan dalam salah satu format berikut:

- Gzip (.gz)
- tar (.tar)
- xbstream Percona (.xbstream)

> **Note**
>
> Percona XtraBackup 8.0 hanya mendukung Percona xbstream untuk kompresi.

Perintah berikut membuat cadangan basis data MySQL Anda yang dipecah menjadi beberapa file Gzip.

```
xtrabackup --backup --user=<myuser> --password=<password> --stream=tar \
   --target-dir=</on-premises/s3-restore/backup> | gzip - | split -d --bytes=500MB \
   - </on-premises/s3-restore/backup/backup>.tar.gz
```

Perintah berikut membuat cadangan basis data MySQL Anda yang dipecah menjadi beberapa file tar.

```
xtrabackup --backup --user=<myuser> --password=<password> --stream=tar \
   --target-dir=</on-premises/s3-restore/backup> | split -d --bytes=500MB \
   - </on-premises/s3-restore/backup/backup>.tar
```

Perintah berikut membuat cadangan basis data MySQL Anda yang dipecah menjadi beberapa file xbstream.

```
xtrabackup --backup --user=<myuser> --password=<password> --stream=xbstream \
   --target-dir=</on-premises/s3-restore/backup> | split -d --bytes=500MB \
   - </on-premises/s3-restore/backup/backup>.xbstream
```

> **Note**
>
> Jika Anda melihat kesalahan berikut, ini mungkin disebabkan perbedaan format file dalam perintah Anda:
>
> ```
> ERROR:/bin/tar: This does not look like a tar archive
> ```

Menggunakan cadangan tambahan dengan Percona XtraBackup

Jika Anda sudah menggunakan Percona XtraBackup untuk melakukan pencadangan penuh dan bertahap dari file database MySQL Anda, Anda tidak perlu membuat cadangan lengkap dan mengunggah file cadangan ke Amazon S3. Sebagai gantinya, Anda dapat menghemat banyak waktu

dengan menyalin direktori dan file cadangan yang sudah ada ke bucket Amazon S3 Anda. Untuk informasi selengkapnya tentang membuat backup tambahan menggunakan Percona XtraBackup, lihat Incremental backup.

Saat menyalin file cadangan penuh dan inkremental yang sudah ada ke bucket Amazon S3, Anda harus menyalin konten direktori dasar secara berulang. Konten tersebut termasuk cadangan penuh dan juga semua direktori dan file cadangan inkremental. Salinan ini harus menjaga struktur direktori di bucket Amazon S3. Amazon RDS melakukan iterasi pada semua file dan direktori. Amazon RDS menggunakan file `xtrabackup-checkpoints` yang disertakan dalam setiap pencadangan inkremental untuk mengidentifikasi direktori dasar, dan untuk memerintahkan pencadangan inkremental berdasarkan rentang nomor urutan log (LSN).

Pertimbangan Backup untuk Percona XtraBackup

Amazon RDS menggunakan file cadangan Anda berdasarkan nama file. Beri nama file cadangan dengan ekstensi file yang sesuai berdasarkan format file—misalnya, `.xbstream` untuk file yang disimpan menggunakan format xbstream Percona.

Amazon RDS menggunakan file cadangan Anda sesuai urutan abjad dan urutan angka alami. Gunakan opsi `split` saat Anda mengeluarkan perintah `xtrabackup` untuk memastikan file cadangan Anda ditulis dan diberi nama dalam urutan yang benar.

Amazon RDS tidak mendukung cadangan sebagian yang dibuat menggunakan Percona. XtraBackup Anda tidak dapat menggunakan opsi berikut untuk membuat cadangan parsial saat mencadangkan file sumber untuk basis data Anda: `--tables`, `--tables-exclude`, `--tables-file`, `--databases`, `--databases-exclude`, atau `--databases-file`.

Amazon RDS mendukung backup tambahan yang dibuat menggunakan Percona. XtraBackup Untuk informasi selengkapnya tentang membuat backup tambahan menggunakan Percona XtraBackup, lihat Incremental backup.

Penciptaan sebuah peran IAM secara manual

Jika Anda tidak memiliki peran IAM, silakan buat baru secara manual. Atau, Anda dapat meminta wizard untuk membuatkan peran IAM baru saat memulihkan basis data menggunakan AWS Management Console. Jika Anda ingin dibuatkan peran IAM baru, ikuti prosedurnya di Konsol.

Untuk membuat peran IAM baru secara manual guna mengimpor basis data dari Amazon S3, buat peran untuk mendelegasikan izin dari Amazon RDS ke bucket Amazon S3 Anda. Saat membuat peran IAM, Anda dapat menyematkan kebijakan kepercayaan dan izin. Untuk mengimpor file

cadangan dari Amazon S3, gunakan kebijakan kepercayaan dan izin yang serupa dengan contoh berikut. Untuk informasi selengkapnya tentang pembuatan peran, lihat Membuat peran untuk mendelegasikan izin ke layanan AWS.

Atau, Anda dapat meminta wizard untuk membuatkan peran IAM baru saat Anda memulihkan basis data menggunakan AWS Management Console. Jika Anda ingin dibuatkan peran IAM baru, ikuti prosedurnya di Konsol

Kebijakan kepercayaan dan izin mengharuskan Anda untuk menyediakan Amazon Resource Name (ARN). Untuk informasi selengkapnya tentang format ARN, lihat Amazon Resource Name (ARN) dan namespace layanan AWS.

Example Kebijakan kepercayaan untuk pengimporan dari Amazon S3

```
{
    "Version": "2012-10-17",
    "Statement":
    [{
        "Effect": "Allow",
        "Principal": {"Service":  "rds.amazonaws.com"},
        "Action": "sts:AssumeRole"
    }]
}
```

Example Kebijakan izin untuk pengimporan dari Amazon S3 — izin pengguna IAM

```
{
    "Version":"2012-10-17",
    "Statement":
    [
        {
            "Sid":"AllowS3AccessRole",
            "Effect":"Allow",
            "Action":"iam:PassRole",
            "Resource":"arn:aws:iam::IAM User ID:role/S3Access"
        }
    ]
}
```

Example Kebijakan izin untuk pengimporan dari Amazon S3 — izin peran

```
{
```

```
    "Version": "2012-10-17",
    "Statement":
    [
        {
        "Effect": "Allow",
        "Action":
            [
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
        "Resource": "arn:aws:s3:::bucket_name"
        },
        {
        "Effect": "Allow",
        "Action":
            [
                "s3:GetObject"
            ],
        "Resource": "arn:aws:s3:::bucket_name/prefix*"
        }
    ]
}
```

> **Note**
>
> Jika Anda menyertakan sebuah prefiks nama file, sertakan tanda bintang (*) setelah prefiks. Jika Anda tidak ingin menyertakan prefiks, sertakan tanda bintang saja.

Mengimpor data dari Amazon S3 ke instans DB MySQL baru

Anda dapat mengimpor data dari Amazon S3 ke instans DB MySQL baru menggunakan AWS Management Console, AWS CLI, atau API RDS.

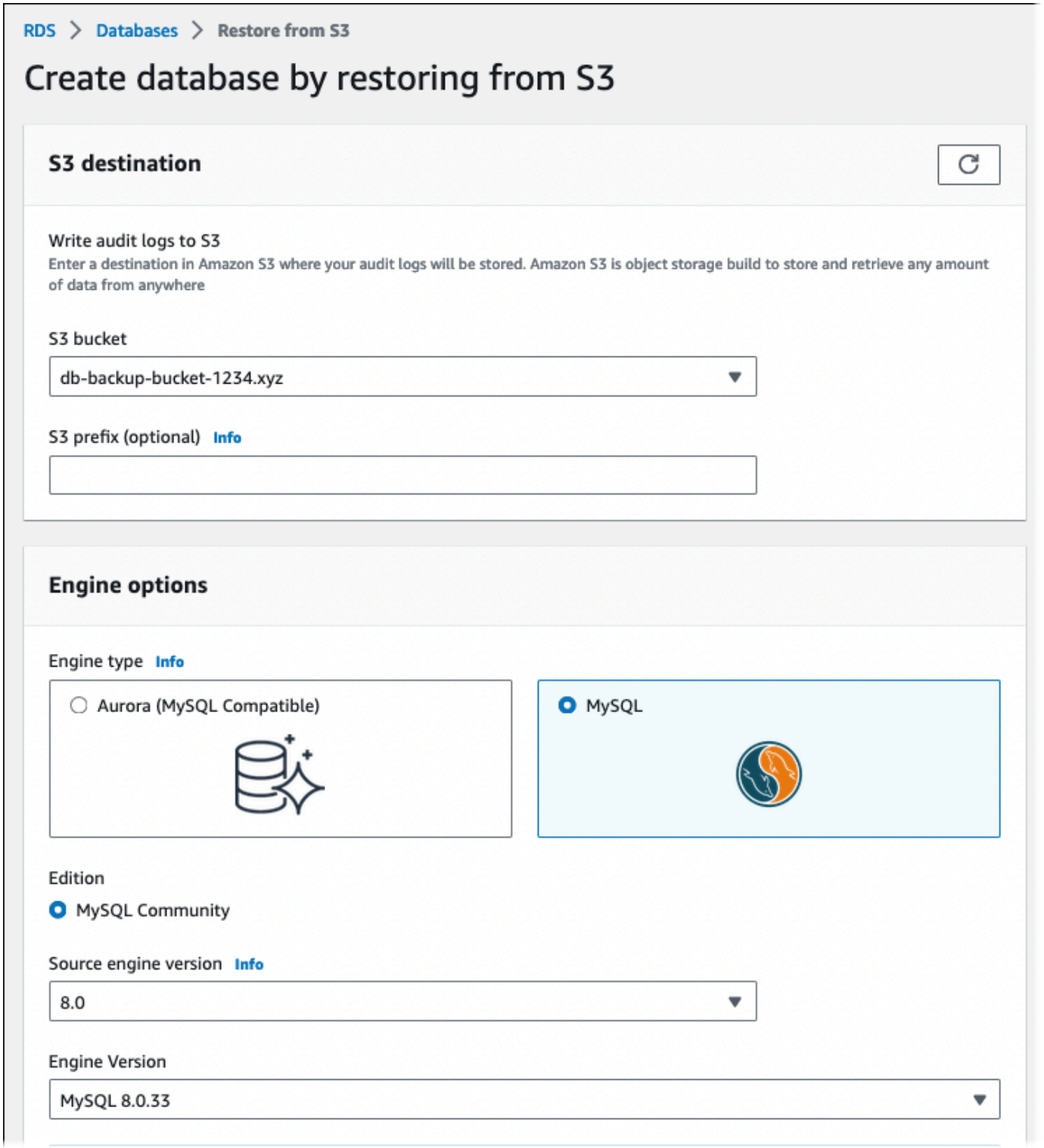
Konsol

Untuk mengimpor data dari Amazon S3 ke instans DB MySQL baru

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.

2. Di sudut kanan atas konsol Amazon RDS, pilih Wilayah AWS untuk membuat instans DB Anda. Pilih Wilayah AWS yang sama dengan bucket Amazon S3 yang berisi cadangan basis data Anda.
3. Di panel navigasi, pilih Basis Data.
4. Pilih Pulihkan dari S3.

   Halaman Buat basis data dengan memulihkan dari S3 akan muncul.



5. Di bagian Tujuan S3:

   a. Pilih Bucket S3 yang berisi cadangan.

b. (Opsional) Untuk Prefiks jalur folder S3, masukkan prefiks jalur file untuk file yang disimpan di bucket Amazon S3 Anda.

   Jika Anda tidak menentukan prefiks, RDS akan membuat instans DB Anda menggunakan semua file dan folder di folder root bucket S3. Jika Anda menentukan prefiks, RDS akan membuat instans DB Anda menggunakan file dan folder dalam bucket S3 yang jalur file-nya dimulai dengan prefiks yang ditentukan.

   Misalnya, Anda menyimpan file cadangan di S3 dalam subfolder bernama “cadangan”, dan Anda memiliki beberapa set file cadangan, masing-masing memiliki direktori tersendiri (gzip_backup1, gzip_backup2, dan sebagainya). Dalam kasus ini, Anda menentukan prefiks backup/gzip_backup1 untuk memulihkan dari file di folder gzip_backup1.

6. Di bagian Opsi mesin:

   a. Untuk Jenis mesin, pilih MySQL.

   b. Untuk Versi mesin sumber, pilih versi utama MySQL basis data sumber Anda.

   c. Untuk Versi, pilih versi minor default versi utama MySQL Anda di versi utama MySQL di Wilayah AWS Anda.

      Di AWS Management Console, hanya versi minor default yang tersedia. Anda dapat meningkatkan instans DB Anda setelah mengimpor.

7. Untuk Peran IAM, Anda dapat memilih peran IAM yang sudah ada.

8. (Opsional) Anda juga dapat meminta dibuatkan peran IAM baru dengan memilih Buat peran baru dan memasukkan Nama peran IAM.

9. Tentukan informasi instans DB Anda. Untuk informasi tentang setiap pengaturan, lihat Pengaturan untuk instans DB.

   > **Note**
   >
   > Pastikan alokasi memori untuk instans DB baru Anda cukup agar operasi pemulihan berhasil.
   > Anda juga dapat memilih Aktifkan penskalaan otomatis penyimpanan untuk mengizinkan pertumbuhan mendatang secara otomatis.

10. Pilih pengaturan tambahan sesuai kebutuhan.

11. Pilih Buat basis data.

AWS CLI

Untuk mengimpor data dari Amazon S3 ke instans MySQL DB baru dengan menggunakanAWS CLI, panggil perintah restore-db-instance-from-s3 dengan parameter berikut. Untuk informasi tentang setiap pengaturan, lihat Pengaturan untuk instans DB.

> **Note**
>
> Pastikan alokasi memori untuk instans DB baru Anda cukup agar operasi pemulihan dapat berhasil.
> Anda juga dapat menggunakan parameter `--max-allocated-storage` untuk mengaktifkan penskalaan otomatis penyimpanan dan memungkinkan pertumbuhan ke depannya secara otomatis.

- `--allocated-storage`
- `--db-instance-identifier`
- `--db-instance-class`
- `--engine`
- `--master-username`
- `--manage-master-user-password`
- `--s3-bucket-name`
- `--s3-ingestion-role-arn`
- `--s3-prefix`
- `--source-engine`
- `--source-engine-version`

Example

Untuk Linux, macOS, atau Unix:

```
aws rds restore-db-instance-from-s3 \
    --allocated-storage 250 \
    --db-instance-identifier myidentifier \
    --db-instance-class db.m5.large \
```

```
    --engine mysql \
    --master-username admin \
    --manage-master-user-password \
    --s3-bucket-name mybucket \
    --s3-ingestion-role-arn arn:aws:iam::account-number:role/rolename \
    --s3-prefix bucketprefix \
    --source-engine mysql \
    --source-engine-version 8.0.32 \
    --max-allocated-storage 1000
```

Untuk Windows:

```
aws rds restore-db-instance-from-s3 ^
    --allocated-storage 250 ^
    --db-instance-identifier myidentifier ^
    --db-instance-class db.m5.large ^
    --engine mysql ^
    --master-username admin ^
    --manage-master-user-password ^
    --s3-bucket-name mybucket ^
    --s3-ingestion-role-arn arn:aws:iam::account-number:role/rolename ^
    --s3-prefix bucketprefix ^
    --source-engine mysql ^
    --source-engine-version 8.0.32 ^
    --max-allocated-storage 1000
```

API RDS

Untuk mengimpor data dari Amazon S3 ke instans MySQL DB baru dengan menggunakan Amazon RDS API, panggil operasi RestoreDB S3. InstanceFrom

Mengimpor data dari MariaDB eksternal atau database MySQL ke RDS untuk MariaDB atau RDS untuk MySQL DB instance

Anda juga dapat mengimpor data dari basis data MariaDB atau MySQL yang sudah ada ke instans DB MySQL atau MariaDB. Hal ini dilakukan dengan menyalin basis data dengan mysqldump dan memasukkannya langsung ke dalam instans DB MariaDB atau MySQL. Utilitas baris perintah `mysqldump` umumnya digunakan untuk membuat pencadangan dan mentransfer data dari satu server MySQL atau MariaDB ke server lainnya. Utilitas ini disertakan dalam perangkat lunak klien MySQL dan MariaDB.

> **Note**
>
> Jika Anda mengimpor atau mengekspor data dalam jumlah besar dengan instans MySQL DB, lebih andal dan lebih cepat untuk memindahkan data masuk dan keluar dari Amazon RDS dengan menggunakan file cadangan dan Amazon S3. `xtrabackup` Untuk informasi selengkapnya, lihat Memulihkan cadangan ke instans DB MySQL.

Perintah `mysqldump` yang umum digunakan untuk memindahkan data dari basis data eksternal ke instans DB Amazon RDS adalah seperti berikut.

```
mysqldump -u local_user \
    --databases database_name \
    --single-transaction \
    --compress \
    --order-by-primary  \
    -plocal_password | mysql -u RDS_user \
        --port=port_number \
        --host=host_name \
        -pRDS_password
```

> **Important**
>
> Pastikan tidak ada spasi di antara opsi `-p` dan kata sandi yang dimasukkan.
> Tentukan kredensial yang berbeda dari perintah yang ditunjukkan di sini sebagai praktik terbaik keamanan.

Perhatikan rekomendasi dan pertimbangan berikut:

- Jangan sertakan skema berikut dalam file dump: `sys`, `performance_schema`, dan `information_schema`. Utilitas `mysqldump` tidak menyertakan skema tersebut secara default.
- Jika Anda perlu memigrasikan pengguna dan hak istimewa, pertimbangkan untuk menggunakan alat yang menghasilkan bahasa kontrol data (DCL) untuk membuatnya kembali, seperti utilitas. pt-show-grants
- Untuk melakukan impor, pastikan pengguna yang melakukannya memiliki akses ke instans DB. Untuk informasi selengkapnya, lihat Mengontrol akses dengan grup keamanan.

Parameternya adalah sebagai berikut:

- `-u local_user` – Gunakan untuk menentukan nama pengguna. Saat menggunakan parameter ini untuk pertama kalinya, Anda harus menentukan nama akun pengguna pada basis data MariaDB atau MySQL lokal yang diidentifikasi oleh parameter `--databases`.
- `--databases database_name` – Gunakan untuk menentukan nama basis data pada instans MariaDB atau MySQL lokal yang ingin Anda impor ke Amazon RDS.
- `--single-transaction` – Gunakan untuk memastikan bahwa semua data yang dimuat dari basis data lokal konsisten dengan satu titik waktu. Jika ada proses lain yang mengubah data saat `mysqldump` membacanya, penggunaan parameter ini dapat membantu menjaga integritas data.
- `--compress` – Gunakan untuk mengurangi konsumsi bandwidth jaringan dengan mengompres data dari basis data lokal sebelum mengirimkannya ke Amazon RDS.
- `--order-by-primary` – Gunakan untuk mengurangi waktu pemuatan dengan mengurutkan setiap tabel data berdasarkan kunci primernya.
- `-plocal_password` – Gunakan untuk menentukan kata sandi. Saat menggunakan parameter ini untuk pertama kalinya, Anda harus menentukan kata sandi untuk akun pengguna yang diidentifikasi oleh parameter `-u`.
- `-u RDS_user` – Gunakan untuk menentukan nama pengguna. Saat menggunakan parameter ini untuk kedua kalinya, Anda harus menentukan nama akun pengguna pada basis data default untuk instans DB MariaDB atau MySQL yang diidentifikasi oleh parameter `--host`.
- `--port port_number` – Gunakan untuk menentukan port instans DB MariaDB atau MySQL Anda. Secara default, port ini adalah 3306, kecuali jika Anda mengubah nilainya saat membuat instans.
- `--host host_name` – Gunakan untuk menentukan nama Sistem Nama Domain (DNS) dari titik akhir instans DB Amazon RDS, misalnya, `myinstance.123456789012.us-east-1.rds.amazonaws.com`. Anda dapat menemukan nilai titik akhir dalam detail instans di Konsol Manajemen Amazon RDS.
- `-pRDS_password` – Gunakan untuk menentukan kata sandi. Saat menggunakan parameter ini untuk kedua kalinya, Anda harus menentukan kata sandi untuk akun pengguna yang diidentifikasi oleh parameter `-u` kedua.

Pastikan Anda membuat prosedur, pemicu, fungsi, atau peristiwa tersimpan apa pun secara manual di dalam basis data Amazon RDS Anda. Jika objek ini berada di basis data yang Anda salin, jangan sertakan saat Anda menjalankan `mysqldump`. Untuk melakukannya, sertakan parameter berikut ke perintah `mysqldump` Anda: `--routines=0 --triggers=0 --events=0`.

Contoh berikut menyalin basis data sampel `world` pada host lokal ke instans DB MySQL.

Untuk Linux, macOS, atau Unix:

```
sudo mysqldump -u localuser \
    --databases world \
    --single-transaction \
    --compress \
    --order-by-primary  \
    --routines=0 \
    --triggers=0 \
    --events=0 \
    -plocalpassword | mysql -u rdsuser \
        --port=3306 \
        --host=myinstance.123456789012.us-east-1.rds.amazonaws.com \
        -prdspassword
```

Untuk Windows, jalankan perintah berikut pada jendela perintah yang telah dibuka dengan mengeklik kanan Jendela Perintah pada menu program Windows dan memilih Jalankan sebagai administrator:

```
mysqldump -u localuser ^
    --databases world ^
    --single-transaction ^
    --compress ^
    --order-by-primary  ^
    --routines=0 ^
    --triggers=0 ^
    --events=0 ^
    -plocalpassword | mysql -u rdsuser ^
        --port=3306 ^
        --host=myinstance.123456789012.us-east-1.rds.amazonaws.com ^
        -prdspassword
```

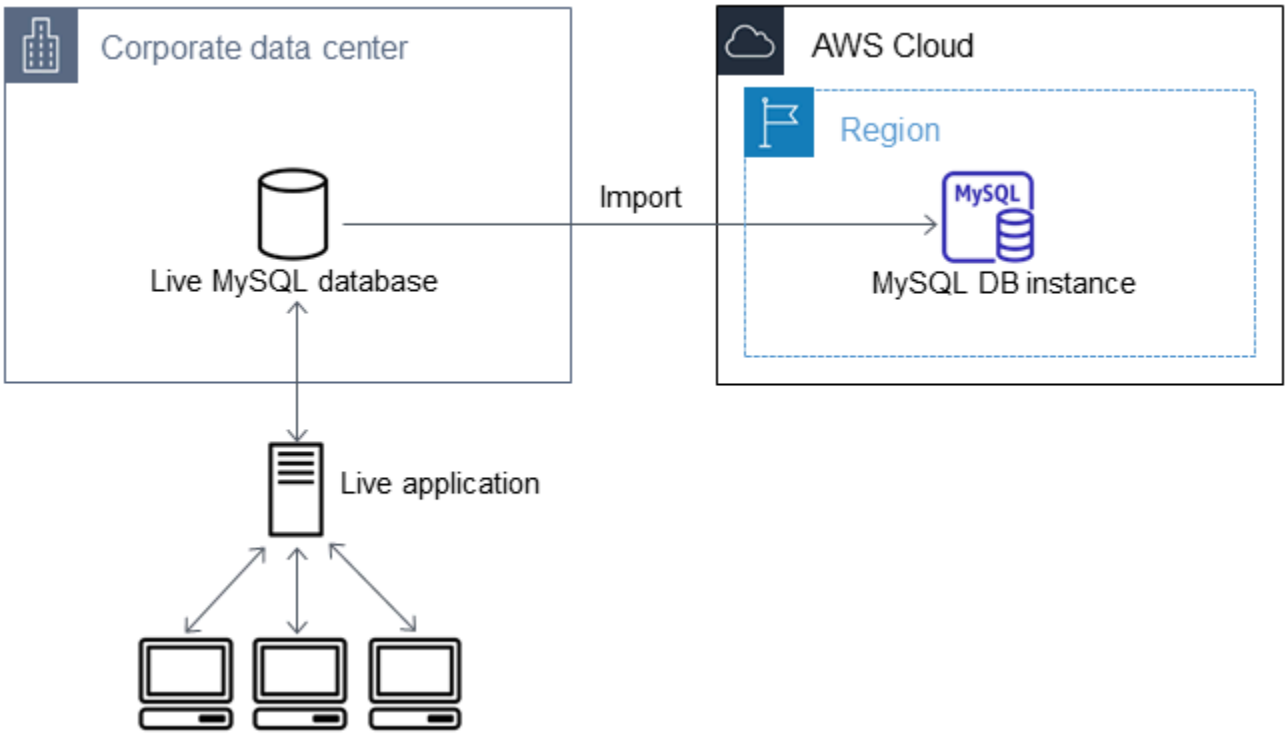
> **Note**
>
> Tentukan kredensial yang berbeda dari perintah yang ditunjukkan di sini sebagai praktik terbaik keamanan.

Mengimpor data ke basis data Amazon RDS MariaDB atau MySQL dengan lebih sedikit waktu henti

Dalam beberapa kasus, mungkin Anda harus mengimpor data dari basis data MariaDB atau MySQL eksternal yang mendukung aplikasi live ke instans DB MariaDB, instans DB MySQL, atau klaster DB Multi-AZ MySQL. Gunakan prosedur berikut untuk meminimalkan dampak terhadap ketersediaan aplikasi. Prosedur ini juga dapat berguna jika Anda menggunakan basis data yang sangat besar. Dengan menggunakan prosedur ini, Anda dapat mengurangi biaya impor dengan mengurangi jumlah data yang dilewatkan di seluruh jaringan AWS.

Dalam prosedur ini, Anda dapat mentransfer salinan data basis data Anda ke instans Amazon EC2 dan mengimpor data ke basis data Amazon RDS baru. Anda kemudian menggunakan replikasi untuk membawa database Amazon RDS up-to-date dengan instans eksternal langsung Anda, sebelum mengarahkan aplikasi Anda ke database Amazon RDS. Lakukan konfigurasi pada replikasi MariaDB berdasarkan pengidentifikasi transaksi global (GTID) jika instans eksternalnya adalah MariaDB 10.0.24 atau yang lebih tinggi dan instans targetnya adalah RDS for MariaDB. Jika tidak, lakukan konfigurasi pada replikasi berdasarkan koordinat log biner. Kami menyarankan replikasi berbasis GTID jika basis data eksternal Anda mendukungnya karena replikasi berbasis GTID adalah metode yang lebih andal. Untuk informasi selengkapnya, lihat Global transaction ID dalam dokumentasi MariaDB.

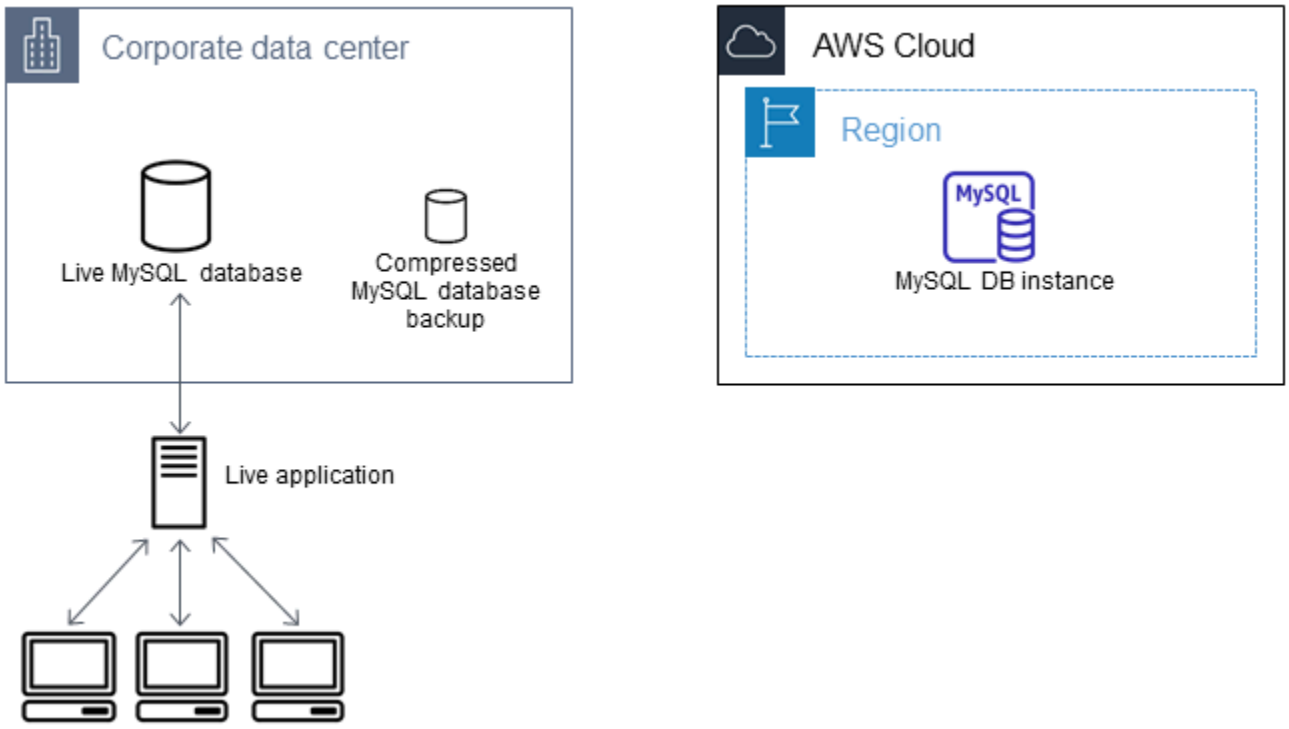
> **Note**
>
> Jika Anda ingin mengimpor data ke instans DB MySQL dan skenario Anda mendukungnya, sebaiknya pindahkan data ke dan dari Amazon RDS dengan menggunakan file cadangan dan Amazon S3. Untuk informasi selengkapnya, lihat Memulihkan cadangan ke instans DB MySQL.



> **Note**
>
> Kami tidak menyarankan Anda untuk menggunakan prosedur ini dengan basis data MySQL sumber dari versi MySQL sebelum versi 5.5 karena ada potensi masalah replikasi. Untuk informasi selengkapnya, lihat Replication compatibility between MySQL versions dalam dokumentasi MySQL.

Membuat salinan basis data yang sudah ada

Langkah pertama dalam proses memigrasikan jumlah data yang besar ke basis data RDS for MariaDB atau RDS for MySQL dengan waktu henti minimal adalah membuat salinan data sumber.



Anda dapat menggunakan utilitas mysqldump untuk membuat cadangan basis data dalam format SQL atau delimited-text. Kami sarankan Anda melakukan uji coba dengan setiap format dalam lingkungan non-produksi untuk melihat metode mana yang dapat meminimalkan jumlah waktu untuk menjalankan mysqldump.

Kami juga sarankan Anda menimbang performa mysqldump dibandingkan dengan manfaat yang ditawarkan dengan menggunakan format delimited-text untuk pemuatan. Pencadangan yang menggunakan format delimited-text akan menciptakan sebuah file teks yang dipisahkan oleh tab untuk setiap tabel yang disalin ke lokasi lain. Untuk mengurangi jumlah waktu yang dibutuhkan untuk mengimpor basis data Anda, Anda dapat memuat file tersebut secara paralel menggunakan perintah `LOAD DATA LOCAL INFILE`. Untuk informasi selengkapnya tentang memilih format mysqldump dan kemudian memuat data, lihat Using mysqldump for backups di dalam dokumentasi MySQL.

Sebelum memulai operasi pencadangan, pastikan Anda mengatur opsi replikasi pada basis data MariaDB atau MySQL yang Anda salin ke Amazon RDS. Opsi replikasi mencakup pengaktifan pencatatan log biner dan pengaturan ID server yang unik. Pengaturan opsi ini menyebabkan server Anda mulai mencatat log transaksi basis data dan menyiapkannya menjadi sebuah instans replikasi sumber di lain waktu dalam proses ini.

> **Note**
>
> Gunakan opsi `--single-transaction` dengan mysqldump karena opsi ini mencadangkan status konsisten basis data. Untuk memastikan file dump valid, jangan menjalankan pernyataan bahasa definisi data (DDL) saat mysqldump sedang berjalan. Anda dapat menjadwalkan jadwal pemeliharaan untuk operasi ini.
> Jangan sertakan skema berikut dalam file dump: `sys`, `performance_schema`, dan `information_schema`. Utilitas mysqldump mengecualikan skema ini secara default.
> Untuk memigrasikan pengguna dan hak istimewa, pertimbangkan untuk menggunakan alat yang menghasilkan bahasa kontrol data (DCL) untuk membuatnya kembali, seperti utilitas. pt-show-grants

Mengatur opsi replikasi

1. Edit file `my.cnf` (file ini biasanya ada di bawah `/etc`).

   ```
   sudo vi /etc/my.cnf
   ```

   Tambahkan opsi `log_bin` dan `server_id` ke bagian `[mysqld]`. Opsi `log_bin` menyediakan sebuah pengidentifikasi nama file untuk file log biner. Opsi `server_id` menyediakan pengidentifikasi unik untuk server dalam hubungan sumber-replika.

   Contoh berikut menunjukkan bagian `[mysqld]` yang diperbarui dari sebuah file `my.cnf`.

   ```
   [mysqld]
   log-bin=mysql-bin
   server-id=1
   ```

   Untuk informasi selengkapnya, lihat dokumentasi MySQL.

2. Untuk replikasi dengan klaster DB Multi-AZ, atur `ENFORCE_GTID_CONSISTENCY` dan parameter `GTID_MODE` ke `ON`.

   ```
   mysql> SET @@GLOBAL.ENFORCE_GTID_CONSISTENCY = ON;
   ```

   ```
   mysql> SET @@GLOBAL.GTID_MODE = ON;
   ```

Pengaturan ini tidak diperlukan untuk replikasi dengan instans DB.

3. Mulai ulang layanan `mysql`.

```
sudo service mysqld restart
```

Membuat salinan cadangan basis data yang sudah ada

1. Buat cadangan data Anda menggunakan utilitas mysqldump, dengan menentukan format SQL atau delimited-text.

   Tentukan `--master-data=2` untuk membuat file cadangan yang dapat digunakan untuk memulai replikasi antar server. Untuk informasi selengkapnya, lihat dokumentasi mysqldump.

   Untuk meningkatkan performa dan memastikan integritas data, gunakan opsi `--order-by-primary` dan `--single-transaction` mysqldump.

   Untuk menghindari penyertaan basis data sistem MySQL di dalam cadangan, jangan gunakan opsi `--all-databases` dengan mysqldump. Untuk informasi selengkapnya, lihat Creating a data snapshot using mysqldump dalam dokumentasi MySQL.

   Gunakan chmod sesuai kebutuhan untuk memastikan bahwa direktori tempat file cadangan diciptakan dapat ditulis.

   > **Important**
   >
   > Pada Windows, jalankan jendela perintah sebagai administrator.

   - Untuk membuat output SQL, gunakan perintah berikut.

     Untuk Linux, macOS, atau Unix:

```
sudo mysqldump \
    --databases database_name \
    --master-data=2  \
    --single-transaction \
    --order-by-primary \
    -r backup.sql \
```

```
    -u local_user \
    -p password
```

> **Note**
>
> Tentukan kredensial selain prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

Untuk Windows:

```
mysqldump ^
    --databases database_name ^
    --master-data=2  ^
    --single-transaction ^
    --order-by-primary ^
    -r backup.sql ^
    -u local_user ^
    -p password
```

> **Note**
>
> Tentukan kredensial selain prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

- Untuk membuat output delimited-text, gunakan perintah berikut.

  Untuk Linux, macOS, atau Unix:

```
sudo mysqldump \
    --tab=target_directory \
    --fields-terminated-by ',' \
    --fields-enclosed-by '"' \
    --lines-terminated-by 0x0d0a \
    database_name \
    --master-data=2 \
    --single-transaction \
    --order-by-primary \
    -p password
```

Untuk Windows:

```
mysqldump ^
    --tab=target_directory ^
    --fields-terminated-by "," ^
    --fields-enclosed-by """ ^
    --lines-terminated-by 0x0d0a ^
    database_name ^
    --master-data=2 ^
    --single-transaction ^
    --order-by-primary ^
    -p password
```

> **Note**
>
> Tentukan kredensial selain prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.
> Pastikan Anda membuat prosedur, pemicu, fungsi, atau peristiwa tersimpan apa pun secara manual di dalam basis data Amazon RDS Anda. Jika Anda memiliki objek-objek tersebut di dalam basis data yang Anda salin, jangan sertakan saat Anda menjalankan mysqldump. Untuk melakukannya, sertakan argumen berikut dengan perintah mysqldump Anda: `--routines=0 --triggers=0 --events=0`.

Saat menggunakan format delimited-text, muncul komentar `CHANGE MASTER TO` saat Anda menjalankan mysqldump. Komentar ini berisi nama dan posisi file log master. Jika instans eksternalnya bukan MariaDB versi 10.0.24 atau yang lebih tinggi, catat nilai untuk `MASTER_LOG_FILE` dan `MASTER_LOG_POS`. Anda memerlukan nilai-nilai ini saat menyiapkan replikasi.

```
-- Position to start replication or point-in-time recovery from
--
-- CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin-changelog.000031',
 MASTER_LOG_POS=107;
```

Jika Anda menggunakan format SQL, Anda dapat memperoleh nama dan posisi file log master pada komentar `CHANGE MASTER TO` tersebut di dalam file cadangan. Jika instans eksternalnya

adalah MariaDB versi 10.0.24 atau yang lebih tinggi, Anda dapat memperoleh GTID pada langkah berikutnya.

2. Jika instans eksternal yang Anda gunakan adalah MariaDB versi 10.0.24 atau yang lebih tinggi, Anda dapat menggunakan replikasi berbasis GTID. Jalankan `SHOW MASTER STATUS` pada instans MariaDB eksternal untuk mendapatkan nama dan posisi file log biner, lalu konversi ke GTID dengan menjalankan `BINLOG_GTID_POS` pada instans MariaDB eksternal.

   ```
   SELECT BINLOG_GTID_POS('binary log file name', binary log file position);
   ```

   Catat GTID yang ditampilkan; Anda membutuhkannya untuk mengonfigurasi replikasi.

3. Kompres data yang disalin untuk mengurangi jumlah sumber daya jaringan yang dibutuhkan untuk menyalin data Anda ke basis data Amazon RDS. Catat ukuran file cadangan. Anda memerlukan informasi ini saat menentukan seberapa besar instans Amazon EC2 yang harus dibuat. Setelah selesai, kompres file cadangan menggunakan GZIP atau utilitas kompresi pilihan Anda.

   - Untuk mengompresi output SQL, gunakan perintah berikut.

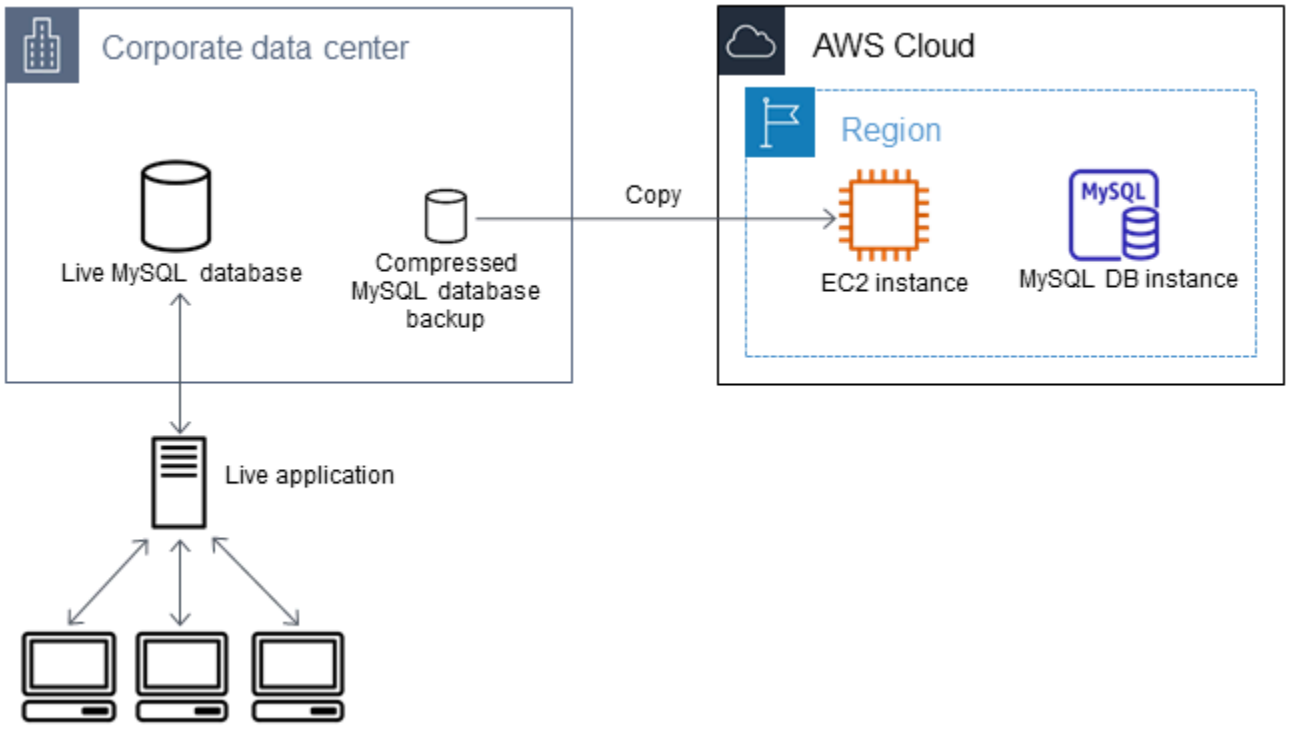
     ```
     gzip backup.sql
     ```

   - Untuk mengompresi output delimited-text, gunakan perintah berikut.

     ```
     tar -zcvf backup.tar.gz target_directory
     ```

Buat sebuah instans Amazon EC2 dan salin basis data terkompresi

Penyalinan file cadangan basis data terkompresi ke sebuah instans Amazon EC2 membutuhkan lebih sedikit sumber daya jaringan dibandingkan dengan melakukan penyalinan langsung data tidak terkompresi antar instans basis data. Setelah data Anda berada di Amazon EC2, Anda dapat menyalinnya dari sana langsung ke basis data MariaDB atau MySQL Anda. Agar Anda dapat menghemat biaya sumber daya jaringan, instans Amazon EC2 Anda harus berada di AWS Wilayah yang sama dengan instans Amazon RDS DB Anda. Memiliki instans Amazon EC2 di AWS Wilayah yang sama dengan database Amazon RDS Anda juga mengurangi latensi jaringan selama impor.



Membuat instans Amazon EC2 dan menyalin data Anda

1. Di Wilayah AWS tempat Anda berencana untuk membuat database RDS, buat virtual private cloud (VPC), grup keamanan VPC, dan subnet VPC. Pastikan aturan masuk untuk grup keamanan VPC Anda mengizinkan alamat IP yang dibutuhkan agar aplikasi Anda dapat terhubung ke AWS. Anda dapat menentukan rentang alamat IP (misalnya, `203.0.113.0/24`), atau grup keamanan VPC lainnya. Anda dapat menggunakan Konsol Manajemen Amazon VPC untuk membuat dan mengelola VPC, subnet, dan grup keamanan. Untuk informasi selengkapnya, lihat Mulai menggunakan Amazon VPC dalam Panduan Memulai Amazon Virtual Private Cloud.
2. Buka Konsol Manajemen Amazon EC2 dan pilih AWS Wilayah yang berisi instans Amazon EC2 dan database Amazon RDS Anda. Luncurkan sebuah instans Amazon EC2 menggunakan VPC, subnet, dan grup keamanan yang Anda buat pada Langkah 1. Pastikan Anda memilih tipe instans dengan penyimpanan yang cukup untuk file cadangan basis data Anda saat tidak terkompresi. Untuk detail tentang instans Amazon EC2 lihat Mulai menggunakan instans Linux Amazon EC2 dalam Panduan Pengguna Amazon Elastic Compute Cloud untuk Linux.
3. Untuk terhubung ke basis data Amazon RDS Anda dari instans Amazon EC2 Anda, edit grup keamanan VPC Anda. Tambahkan aturan masuk yang menentukan alamat IP privat instans EC2 Anda. Anda dapat menemukan alamat IP pribadi pada tab Detail dari panel Instans dalam jendela konsol EC2. Untuk mengedit grup keamanan VPC dan menambahkan aturan masuk, pilih Grup Keamanan dalam panel navigasi konsol EC2, pilih grup keamanan Anda, lalu tambahkan aturan

masuk untuk MySQL atau Aurora yang menentukan alamat IP privat instans EC2 Anda. Untuk mempelajari cara menambahkan aturan ke sebuah grup keamanan VPC, lihat Menambahkan dan menghapus aturan di dalam Panduan Pengguna Amazon VPC.

4. Salin file cadangan basis data terkompresi Anda dari sistem lokal ke instans Amazon EC2 Anda. Gunakan `chmod` sesuai kebutuhan untuk memastikan Anda memiliki izin menulis pada direktori target instans Amazon EC2. Anda dapat menggunakan `scp` atau klien Secure Shell (SSH) untuk menyalin file. Berikut adalah contohnya.

```
scp -r -i key pair.pem backup.sql.gz ec2-user@EC2 DNS:/target_directory/backup.sql.gz
```

> **Important**
>
> Pastikan untuk menyalin data sensitif menggunakan protokol transfer jaringan yang aman.

5. Hubungkan ke instans Amazon EC2 Anda dan instal pembaruan terkini dan alat klien MySQL dengan menggunakan perintah berikut.

```
sudo yum update -y
sudo yum install mysql -y
```

Untuk informasi selengkapnya, lihat Membuat koneksi ke instans Anda dalam Panduan Pengguna Amazon Elastic Compute Cloud untuk Linux.

> **Important**
>
> Contoh ini menginstal klien MySQL pada Amazon Machine Image (AMI) untuk distribusi Linux Amazon. Untuk menginstal klien MySQL pada distribusi yang berbeda, seperti Linux Ubuntu atau Red Hat Enterprise, contoh ini tidak berlaku. Untuk informasi tentang penginstalan MySQL, lihat Installing and Upgrading MySQL dalam dokumentasi MySQL.

6. Saat terhubung ke instans Amazon EC2, dekompresi file cadangan basis data Anda. Berikut ini adalah beberapa contohnya.
   - Untuk mendekompresi output SQL, gunakan perintah berikut.

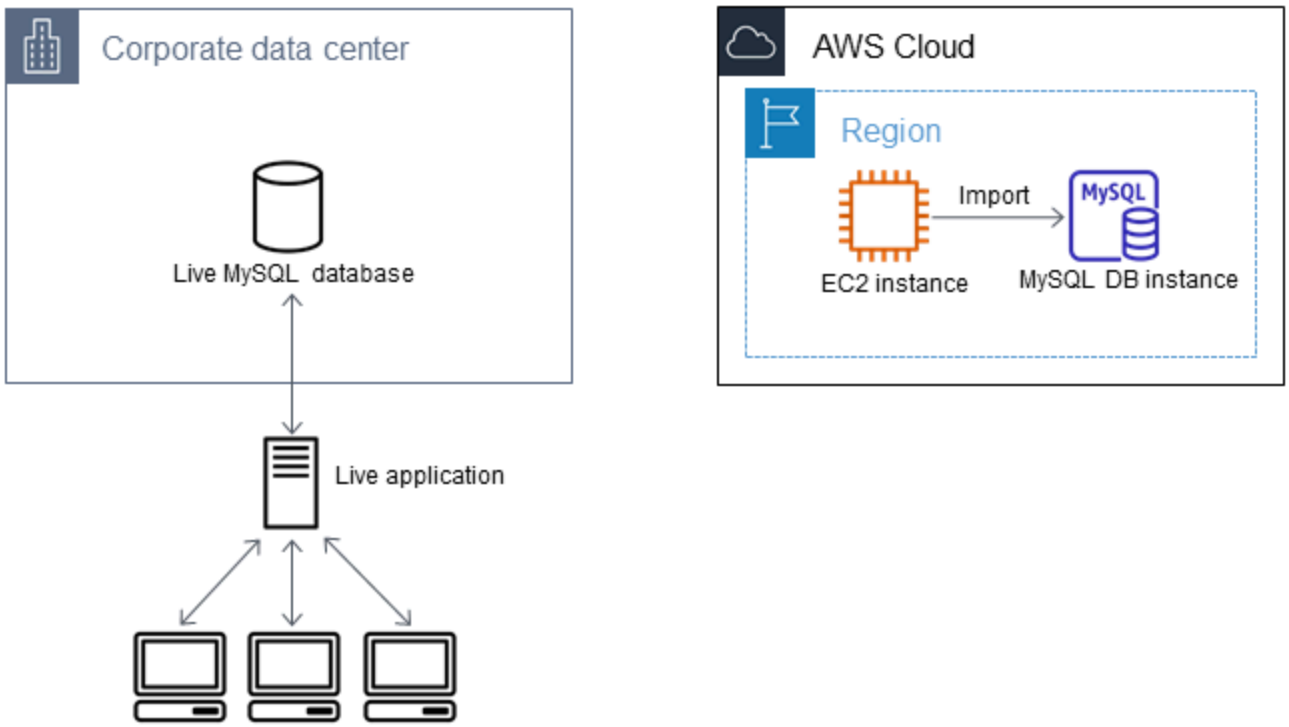
     ```
     gzip backup.sql.gz -d
     ```

   - Untuk mendekompresi output delimited-text, gunakan perintah berikut:

```
tar xzvf backup.tar.gz
```

Buat basis data MySQL atau MariaDB dan impor data dari instans Amazon EC2 Anda

Dengan membuat instance MariaDB, instans MySQL DB, atau kluster DB MySQL Multi-AZ di Wilayah yang AWS sama dengan instans Amazon EC2 Anda, Anda dapat mengimpor file cadangan database dari EC2 lebih cepat daripada melalui internet.



Untuk membuat basis data MariaDB atau MySQL dan mengimpor data Anda

1. Tentukan kelas instans DB dan jumlah ruang penyimpanan yang dibutuhkan untuk mendukung perkiraan beban kerja untuk basis data Amazon RDS ini. Sebagai bagian dari proses ini, putuskan berapa ruang dan kapasitas pemrosesan yang memadai untuk prosedur pemuatan data Anda. Putuskan juga apa yang diperlukan untuk menangani beban kerja produksi. Anda dapat memperkirakan ini berdasarkan ukuran dan sumber daya dari basis data MariaDB atau MySQL sumber. Untuk informasi selengkapnya, lihat Kelas instans DB .
2. Buat instans DB atau cluster DB multi-AZ di AWS Wilayah yang berisi instans Amazon EC2 Anda.

   Untuk membuat klaster DB Multi-AZ MySQL, ikuti petunjuk di Membuat klaster DB Multi-AZ.

Untuk membuat instans DB MariaDB atau MySQL, ikuti petunjuk di Membuat instans DB Amazon RDS dan gunakan pedoman berikut ini:

- Tentukan versi mesin DB yang kompatibel dengan instans DB sumber Anda, seperti berikut:
  - Jika instans sumber Anda adalah MySQL 5.5.x, instans DB Amazon RDS harus MySQL.
  - Jika instans sumber Anda adalah MySQL 5.6.x atau 5.7.x, instans DB Amazon RDS harus MySQL atau MariaDB.
  - Jika instans sumber Anda adalah MySQL 8.0.x, instans DB Amazon RDS harus MySQL 8.0.x.
  - Jika instans sumber Anda adalah MariaDB 5.5 atau yang lebih tinggi, instans DB Amazon RDS harus MariaDB.
- Tentukan cloud privat virtual (VPC) dan grup keamanan VPC yang sama untuk instans Amazon EC2 Anda. Pendekatan ini memastikan bahwa instans Amazon EC2 dan instans Amazon RDS Anda terlihat oleh satu sama lain pada jaringan. Pastikan instans DB Anda dapat diakses publik. Untuk mengatur replikasi dengan basis data sumber Anda yang akan dijelaskan nanti, instans DB Anda harus dapat diakses publik.
- Jangan mengonfigurasikan lebih dari satu Zona Ketersediaan, retensi cadangan, atau replika baca sebelum Anda selesai mengimpor cadangan basis data. Setelah impor selesai, Anda dapat mengonfigurasi Multi-AZ dan retensi cadangan untuk instans produksi.

3. Tinjau opsi konfigurasi default untuk basis data Amazon RDS. Jika grup parameter default untuk basis data tidak memiliki opsi konfigurasi yang Anda inginkan, temukan grup parameter lain atau buat grup parameter baru. Untuk informasi selengkapnya tentang pembuatan grup parameter, lihat Bekerja dengan grup parameter.
4. Hubungkan ke basis data Amazon RDS baru sebagai pengguna master. Buat pengguna yang diperlukan untuk mendukung administrator, aplikasi, dan layanan yang harus mengakses instans. Nama host untuk basis data Amazon RDS adalah nilai Titik akhir untuk instans ini tanpa menyertakan nomor port. Contohnya adalah `mysampledb.123456789012.us-west-2.rds.amazonaws.com`. Anda dapat menemukan nilai titik akhir dalam detail basis data di Konsol Manajemen Amazon RDS.
5. Hubungkan ke instans Amazon EC2 Anda. Untuk informasi selengkapnya, lihat Membuat koneksi ke instans Anda dalam Panduan Pengguna Amazon Elastic Compute Cloud untuk Linux.
6. Hubungkan ke basis data Amazon RDS Anda sebagai sebuah host jarak jauh dari instans Amazon EC2 Anda menggunakan perintah `mysql`. Berikut adalah contohnya.

```
mysql -h host_name -P 3306 -u db_master_user -p
```

Nama host adalah titik akhir basis data Amazon RDS.

7. Pada prompt `mysql`, jalankan perintah `source` dan berikan nama file dump basis data Anda untuk memuat data ke dalam instans DB Amazon RDS:

   - Untuk format SQL, gunakan perintah berikut.

     ```
     mysql> source backup.sql;
     ```

   - Untuk format delimited-text, pertama-tama buat basis data, jika ini bukan basis data default yang Anda buat saat mengatur basis data Amazon RDS.

     ```
     mysql> create database database_name;
     mysql> use database_name;
     ```

     Lalu buat tabel.

     ```
     mysql> source table1.sql
     mysql> source table2.sql
     etc...
     ```

     Lalu impor data.

     ```
     mysql> LOAD DATA LOCAL INFILE 'table1.txt' INTO TABLE table1 FIELDS TERMINATED BY
      ',' ENCLOSED BY '"' LINES TERMINATED BY '0x0d0a';
     mysql> LOAD DATA LOCAL INFILE 'table2.txt' INTO TABLE table2 FIELDS TERMINATED BY
      ',' ENCLOSED BY '"' LINES TERMINATED BY '0x0d0a';
     etc...
     ```

     Untuk meningkatkan performa, Anda dapat melakukan operasi ini secara paralel dari beberapa koneksi sehingga semua tabel Anda akan diciptakan dan kemudian dimuat secara bersamaan.

     > **Note**
     >
     > Jika Anda menggunakan opsi pemformatan data apa pun dengan mysqldump saat Anda pertama kali membuang tabel, pastikan untuk menggunakan opsi yang sama

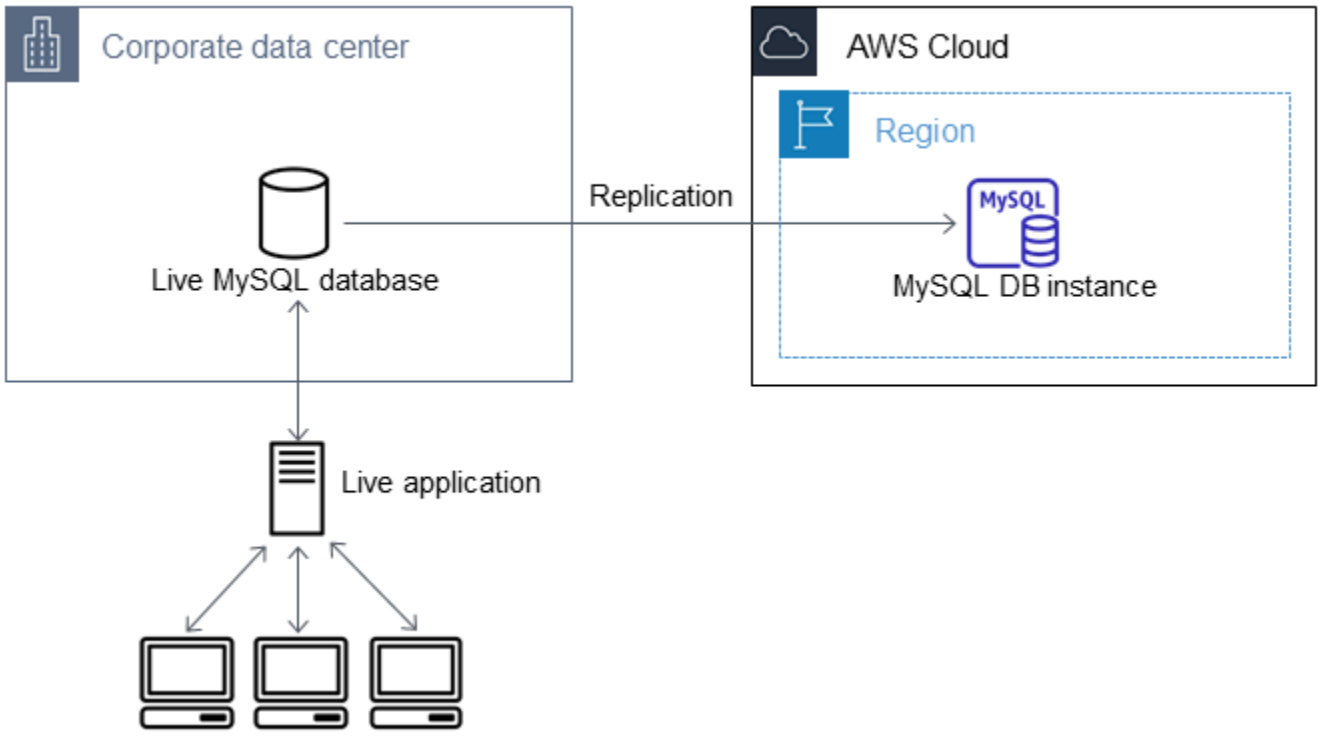
untuk memastikan interpretasi yang tepat dari konten file data. `LOAD DATA LOCAL INFILE`

8. Jalankan `SELECT` kueri sederhana terhadap satu atau dua tabel dalam database yang diimpor untuk memverifikasi bahwa impor berhasil.

Jika Anda tidak lagi memerlukan instans Amazon EC2 yang digunakan dalam prosedur ini, hentikan instans EC2 untuk mengurangi penggunaan sumber daya Anda. AWS Untuk mengakhiri sebuah instans EC2, lihat Mengakhiri instans di Panduan Pengguna Amazon EC2.

Replikasi antara basis data eksternal Anda dan basis data Amazon RDS baru

Basis data sumber Anda kemungkinan diperbarui pada saat menyalin dan mentransfer data ke basis data MariaDB atau MySQL. Dengan demikian, Anda dapat menggunakan replikasi untuk membawa database yang disalin up-to-date dengan database sumber.



Izin yang dibutuhkan untuk memulai replikasi pada basis data Amazon RDS dibatasi dan tidak tersedia untuk pengguna master Amazon RDS Anda. Karena itu, pastikan gunakan perintah Amazon RDS atau perintah mysql.rds_set_external_master_gtid untuk mengonfigurasi replikasi, dan perintah mysql.rds_start_replication untuk memulai replikasi antara basis data live Anda dan basis data Amazon RDS Anda.

Memulai replikasi

Sebelumnya, Anda sudah mengaktifkan pencatatan log biner dan mengatur ID server unik untuk basis data sumber Anda. Sekarang Anda dapat mengatur basis data Amazon RDS Anda sebagai replika dengan basis data live Anda sebagai instans replikasi sumber.

1. Di Konsol Manajemen Amazon RDS, tambahkan alamat IP server yang meng-host basis data sumber ke grup keamanan VPC untuk basis data Amazon RDS. Untuk informasi selengkapnya tentang cara memodifikasi grup keamanan VPC, lihat Grup keamanan untuk VPC Anda dalam Panduan Pengguna Amazon Virtual Private Cloud.

   Anda juga mungkin harus mengonfigurasi jaringan lokal Anda untuk mengizinkan koneksi dari alamat IP basis data Amazon RDS Anda agar klaster DB ini dapat berkomunikasi dengan instans sumber Anda. Untuk menemukan alamat IP basis data Amazon RDS, gunakan perintah `host`.

   ```
   host rds_db_endpoint
   ```

   Nama host adalah nama DNS dari titik akhir basis data Amazon RDS, misalnya, `myinstance.123456789012.us-east-1.rds.amazonaws.com`. Anda dapat menemukan nilai titik akhir dalam detail instans di Konsol Manajemen Amazon RDS.

2. Menggunakan klien pilihan Anda, hubungkan ke instans sumber dan buat pengguna untuk digunakan untuk replikasi. Akun ini digunakan hanya untuk replikasi dan harus dibatasi pada domain Anda untuk meningkatkan keamanan. Berikut adalah contohnya.

   MySQL 5.5, 5.6, dan 5.7

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
   ```

   MySQL 8.0

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED WITH mysql_native_password BY
    'password';
   ```

   > **Note**
   >
   > Tentukan kredensial selain prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

3. Untuk instans sumber, berikan hak istimewa `REPLICATION CLIENT` dan `REPLICATION SLAVE` kepada pengguna replikasi Anda. Misalnya, untuk memberikan hak akses `REPLICATION CLIENT` dan `REPLICATION SLAVE` pada semua basis data untuk pengguna '`repl_user`' bagi domain Anda, jalankan perintah berikut.

   MySQL 5.5, 5.6, dan 5.7

   ```
   GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com'
    IDENTIFIED BY 'password';
   ```

   MySQL 8.0

   ```
   GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
   ```

   > **Note**
   >
   > Tentukan kredensial selain prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

4. Jika Anda menggunakan format SQL untuk menciptakan file cadangan dan instans eksternalnya bukan MariaDB 10.0.24 atau yang lebih tinggi, lihat konten dari file tersebut.

   ```
   cat backup.sql
   ```

   File tersebut menyertakan sebuah komentar `CHANGE MASTER TO` yang berisi nama dan posisi file log master. Komentar ini disertakan dalam file cadangan saat Anda menggunakan opsi `--master-data` dengan mysqldump. Catat nilai untuk `MASTER_LOG_FILE` dan `MASTER_LOG_POS`.

   ```
   --
   -- Position to start replication or point-in-time recovery from
   --

   -- CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin-changelog.000031', MASTER_LOG_POS=107;
   ```

   Jika Anda menggunakan format delimited-text untuk membuat file cadangan dan instans eksternalnya bukan MariaDB 10.0.24 atau yang lebih tinggi, Anda seharusnya sudah memiliki koordinat log biner dari langkah 1 pada prosedur “Membuat salinan cadangan basis data yang sudah ada” dalam topik ini.

Jika instans eksternalnya adalah MariaDB 10.0.24 atau yang lebih tinggi, Anda seharusnya sudah memiliki GTID untuk memulai replikasi dari langkah 2 pada prosedur “Membuat salinan cadangan basis data yang sudah ada” dalam topik ini.

5. Jadikan basis data Amazon RDS sebagai replika. Jika instans eksternalnya bukan MariaDB 10.0.24 atau yang lebih tinggi, hubungkan basis data Amazon RDS sebagai pengguna master dan identifikasi basis data sumber sebagai instans replikasi sumber dengan menggunakan perintah . Gunakan nama file log master dan posisi log master yang Anda tentukan dalam langkah sebelumnya jika Anda memiliki sebuah file cadangan format SQL. Atau gunakan nama dan posisi yang Anda tentukan saat membuat file cadangan jika Anda menggunakan format delimited-text. Berikut adalah contohnya.

```
CALL mysql.rds_set_external_master ('myserver.mydomain.com', 3306,
    'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 0);
```

> **Note**
>
> Tentukan kredensial selain prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

Jika instans eksternalnya adalah MariaDB 10.0.24 atau yang lebih tinggi, hubungkan basis data Amazon RDS sebagai pengguna master dan identifikasi basis data sumber sebagai instans replikasi sumber dengan menggunakan perintah mysql.rds_set_external_master_gtid. Gunakan GTID yang Anda tentukan pada langkah 2 dalam prosedur “Membuat salinan cadangan basis data yang sudah ada” dalam topik ini. Berikut adalah contohnya.

```
CALL mysql.rds_set_external_master_gtid ('source_server_ip_address', 3306,
 'ReplicationUser', 'password', 'GTID', 0);
```

`source_server_ip_address` adalah alamat IP instans replikasi sumber. Alamat DNS privat EC2 saat ini tidak didukung.

> **Note**
>
> Tentukan kredensial selain prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

6. Pada basis data Amazon RDS, terbitkan perintah mysql.rds_start_replication untuk memulai replikasi.

```
CALL mysql.rds_start_replication;
```

7. Pada database Amazon RDS, jalankan perintah SHOW REPLICA STATUS untuk menentukan kapan replika up-to-date dengan instance replikasi sumber. Hasil perintah `SHOW REPLICA STATUS` mencakup bidang `Seconds_Behind_Master`. Ketika `Seconds_Behind_Master` bidang mengembalikan 0, maka replika adalah up-to-date dengan contoh replikasi sumber.

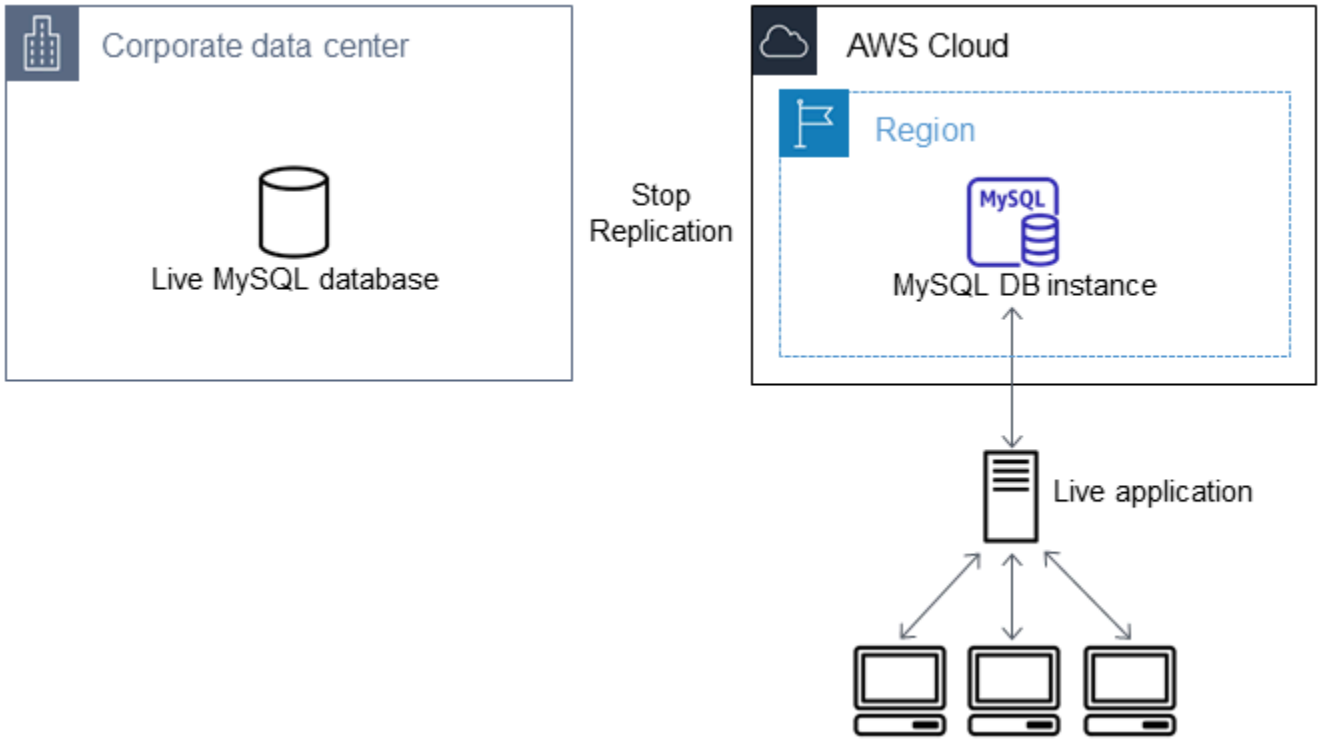
   > **Note**
   >
   > Versi MySQL sebelumnya menggunakan `SHOW SLAVE STATUS`, bukan `SHOW REPLICA STATUS`. Jika Anda menggunakan versi MySQL sebelum 8.0.23, gunakan `SHOW SLAVE STATUS`.

   Untuk instans DB MariaDB 10.5, 10.6, atau 10.11, jalankan prosedur mysql.rds_replica_status, bukan perintah MySQL.

8. Setelah database Amazon RDS up-to-date, aktifkan backup otomatis sehingga Anda dapat memulihkan database tersebut jika diperlukan. Anda dapat mengaktifkan atau memodifikasi pencadangan otomatis untuk basis data Amazon RDS Anda menggunakan Konsol Manajemen Amazon RDS. Untuk informasi selengkapnya, lihat Pengantar cadangan.

Mengalihkan aplikasi live Anda ke instans Amazon RDS Anda

Setelah up-to-date database MariaDB atau MySQL dengan instance replikasi sumber, Anda sekarang dapat memperbarui aplikasi langsung Anda untuk menggunakan instans Amazon RDS.



Mengalihkan aplikasi live Anda ke basis data MariaDB atau MySQL Anda dan menghentikan replikasi

1. Untuk menambahkan grup keamanan VPC untuk basis data Amazon RDS, tambahkan alamat IP server yang meng-host aplikasi. Untuk informasi selengkapnya tentang cara memodifikasi grup keamanan VPC, lihat Grup keamanan untuk VPC Anda dalam Panduan Pengguna Amazon Virtual Private Cloud.
2. Verifikasi bahwa `Seconds_Behind_Master` bidang dalam hasil perintah SHOW REPLICA STATUS adalah 0, yang menunjukkan bahwa replika up-to-date dengan contoh replikasi sumber.

```
SHOW REPLICA STATUS;
```

> **Note**
>
> Versi MySQL sebelumnya menggunakan `SHOW SLAVE STATUS`, bukan `SHOW REPLICA STATUS`. Jika Anda menggunakan versi MySQL sebelum 8.0.23, gunakan `SHOW SLAVE STATUS`.

Untuk instans DB MariaDB 10.5, 10.6, atau 10.11, jalankan prosedur mysql.rds_replica_status, bukan perintah MySQL.

3. Tutup semua koneksi ke sumber setelah transaksi selesai.
4. Perbarui aplikasi Anda untuk menggunakan basis data Amazon RDS. Pembaruan ini biasanya melibatkan perubahan pengaturan koneksi untuk mengidentifikasi nama host dan port basis data Amazon RDS, akun pengguna dan kata sandi untuk terhubung, dan basis data yang digunakan.
5. Hubungkan ke instans DB.

   Untuk klaster DB Multi-AZ, hubungkan ke instans DB penulis.
6. Hentikan replikasi untuk instans Amazon RDS menggunakan perintah mysql.rds_stop_replication.

```
CALL mysql.rds_stop_replication;
```

7. Jalankan perintah mysql.rds_reset_external_master pada basis data Amazon RDS Anda untuk mereset konfigurasi replikasi sehingga instans ini tidak lagi diidentifikasi sebagai replika.

```
CALL mysql.rds_reset_external_master;
```

8. Aktifkan fitur Amazon RDS tambahan seperti dukungan Multi-AZ dan replika baca. Lihat informasi yang lebih lengkap di Mengonfigurasi dan mengelola deployment Multi-AZ dan Menggunakan replika baca instans DB.

Mengimpor data dari sumber mana pun ke instans DB MySQL atau MariaDB

Sebaiknya buat snapshot DB dari instans Amazon RDS DB target sebelum dan sesudah pemuatan data. Snapshot DB Amazon RDS adalah cadangan lengkap instans DB Anda yang dapat digunakan untuk memulihkan instans DB Anda ke status yang diketahui. Saat Anda memulai sebuah snapshot DB, operasi I/O ke instans DB Anda untuk sementara ditangguhkan selama basis data Anda dicadangkan.

Jika perlu, segera buat sebuah snapshot DB sebelum pemuatan agar Anda dapat memulihkan basis data ke statusnya sebelum pemuatan. Dengan snapshot DB yang diambil segera setelah pemuatan, Anda tidak perlu memuat data lagi jika terjadi hal-hal yang tidak diinginkan dan snapshot ini juga dapat digunakan untuk melakukan seeding instans basis data baru.

Daftar berikut menunjukkan langkah-langkah yang harus diambil. Setiap langkah dibahas secara lebih mendetail di bawah.

1. Buat file datar yang berisi data yang akan dimuat.

2. Hentikan aplikasi apa pun yang mengakses instans DB target.
3. Buat sebuah snapshot DB.
4. Pertimbangkan menonaktifkan pencadangan otomatis Amazon RDS.
5. Muat data.
6. Aktifkan pencadangan otomatis lagi.

Langkah 1: Buat file datar yang berisi data yang akan dimuat

Gunakan format umum, seperti CSV (Comma-Separated Values), untuk menyimpan data yang akan dimuat. Setiap tabel harus memiliki file sendiri; Anda tidak dapat menggabungkan data untuk beberapa tabel dalam file yang sama. Beri setiap file nama yang sama dengan tabelnya. Ekstensi file dapat berupa apa pun yang Anda inginkan. Misalnya, jika nama tabelnya `sales`, nama file-nya mungkin `sales.csv` atau `sales.txt`, tetapi bukan `sales_01.csv`.

Jika memungkinkan, urutkan data berdasarkan kunci primer tabel yang dimuat. Tindakan ini secara drastis meningkatkan waktu pemuatan dan meminimalkan kebutuhan penyimpanan disk.

Kecepatan dan efisiensi prosedur ini ditentukan oleh kemampuan untuk mempertahankan ukuran file tetap kecil. Jika ukuran file individual yang tidak terkompresi lebih besar dari 1 GiB, bagilah ke dalam beberapa file dan muat setiap file secara terpisah.

Pada sistem seperti Unix (termasuk Linux), gunakan perintah `split`. Misalnya, perintah berikut membagi file `sales.csv` menjadi beberapa file berukuran kurang dari 1 GiB, yang hanya membagi pada jeda baris (-C 1024m). File-file baru tersebut diberi nama `sales.part_00`, `sales.part_01`, dan seterusnya.

```
split -C 1024m -d sales.csv sales.part_
```

Utilitas yang serupa juga tersedia untuk sistem operasi lain.

Langkah 2: Hentikan aplikasi apa pun yang mengakses instans DB target

Sebelum memulai pemuatan besar, hentikan semua aktivitas aplikasi yang mengakses instans DB target yang akan Anda muati. Kami menyarankan hal ini terutama jika sesi lain akan memodifikasi tabel yang sedang dimuat atau tabel yang menjadi rujukan. Tindakan ini dapat mengurangi risiko pelanggaran pembatasan yang terjadi selama pemuatan dan meningkatkan performa pemuatan.

Tindakan ini juga memungkinkan untuk memulihkan instans DB ke titik tepat sebelum pemuatan tanpa kehilangan perubahan yang dibuat oleh proses yang tidak terlibat dalam pemuatan.

Tentu saja, terkadang ini tidak memungkinkan atau tidak praktis. Jika Anda tidak dapat menghentikan aplikasi dari mengakses instans DB sebelum pemuatan, lakukan langkah-langkah untuk memastikan ketersediaan dan integritas data Anda. Langkah-langkah tertentu yang dibutuhkan dapat bervariasi tergantung kasus penggunaan dan persyaratan situs tertentu.

Langkah 3: Buat snapshot DB

Jika Anda berencana memuat data ke dalam instans DB baru yang tidak berisi data, Anda dapat melompati langkah ini. Sebaliknya, dengan membuat snapshot DB dari instans DB Anda, Anda dapat memulihkan instans DB ke titik tepat sebelum pemuatan, jika memang diperlukan. Seperti yang disebutkan sebelumnya, saat Anda memulai snapshot DB, operasi I/O ke instans DB Anda ditangguhkan selama beberapa menit selama basis data dicadangkan.

Contoh berikut menggunakan AWS CLI `create-db-snapshot` perintah untuk membuat snapshot DB dari `AcmeRDS` instance dan memberikan snapshot DB pengenal. `"preload"`

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-snapshot \
    --db-instance-identifier AcmeRDS \
    --db-snapshot-identifier preload
```

Untuk Windows:

```
aws rds create-db-snapshot ^
    --db-instance-identifier AcmeRDS ^
    --db-snapshot-identifier preload
```

Anda juga dapat menggunakan pemulihan dari fungsionalitas snapshot DB untuk membuat instans DB pengujian untuk melakukan dry run atau untuk membatalkan perubahan yang dibuat selama pemuatan.

Harap diingat bahwa pemulihan basis data dari sebuah snapshot DB akan menciptakan sebuah instans DB baru yang, seperti semua instans DB, memiliki pengidentifikasi yang unik dan titik akhir. Untuk memulihkan instans DB tanpa mengubah titik akhir, pertama-tama hapus instans DB sehingga Anda dapat menggunakan ulang titik akhir.

Misalnya, untuk membuat instans DB untuk dry run atau pengujian lainnya, beri instans DB tersebut pengidentifikasinya sendiri. Dalam contoh ini, AcmeRDS-2" adalah pengidentifikasinya. Contoh ini terhubung ke instans DB menggunakan titik akhir yang terkait dengan AcmeRDS-2.

Untuk Linux, macOS, atau Unix:

```
aws rds restore-db-instance-from-db-snapshot \
    --db-instance-identifier AcmeRDS-2 \
    --db-snapshot-identifier preload
```

Untuk Windows:

```
aws rds restore-db-instance-from-db-snapshot ^
    --db-instance-identifier AcmeRDS-2 ^
    --db-snapshot-identifier preload
```

Untuk menggunakan ulang titik akhir yang sudah ada, pertama-tama hapus instans DB kemudian berikan pengidentifikasi yang sama kepada basis data yang dipulihkan.

Untuk Linux, macOS, atau Unix:

```
aws rds delete-db-instance \
    --db-instance-identifier AcmeRDS \
    --final-db-snapshot-identifier AcmeRDS-Final

aws rds restore-db-instance-from-db-snapshot \
    --db-instance-identifier AcmeRDS \
    --db-snapshot-identifier preload
```

Untuk Windows:

```
aws rds delete-db-instance ^
    --db-instance-identifier AcmeRDS ^
    --final-db-snapshot-identifier AcmeRDS-Final

aws rds restore-db-instance-from-db-snapshot ^
    --db-instance-identifier AcmeRDS ^
    --db-snapshot-identifier preload
```

Contoh sebelumnya mengambil snapshot DB akhir sebuah instans DB sebelum menghapusnya. Ini adalah langkah opsional, tetapi direkomendasikan.

Langkah 4: Pertimbangkan untuk menonaktifkan pencadangan otomatis Amazon RDS

> **Warning**
>
> Jangan mematikan cadangan otomatis jika Anda perlu melakukan point-in-time pemulihan.

Mematikan pencadangan otomatis menghapus semua cadangan yang ada, jadi point-in-time pemulihan tidak mungkin dilakukan setelah pencadangan otomatis dimatikan. Penonaktifan pencadangan otomatis adalah sebuah optimisasi performa dan tidak dibutuhkan untuk pemuatan data. Snapshot DB manual tidak terpengaruh dengan menonaktifkan pencadangan otomatis. Semua snapshot DB manual yang sudah ada tetap tersedia untuk pemulihan.

Penonaktifan pencadangan otomatis mengurangi waktu pemuatan sekitar 25 persen dan mengurangi jumlah ruang penyimpanan yang dibutuhkan selama pemuatan. Jika Anda berencana memuat data ke dalam sebuah instans DB baru yang tidak berisi data, penonaktifan pencadangan adalah cara yang mudah untuk mempercepat pemuatan dan menghindari penggunaan penyimpanan tambahan yang diperlukan untuk pencadangan. Namun, dalam beberapa kasus Anda mungkin berencana untuk melakukan pemuatan ke dalam instans DB yang sudah berisi data. Jika demikian, pertimbangkan manfaat mematikan cadangan terhadap dampak kehilangan kemampuan untuk melakukan. point-in-time-recovery

Instans DB memiliki pencadangan otomatis yang diaktifkan secara default (dengan periode retensi satu hari). Untuk menonaktifkan pencadangan otomatis, atur periode retensi pencadangan ke nol. Setelah pemuatan selesai, Anda dapat mengaktifkan kembali pencadangan dengan mengatur periode retensi pencadangan ke nilai selain nol. Untuk mengaktifkan atau menonaktifkan pencadangan, Amazon RDS mematikan instans DB dan memulainya kembali untuk mengaktifkan atau menonaktifkan pencatatan log MariaDB atau MySQL.

Gunakan AWS CLI `modify-db-instance` perintah untuk mengatur retensi cadangan ke nol dan segera terapkan perubahan. Untuk mengatur periode retensi menjadi nol diperlukan mulai ulang instans DB, jadi tunggu hingga mulai ulang selesai sebelum melanjutkan.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier AcmeRDS \
    --apply-immediately \
    --backup-retention-period 0
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier AcmeRDS ^
    --apply-immediately ^
    --backup-retention-period 0
```

Anda dapat memeriksa status instans DB Anda dengan AWS CLI `describe-db-instances` perintah. Contoh berikut ini menampilkan status instans DB dari instans DB `AcmeRDS`.

```
aws rds describe-db-instances --db-instance-identifier AcmeRDS --query "*[].
{DBInstanceStatus:DBInstanceStatus}"
```

Saat status instans DB adalah `available`, Anda dapat melanjutkan.

Langkah 5: Muat data

Gunakan pernyataan `LOAD DATA LOCAL INFILE` MySQL untuk membaca baris dari file datar Anda ke dalam tabel database.

Contoh berikut menunjukkan cara memuat data dari file bernama `sales.txt` ke dalam tabel bernama `Sales` dalam database.

```
mysql> LOAD DATA LOCAL INFILE 'sales.txt' INTO TABLE Sales FIELDS TERMINATED BY ' '
 ENCLOSED BY '' ESCAPED BY '\\';
Query OK, 1 row affected (0.01 sec)
Records: 1  Deleted: 0  Skipped: 0  Warnings: 0
```

Untuk informasi selengkapnya tentang `LOAD DATA` pernyataan tersebut, lihat dokumentasi MySQL.

Langkah 6: Aktifkan pencadangan otomatis Amazon RDS

Setelah pemuatan selesai, aktifkan pencadangan otomatis Amazon RDS dengan mengatur periode retensi pencadangan kembali ke nilai sebelum pemuatan. Sebagaimana dijelaskan sebelumnya, Amazon RDS akan memulai ulang instans DB, jadi bersiaplah untuk pemadaman singkat.

Contoh berikut menggunakan AWS CLI `modify-db-instance` perintah untuk mengaktifkan backup otomatis untuk instans `AcmeRDS` DB dan mengatur periode retensi menjadi satu hari.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier AcmeRDS \
    --backup-retention-period 1 \
    --apply-immediately
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier AcmeRDS ^
    --backup-retention-period 1 ^
    --apply-immediately
```

Menggunakan replikasi MySQL di Amazon RDS

Anda biasanya menggunakan replika baca untuk mengonfigurasi replikasi antara instans DB Amazon RDS. Untuk informasi umum tentang replika baca, lihat Menggunakan replika baca instans DB. Untuk informasi spesifik tentang cara menggunakan replika baca pada Amazon RDS for MySQL, lihat Menggunakan replika baca MySQL.

Anda dapat menggunakan pengidentifikasi transaksi global (GTID) untuk replikasi dengan RDS for MySQL. Untuk informasi selengkapnya, lihat Menggunakan replikasi berbasis GTID untuk Amazon RDS for MySQL.

Anda juga dapat menyiapkan replikasi antara instans DB RDS for MySQL dan instans MariaDB atau MySQL yang berada di luar Amazon RDS. Untuk informasi tentang cara mengonfigurasi replikasi dengan sumber eksternal, lihat Mengonfigurasi replikasi posisi file log biner dengan instans sumber eksternal.

Untuk opsi replikasi ini, Anda dapat menggunakan replikasi berbasis baris, replikasi berbasis pernyataan, atau replikasi campuran. Replikasi berbasis baris hanya mereplikasi baris yang diubah yang dihasilkan dari laporan SQL. Replikasi berbasis pernyataan mereplikasi seluruh pernyataan SQL. Replikasi campuran menggunakan replikasi berbasis pernyataan jika memungkinkan, tetapi ke replikasi berbasis baris ketika pernyataan SQL yang tidak aman untuk replikasi berbasis pernyataan dijalankan. Dalam sebagian besar kasus, replikasi campuran direkomendasikan. Format biner log dari DB instans menentukan apakah replikasi berbasis baris, berbasis pernyataan, atau campuran. Untuk informasi mengenai pengaturan format log biner, lihat Mengkonfigurasi pengelogan biner MySQL.

> **Note**
>
> Anda dapat mengonfigurasi replikasi untuk mengimpor basis data dari instans MariaDB atau MySQL yang berada di luar Amazon RDS, atau untuk mengekspor basis data ke instans tersebut. Lihat informasi yang lebih lengkap di Mengimpor data ke basis data Amazon RDS MariaDB atau MySQL dengan lebih sedikit waktu henti dan Mengekspor data dari instans DB MySQL dengan menggunakan replikasi.

Topik

- Menggunakan replika baca MySQL

- Menggunakan replikasi berbasis GTID untuk Amazon RDS for MySQL
- Mengonfigurasi replikasi posisi file log biner dengan instans sumber eksternal
- Mengkonfigurasi multi-source-replication untuk RDS untuk MySQL

Menggunakan replika baca MySQL

Setelah itu, Anda bisa menemukan informasi spesifik tentang menggunakan replika baca di RDS for MySQL. Untuk informasi umum tentang replika baca dan petunjuk penggunaannya, lihat Menggunakan replika baca instans DB.

Topik

- Mengonfigurasi replika baca dengan MySQL
- Mengonfigurasi filter replikasi dengan MySQL
- Mengonfigurasi replikasi tertunda dengan MySQL
- Memperbarui replika baca dengan MySQL
- Bekerja dengan deployment replika baca multi-AZ dengan MySQL
- Menggunakan replika baca kaskade dengan RDS for MySQL
- Memantau replika baca MySQL
- Memulai dan menghentikan replikasi dengan replika baca MySQL
- Pemecahan Masalah batasan replika baca MySQL

Mengonfigurasi replika baca dengan MySQL

Sebelum instans DB MySQL dapat berfungsi sebagai sumber replikasi, pastikan untuk mengaktifkan pencadangan otomatis pada instans DB sumber. Untuk melakukannya, atur periode retensi cadangan ke nilai selain 0. Persyaratan ini juga berlaku untuk replika baca yang merupakan instans DB sumber untuk replika baca lain. Pencadangan otomatis didukung untuk replika baca yang menjalankan versi MySQL apa pun. Anda dapat mengonfigurasi replikasi berdasarkan koordinat log biner untuk instans DB MySQL.

Pada RDS untuk MySQL versi 5.7.43 dan versi MySQL 5.7 yang lebih tinggi dan RDS untuk MySQL 8.0.28 dan versi 8.0 yang lebih tinggi, Anda dapat mengonfigurasi replikasi menggunakan pengidentifikasi transaksi global (GTID). Untuk informasi selengkapnya, lihat Menggunakan replikasi berbasis GTID untuk Amazon RDS for MySQL.

Anda dapat membuat hingga 15 replika baca dari satu instans DB dalam Region yang sama. Agar replikasi beroperasi secara efektif, setiap replika baca harus memiliki jumlah sumber daya komputasi dan penyimpanan yang sama seperti instans DB sumber. Jika Anda menskalakan instans DB sumber, maka replika baca juga perlu diskalakan.

RDS for MySQL mendukung replika baca kaskade. Untuk mempelajari cara mengonfigurasi replika baca kaskade, lihat Menggunakan replika baca kaskade dengan RDS for MySQL.

Anda dapat menjalankan beberapa replika baca, membuat dan menghapus tindakan pada saat yang sama yang mereferensikan instans DB sumber yang sama. Saat Anda melakukan tindakan ini, tidak boleh ada lebih dari 15 replika baca untuk setiap instans sumber.

Replika baca dari instans DB MySQL tidak dapat menggunakan versi mesin DB yang lebih rendah dari instans DB sumbernya.

Mempersiapkan instans MySQL DB yang menggunakan MyISAM

Jika instans DB MySQL Anda menggunakan mesin nontransaksional seperti, Anda perlu melakukan langkah-langkah berikut agar berhasil menyiapkan replika baca Anda. Langkah-langkah ini diperlukan untuk memastikan replika baca memiliki salinan data yang konsisten. Langkah-langkah ini tidak diperlukan jika semua tabel Anda menggunakan mesin transaksional seperti InnoDB.

1. Hentikan semua operasi data manipulation language (DML) dan data definition language (DDL) pada tabel non-transaksional dalam instans DB sumber dan tunggu sampai selesai. Pernyataan SELECT dapat terus berjalan.
2. Flush kunci tabel di instans DB sumber.
3. Buat replika baca menggunakan salah satu metode di bagian berikut.
4. Periksa kemajuan pembuatan replika baca menggunakan, misalnya, operasi API `DescribeDBInstances`. Setelah replika baca tersedia, buka kunci tabel instans DB sumber dan lanjutkan operasi basis data normal.

Mengonfigurasi filter replikasi dengan MySQL

Anda dapat menggunakan filter replikasi untuk menentukan basis data dan tabel mana yang direplikasi dengan replika baca. Filter replikasi dapat menyertakan basis data dan tabel dalam replikasi atau mengecualikannya dari replikasi.

Berikut ini adalah beberapa kasus penggunaan untuk filter replikasi:

- Untuk mengurangi ukuran replika baca. Dengan filter replikasi, Anda dapat mengecualikan basis data dan tabel yang tidak diperlukan pada replika baca.
- Untuk mengecualikan basis data dan tabel dari replika baca untuk alasan keamanan.
- Untuk mereplikasi basis data yang berbeda dan tabel untuk kasus penggunaan tertentu di replika baca yang berbeda. Misalnya, Anda mungkin menggunakan replika baca khusus untuk analitik atau penyerpihan.
- Untuk contoh DB yang telah membaca replika yang berbeda Wilayah AWS, untuk mereplikasi database atau tabel yang berbeda dalam bentuk yang berbeda. Wilayah AWS

> **Note**
>
> Anda juga dapat menggunakan filter replikasi untuk menentukan basis data dan tabel apa yang direplikasi dengan instans DB MySQL primer yang dikonfigurasi sebagai replika dalam topologi replikasi masuk. Untuk informasi selengkapnya tentang konfigurasi ini, silakan lihat Mengonfigurasi replikasi posisi file log biner dengan instans sumber eksternal.

Topik

Mengatur parameter filter replikasi untuk RDS for MySQL

Untuk mengonfigurasi filter replikasi, atur parameter filter replikasi berikut pada replika baca:

- `replicate-do-db` – Mereplikasi perubahan ke basis data yang ditentukan. Ketika Anda menetapkan parameter ini untuk replika baca, hanya basis data yang ditentukan dalam parameter yang direplikasi.
- `replicate-ignore-db` – Jangan mereplikasi perubahan ke basis data yang ditentukan. Ketika parameter `replicate-do-db` diatur untuk replika baca, parameter ini tidak dievaluasi.
- `replicate-do-table` – Mereplikasi perubahan ke tabel yang ditentukan. Ketika Anda menetapkan parameter ini untuk replika baca, hanya tabel yang ditentukan dalam parameter yang

direplikasi. Juga, ketika parameter `replicate-do-db` atau `replicate-ignore-db` diatur, pastikan untuk menyertakan basis data yang mencakup tabel tertentu dalam replikasi dengan replika baca.

- `replicate-ignore-table` — Jangan mereplikasi perubahan ke tabel yang ditentukan. Ketika parameter `replicate-do-table` diatur untuk replika baca, parameter ini tidak dievaluasi.
- `replicate-wild-do-table` – Mereplikasi tabel berdasarkan basis data dan pola nama tabel yang ditentukan. Karakter wildcard `%` dan `_` didukung. Ketika parameter `replicate-do-db` atau `replicate-ignore-db` diatur, pastikan untuk menyertakan basis data yang mencakup tabel tertentu dalam replikasi dengan replika baca.
- `replicate-wild-ignore-table` – Jangan mereplikasi tabel berdasarkan basis data dan pola nama tabel yang ditentukan. Karakter wildcard `%` dan `_` didukung. Ketika parameter `replicate-do-table` atau `replicate-wild-do-table` diatur untuk replika baca, parameter ini tidak dievaluasi.

Parameter dievaluasi dalam urutan dalam daftar. Untuk informasi selengkapnya tentang cara kerja parameter ini, lihat dokumentasi MySQL:

- Untuk informasi umum, lihat Opsi dan Variabel Server Replika.
- Untuk informasi tentang cara parameter pemfilteran replikasi basis data dievaluasi, lihat Evaluation of Database-Level Replication and Binary Logging Options.
- Untuk informasi tentang cara parameter filter replikasi basis data dievaluasi, lihat Evaluasi Opsi Replikasi Tingkat Tabel.

Secara default, masing-masing parameter ini memiliki nilai kosong. Pada setiap replika baca, Anda dapat menggunakan parameter ini untuk mengatur, mengubah, dan menghapus filter replikasi. Ketika Anda menetapkan salah satu parameter ini, pisahkan masing-masing filter dari yang lain dengan koma.

Anda dapat menggunakan karakter wildcard `%` dan `_` dalam parameter `replicate-wild-do-table` dan `replicate-wild-ignore-table`. Parameter wildcard `%` mencocokkan jumlah karakter berapa pun, dan wildcard `_` hanya mencocokkan satu karakter.

Format pencatatan log biner instans DB sumber penting untuk replikasi karena menentukan catatan perubahan data. Pengaturan parameter `binlog_format` menentukan apakah replikasi berbasis baris atau berbasis pernyataan. Untuk informasi selengkapnya, lihat Mengkonfigurasi pengelogan biner MySQL.

> **Note**
>
> Semua pernyataan bahasa definisi data (DDL) direplikasi sebagai pernyataan, terlepas dari pengaturan `binlog_format` pada instans DB sumber.

Batasan filter replikasi untuk RDS for MySQL

Batasan tersebut berlaku kepada filter replikasi untuk RDS for MySQL:

- Setiap parameter filter replikasi memiliki batas 2.000 karakter.
- Koma tidak didukung dalam filter replikasi untuk nilai parameter. Dalam daftar parameter, koma hanya dapat digunakan sebagai pemisah nilai. Misalnya, `ParameterValue='`a,b`'` tidak didukung, `ParameterValue='a,b'` tetapi.
- Opsi MySQL `--binlog-do-db` dan `--binlog-ignore-db` untuk filter log biner tidak didukung.
- Filter replikasi tidak mendukung transaksi XA.

  Untuk informasi selengkapnya, lihat Restrictions on XA Transactions dalam dokumentasi MySQL.

Contoh filter replikasi untuk RDS for MySQL

Untuk mengonfigurasi filter replikasi untuk replika baca, modifikasi parameter filter replikasi dalam grup parameter yang terkait dengan replika baca tersebut.

> **Note**
>
> Anda tidak dapat mengubah grup parameter default. Jika replika baca menggunakan grup parameter default, buat grup parameter baru dan kaitkan dengan replika baca tersebut. Untuk mengetahui informasi selengkapnya tentang grup parameter DB, lihat Bekerja dengan grup parameter.

Anda dapat mengatur parameter dalam grup parameter menggunakan AWS Management Console, AWS CLI, atau RDS API. Untuk mengetahui informasi tentang mengatur parameter, lihat Memodifikasi parameter dalam grup parameter DB. Ketika Anda mengatur parameter dalam grup parameter, semua instans DB yang terkait dengan grup parameter tersebut menggunakan pengaturan parameter. Jika Anda mengatur parameter filter replikasi dalam grup parameter, pastikan

bahwa grup parameter dikaitkan hanya dengan replika baca. Biarkan parameter filter replikasi kosong untuk instans DB sumber.

Contoh berikut mengatur parameter menggunakan AWS CLI. Contoh ini menetapkan `ApplyMethod` ke `immediate` sehingga perubahan parameter terjadi segera setelah perintah CLI selesai. Jika Anda ingin menerapkan perubahan tertunda setelah replika baca di-boot ulang, atur `ApplyMethod` ke `pending-reboot`.

Contoh berikut mengatur filter replikasi:

- Including databases in replication
- Including tables in replication
- Including tables in replication with wildcard characters
- Excluding databases from replication
- Excluding tables from replication
- Excluding tables from replication using wildcard characters

Example Menyertakan basis data dalam replikasi

Contoh berikut menyertakan basis data `mydb1` dan `mydb2` dalam replikasi.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "ParameterName=replicate-do-
db,ParameterValue='mydb1,mydb2',ApplyMethod=immediate"
```

Untuk Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "ParameterName=replicate-do-
db,ParameterValue='mydb1,mydb2',ApplyMethod=immediate"
```

Example Menyertakan tabel dalam replikasi

Contoh berikut menyertakan tabel `table1` dan `table2` dalam `mydb1` basis data dalam replikasi.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "ParameterName=replicate-do-
table,ParameterValue='mydb1.table1,mydb1.table2',ApplyMethod=immediate"
```

Untuk Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "ParameterName=replicate-do-
table,ParameterValue='mydb1.table1,mydb1.table2',ApplyMethod=immediate"
```

Example Menyertakan tabel dalam replikasi menggunakan karakter wildcard

Contoh berikut menyertakan tabel dengan nama berawalan `order` dan `return` dalam basis data `mydb` dalam replikasi.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "ParameterName=replicate-wild-do-table,ParameterValue='mydb.order
%,mydb.return%',ApplyMethod=immediate"
```

Untuk Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "ParameterName=replicate-wild-do-table,ParameterValue='mydb.order
%,mydb.return%',ApplyMethod=immediate"
```

Example Mengecualikan basis data dari replikasi

Contoh berikut mengecualikan basis data `mydb5` dan `mydb6` dari replikasi.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "ParameterName=replicate-ignore-
db,ParameterValue='mydb5,mydb6',ApplyMethod=immediate"
```

Untuk Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "ParameterName=replicate-ignore-
db,ParameterValue='mydb5,mydb6',ApplyMethod=immediate"
```

Example Mengecualikan tabel dari replikasi

Contoh berikut mengecualikan tabel `table1` dalam basis data `mydb5` dan `table2` dalam basis data `mydb6` dari replikasi.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "ParameterName=replicate-ignore-
table,ParameterValue='mydb5.table1,mydb6.table2',ApplyMethod=immediate"
```

Untuk Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "ParameterName=replicate-ignore-
table,ParameterValue='mydb5.table1,mydb6.table2',ApplyMethod=immediate"
```

Example Mengecualikan tabel dari replikasi menggunakan karakter wildcard

Contoh berikut mengecualikan tabel dengan nama berawalan `order` dan `return` dalam basis data `mydb7` dari replikasi.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-parameter-group \
```

```
  --db-parameter-group-name myparametergroup \
  --parameters "ParameterName=replicate-wild-ignore-table,ParameterValue='mydb7.order
%,mydb7.return%',ApplyMethod=immediate"
```

Untuk Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "ParameterName=replicate-wild-ignore-table,ParameterValue='mydb7.order
%,mydb7.return%',ApplyMethod=immediate"
```

Melihat filter replikasi untuk replika baca

Anda dapat melihat filter replikasi untuk replika baca dengan cara berikut:

- Memeriksa pengaturan parameter filter replikasi dalam grup parameter yang terkait dengan replika baca.

  Untuk petunjuk, lihat Melihat nilai parameter untuk grup parameter DB.
- Dalam klien MySQL, hubungkan ke replika baca dan jalankan pernyataan `SHOW REPLICA STATUS`.

  Dalam output, bidang berikut menunjukkan filter replikasi untuk replika baca:
  - `Replicate_Do_DB`
  - `Replicate_Ignore_DB`
  - `Replicate_Do_Table`
  - `Replicate_Ignore_Table`
  - `Replicate_Wild_Do_Table`
  - `Replicate_Wild_Ignore_Table`

  Untuk mengetahui informasi selengkapnya tentang bidang ini, lihat Checking Replication Status dalam dokumentasi MySQL.

  > **Note**
  >
  > Versi MySQL sebelumnya menggunakan `SHOW SLAVE STATUS`, bukan `SHOW REPLICA STATUS`. Jika Anda menggunakan versi MySQL sebelum 8.0.23, gunakan `SHOW SLAVE STATUS`.

Mengonfigurasi replikasi tertunda dengan MySQL

Anda dapat menggunakan replikasi tertunda sebagai strategi pemulihan bencana. Dengan replikasi tertunda, Anda menentukan jumlah waktu minimum, dalam detik, untuk menunda replikasi dari sumber ke replika baca. Jika terjadi bencana, seperti tabel yang terhapus secara tidak sengaja, Anda menyelesaikan langkah-langkah berikut untuk memulihkan dari bencana dengan cepat:

- Hentikan replikasi ke replika baca sebelum perubahan yang menyebabkan bencana dikirim ke replika tersebut.

  Gunakan prosedur yang tersimpan di mysql.rds_stop_replication untuk menghentikan replikasi.
- Mulai replikasi dan tentukan bahwa replikasi berhenti secara otomatis di lokasi file log.

  Anda menentukan lokasi tepat sebelum bencana menggunakan prosedur tersimpan mysql.rds_start_replication_until.
- Tingkatkan replika baca menjadi instans DB sumber baru dengan menggunakan petunjuk di Mempromosikan replika baca menjadi instans DB mandiri.

> **Note**
>
> - Pada RDS for MySQL 8.0, replikasi tertunda didukung untuk MySQL 8.0.28 dan yang lebih tinggi. Pada RDS untuk MySQL 5.7, replikasi tertunda didukung untuk MySQL 5.7.43 dan yang lebih tinggi.
> - Gunakan prosedur yang tersimpan untuk mengonfigurasi replikasi tertunda. Anda tidak dapat mengonfigurasi replikasi yang tertunda dengan AWS CLI, API, atau Amazon RDS. AWS Management Console
> - Pada RDS untuk MySQL 5.7.43 dan versi MySQL 5.7 yang lebih tinggi dan RDS untuk MySQL 8.0.28 dan versi 8.0 yang lebih tinggi, Anda dapat menggunakan replikasi berdasarkan pengidentifikasi transaksi global (GTID) dalam konfigurasi replikasi tertunda. Jika Anda menggunakan replikasi berbasis GTID, gunakan prosedur tersimpan mysql.rds_start_replication_until_gtid, bukan prosedur tersimpan mysql.rds_start_replication_until. Untuk informasi selengkapnya tentang replikasi, lihat Menggunakan replikasi berbasis GTID untuk Amazon RDS for MySQL.

Topik

- Mengonfigurasi replikasi tertunda selama pembuatan replika baca
- Mengubah replikasi tertunda untuk replika baca yang sudah ada
- Mengatur lokasi untuk menghentikan replikasi ke replika baca
- Menaikkan replika baca

Mengonfigurasi replikasi tertunda selama pembuatan replika baca

Untuk mengonfigurasi replikasi tertunda untuk replika baca di masa mendatang yang dibuat dari instans DB, jalankan prosedur tersimpan mysql.rds_set_configuration dengan parameter `target delay`.

Untuk mengonfigurasi replikasi tertunda selama pembuatan replika baca

1. Dengan menggunakan klien MySQL, hubungkan ke instans DB MySQL untuk menjadi sumber replika baca sebagai pengguna master.
2. Jalankan prosedur tersimpan mysql.rds_set_configuration dengan parameter `target delay`.

   Misalnya, jalankan prosedur tersimpan berikut untuk menentukan bahwa replikasi ditunda setidaknya satu jam (3.600 detik) untuk replika baca yang dibuat dari instans DB saat ini.

   ```
   call mysql.rds_set_configuration('target delay', 3600);
   ```

   > **Note**
   >
   > Setelah menjalankan prosedur tersimpan ini, replika baca apa pun yang Anda buat menggunakan AWS CLI atau Amazon RDS API dikonfigurasi dengan replikasi yang tertunda oleh jumlah detik yang ditentukan.

Mengubah replikasi tertunda untuk replika baca yang sudah ada

Untuk mengubah replikasi tertunda untuk replika baca yang ada, jalankan prosedur tersimpan mysql.rds_set_source_delay.

Untuk mengubah replikasi tertunda untuk replika baca yang sudah ada

1. Dengan menggunakan klien MySQL, hubungkan ke replika baca sebagai pengguna master.
2. Gunakan prosedur yang tersimpan di mysql.rds_stop_replication untuk menghentikan replikasi.

3. Jalankan prosedur tersimpan mysql.rds_set_source_delay.

   Misalnya, jalankan prosedur tersimpan berikut untuk menentukan bahwa replikasi ke replika baca ditunda setidaknya satu jam (3600 detik).

   ```
   call mysql.rds_set_source_delay(3600);
   ```

4. Gunakan prosedur yang tersimpan di mysql.rds_start_replication untuk memulai replikasi.

Mengatur lokasi untuk menghentikan replikasi ke replika baca

Setelah menghentikan replikasi ke replika baca, Anda dapat memulai replikasi dan kemudian menghentikannya di lokasi file log biner yang ditentukan menggunakan prosedur tersimpan mysql.rds_start_replication_until.

Untuk memulai replikasi ke replika baca dan menghentikan replikasi di lokasi tertentu

1. Dengan menggunakan klien MySQL, hubungkan ke sumber instans DB MySQL sebagai pengguna master.
2. Jalankan prosedur yang tersimpan di mysql.rds_start_replication_until.

   Contoh berikut memulai replikasi dan mereplikasi perubahan hingga mencapai lokasi `120` di file biner `mysql-bin-changelog.000777`. Dalam skenario pemulihan bencana, asumsikan bahwa lokasi `120` tepat sebelum bencana.

   ```
   call mysql.rds_start_replication_until(
     'mysql-bin-changelog.000777',
     120);
   ```

Replikasi berhenti secara otomatis ketika stop point tercapai. Peristiwa RDS berikut dibuat: `Replication has been stopped since the replica reached the stop point specified by the rds_start_replication_until stored procedure`.

Menaikkan replika baca

Setelah replikasi dihentikan, dalam skenario pemulihan bencana, Anda dapat mempromosikan replika baca menjadi instans DB sumber baru. Untuk informasi tentang mempromosikan replika baca, lihat Mempromosikan replika baca menjadi instans DB mandiri.

Memperbarui replika baca dengan MySQL

Replika baca dirancang untuk mendukung kueri baca, tetapi Anda mungkin memerlukan pembaruan sesekali. Misalnya, Anda mungkin perlu menambahkan indeks untuk mengoptimalkan jenis kueri tertentu yang mengakses replika.

Meskipun Anda dapat mengaktifkan pembaruan dengan mengatur `read_only` parameter ke `0` dalam grup parameter DB untuk replika baca, sebaiknya Anda tidak melakukannya karena dapat menyebabkan masalah jika replika baca menjadi tidak kompatibel dengan instans DB sumber. Untuk operasi pemeliharaan, kami menyarankan Anda menggunakan deployment blue/green. Untuk informasi selengkapnya, lihat Menggunakan Deployment Blue/Green untuk pembaruan basis data.

Jika Anda menonaktifkan read-only pada replika baca, ubah nilai `read_only` parameter kembali `1` sesegera mungkin.

Bekerja dengan deployment replika baca multi-AZ dengan MySQL

Anda dapat membuat replika baca dari deployment instans DB Multi-AZ atau tunggal-AZ. Anda menggunakan deployment Multi-AZ untuk meningkatkan dan ketersediaan data kritis, tetapi Anda tidak dapat menggunakan sekunder Multi-AZ untuk melayani kueri hanya baca. Sebagai gantinya, Anda dapat membuat replika baca dari instans DB Multi-AZ multi-lalu lintas tinggi untuk mengeluarkan kueri hanya baca. Jika instans sumber dari deployment Multi-AZ gagal karena replika baca sekunder, setiap replika baca terkait akan otomatis untuk menggunakan sumber sekunder (sekarang primer) sebagai sumber replikasinya. Untuk informasi selengkapnya, lihat Mengonfigurasi dan mengelola deployment Multi-AZ.

Anda dapat membuat replika baca sebagai instans DB Multi-AZ. Amazon RDS membuat instans siaga replika Anda di Zona Ketersediaan lain untuk dukungan failover untuk replika tersebut. Membuat replika baca Anda sebagai instans DB Multi-AZ tidak tergantung pada apakah basis data sumber adalah instans DB Multi-AZ.

Menggunakan replika baca kaskade dengan RDS for MySQL

RDS for MySQL mendukung replika baca kaskade. Dengan replika baca kaskade, Anda dapat menskalakan pembacaan tanpa menambahkan overhead ke instans DB RDS for MySQL Anda.

Dengan replika baca kaskade, instans DB RDS for MySQL Anda mengirimkan data ke replika baca pertama dalam rantai. Replika baca itu kemudian mengirimkan data ke replika kedua dalam rantai, dan seterusnya. Hasil akhirnya adalah bahwa semua replika baca dalam rantai memiliki perubahan dari instans DB RDS for MySQL, tetapi tanpa overhead hanya pada instans DB sumber.

Anda dapat membuat serangkaian hingga tiga replika baca dalam rantai dari instans DB RDS for MySQL sumber. Misalnya, anggaplah bahwa Anda memiliki instans DB RDS for MySQL, `mysql-main`. Anda dapat melakukan hal berikut:

- Dimulai dengan `mysql-main`, buat replika baca pertama dalam rantai, `read-replica-1`.
- Selanjutnya, dari `read-replica-1`, buat replika baca berikutnya dalam rantai, `read-replica-2`.
- Akhirnya, dari `read-replica-2`, buat replika baca ketiga dalam rantai, `read-replica-3`.

Anda tidak dapat membuat replika baca lain di luar replika baca kaskade ketiga ini dalam rangkaian untuk `mysql-main`. Serangkaian instans lengkap dari instans DB RDS for MySQL hingga akhir serangkaian replika baca kaskade dapat terdiri dari paling banyak empat instans DB.

Agar replika baca kaskade berfungsi, setiap sumber instans DB RDS for MySQL harus mengaktifkan pencadangan otomatis. Untuk mengaktifkan pencadangan otomatis pada replika baca, pertama-tama buat replika baca, lalu ubah replika baca untuk mengaktifkan pencadangan otomatis. Untuk informasi selengkapnya, lihat Membuat replika baca.

Seperti halnya replika baca lainnya, Anda dapat mempromosikan replika baca yang merupakan bagian dari kaskade. Mempromosikan replika baca dari dalam rantai replika baca menghilangkan replika tersebut dari rantai. Misalnya, misalkan Anda ingin memindahkan sebagian beban kerja dari instans `mysql-main` DB Anda ke instans baru untuk digunakan oleh departemen akuntansi saja. Dengan asumsi rantai tiga replika baca dari contoh, Anda memutuskan untuk mempromosikan `read-replica-2`. Rantai terpengaruh sebagai berikut:

- Mempromosikan `read-replica-2` menghapusnya dari rantai replikasi.
  - Replika ini sekarang menjadi instans DB baca/tulis penuh.
  - Replika ini terus mereplikasi menjadi `read-replica-3`, seperti yang dilakukan sebelum promosi.
- `mysql-main` Anda terus mereplikasi ke `read-replica-1`.

Untuk informasi lebih lanjut tentang mempromosikan replika baca, lihat Mempromosikan replika baca menjadi instans DB mandiri.

Memantau replika baca MySQL

Untuk replika baca MySQL, Anda dapat memantau kelambatan replikasi di Amazon CloudWatch dengan melihat metrik Amazon RDS. `ReplicaLag` Metrik `ReplicaLag` melaporkan nilai dari kolom `Seconds_Behind_Master` dari perintah `SHOW REPLICA STATUS`.

> **Note**
>
> Versi sebelumnya dari MySQL menggunakan `SHOW SLAVE STATUS` bukan `SHOW REPLICA STATUS`. Jika Anda menggunakan MySQL versi sebelum 8.0.23, gunakan `SHOW SLAVE STATUS`.

Penyebab umum keterlambatan replikasi untuk MySQL adalah sebagai berikut:

- Pemadaman jaringan.
- Menulis ke tabel yang memiliki indeks berbeda pada replika baca. Jika parameter `read_only` diatur ke `0` pada replika baca, replikasi dapat rusak jika replika baca menjadi tidak kompatibel dengan instans DB sumber. Setelah Anda melakukan tugas pemeliharaan pada replika baca, sebaiknya Anda mengatur kembali parameter `read_only` ke `1`.
- Gunakan mesin penyimpanan nontransaksional seperti MyISAM. Replikasi hanya didukung untuk mesin penyimpanan InnoDB pada MySQL.

Saat metrik `ReplicaLag` mencapai 0, replika telah menyamai instans DB sumber. Jika metrik `ReplicaLag` mengembalikan -1, maka replikasi saat ini tidak aktif. `ReplicaLag` = -1 setara dengan `Seconds_Behind_Master` = `NULL`.

Memulai dan menghentikan replikasi dengan replika baca MySQL

Anda dapat menghentikan dan memulai ulang proses replikasi di instans DB Amazon RDS dengan memanggil prosedur yang disimpan sistem mysql.rds_stop_replication dan mysql.rds_start_replication. Anda dapat melakukan ini saat mereplikasi antara dua instans Amazon RDS untuk operasi jangka panjang seperti membuat indeks besar. Anda juga perlu menghentikan dan memulai replikasi saat mengimpor atau mengekspor basis data. Untuk informasi lebih lanjut, lihat Mengimpor data ke basis data Amazon RDS MariaDB atau MySQL dengan lebih sedikit waktu henti dan Mengekspor data dari instans DB MySQL dengan menggunakan replikasi.

Jika replikasi dihentikan selama lebih dari 30 hari berturut-turut, baik secara manual atau karena kesalahan replikasi, Amazon RDS menghentikan replikasi antara instans DB sumber dan semua replika baca. Hal ini dilakukan untuk mencegah peningkatan persyaratan penyimpanan pada instans DB sumber dan waktu failover yang lama. Instans DB replika baca masih tersedia. Namun, replikasi tidak dapat dilanjutkan karena log biner yang diperlukan oleh replika baca dihapus dari instans DB sumber setelah replikasi dihentikan. Anda dapat membuat replika baca baru untuk instans DB sumber untuk memulihkan replikasi.

Pemecahan Masalah batasan replika baca MySQL

Untuk instans DB MySQL, dalam beberapa kasus replika baca menghasilkan kesalahan replikasi atau inkonsistensi data (atau keduanya) antara replika baca dan instans DB sumbernya. Masalah ini terjadi ketika beberapa peristiwa log biner (binlog) atau log redo InnoDB tidak dialirkan selama kegagalan replika baca atau instans DB sumber. Dalam kasus ini, hapus dan buat ulang replika baca secara manual. Anda dapat mengurangi kemungkinan terjadinya hal ini dengan menetapkan nilai parameter berikut: `sync_binlog=1` dan `innodb_flush_log_at_trx_commit=1`. Pengaturan ini dapat mengurangi performa, jadi uji dampaknya sebelum menerapkan perubahan di lingkungan produksi.

> **Warning**
>
> Dalam grup parameter yang terkait dengan instans DB sumber, kami sarankan untuk menjaga nilai parameter ini: `sync_binlog=1` dan `innodb_flush_log_at_trx_commit=1`. Parameter ini dinamis. Jika Anda tidak ingin menggunakan pengaturan ini, sebaiknya setel sementara nilai-nilai tersebut sebelum menjalankan operasi apa pun pada instans DB sumber yang mungkin menyebabkannya dimulai ulang. Operasi ini termasuk, namun tidak terbatas pada, boot ulang, boot ulang dengan failover, tingkatkan versi basis data, dan mengubah kelas instans DB atau penyimpanannya. Rekomendasi yang sama berlaku untuk membuat replika baca baru untuk instans DB sumber.
>
> Kegagalan untuk mengikuti panduan ini meningkatkan risiko replika baca yang menghadirkan kesalahan replikasi atau inkonsistensi data (atau keduanya) antara replika baca dan instans DB sumbernya.

Teknologi replikasi untuk MySQL bersifat asinkron. Karena mereka tidak sinkron, sesekali `BinLogDiskUsage` meningkatkan instans DB sumber dan `ReplicaLag` pada replika baca diharapkan. Misalnya, volume operasi tulis tinggi ke instans DB sumber dapat terjadi secara

paralel. Sebaliknya, operasi ke replika baca diseret menggunakan utas I/O tunggal, yang dapat menyebabkan jeda antara instans sumber dan replika baca. Untuk informasi selengkapnya tentang replika read-only di dokumentasi MySQL, lihat Replication implementation details.

Anda dapat melakukan beberapa hal untuk mengurangi keterlambatan antara pembaruan ke instans DB sumber dan pembaruan berikutnya ke replika baca, seperti berikut:

- Mengukur replika baca untuk memiliki ukuran penyimpanan dan kelas instans DB yang sebanding dengan instans DB sumber.
- Memastikan kompatibilitas pengaturan parameter di grup parameter DB yang digunakan oleh instans DB sumber dan replika baca. Untuk mengetahui informasi selengkapnya dan instans, lihat diskusi tentang `max_allowed_packet` nanti di bagian ini.

Amazon RDS memantau status replikasi replika baca Anda dan memperbarui `Replication State` bidang instans replika baca untuk `Error` jika replikasi berhenti karena alasan apa pun. Contohnya adalah jika kueri DML berjalan di atas replika baca, pembaruan yang dibuat di instans DB sumber.

Anda dapat meninjau perincian kesalahan terkait yang disebabkan oleh mesin MySQL dengan melihat bidang `Replication Error`. Peristiwa yang menunjukkan status replika baca juga dihasilkan, termasuk RDS-EVENT-0045, RDS-EVENT-0046, dan RDS-EVENT-0047. Untuk informasi selengkapnya tentang peristiwa dan berlangganan peristiwa, lihat Menggunakan pemberitahuan peristiwa Amazon RDS. Jika pesan kesalahan MySQL dikembalikan, tinjau nomor kesalahan dalam dokumentasi pesan kesalahan MySQL.

Satu masalah umum yang dapat menyebabkan kesalahan replikasi adalah ketika nilai untuk `max_allowed_packet` parameter untuk replika baca lebih kecil dari `max_allowed_packet` untuk instans DB sumber. Parameter `max_allowed_packet` adalah parameter kustom yang dapat Anda atur di grup parameter DB. Anda menggunakan `max_allowed_packet` untuk menentukan ukuran maksimum kode DML yang dapat dijalankan di basis data. Dalam beberapa kasus, nilai `max_allowed_packet` dalam grup parameter DB yang dikaitkan dengan replika baca lebih kecil daripada nilai `max_allowed_packet` dalam grup parameter DB yang terkait dengan instans DB sumber. Dalam kasus ini, proses replikasi dapat melempar kesalahan `Packet bigger than 'max_allowed_packet' bytes` dan menghentikan replikasi. Untuk memperbaiki kesalahan, miliki instans DB sumber dan replika baca menggunakan grup parameter DB dengan nilai parameter `max_allowed_packet` yang sama.

Situasi umum lainnya yang dapat menyebabkan kesalahan replikasi mencakup hal-hal berikut:

- Tuliskan ke tabel di replika baca. Dalam beberapa kasus, Anda mungkin membuat indeks pada replika baca yang berbeda dari indeks pada instans DB sumber. Jika Anda melakukannya, atur parameter `read_only` ke `0` untuk membuat indeks. Jika Anda menulis ke tabel pada replika baca, replikasi dapat rusak jika replika baca menjadi tidak kompatibel dengan instans DB sumber. Setelah Anda melakukan tugas pemeliharaan pada replika baca, sebaiknya Anda mengatur kembali parameter `read_only` ke `1`.
- Gunakan mesin penyimpanan non-transaksional seperti MyISAM. Replika baca membutuhkan mesin penyimpanan transaksional. Replikasi hanya didukung untuk mesin penyimpanan InnoDB pada MySQL.
- Gunakan kueri nondeterministik yang tidak aman seperti `SYSDATE()`. Untuk informasi selengkapnya, lihat Determination of safe and unsafe statements in binary logging.

Jika Anda memutuskan bahwa Anda dapat melewati kesalahan dengan aman, Anda dapat mengikuti langkah-langkah yang dijelaskan di bagian Melewati kesalahan replikasi saat ini. Jika tidak, Anda dapat menghapus replika baca terlebih dahulu. Selanjutnya Anda membuat instans menggunakan pengenal instans DB yang sama sehingga titik akhir tetap sama dengan replika baca lama Anda. Jika kesalahan replikasi diperbaiki, `Replication State` berubah menjadi mereplikasi.

Menggunakan replikasi berbasis GTID untuk Amazon RDS for MySQL

Berikut ini, Anda dapat mempelajari cara menggunakan pengidentifikasi transaksi global (GTID) dengan replikasi log biner (binlog) di antara instans Amazon RDS for MySQL.

Jika Anda menggunakan replikasi binlog dan tidak familier dengan replikasi berbasis GTID dengan MySQL, ketahui latar belakangnya di Replication with global transaction identifiers dalam dokumentasi MySQL.

Replikasi berbasis GTID didukung untuk semua RDS for MySQL versi 5.7, dan RDS for MySQL versi 8.0.26, serta MySQL versi 8.0 yang lebih tinggi. Semua instans DB MySQL dalam konfigurasi replikasi harus memenuhi persyaratan ini.

Topik

- Ikhtisar pengidentifikasi transaksi global (GTID)
- Parameter untuk replikasi berbasis GTID
- Pengonfigurasian replikasi berbasis GTID untuk replika baca baru
- Pengonfigurasian replikasi berbasis GTID untuk replika baca yang sudah ada
- Menonaktifkan replikasi berbasis GTID untuk instans DB MySQL dengan replika baca

Ikhtisar pengidentifikasi transaksi global (GTID)

Pengidentifikasi transaksi global (GTID) adalah pengidentifikasi unik yang dibuat untuk transaksi MySQL yang dilakukan. Anda dapat menggunakan GTID agar pemecahan masalah pada replikasi binlog bisa dilakukan dengan lebih mudah dan sederhana.

MySQL menggunakan dua jenis transaksi untuk replikasi binlog:

- Transaksi GTID – Transaksi yang diidentifikasi oleh GTID.
- Transaksi anonim – Transaksi yang tidak memiliki GTID.

Dalam konfigurasi replikasi, GTID bersifat unik di semua instans DB. GTID menyederhanakan konfigurasi replikasi karena saat Anda menggunakannya, Anda tidak harus merujuk ke posisi file log. GTID juga mempermudah pelacakan transaksi yang direplikasi dan menentukan apakah instans sumber dan replika konsisten.

Anda dapat menggunakan replikasi berbasis GTID untuk mereplikasi data dengan replika baca RDS for MySQL. Anda dapat mengonfigurasi replikasi berbasis GTID saat membuat replika baca baru, atau mengonversi replika baca yang ada untuk menggunakan replikasi berbasis GTID.

Anda juga dapat menggunakan replikasi berbasis GTID dalam konfigurasi replikasi tertunda dengan RDS for MySQL. Untuk informasi selengkapnya, lihat Mengonfigurasi replikasi tertunda dengan MySQL.

Parameter untuk replikasi berbasis GTID

Gunakan parameter berikut untuk mengonfigurasi replikasi berbasis GTID.

Parameter	Nilai valid	Deskripsi
`gtid_mode`	`OFF`, `OFF_PERMISSIVE` , `ON_PERMISSIVE` , `ON`	`OFF` menentukan bahwa transaksi baru adalah transaksi anonim (yaitu, tidak memiliki GTID), dan transaksi harus anonim agar dapat direplikasi. `OFF_PERMISSIVE` menentukan bahwa transaksi baru adalah transaksi anonim, tetapi semua transaksi dapat direplikasi.

Parameter	Nilai valid	Deskripsi
		`ON_PERMISSIVE` menentukan bahwa transaksi baru adalah transaksi GTID, tetapi semua transaksi dapat direplikasi. `ON` menentukan bahwa transaksi baru adalah transaksi GTID, dan transaksi harus berupa transaksi GTID untuk bisa direplikasi.
`enforce_gtid_consistency`	`OFF`, `ON`, `WARN`	`OFF` memperbolehkan transaksi melanggar konsistensi GTID. `ON` mencegah transaksi melanggar konsistensi GTID. `WARN` memperbolehkan transaksi melanggar konsistensi GTID, tetapi menghasilkan peringatan apabila terjadi pelanggaran.

> **Note**
>
> Dalam AWS Management Console, parameter `gtid_mode` muncul sebagai `gtid-mode`.

Untuk replikasi berbasis GTID, gunakan pengaturan ini untuk grup parameter pada instans DB atau replika baca Anda:

- `ON` dan `ON_PERMISSIVE` hanya berlaku pada replikasi keluar dari instans DB RDS. Kedua nilai ini menyebabkan instans DB RDS Anda menggunakan GTID untuk transaksi yang direplikasi. `ON` mengharuskan basis data target juga menggunakan replikasi berbasis GTID. `ON_PERMISSIVE` membuat replikasi berbasis GTID bersifat opsional di basis data target.
- `OFF_PERMISSIVE`, jika diatur, artinya instans DB RDS Anda dapat menerima replikasi masuk dari basis data sumber. Instans tersebut dapat melakukan ini terlepas dari apakah basis data sumber tersebut menggunakan replikasi berbasis GTID atau tidak.
- `OFF`, jika diatur, artinya instans DB RDS Anda hanya dapat menerima replikasi masuk dari basis data sumber yang tidak menggunakan replikasi berbasis GTID.

Untuk informasi selengkapnya tentang grup parameter, lihat Bekerja dengan grup parameter.

Pengonfigurasian replikasi berbasis GTID untuk replika baca baru

Jika replikasi berbasis GTID diaktifkan untuk instans DB RDS for MySQL, replikasi berbasis GTID dikonfigurasi secara otomatis untuk replika baca dari instans DB.

Untuk mengaktifkan replikasi berbasis GTID untuk replika baca baru

1. Pastikan grup parameter yang terkait dengan instans DB memiliki pengaturan parameter sebagai berikut:

   - `gtid_mode` – `ON` atau `ON_PERMISSIVE`
   - `enforce_gtid_consistency` – `ON`

   Untuk informasi selengkapnya tentang cara mengatur parameter konfigurasi menggunakan grup parameter, lihat Bekerja dengan grup parameter.
2. Jika Anda mengubah grup parameter dari instans DB, reboot instans DB tersebut. Untuk informasi selengkapnya tentang cara melakukannya, lihat Mem-boot ulang instans DB.
3. Ciptakan satu replika baca atau lebih dari instans DB. Untuk informasi selengkapnya tentang cara melakukannya, lihat Membuat replika baca.

Amazon RDS mencoba membuat replikasi berbasis GTID antara instans DB MySQL dan replika baca menggunakan `MASTER_AUTO_POSITION`. Jika upaya tersebut gagal, Amazon RDS menggunakan posisi file log untuk replikasi dengan replika baca. Untuk informasi selengkapnya tentang `MASTER_AUTO_POSITION`, lihat GTID auto-positioning dalam dokumentasi MySQL.

Pengonfigurasian replikasi berbasis GTID untuk replika baca yang sudah ada

Untuk instans DB MySQL yang sudah ada dengan replika baca yang tidak menggunakan replikasi berbasis GTID, Anda dapat mengonfigurasi replikasi berbasis GTID antara instans DB dan replika baca.

Untuk mengaktifkan replikasi berbasis GTID untuk replika baca yang sudah ada

1. Jika instans DB atau replika baca apa pun menggunakan RDS for MySQL versi 8.0 yang lebih rendah dari versi 8.0.26, tingkatkan instans DB atau replika baca ke 8.0.26 atau MySQL versi 8.0 yang lebih tinggi. Dukungan untuk replikasi berbasis GTID untuk RDS for MySQL versi 5.7

Untuk informasi selengkapnya, lihat Meng-upgrade mesin DB MySQL.

2. (Opsional) Atur ulang parameter GTID dan uji perilaku instans DB dan replika baca:

   a. Pastikan grup parameter yang terkait dengan instans DB MySQL dan setiap replika baca memiliki `enforce_gtid_consistency` yang diatur ke `WARN`.

      Untuk informasi selengkapnya tentang cara mengatur parameter konfigurasi menggunakan grup parameter, lihat Bekerja dengan grup parameter.

   b. Jika Anda mengubah grup parameter dari instans DB, reboot instans DB tersebut. Jika Anda mengubah grup parameter untuk sebuah replika baca, reboot replika baca tersebut.

      Untuk informasi selengkapnya, lihat Mem-boot ulang instans DB.

   c. Jalankan instans DB dan replika baca Anda dengan beban kerja normal Anda dan monitor file log.

      Jika Anda melihat peringatan tentang transaksi yang tidak kompatibel dengan GTID, sesuaikan agar aplikasi Anda hanya menggunakan fitur yang kompatibel dengan GTID. Pastikan instans DB tidak menghasilkan peringatan apa pun tentang transaksi yang tidak kompatibel dengan GTID sebelum melanjutkan ke langkah berikutnya.

3. Atur ulang parameter GTID untuk replikasi berbasis GTID yang memperbolehkan transaksi anonim hingga replika baca telah memproses semuanya.

   a. Pastikan grup parameter yang terkait dengan instans DB dan setiap replika baca memiliki pengaturan parameter sebagai berikut:

      - `gtid_mode` – `ON_PERMISSIVE`
      - `enforce_gtid_consistency` – `ON`

   b. Jika Anda mengubah grup parameter dari instans DB, reboot instans DB tersebut. Jika Anda mengubah grup parameter untuk sebuah replika baca, reboot replika baca tersebut.

4. Tunggu hingga semua transaksi anonim Anda direplikasi. Untuk memeriksa apakah semua transaksi tersebut sudah direplikasi, lakukan hal berikut:

   a. Jalankan pernyataan berikut pada instans DB sumber Anda.

      ```
      SHOW MASTER STATUS;
      ```

      Perhatikan nilai di kolom `File` dan `Position`.

b. Pada setiap replika baca, gunakan informasi file dan posisi dari instans sumber dalam langkah sebelumnya untuk menjalankan kueri berikut.

```
SELECT MASTER_POS_WAIT('file', position);
```

Misalnya, jika nama file-nya adalah `mysql-bin-changelog.000031` dan posisinya adalah `107`, jalankan pernyataan berikut.

```
SELECT MASTER_POS_WAIT('mysql-bin-changelog.000031', 107);
```

Jika replika baca melewati posisi yang ditentukan, kueri akan segera ditampilkan. Jika tidak, fungsi tersebut akan menunggu. Lanjutkan ke langkah berikutnya jika kueri menampilkan semua replika baca.

5. Atur ulang parameter GTID untuk replikasi berbasis GTID saja.

a. Pastikan grup parameter yang terkait dengan instans DB dan setiap replika baca memiliki pengaturan parameter sebagai berikut:

- `gtid_mode` – `ON`
- `enforce_gtid_consistency` – `ON`

b. Reboot instans DB dan setiap replika baca.

6. Pada setiap replika baca, jalankan prosedur berikut.

```
CALL mysql.rds_set_master_auto_position(1);
```

Menonaktifkan replikasi berbasis GTID untuk instans DB MySQL dengan replika baca

Anda dapat menonaktifkan replikasi berbasis GTID untuk instans DB MySQL dengan replika baca.

Untuk menonaktifkan replikasi berbasis GTID untuk instans DB MySQL dengan replika baca

1. Pada setiap replika baca, jalankan prosedur berikut.

```
CALL mysql.rds_set_master_auto_position(0); (Aurora MySQL version 2)
CALL mysql.rds_set_source_auto_position(0); (Aurora MySQL version 3)
```

2. Atur ulang `gtid_mode` ke `ON_PERMISSIVE`.

a. Pastikan grup parameter yang terkait dengan instans DB MySQL dan setiap replika baca memiliki `gtid_mode` yang diatur ke `ON_PERMISSIVE`.

   Untuk informasi selengkapnya tentang cara mengatur parameter konfigurasi menggunakan grup parameter, lihat Bekerja dengan grup parameter.

b. Reboot instans DB MySQL dan setiap replika baca. Untuk informasi selengkapnya tentang melakukan reboot, lihat Mem-boot ulang instans DB.

3. Atur ulang `gtid_mode` ke `OFF_PERMISSIVE`:

   a. Pastikan grup parameter yang terkait dengan instans DB MySQL dan setiap replika baca memiliki `gtid_mode` yang diatur ke `OFF_PERMISSIVE`.

   b. Reboot instans DB MySQL dan setiap replika baca.

4. Tunggu hingga semua transaksi GTID diterapkan pada semua replika baca. Untuk memastikan apakah semua ini diterapkan, lakukan hal berikut:

   Tunggu hingga semua transaksi GTID diterapkan pada instans primer Aurora. Untuk memastikan apakah semua ini diterapkan, lakukan hal berikut:

   a. Pada instans DB MySQL , jalankan perintah `SHOW MASTER STATUS`.

      Output Anda semestinya serupa dengan yang berikut ini.

```
File                        Position
-------------------------------------
mysql-bin-changelog.000031      107
-------------------------------------
```

      Perhatikan file dan posisi dalam output Anda.

   b. Pada setiap replika baca, gunakan informasi file dan posisi dari instans sumber dalam langkah sebelumnya untuk menjalankan kueri berikut.

```
SELECT MASTER_POS_WAIT('file', position);
```

      Misalnya, jika nama file-nya adalah `mysql-bin-changelog.000031` dan posisinya adalah `107`, jalankan pernyataan berikut.

```
SELECT MASTER_POS_WAIT('mysql-bin-changelog.000031', 107);
```

Jika replika baca melewati posisi yang ditentukan, kueri akan segera ditampilkan. Jika tidak, fungsi tersebut akan menunggu. Saat kueri menampilkan semua replika baca, lanjutkan ke langkah berikutnya.

5. Atur ulang parameter GTID untuk menonaktifkan replikasi berbasis GTID:

   a. Pastikan grup parameter yang terkait dengan instans DB MySQL dan setiap replika baca memiliki pengaturan parameter sebagai berikut:

      - `gtid_mode` – `OFF`
      - `enforce_gtid_consistency` – `OFF`

   b. Reboot instans DB MySQL dan setiap replika baca.

Mengonfigurasi replikasi posisi file log biner dengan instans sumber eksternal

Anda dapat menyiapkan replikasi antara instans DB RDS for MySQL atau MariaDB dan instans MySQL atau MariaDB yang berada di luar Amazon RDS menggunakan replikasi file log biner.

Topik

Sebelum Anda mulai

Anda dapat mengonfigurasi replikasi menggunakan posisi file log biner transaksi yang direplikasi.

Izin yang diperlukan untuk memulai replikasi pada instans DB Amazon RDS dibatasi dan tidak tersedia untuk pengguna master Amazon RDS Anda. Karena itu, pastikan Anda menggunakan perintah dan mysql.rds_start_replication Amazon RDS untuk mengatur replikasi antara basis data live dan basis data Amazon RDS Anda.

Untuk mengatur format pencatatan log biner untuk basis data MySQL atau MariaDB, perbarui parameter `binlog_format`. Jika instans DB Anda menggunakan grup parameter instans DB default, buat grup parameter DB baru untuk mengubah pengaturan `binlog_format`. Kami sarankan

Anda menggunakan pengaturan default untuk `binlog_format`, yaitu `MIXED`. Namun, Anda juga dapat mengatur `binlog_format` ke `ROW` atau `STATEMENT` jika Anda memerlukan format log biner (binlog) tertentu. Boot ulang instans DB Anda agar perubahan diterapkan.

Untuk informasi tentang mengatur parameter `binlog_format`, lihat Mengkonfigurasi pengelogan biner MySQL. Untuk informasi tentang implikasi tipe replikasi MySQL yang berbeda-beda, lihat Keuntungan dan kerugian replikasi berbasis pernyataan dan berbasis baris dalam dokumentasi MySQL.

Mengonfigurasi replikasi posisi file log biner dengan instans sumber eksternal

Ikuti pedoman ini saat Anda menyiapkan instans sumber eksternal dan replika di Amazon RDS:

- Pantau peristiwa failover untuk instans DB Amazon RDS yang merupakan replika Anda. Jika terjadi failover, maka instans DB yang merupakan replika Anda dapat dibuat ulang pada host baru dengan alamat jaringan yang berbeda. Untuk informasi tentang cara pemonitoran peristiwa failover, lihat Menggunakan pemberitahuan peristiwa Amazon RDS.
- Pertahankan binlog pada instans sumber Anda hingga Anda memverifikasi bahwa binlog tersebut telah diterapkan ke replika. Dengan mempertahankannya, Anda dapat memulihkan instans sumber Anda jika terjadi kegagalan.
- Aktifkan pencadangan otomatis pada instans DB Amazon RDS Anda. Dengan mengaktifkan pencadangan otomatis, Anda dapat memulihkan replika ke titik waktu tertentu jika Anda perlu menyinkronkan ulang instans sumber dan replika Anda. Untuk informasi tentang pencadangan dan point-in-time pemulihan, lihat. Mencadangkan, memulihkan, dan mengekspor data

Mengonfigurasi replikasi file log biner dengan Instans sumber eksternal

1. Jadikan instans MySQL atau MariaDB sumber sebagai hanya-baca.

```
mysql> FLUSH TABLES WITH READ LOCK;
mysql> SET GLOBAL read_only = ON;
```

2. Jalankan perintah `SHOW MASTER STATUS` pada instans MySQL atau MariaDB sumber untuk menentukan lokasi binlog.

   Anda menerima output yang mirip dengan contoh berikut.

```
File                      Position
-----------------------------------
```

```
 mysql-bin-changelog.000031      107
-----------------------------------
```

3. Salin basis data dari instans eksternal ke instans DB Amazon RDS menggunakan `mysqldump`. Untuk basis data yang sangat besar, Anda mungkin ingin menggunakan prosedur di Mengimpor data ke basis data Amazon RDS MariaDB atau MySQL dengan lebih sedikit waktu henti.

   Untuk Linux, macOS, atau Unix:

```
mysqldump --databases database_name \
    --single-transaction \
    --compress \
    --order-by-primary \
    -u local_user \
    -plocal_password | mysql \
        --host=hostname \
        --port=3306 \
        -u RDS_user_name \
        -pRDS_password
```

   Untuk Windows:

```
mysqldump --databases database_name ^
    --single-transaction ^
    --compress ^
    --order-by-primary ^
    -u local_user ^
    -plocal_password | mysql ^
        --host=hostname ^
        --port=3306 ^
        -u RDS_user_name ^
        -pRDS_password
```

> **Note**
>
> Pastikan tidak ada spasi di antara opsi `-p` dan sandi yang dimasukkan.

   Untuk menentukan nama host, nama pengguna, port, dan kata sandi untuk menghubungkan ke instans Amazon RDS DB Anda, gunakan opsi `--host`, `--user (-u)`, `--port`, dan `-p` dalam

perintah `mysql`. Nama host adalah nama Domain Name Service (DNS) dari titik akhir instans DB Amazon RDS, misalnya `myinstance.123456789012.us-east-1.rds.amazonaws.com`. Anda dapat menemukan nilai titik akhir dalam detail instans di AWS Management Console.

4. Jadikan instans DB MySQL atau MariaDB sumber sebagai writable (dapat diubah) lagi.

```
mysql> SET GLOBAL read_only = OFF;
mysql> UNLOCK TABLES;
```

Untuk informasi lebih lanjut tentang cara membuat cadangan untuk digunakan dengan replikasi, lihat dokumentasi MySQL.

5. Di AWS Management Console, tambahkan alamat IP server yang meng-host basis data eksternal ke grup keamanan cloud privat virtual (VPC) untuk instans DB Amazon RDS. Untuk informasi selengkapnya tentang cara memodifikasi grup keamanan VPC, lihat Grup keamanan untuk VPC Anda dalam Panduan Pengguna Amazon Virtual Private Cloud.

Alamat IP dapat berubah jika kondisi berikut terpenuhi:

- Anda menggunakan alamat IP publik untuk komunikasi antara instans sumber eksternal dan instans basis data.
- Instans sumber eksternal dihentikan dan dimulai ulang.

Jika semua kondisi ini terpenuhi, verifikasikan alamat IP sebelum menambahkannya.

Anda mungkin juga perlu mengonfigurasi jaringan lokal Anda untuk mengizinkan koneksi dari alamat IP instans Amazon RDS DB Anda. Anda melakukan ini agar jaringan lokal Anda dapat berkomunikasi dengan instans MySQL atau MariaDB eksternal Anda. Untuk menemukan alamat IP dari instans Amazon RDS DB, gunakan perintah `host`.

```
host db_instance_endpoint
```

Nama host adalah nama DNS dari titik akhir instans DB Amazon RDS.

6. Menggunakan klien pilihan Anda, hubungkan ke instans eksternal dan buat pengguna untuk digunakan untuk replikasi. Gunakan akun ini semata-mata untuk replikasi dan batasi hanya untuk domain Anda guna meningkatkan keamanan. Berikut adalah contohnya.

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
```

> **Note**
>
> Tentukan kata sandi selain prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

7. Untuk instans eksternal, berikan hak akses `REPLICATION CLIENT` dan `REPLICATION SLAVE` kepada pengguna replikasi Anda. Misalnya, untuk memberikan hak akses `REPLICATION CLIENT` dan `REPLICATION SLAVE` pada semua basis data untuk pengguna '`repl_user`' bagi domain Anda, jalankan perintah berikut.

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
```

8. Jadikan instans Amazon RDS DB sebagai replika. Untuk melakukannya, pertama-tama hubungkan ke instans DB Amazon RDS sebagai pengguna master. Kemudian identifikasi basis data MySQL atau MariaDB eksternal sebagai instans sumber menggunakan perintah . Gunakan nama file log master dan posisi log master yang Anda tentukan pada langkah 2. Berikut adalah contohnya.

```
CALL mysql.rds_set_external_master ('mymasterserver.mydomain.com', 3306,
 'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 0);
```

> **Note**
>
> Di RDS for MySQL, Anda dapat memilih untuk menggunakan replikasi tertunda dengan menjalankan prosedur tersimpan mysql.rds_set_external_master_with_delay sebagai gantinya. Di RDS for MySQL, salah satu alasan menggunakan replikasi tertunda adalah untuk mengaktifkan pemulihan bencana dengan prosedur tersimpan mysql.rds_start_replication_until. Saat ini, RDS untuk MariaDB mendukung replikasi tertunda tetapi tidak mendukung prosedur `mysql.rds_start_replication_until`.

9. Di instans DB Amazon RDS, terbitkan perintah mysql.rds_start_replication untuk memulai replikasi.

```
CALL mysql.rds_start_replication;
```

Mengkonfigurasi multi-source-replication untuk RDS untuk MySQL

Dengan replikasi multi-sumber, Anda dapat menyiapkan instans Amazon RDS for MySQL DB sebagai replika yang menerima peristiwa log biner dari lebih dari satu RDS untuk instans DB sumber MySQL. Replikasi multi-sumber didukung untuk RDS untuk instance MySQL DB yang menjalankan versi mesin berikut:

- 8.0.35 dan versi minor yang lebih tinggi
- 5.7.44 dan versi minor yang lebih tinggi

Untuk informasi tentang replikasi multi-sumber MySQL, lihat Replikasi Multi-Sumber MySQL dalam dokumentasi MySQL. Dokumentasi MySQL berisi informasi rinci tentang fitur ini, sementara topik ini menjelaskan cara mengkonfigurasi dan mengelola saluran replikasi multi-sumber pada RDS Anda untuk instance MySQL DB.

Topik

Kasus penggunaan untuk replikasi multi-sumber

Kasus-kasus berikut adalah kandidat yang baik untuk menggunakan replikasi multi-sumber pada RDS untuk MySQL:

- Aplikasi yang perlu menggabungkan atau menggabungkan beberapa pecahan pada instance DB terpisah menjadi satu pecahan.
- Aplikasi yang perlu menghasilkan laporan dari data yang dikonsolidasikan dari berbagai sumber.
- Persyaratan untuk membuat cadangan data jangka panjang terkonsolidasi yang didistribusikan di antara beberapa RDS untuk instance MySQL DB.

Pertimbangan dan praktik terbaik untuk replikasi multi-sumber

Sebelum Anda menggunakan replikasi multi-sumber pada RDS untuk MySQL, tinjau pertimbangan dan praktik terbaik berikut:

- Pastikan instans DB yang dikonfigurasi sebagai replika multi-sumber memiliki sumber daya yang cukup seperti throughput, memori, CPU, dan IOPS untuk menangani beban kerja dari beberapa instance sumber.
- Pantau pemanfaatan sumber daya secara teratur pada replika multi-sumber Anda dan sesuaikan konfigurasi penyimpanan atau instans untuk menangani beban kerja tanpa membebani sumber daya.
- Anda dapat mengonfigurasi replikasi multi-utas pada replika multi-sumber dengan mengatur variabel sistem `replica_parallel_workers` atau `slave_parallel_workers` ke nilai yang lebih besar dari 0. Dalam hal ini, jumlah utas yang dialokasikan untuk setiap saluran adalah nilai variabel ini, ditambah satu utas koordinator untuk mengelola utas applier.
- Konfigurasikan filter replikasi dengan tepat untuk menghindari konflik. Anda dapat mengkonfigurasi `replica_rewrite_db` untuk mereplikasi seluruh database ke database lain pada replika. Misalnya, semua tabel dalam database A dapat direplikasi ke database B pada contoh replika. Pendekatan ini dapat membantu ketika semua instance sumber menggunakan konvensi penamaan skema yang sama.
- Untuk menghindari kesalahan replikasi, hindari menulis ke replika. Kami menyarankan Anda mengaktifkan `read_only` parameter pada replika multi-sumber untuk memblokir operasi penulisan. Melakukannya membantu menghilangkan masalah replikasi yang disebabkan oleh operasi penulisan yang bertentangan.
- Untuk meningkatkan kinerja operasi baca seperti jenis dan gabungan beban tinggi yang dijalankan pada replika multi-sumber, pertimbangkan untuk menggunakan RDS Optimized Reads. Fitur ini dapat membantu dengan kueri yang bergantung pada tabel sementara besar atau mengurutkan file. Untuk informasi selengkapnya, lihat the section called “Meningkatkan performa kueri dengan RDS Optimized Reads”.
- Untuk meminimalkan kelambatan replikasi dan meningkatkan kinerja replika multi-sumber, pertimbangkan untuk mengaktifkan penulisan yang dioptimalkan. Untuk informasi selengkapnya, lihat the section called “Meningkatkan performa penulisan dengan RDS Optimized Writes for MySQL”.
- Lakukan operasi manajemen (seperti mengubah konfigurasi) pada satu saluran pada satu waktu, dan hindari melakukan perubahan pada beberapa saluran dari beberapa koneksi. Praktik-praktik ini dapat menyebabkan konflik dalam operasi replikasi. Misalnya, mengeksekusi

`rds_skip_repl_error_for_channel` dan `rds_start_replication_for_channel` prosedur secara bersamaan dari beberapa koneksi dapat menyebabkan melewatkan peristiwa pada saluran yang berbeda dari yang dimaksudkan.

- Anda dapat mengaktifkan pencadangan pada instans replikasi multi-sumber dan mengekspor data dari instans tersebut ke bucket Amazon S3 untuk menyimpannya untuk tujuan jangka panjang. Namun, penting juga untuk mengonfigurasi cadangan dengan retensi yang sesuai pada instance sumber individual. Untuk informasi tentang mengekspor data snapshot ke Amazon S3, lihat. the section called “Mengekspor data snapshot DB ke Amazon S3”
- Untuk mendistribusikan beban kerja baca pada replika multi-sumber, Anda dapat membuat replika baca dari replika multi-sumber. Anda dapat menemukan replika baca ini secara berbeda Wilayah AWS berdasarkan persyaratan aplikasi Anda. Untuk informasi selengkapnya tentang replika baca, lihat the section called “Menggunakan replika baca MySQL”.

Prasyarat untuk replikasi multi-sumber

Sebelum Anda mengonfigurasi replikasi multi-sumber, selesaikan prasyarat berikut.

- Pastikan bahwa setiap sumber RDS untuk MySQL DB instance memiliki backup otomatis diaktifkan. Mengaktifkan backup otomatis memungkinkan logging biner. Untuk mempelajari cara mengaktifkan pencadangan otomatis, lihat. the section called “Mengaktifkan pencadangan otomatis”
- Untuk menghindari kesalahan replikasi, sebaiknya Anda memblokir operasi penulisan ke instans DB sumber. Anda dapat melakukannya dengan mengatur `read-only` parameter ke `0N` dalam grup parameter khusus yang dilampirkan ke RDS untuk instance DB sumber MySQL. Anda dapat menggunakan AWS Management Console atau AWS CLI untuk membuat grup parameter kustom baru atau untuk memodifikasi yang sudah ada. Lihat informasi yang lebih lengkap di the section called “Membuat grup parameter DB” dan the section called “Memodifikasi parameter dalam grup parameter DB”.
- Untuk setiap instans DB sumber, tambahkan alamat IP instans ke grup keamanan Amazon virtual private cloud (VPC) untuk instans DB multi-sumber. Untuk mengidentifikasi alamat IP dari instans DB sumber, Anda dapat menjalankan perintah`dig` *`RDS Endpoint`*. Jalankan perintah dari instans Amazon EC2 di VPC yang sama dengan instans DB multi-sumber tujuan.
- Untuk setiap instans DB sumber, gunakan klien untuk terhubung ke instans DB dan buat pengguna database dengan hak istimewa yang diperlukan untuk replikasi, seperti pada contoh berikut.

```
CREATE USER 'repl_user' IDENTIFIED BY 'password';
```

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user';
```

Mengkonfigurasi saluran replikasi multi-sumber pada RDS untuk instans MySQL DB

Mengkonfigurasi saluran replikasi multi-sumber mirip dengan mengkonfigurasi replikasi sumber tunggal. Untuk replikasi multi-sumber, pertama-tama Anda mengaktifkan logging biner pada instance sumber. Kemudian, Anda mengimpor data dari sumber ke replika multi-sumber. Kemudian, Anda memulai replikasi dari setiap sumber dengan menggunakan koordinat log biner atau dengan menggunakan pemosisian otomatis GTID.

Untuk mengonfigurasi RDS untuk instance MySQL DB sebagai replika multi-sumber dari dua atau lebih RDS untuk instance MySQL DB, lakukan langkah-langkah berikut.

Topik

Langkah 1: Impor data dari instans DB sumber ke replika multi-sumber

Lakukan langkah-langkah berikut pada setiap instans DB sumber.

Sebelum Anda mengimpor data dari sumber ke replika multi-sumber, tentukan file log biner saat ini dan posisikan dengan menjalankan perintah. `SHOW MASTER STATUS` Catat detail ini untuk digunakan pada langkah berikutnya. Dalam contoh output ini, file adalah `mysql-bin-changelog.000031` dan posisinya adalah`107`.

```
File                        Position
-----------------------------------
mysql-bin-changelog.000031      107
-----------------------------------
```

Sekarang salin database dari instance DB sumber ke replika multi-sumber dengan menggunakan`mysqldump`, seperti pada contoh berikut.

```
mysqldump --databases database_name \
 --single-transaction \
 --compress \
 --order-by-primary \
 -u RDS_user_name \
```

```
 -p RDS_password \
 --host=RDS Endpoint | mysql \
 --host=RDS Endpoint \
 --port=3306 \
 -u RDS_user_name \
-p RDS_password
```

Setelah menyalin database, Anda dapat mengatur parameter read-only ke instans OFF DB sumber.

Langkah 2: Mulai replikasi dari instans DB sumber ke replika multi-sumber

Untuk setiap instans DB sumber, gunakan kredensi pengguna master untuk terhubung ke instance, dan jalankan dua prosedur tersimpan berikut. Prosedur tersimpan ini mengonfigurasi replikasi pada saluran dan memulai replikasi. Contoh ini menggunakan nama file binlog dan posisi dari output contoh pada langkah sebelumnya.

```
CALL mysql.rds_set_external_source_for_channel('mysourcehost.example.com', 3306,
 'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 0, 'channel_1');
CALL mysql.rds_start_replication_for_channel('channel_1');
```

Untuk informasi selengkapnya tentang menggunakan prosedur tersimpan ini dan lainnya untuk menyiapkan dan mengelola saluran replikasi Anda, lihatthe section called “Mengelola replikasi multi-sumber”.

Menggunakan filter dengan replikasi multi-sumber

Anda dapat menggunakan filter replikasi untuk menentukan database dan tabel mana yang direplikasi dalam replika multi-sumber. Filter replikasi dapat menyertakan basis data dan tabel ke dalam replikasi atau mengecualikan mereka dari replikasi. Untuk informasi selengkapnya tentang filter replikasi, lihatthe section called “Mengonfigurasi filter replikasi dengan MySQL”.

Dengan replikasi multi-sumber, Anda dapat mengonfigurasi filter replikasi secara global atau di tingkat saluran. Pemfilteran tingkat saluran hanya tersedia dengan instans DB yang didukung yang menjalankan versi 8.0. Contoh berikut menunjukkan cara mengonfigurasi filter secara global atau di tingkat saluran.

Perhatikan persyaratan dan perilaku berikut dengan pemfilteran dalam replikasi multi-sumber:

- Kutipan belakang (``) di sekitar nama saluran diperlukan.
- Jika Anda mengubah filter replikasi dalam grup parameter, replika multi-sumber `sql_thread` untuk semua saluran dengan pembaruan akan dimulai ulang untuk menerapkan perubahan secara

dinamis. Jika pembaruan melibatkan filter global, maka semua saluran replikasi dalam status berjalan dimulai ulang.

- Semua filter global diterapkan sebelum filter khusus saluran apa pun.
- Jika filter diterapkan secara global dan pada tingkat saluran, maka hanya filter tingkat saluran yang diterapkan. Misalnya, jika filternya`replicate_ignore_db="db1,`channel_22`:db2"`, maka `replicate_ignore_db` disetel ke `db1` diterapkan ke semua saluran kecuali untuk`channel_22`, dan hanya `channel_22` mengabaikan perubahan dari`db2`.

Contoh 1: Mengatur filter global

Dalam contoh berikut, `temp_data` database dikecualikan dari replikasi di setiap saluran.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-parameter-group \
--db-parameter-group-name myparametergroup \
--parameters "ParameterName=replicate-ignore-
db,ParameterValue='temp_data',ApplyMethod=immediate"
```

Contoh 2: Mengatur filter tingkat saluran

Dalam contoh berikut, perubahan dari `sample22` database hanya disertakan dalam saluran`channel_22`. Demikian pula, perubahan dari `sample99` database hanya disertakan dalam saluran`channel_99`.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-parameter-group \
--db-parameter-group-name myparametergroup \
--parameters "ParameterName=replicate-do-db,ParameterValue='\`channel_22\`:sample22,
\`channel_99\`:sample99',ApplyMethod=immediate"
```

Memantau saluran replikasi multi-sumber

Anda dapat memantau saluran individual dalam replika multi-sumber dengan menggunakan metode berikut:

- Untuk memantau status semua saluran atau saluran tertentu, sambungkan ke replika multi-sumber dan jalankan perintah `SHOW REPLICA STATUS` atau`SHOW REPLICA STATUS FOR CHANNEL`

`'channel_name'`. Untuk informasi selengkapnya, lihat Memeriksa Status Replikasi dalam dokumentasi MySQL.

- Untuk menerima pemberitahuan saat saluran replikasi dimulai, dihentikan, atau dihapus, gunakan pemberitahuan acara RDS. Untuk informasi selengkapnya, lihat the section called “Menggunakan pemberitahuan peristiwa Amazon RDS”.
- Untuk memantau lag untuk saluran tertentu, periksa `ReplicationChannelLag` metrik untuk itu. Titik data untuk metrik ini memiliki jangka waktu 60 detik (1 menit) yang tersedia selama 15 hari. Untuk menemukan lag saluran replikasi untuk saluran, gunakan pengenal instance dan nama saluran replikasi. Untuk menerima pemberitahuan ketika lag ini melebihi ambang tertentu, Anda dapat mengatur CloudWatch alarm. Untuk informasi selengkapnya, lihat the section called “Memantau RDS dengan CloudWatch”.

Keterbatasan untuk replikasi multi-sumber pada RDS untuk MySQL

Batasan berikut berlaku untuk replikasi multi-sumber pada RDS untuk MySQL:

- Saat ini, RDS untuk MySQL mendukung konfigurasi maksimum 15 saluran untuk replika multi-sumber.
- Instance replika baca tidak dapat dikonfigurasi sebagai replika multi-sumber.
- Untuk mengonfigurasi replikasi multi-sumber pada RDS untuk MySQL yang menjalankan engine versi 5.7, Performance Schema harus diaktifkan pada instance replika. Mengaktifkan Skema Kinerja adalah opsional pada RDS untuk MySQL yang menjalankan engine versi 8.0.
- Untuk RDS untuk MySQL yang menjalankan engine versi 5.7, filter replikasi berlaku untuk semua saluran replikasi. Untuk RDS untuk MySQL yang menjalankan engine versi 8.0, Anda dapat mengonfigurasi filter yang berlaku untuk semua saluran replikasi atau saluran individual.
- Memulihkan snapshot RDS atau melakukan P oint-in-time -Restore (PITR) tidak memulihkan konfigurasi saluran replika multi-sumber.
- Saat Anda membuat replika baca dari replika multi-sumber, itu hanya mereplikasi data dari instance multi-sumber. Itu tidak mengembalikan konfigurasi saluran apa pun.
- MySQL tidak mendukung pengaturan jumlah pekerja paralel yang berbeda untuk setiap saluran. Setiap saluran mendapatkan jumlah pekerja paralel yang sama berdasarkan `replica_parallel_workers` nilainya.

Batasan tambahan berikut berlaku jika target replikasi multi-sumber Anda adalah cluster DB multi-AZ:

- Saluran harus dikonfigurasi untuk sumber RDS untuk instance MySQL sebelum penulisan ke instance itu terjadi.
- Setiap sumber RDS untuk instance MySQL harus mengaktifkan replikasi berbasis GTID.
- Peristiwa failover pada cluster DB menghapus konfigurasi replikasi multi-sumber. Memulihkan konfigurasi itu membutuhkan pengulangan langkah-langkah konfigurasi.

Mengonfigurasikan klaster aktif-aktif untuk RDS for MySQL

Anda dapat mengatur klaster aktif-aktif untuk RDS for MySQL dengan menggunakan pengaya MySQL Group Replication. Pengaya Group Replication didukung untuk instans basis data RDS for MySQL yang menjalankan versi 8.0.35 dan versi-versi kecil yang lebih tinggi.

Lihat informasi tentang MySQL Group Replication di Replikasi Grup dalam dokumentasi MySQL. Dokumentasi MySQL berisi informasi terperinci tentang fitur ini, sementara topik ini menjelaskan cara mengonfigurasikan dan mengelola pengaya pada instans basis data RDS for MySQL Anda.

> **Note**
>
> Singkat cerita, semua penyebutan klaster "aktif-aktif" dalam topik ini mengacu kepada klaster aktif-aktif yang menggunakan pengaya MySQL Group Replication.

Topik

Kasus penggunaan untuk klaster aktif-aktif

Kasus-kasus berikut adalah calon yang baik untuk menggunakan klaster aktif-aktif:

- Aplikasi yang membutuhkan semua instans basis data di dalam klaster untuk mendukung operasi tulis. Pengaya Group Replication menjaga data tetap konsisten pada setiap instans basis data di dalam klaster aktif-aktif. Lihat informasi yang lebih lengkap tentang cara kerjanya di Replikasi Grup dalam dokumentasi MySQL.
- Aplikasi yang membutuhkan ketersediaan sinambung basis data. Dengan klaster aktif-aktif, data disimpan pada semua instans basis data di dalam klaster. Jika satu instans basis data gagal, aplikasi dapat mengalihkan lalu lintas ke instans basis data yang lain di dalam klaster.
- Aplikasi yang mungkin perlu membagi operasi baca dan tulis di antara instans basis data yang berbeda di klaster untuk tujuan penyeimbangan beban. Dengan klaster aktif-aktif, aplikasi Anda dapat mengirim lalu lintas baca ke instans basis data tertentu dan menulis lalu lintas ke instans lain. Anda juga dapat mengganti kapan saja instans basis data yang akan dikirim lalu lintas baca atau tulis.

Pertimbangan dan praktik terbaik untuk klaster aktif-aktif

Sebelum Anda menggunakan klaster aktif-aktif RDS for MySQL, tinjau beberapa pertimbangan dan praktik terbaik berikut:

- Klaster aktif-aktif tidak dapat memiliki lebih dari sembilan instans basis data.
- Dengan pengaya Group Replication, Anda dapat mengendalikan jaminan konsistensi transaksi klaster aktif-aktif. Lihat informasi yang lebih lengkap di Jaminan Konsistensi Transaksi dalam dokumentasi MySQL.
- Konflik mungkin terjadi ketika instans basis data yang berbeda memperbarui baris yang sama di klaster aktif-aktif. Lihat informasi tentang konflik dan resolusi konflik di Replikasi Grup dalam dokumentasi MySQL.
- Untuk toleransi kesalahan, sertakan setidaknya tiga instans basis data di klaster aktif-aktif Anda. Mengonfigurasikan klaster aktif-aktif hanya dengan satu atau dua instans basis data dapat dilakukan, tetapi klaster tidak akan toleran terhadap kesalahan. Lihat informasi tentang toleransi kesalahan di Toleransi kesalahan dalam dokumentasi MySQL.
- Ketika instans basis data bergabung dengan klaster aktif-aktif yang ada dan menjalankan versi mesin yang sama dengan versi mesin terendah di klaster, instans basis data bergabung dalam mode baca tulis.
- Ketika instans basis data bergabung dengan klaster aktif-aktif yang ada dan menjalankan versi mesin yang lebih tinggi daripada versi mesin terendah di klaster, instans basis data harus tetap dalam mode hanya-baca.

- Jika Anda mengaktifkan Replikasi Grup untuk instans DB dengan menyetel `rds.group_replication_enabled` parameternya ke `1` dalam grup parameter DB, tetapi replikasi belum dimulai atau gagal dimulai, instans DB ditempatkan dalam super-read-only mode untuk mencegah inkonsistensi data. Untuk informasi tentang super-read-only mode, lihat dokumentasi MySQL.
- Anda dapat memutakhirkan instans basis data di klaster aktif-aktif, tetapi instans basis data akan hanya baca hingga semua instans basis data lain di klaster aktif-aktif ditingkatkan ke versi mesin yang sama atau versi mesin yang lebih tinggi. Saat Anda memutakhirkan instans basis data, instans basis data bergabung secara otomatis dengan klaster aktif-aktif yang sama saat pemutakhiran selesai. Untuk menghindari peralihan yang tidak diinginkan ke mode hanya baca untuk instans basis data, nonaktifkan peningkatan versi kecil otomatis untuknya. Lihat informasi tentang pemutakhiran instans basis data MySQL di Meng-upgrade mesin DB MySQL.
- Anda dapat menambahkan instans basis data dalam deployment instans basis data Multi-AZ ke klaster aktif-aktif yang ada. Anda juga dapat mengonversi instans basis data AZ Tunggal dalam klaster aktif-aktif menjadi deployment instans basis data Multi-AZ. Jika instans basis data utama dalam deployment Multi-AZ gagal, instans utama melakukan pindah saat gagal/failover ke instans siaga. Instans basis data utama baru bergabung secara otomatis dengan klaster yang sama setelah pindah saat gagal/failover selesai. Lihat informasi yang lebih lengkap tentang deployment instans basis data Multi-AZ di Deployment instans DB Multi-AZ.
- Kami menganjurkan supaya instans-instans basis data dalam klaster aktif-aktif memiliki rentang waktu yang berbeda-beda untuk jendela pemeliharaannya. Praktik ini menghindari beberapa instans basis data di klaster menjadi serentak luring selama pemeliharaan. Lihat informasi yang lebih lengkap di Periode pemeliharaan Amazon RDS.
- Klaster aktif-aktif dapat menggunakan SSL untuk koneksi di antara instans-instans basis data. Untuk mengonfigurasikan koneksi SSL, atur parameter-parameter group_replication_recovery_use_ssl dan group_replication_ssl_mode. Nilai untuk parameter-parameter ini harus cocok untuk semua instans basis data di klaster aktif-aktif.

  Saat ini, klaster aktif-aktif tidak mendukung verifikasi otoritas sertifikat (CA) untuk koneksi di antara Wilayah AWS. Jadi, parameter group_replication_ssl_mode harus diatur ke `DISABLED` (bawaan) atau `REQUIRED` untuk klaster lintas Kawasan.
- Sebuah klaster aktif-aktif RDS for MySQL berjalan dalam mode multi-utama. Nilai bawaan group_replication_enforce_update_everywhere_checks adalah `ON` dan parameter ini statis. Ketika parameter ini diatur ke `ON`, aplikasi tidak dapat menyisipkan ke dalam tabel yang memiliki kendala kunci asing berjenjang.

- Klaster aktif-aktif RDS for MySQL menggunakan tumpukan komunikasi MySQL alih-alih XCOM untuk keamanan koneksi. Lihat informasi yang lebih lengkap di Tumpukan Komunikasi untuk Pengelolaan Keamanan Koneksi dalam dokumentasi MySQL.
- Ketika grup parameter basis data dikaitkan dengan instans basis data dalam klaster aktif-aktif, kami menyarankan supaya hanya mengaitkan grup parameter basis data ini dengan instans-instans basis data lain yang ada di klaster.
- Klaster aktif-aktif hanya mendukung instans basis data RDS for MySQL. Instans basis data ini harus menjalankan versi mesin basis data yang didukung.
- Ketika instans basis data di klaster aktif-aktif mengalami kegagalan yang tidak terduga, RDS memulai secara otomatis pemulihan instans basis data. Jika instans DB tidak pulih, kami sarankan untuk menggantinya dengan instans DB baru dengan melakukan point-in-time pemulihan dengan instans DB yang sehat di cluster. Untuk petunjuk, lihat Menambahkan instans DB ke cluster aktif-aktif menggunakan pemulihan point-in-time .
- Anda dapat menghapus instans basis data di klaster aktif-aktif tanpa memengaruhi instans basis data yang lain di situ. Lihat informasi tentang menghapus instans basis data di Menghapus instans DB.

Prasyarat untuk klaster aktif-aktif lintas VPC

Anda dapat mengonfigurasikan klaster aktif-aktif dengan instans basis data di lebih dari satu VPC. VPC boleh berada di Wilayah AWS yang sama atau Wilayah AWS yang berbeda.

> **Note**
>
> Mengirim lalu lintas di antara beberapa Wilayah AWS mungkin menimbulkan biaya tambahan. Lihat informasi yang lebih lengkap di Ikhtisar Biaya Transfer Data untuk Arsitektur Umum.

Jika Anda mengonfigurasikan klaster aktif-aktif dalam satu VPC, Anda dapat melewati langkah-langkah ini dan melanjutkan ke Menyiapkan klaster aktif-aktif dengan instans basis data baru.

Untuk mempersiapkan klaster aktif-aktif dengan instans basis data di lebih dari satu VPC

1. Pastikan bahwa rentang alamat IPv4 di blok CIDR memenuhi persyaratan berikut:
   - Rentang alamat IPv4 di blok CIDR dari VPC tidak boleh bertumpang tindih.

- Semua rentang alamat IPv4 di blok CIDR harus lebih rendah dari `128.0.0.0/`*`subnet_mask`* atau lebih tinggi dari 128.0.0.0/*`subnet_mask`*.

  Rentang-rentang berikut menggambarkan persyaratan ini:

  - `10.1.0.0/16` di satu VPC dan `10.2.0.0/16` di VPC yang lain didukung.
  - `172.1.0.0/16` di satu VPC dan `172.2.0.0/16` di VPC yang lain didukung.
  - `10.1.0.0/16` di satu VPC dan `10.1.0.0/16` di VPC yang lain tidak didukung karena bertumpang tindih.
  - `10.1.0.0/16` di satu VPC dan `172.1.0.0/16` di VPC yang lain tidak didukung karena yang satu di bawah `128.0.0.0/`*`subnet_mask`* dan yang lain di atas `128.0.0.0/`*`subnet_mask`*.

  Lihat informasi tentang blok CIDR di Blok CIDR VPC dalam Panduan Pengguna Amazon VPC.

2. Di setiap VPC, pastikan bahwa resolusi DNS dan nama host DNS keduanya diaktifkan.

   Lihat petunjuknya di Melihat dan memperbarui atribut-atribut DNS untuk VPC dalam Panduan Pengguna Amazon VPC.

3. Konfigurasikan semua VPC sehingga Anda dapat merutekan lalu lintas di antara VPC dengan salah satu cara berikut:

   - Buat koneksi peering VPC di antara VPC-VPC.

     Lihat petunjuknya di Membuat koneksi peering VPC dalam Panduan Perekanan/Peering Amazon VPC. Di setiap VPC, pastikan ada aturan masuk untuk grup keamanan Anda yang merujuk ke grup keamanan di VPC yang menjadi rekan/peer. Melakukan hal itu memungkinkan lalu lintas mengalir ke dan dari instans yang terkait dengan grup keamanan yang dirujuk di VPC yang menjadi rekan/peer. Lihat petunjuknya di Memperbarui grup keamanan untuk merujuk ke grup keamanan rekan dalam Panduan Perekanan/Peering Amazon VPC.

   - Buat gateway transit di antara VPC-VPC.

     Lihat petunjuknya di Memulai gateway transit dalam Gerbang Transit Amazon VPC. Di setiap VPC, pastikan ada aturan masuk untuk grup keamanan Anda yang mengizinkan lalu lintas dari VPC lain, seperti aturan masuk yang menentukan CIDR VPC yang lain. Melakukan hal itu akan memungkinkan lalu lintas mengalir ke dan dari instans yang dikaitkan dengan grup keamanan yang dirujuk di klaster aktif-aktif. Lihat informasi yang lebih lengkap di

Mengendalikan lalu lintas ke sumber daya AWS dengan menggunakan grup keamanan dalam Panduan Pengguna Amazon VPC.

Setelan parameter yang diperlukan untuk klaster aktif-aktif

Setelan parameter berikut diperlukan saat Anda menyiapkan klaster aktif-aktif RDS for MySQL.

Parameter	Deskripsi	Setelan diperlukan
`binlog_format`	Mengatur format pengelogan biner. Nilai bawaan untuk RDS for MySQL adalah `MIXED`. Lihat informasi yang lebih lengkap dalam dokumentasi MySQL.	`ROW`
`enforce_gtid_consistency`	Menerapkan konsistensi GTID untuk eksekusi pernyataan. Nilai bawaan untuk RDS for MySQL adalah `OFF`. Lihat informasi yang lebih lengkap dalam dokumentasi MySQL.	`ON`
`group_replication_group_name`	Menetapkan nama Group Replication ke UUID. Format UUID adalah `11111111-2222-3333-4444-555555555555`. Anda dapat menghasilkan UUID MySQL dengan menghubungi instans basis data MySQL dan menjalankan `SELECT UUID()`. Nilai harus sama untuk semua instans basis data di klaster aktif-aktif. Lihat informasi yang lebih lengkap dalam dokumentasi MySQL.	UUID MySQL

Parameter	Deskripsi	Setelan diperlukan
`gtid-mode`	Mengendalikan pengelogan berbasis GTID. Nilai bawaan untuk RDS for MySQL adalah `OFF_PERMISSIVE` . Lihat informasi yang lebih lengkap dalam dokumentasi MySQL.	`ON`
`rds.custom_dns_resolution`	Menentukan apakah mengizinkan resolusi DNS dari server DNS Amazon di VPC Anda. Resolusi DNS harus diaktifkan apabila Group Replication diaktifkan dengan parameter `rds.group_replication_enabled` . Resolusi DNS tidak dapat diaktifkan apabila Group Replication dinonaktifkan dengan parameter `rds.group_replication_enabled` . Lihat informasi yang lebih lengkap di Server DNS Amazon dalam Panduan Pengguna Amazon VPC.	`1`
`rds.group_replication_enabled`	Menentukan apakah Group Replication diaktifkan untuk instans basis data. Group Replication harus diaktifkan pada instans basis data dalam klaster aktif-aktif.	`1`

Parameter	Deskripsi	Setelan diperlukan
`slave_preserve_commit_order`	Mengendalikan urutan transaksi dituntaskan/commit pada replika. Nilai bawaan untuk RDS for MySQL adalah `ON`. Lihat informasi yang lebih lengkap dalam dokumentasi MySQL.	`ON`

Mengonversi instans basis data yang ada ke klaster aktif-aktif

Versi mesin basis data dari instans basis data yang ingin Anda migrasikan ke klaster aktif-aktif harus MySQL 8.0.35 atau lebih tinggi. Jika Anda perlu memutakhirkan versi mesin, lihat Meng-upgrade mesin DB MySQL.

Jika Anda menyiapkan klaster aktif-aktif dengan instans basis data di lebih dari satu VPC, pastikan untuk menyelesaikan prasyarat di Prasyarat untuk klaster aktif-aktif lintas VPC.

Selesaikan langkah-langkah berikut untuk memigrasikan instans basis data yang ada ke klaster aktif-aktif untuk RDS for MySQL.

Topik

- Langkah 1: Atur parameter klaster aktif-aktif dalam satu atau beberapa grup parameter kustom
- Langkah 2: Kaitkan instans basis data dengan grup parameter basis data yang telah mengatur parameter Group Replication yang diperlukan
- Langkah 3: Buat klaster aktif-aktif
- Langkah 4: Buat instans basis data RDS for MySQL tambahan untuk klaster aktif-aktif
- Langkah 5: Inisialisasikan grup pada instans basis data yang Anda konversi
- Langkah 6: Mulai replikasi pada instans basis data lain di klaster aktif-aktif
- Langkah 7: (Disarankan) Periksa status klaster aktif-aktif

Langkah 1: Atur parameter klaster aktif-aktif dalam satu atau beberapa grup parameter kustom

Instans basis data RDS for MySQL dalam klaster aktif-aktif harus dikaitkan dengan grup parameter kustom yang memiliki setelan yang benar untuk parameter-parameter yang diperlukan. Lihat informasi tentang parameter-parameter dan setelan yang diperlukan untuk masing-masing parameter di Setelan parameter yang diperlukan untuk klaster aktif-aktif.

Anda dapat mengatur semua parameter ini di grup parameter baru atau di grup parameter yang ada. Namun, untuk menghindari mempengaruhi tanpa sengaja instans basis data yang bukan bagian dari klaster aktif-aktif, kami sangat menyarankan supaya Anda membuat grup parameter kustom baru. Instans basis data dalam klaster aktif-aktif dapat dikaitkan dengan grup parameter basis data yang sama atau berbeda.

Anda dapat menggunakan AWS Management Console atau AWS CLI untuk membuat grup parameter kustom baru. Untuk informasi selengkapnya, lihat Membuat grup parameter DB. Contoh berikut menjalankan create-db-parameter-groupAWS CLIperintah untuk membuat grup parameter DB kustom bernama*myactivepg*:

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-parameter-group \
  --db-parameter-group-name myactivepg \
  --db-parameter-group-family mysql8.0 \
  --description "Parameter group for active-active clusters"
```

Untuk Windows:

```
aws rds create-db-parameter-group ^
  --db-parameter-group-name myactivepg ^
  --db-parameter-group-family mysql8.0 ^
  --description "Parameter group for active-active clusters"
```

Anda juga dapat menggunakan AWS Management Console atau AWS CLI untuk mengatur parameter dalam grup parameter kustom. Untuk informasi selengkapnya, lihat Memodifikasi parameter dalam grup parameter DB.

Contoh berikut menjalankan modify-db-parameter-groupAWS CLIperintah untuk mengatur parameter:

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myactivepg \
  --parameters
 "ParameterName='rds.group_replication_enabled',ParameterValue='1',ApplyMethod=pending-
reboot" \

 "ParameterName='rds.custom_dns_resolution',ParameterValue='1',ApplyMethod=pending-
reboot" \

 "ParameterName='enforce_gtid_consistency',ParameterValue='ON',ApplyMethod=pending-
reboot" \
              "ParameterName='gtid-mode',ParameterValue='ON',ApplyMethod=pending-
reboot" \

 "ParameterName='binlog_format',ParameterValue='ROW',ApplyMethod=immediate" \

 "ParameterName='slave_preserve_commit_order',ParameterValue='ON',ApplyMethod=immediate"
 \

 "ParameterName='group_replication_group_name',ParameterValue='11111111-2222-3333-4444-55555555
reboot"
```

Untuk Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myactivepg ^
  --parameters
 "ParameterName='rds.group_replication_enabled',ParameterValue='1',ApplyMethod=pending-
reboot" ^

 "ParameterName='rds.custom_dns_resolution',ParameterValue='1',ApplyMethod=pending-
reboot" ^

 "ParameterName='enforce_gtid_consistency',ParameterValue='ON',ApplyMethod=pending-
reboot" ^
              "ParameterName='gtid-mode',ParameterValue='ON',ApplyMethod=pending-
reboot" ^

 "ParameterName='binlog_format',ParameterValue='ROW',ApplyMethod=immediate" ^

 "ParameterName='slave_preserve_commit_order',ParameterValue='ON',ApplyMethod=immediate"
 ^
```

```
 "ParameterName='group_replication_group_name',ParameterValue='11111111-2222-3333-4444-55555555
reboot"
```

Langkah 2: Kaitkan instans basis data dengan grup parameter basis data yang telah mengatur parameter Group Replication yang diperlukan

Kaitkan instans basis data dengan grup parameter yang Anda buat atau ubah dalam langkah sebelumnya. Lihat petunjuk di Mengaitkan grup parameter DB dengan instans DB.

But ulang instans basis data agar setelan parameter baru berlaku. Lihat petunjuk di Mem-boot ulang instans DB.

Langkah 3: Buat klaster aktif-aktif

Dalam grup parameter basis data yang terkait dengan instans basis data, atur parameter `group_replication_group_seeds` ke titik akhir instans basis data yang Anda konversi.

Anda dapat menggunakan AWS Management Console atau AWS CLI untuk mengatur parameter. Anda tidak perlu membut ulang instans basis data setelah mengatur parameter ini. Lihat informasi yang lebih lengkap tentang pengaturan parameter di Memodifikasi parameter dalam grup parameter DB.

Contoh berikut menjalankan modify-db-parameter-groupAWS CLIperintah untuk mengatur parameter:

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myactivepg \
  --parameters
 "ParameterName='group_replication_group_seeds',ParameterValue='myactivedb1.123456789012.us-
east-1.rds.amazonaws.com:3306',ApplyMethod=immediate"
```

Untuk Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myactivepg ^
  --parameters
 "ParameterName='group_replication_group_seeds',ParameterValue='myactivedb1.123456789012.us-
east-1.rds.amazonaws.com:3306',ApplyMethod=immediate"
```

Langkah 4: Buat instans basis data RDS for MySQL tambahan untuk klaster aktif-aktif

Untuk membuat instance DB tambahan untuk cluster aktif-aktif, lakukan point-in-time pemulihan pada instans DB yang Anda konversi. Lihat petunjuk di Menambahkan instans DB ke cluster aktif-aktif menggunakan pemulihan point-in-time .

Klaster aktif-aktif dapat memiliki hingga sembilan instans basis data. Lakukan point-in-time pemulihan pada instans DB hingga Anda memiliki jumlah instans DB yang Anda inginkan untuk cluster. Saat Anda melakukan point-in-recovery, pastikan Anda mengaitkan instans DB yang Anda tambahkan dengan grup parameter DB yang telah `rds.group_replication_enabled` disetel1. Jika tidak, Group Replication tidak akan dimulai pada instans basis data yang baru ditambahkan.

Langkah 5: Inisialisasikan grup pada instans basis data yang Anda konversi

Inisialisasikan grup dan mulai replikasi:

1. Hubungi instans basis data yang Anda konversi dalam klien SQL. Lihat informasi yang lebih lengkap tentang cara menghubungi instans basis data RDS for MySQL di Menghubungkan ke instans DB yang menjalankan mesin basis data MySQL.
2. Di klien SQL, jalankan prosedur tersimpan berikut dan ganti *`group_replication_user_password`* dengan kata sandi untuk pengguna `rdsgrprepladmin`. Pengguna `rdsgrprepladmin` dicadangkan untuk koneksi Group Replication dalam klaster aktif-aktif. Kata sandi untuk pengguna ini harus sama pada semua instans basis data di klaster aktif-aktif.

```
call mysql.rds_set_configuration('binlog retention hours', 168); -- 7 days binlog
call mysql.rds_group_replication_create_user('group_replication_user_password');
call
 mysql.rds_group_replication_set_recovery_channel('group_replication_user_password');
call mysql.rds_group_replication_start(1);
```

   Contoh ini menetapkan nilai `binlog retention hours` ke 168, yang berarti bahwa file log biner dipertahankan selama tujuh hari pada instans basis data. Anda dapat menyesuaikan nilai ini untuk memenuhi kebutuhan Anda.

   Contoh-contoh ini menentukan 1 dalam prosedur tersimpan `mysql.rds_group_replication_start` untuk menginisialisasikan grup baru dengan instans basis data saat ini.

Lihat informasi yang lebih lengkap tentang prosedur tersimpan yang dipanggil dalam contoh ini di Mengelola klaster aktif-aktif.

Langkah 6: Mulai replikasi pada instans basis data lain di klaster aktif-aktif

Untuk setiap instans basis data di klaster aktif-aktif, gunakan klien SQL untuk menghubungi instans itu, dan jalankan prosedur tersimpan berikut. Ganti *group_replication_user_password* dengan kata sandi untuk pengguna `rdsgrprepladmin`.

```
call mysql.rds_set_configuration('binlog retention hours', 168); -- 7 days binlog
call mysql.rds_group_replication_create_user('group_replication_user_password');
call
 mysql.rds_group_replication_set_recovery_channel('group_replication_user_password');
call mysql.rds_group_replication_start(0);
```

Contoh ini menetapkan nilai `binlog retention hours` ke `168`, yang berarti bahwa file log biner dipertahankan selama tujuh hari pada setiap instans basis data. Anda dapat menyesuaikan nilai ini untuk memenuhi kebutuhan Anda.

Contoh ini menentukan `0` dalam prosedur tersimpan `mysql.rds_group_replication_start` untuk menggabungkan instans basis data saat ini dengan grup yang ada.

> **Tip**
>
> Pastikan untuk menjalankan prosedur tersimpan ini pada semua instans basis data yang lain di klaster aktif-aktif.

Langkah 7: (Disarankan) Periksa status klaster aktif-aktif

Untuk memastikan bahwa setiap anggota klaster dikonfigurasikan dengan benar, periksa status klaster dengan menghubungi instans basis data di klaster aktif-aktif, dan menjalankan perintah SQL berikut:

```
SELECT * FROM performance_schema.replication_group_members;
```

Output Anda semestinya menampilkan `ONLINE` untuk `MEMBER_STATE` setiap instans basis data, seperti pada output sampel berikut:

```
+---------------------------+-------------------------------------
+---------------+-------------+--------------+-------------+---------------
+----------------------------+
| CHANNEL_NAME              | MEMBER_ID                           | MEMBER_HOST    |
 MEMBER_PORT | MEMBER_STATE | MEMBER_ROLE | MEMBER_VERSION | MEMBER_COMMUNICATION_STACK
 |
+---------------------------+-------------------------------------
+---------------+-------------+--------------+-------------+---------------
+----------------------------+
| group_replication_applier | 9854d4a2-5d7f-11ee-b8ec-0ec88c43c251 | ip-10-15-3-137 |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
| group_replication_applier | 9e2e9c28-5d7f-11ee-8039-0e5d58f05fef | ip-10-15-3-225 |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
| group_replication_applier | a6ba332d-5d7f-11ee-a025-0a5c6971197d | ip-10-15-1-83  |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
+---------------------------+-------------------------------------
+---------------+-------------+--------------+-------------+---------------
+----------------------------+
3 rows in set (0.00 sec)
```

Lihat informasi tentang nilai `MEMBER_STATE` yang mungkin di Status Server Replikasi Grup dalam dokumentasi MySQL.

Menyiapkan klaster aktif-aktif dengan instans basis data baru

Selesaikan langkah-langkah berikut untuk menyiapkan klaster aktif-aktif dengan menggunakan instans basis data RDS for MySQL baru.

Jika Anda menyiapkan klaster aktif-aktif dengan instans basis data di lebih dari satu VPC, pastikan untuk menyelesaikan prasyarat di Prasyarat untuk klaster aktif-aktif lintas VPC.

Topik

- Langkah 1: Atur parameter klaster aktif-aktif dalam satu atau beberapa grup parameter kustom
- Langkah 2: Buat instans basis data RDS for MySQL baru untuk klaster aktif-aktif
- Langkah 4: Tentukan instans basis data di klaster aktif-aktif
- Langkah 5: Inisialisasikan grup pada instans basis data dan mulai replikasi
- Langkah 6: Mulai replikasi pada instans basis data lain di klaster aktif-aktif
- Langkah 7: (Disarankan) Periksa status klaster aktif-aktif
- Langkah 8: (Opsional) Impor data ke instans basis data di klaster aktif-aktif

Langkah 1: Atur parameter klaster aktif-aktif dalam satu atau beberapa grup parameter kustom

Instans basis data RDS for MySQL dalam klaster aktif-aktif harus dikaitkan dengan grup parameter kustom yang memiliki setelan yang benar untuk parameter-parameter yang diperlukan. Lihat informasi tentang parameter-parameter dan setelan yang diperlukan untuk masing-masing parameter di Setelan parameter yang diperlukan untuk klaster aktif-aktif.

Anda dapat mengatur semua parameter ini di grup parameter baru atau di grup parameter yang ada. Namun, untuk menghindari mempengaruhi tanpa sengaja instans basis data yang bukan bagian dari klaster aktif-aktif, kami sangat menyarankan supaya Anda membuat grup parameter kustom baru. Instans basis data dalam klaster aktif-aktif dapat dikaitkan dengan grup parameter basis data yang sama atau berbeda.

Anda dapat menggunakan AWS Management Console atau AWS CLI untuk membuat grup parameter kustom baru. Untuk informasi selengkapnya, lihat Membuat grup parameter DB. Contoh berikut menjalankan create-db-parameter-groupAWS CLIperintah untuk membuat grup parameter DB kustom bernama*myactivepg*:

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-parameter-group \
  --db-parameter-group-name myactivepg \
  --db-parameter-group-family mysql8.0 \
  --description "Parameter group for active-active clusters"
```

Untuk Windows:

```
aws rds create-db-parameter-group ^
  --db-parameter-group-name myactivepg ^
  --db-parameter-group-family mysql8.0 ^
  --description "Parameter group for active-active clusters"
```

Anda juga dapat menggunakan AWS Management Console atau AWS CLI untuk mengatur parameter dalam grup parameter kustom. Untuk informasi selengkapnya, lihat Memodifikasi parameter dalam grup parameter DB.

Contoh berikut menjalankan modify-db-parameter-groupAWS CLIperintah untuk mengatur parameter:

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myactivepg \
  --parameters
 "ParameterName='rds.group_replication_enabled',ParameterValue='1',ApplyMethod=pending-
reboot" \

 "ParameterName='rds.custom_dns_resolution',ParameterValue='1',ApplyMethod=pending-
reboot" \

 "ParameterName='enforce_gtid_consistency',ParameterValue='ON',ApplyMethod=pending-
reboot" \
              "ParameterName='gtid-mode',ParameterValue='ON',ApplyMethod=pending-
reboot" \

 "ParameterName='binlog_format',ParameterValue='ROW',ApplyMethod=immediate" \

 "ParameterName='slave_preserve_commit_order',ParameterValue='ON',ApplyMethod=immediate"
 \

 "ParameterName='group_replication_group_name',ParameterValue='11111111-2222-3333-4444-55555555
reboot"
```

Untuk Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myactivepg ^
  --parameters
 "ParameterName='rds.group_replication_enabled',ParameterValue='1',ApplyMethod=pending-
reboot" ^

 "ParameterName='rds.custom_dns_resolution',ParameterValue='1',ApplyMethod=pending-
reboot" ^

 "ParameterName='enforce_gtid_consistency',ParameterValue='ON',ApplyMethod=pending-
reboot" ^
              "ParameterName='gtid-mode',ParameterValue='ON',ApplyMethod=pending-
reboot" ^

 "ParameterName='binlog_format',ParameterValue='ROW',ApplyMethod=immediate" ^

 "ParameterName='slave_preserve_commit_order',ParameterValue='ON',ApplyMethod=immediate"
 ^
```

```
 "ParameterName='group_replication_group_name',ParameterValue='11111111-2222-3333-4444-55555555
reboot"
```

Langkah 2: Buat instans basis data RDS for MySQL baru untuk klaster aktif-aktif

Klaster aktif-aktif didukung untuk instans basis data RDS for MySQL versi 8.0.35 dan yang lebih tinggi. Anda dapat membuat hingga sembilan instans basis data baru untuk klaster.

Anda dapat menggunakan AWS Management Console atau AWS CLI untuk membuat instans basis data baru. Lihat informasi yang lebih lengkap tentang cara membuat instans basis data di Membuat instans DB Amazon RDS. Saat Anda membuat instans basis data, kaitkan dengan grup parameter basis data yang Anda buat atau ubah dalam langkah sebelumnya.

Langkah 4: Tentukan instans basis data di klaster aktif-aktif

Dalam grup parameter basis data yang terkait dengan setiap instans basis data, atur parameter `group_replication_group_seeds` ke titik akhir instans basis data yang ingin Anda sertakan dalam klaster.

Anda dapat menggunakan AWS Management Console atau AWS CLI untuk mengatur parameter. Anda tidak perlu membut ulang instans basis data setelah mengatur parameter ini. Lihat informasi yang lebih lengkap tentang pengaturan parameter di Memodifikasi parameter dalam grup parameter DB.

Contoh berikut menjalankan modify-db-parameter-groupAWS CLIperintah untuk mengatur parameter:

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myactivepg \
  --parameters
 "ParameterName='group_replication_group_seeds',ParameterValue='myactivedb1.123456789012.us-
east-1.rds.amazonaws.com:3306,myactivedb2.123456789012.us-
east-1.rds.amazonaws.com:3306,myactivedb3.123456789012.us-
east-1.rds.amazonaws.com:3306',ApplyMethod=immediate"
```

Untuk Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myactivepg ^
```

```
  --parameters
 "ParameterName='group_replication_group_seeds',ParameterValue='myactivedb1.123456789012.us-
east-1.rds.amazonaws.com:3306,myactivedb2.123456789012.us-
east-1.rds.amazonaws.com:3306,myactivedb3.123456789012.us-
east-1.rds.amazonaws.com:3306',ApplyMethod=immediate"
```

> **Tip**
>
> Pastikan untuk mengatur parameter `group_replication_group_seeds` di setiap grup parameter basis data yang terkait dengan instans basis data di klaster aktif-aktif.

Langkah 5: Inisialisasikan grup pada instans basis data dan mulai replikasi

Anda dapat memilih basis data baru untuk menginisialisasikan grup dan memulai replikasi. Untuk melakukannya, selesaikan langkah-langkah berikut:

1. Pilih instans basis data di klaster aktif-aktif, dan hubungi instans basis data itu di klien SQL. Lihat informasi yang lebih lengkap tentang cara menghubungi instans basis data RDS for MySQL di Menghubungkan ke instans DB yang menjalankan mesin basis data MySQL.
2. Di klien SQL, jalankan prosedur tersimpan berikut dan ganti *group_replication_user_password* dengan kata sandi untuk pengguna `rdsgrprepladmin`. Pengguna `rdsgrprepladmin` dicadangkan untuk koneksi Group Replication dalam klaster aktif-aktif. Kata sandi untuk pengguna ini harus sama pada semua instans basis data di klaster aktif-aktif.

   ```
   call mysql.rds_set_configuration('binlog retention hours', 168); -- 7 days binlog
   call mysql.rds_group_replication_create_user('group_replication_user_password');
   call
    mysql.rds_group_replication_set_recovery_channel('group_replication_user_password');
   call mysql.rds_group_replication_start(1);
   ```

   Contoh ini menetapkan nilai `binlog retention hours` ke 168, yang berarti bahwa file log biner dipertahankan selama tujuh hari pada instans basis data. Anda dapat menyesuaikan nilai ini untuk memenuhi kebutuhan Anda.

   Contoh-contoh ini menentukan 1 dalam prosedur tersimpan `mysql.rds_group_replication_start` untuk menginisialisasikan grup baru dengan instans basis data saat ini.

Lihat informasi yang lebih lengkap tentang prosedur tersimpan yang dipanggil dalam contoh ini di Mengelola klaster aktif-aktif.

Langkah 6: Mulai replikasi pada instans basis data lain di klaster aktif-aktif

Untuk setiap instans basis data di klaster aktif-aktif, gunakan klien SQL untuk menghubungi instans itu, dan jalankan prosedur tersimpan berikut. Ganti *group_replication_user_password* dengan kata sandi untuk pengguna `rdsgrprepladmin`.

```
call mysql.rds_set_configuration('binlog retention hours', 168); -- 7 days binlog
call mysql.rds_group_replication_create_user('group_replication_user_password');
call
 mysql.rds_group_replication_set_recovery_channel('group_replication_user_password');
call mysql.rds_group_replication_start(0);
```

Contoh ini menetapkan nilai `binlog retention hours` ke `168`, yang berarti bahwa file log biner dipertahankan selama tujuh hari pada setiap instans basis data. Anda dapat menyesuaikan nilai ini untuk memenuhi kebutuhan Anda.

Contoh ini menentukan `0` dalam prosedur tersimpan `mysql.rds_group_replication_start` untuk menggabungkan instans basis data saat ini dengan grup yang ada.

> **Tip**
>
> Pastikan untuk menjalankan prosedur tersimpan ini pada semua instans basis data yang lain di klaster aktif-aktif.

Langkah 7: (Disarankan) Periksa status klaster aktif-aktif

Untuk memastikan bahwa setiap anggota klaster dikonfigurasikan dengan benar, periksa status klaster dengan menghubungi instans basis data di klaster aktif-aktif, dan menjalankan perintah SQL berikut:

```
SELECT * FROM performance_schema.replication_group_members;
```

Output Anda semestinya menampilkan `ONLINE` untuk `MEMBER_STATE` setiap instans basis data, seperti pada output sampel berikut:

```
+---------------------------+-------------------------------------
+---------------+-------------+--------------+-------------+---------------
+----------------------------+
| CHANNEL_NAME               | MEMBER_ID                           | MEMBER_HOST    |
 MEMBER_PORT | MEMBER_STATE | MEMBER_ROLE | MEMBER_VERSION | MEMBER_COMMUNICATION_STACK
 |
+---------------------------+-------------------------------------
+---------------+-------------+--------------+-------------+---------------
+----------------------------+
| group_replication_applier | 9854d4a2-5d7f-11ee-b8ec-0ec88c43c251 | ip-10-15-3-137 |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
| group_replication_applier | 9e2e9c28-5d7f-11ee-8039-0e5d58f05fef | ip-10-15-3-225 |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
| group_replication_applier | a6ba332d-5d7f-11ee-a025-0a5c6971197d | ip-10-15-1-83  |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
+---------------------------+-------------------------------------
+---------------+-------------+--------------+-------------+---------------
+----------------------------+
3 rows in set (0.00 sec)
```

Lihat informasi tentang nilai `MEMBER_STATE` yang mungkin di Status Server Replikasi Grup dalam dokumentasi MySQL.

Langkah 8: (Opsional) Impor data ke instans basis data di klaster aktif-aktif

Anda dapat mengimpor data dari basis data MySQL ke instans basis data di klaster aktif-aktif. Setelah data diimpor, Group Replication mereplikasi data ke instans basis data lain di klaster.

Lihat informasi tentang cara mengimpor data di Mengimpor data ke basis data Amazon RDS MariaDB atau MySQL dengan lebih sedikit waktu henti.

Menambahkan instans basis data ke klaster aktif-aktif

Anda dapat menambahkan instans basis data ke klaster aktif-aktif dengan memulihkan cuplikan basis data atau dengan memulihkan instans basis data ke suatu titik waktu. Klaster aktif-aktif dapat mencakup hingga sembilan instans basis data.

Ketika Anda memulihkan instans basis data ke suatu titik waktu, pemulihan itu biasanya mencakup transaksi yang lebih baru daripada yang ada di instans basis data yang dipulihkan dari cuplikan basis data. Apabila instans basis data memiliki transaksi yang lebih baru, lebih sedikit transaksi yang

perlu diterapkan saat Anda memulai replikasi. Jadi, menggunakan point-in-time pemulihan untuk menambahkan instance DB ke cluster biasanya lebih cepat daripada memulihkan dari snapshot DB.

Topik

- Menambahkan instans DB ke cluster aktif-aktif menggunakan pemulihan point-in-time
- Menambahkan instans basis data ke klaster aktif-aktif dengan menggunakan cuplikan basis data

Menambahkan instans DB ke cluster aktif-aktif menggunakan pemulihan point-in-time

Anda dapat menambahkan instans DB ke cluster aktif-aktif dengan melakukan point-in-time pemulihan pada instans DB di cluster.

Lihat informasi tentang pemulihan instans basis data ke titik waktu dalam Wilayah AWS yang berbeda di Mereplikasi backup otomatis ke yang lain Wilayah AWS.

Untuk menambahkan instans DB ke cluster aktif-aktif menggunakan pemulihan point-in-time

1. Buat instans DB baru dengan melakukan point-in-time pemulihan pada instans DB di cluster aktif-aktif.

   Anda dapat melakukan point-in-time pemulihan pada instans DB apa pun di cluster untuk membuat instans DB baru. Untuk petunjuk, lihat Memulihkan instans DB dengan waktu yang ditentukan.

   > **Important**
   >
   > Selama point-in-time-recovery, kaitkan instans DB baru dengan grup parameter DB yang memiliki parameter cluster aktif-aktif yang ditetapkan. Jika tidak, Group Replication tidak akan mulai pada instans basis data baru. Lihat informasi tentang parameter-parameter dan setelan yang diperlukan untuk masing-masing parameter di Setelan parameter yang diperlukan untuk klaster aktif-aktif.

   > **Tip**
   >
   > Jika Anda mengambil snapshot dari instans DB sebelum memulai point-in-time pemulihan, Anda mungkin dapat mengurangi jumlah waktu yang diperlukan untuk menerapkan transaksi pada instans DB baru.

2. Tambahkan instans basis data ke parameter `group_replication_group_seeds` di setiap grup parameter basis data yang terkait dengan instans basis data di klaster aktif-aktif, yang meliputi grup parameter basis data yang Anda kaitkan dengan instans basis data baru.

   Lihat informasi yang lebih lengkap tentang pengaturan parameter di Memodifikasi parameter dalam grup parameter DB.

3. Dalam klien SQL, hubungi instans basis data baru, dan panggil prosedur tersimpan mysql.rds_group_replication_set_recovery_channel. Ganti *group_replication_user_password* dengan kata sandi untuk pengguna `rdsgrprepladmin`.

   ```
   call
    mysql.rds_group_replication_set_recovery_channel('group_replication_user_password');
   ```

4. Dengan menggunakan klien SQL, panggil prosedur tersimpan mysql.rds_group_replication_start untuk memulai replikasi:

   ```
   call mysql.rds_group_replication_start(0);
   ```

Menambahkan instans basis data ke klaster aktif-aktif dengan menggunakan cuplikan basis data

Anda dapat menambahkan instans basis data ke klaster aktif-aktif dengan membuat cuplikan basis data sebuah instans basis data di klaster, lalu memulihkan cuplikan itu.

Lihat informasi tentang cara menyalin cuplikan ke Wilayah AWS lain di the section called “Penyalinan lintas Wilayah”.

Untuk menambahkan instans basis data ke klaster aktif-aktif dengan menggunakan cuplikan basis data

1. Buat cuplikan basis data sebuah instans basis data di klaster aktif-aktif.

   Anda dapat membuat cuplikan basis data sebarang instans basis data di klaster. Lihat petunjuk di Membuat snapshot DB untuk instans DB Single-AZ.

2. Pulihkan instans basis data dari cuplikan basis data.

Selama operasi pemulihan cuplikan, kaitkan instans basis data baru dengan grup parameter basis data yang telah mengatur parameter klaster aktif-aktif. Lihat informasi tentang parameter-parameter dan setelan yang diperlukan untuk masing-masing parameter di Setelan parameter yang diperlukan untuk klaster aktif-aktif.

Lihat informasi tentang cara memulihkan instans basis data dari cuplikan basis data di Memulihkan dari snapshot DB.

3. Tambahkan instans basis data ke parameter `group_replication_group_seeds` di setiap grup parameter basis data yang terkait dengan instans basis data di klaster aktif-aktif, yang meliputi grup parameter basis data yang Anda kaitkan dengan instans basis data baru.

   Lihat informasi yang lebih lengkap tentang pengaturan parameter di Memodifikasi parameter dalam grup parameter DB.

4. Dalam klien SQL, hubungi instans basis data baru, dan panggil prosedur tersimpan mysql.rds_group_replication_set_recovery_channel. Ganti *`group_replication_user_password`* dengan kata sandi untuk pengguna `rdsgrprepladmin`.

   ```
   call
    mysql.rds_group_replication_set_recovery_channel('group_replication_user_password');
   ```

5. Dengan menggunakan klien SQL, panggil prosedur tersimpan mysql.rds_group_replication_start untuk memulai replikasi:

   ```
   call mysql.rds_group_replication_start(0);
   ```

Memantau klaster aktif-aktif

Anda dapat memantau klaster aktif-aktif dengan menghubungi instans basis data di klaster dan menjalankan perintah SQL berikut:

```
SELECT * FROM performance_schema.replication_group_members;
```

Output Anda semestinya menampilkan `ONLINE` untuk `MEMBER_STATE` setiap instans basis data, seperti pada output sampel berikut:

```
+---------------------------+-------------------------------------
+---------------+-------------+--------------+-------------+---------------
+----------------------------+
| CHANNEL_NAME               | MEMBER_ID                           | MEMBER_HOST    |
 MEMBER_PORT | MEMBER_STATE | MEMBER_ROLE | MEMBER_VERSION | MEMBER_COMMUNICATION_STACK
 |
+---------------------------+-------------------------------------
+---------------+-------------+--------------+-------------+---------------
+----------------------------+
| group_replication_applier | 9854d4a2-5d7f-11ee-b8ec-0ec88c43c251 | ip-10-15-3-137 |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
| group_replication_applier | 9e2e9c28-5d7f-11ee-8039-0e5d58f05fef | ip-10-15-3-225 |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
| group_replication_applier | a6ba332d-5d7f-11ee-a025-0a5c6971197d | ip-10-15-1-83  |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
+---------------------------+-------------------------------------
+---------------+-------------+--------------+-------------+---------------
+----------------------------+
3 rows in set (0.00 sec)
```

Lihat informasi tentang nilai `MEMBER_STATE` yang mungkin di Status Server Replikasi Grup dalam dokumentasi MySQL.

Menghentikan Group Replication pada instans basis data dalam klaster aktif-aktif

Anda dapat menghentikan Group Replication pada instans basis data di klaster aktif-aktif. Saat Anda menghentikan Replikasi Grup, instans DB ditempatkan dalam super-read-only mode hingga replikasi dimulai ulang atau instans DB dihapus dari cluster aktif-aktif. Untuk informasi tentang super-read-only mode, lihat dokumentasi MySQL.

Untuk menghentikan sementara Group Replication bagi klaster aktif-aktif

1. Hubungi instans basis data di klaster aktif-aktif dengan menggunakan klien SQL.

   Lihat informasi yang lebih lengkap tentang cara menghubungi instans basis data RDS for MySQL di Menghubungkan ke instans DB yang menjalankan mesin basis data MySQL.
2. Di klien SQL, panggil prosedur tersimpan mysql.rds_group_replication_stop:

   ```
   call mysql.rds_group_replication_stop();
   ```

Mengganti nama instans basis data di klaster aktif-aktif

Anda dapat mengganti nama instans basis data di klaster aktif-aktif. Untuk mengganti nama lebih dari satu instans basis data dalam klaster aktif-aktif, ganti nama instans basis data satu per satu. Jadi, ganti nama satu instans basis data dan gabungkan kembali dengan klaster sebelum Anda mengganti nama instans basis data berikutnya.

Untuk mengganti nama instans basis data di klaster aktif-aktif

1. Hubungi instans basis data di klien SQL, dan panggil prosedur tersimpan mysql.rds_group_replication_stop:

   ```
   call mysql.rds_group_replication_stop();
   ```

2. Ganti nama instans basis data dengan mengikuti petunjuk di Mengganti nama instans DB.
3. Ubah parameter `group_replication_group_seeds` di setiap grup parameter basis data yang terkait dengan instans basis data di klaster aktif-aktif.

   Pada setelan parameter, ganti titik akhir instans basis data lama dengan titik akhir instans basis data baru. Lihat informasi yang lebih lengkap tentang pengaturan parameter di Memodifikasi parameter dalam grup parameter DB.
4. Hubungi instans basis data di klien SQL, dan panggil prosedur tersimpan mysql.rds_group_replication_start:

   ```
   call mysql.rds_group_replication_start(0);
   ```

Mengeluarkan instans basis data dari klaster aktif-aktif

Saat Anda mengeluarkan suatu instans basis data dari klaster aktif-aktif, instans itu akan kembali menjadi instans basis data mandiri.

Untuk mengeluarkan instans basis data dari klaster aktif-aktif

1. Hubungi instans basis data di klien SQL, dan panggil prosedur tersimpan mysql.rds_group_replication_stop:

   ```
   call mysql.rds_group_replication_stop();
   ```

2. Ubah parameter `group_replication_group_seeds` untuk instans basis data yang akan tetap berada di klaster aktif-aktif.

   Dalam parameter `group_replication_group_seeds`, hapus instans basis data yang Anda keluarkan dari klaster aktif-aktif. Lihat informasi yang lebih lengkap tentang pengaturan parameter di Memodifikasi parameter dalam grup parameter DB.

3. Ubah parameter-parameter instans basis data yang Anda keluarkan dari klaster aktif-aktif sehingga tidak lagi menjadi bagian dari klaster.

   Anda dapat mengaitkan instans basis data dengan grup parameter yang berbeda, atau mengubah parameter-parameter dalam grup parameter basis data yang terkait dengan instans basis data. Parameter-parameter untuk diubah meliputi `group_replication_group_name`, `rds.group_replication_enabled`, dan `group_replication_group_seeds`. Lihat informasi yang lebih lengkap tentang parameter klaster aktif-aktif di Setelan parameter yang diperlukan untuk klaster aktif-aktif.

   Jika Anda mengubah parameter-parameter dalam grup parameter basis data, pastikan bahwa grup itu tidak terkait dengan instans basis data lain di klaster aktif-aktif.

4. But ulang instans basis data yang Anda keluarkan dari klaster aktif-aktif agar setelan parameter baru berlaku.

   Lihat petunjuk di Mem-boot ulang instans DB.

Keterbatasan untuk klaster aktif-aktif RDS for MySQL

Keterbatasan berikut berlaku bagi klaster aktif-aktif untuk RDS for MySQL:

- Nama pengguna master tidak boleh `rdsgrprepladmin` untuk instans basis data di klaster aktif-aktif. Nama pengguna ini dicadangkan untuk koneksi Group Replication.
- Untuk instans basis data dengan replika baca di klaster aktif-aktif, status replikasi yang berkepanjangan selain `Replicating` dapat menyebabkan file log melebihi batas penyimpanan. Lihat informasi tentang status replika baca di Memantau replikasi baca.
- Deployment blue/green tidak didukung untuk instans basis data di klaster aktif-aktif. Untuk informasi selengkapnya, lihat Menggunakan Deployment Blue/Green Amazon RDS untuk pembaruan basis data.
- Autentikasi Kerberos tidak didukung untuk instans basis data di klaster aktif-aktif. Untuk informasi selengkapnya, lihat Menggunakan autentikasi Kerberos untuk MySQL.

- Instans basis data di klaster basis data Multi-AZ tidak dapat ditambahkan ke klaster aktif-aktif.

  Namun, instans basis data dalam deployment instans basis data Multi-AZ dapat ditambahkan ke klaster aktif-aktif.

  Untuk informasi selengkapnya, lihat Mengonfigurasi dan mengelola deployment Multi-AZ.
- Tabel yang tidak memiliki kunci primer tidak direplikasi dalam klaster aktif-aktif karena penulisan ditolak oleh pengaya Group Replication.
- Tabel non-InnoDB tidak direplikasi dalam klaster aktif-aktif.
- Klaster aktif-aktif tidak mendukung pernyataan-pernyataan DHTML dan DDL yang konkuren pada instans basis data yang berbeda di klaster.
- Anda tidak dapat mengonfigurasikan klaster aktif-aktif untuk menggunakan mode primer tunggal untuk mode replikasi grup. Untuk konfigurasi ini, sebaiknya gunakan klaster basis data Multi-AZ sebagai gantinya. Untuk informasi selengkapnya, lihat Deployment klaster basis data Multi-AZ.
- Replikasi multisumber tidak didukung untuk instans basis data di klaster aktif-aktif.
- Klaster aktif-aktif lintas Kawasan tidak dapat menerapkan verifikasi otoritas sertifikat (CA) untuk koneksi Group Replication.

Mengekspor data dari instans DB MySQL dengan menggunakan replikasi

Untuk mengekspor data dari instans DB MySQL ke instans MySQL yang berjalan di luar Amazon RDS, Anda dapat menggunakan replikasi. Dalam skenario ini, instans DB MySQL adalah instans DB MySQL sumber, dan instans MySQL yang berjalan di luar Amazon RDS adalah basis data MySQL eksternal.

Basis data MySQL eksternal dapat berjalan secara on-premise di pusat data Anda, atau pada instans Amazon EC2. Basis data MySQL eksternal harus menjalankan versi yang sama dengan instans DB MySQL sumber, atau versi lebih baru.

Replikasi ke basis data MySQL eksternal hanya didukung selama waktu yang diperlukan untuk mengekspor basis data dari instans DB MySQL sumber. Replikasi harus diakhiri ketika data telah diekspor dan aplikasi dapat mulai mengakses instans MySQL eksternal.

Daftar berikut menunjukkan langkah-langkah yang harus diambil. Setiap langkah dibahas secara lebih mendetail di bagian berikutnya.

1. Siapkan instans DB MySQL eksternal.
2. Siapkan instans DB MySQL sumber untuk direplikasi.
3. Gunakan utilitas mysqldump untuk mentransfer basis data dari instans DB MySQL sumber ke basis data MySQL eksternal.
4. Mulai replikasi ke basis data MySQL eksternal.
5. Setelah selesai mengekspor, hentikan replikasi.

Menyiapkan basis data MySQL eksternal

Lakukan langkah-langkah berikut untuk menyiapkan basis data MySQL eksternal.

Menyiapkan basis data MySQL eksternal

1. Instal basis data MySQL eksternal.
2. Hubungkan ke basis data MySQL eksternal sebagai pengguna master. Kemudian buat pengguna yang diperlukan untuk mendukung administrator, aplikasi, dan layanan yang mengakses basis data.

3. Ikuti petunjuk di dokumentasi MySQL untuk menyiapkan basis data MySQL eksternal sebagai replika. Untuk informasi selengkapnya, lihat dokumentasi MySQL.
4. Lakukan konfigurasi aturan egress untuk basis data MySQL eksternal agar beroperasi sebagai replika baca selama pengeksporan. Aturan egress memungkinkan basis data MySQL eksternal untuk terhubung ke instans DB MySQL sumber selama replikasi. Tentukan aturan egress yang memungkinkan koneksi Transmission Control Protocol (TCP) ke port dan alamat IP instans DB MySQL sumber.

   Tentukan aturan egress yang sesuai untuk lingkungan Anda:

   - Jika basis data MySQL eksternal berjalan di instans Amazon EC2 di dalam cloud privat virtual (VPC) berdasarkan layanan Amazon VPC, tentukan aturan egress dalam grup keamanan VPC. Untuk informasi selengkapnya, lihat Mengontrol akses dengan grup keamanan.
   - Jika basis data MySQL eksternal diinstal secara on-premise, tentukan aturan egress di firewall.
5. Jika basis data MySQL eksternal berjalan dalam VPC, lakukan konfigurasi aturan untuk aturan daftar kontrol akses (ACL) VPC selain aturan egress grup keamanan:

   - Lakukan konfigurasi aturan ingress ACL yang memungkinkan lalu lintas TCP ke port 1024–65535 dari alamat IP instans DB MySQL sumber.
   - Lakukan konfigurasi aturan ingress ACL yang memungkinkan lalu lintas TCP keluar ke port dan alamat IP instans DB MySQL sumber.

   Untuk informasi selengkapnya tentang ACL jaringan Amazon VPC, lihat ACL Jaringan di Panduan Pengguna Amazon VPC.
6. (Opsional) Tetapkan parameter `max_allowed_packet` ke ukuran maksimum untuk menghindari kesalahan replikasi. Kami merekomendasikan pengaturan ini.

Siapkan instans DB MySQL sumber

Lakukan langkah-langkah berikut untuk menyiapkan instans DB MySQL sumber sebagai sumber replikasi.

Menyiapkan instans DB MySQL sumber

1. Pastikan komputer klien Anda memiliki ruang disk yang cukup untuk menyimpan log biner saat menyiapkan replikasi.

2. Hubungkan ke instans DB MySQL sumber, dan buat akun replikasi sesuai petunjuk di Membuat pengguna untuk replikasi dalam dokumentasi MySQL.
3. Lakukan konfigurasi aturan ingress pada sistem yang menjalankan instans DB MySQL sumber agar basis data MySQL eksternal dapat terhubung selama replikasi. Tentukan aturan ingress yang mengizinkan koneksi TCP ke port yang digunakan oleh instans DB MySQL sumber dari alamat IP basis data MySQL eksternal.
4. Tentukan aturan egress:

   - Jika instans DB MySQL sumber berjalan di VPC, tentukan aturan ingress dalam grup keamanan VPC. Untuk informasi selengkapnya, lihat Mengontrol akses dengan grup keamanan.
5. Jika instans DB MySQL sumber berjalan di VPC, lakukan konfigurasi aturan ACL VPC, selain aturan ingress grup keamanan:

   - Lakukan konfigurasi aturan ingress ACL untuk mengizinkan koneksi TCP ke port yang digunakan oleh instans Amazon RDS dari alamat IP basis data MySQL eksternal.
   - Lakukan konfigurasi aturan egress ACL untuk mengizinkan koneksi TCP dari port 1024–65535 ke alamat IP basis data MySQL eksternal.

   Untuk informasi selengkapnya tentang ACL jaringan Amazon VPC, lihat ACL Jaringan di Panduan Pengguna Amazon VPC.
6. Pastikan periode retensi cadangan ditetapkan cukup lama sehingga tidak ada log biner yang dihapus selama ekspor. Jika terdapat log yang dihapus sebelum ekspor selesai, Anda harus memulai ulang replikasi dari awal. Untuk informasi selengkapnya tentang cara mengatur periode retensi cadangan, lihat Pengantar cadangan.
7. Gunakan prosedur tersimpan `mysql.rds_set_configuration` untuk menetapkan periode penyimpanan log biner dalam waktu yang cukup lama sehingga log biner tidak dihapus selama ekspor. Untuk informasi selengkapnya, lihat Mengakses log biner MySQL.
8. Buat replika baca Amazon RDS dari instans DB MySQL sumber untuk lebih memastikan bahwa log biner instans DB MySQL sumber tidak dibersihkan. Untuk informasi selengkapnya, lihat Membuat replika baca.
9. Setelah replika baca Amazon RDS dibuat, panggil prosedur tersimpan `mysql.rds_stop_replication` untuk menghentikan proses replikasi. Instans DB MySQL sumber sudah tidak membersihkan file log biner, sehingga file tersedia untuk proses replikasi.

10. (Opsional) Tetapkan parameter `max_allowed_packet` dan parameter `slave_max_allowed_packet` ke ukuran maksimum untuk menghindari kesalahan replikasi. Ukuran maksimum untuk kedua parameter tersebut adalah 1 GB. Kami merekomendasikan pengaturan ini untuk kedua parameter. Untuk informasi tentang mengatur parameter, lihat Memodifikasi parameter dalam grup parameter DB.

Menyalin basis data

Lakukan langkah-langkah berikut untuk menyalin basis data.

Untuk menyalin basis data

1. Hubungkan ke replika baca RDS dari instans DB MySQL sumber, dan jalankan pernyataan `SHOW REPLICA STATUS\G` MySQL. Catat nilai untuk hal berikut:

   - `Master_Host`
   - `Master_Port`
   - `Master_Log_File`
   - `Exec_Master_Log_Pos`

   > **Note**
   >
   > Versi sebelumnya dari MySQL menggunakan `SHOW SLAVE STATUS` bukan `SHOW REPLICA STATUS`. Jika Anda menggunakan MySQL versi sebelum 8.0.23, gunakan `SHOW SLAVE STATUS`.

2. Gunakan utilitas mysqldump untuk membuat snapshot, yang menyalin data dari Amazon RDS ke komputer klien lokal Anda. Pastikan komputer klien Anda memiliki cukup ruang untuk menyimpan file `mysqldump` dari basis data yang akan direplikasi. Untuk basis data yang sangat besar, proses ini dapat memakan waktu beberapa jam. Ikuti petunjuk di Membuat snapshot data menggunakan mysqldump di dokumentasi MySQL.

   Contoh berikut menjalankan `mysqldump` pada klien dan menuliskan dump ke sebuah file.

   Untuk Linux, macOS, atau Unix:

   ```
   mysqldump -h source_MySQL_DB_instance_endpoint \
   ```

```
    -u user \
    -ppassword \
    --port=3306 \
    --single-transaction \
    --routines \
    --triggers \
    --databases  database database2 > path/rds-dump.sql
```

Untuk Windows:

```
mysqldump -h source_MySQL_DB_instance_endpoint ^
    -u user ^
    -ppassword ^
    --port=3306 ^
    --single-transaction ^
    --routines ^
    --triggers ^
    --databases  database database2 > path\rds-dump.sql
```

Anda dapat memuat file cadangan ke dalam basis data MySQL eksternal. Untuk informasi selengkapnya, lihat Memuat Ulang Cadangan Format SQL di dokumentasi MySQL. Anda dapat menjalankan utilitas lain untuk memuat data ke dalam basis data MySQL eksternal.

Menyelesaikan ekspor

Lakukan langkah-langkah berikut untuk menyelesaikan ekspor.

Untuk menyelesaikan ekspor

1. Gunakan pernyataan `CHANGE MASTER` MySQL untuk mengonfigurasi basis data MySQL eksternal. Tentukan ID dan kata sandi pengguna yang diberi izin `REPLICATION SLAVE`. Tentukan nilai `Master_Host`, `Master_Port`, `Relay_Master_Log_File`, dan `Exec_Master_Log_Pos` yang Anda dapatkan dari pernyataan `SHOW REPLICA STATUS \G` MySQL yang Anda jalankan pada replika baca RDS. Untuk informasi selengkapnya, lihat dokumentasi MySQL.

> **Note**
>
> Versi sebelumnya dari MySQL menggunakan `SHOW SLAVE STATUS` bukan `SHOW REPLICA STATUS`. Jika Anda menggunakan MySQL versi sebelum 8.0.23, gunakan `SHOW SLAVE STATUS`.

2. Gunakan perintah `START REPLICA` MySQL untuk memulai replikasi dari instans DB MySQL sumber ke basis data MySQL eksternal.

   Tindakan ini akan memulai replikasi dari instans DB MySQL sumber dan mengekspor semua perubahan sumber yang telah terjadi setelah Anda menghentikan replikasi dari replika baca Amazon RDS.

   > **Note**
   >
   > Versi MySQL sebelumnya menggunakan `START SLAVE`, bukan `START REPLICA`. Jika Anda menggunakan versi MySQL sebelum 8.0.23, gunakan `START SLAVE`.

3. Jalankan perintah `SHOW REPLICA STATUS\G` MySQL pada basis data MySQL eksternal untuk memverifikasi bahwa basis data tersebut beroperasi sebagai replika baca. Untuk informasi selengkapnya tentang penafsiran hasil, lihat dokumentasi MySQL.
4. Setelah replikasi pada basis data MySQL eksternal berhasil mengejar instans DB MySQL sumber, gunakan perintah `STOP REPLICA` MySQL DB untuk menghentikan replikasi dari instans DB MySQL sumber.

   > **Note**
   >
   > Versi MySQL sebelumnya menggunakan `STOP SLAVE`, bukan `STOP REPLICA`. Jika Anda menggunakan versi MySQL sebelum 8.0.23, gunakan `STOP SLAVE`.

5. Di replika baca Amazon RDS, panggil prosedur tersimpan `mysql.rds_start_replication`. Tindakan ini memungkinkan Amazon RDS untuk mulai membersihkan file log biner dari instans DB MySQL sumber.

Opsi untuk instans DB MySQL

Pada bagian berikut ini, Anda dapat menemukan deskripsi opsi, atau fitur tambahan, yang tersedia untuk instans Amazon RDS yang menjalankan mesin DB MySQL. Untuk mengaktifkan opsi ini, Anda dapat menambahkannya ke grup opsi khusus, lalu mengaitkan grup opsi dengan instans DB Anda. Untuk informasi selengkapnya tentang cara menggunakan grup opsi, lihat Menggunakan grup opsi.

Amazon RDS mendukung opsi berikut untuk MySQL:

Opsi	ID Opsi	Versi mesin
Dukungan MariaDB Audit Plugin untuk MySQL	`MARIADB_AUDIT_PLUGIN`	MySQL 8.0.28 dan versi 8.0 yang lebih tinggi Semua versi MySQL 5.7
Dukungan memcached MySQL	`MEMCACHED`	Semua versi MySQL 5.7 dan 8.0

Dukungan MariaDB Audit Plugin untuk MySQL

Amazon RDS menawarkan plugin audit untuk instans basis data MySQL berdasarkan MariaDB Audit Plugin sumber terbuka. Untuk informasi selengkapnya, lihat Repositori GitHub Audit Plugin untuk Server MySQL.

> **Note**
>
> Plugin audit untuk MySQL didasarkan pada MariaDB Audit Plugin. Di sepanjang artikel ini, kami menyebutnya sebagai MariaDB Audit Plugin.

MariaDB Audit Plugin. mencatat aktivitas basis data, termasuk pengguna yang masuk ke basis data dan kueri yang dijalankan terhadap basis data. Catatan aktivitas basis data disimpan dalam file log.

> **Note**
>
> Saat ini, MariaDB Plugin Audit hanya didukung untuk versi RDS for MySQL berikut:
>
> - MySQL 8.0.28 dan versi 8.0 yang lebih tinggi
> - Semua versi MySQL 5.7

Pengaturan opsi Audit Plugin

Amazon RDS mendukung pengaturan berikut untuk opsi MariaDB Audit Plugin.

Pengaturan opsi	Nilai valid	Nilai default	Deskripsi
`SERVER_AUDIT_FILE_PATH`	`/rdsdbdata/log/audit/`	`/rdsdbdata/log/audit/`	Lokasi file log. File log berisi catatan aktivitas yang ditentukan dalam `SERVER_AUDIT_EVENTS` . Untuk informasi selengkapnya, lihat Melihat dan mencantumkan file log basis data dan File log basis data MySQL.
`SERVER_AUDIT_FILE_`	1–1000000000	1000000	Ukuran dalam byte. Jika ukuran ini tercapai, file akan diputar. Untuk informasi selengkapnya, lihat Ikhtisar log basis data RDS for MySQL.

Pengaturan opsi	Nilai valid	Nilai default	Deskripsi
ROTATE_SIZE			
SERVER_AUDIT_FILE_ROTATIONS	0–100	9	Jumlah rotasi log yang akan disimpan ketika `server_audit_output_type=file` . Jika diatur ke 0, file log tidak akan pernah diputar. Untuk informasi selengkapnya, lihat Ikhtisar log basis data RDS for MySQL dan Mengunduh file log basis data.

Pengaturan opsi	Nilai valid	Nilai default	Deskripsi
SERVER_AUDIT_EVENTS	CONNECT, QUERY, QUERY_DDL, QUERY_DML, QUERY_DML_NO_SELECT, QUERY_DCL	CONNECT, QUERY	Jenis aktivitas yang akan dicatat di log. Penginstalan MariaDB Audit Plugin itu sendiri juga akan dicatat. • CONNECT: Mencatat koneksi yang berhasil dan tidak berhasil ke basis data, dan pemutusan dari basis data. • QUERY: Mencatat teks semua kueri yang dijalankan terhadap basis data. • QUERY_DDL : Mirip dengan peristiwa QUERY, tetapi hanya mengembalikan kueri bahasa definisi data (DDL) (CREATE, ALTER, dan sebagainya). • QUERY_DML : Mirip dengan peristiwa QUERY, tetapi hanya mengembalikan kueri bahasa manipulasi data (DML) (INSERT, UPDATE, dan sebagainya, serta SELECT). • QUERY_DML_NO_SELECT : Mirip dengan peristiwa QUERY_DML , tetapi tidak mencatat kueri SELECT. Pengaturan QUERY_DML_NO_SELECT didukung hanya untuk RDS for MySQL 5.7.34 dan versi 5.7 yang lebih tinggi, serta versi 8.0.25 dan versi 8.0 yang lebih tinggi. • QUERY_DCL : Mirip dengan peristiwa QUERY, tetapi hanya mengembalikan kueri bahasa kontrol data (DCL) (GRANT, REVOKE, dan sebagainya). Untuk MySQL, TABLE tidak didukung.

Pengaturan opsi	Nilai valid	Nilai default	Deskripsi
SERVER_AUDIT_INCL_USERS	Beberapa nilai yang dipisahkan koma	Tidak ada	Hanya menyertakan aktivitas dari pengguna tertentu. Secara default, aktivitas dicatat untuk semua pengguna SERVER_AUDIT_INCL_USERS dan SERVER_AUDIT_EXCL_USERS sama-sama bersifat eksklusif. Jika Anda menambahkan nilai ke SERVER_AUDIT_INCL_USERS , pastikan tidak ada nilai yang ditambahkan ke SERVER_AUDIT_EXCL_USERS .

Pengaturan opsi	Nilai valid	Nilai default	Deskripsi
SERVER_AUDIT_EXCL_USERS	Beberapa nilai yang dipisahkan koma	Tidak ada	Mengecualikan aktivitas dari pengguna tertentu. Secara default, aktivitas dicatat untuk semua pengguna SERVER_AUDIT_INCL_USERS dan SERVER_AUDIT_EXCL_USERS sama-sama bersifat eksklusif. Jika Anda menambahkan nilai ke SERVER_AUDIT_EXCL_USERS , pastikan tidak ada nilai yang ditambahkan ke SERVER_AUDIT_INCL_USERS . Pengguna rdsadmin membuat kueri basis data setiap detik untuk memeriksa kondisi basis data. Bergantung pada pengaturan Anda yang lain, aktivitas ini mungkin dapat menyebabkan ukuran file log Anda bertambah sangat besar dengan sangat cepat. Jika Anda tidak perlu mencatat aktivitas ini, tambahkan pengguna rdsadmin ke daftar SERVER_AUDIT_EXCL_USERS . **Note** Aktivitas CONNECT selalu dicatat untuk semua pengguna meskipun pengguna ditentukan untuk pengaturan opsi ini.
SERVER_AUDIT_LOGGING	ON	ON	Pencatatan aktif. Satu-satunya nilai yang valid adalah ON. Amazon RDS tidak mendukung penonaktifan pencatatan. Jika Anda ingin menonaktifkan pencatatan, hapus MariaDB Audit Plugin. Untuk informasi selengkapnya, lihat Menghapus MariaDB Audit Plugin.

Pengaturan opsi	Nilai valid	Nilai default	Deskripsi
SERVER_AUDIT_QUERY_LOG_LIMIT	0–2147483647	1024	Batas panjang string kueri dalam sebuah catatan.

Menambahkan MariaDB Audit Plugin

Proses umum untuk menambahkan MariaDB Audit Plugin ke instans DB adalah sebagai berikut:

- Buat grup opsi baru, atau salin atau ubah grup opsi yang ada
- Tambahkan opsi ke grup opsi
- Kaitkan grup opsi dengan instans DB

Setelah menambahkan MariaDB Audit Plugin, Anda tidak perlu memulai ulang instans DB Anda. Setelah grup opsi aktif, audit akan segera dimulai.

> **Important**
>
> Penambahan MariaDB Plugin Audit ke instans DB dapat menyebabkan gangguan. Sebaiknya tambahkan MariaDB Audit Plugin selama waktu pemeliharaan atau selama waktu beban kerja basis data rendah.

Untuk menambahkan MariaDB Audit Plugin

1. Tentukan grup opsi yang ingin Anda gunakan. Anda dapat membuat grup opsi baru atau menggunakan grup opsi yang ada. Jika Anda ingin menggunakan grup opsi yang ada, lanjutkan ke langkah berikutnya. Jika tidak, buat grup opsi DB kustom. Pilih mysql untuk Mesin, dan pilih 5.7 atau 8.0 untuk Versi mesin utama. Untuk informasi selengkapnya, lihat Membuat grup opsi.
2. Tambahkan opsi MARIADB_AUDIT_PLUGIN ke grup opsi, lalu konfigurasi pengaturan opsi. Untuk informasi selengkapnya tentang cara menambahkan opsi, lihat Menambahkan opsi ke grup opsi. Untuk informasi selengkapnya tentang setiap pengaturan, lihat Pengaturan opsi Audit Plugin.

3. Terapkan grup opsi ke instans DB baru atau yang sudah ada.

   - Untuk instans DB baru, Anda menerapkan grup opsi saat Anda meluncurkan instans. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.
   - Untuk instans DB yang ada, Anda menerapkan grup opsi dengan memodifikasi instans dan melampirkan grup opsi baru. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Format log Audit

File log direpresentasikan sebagai file variabel dipisahkan koma (CSV) dalam format UTF-8.

> **Tip**
>
> Entri file log tidak berurutan. Untuk mengurutkan entri, gunakan nilai stempel waktu. Untuk melihat peristiwa terbaru, Anda mungkin harus meninjau semua file log. Untuk fleksibilitas yang lebih besar dalam menyortir dan mencari data log, aktifkan pengaturan untuk mengunggah log audit ke CloudWatch dan lihat menggunakan antarmuka CloudWatch. Untuk melihat data audit dengan lebih banyak jenis bidang dan dengan output dalam format JSON, Anda juga dapat menggunakan fitur Aliran Aktivitas Basis Data. Untuk informasi selengkapnya, lihat Memantau Amazon RDS dengan Aliran Aktivitas Basis Data.

File log audit meliputi informasi yang dipisahkan koma berikut dalam baris, dalam urutan yang ditentukan:

Bidang	Deskripsi
timestamp	`YYYYMMDD` diikuti oleh `HH:MI:SS` (waktu 24 jam) untuk peristiwa yang dicatat.
serverhost	Nama instans tempat peristiwa dicatat.
nama pengguna	Nama pengguna yang terhubung dengan pengguna.
host	Host yang terhubung ke pengguna.
connectionid	Nomor ID koneksi untuk operasi yang dicatat.

Bidang	Deskripsi
queryid	Nomor ID kueri, yang dapat digunakan untuk menemukan peristiwa tabel hubungan dan kueri terkait. Untuk peristiwa `TABLE`, beberapa baris ditambahkan.
operation	Jenis tindakan terekam. Kemungkinan nilainya adalah: `CONNECT`, `QUERY`, `READ`, `WRITE`, `CREATE`, `ALTER`, `RENAME`, dan `DROP`.
database	Basis data aktif, yang diatur oleh perintah `USE`.
object	Untuk peristiwa `QUERY`, nilai ini menunjukkan kueri yang dilakukan basis data. Untuk peristiwa `TABLE`, nilai ini menunjukkan nama tabel.
retcode	Kode hasil dari operasi yang dicatat.
connection_type	Status keamanan koneksi ke server. Kemungkinan nilainy adalah: • `0` – Undefined • `1`— TCP/IP • `2` – Soket • `3` – Pipa bernama • `4` – SSL/TLS • `5` – Memori bersama Bidang ini disertakan hanya untuk RDS for MySQL versi 5.7.34 dan versi 5.7 yang lebih tinggi, serta semua versi 8.0.

Melihat dan mengunduh log MariaDB Audit Plugin.

Setelah mengaktifkan MariaDB Audit Plugin, Anda dapat mengakses hasilnya di file log dengan cara yang sama seperti Anda mengakses file log berbasis teks lainnya. File log audit terletak di `/rdsdbdata/log/audit/`. Untuk informasi tentang cara melihat file log di konsol, lihat Melihat dan mencantumkan file log basis data. Untuk informasi tentang cara mengunduh file log, lihat Mengunduh file log basis data.

Mengubah pengaturan MariaDB Audit Plugin

Setelah mengaktifkan MariaDB Plugin Audit, Anda dapat mengubah pengaturan. Untuk informasi selengkapnya tentang cara mengubah pengaturan opsi, lihat Memodifikasi pengaturan opsi. Untuk informasi selengkapnya tentang setiap pengaturan, lihat Pengaturan opsi Audit Plugin.

Menghapus MariaDB Audit Plugin

Amazon RDS tidak mendukung penonaktifan log di MariaDB Audit Plugin. Namun, Anda dapat menghapus plugin dari instans DB. Jika Anda menghapus MariaDB Audit Plugin, instans DB dimulai ulang secara otomatis untuk menghentikan audit.

Untuk menghapus MariaDB Audit Plugin dari instans DB, lakukan salah satu hal berikut:

- Hapus opsi MariaDB Plugin Audit dari grup opsi yang menjadi miliknya. Perubahan ini memengaruhi semua instans DB yang menggunakan grup opsi tersebut. Untuk informasi selengkapnya, lihat Menghapus opsi dari grup opsi
- Ubah instans DB dan tentukan grup opsi berbeda yang tidak menyertakan plugin. Perubahan ini memengaruhi instans DB tunggal. Anda dapat menentukan grup opsi default (kosong) atau grup opsi kustom yang berbeda. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Dukungan memcached MySQL

Amazon RDS mendukung penggunaan antarmuka `memcached` pada tabel InnoDB yang diperkenalkan di MySQL 5.6. API `memcached` memungkinkan aplikasi menggunakan tabel InnoDB dengan cara yang serupa dengan penyimpanan data nilai kunci NoSQL.

Antarmuka `memcached` adalah cache sederhana berbasis kunci. Aplikasi menggunakan `memcached` untuk menyisipkan, memanipulasi, dan mengambil pasangan data nilai kunci dari cache. MySQL 5.6 memperkenalkan plugin yang menerapkan layanan daemon yang mengekspos data dari tabel InnoDB melalui protokol `memcached`. Untuk informasi selengkapnya tentang plugin `memcached` MySQL, lihat InnoDB integration with memcached.

Untuk mengaktifkan dukungan memcached instans DB RDS for MySQL

1. Tentukan grup keamanan yang akan digunakan untuk mengontrol akses ke antarmuka `memcached`. Jika set aplikasi yang sudah menggunakan antarmuka SQL sama dengan yang akan mengakses antarmuka `memcached`, Anda dapat menggunakan grup keamanan VPC yang sudah ada yang digunakan oleh antarmuka SQL. Jika set aplikasi yang berbeda akan mengakses antarmuka `memcached`, tentukan VPC atau grup keamanan DB baru. Untuk informasi selengkapnya tentang cara mengelola grup keamanan, lihat Mengontrol akses dengan grup keamanan
2. Buat grup opsi DB kustom, dengan memilih MySQL sebagai jenis dan versi mesin. Untuk informasi selengkapnya tentang cara membuat grup opsi, lihat Membuat grup opsi.
3. Tambahkan opsi `MEMCACHED` untuk grup opsi. Tentukan port yang akan digunakan antarmuka `memcached`, dan grup keamanan yang akan digunakan untuk mengontrol akses ke antarmuka. Untuk informasi selengkapnya tentang cara menambahkan opsi, lihat Menambahkan opsi ke grup opsi.
4. Ubah pengaturan opsi untuk mengonfigurasi parameter `memcached`, jika perlu. Untuk informasi selengkapnya tentang cara mengubah pengaturan opsi, lihat Memodifikasi pengaturan opsi.
5. Terapkan grup opsi ke instans. Amazon RDS memungkinkan dukungan `memcached` untuk instans tersebut ketika grup opsi diterapkan:

   - Anda mengaktifkan dukungan `memcached` untuk instans baru dengan menentukan grup opsi kustom saat Anda meluncurkan instans. Untuk informasi selengkapnya tentang cara meluncurkan instans MySQL, lihat Membuat instans DB Amazon RDS.

- Anda mengaktifkan dukungan memcached untuk instans yang ada dengan menentukan grup opsi kustom saat Anda memodifikasi instans. Untuk informasi selengkapnya tentang cara memodifikasi instans DB, lihat Memodifikasi instans DB Amazon RDS.

6. Tentukan kolom mana dalam tabel MySQL Anda yang dapat diakses melalui antarmuka memcached. Plugin memcached membuat tabel katalog bernama containers dalam basis data khusus bernama innodb_memcache. Anda memasukkan baris ke tabel containers guna memetakan tabel InnoDB untuk akses melalui memcached. Anda menentukan kolom dalam tabel InnoDB yang digunakan untuk menyimpan nilai kunci memcached, dan satu atau lebih kolom yang digunakan untuk menyimpan nilai-nilai data yang terkait dengan kunci. Anda juga menentukan nama yang digunakan aplikasi memcached untuk merujuk ke set kolom tersebut. Untuk detail tentang cara menyisipkan baris ke dalam tabel containers, lihat Internal plugin memcached InnoDB. Untuk contoh pemetaan tabel InnoDB dan cara mengaksesnya melalui memcached, lihat Writing applications for the InnoDB memcached plugin.
7. Jika aplikasi yang mengakses antarmuka memcached berada di komputer atau instans EC2 yang berbeda dengan aplikasi yang menggunakan antarmuka SQL, tambahkan informasi koneksi untuk komputer tersebut ke grup keamanan VPC yang terkait dengan instans MySQL. Untuk informasi selengkapnya tentang cara mengelola grup keamanan, lihat Mengontrol akses dengan grup keamanan.

Anda menonaktifkan dukungan memcached untuk instans dengan memodifikasi instans dan menentukan grup opsi default untuk versi MySQL. Untuk informasi selengkapnya tentang cara memodifikasi instans DB, lihat Memodifikasi instans DB Amazon RDS.

Pertimbangan keamanan memcached MySQL

Protokol memcached tidak mendukung autentikasi pengguna. Untuk informasi selengkapnya tentang pertimbangan keamanan memcached MySQL, lihat Security Considerations for the InnoDB memcached Plugin di dokumentasi MySQL.

Anda dapat mengambil tindakan berikut untuk membantu meningkatkan keamanan antarmuka memcached:

- Tentukan port yang berbeda dengan default 11211 ketika menambahkan opsi MEMCACHED untuk grup opsi.
- Pastikan bahwa Anda mengaitkan antarmuka memcached dengan grup keamanan VPC yang membatasi akses ke alamat klien serta instans EC2 yang diketahui dan tepercaya. Untuk informasi

selengkapnya tentang cara mengelola grup keamanan, lihat Mengontrol akses dengan grup keamanan.

Informasi koneksi memcached MySQL

Untuk mengakses antarmuka `memcached`, aplikasi harus menentukan nama DNS instans Amazon RDS dan nomor port `memcached`. Misalnya, jika suatu instans memiliki nama DNS `my-cache-instance.cg034hpkmmjt.region.rds.amazonaws.com` dan antarmuka memcached menggunakan port 11212, informasi koneksi yang ditentukan dalam PHP akan:

```
<?php

$cache = new Memcache;
$cache->connect('my-cache-instance.cg034hpkmmjt.region.rds.amazonaws.com',11212);
?>
```

Untuk menemukan nama DNS dan memcached port instans DB MySQL

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di sudut kanan atas AWS Management Console, pilih wilayah yang berisi instans DB.
3. Di panel navigasi, pilih Basis data.
4. Pilih nama instans DB MySQL untuk menampilkan detailnya.
5. Di bagian Terhubung, perhatikan nilai bidang Titik Akhir. Nama DNS sama dengan titik akhir. Perhatikan juga bahwa port di bagian Terhubung tidak digunakan untuk mengakses antarmuka `memcached`.
6. Di bagian Detail, perhatikan nama yang tercantum di kolom Grup Opsi.
7. Di panel navigasi, pilih Grup opsi.
8. Pilih nama grup opsi yang digunakan oleh instans DB MySQL untuk menampilkan detail grup opsi. Di bagian Opsi, perhatikan nilai pengaturan Port untuk opsi MEMCACHED.

Pengaturan opsi memcached MySQL

Amazon RDS mengekspos parameter `memcached` MySQL sebagai pengaturan opsi di opsi `MEMCACHED` Amazon RDS.

Parameter memcached MySQL

- DAEMON_MEMCACHED_R_BATCH_SIZE – bilangan bulat yang menentukan jumlah operasi baca memcached (get) yang harus dilakukan sebelum melakukan COMMIT untuk memulai transaksi baru. Nilai yang diizinkan adalah 1 hingga 4294967295 dan default-nya adalah 1. Opsi tidak berlaku sampai instans dimulai ulang.
- DAEMON_MEMCACHED_W_BATCH_SIZE – bilangan bulat yang menentukan berapa banyak operasi tulis memcached, seperti add, set, atau incr, yang harus dilakukan sebelum melakukan COMMIT untuk memulai transaksi baru. Nilai yang diizinkan adalah 1 hingga 4294967295 dan default-nya adalah 1. Opsi tidak berlaku sampai instans dimulai ulang.
- INNODB_API_BK_COMMIT_INTERVAL – bilangan bulat yang menentukan seberapa sering penerapan commit otomatis koneksi idle yang menggunakan antarmuka memcached InnoDB. Nilai yang diizinkan adalah 1 hingga 1073741824 dan default-nya adalah 5. Opsi akan langsung diterapkan tanpa harus memulai ulang instans.
- INNODB_API_DISABLE_ROWLOCK – Boolean yang menonaktifkan (1 (true)) atau mengaktifkan (0 (false)) penggunaan kunci baris saat menggunakan antarmuka memcached InnoDB. Default-nya adalah 0 (false). Opsi tidak berlaku sampai instans dimulai ulang.
- INNODB_API_ENABLE_MDL – Boolean yang saat ditetapkan ke 0 (false) akan mengunci tabel yang digunakan oleh plugin memcached InnoDB, sehingga tidak dapat dihapus atau diubah oleh DDL melalui antarmuka SQL. Default-nya adalah 0 (false). Opsi tidak berlaku sampai instans dimulai ulang.
- INNODB_API_TRX_LEVEL – bilangan bulat yang menentukan tingkat isolasi transaksi untuk kueri yang diproses oleh antarmuka memcached. Nilai yang diizinkan adalah 0 hingga 3. Default-nya adalah 0. Opsi tidak berlaku sampai instans dimulai ulang.

Amazon RDS mengonfigurasi parameter memcached MySQL berikut dan parameter ini tidak dapat dimodifikasi: DAEMON_MEMCACHED_LIB_NAME, DAEMON_MEMCACHED_LIB_PATH, dan INNODB_API_ENABLE_BINLOG. Parameter yang diatur administrator MySQL dengan menggunakan daemon_memcached_options yang tersedia sebagai pengaturan opsi MEMCACHED individu di Amazon RDS.

Parameter daemon_memcached_options MySQL

- BINDING_PROTOCOL – string yang menentukan protokol pengikatan yang akan digunakan. Nilai yang diizinkan adalah auto, ascii, atau binary. Default-nya adalah auto, yang berarti server

menegosiasikan protokol secara otomatis dengan klien. Opsi tidak berlaku sampai instans dimulai ulang.

- `BACKLOG_QUEUE_LIMIT` – bilangan bulat yang menentukan jumlah koneksi jaringan yang dapat menunggu untuk diproses oleh `memcached`. Meningkatkan batas ini dapat mengurangi kesalahan yang diterima oleh klien yang tidak dapat terhubung ke instans `memcached`, tetapi tidak meningkatkan performa server. Nilai yang diizinkan adalah 1 hingga 2048 dan default-nya adalah 1024. Opsi tidak berlaku sampai instans dimulai ulang.
- `CAS_DISABLED` – Boolean yang mengaktifkan (1 (true)) atau menonaktifkan (0 (false)) penggunaan perbandingan dan pertukaran (CAS), yang mengurangi ukuran per item sebesar 8 byte. Default-nya adalah 0 (false). Opsi tidak berlaku sampai instans dimulai ulang.
- `CHUNK_SIZE` – bilangan bulat yang menentukan ukuran potongan minimum, dalam byte, untuk mengalokasikan kunci, nilai, dan bendera item terkecil. Nilai yang diizinkan adalah 1 hingga 48. Default-nya adalah 48 dan Anda dapat meningkatkan efisiensi memori secara signifikan dengan nilai yang lebih rendah. Opsi tidak berlaku sampai instans dimulai ulang.
- `CHUNK_SIZE_GROWTH_FACTOR` – bilangan desimal yang mengontrol ukuran potongan yang baru. Ukuran potongan baru adalah ukuran potongan sebelumnya dikalikan `CHUNK_SIZE_GROWTH_FACTOR`. Nilai yang diizinkan adalah 1 hingga 2, default-nya adalah 1,25. Opsi tidak berlaku sampai instans dimulai ulang.
- `ERROR_ON_MEMORY_EXHAUSTED` – Boolean yang jika ditetapkan ke 1 (true) akan menentukan bahwa `memcached` akan menghasilkan kesalahan daripada mengosongkan item ketika tidak ada lagi memori untuk menyimpan item. Jika ditetapkan ke 0 (false), `memcached` akan mengosongkan item jika tidak ada lagi memori. Default-nya adalah 0 (false). Opsi tidak berlaku sampai instans dimulai ulang.
- `MAX_SIMULTANEOUS_CONNECTIONS` – bilangan bulat yang menentukan jumlah maksimum koneksi serentak. Menetapkan nilai ini ke nilai di bawah 10 akan mencegah MySQL dimulai. Nilai yang diizinkan adalah 10 hingga 1024 dan default-nya adalah 1024. Opsi tidak berlaku sampai instans dimulai ulang.
- `VERBOSITY` – string yang menentukan tingkat informasi yang dicatat dalam log kesalahan MySQL dengan layanan `memcached`. Default-nya adalah v. Opsi ini tidak berlaku hingga instans dimulai ulang. Nilai yang diizinkan adalah:
  - `v` – Mencatat log kesalahan dan peringatan saat menjalankan loop peristiwa utama.
  - `vv` – Selain informasi yang dicatat oleh v, nilai ini juga mencatat setiap perintah dan respons klien.
  - `vvv` – Selain informasi yang dicatat oleh vv, nilai ini juga mencatat transisi keadaan internal.

Amazon RDS mengonfigurasi parameter DAEMON_MEMCACHED_OPTIONS MySQL berikut, dan bersifat tidak dapat dimodifikasi: DAEMON_PROCESS, LARGE_MEMORY_PAGES, MAXIMUM_CORE_FILE_LIMIT, MAX_ITEM_SIZE, LOCK_DOWN_PAGE_MEMORY, MASK, IDFILE, REQUESTS_PER_EVENT, SOCKET, dan USER.

Parameter untuk MySQL

Secara default, instans DB MySQL menggunakan grup parameter DB yang spesifik untuk basis data MySQL. Grup parameter ini berisi parameter untuk mesin basis data MySQL. Untuk informasi tentang cara menangani grup parameter dan mengatur parameter, lihat Bekerja dengan grup parameter.

Parameter RDS for MySQL diatur ke nilai default dari mesin penyimpanan yang Anda pilih. Untuk mengetahui informasi selengkapnya tentang parameter MySQL, lihat dokumentasi MySQL. Untuk informasi selengkapnya tentang mesin penyimpanan MySQL, lihat Mesin penyimpanan yang didukung untuk RDS for MySQL.

Anda dapat melihat parameter yang tersedia untuk versi RDS for MySQL tertentu menggunakan konsol RDS atau AWS CLI. Untuk informasi tentang cara melihat parameter dalam grup parameter MySQL di konsol RDS, lihat Melihat nilai parameter untuk grup parameter DB.

Dengan menggunakan AWS CLI, Anda dapat melihat parameter untuk versi RDS for MySQL dengan menjalankan perintah `describe-engine-default-parameters`. Tentukan salah satu dari nilai-nilai berikut untuk opsi `--db-parameter-group-family`:

- `mysql8.0`
- `mysql5.7`

Misalnya, untuk melihat parameter untuk RDS for MySQL versi 8.0, jalankan perintah berikut.

```
aws rds describe-engine-default-parameters --db-parameter-group-family mysql8.0
```

Output Anda akan terlihat serupa dengan yang berikut ini.

```
{
    "EngineDefaults": {
        "Parameters": [
            {
                "ParameterName": "activate_all_roles_on_login",
                "ParameterValue": "0",
                "Description": "Automatically set all granted roles as active after the
 user has authenticated successfully.",
                "Source": "engine-default",
                "ApplyType": "dynamic",
                "DataType": "boolean",
```

```
                "AllowedValues": "0,1",
                "IsModifiable": true
            },
            {
                "ParameterName": "allow-suspicious-udfs",
                "Description": "Controls whether user-defined functions that have only
 an xxx symbol for the main function can be loaded",
                "Source": "engine-default",
                "ApplyType": "static",
                "DataType": "boolean",
                "AllowedValues": "0,1",
                "IsModifiable": false
            },
            {
                "ParameterName": "auto_generate_certs",
                "Description": "Controls whether the server autogenerates SSL key and
 certificate files in the data directory, if they do not already exist.",
                "Source": "engine-default",
                "ApplyType": "static",
                "DataType": "boolean",
                "AllowedValues": "0,1",
                "IsModifiable": false
            },
        ...
```

Untuk hanya mencantumkan parameter yang dapat dimodifikasi untuk MySQL versi 8.0, jalankan perintah berikut.

Untuk Linux, macOS, atau Unix:

```
aws rds describe-engine-default-parameters --db-parameter-group-family mysql8.0 \
   --query 'EngineDefaults.Parameters[?IsModifiable==`true`]'
```

Untuk Windows:

```
aws rds describe-engine-default-parameters --db-parameter-group-family mysql8.0 ^
   --query "EngineDefaults.Parameters[?IsModifiable==`true`]"
```

Tugas umum DBA untuk instans DB MySQL

Dalam konten berikut, Anda dapat menemukan deskripsi implementasi khusus Amazon RDS dari beberapa tugas DBA umum untuk instans DB yang menjalankan mesin database MySQL. Untuk memberikan pengalaman layanan terkelola, Amazon RDS tidak memberikan akses shell ke instans DB. Selain itu, Amazon RDS juga membatasi akses ke prosedur dan tabel sistem tertentu yang memerlukan hak istimewa tingkat lanjut.

Untuk informasi tentang cara bekerja dengan file log MySQL di Amazon RDS, lihat File log basis data MySQL.

Topik

Memahami pengguna yang telah ditentukan

Amazon RDS secara otomatis membuat beberapa pengguna yang telah ditentukan dengan RDS baru untuk instans MySQL DB. Pengguna yang telah ditentukan sebelumnya dan hak istimewa mereka tidak dapat diubah. Anda tidak dapat menghapus, mengganti nama, atau memodifikasi hak istimewa untuk pengguna yang telah ditentukan ini. Mencoba untuk melakukannya akan menghasilkan kesalahan.

- rdsadmin — Pengguna yang dibuat untuk menangani banyak tugas manajemen yang administrator dengan `superuser` hak istimewa akan melakukan pada database MySQL mandiri. Pengguna ini digunakan secara internal oleh RDS untuk MySQL untuk banyak tugas manajemen.
- rdsrepladmin — Pengguna yang digunakan secara internal oleh Amazon RDS untuk mendukung aktivitas replikasi pada RDS untuk instans dan cluster MySQL DB.

Mengakhiri sesi atau kueri

Anda dapat mengakhiri sesi atau kueri pengguna di instans DB dengan menggunakan perintah `rds_kill` dan `rds_kill_query`. Pertama, hubungkan ke instans DB MySQL Anda, lalu jalankan

perintah yang sesuai seperti yang ditunjukkan berikut ini. Untuk informasi selengkapnya, lihat Menghubungkan ke instans DB yang menjalankan mesin basis data MySQL.

```
CALL mysql.rds_kill(thread-ID)
CALL mysql.rds_kill_query(thread-ID)
```

Misalnya, untuk mengakhiri sesi yang berjalan di thread 99, ketikkan hal berikut:

```
CALL mysql.rds_kill(99);
```

Untuk mengakhiri kueri yang sedang berjalan pada thread 99, ketikkan hal berikut:

```
CALL mysql.rds_kill_query(99);
```

Melewati kesalahan replikasi saat ini

Anda dapat melewati kesalahan pada replika baca jika kesalahan tersebut menyebabkan replika baca Anda berhenti merespons dan kesalahan tersebut tidak memengaruhi integritas data Anda.

> **Note**
>
> Pertama, pastikan bahwa kesalahan tersebut dapat dilewati dengan aman. Di utilitas MySQL, hubungkan ke replika baca dan jalankan perintah MySQL berikut:
>
> ```
> SHOW REPLICA STATUS\G
> ```
>
> Untuk informasi tentang nilai yang dikembalikan, lihat dokumentasi MySQL.
> Versi MySQL sebelumnya menggunakan `SHOW SLAVE STATUS`, bukan `SHOW REPLICA STATUS`. Jika Anda menggunakan versi MySQL sebelum 8.0.23, gunakan `SHOW SLAVE STATUS`.

Anda dapat melewati kesalahan pada replika baca dengan cara berikut.

Topik

- Memanggil prosedur mysql.rds_skip_repl_error
- Mengatur parameter slave_skip_errors

Memanggil prosedur mysql.rds_skip_repl_error

Amazon RDS menyediakan prosedur tersimpan yang dapat Anda panggil untuk melewati kesalahan pada replika baca Anda. Pertama, hubungkan ke replika baca Anda, lalu jalankan perintah yang sesuai seperti yang ditunjukkan berikut ini. Untuk informasi selengkapnya, lihat Menghubungkan ke instans DB yang menjalankan mesin basis data MySQL.

Untuk melewati kesalahan, jalankan perintah berikut.

```
CALL mysql.rds_skip_repl_error;
```

Perintah ini tidak berpengaruh jika Anda menjalankannya di instans DB sumber, atau di replika baca yang belum mengalami kesalahan replikasi.

Untuk informasi selengkapnya, seperti versi MySQL yang mendukung `mysql.rds_skip_repl_error`, lihat mysql.rds_skip_repl_error.

> **Important**
>
> Jika Anda mencoba untuk memanggil `mysql.rds_skip_repl_error` dan menemukan kesalahan berikut: `ERROR 1305 (42000): PROCEDURE mysql.rds_skip_repl_error does not exist`, tingkatkan instans DB MySQL Anda ke versi minor terbaru atau salah satu versi minor minimum yang tercantum dalam mysql.rds_skip_repl_error.

Mengatur parameter slave_skip_errors

Untuk melewati satu atau beberapa kesalahan, Anda dapat mengatur parameter `slave_skip_errors` statis pada replika baca. Anda dapat mengatur parameter ini untuk melewati satu atau beberapa kode kesalahan replikasi spesifik. Saat ini, Anda dapat mengatur parameter ini hanya untuk instans DB RDS for MySQL 5.7. Setelah Anda mengubah pengaturan untuk parameter ini, pastikan untuk melakukan boot ulang instans DB Anda agar pengaturan baru dapat diterapkan. Untuk informasi tentang cara mengatur parameter ini, lihat dokumentasi MySQL.

Sebaiknya atur parameter ini dalam grup parameter DB terpisah. Anda dapat mengaitkan grup parameter DB ini hanya dengan replika baca yang perlu melewati kesalahan. Mengikuti praktik terbaik ini akan mengurangi potensi dampak pada instans DB dan replica baca lainnya.

> **Important**
>
> Mengatur nilai nondefault untuk parameter ini dapat menyebabkan inkonsistensi replikasi. Atur parameter ini ke nilai nondefault hanya jika Anda tidak memiliki opsi lain untuk menyelesaikan masalah dan Anda yakin bahwa akan ada potensi dampak pada data replika baca Anda.

Bekerja dengan tablespace InnoDB untuk meningkatkan waktu pemulihan kerusakan

Setiap tabel di MySQL terdiri dari definisi, data, dan indeks tabel. Mesin penyimpanan InnoDB MySQL menyimpan data dan indeks tabel di tablespace. InnoDB membuat tablespace bersama global yang berisi kamus data dan metadata relevan lainnya, serta dapat berisi data dan indeks tabel. InnoDB juga dapat membuat tablespace terpisah untuk setiap tabel dan partisi. Tablespace terpisah ini disimpan dalam file berekstensi .ibd dan judul setiap tablespace berisi nomor pengidentifikasi unik.

Amazon RDS menyediakan parameter dalam grup parameter MySQL yang disebut `innodb_file_per_table`. Parameter ini mengontrol apakah InnoDB menambahkan data dan indeks tabel baru ke tablespace bersama (dengan menetapkan nilai parameter ke 0) atau ke tablespace terpisah (dengan menetapkan nilai parameter ke 1). Amazon RDS menetapkan nilai default untuk parameter `innodb_file_per_table` ke 1, yang memungkinkan Anda meletakkan tabel InnoDB terpisah dan mengklaim ulang penyimpanan yang digunakan oleh tabel tersebut untuk instans DB. Biasanya, penetapan parameter `innodb_file_per_table` ke 1 adalah pengaturan yang direkomendasikan.

Anda harus menetapkan parameter `innodb_file_per_table` ke 0 ketika Anda memiliki tabel dalam jumlah besar. Misalnya, Anda memiliki lebih dari 1.000 tabel saat menggunakan penyimpanan SSD standar (magnetik) atau SSD umum atau lebih dari 10.000 tabel saat Anda menggunakan penyimpanan IOPS yang Tersedia. Saat Anda menetapkan parameter ini ke 0, tablespace terpisah tidak dibuat dan hal ini dapat meningkatkan waktu yang diperlukan untuk pemulihan kerusakan basis data.

MySQL memproses setiap file metadata, yang mencakup tablespace, selama siklus pemulihan crash. Waktu yang diperlukan oleh MySQL untuk memproses informasi metadata di tablespace bersama dapat diabaikan, dibandingkan dengan waktu yang diperlukan untuk memproses ribuan file tablespace ketika ada beberapa tablespace. Karena nomor tablespace disimpan di judul setiap file, waktu gabungan untuk membaca semua file tablespace tersebut bisa mencapai beberapa

jam. Misalnya, pemrosesan sejuta tablespace InnoDB pada penyimpanan standar selama siklus pemulihan crash dapat memakan waktu mulai lima hingga delapan jam. Dalam beberapa kasus, InnoDB dapat menentukan apakah pembersihan tambahan diperlukan setelah siklus pemulihan crash selesai sehingga siklus pemulihan crash akan dimulai lagi. Hal ini akan memperpanjang waktu pemulihan. Perlu diperhatikan bahwa siklus pemulihan crash juga melibatkan transaksi rolling back, perbaikan halaman yang rusak, dan operasi lain selain pemrosesan informasi tablespace.

Karena parameter `innodb_file_per_table` berada di grup parameter, Anda dapat mengubah nilai parameter dengan mengedit grup parameter yang digunakan oleh instans DB Anda tanpa perlu melakukan boot ulang instans DB. Setelah pengaturan diubah, misalnya, dari 1 (buat tabel terpisah) ke 0 (gunakan tablespace bersama), tabel InnoDB baru akan ditambahkan ke tablespace bersama sedangkan tabel yang ada tetap memiliki tablespace terpisah. Untuk memindahkan tabel InnoDB ke tablespace bersama, Anda harus menggunakan perintah `ALTER TABLE`.

Memigrasikan beberapa tablespace ke tablespace bersama

Anda dapat memindahkan metadata tabel InnoDB dari tablespace-nya ke tablespace bersama. Hal ini akan membuat ulang metadata tabel sesuai dengan pengaturan parameter `innodb_file_per_table`. Pertama, hubungkan ke instans MySQL DB Anda, lalu jalankan perintah yang sesuai seperti yang ditunjukkan berikut. Untuk informasi selengkapnya, lihat Menghubungkan ke instans DB yang menjalankan mesin basis data MySQL.

```
ALTER TABLE table_name ENGINE = InnoDB, ALGORITHM=COPY;
```

Misalnya, kueri berikut mengembalikan pernyataan `ALTER TABLE` untuk setiap tabel InnoDB yang tidak ada di tablespace bersama.

Untuk instans DB MySQL 5.7:

```
SELECT CONCAT('ALTER TABLE `',
REPLACE(LEFT(NAME , INSTR((NAME), '/') - 1), '`', '``'), '`.`',
REPLACE(SUBSTR(NAME FROM INSTR(NAME, '/') + 1), '`', '``'), '` ENGINE=InnoDB,
 ALGORITHM=COPY;') AS Query
FROM INFORMATION_SCHEMA.INNODB_SYS_TABLES
WHERE SPACE <> 0 AND LEFT(NAME, INSTR((NAME), '/') - 1) NOT IN ('mysql','');
```

Untuk instans DB MySQL 8.0:

```
SELECT CONCAT('ALTER TABLE `',
REPLACE(LEFT(NAME , INSTR((NAME), '/') - 1), '`', '``'), '`.`',
```

```
REPLACE(SUBSTR(NAME FROM INSTR(NAME, '/') + 1), '``', '```'), '` ENGINE=InnoDB,
 ALGORITHM=COPY;') AS Query
FROM INFORMATION_SCHEMA.INNODB_TABLES
WHERE SPACE <> 0 AND LEFT(NAME, INSTR((NAME), '/') - 1) NOT IN ('mysql','');
```

Pembuatan ulang tabel MySQL untuk memindahkan metadata tabel tersebut ke tablespace bersama memerlukan ruang penyimpanan tambahan sementara guna membuat ulang tabel. Jadi, instans DB harus memiliki ruang penyimpanan yang tersedia. Selama pembuatan ulang, tabel dikunci dan tidak dapat diakses oleh kueri. Untuk tabel kecil atau tabel yang jarang diakses, hal ini mungkin tidak menjadi masalah. Untuk tabel besar atau tabel yang sering diakses dalam lingkungan yang sangat serentak, Anda dapat membuat ulang tabel pada replika baca.

Anda dapat membuat replika baca dan memigrasi metadata tabel ke tablespace bersama pada replika baca. Meskipun pernyataan ALTER TABLE memblokir akses pada replika baca, instans DB sumber tidak terpengaruh. Instans DB sumber akan terus membuat log binernya, sementara replika baca mengalami lag selama proses pembuatan ulang tabel. Karena pembuatan ulang ini memerlukan ruang penyimpanan tambahan dan file log pemutaran ulang bisa menjadi besar, Anda harus membuat replika baca dengan alokasi penyimpanan yang lebih besar dari instans DB sumber.

Guna membuat replika baca dan membuat ulang InnoDB untuk menggunakan tablespace bersama, lakukan langkah-langkah berikut:

1. Pastikan retensi cadangan diaktifkan di instans DB sumber sehingga pencatatan biner diaktifkan.
2. Gunakan AWS Management Console atau AWS CLI untuk membuat replika baca untuk instance DB sumber. Karena pembuatan replika baca melibatkan banyak proses yang sama seperti pemulihan crash, proses pembuatannya dapat memakan waktu lama jika ada tablespace InnoDB dalam jumlah besar. Alokasikan lebih banyak ruang penyimpanan pada replika baca dibandingkan dengan yang saat ini digunakan pada instans DB sumber.
3. Setelah replika baca dibuat, buat grup parameter dengan pengaturan parameter `read_only = 0` dan `innodb_file_per_table = 0`. Kemudian hubungkan grup parameter dengan replika baca.
4. Terbitkan pernyataan SQL berikut untuk semua tabel yang ingin Anda migrasikan pada replika:

   ```
   ALTER TABLE name ENGINE = InnoDB
   ```

5. Saat semua pernyataan `ALTER TABLE` Anda pada replika baca selesai, pastikan bahwa replika baca tersebut telah terhubung ke instans DB sumber dan bahwa kedua instans tersebut sudah sinkron.

6. Gunakan konsol atau CLI untuk mempromosikan replika baca sebagai instans. Pastikan grup parameter yang digunakan untuk instans DB mandiri baru telah menetapkan parameter `innodb_file_per_table` ke 0. Ubah nama instans DB mandiri baru, lalu arahkan aplikasi apa pun ke instans DB mandiri baru.

Mengelola Global Status History

> **Tip**
>
> Untuk menganalisis performa basis data, Anda juga dapat menggunakan Wawasan Performa di Amazon RDS. Untuk informasi selengkapnya, lihat Memantau muatan DB dengan Wawasan Performa di Amazon RDS.

MySQL mempertahankan banyak variabel status yang memberikan informasi tentang operasinya. Nilai variabel tersebut dapat membantu Anda mendeteksi masalah penguncian atau memori pada instans DB. Nilai variabel status ini bersifat kumulatif sejak instans DB dimulai terakhir kali. Anda dapat menetapkan ulang sebagian besar variabel status ke 0 dengan menggunakan perintah `FLUSH STATUS`.

Agar dapat memantau nilai ini dari waktu ke waktu, Amazon RDS menyediakan serangkaian prosedur yang akan mengambil snapshot nilai-nilai variabel status ini dari waktu ke waktu dan menuliskannya ke tabel, beserta perubahan apa pun yang dibuat sejak snapshot terakhir. Infrastruktur ini, yang disebut Global Status History (GoSH), diinstal di seluruh instans DB MySQL mulai versi 5.5.23. GoSH dinonaktifkan secara default.

Untuk mengaktifkan GoSH, aktifkan penjadwal peristiwa terlebih dahulu dari grup parameter DB dengan menetapkan parameter `event_scheduler` ke `ON`. Untuk instans DB MySQL yang menjalankan MySQL 5.7, tetapkan juga parameter `show_compatibility_56` ke `1`. Untuk informasi tentang cara membuat dan memodifikasi grup parameter DB, lihat Bekerja dengan grup parameter. Untuk informasi tentang dampak dari pengaktifan parameter ini, lihat show_compatibility_56 di Manual Referensi MySQL 5.7.

Anda kemudian dapat menggunakan prosedur dalam tabel berikut untuk mengaktifkan dan mengonfigurasi GoSH. Pertama, hubungkan ke instans DB MySQL Anda, lalu jalankan perintah yang sesuai seperti yang ditunjukkan berikut. Untuk informasi selengkapnya, lihat Menghubungkan ke instans DB yang menjalankan mesin basis data MySQL. Untuk setiap prosedur, ketikkan berikut ini:

```
CALL procedure-name;
```

Dengan procedure-name merupakan salah satu prosedur dalam tabel.

Prosedur	Deskripsi
`mysql.rds_enable_gsh_collector`	Mengaktifkan GoSH untuk mengambil snapshot default pada interval yang ditentukan oleh `rds_set_gsh_collector` .
`mysql.rds_set_gsh_collector`	Menentukan interval, dalam menit, antara snapshot. Nilai defaultnya adalah 5.
`mysql.rds_disable_gsh_collector`	Menonaktifkan snapshot.
`mysql.rds_collect_global_status_history`	Mengambil snapshot sesuai permintaan.
`mysql.rds_enable_gsh_rotation`	Mengaktifkan rotasi isi tabel `mysql.rds_global_status_history` ke `mysql.rds_global_status_history_old` pada interval yang ditentukan oleh `rds_set_gsh_rotation` .
`mysql.rds_set_gsh_rotation`	Menentukan interval, dalam hari, antara rotasi tabel. Nilai defaultnya adalah 7.
`mysql.rds_disable_gsh_rotation`	Menonaktifkan rotasi tabel.
`mysql.rds_rotate_global_status_history`	Merotasi isi tabel `mysql.rds_global_status_history` ke `mysql.rds_global_status_history_old` sesuai permintaan.

Saat GoSH berjalan, Anda dapat mengkueri tabel tujuan penulisannya. Misalnya, untuk mengkueri rasio hit kumpulan buffer Innodb, Anda dapat menjalankan kueri berikut:

```
select a.collection_end, a.collection_start, (( a.variable_Delta-b.variable_delta)/
a.variable_delta)*100 as "HitRatio"
    from mysql.rds_global_status_history as a join mysql.rds_global_status_history as b
 on a.collection_end = b.collection_end
    where a. variable_name = 'Innodb_buffer_pool_read_requests' and b.variable_name =
 'Innodb_buffer_pool_reads'
```

Zona waktu lokal untuk instans DB MySQL

Secara bawaan, zona waktu untuk instans DB MySQL adalah Waktu Universal Terkoordinasi (UTC). Anda dapat mengatur zona waktu untuk instans DB Anda ke zona waktu lokal untuk aplikasi Anda.

Untuk mengatur zona waktu lokal untuk instans DB, atur parameter `time_zone` di grup parameter untuk instans DB Anda ke salah satu nilai yang didukung yang dicantumkan nanti di bagian ini. Saat Anda mengatur parameter `time_zone` untuk grup parameter, semua instans DB dan replika baca yang menggunakan grup parameter tersebut berubah untuk menggunakan zona waktu lokal baru. Untuk informasi tentang pengaturan parameter di grup parameter, lihat Bekerja dengan grup parameter.

Setelah Anda mengatur zona waktu lokal, semua koneksi baru ke basis data mencerminkan perubahan tersebut. Jika Anda memiliki koneksi terbuka ke basis data Anda ketika mengubah zona waktu lokal, Anda tidak akan melihat pembaruan zona waktu lokal sampai Anda menutup koneksi dan membuka koneksi baru.

Anda dapat mengatur zona waktu lokal yang berbeda untuk instans DB dan satu atau beberapa replika baca. Untuk melakukannya, gunakan grup parameter yang berbeda untuk instans DB dan replika dan atur parameter `time_zone` di dalam setiap grup parameter ke zona waktu lokal yang berbeda.

Jika Anda mereplikasi di seluruh Wilayah AWS, instans DB sumber dan replika baca menggunakan grup parameter yang berbeda (grup parameter tidak ada yang menyamai di suatu Wilayah AWS). Untuk menggunakan zona waktu lokal yang sama untuk setiap instans, Anda harus mengatur parameter `time_zone` di dalam grup parameter instans dan replika baca.

Saat Anda memulihkan instans DB dari snapshot DB, zona waktu lokal diatur menjadi UTC. Anda dapat memperbarui zona waktu ke zona waktu lokal Anda setelah pemulihan selesai. Jika Anda memulihkan instans DB ke titik waktu tertentu, zona waktu lokal untuk instans DB yang dipulihkan adalah pengaturan zona waktu dari grup parameter instans DB yang dipulihkan.

Internet Assigned Numbers Authority (IANA) menerbitkan zona waktu baru di https://www.iana.org/time-zones beberapa kali dalam setahun. Setiap kali RDS menerbitkan rilis pemeliharaan minor baru MySQL, rilis tersebut akan dikirimkan beserta data zona waktu terbaru pada saat rilis. Jika Anda menggunakan versi RDS for MySQL terbaru, Anda akan memiliki data zona waktu terbaru dari RDS. Untuk memastikan bahwa instans basis data Anda memiliki data zona waktu terbaru, sebaiknya mutakhirkan ke versi mesin basis data yang lebih tinggi. Atau, Anda dapat mengubah secara manual

tabel zona waktu dalam instans basis data MariaDB. Untuk itu, Anda dapat menggunakan perintah-perintah SQL atau menjalankan alat mysql_tzinfo_to_sql di klien SQL. Setelah memperbarui data zona waktu secara manual, hidupkan ulang instans basis data Anda sehingga perubahan berlaku. RDS tidak mengubah atau mengatur ulang data zona waktu instans basis data yang sedang berjalan. Data zona waktu baru diinstal hanya ketika Anda melakukan pemutakhiran versi mesin basis data.

Anda dapat mengatur zona waktu lokal Anda menjadi salah satu dari nilai-nilai berikut ini.

`Africa/Cairo`	`Asia/Riyadh`
`Africa/Casablanca`	`Asia/Seoul`
`Africa/Harare`	`Asia/Shanghai`
`Africa/Monrovia`	`Asia/Singapore`
`Africa/Nairobi`	`Asia/Taipei`
`Africa/Tripoli`	`Asia/Tehran`
`Africa/Windhoek`	`Asia/Tokyo`
`America/Araguaina`	`Asia/Ulaanbaatar`
`America/Asuncion`	`Asia/Vladivostok`
`America/Bogota`	`Asia/Yakutsk`
`America/Buenos_Aires`	`Asia/Yerevan`
`America/Caracas`	`Atlantic/Azores`
`America/Chihuahua`	`Australia/Adelaide`
`America/Cuiaba`	`Australia/Brisbane`
`America/Denver`	`Australia/Darwin`
`America/Fortaleza`	`Australia/Hobart`
`America/Guatemala`	`Australia/Perth`

America/Halifax	Australia/Sydney
America/Manaus	Brazil/East
America/Matamoros	Canada/Newfoundland
America/Monterrey	Canada/Saskatchewan
America/Montevideo	Canada/Yukon
America/Phoenix	Europe/Amsterdam
America/Santiago	Europe/Athens
America/Tijuana	Europe/Dublin
Asia/Amman	Europe/Helsinki
Asia/Ashgabat	Europe/Istanbul
Asia/Baghdad	Europe/Kaliningrad
Asia/Baku	Europe/Moscow
Asia/Bangkok	Europe/Paris
Asia/Beirut	Europe/Prague
Asia/Calcutta	Europe/Sarajevo
Asia/Damascus	Pacific/Auckland
Asia/Dhaka	Pacific/Fiji
Asia/Irkutsk	Pacific/Guam
Asia/Jerusalem	Pacific/Honolulu
Asia/Kabul	Pacific/Samoa
Asia/Karachi	US/Alaska

Asia/Kathmandu	US/Central
Asia/Krasnoyarsk	US/Eastern
Asia/Magadan	US/East-Indiana
Asia/Muscat	US/Pacific
Asia/Novosibirsk	UTC

Masalah umum dan batasan untuk Amazon RDS for MySQL

Berikut ini adalah masalah umum dan batasan untuk menggunakan Amazon RDS for MySQL.

Topik

Kata yang dicadangkan InnoDB

`InnoDB` adalah kata yang dicadangkan untuk RDS for MySQL. Anda tidak dapat menggunakan nama ini untuk basis data MySQL.

Perilaku penyimpanan penuh untuk Amazon RDS for MySQL

Ketika penyimpanan menjadi penuh untuk instans DB MySQL, terkadang akan ada inkonsistensi metadata, ketidaksesuaian kamus, dan tabel orphan. Untuk mencegah masalah ini, Amazon RDS secara otomatis menghentikan instans DB yang mencapai status `storage-full`.

Instans DB MySQL mencapai status `storage-full` dalam kasus berikut:

- Instans DB memiliki penyimpanan kurang dari 20.000 MiB, dan penyimpanan yang tersedia mencapai 200 MiB atau kurang.
- Instans DB memiliki penyimpanan lebih dari 102.400 MiB, dan penyimpanan yang tersedia mencapai 1024 MiB atau kurang.

- Instans DB memiliki penyimpanan antara 20.000 MiB dan 102.400 MiB, dan memiliki kurang dari 1% dari penyimpanan yang tersedia.

Setelah Amazon RDS menghentikan instans DB secara otomatis karena mencapai status `storage-full`, Anda masih dapat memodifikasinya. Untuk memulai ulang instans DB, selesaikan setidaknya salah satu hal berikut:

- Modifikasi instans DB untuk mengaktifkan penskalaan otomatis penyimpanan.

  Untuk informasi selengkapnya tentang penskalaan otomatis penyimpanan, lihat Mengelola kapasitas secara otomatis dengan penskalaan otomatis penyimpanan Amazon RDS.
- Modifikasi instans DB untuk meningkatkan kapasitas penyimpanan.

  Untuk informasi selengkapnya tentang peningkatan kapasitas penyimpanan, lihat Meningkatkan kapasitas penyimpanan instans DB.

Setelah Anda melakukan salah satu perubahan ini, instans DB dimulai ulang secara otomatis. Untuk informasi tentang memodifikasi instans DB, lihat Memodifikasi instans DB Amazon RDS.

Ukuran pool buffer InnoDB tidak konsisten

Untuk MySQL 5.7, saat ini terdapat bug dalam cara pengelolaan ukuran pool buffer. MySQL 5.7 mungkin menyesuaikan nilai parameter `innodb_buffer_pool_size` ke nilai yang besar yang dapat mengakibatkan pool buffer InnoDB tumbuh terlalu besar dan menggunakan terlalu banyak memori. Efek ini dapat menyebabkan mesin basis data MySQL berhenti berjalan atau dapat mencegahnya untuk menyala. Masalah ini lebih umum untuk kelas instans DB yang memiliki memori lebih sedikit.

Untuk mengatasi masalah ini, tetapkan nilai parameter `innodb_buffer_pool_size` ke beberapa produk nilai parameter `innodb_buffer_pool_instances` dan nilai parameter `innodb_buffer_pool_chunk_size`. Misalnya, Anda dapat mengatur nilai parameter `innodb_buffer_pool_size` ke beberapa kali delapan kali lipat produk nilai parameter `innodb_buffer_pool_instances` dan `innodb_buffer_pool_chunk_size`, seperti yang ditunjukkan dalam contoh berikut.

```
innodb_buffer_pool_chunk_size = 536870912
innodb_buffer_pool_instances = 4
innodb_buffer_pool_size = (536870912 * 4) * 8 = 17179869184
```

Untuk detail tentang bug MySQL 5.7, lihat https://bugs.mysql.com/bug.php?id=79379 dalam dokumentasi MySQL.

Pengoptimalan penggabungan indeks memberikan hasil yang salah

Kueri yang menggunakan pengoptimalan penggabungan indeks mungkin memberikan hasil yang salah karena bug di pengoptimal kueri MySQL yang diperkenalkan di MySQL 5.5.37. Saat Anda mengeluarkan kueri terhadap tabel dengan beberapa indeks, pengoptimal memindai rentang baris berdasarkan beberapa indeks, tetapi tidak menggabungkan hasil bersama-sama dengan benar. Untuk informasi selengkapnya tentang bug pengoptimal kueri, lihat http://bugs.mysql.com/bug.php?id=72745 dan http://bugs.mysql.com/bug.php?id=68194 di basis data bug MySQL.

Misalnya, pertimbangkan kueri pada tabel dengan dua indeks di mana argumen pencarian mereferensikan kolom yang diindeks.

```
SELECT * FROM table1
WHERE indexed_col1 = 'value1' AND indexed_col2 = 'value2';
```

Dalam kasus ini, mesin pencari akan mencari kedua indeks. Namun, karena adanya bug, hasil gabungan salah.

Untuk mengatasi masalah ini, Anda dapat melakukan salah satu dari yang berikut:

- Tetapkan parameter `optimizer_switch` ke `index_merge=off` di grup parameter DB untuk instans DB MySQL Anda. Untuk informasi tentang pengaturan parameter grup parameter DB, lihat Bekerja dengan grup parameter.
- Tingkatkan instans DB MySQL Anda ke MySQL versi 5.7 atau 8.0. Untuk informasi selengkapnya, lihat Meng-upgrade mesin DB MySQL.
- Jika Anda tidak dapat meningkatkan instans Anda atau mengubah parameter `optimizer_switch`, Anda dapat mengatasi bug dengan mengidentifikasi indeks untuk kueri secara eksplisit, misalnya:

  ```
  SELECT * FROM table1
  USE INDEX covering_index
  WHERE indexed_col1 = 'value1' AND indexed_col2 = 'value2';
  ```

Untuk informasi selengkapnya, lihat Index merge optimization dalam dokumentasi MySQL.

Pengecualian parameter MySQL untuk instans DB Amazon RDS

Beberapa parameter MySQL memerlukan pertimbangan khusus saat digunakan dengan instans DB Amazon RDS.

lower_case_table_names

Karena Amazon RDS menggunakan sistem file yang peka huruf besar/kecil, pengaturan nilai parameter server `lower_case_table_names` menjadi 2 (nama disimpan seperti yang diberikan tetapi dibandingkan dalam huruf kecil) tidak didukung. Berikut adalah nilai yang didukung untuk instans DB Amazon RDS for MySQL:

- 0 (nama yang disimpan seperti yang diberikan dan perbandingan peka terhadap huruf besar-kecil) didukung untuk semua versi RDS for MySQL.
- 1 (nama yang disimpan dalam huruf kecil dan perbandingan tidak peka terhadap huruf besar-kecil) didukung untuk RDS for MySQL versi 5.7 dan versi 8.0.28 dan versi 8.0 yang lebih tinggi.

Atur parameter `lower_case_table_names` di dalam grup parameter DB kustom sebelum membuat instans DB. Kemudian, tentukan grup parameter DB kustom ketika Anda membuat instans DB.

Ketika grup parameter dikaitkan dengan instans DB MySQL dengan versi di bawah 8.0, kami sarankan Anda tidak mengubah parameter `lower_case_table_names` di dalam grup parameter. Mengubahnya dapat menyebabkan ketidakkonsistenan dengan cadangan point-in-time pemulihan dan membaca instance replika DB.

Ketika sebuah grup parameter dikaitkan dengan instans DB MySQL versi 8.0, Anda tidak dapat mengubah parameter `lower_case_table_names` di dalam grup parameter tersebut.

Replika baca harus selalu menggunakan nilai parameter `lower_case_table_names` yang sama dengan instans DB sumber.

long_query_time

Anda dapat menyetel parameter `long_query_time` ke nilai titik mengambang sehingga Anda dapat mencatatkan kueri lambat ke log kueri lambat MySQL dengan resolusi mikrodetik. Anda dapat mengatur nilai seperti 0,1 detik, yang akan menjadi 100 milidetik, untuk memudahkan debugging transaksi lambat yang membutuhkan waktu kurang dari satu detik.

Batas ukuran file MySQL di Amazon RDS

Untuk instance MySQL DB, batas penyimpanan maksimum yang disediakan membatasi ukuran tabel hingga ukuran maksimum 16 TB saat menggunakan ruang tabel InnoDB. file-per-table Batasan ini juga membatasi ruang tabel sistem hingga ukuran maksimum sebesar 16 TB. file-per-table Ruang tabel InnoDB (dengan tabel masing-masing di tablespace mereka sendiri) diatur secara default untuk instance MySQL DB.

> **Note**
>
> Beberapa instans DB yang ada memiliki batas yang lebih rendah. Misalnya, instans DB MySQL yang dibuat sebelum April 2014 memiliki batas ukuran file dan tabel 2 TB. Batas ukuran file 2 TB ini juga berlaku untuk instans DB atau replika baca yang dibuat dari snapshot DB yang diambil sebelum April 2014, terlepas dari kapan instans DB dibuat.

Ada keuntungan dan kerugian untuk menggunakan InnoDB file-per-table tablespaces, tergantung pada aplikasi Anda. Untuk menentukan pendekatan terbaik untuk aplikasi Anda, lihat F ile-per-table tablespace dalam dokumentasi MySQL.

Kami tidak menyarankan Anda membiarkan tabel berkembang hingga ukuran file maksimum. Secara umum, praktik yang lebih baik adalah membagi data menjadi tabel yang lebih kecil, yang dapat meningkatkan waktu performa dan pemulihan.

Salah satu opsi yang dapat Anda gunakan untuk memecah tabel ke dalam tabel yang lebih kecil adalah partisi. Partisi mendistribusikan porsi tabel besar ke dalam file terpisah berdasarkan aturan yang Anda tentukan. Misalnya, jika menyimpan transaksi berdasarkan tanggal, Anda dapat membuat aturan partisi yang mendistribusikan transaksi lama ke dalam file terpisah menggunakan partisi. Kemudian, Anda secara berkala dapat mengarsipkan data transaksi historis yang tidak diperlukan aplikasi Anda. Untuk informasi selengkapnya, lihat Partitioning dalam dokumentasi MySQL.

Karena tidak ada tabel sistem atau tampilan tunggal yang menyediakan ukuran semua tabel dan ruang tabel sistem InnoDB, Anda harus mengueri beberapa tabel untuk menentukan ukuran ruang tabelnya.

Untuk menentukan ukuran ruang tabel sistem InnoDB dan ruang tabel kamus data

- Gunakan perintah SQL berikut untuk menentukan apakah ada ruang tabel Anda yang terlalu besar dan merupakan kandidat untuk partisi.

> **Note**
>
> Ruang tabel kamus data ditujukan khusus untuk MySQL 8.0.

```
select FILE_NAME,TABLESPACE_NAME, ROUND(((TOTAL_EXTENTS*EXTENT_SIZE)
/1024/1024/1024), 2)  as "File Size (GB)" from information_schema.FILES
where tablespace_name in ('mysql','innodb_system');
```

Untuk menentukan ukuran tabel pengguna InnoDB di luar ruang tabel sistem InnoDB (untuk versi MySQL 5.7)

- Gunakan perintah SQL berikut untuk menentukan apakah salah satu tabel Anda terlalu besar dan merupakan kandidat untuk partisi.

```
SELECT SPACE,NAME,ROUND((ALLOCATED_SIZE/1024/1024/1024), 2)
as "Tablespace Size (GB)"
FROM information_schema.INNODB_SYS_TABLESPACES ORDER BY 3 DESC;
```

Untuk menentukan ukuran tabel pengguna InnoDB di luar ruang tabel sistem InnoDB (untuk versi MySQL 8.0)

- Gunakan perintah SQL berikut untuk menentukan apakah salah satu tabel Anda terlalu besar dan merupakan kandidat untuk partisi.

```
SELECT SPACE,NAME,ROUND((ALLOCATED_SIZE/1024/1024/1024), 2)
as "Tablespace Size (GB)"
FROM information_schema.INNODB_TABLESPACES ORDER BY 3 DESC;
```

Untuk menentukan ukuran tabel pengguna non-InnoDB

- Gunakan perintah SQL berikut untuk menentukan apakah ada tabel pengguna non-InnoDB Anda yang terlalu besar.

```
SELECT TABLE_SCHEMA, TABLE_NAME, round(((DATA_LENGTH + INDEX_LENGTH+DATA_FREE)
/ 1024 / 1024/ 1024), 2) As "Approximate size (GB)" FROM information_schema.TABLES
```

```
WHERE TABLE_SCHEMA NOT IN ('mysql', 'information_schema', 'performance_schema')
and ENGINE<>'InnoDB';
```

Untuk mengaktifkan ruang meja file-per-table InnoDB

- Setel parameter innodb_file_per_table ke 1 di dalam grup parameter untuk instans DB.

Untuk menonaktifkan ruang meja file-per-table InnoDB

- Setel parameter innodb_file_per_table ke `0` di dalam grup parameter untuk instans DB.

Untuk informasi tentang pembaruan grup parameter, lihat Bekerja dengan grup parameter.

Bila Anda telah mengaktifkan atau menonaktifkan file-per-table ruang tabel InnoDB, Anda dapat mengeluarkan `ALTER TABLE` perintah untuk memindahkan tabel dari tablespace global ke tablespace sendiri, atau dari tablespace sendiri ke tablespace global seperti yang ditunjukkan pada contoh berikut:

```
ALTER TABLE table_name ENGINE=InnoDB;
```

Plugin Keyring MySQL tidak didukung

Saat ini, Amazon RDS for MySQL tidak mendukung Plugin Keyring Amazon Web Services `keyring_aws` MySQL.

Port kustom

Amazon RDS memblokir koneksi ke port kustom 33060 untuk mesin MySQL. Pilih port yang berbeda untuk mesin MySQL Anda.

Batasan prosedur tersimpan MySQL

Prosedur tersimpan mysql.rds_kill dan mysql.rds_kill_query tidak dapat mengakhiri sesi atau kueri yang dimiliki oleh pengguna MySQL dengan nama pengguna lebih dari 16 karakter pada versi RDS for MySQL berikut:

- 8.0.32 dan versi 8 yang lebih rendah
- 5.7.41 dan versi 5.7 yang lebih rendah

Replikasi berbasis GTID dengan instans sumber eksternal

Amazon RDS tidak mendukung replikasi berdasarkan pengidentifikasi transaksi global (GTID) dari instans MySQL eksternal ke instans DB Amazon RDS for MySQL yang memerlukan pengaturan GTID_PURGED selama konfigurasi.

Plugin otentikasi default MySQL

RDS untuk MySQL versi 8.0.34 dan yang lebih tinggi menggunakan plugin. `mysql_native_password` Anda tidak dapat mengubah pengaturan `default_authentication_plugin`.

RDS for MySQL

Topik ini menjelaskan prosedur tersimpan sistem yang tersedia untuk instans Amazon RDS yang menjalankan mesin DB MySQL. Pengguna utama harus menjalankan prosedur ini.

Topik

Melakukan konfigurasi

Prosedur tersimpan berikut mengatur dan menampilkan parameter konfigurasi, seperti untuk retensi file log biner.

Topik

- mysql.rds_set_configuration
- mysql.rds_show_configuration

mysql.rds_set_configuration

Menentukan jumlah jam untuk mempertahankan log biner atau jumlah detik untuk menunda replikasi.

Sintaksis

```
CALL mysql.rds_set_configuration(name,value);
```

Parameter

name

Nama parameter konfigurasi yang akan diatur.

value

Nilai parameter konfigurasi.

Catatan penggunaan

Prosedur `mysql.rds_set_configuration` mendukung parameter konfigurasi berikut:

- binlog retention hours
- source delay
- target delay

Parameter konfigurasi disimpan secara permanen dan bertahan dari reboot atau failover instans DB apa pun.

binlog retention hours

Parameter `binlog retention hours` digunakan untuk menentukan jumlah jam untuk mempertahankan file log biner. Amazon RDS biasanya membersihkan log biner sesegera mungkin, tetapi log biner mungkin masih diperlukan untuk replikasi dengan basis data MySQL di luar RDS.

Nilai default `binlog retention hours` adalah `NULL`. Untuk RDS for MySQL, `NULL` menandakan bahwa log biner tidak dipertahankan (0 jam).

Untuk menentukan jumlah jam guna mempertahankan log biner pada instans DB, gunakan prosedur tersimpan `mysql.rds_set_configuration` dan tentukan periode dengan waktu yang cukup untuk terjadinya proses replikasi, seperti yang diperlihatkan dalam contoh berikut.

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

> **Note**
>
> Anda tidak dapat menggunakan nilai `0` untuk `binlog retention hours`.

Untuk instans DB MySQL, nilai `binlog retention hours` maksimumnya adalah 168 (7 hari).

Setelah Anda mengatur periode retensi, pantau penggunaan penyimpanan untuk instans DB guna memastikan bahwa log biner yang dipertahankan tidak memakan terlalu banyak ruang penyimpanan.

source delay

Gunakan parameter `source delay` dalam replika baca untuk menentukan jumlah detik untuk menunda replikasi dari replika baca ke instans DB sumbernya. Amazon RDS biasanya mereplikasi perubahan sesegera mungkin, tetapi Anda mungkin ingin menunda replikasi di beberapa lingkungan. Misalnya, saat replikasi tertunda, Anda dapat menggulirkan replika baca tertunda ke waktu sebelum bencana terjadi. Jika tabel jatuh secara tidak sengaja, Anda dapat menggunakan replikasi tertunda untuk memulihkannya dengan cepat. Nilai default `target delay` adalah `0` (tidak menunda replikasi).

Saat digunakan, parameter ini menjalankan mysql.rds_set_source_delay dan menerapkan CHANGE primary TO MASTER_DELAY = nilai input. Jika berhasil, prosedur menyimpan parameter `source delay` ke tabel `mysql.rds_configuration`.

Untuk menentukan jumlah detik bagi Amazon RDS untuk menunda replikasi ke instans DB sumber, gunakan prosedur tersimpan `mysql.rds_set_configuration` dan tentukan jumlah detik untuk menunda replikasi. Dalam contoh berikut, replikasi tertunda setidaknya satu jam (3.600 detik).

```
call mysql.rds_set_configuration('source delay', 3600);
```

Lalu, prosedur menjalankan `mysql.rds_set_source_delay(3600)`.

Batas untuk parameter `source delay` adalah satu hari (86.400 detik).

> **Note**
>
> Parameter `source delay` tidak didukung untuk RDS for MySQL versi 8.0 atau MariaDB versi di bawah 10.2.

target delay

Gunakan parameter `target delay` untuk menentukan jumlah detik guna menunda replikasi antara instans DB dan replika baca yang dikelola RDS mendatang yang dibuat dari instans ini. Parameter ini diabaikan untuk replika baca yang tidak dikelola RDS. Amazon RDS biasanya mereplikasi perubahan sesegera mungkin, tetapi Anda mungkin ingin menunda replikasi di beberapa lingkungan. Misalnya, saat replikasi tertunda, Anda dapat menggulirkan replika baca tertunda ke waktu sebelum bencana terjadi. Jika tabel jatuh secara tidak sengaja, Anda dapat menggunakan replikasi tertunda untuk memulihkannya dengan cepat. Nilai default `target delay` adalah `0` (tidak menunda replikasi).

Untuk pemulihan bencana, Anda dapat menggunakan parameter konfigurasi ini dengan prosedur tersimpan mysql.rds_start_replication_until atau mysql.rds_start_replication_until_gtid. Untuk meneruskan perubahan ke replika baca yang tertunda ke waktu sebelum bencana, Anda dapat menjalankan prosedur `mysql.rds_set_configuration` dengan pengaturan parameter ini. Setelah prosedur `mysql.rds_start_replication_until` atau `mysql.rds_start_replication_until_gtid` menghentikan replikasi, Anda dapat mempromosikan replika baca menjadi instans DB primer baru dengan mengikuti petunjuk di Mempromosikan replika baca menjadi instans DB mandiri.

Untuk menggunakan prosedur `mysql.rds_rds_start_replication_until_gtid`, replikasi berbasis GTID harus diaktifkan. Untuk melewati transaksi berbasis GTID tertentu yang diketahui menyebabkan bencana, Anda dapat menggunakan prosedur tersimpan mysql.rds_skip_transaction_with_gtid. Untuk informasi selengkapnya tentang cara menggunakan replikasi berbasis GTID, lihat Menggunakan replikasi berbasis GTID untuk Amazon RDS for MySQL.

Untuk menentukan jumlah detik bagi Amazon RDS untuk menunda replikasi ke replika baca, gunakan prosedur tersimpan `mysql.rds_set_configuration` dan tentukan jumlah detik untuk menunda replikasi. Contoh berikut menunjukkan bahwa replikasi tertunda setidaknya satu jam (3.600 detik).

```
call mysql.rds_set_configuration('target delay', 3600);
```

Batas untuk parameter `target delay` adalah satu hari (86.400 detik).

> **Note**
>
> Parameter `target delay` tidak didukung untuk RDS for MySQL versi 8.0 atau MariaDB versi sebelum 10.2.

mysql.rds_show_configuration

Jumlah jam untuk mempertahankan log biner.

Sintaksis

```
CALL mysql.rds_show_configuration;
```

Catatan penggunaan

Untuk memverifikasi jumlah jam Amazon RDS mempertahankan log biner, gunakan prosedur tersimpan `mysql.rds_show_configuration`.

Contoh

Contoh berikut menampilkan periode retensi:

```
call mysql.rds_show_configuration;
                name                          value     description
                binlog retention hours        24        binlog retention hours specifies
 the duration in hours before binary logs are automatically deleted.
```

Mengakhiri sesi atau kueri

Prosedur tersimpan berikut mengakhiri sesi atau kueri.

Topik

- mysql.rds_kill
- mysql.rds_kill_query

mysql.rds_kill

Mengakhiri koneksi ke server MySQL.

Sintaksis

```
CALL mysql.rds_kill(processID);
```

Parameter

processID

Identitas utas koneksi akan diakhiri.

Catatan penggunaan

Setiap koneksi ke server MySQL berjalan di utas terpisah. Untuk mengakhiri koneksi, gunakan prosedur `mysql.rds_kill` dan teruskan ID utas koneksi tersebut. Untuk mendapatkan ID utas, gunakan perintah SHOW PROCESSLIST MySQL.

Untuk informasi tentang batasan, lihat Batasan prosedur tersimpan MySQL.

Contoh

Contoh berikut mengakhiri koneksi dengan ID utas 4243:

```
CALL mysql.rds_kill(4243);
```

mysql.rds_kill_query

Mengakhiri kueri yang berjalan pada server MySQL.

Sintaksis

```
CALL mysql.rds_kill_query(processID);
```

Parameter

processID

Identitas proses atau utas yang menjalankan kueri yang akan diakhiri.

Catatan penggunaan

Untuk mengakhiri kueri yang berjalan pada server MySQL, gunakan prosedur `mysql_rds_kill_query` dan teruskan ID koneksi utas yang menjalankan kueri. Lalu, prosedur akan mengakhiri koneksi.

Untuk mendapatkan ID, lakukan kueri pada tabel INFORMATION_SCHEMA PROCESSLIST MySQL atau gunakan perintah SHOW PROCESSLIST MySQL. Nilai dalam kolom ID dari `SHOW PROCESSLIST` atau `SELECT * FROM INFORMATION_SCHEMA.PROCESSLIST` adalah *processID*.

Untuk informasi tentang batasan, lihat Batasan prosedur tersimpan MySQL.

Contoh

Contoh berikut mengakhiri kueri dengan ID utas kueri 230040:

```
CALL mysql.rds_kill_query(230040);
```

Pencatatan log

Prosedur tersimpan berikut merotasi log MySQL ke tabel cadangan. Untuk informasi selengkapnya, lihat File log basis data MySQL.

Topik

mysql.rds_rotate_general_log

Merotasi tabel `mysql.general_log` ke tabel cadangan.

Sintaksis

```
CALL mysql.rds_rotate_general_log;
```

Catatan penggunaan

Anda bisa merotasi tabel `mysql.general_log` ke tabel cadangan dengan memanggil prosedur `mysql.rds_rotate_general_log`. Saat tabel log dirotasi, tabel log saat ini disalin ke tabel log cadangan dan entri di tabel log saat ini dihapus. Jika sudah ada, tabel log cadangan akan dihapus sebelum tabel log saat ini disalin ke cadangan. Anda dapat meminta tabel log cadangan jika diperlukan. Tabel log cadangan untuk tabel `mysql.general_log` bernama `mysql.general_log_backup`.

Anda dapat menjalankan prosedur ini hanya ketika parameter `log_output` diatur ke `TABLE`.

mysql.rds_rotate_slow_log

Merotasi tabel `mysql.slow_log` ke tabel cadangan.

Sintaksis

```
CALL mysql.rds_rotate_slow_log;
```

Catatan penggunaan

Anda bisa merotasi tabel `mysql.slow_log` ke tabel cadangan dengan memanggil prosedur `mysql.rds_rotate_slow_log`. Saat tabel log dirotasi, tabel log saat ini disalin ke tabel log cadangan dan entri di tabel log saat ini dihapus. Jika sudah ada, tabel log cadangan akan dihapus sebelum tabel log saat ini disalin ke cadangan.

Anda dapat meminta tabel log cadangan jika diperlukan. Tabel log cadangan untuk tabel `mysql.slow_log` bernama `mysql.slow_log_backup`.

Mengelola klaster aktif-aktif

Prosedur tersimpan berikut menyiapkan dan mengelola klaster aktif-aktif RDS for MySQL. Untuk informasi selengkapnya, lihat the section called “Mengonfigurasikan klaster aktif-aktif”.

Prosedur tersimpan ini hanya tersedia dengan RDS untuk instans MySQL DB yang menjalankan versi 8.0.35 dan versi minor yang lebih tinggi.

Topik

- mysql.rds_group_replication_advance_gtid
- mysql.rds_group_replication_create_user
- mysql.rds_group_replication_set_recovery_channel
- mysql.rds_group_replication_start
- mysql.rds_group_replication_stop

mysql.rds_group_replication_advance_gtid

Membuat GTID placeholder pada instans DB saat ini.

Sintaks

```
CALL mysql.rds_group_replication_advance_gtid(
begin_id
, end_id
, server_uuid
);
```

Parameter

begin_id

ID transaksi awal yang akan dibuat.

end_id

ID transaksi akhir yang akan dibuat.

begin_id

`group_replication_group_name` untuk transaksi yang akan dibuat. `group_replication_group_name` ditetapkan sebagai UUID dalam grup parameter DB yang terkait dengan instans DB.

Catatan penggunaan

Dalam klaster aktif-aktif, supaya instans DB dapat digabungkan ke dalam grup, semua transaksi GTID yang dieksekusi pada instans DB baru harus ada pada anggota lain dalam klaster. Dalam kasus tertentu, instans DB baru mungkin memiliki lebih banyak transaksi ketika transaksi dieksekusi sebelum menggabungkan instans ke grup. Dalam kasus ini, Anda tidak dapat menghapus transaksi yang sudah ada, tetapi Anda dapat menggunakan prosedur ini untuk membuat GTID placeholder yang sesuai pada instans DB lain dalam grup. Sebelum itu, pastikan transaksi tidak memengaruhi data yang direplikasi.

Saat Anda memanggil prosedur ini, transaksi GTID `server_uuid:begin_id-end_id` dibuat dengan konten kosong. Untuk menghindari masalah replikasi, jangan gunakan prosedur ini dalam kondisi lain.

> **Important**
>
> Jangan panggil prosedur ini ketika klaster aktif-aktif berfungsi normal. Jangan panggil prosedur ini kecuali Anda memahami kemungkinan konsekuensi dari transaksi yang Anda buat. Memanggil prosedur ini dapat menyebabkan inkonsistensi data.

Contoh

Contoh berikut membuat GTID placeholder pada instans DB saat ini:

```
CALL mysql.rds_group_replication_advance_gtid(5, 6,
 '11111111-2222-3333-4444-555555555555');
```

mysql.rds_group_replication_create_user

Membuat pengguna replikasi `rdsgrprepladmin` untuk replikasi grup pada instans DB.

Sintaks

```
CALL mysql.rds_group_replication_create_user(
replication_user_password
);
```

Parameter

replication_user_password

Kata sandi pengguna replikasi `rdsgrprepladmin`.

Catatan penggunaan

- Kata sandi pengguna replikasi `rdsgrprepladmin` harus sama di semua instans DB dalam klaster aktif-aktif.
- Nama pengguna `rdsgrprepladmin` dipesan untuk koneksi replikasi grup. Tidak ada pengguna lain, termasuk pengguna master, yang dapat memiliki nama pengguna ini.

Contoh

Contoh berikut membuat pengguna replikasi `rdsgrprepladmin` untuk replikasi grup pada instans DB:

```
CALL mysql.rds_group_replication_create_user('password');
```

mysql.rds_group_replication_set_recovery_channel

Menetapkan saluran `group_replication_recovery` untuk klaster aktif-aktif. Prosedur ini menggunakan pengguna `rdsgrprepladmin` terpesan untuk mengonfigurasi saluran.

Sintaks

```
CALL mysql.rds_group_replication_set_recovery_channel(
replication_user_password);
```

Parameter

replication_user_password

Kata sandi pengguna replikasi `rdsgrprepladmin`.

Catatan penggunaan

Kata sandi pengguna replikasi `rdsgrprepladmin` harus sama di semua instans DB dalam klaster aktif-aktif. Panggilan ke `mysql.rds_group_replication_create_user` menentukan kata sandi.

Contoh

Contoh berikut menetapkan saluran `group_replication_recovery` untuk klaster aktif-aktif:

```
CALL mysql.rds_group_replication_set_recovery_channel('password');
```

mysql.rds_group_replication_start

Memulai replikasi grup pada instans DB saat ini.

Sintaks

```
CALL mysql.rds_group_replication_start(
bootstrap
);
```

Parameter

bootstrap

Nilai yang menentukan apakah akan menginisialisasi grup baru atau bergabung dengan grup yang sudah ada. 1 menginisialisasi grup baru dengan instans DB saat ini. `0` menggabungkan instans DB saat ini ke grup yang sudah ada dengan menghubungkan ke titik akhir yang ditentukan dalam parameter `group_replication_group_seeds` dalam grup parameter DB yang terkait dengan instans DB.

Contoh

Contoh berikut menginisialisasi grup baru dengan instans DB saat ini:

```
CALL mysql.rds_group_replication_start(1);
```

mysql.rds_group_replication_stop

Menghentikan replikasi grup pada instans DB saat ini.

Sintaks

```
CALL mysql.rds_group_replication_stop();
```

Catatan penggunaan

Saat Anda menghentikan replikasi pada instans DB, tindakan tersebut tidak memengaruhi instans DB lain dalam klaster aktif-aktif.

Mengelola replikasi multi-sumber

Prosedur tersimpan berikut mengatur dan mengelola saluran replikasi pada RDS untuk replika multi-sumber MySQL. Untuk informasi selengkapnya, lihat the section called “Mengkonfigurasi replikasi multi-sumber”.

Prosedur tersimpan ini hanya tersedia dengan RDS untuk instance MySQL DB yang menjalankan versi mesin berikut:

- 8.0.35 dan versi minor yang lebih tinggi
- 5.7.44 dan versi minor yang lebih tinggi

> **Note**
>
> Meskipun dokumentasi ini mengacu pada instance DB sumber sebagai RDS untuk instance MySQL DB, prosedur ini juga berfungsi untuk instance MySQL yang berjalan di luar Amazon RDS.

Topik

mysql.rds_next_source_log_for_channel

Mengubah posisi log instans DB sumber ke awal log biner berikutnya pada instans DB sumber untuk saluran. Gunakan prosedur ini hanya jika Anda menerima replikasi I/O kesalahan 1236 pada replika multi-sumber.

Sintaksis

```
CALL mysql.rds_next_source_log_for_channel(
curr_master_log,
channel_name
);
```

Parameter

curr_master_log

Indeks file log sumber saat ini. Misalnya, jika file saat ini bernama `mysql-bin-changelog.012345`, maka indeksnya adalah 12345. Untuk menentukan nama file log sumber saat ini, jalankan perintah `SHOW REPLICA STATUS FOR CHANNEL '`*`channel_name`*`'` dan lihat kolom `Source_Log_File`.

> **Note**
>
> Versi sebelumnya dari MySQL menggunakan `SHOW SLAVE STATUS` bukan `SHOW REPLICA STATUS`. Jika Anda menggunakan MySQL versi sebelum 8.0.23, gunakan `SHOW SLAVE STATUS`.

channel_name

Nama saluran replikasi pada replika multi-sumber. Setiap saluran replikasi menerima peristiwa log biner dari RDS sumber tunggal untuk instance MySQL DB yang berjalan pada host dan port tertentu.

Catatan penggunaan

Pengguna master harus menjalankan prosedur `mysql.rds_next_source_log_for_channel`. Jika ada kesalahan IO_thread, misalnya, Anda dapat menggunakan prosedur ini untuk melewati

semua peristiwa dalam file log biner saat ini dan melanjutkan replikasi dari file log biner berikutnya untuk saluran yang ditentukan dalam. `channel_name`

Contoh

Asumsikan replikasi gagal pada saluran pada replika multi-sumber. Berjalan `SHOW REPLICA STATUS FOR CHANNEL 'channel_1'\G` pada replika multi-sumber mengembalikan hasil berikut:

```
mysql> SHOW REPLICA STATUS FOR CHANNEL 'channel_1'\G
*************************** 1. row ***************************
             Replica_IO_State: Waiting for source to send event
                  Source_Host: myhost.XXXXXXXXXXXXXX.rr-rrrr-1.rds.amazonaws.com
                  Source_User: ReplicationUser
                  Source_Port: 3306
                Connect_Retry: 60
              Source_Log_File: mysql-bin-changelog.012345
          Read_Source_Log_Pos: 1219393
               Relay_Log_File: replica-relay-bin.000003
                Relay_Log_Pos: 30223388
        Relay_Source_Log_File: mysql-bin-changelog.012345
           Replica_IO_Running: No
          Replica_SQL_Running: Yes
              Replicate_Do_DB:.
              .
              .
                Last_IO_Errno: 1236
                Last_IO_Error: Got fatal error 1236 from master when reading data from
 binary log: 'Client requested master to start replication from impossible position;
 the first event 'mysql-bin-changelog.013406' at 1219393, the last event read from
 '/rdsdbdata/log/binlog/mysql-bin-changelog.012345' at 4, the last byte read from '/
rdsdbdata/log/binlog/mysql-bin-changelog.012345' at 4.'
               Last_SQL_Errno: 0
               Last_SQL_Error:
               .
               .
                 Channel_name: channel_1
              .
              .
 -- Some fields are omitted in this example output
```

Kolom `Last_IO_Errno` menunjukkan bahwa instans menerima kesalahan I/O 1236. Kolom `Source_Log_File` menunjukkan bahwa nama file adalah `mysql-bin-changelog.012345`,

yang berarti indeks file log adalah `12345`. Untuk mengatasi kesalahan, Anda dapat menelepon `mysql.rds_next_source_log_for_channel` dengan parameter berikut:

```
CALL mysql.rds_next_source_log_for_channel(12345,'channel_1');
```

> **Note**
>
> Versi sebelumnya dari MySQL menggunakan `SHOW SLAVE STATUS` bukan `SHOW REPLICA STATUS`. Jika Anda menggunakan MySQL versi sebelum 8.0.23, gunakan `SHOW SLAVE STATUS`.

mysql.rds_reset_external_source_for_channel

Menghentikan proses replikasi pada saluran yang ditentukan, dan menghapus saluran dan konfigurasi terkait dari replika multi-sumber.

> **Important**
>
> Untuk menjalankan prosedur ini, `autocommit` harus diaktifkan. Untuk mengaktifkannya, atur parameter `autocommit` ke 1. Lihat informasi tentang cara mengubah parameter di Memodifikasi parameter dalam grup parameter DB.

Sintaksis

```
CALL mysql.rds_reset_external_source_for_channel (channel_name);
```

Parameter-parameter

`channel_name`

Nama saluran replikasi pada replika multi-sumber. Setiap saluran replikasi menerima peristiwa log biner dari RDS sumber tunggal untuk instance MySQL DB yang berjalan pada host dan port tertentu.

Catatan penggunaan

Pengguna master harus menjalankan prosedur `mysql.rds_reset_external_source_for_channel`. Prosedur ini menghapus semua log relai milik saluran yang dihapus.

mysql.rds_set_external_source_for_channel

Mengkonfigurasi saluran replikasi pada RDS untuk MySQL DB instance untuk mereplikasi data dari RDS lain untuk MySQL DB instance.

> **Important**
>
> Untuk menjalankan prosedur ini, `autocommit` harus diaktifkan. Untuk mengaktifkannya, atur parameter `autocommit` ke 1. Lihat informasi tentang cara mengubah parameter di Memodifikasi parameter dalam grup parameter DB.

> **Note**
>
> Anda dapat menggunakan prosedur the section called "mysql.rds_set_external_source_with_delay_for_channel" tersimpan sebagai gantinya untuk mengonfigurasi saluran ini dengan replikasi tertunda.

Sintaks

```
CALL mysql.rds_set_external_source_for_channel (
  host_name
  , host_port
  , replication_user_name
  , replication_user_password
  , mysql_binary_log_file_name
  , mysql_binary_log_file_location
  , ssl_encryption
  , channel_name
);
```

Parameter

host_name

Nama host atau alamat IP dari RDS untuk contoh DB sumber MySQL.

host_port

Port yang digunakan oleh RDS untuk instance DB sumber MySQL. Jika konfigurasi jaringan Anda mencakup replikasi port Secure Shell (SSH) yang mengubah nomor port, tentukan nomor port yang diekspos oleh SSH.

replication_user_name

ID pengguna dengan `REPLICATION CLIENT` dan `REPLICATION SLAVE` izin pada RDS untuk instance DB sumber MySQL. Kami menyarankan Anda memberikan akun yang hanya digunakan untuk replikasi dengan instans DB sumber.

replication_user_password

Kata sandi ID pengguna yang ditentukan di `replication_user_name`.

mysql_binary_log_file_name

Nama log biner pada instance DB sumber yang berisi informasi replikasi.

mysql_binary_log_file_location

Lokasi di log biner `mysql_binary_log_file_name` tempat replikasi mulai membaca informasi replikasi.

Anda dapat menentukan nama dan lokasi file binlog dengan menjalankan `SHOW MASTER STATUS` instans DB sumber.

ssl_encryption

Nilai yang menentukan apakah enkripsi Lapisan Soket Aman (SSL) digunakan pada sambungan replikasi. 1 menentukan untuk menggunakan enkripsi SSL, 0 menentukan untuk tidak menggunakan enkripsi. Defaultnya adalah 0.

> **Note**
>
> Opsi `MASTER_SSL_VERIFY_SERVER_CERT` tidak didukung. Opsi ini diatur ke 0, yang berarti koneksi dienkripsi, tetapi sertifikat tidak diverifikasi.

channel_name

Nama saluran replikasi. Setiap saluran replikasi menerima peristiwa log biner dari RDS sumber tunggal untuk instance MySQL DB yang berjalan pada host dan port tertentu.

Catatan penggunaan

Pengguna master harus menjalankan prosedur `mysql.rds_set_external_source_for_channel`. Prosedur ini harus dijalankan pada RDS target untuk instance MySQL DB tempat Anda membuat saluran replikasi.

Sebelum Anda menjalankan`mysql.rds_set_external_source_for_channel`, konfigurasikan pengguna replikasi pada instans DB sumber dengan hak istimewa yang diperlukan untuk replika multi-sumber. Untuk menghubungkan replika multi-sumber ke instans DB sumber, Anda harus menentukan `replication_user_name` dan `replication_user_password` nilai pengguna replikasi yang memiliki `REPLICATION CLIENT` dan `REPLICATION SLAVE` izin pada instans DB sumber.

Untuk mengkonfigurasi pengguna replikasi pada instans DB sumber

1. Menggunakan klien MySQL pilihan Anda, sambungkan ke instans DB sumber dan buat akun pengguna yang akan digunakan untuk replikasi. Berikut sebuah contoh.

   > **Important**
   >
   > Sebagai praktik keamanan terbaik, tentukan kata sandi selain nilai placeholder yang ditunjukkan dalam contoh berikut.

   MySQL 8.0

   ```
   CREATE USER 'repl_user'@'example.com' IDENTIFIED WITH mysql_native_password BY
    'password';
   ```

   MySQL 5.7

   ```
   CREATE USER 'repl_user'@'example.com' IDENTIFIED BY 'password';
   ```

2. Pada instans DB sumber, berikan `REPLICATION CLIENT` dan `REPLICATION SLAVE` hak istimewa kepada pengguna replikasi Anda. Contoh berikut memberikan hak akses `REPLICATION CLIENT` dan `REPLICATION SLAVE` pada semua basis data untuk pengguna 'repl_user' domain Anda.

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'example.com';
```

Untuk menggunakan replikasi terenkripsi, konfigurasikan instans DB sumber untuk menggunakan koneksi SSL.

Setelah menelepon `mysql.rds_set_external_source_for_channel` untuk mengonfigurasi saluran replikasi ini, Anda dapat memanggil mysql.rds_start_replication_for_channel replika untuk memulai proses replikasi di saluran. Anda dapat menelepon the section called "mysql.rds_reset_external_source_for_channel" untuk menghentikan replikasi pada saluran dan menghapus konfigurasi saluran dari replika.

Saat Anda meneleponmysql.rds_set_external_source_for_channel, Amazon RDS mencatat waktu, pengguna, dan tindakan `set channel source` dalam `mysql.rds_history` tabel tanpa detail khusus saluran, dan dalam `mysql.rds_replication_status` tabel, dengan nama saluran. Informasi ini dicatat hanya untuk tujuan penggunaan dan pemantauan internal. Untuk mencatat panggilan prosedur lengkap untuk tujuan audit, pertimbangkan untuk mengaktifkan log audit atau log umum, berdasarkan persyaratan spesifik aplikasi Anda.

Contoh-contoh

Ketika dijalankan pada RDS untuk MySQL DB instance, contoh berikut mengkonfigurasi saluran replikasi `channel_1` bernama pada instance DB ini untuk mereplikasi data dari sumber yang ditentukan oleh host dan port. `sourcedb.example.com` `3306`

```
call mysql.rds_set_external_source_for_channel(
  'sourcedb.example.com',
  3306,
  'repl_user',
  'password',
  'mysql-bin-changelog.0777',
  120,
  0,
  'channel_1');
```

mysql.rds_set_external_source_with_auto_position_for_channel

Mengkonfigurasi saluran replikasi pada RDS untuk instance MySQL DB dengan penundaan replikasi opsional. Replikasi didasarkan pada pengidentifikasi transaksi global (GTID).

> **Important**
>
> Untuk menjalankan prosedur ini, `autocommit` harus diaktifkan. Untuk mengaktifkannya, atur parameter `autocommit` ke 1. Lihat informasi tentang cara mengubah parameter di Memodifikasi parameter dalam grup parameter DB.

Sintaksis

```
CALL mysql.rds_set_external_source_with_auto_position_for_channel (
  host_name
  , host_port
  , replication_user_name
  , replication_user_password
  , ssl_encryption
  , delay
  , channel_name
);
```

Parameter

host_name

Nama host atau alamat IP dari RDS untuk contoh DB sumber MySQL.

host_port

Port yang digunakan oleh RDS untuk instance DB sumber MySQL. Jika konfigurasi jaringan Anda mencakup replikasi port Secure Shell (SSH) yang mengubah nomor port, tentukan nomor port yang diekspos oleh SSH.

replication_user_name

ID pengguna dengan `REPLICATION CLIENT` dan `REPLICATION SLAVE` izin pada RDS untuk instance DB sumber MySQL. Kami menyarankan Anda memberikan akun yang hanya digunakan untuk replikasi dengan instans DB sumber.

replication_user_password

Kata sandi ID pengguna yang ditentukan dalam `replication_user_name`.

ssl_encryption

Nilai yang menentukan apakah enkripsi Lapisan Soket Aman (SSL) digunakan pada sambungan replikasi. 1 menentukan untuk menggunakan enkripsi SSL, 0 menentukan untuk tidak menggunakan enkripsi. Defaultnya adalah 0.

> **Note**
>
> Opsi `MASTER_SSL_VERIFY_SERVER_CERT` tidak didukung. Opsi ini diatur ke 0, yang berarti koneksi dienkripsi, tetapi sertifikat tidak diverifikasi.

delay

Jumlah minimum detik untuk menunda replikasi dari instans DB sumber.

Batas untuk parameter ini adalah satu hari (86.400 detik).

channel_name

Nama saluran replikasi. Setiap saluran replikasi menerima peristiwa log biner dari RDS sumber tunggal untuk instance MySQL DB yang berjalan pada host dan port tertentu.

Catatan penggunaan

Pengguna master harus menjalankan prosedur `mysql.rds_set_external_source_with_auto_position_for_channel`. Prosedur ini harus dijalankan pada RDS target untuk instance MySQL DB tempat Anda membuat saluran replikasi.

Sebelum Anda menjalankan`rds_set_external_source_with_auto_position_for_channel`, konfigurasikan pengguna replikasi pada instans DB sumber dengan hak istimewa yang diperlukan untuk replika multi-sumber. Untuk menghubungkan replika multi-sumber ke instans DB sumber, Anda harus menentukan `replication_user_name` dan `replication_user_password` nilai pengguna replikasi yang memiliki `REPLICATION CLIENT` dan `REPLICATION SLAVE` izin pada instans DB sumber.

Untuk mengkonfigurasi pengguna replikasi pada instans DB sumber

1. Menggunakan klien MySQL pilihan Anda, sambungkan ke instans DB sumber dan buat akun pengguna yang akan digunakan untuk replikasi. Berikut sebuah contoh.

   > **Important**
   >
   > Sebagai praktik keamanan terbaik, tentukan kata sandi selain nilai placeholder yang ditunjukkan dalam contoh berikut.

   MySQL 8.0

   ```
   CREATE USER 'repl_user'@'example.com' IDENTIFIED WITH mysql_native_password BY
    'password';
   ```

   MySQL 5.7

   ```
   CREATE USER 'repl_user'@'example.com' IDENTIFIED BY 'password';
   ```

2. Pada instans DB sumber, berikan `REPLICATION CLIENT` dan `REPLICATION SLAVE` hak istimewa kepada pengguna replikasi Anda. Contoh berikut memberikan hak akses `REPLICATION CLIENT` dan `REPLICATION SLAVE` pada semua basis data untuk pengguna 'repl_user' domain Anda.

   ```
   GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'example.com';
   ```

Untuk menggunakan replikasi terenkripsi, konfigurasikan instans DB sumber untuk menggunakan koneksi SSL.

Setelah menelepon `mysql.rds_set_external_source_with_auto_position_for_channel` untuk mengonfigurasi instans Amazon RDS DB sebagai replika baca pada saluran tertentu, Anda dapat memanggil replika baca untuk memulai proses replikasi the section called "mysql.rds_start_replication_for_channel" pada saluran tersebut.

Setelah menelepon `mysql.rds_set_external_source_with_auto_position_for_channel` untuk mengonfigurasi saluran replikasi ini, Anda dapat memanggil mysql.rds_start_replication_for_channel replika untuk memulai proses replikasi di saluran. Anda dapat

menelepon the section called “mysql.rds_reset_external_source_for_channel” untuk menghentikan replikasi pada saluran dan menghapus konfigurasi saluran dari replika.

Contoh-contoh

Ketika dijalankan pada RDS untuk MySQL DB instance, contoh berikut mengkonfigurasi saluran replikasi `channel_1` bernama pada instance DB ini untuk mereplikasi data dari sumber yang ditentukan oleh `sourcedb.example.com` host dan `3306` port Ini menetapkan penundaan replikasi minimum menjadi satu jam (3.600 detik). Ini berarti bahwa perubahan dari sumber RDS untuk instance MySQL DB tidak diterapkan pada replika multi-sumber setidaknya selama satu jam.

```
call mysql.rds_set_external_source_with_auto_position_for_channel(
  'sourcedb.example.com',
  3306,
  'repl_user',
  'password',
  0,
  3600,
  'channel_1');
```

mysql.rds_set_external_source_with_delay_for_channel

Mengkonfigurasi saluran replikasi pada RDS untuk instance MySQL DB dengan penundaan replikasi tertentu.

> **Important**
>
> Untuk menjalankan prosedur ini, `autocommit` harus diaktifkan. Untuk mengaktifkannya, atur parameter `autocommit` ke 1. Lihat informasi tentang cara mengubah parameter di Memodifikasi parameter dalam grup parameter DB.

Sintaksis

```
CALL mysql.rds_set_external_source_with_delay_for_channel (
  host_name
  , host_port
  , replication_user_name
  , replication_user_password
  , mysql_binary_log_file_name
```

```
  , mysql_binary_log_file_location
  , ssl_encryption
  , delay
  , channel_name
);
```

Parameter

host_name

Nama host atau alamat IP dari RDS untuk contoh DB sumber MySQL.

host_port

Port yang digunakan oleh RDS untuk instance DB sumber MySQL. Jika konfigurasi jaringan Anda mencakup replikasi port Secure Shell (SSH) yang mengubah nomor port, tentukan nomor port yang diekspos oleh SSH.

replication_user_name

ID pengguna dengan `REPLICATION CLIENT` dan `REPLICATION SLAVE` izin pada RDS untuk instance DB sumber MySQL. Kami menyarankan Anda memberikan akun yang hanya digunakan untuk replikasi dengan instans DB sumber.

replication_user_password

Kata sandi ID pengguna yang ditentukan di `replication_user_name`.

mysql_binary_log_file_name

Nama log biner pada instance DB sumber berisi informasi replikasi.

mysql_binary_log_file_location

Lokasi di log biner `mysql_binary_log_file_name` tempat replikasi akan mulai membaca informasi replikasi.

Anda dapat menentukan nama dan lokasi file binlog dengan menjalankan `SHOW MASTER STATUS` pada instans basis data sumber.

ssl_encryption

Nilai yang menentukan apakah enkripsi Lapisan Soket Aman (SSL) digunakan pada sambungan replikasi. 1 menentukan untuk menggunakan enkripsi SSL, 0 menentukan untuk tidak menggunakan enkripsi. Defaultnya adalah 0.

> **Note**
>
> Opsi `MASTER_SSL_VERIFY_SERVER_CERT` tidak didukung. Opsi ini diatur ke 0, yang berarti koneksi dienkripsi, tetapi sertifikat tidak diverifikasi.

`delay`

Jumlah minimum detik untuk menunda replikasi dari instans DB sumber.

Batas untuk parameter ini adalah satu hari (86.400 detik).

`channel_name`

Nama saluran replikasi. Setiap saluran replikasi menerima peristiwa log biner dari RDS sumber tunggal untuk instance MySQL DB yang berjalan pada host dan port tertentu.

Catatan penggunaan

Pengguna master harus menjalankan prosedur `mysql.rds_set_external_source_with_delay_for_channel`. Prosedur ini harus dijalankan pada RDS target untuk instance MySQL DB tempat Anda membuat saluran replikasi.

Sebelum Anda menjalankan`mysql.rds_set_external_source_with_delay_for_channel`, konfigurasikan pengguna replikasi pada instans DB sumber dengan hak istimewa yang diperlukan untuk replika multi-sumber. Untuk menghubungkan replika multi-sumber ke instans DB sumber, Anda harus menentukan `replication_user_name` dan `replication_user_password` nilai pengguna replikasi yang memiliki `REPLICATION CLIENT` dan `REPLICATION SLAVE` izin pada instans DB sumber.

Untuk mengkonfigurasi pengguna replikasi pada instans DB sumber

1. Menggunakan klien MySQL pilihan Anda, sambungkan ke instans DB sumber dan buat akun pengguna yang akan digunakan untuk replikasi. Berikut sebuah contoh.

   > **Important**
   >
   > Sebagai praktik keamanan terbaik, tentukan kata sandi selain nilai placeholder yang ditunjukkan dalam contoh berikut.

MySQL 8.0

```
CREATE USER 'repl_user'@'example.com' IDENTIFIED WITH mysql_native_password BY
 'password';
```

MySQL 5.7

```
CREATE USER 'repl_user'@'example.com' IDENTIFIED BY 'password';
```

2. Pada instans DB sumber, berikan `REPLICATION CLIENT` dan `REPLICATION SLAVE` hak istimewa kepada pengguna replikasi Anda. Contoh berikut memberikan hak akses `REPLICATION CLIENT` dan `REPLICATION SLAVE` pada semua basis data untuk pengguna 'repl_user' domain Anda.

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'example.com';
```

Untuk menggunakan replikasi terenkripsi, konfigurasikan instans DB sumber untuk menggunakan koneksi SSL.

Setelah menelepon `mysql.rds_set_external_source_with_delay_for_channel` untuk mengonfigurasi saluran replikasi ini, Anda dapat memanggil mysql.rds_start_replication_for_channel replika untuk memulai proses replikasi di saluran. Anda dapat menelepon the section called "mysql.rds_reset_external_source_for_channel" untuk menghentikan replikasi pada saluran dan menghapus konfigurasi saluran dari replika.

Saat Anda meneleponmysql.rds_set_external_source_with_delay_for_channel, Amazon RDS mencatat waktu, pengguna, dan tindakan `set channel source` dalam `mysql.rds_history` tabel tanpa detail khusus saluran, dan dalam `mysql.rds_replication_status` tabel, dengan nama saluran. Informasi ini dicatat hanya untuk tujuan penggunaan dan pemantauan internal. Untuk mencatat panggilan prosedur lengkap untuk tujuan audit, pertimbangkan untuk mengaktifkan log audit atau log umum, berdasarkan persyaratan spesifik aplikasi Anda.

Contoh-contoh

Ketika dijalankan pada RDS untuk MySQL DB instance, contoh berikut mengkonfigurasi saluran replikasi `channel_1` bernama pada instance DB ini untuk mereplikasi data dari sumber yang

ditentukan oleh `sourcedb.example.com` host dan `3306` port Ini menetapkan penundaan replikasi minimum menjadi satu jam (3.600 detik). Ini berarti bahwa perubahan dari sumber RDS untuk instance MySQL DB tidak diterapkan pada replika multi-sumber setidaknya selama satu jam.

```
call mysql.rds_set_external_source_with_delay_for_channel(
  'sourcedb.example.com',
  3306,
  'repl_user',
  'password',
  'mysql-bin-changelog.000777',
  120,
  0,
  3600,
  'channel_1');
```

mysql.rds_set_source_auto_position_for_channel

Menetapkan mode replikasi untuk saluran yang ditentukan untuk didasarkan pada posisi file log biner atau pada pengidentifikasi transaksi global (GTID).

Sintaks

```
CALL mysql.rds_set_source_auto_position_for_channel (
auto_position_mode
 , channel_name
);
```

Parameter

auto_position_mode

Nilai yang menunjukkan apakah akan menggunakan replikasi posisi file log atau replikasi berbasis GTID:

- `0` – Gunakan metode replikasi berdasarkan posisi file log biner. Default-nya adalah `0`.
- 1 – Gunakan metode replikasi berbasis GTID.

channel_name

Nama saluran replikasi pada replika multi-sumber. Setiap saluran replikasi menerima peristiwa log biner dari RDS sumber tunggal untuk instance MySQL DB yang berjalan pada host dan port tertentu.

Catatan penggunaan

Pengguna master harus menjalankan prosedur `mysql.rds_set_source_auto_position_for_channel`. Prosedur ini memulai ulang replikasi pada saluran yang ditentukan untuk menerapkan mode posisi auto yang ditentukan.

Contoh-contoh

Contoh berikut menetapkan mode posisi auto untuk channel_1 untuk menggunakan metode replikasi berbasis GTID.

```
call mysql.rds_set_source_auto_position_for_channel(1,'channel_1');
```

mysql.rds_set_source_delay_for_channel

Menetapkan jumlah minimum detik untuk menunda replikasi dari instance database sumber ke replika multi-sumber untuk saluran yang ditentukan.

Sintaks

```
CALL mysql.rds_set_source_delay_for_channel(delay, channel_name);
```

Parameter

`delay`

Jumlah minimum detik untuk menunda replikasi dari instans DB sumber.

Batas untuk parameter ini adalah satu hari (86.400 detik).

`channel_name`

Nama saluran replikasi pada replika multi-sumber. Setiap saluran replikasi menerima peristiwa log biner dari RDS sumber tunggal untuk instance MySQL DB yang berjalan pada host dan port tertentu.

Catatan penggunaan

Pengguna master harus menjalankan prosedur `mysql.rds_set_source_delay_for_channel`. Untuk menggunakan prosedur ini, panggilan pertama `mysql.rds_stop_replication_for_channel` untuk menghentikan replikasi. Kemudian,

panggil prosedur ini untuk mengatur nilai penundaan replikasi. Saat penundaan diatur, panggil `mysql.rds_start_replication_for_channel` untuk memulai ulang replikasi.

Contoh-contoh

Contoh berikut menetapkan penundaan replikasi dari instance database sumber pada `channel_1` replika multi-sumber setidaknya selama satu jam (3.600 detik).

```
CALL mysql.rds_set_source_delay_for_channel(3600,'channel_1');
```

mysql.rds_skip_repl_error_for_channel

Melewatkan peristiwa log biner dan menghapus kesalahan replikasi pada replika multi-sumber MySQL DB untuk saluran yang ditentukan.

Sintaks

```
CALL mysql.rds_skip_repl_error_for_channel(channel_name);
```

Parameter-parameter

`channel_name`

Nama saluran replikasi pada replika multi-sumber. Setiap saluran replikasi menerima peristiwa log biner dari RDS sumber tunggal untuk instance MySQL DB yang berjalan pada host dan port tertentu.

Catatan penggunaan

Pengguna utama harus menjalankan prosedur `mysql.rds_skip_repl_error_for_channel` pada replika baca. Anda dapat menggunakan prosedur ini dengan cara yang sama `mysql.rds_skip_repl_error` digunakan untuk melewati kesalahan pada replika baca. Untuk informasi selengkapnya, lihat Memanggil prosedur mysql.rds_skip_repl_error.

> **Note**
>
> Untuk melewati kesalahan dalam replikasi berbasis GTID, kami sarankan Anda menggunakan prosedur sebagai gantinya. the section called “mysql.rds_skip_transaction_with_gtid”

Untuk menentukan apakah ada kesalahan, jalankan perintah `SHOW REPLICA STATUS FOR CHANNEL 'channel_name'\G` MySQL. Jika kesalahan replikasi tidak parah, Anda dapat menjalankan `mysql.rds_skip_repl_error_for_channel` untuk melewati kesalahan tersebut. Jika ada beberapa kesalahan, `mysql.rds_skip_repl_error_for_channel` menghapus kesalahan pertama pada saluran replikasi yang ditentukan, kemudian memperingatkan bahwa orang lain hadir. Anda kemudian dapat menggunakan `SHOW REPLICA STATUS FOR CHANNEL 'channel_name'\G` untuk menentukan tindakan yang benar untuk kesalahan berikutnya. Untuk informasi tentang nilai yang ditampilkan, lihat pernyataan TUNJUKKAN STATUS REPLIKA di dokumentasi MySQL.

mysql.rds_start_replication_for_channel

Memulai replikasi dari RDS untuk instance MySQL DB ke replika multi-sumber pada saluran yang ditentukan.

> **Note**
>
> Anda dapat menggunakan prosedur tersimpan mysql.rds_start_replication_until_for_channel atau mysql.rds_start_replication_until_gtid_for_channel untuk memulai replikasi dari instans DB RDS for MySQL dan menghentikan replikasi di lokasi file log biner yang ditentukan.

Sintaks

```
CALL mysql.rds_start_replication_for_channel(channel_name);
```

Parameter-parameter

channel_name

Nama saluran replikasi pada replika multi-sumber. Setiap saluran replikasi menerima peristiwa log biner dari RDS sumber tunggal untuk instance MySQL DB yang berjalan pada host dan port tertentu.

Catatan penggunaan

Pengguna master harus menjalankan prosedur `mysql.rds_start_replication_for_channel`. Setelah Anda mengimpor data dari sumber RDS untuk instance MySQL DB, jalankan perintah ini pada replika multi-sumber untuk memulai replikasi pada saluran yang ditentukan.

Contoh-contoh

Contoh berikut memulai replikasi pada `channel_1` replika multi-sumber.

```
CALL mysql.rds_start_replication_for_channel('channel_1');
```

mysql.rds_start_replication_until_for_channel

Memulai replikasi dari RDS untuk instance MySQL DB pada saluran yang ditentukan dan menghentikan replikasi di lokasi file log biner yang ditentukan.

Sintaks

```
CALL mysql.rds_start_replication_until_for_channel (
replication_log_file
  , replication_stop_point
  , channel_name
);
```

Parameter

`replication_log_file`

Nama log biner pada instance DB sumber berisi informasi replikasi.

`replication_stop_point`

Lokasi di log biner `replication_log_file` tempat replikasi akan berhenti.

`channel_name`

Nama saluran replikasi pada replika multi-sumber. Setiap saluran replikasi menerima peristiwa log biner dari RDS sumber tunggal untuk instance MySQL DB yang berjalan pada host dan port tertentu.

Catatan penggunaan

Pengguna master harus menjalankan prosedur `mysql.rds_start_replication_until_for_channel`. Dengan prosedur ini, replikasi dimulai dan kemudian berhenti ketika posisi file binlog yang ditentukan tercapai. Untuk versi 8.0, prosedur hanya berhenti. `SQL_Thread` Untuk versi 5.7, prosedur menghentikan kedua `SQL_Thread` dan. `IO_Thread`

Nama file yang ditentukan untuk `replication_log_file` parameter harus cocok dengan nama file binlog instance DB sumber.

Ketika `replication_stop_point` parameter menentukan lokasi berhenti yang di masa lalu, replikasi segera dihentikan.

Contoh-contoh

Contoh berikut memulai replikasi`channel_1`, dan mereplikasi perubahan hingga mencapai lokasi `120` dalam file log `mysql-bin-changelog.000777` biner.

```
call mysql.rds_start_replication_until_for_channel(
  'mysql-bin-changelog.000777',
  120,
  'channel_1'
  );
```

mysql.rds_start_replication_until_gtid_for_channel

Memulai replikasi pada saluran yang ditentukan dari RDS untuk instance MySQL DB dan menghentikan replikasi pada pengidentifikasi transaksi global (GTID) yang ditentukan.

Sintaks

```
CALL mysql.rds_start_replication_until_gtid_for_channel(gtid,channel_name);
```

Parameter

`gtid`

GTID setelah itu untuk menghentikan replikasi.

channel_name

Nama saluran replikasi pada replika multi-sumber. Setiap saluran replikasi menerima peristiwa log biner dari RDS sumber tunggal untuk instance MySQL DB yang berjalan pada host dan port tertentu.

Catatan penggunaan

Pengguna master harus menjalankan prosedur `mysql.rds_start_replication_until_gtid_for_channel`. Prosedur memulai replikasi pada saluran yang ditentukan dan menerapkan semua perubahan hingga nilai GTID yang ditentukan. Kemudian, ia menghentikan replikasi pada saluran.

Saat parameter `gtid` menentukan transaksi yang telah dijalankan oleh replika, replikasi akan segera dihentikan.

Sebelum Anda menjalankan prosedur ini, Anda harus menonaktifkan replikasi multi-threaded dengan menetapkan nilai dari atau ke. `replica_parallel_workers` `slave_parallel_workers` `0`

Contoh-contoh

Contoh berikut memulai replikasi`channel_1`, dan mereplikasi perubahan hingga mencapai GTID. `3E11FA47-71CA-11E1-9E33-C80AA9429562:23`

```
call mysql.rds_start_replication_until_gtid_for_channel('3E11FA47-71CA-11E1-9E33-
C80AA9429562:23','channel_1');
```

mysql.rds_stop_replication_for_channel

Menghentikan replikasi dari instance MySQL DB pada saluran yang ditentukan.

Sintaks

```
CALL mysql.rds_stop_replication_for_channel(channel_name);
```

Parameter-parameter

channel_name

Nama saluran replikasi pada replika multi-sumber. Setiap saluran replikasi menerima peristiwa log biner dari RDS sumber tunggal untuk instance MySQL DB yang berjalan pada host dan port tertentu.

Catatan penggunaan

Pengguna master harus menjalankan prosedur `mysql.rds_stop_replication_for_channel`.

Contoh-contoh

Contoh berikut menghentikan replikasi pada `channel_1` replika multi-sumber.

```
CALL mysql.rds_stop_replication_for_channel('channel_1');
```

Mengelola Global Status History

Amazon RDS menyediakan serangkaian prosedur yang mengambil snapshot nilai-nilai variabel status dari waktu ke waktu dan menuliskannya ke tabel, beserta perubahan apa pun yang dibuat sejak snapshot terakhir. Infrastruktur ini disebut Global Status History. Untuk informasi selengkapnya, lihat Mengelola Global Status History.

Prosedur tersimpan berikut mengelola bagaimana Global Status History dikumpulkan dan dipelihara.

Topik

mysql.rds_collect_global_status_history

Mengambil snapshot sesuai permintaan untuk Global Status History.

Sintaksis

```
CALL mysql.rds_collect_global_status_history;
```

mysql.rds_disable_gsh_collector

Menonaktifkan snapshot yang diambil oleh Global Status History.

Sintaksis

```
CALL mysql.rds_disable_gsh_collector;
```

mysql.rds_disable_gsh_rotation

Menonaktifkan rotasi tabel `mysql.global_status_history`.

Sintaksis

```
CALL mysql.rds_disable_gsh_rotation;
```

mysql.rds_enable_gsh_collector

Mengaktifkan Global Status History untuk mengambil snapshot default pada interval yang ditentukan oleh `rds_set_gsh_collector`.

Sintaksis

```
CALL mysql.rds_enable_gsh_collector;
```

mysql.rds_enable_gsh_rotation

Mengaktifkan rotasi isi tabel `mysql.global_status_history` ke `mysql.global_status_history_old` pada interval yang ditentukan oleh `rds_set_gsh_rotation`.

Sintaksis

```
CALL mysql.rds_enable_gsh_rotation;
```

mysql.rds_rotate_global_status_history

Merotasi isi tabel `mysql.global_status_history` ke `mysql.global_status_history_old` sesuai permintaan.

Sintaksis

```
CALL mysql.rds_rotate_global_status_history;
```

mysql.rds_set_gsh_collector

Menentukan interval, dalam menit, di antara snapshot yang diambil oleh Global Status History.

Sintaksis

```
CALL mysql.rds_set_gsh_collector(intervalPeriod);
```

Parameter

intervalPeriod

Interval, dalam menit, antara snapshot. Nilai default-nya adalah 5.

mysql.rds_set_gsh_rotation

Menentukan interval, dalam hari, antara rotasi tabel `mysql.global_status_history`.

Sintaksis

```
CALL mysql.rds_set_gsh_rotation(intervalPeriod);
```

Parameter

intervalPeriod

Interval, dalam hari, antara rotasi tabel. Nilai default-nya adalah 7.

Mereplikasi

Prosedur tersimpan berikut mengontrol bagaimana transaksi direplikasi dari basis data eksternal ke dalam RDS for MySQL, atau dari RDS for MySQL ke basis data eksternal. Untuk mempelajari cara menggunakan replikasi berdasarkan pengidentifikasi transaksi global (GTID) dengan RDS for MySQL, lihat Menggunakan replikasi berbasis GTID untuk Amazon RDS for MySQL.

Topik

mysql.rds_next_master_log

Mengubah posisi log instans basis data sumber menjadi awal log biner berikutnya pada instans basis data sumber. Gunakan prosedur ini hanya jika Anda menerima kesalahan I/O 1236 replikasi pada replika baca.

Sintaksis

```
CALL mysql.rds_next_master_log(
curr_master_log
);
```

Parameter

curr_master_log

Indeks file log master saat ini. Misalnya, jika file saat ini bernama `mysql-bin-changelog.012345`, maka indeksnya adalah 12345. Untuk menentukan nama file log master saat ini, jalankan perintah `SHOW REPLICA STATUS` dan lihat kolom `Master_Log_File`.

> **Note**
>
> Versi MySQL sebelumnya menggunakan `SHOW SLAVE STATUS`, bukan `SHOW REPLICA STATUS`. Jika Anda menggunakan versi MySQL sebelum 8.0.23, gunakan `SHOW SLAVE STATUS`.

Catatan penggunaan

Pengguna utama harus menjalankan prosedur `mysql.rds_next_master_log`.

> **Warning**
>
> Panggil `mysql.rds_next_master_log` hanya jika replikasi gagal setelah failover dari instans DB Multi-AZ yang merupakan sumber replikasi, dan kolom `Last_IO_Errno` dari laporan kesalahan I/O 1236 `SHOW REPLICA STATUS`.
> Memanggil `mysql.rds_next_master_log` dapat mengakibatkan hilangnya data di replika baca jika transaksi dalam instans sumber tidak ditulis ke log biner di disk sebelum peristiwa failover terjadi.
> Anda dapat mengurangi kemungkinan hal ini terjadi dengan mengatur parameter instans sumber `sync_binlog` dan `innodb_support_xa` ke 1, meskipun ini dapat mengurangi performa. Untuk informasi selengkapnya, lihat Pemecahan Masalah batasan replika baca MySQL.

Contoh-contoh

Asumsikan replikasi gagal pada replika baca RDS for MySQL. Menjalankan `SHOW REPLICA STATUS\G` pada replika baca akan menampilkan hasil berikut:

```
*************************** 1. row ***************************
             Replica_IO_State:
```

```
                  Source_Host: myhost.XXXXXXXXXXXXXXX.rr-rrrr-1.rds.amazonaws.com
                  Source_User: MasterUser
                  Source_Port: 3306
                Connect_Retry: 10
              Source_Log_File: mysql-bin-changelog.012345
          Read_Source_Log_Pos: 1219393
               Relay_Log_File: relaylog.012340
                Relay_Log_Pos: 30223388
        Relay_Source_Log_File: mysql-bin-changelog.012345
           Replica_IO_Running: No
          Replica_SQL_Running: Yes
              Replicate_Do_DB:
          Replicate_Ignore_DB:
           Replicate_Do_Table:
       Replicate_Ignore_Table:
      Replicate_Wild_Do_Table:
  Replicate_Wild_Ignore_Table:
                   Last_Errno: 0
                   Last_Error:
                 Skip_Counter: 0
          Exec_Source_Log_Pos: 30223232
              Relay_Log_Space: 5248928866
              Until_Condition: None
               Until_Log_File:
                Until_Log_Pos: 0
           Source_SSL_Allowed: No
           Source_SSL_CA_File:
           Source_SSL_CA_Path:
              Source_SSL_Cert:
            Source_SSL_Cipher:
               Source_SSL_Key:
        Seconds_Behind_Master: NULL
Source_SSL_Verify_Server_Cert: No
                Last_IO_Errno: 1236
                Last_IO_Error: Got fatal error 1236 from master when reading data from
 binary log: 'Client requested master to start replication from impossible position;
 the first event 'mysql-bin-changelog.013406' at 1219393, the last event read from
 '/rdsdbdata/log/binlog/mysql-bin-changelog.012345' at 4, the last byte read from '/
rdsdbdata/log/binlog/mysql-bin-changelog.012345' at 4.'
               Last_SQL_Errno: 0
               Last_SQL_Error:
  Replicate_Ignore_Server_Ids:
             Source_Server_Id: 67285976
```

Kolom `Last_IO_Errno` menunjukkan bahwa instans menerima kesalahan I/O 1236. Kolom `Master_Log_File` menunjukkan bahwa nama file adalah `mysql-bin-changelog.012345`, yang berarti indeks file log adalah `12345`. Untuk mengatasi kesalahan, Anda dapat memanggil `mysql.rds_next_master_log` dengan parameter berikut:

```
CALL mysql.rds_next_master_log(12345);
```

> **Note**
>
> Versi MySQL sebelumnya menggunakan `SHOW SLAVE STATUS`, bukan `SHOW REPLICA STATUS`. Jika Anda menggunakan versi MySQL sebelum 8.0.23, gunakan `SHOW SLAVE STATUS`.

mysql.rds_reset_external_master

Mengonfigurasi ulang instans DB RDS for MySQL agar tidak lagi menjadi replika baca dari instans MySQL yang berjalan di luar Amazon RDS.

> **Important**
>
> Untuk menjalankan prosedur ini, `autocommit` harus diaktifkan. Untuk mengaktifkannya, atur parameter `autocommit` ke 1. Lihat informasi tentang cara mengubah parameter di Memodifikasi parameter dalam grup parameter DB.

Sintaksis

```
CALL mysql.rds_reset_external_master;
```

Catatan penggunaan

Pengguna utama harus menjalankan prosedur `mysql.rds_reset_external_master`. Prosedur ini harus dijalankan pada instans DB MySQL agar dihapus sebagai replika baca dari instans MySQL yang berjalan di luar Amazon RDS.

> **Note**
>
> Kami menyarankan Anda menggunakan replikasi baca untuk mengelola replikasi antara dua instans DB Amazon RDS jika memungkinkan. Saat Anda melakukannya, sebaiknya hanya gunakan replika baca ini dan prosedur tersimpan terkait replikasi lainnya. Praktik ini memungkinkan topologi replikasi yang lebih kompleks antara instans DB Amazon RDS. Kami menawarkan prosedur tersimpan ini terutama untuk mengaktifkan replikasi dengan instans MySQL yang berjalan di luar Amazon RDS. Untuk informasi tentang mengelola replikasi di antara instans DB Amazon RDS, lihat Menggunakan replika baca instans DB.

Untuk informasi selengkapnya tentang cara menggunakan replikasi untuk mengimpor data dari instans MySQL yang berjalan di luar Amazon RDS, lihat Mengonfigurasi replikasi posisi file log biner dengan instans sumber eksternal.

Mengonfigurasi instans DB RDS for MySQL agar tidak lagi menjadi replika baca dari instans MySQL yang berjalan di luar Amazon RDS.

> **Important**
>
> Untuk menjalankan prosedur ini, `autocommit` harus diaktifkan. Untuk mengaktifkannya, atur parameter `autocommit` ke 1. Lihat informasi tentang cara mengubah parameter di Memodifikasi parameter dalam grup parameter DB.

> **Note**
>
> Anda bisa menggunakan prosedur tersimpan mysql.rds_set_external_master_with_delay untuk mengonfigurasi instans basis data sumber eksternal dan replikasi tertunda.

Sintaksis

```
CALL mysql.rds_set_external_master (
  host_name
  , host_port
  , replication_user_name
```

```
  , replication_user_password
  , mysql_binary_log_file_name
  , mysql_binary_log_file_location
  , ssl_encryption
);
```

Parameter

host_name

Nama host atau alamat IP instans MySQL yang berjalan di luar Amazon RDS untuk menjadi instans basis data sumber.

host_port

Port yang digunakan oleh instans MySQL yang berjalan di luar Amazon RDS untuk dikonfigurasi sebagai instans basis data sumber. Jika konfigurasi jaringan Anda mencakup replikasi port Secure Shell (SSH) yang mengubah nomor port, tentukan nomor port yang diekspos oleh SSH.

replication_user_name

ID pengguna dengan izin `REPLICATION CLIENT` dan `REPLICATION SLAVE` pada instans MySQL yang berjalan di luar Amazon RDS. Kami menyarankan Anda memberikan akun yang digunakan sepenuhnya untuk replikasi dengan instans eksternal.

replication_user_password

Kata sandi ID pengguna yang ditentukan di `replication_user_name`.

mysql_binary_log_file_name

Nama log biner pada instans basis data sumber yang berisi informasi replikasi.

mysql_binary_log_file_location

Lokasi di log biner `mysql_binary_log_file_name` tempat replikasi mulai membaca informasi replikasi.

Anda dapat menentukan nama dan lokasi file binlog dengan menjalankan `SHOW MASTER STATUS` pada instans basis data sumber.

ssl_encryption

Nilai yang menentukan apakah enkripsi Lapisan Soket Aman (SSL) digunakan pada sambungan replikasi. 1 menentukan untuk menggunakan enkripsi SSL, 0 menentukan untuk tidak menggunakan enkripsi. Default-nya adalah 0.

> **Note**
>
> Opsi `MASTER_SSL_VERIFY_SERVER_CERT` tidak didukung. Opsi ini diatur ke 0, yang berarti koneksi dienkripsi, tetapi sertifikat tidak diverifikasi.

Catatan penggunaan

Pengguna utama harus menjalankan prosedur `mysql.rds_set_external_master`. Prosedur ini harus dijalankan pada instans DB MySQL untuk dikonfigurasi sebagai replika baca dari instans MySQL yang berjalan di luar Amazon RDS.

Sebelum menjalankan `mysql.rds_set_external_master`, Anda harus mengonfigurasi instans MySQL yang berjalan di luar Amazon RDS menjadi instans basis data sumber. Untuk terhubung ke instans MySQL yang berjalan di luar Amazon RDS, Anda harus menentukan nilai `replication_user_name` dan `replication_user_password` yang menunjukkan pengguna replikasi yang memiliki izin `REPLICATION CLIENT` dan `REPLICATION SLAVE` pada instans eksternal MySQL.

Untuk mengonfigurasi instans eksternal MySQL sebagai instans basis data sumber

1. Dengan menggunakan klien MySQL pilihan Anda, hubungkan ke instans eksternal MySQL dan buat akun pengguna yang akan digunakan untuk replikasi. Berikut adalah contohnya.

   MySQL 5.7

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
   ```

   MySQL 8.0

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED WITH mysql_native_password BY
    'password';
   ```

   > **Note**
   >
   > Tetapkan kata sandi selain penggugah (prompt) yang ditampilkan di sini sebagai praktik terbaik keamanan.

2. Pada instans eksternal MySQL, berikan hak istimewa `REPLICATION CLIENT` dan `REPLICATION SLAVE` kepada pengguna replikasi Anda. Contoh berikut memberikan hak akses `REPLICATION CLIENT` dan `REPLICATION SLAVE` pada semua basis data untuk pengguna 'repl_user' domain Anda.

   MySQL 5.7

   ```
   GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com'
    IDENTIFIED BY 'password';
   ```

   MySQL 8.0

   ```
   GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
   ```

Untuk menggunakan replikasi terenkripsi, konfigurasikan instans basis data sumber untuk menggunakan koneksi SSL.

> **Note**
>
> Kami menyarankan Anda menggunakan replikasi baca untuk mengelola replikasi antara dua instans DB Amazon RDS jika memungkinkan. Saat Anda melakukannya, sebaiknya hanya gunakan replika baca ini dan prosedur tersimpan terkait replikasi lainnya. Praktik ini memungkinkan topologi replikasi yang lebih kompleks antara instans DB Amazon RDS. Kami menawarkan prosedur tersimpan ini terutama untuk mengaktifkan replikasi dengan instans MySQL yang berjalan di luar Amazon RDS. Untuk informasi tentang mengelola replikasi di antara instans DB Amazon RDS, lihat Menggunakan replika baca instans DB.

Setelah memanggil `mysql.rds_set_external_master` untuk mengonfigurasi instans DB Amazon RDS sebagai replika baca, Anda dapat memanggil mysql.rds_start_replication pada replika baca untuk memulai proses replikasi. Anda dapat memanggil mysql.rds_reset_external_master untuk menghapus konfigurasi replika baca.

Saat `mysql.rds_set_external_master` dipanggil, Amazon RDS mencatat waktu, pengguna, dan tindakan `set master` di tabel `mysql.rds_history` dan `mysql.rds_replication_status`.

Contoh-contoh

Ketika dijalankan pada instans DB MySQL, contoh berikut mengonfigurasi instans DB menjadi replika baca dari instans MySQL yang berjalan di luar Amazon RDS.

```
call mysql.rds_set_external_master(
  'Externaldb.some.com',
  3306,
  'repl_user',
  'password',
  'mysql-bin-changelog.0777',
  120,
  0);
```

mysql.rds_set_external_master_with_auto_position

Mengonfigurasi instans DB RDS for MySQL menjadi replika baca dari instans MySQL yang berjalan di luar Amazon RDS. Prosedur ini juga mengonfigurasi replikasi tertunda dan replikasi berdasarkan pengidentifikasi transaksi global (GTID).

> **Important**
>
> Untuk menjalankan prosedur ini, `autocommit` harus diaktifkan. Untuk mengaktifkannya, atur parameter `autocommit` ke 1. Lihat informasi tentang cara mengubah parameter di Memodifikasi parameter dalam grup parameter DB.

Sintaksis

```
CALL mysql.rds_set_external_master_with_auto_position (
  host_name
  , host_port
  , replication_user_name
  , replication_user_password
  , ssl_encryption
  , delay
);
```

Parameter

host_name

Nama host atau alamat IP instans MySQL yang berjalan di luar Amazon RDS untuk menjadi instans basis data sumber.

host_port

Port yang digunakan oleh instans MySQL yang berjalan di luar Amazon RDS untuk dikonfigurasi sebagai instans basis data sumber. Jika konfigurasi jaringan Anda mencakup replikasi port Secure Shell (SSH) yang mengubah nomor port, tentukan nomor port yang diekspos oleh SSH.

replication_user_name

ID pengguna dengan izin `REPLICATION CLIENT` dan `REPLICATION SLAVE` pada instans MySQL yang berjalan di luar Amazon RDS. Kami menyarankan Anda memberikan akun yang digunakan sepenuhnya untuk replikasi dengan instans eksternal.

replication_user_password

Kata sandi ID pengguna yang ditentukan dalam `replication_user_name`.

ssl_encryption

Nilai yang menentukan apakah enkripsi Lapisan Soket Aman (SSL) digunakan pada sambungan replikasi. 1 menentukan untuk menggunakan enkripsi SSL, 0 menentukan untuk tidak menggunakan enkripsi. Default-nya adalah 0.

> **Note**
>
> Opsi `MASTER_SSL_VERIFY_SERVER_CERT` tidak didukung. Opsi ini diatur ke 0, yang berarti koneksi dienkripsi, tetapi sertifikat tidak diverifikasi.

delay

Jumlah detik minimum untuk menunda replikasi dari instans basis data sumber.

Batas untuk parameter ini adalah satu hari (86.400 detik).

Catatan penggunaan

Pengguna utama harus menjalankan prosedur `mysql.rds_set_external_master_with_auto_position`. Prosedur ini harus dijalankan pada instans DB MySQL untuk dikonfigurasi sebagai replika baca dari instans MySQL yang berjalan di luar Amazon RDS.

Prosedur ini didukung untuk semua versi RDS for MySQL 5.7, dan RDS for MySQL 8.0.26, dan versi 8.0 yang lebih tinggi.

Sebelum menjalankan `mysql.rds_set_external_master_with_auto_position`, Anda harus mengonfigurasi instans MySQL yang berjalan di luar Amazon RDS menjadi instans basis data sumber. Untuk terhubung ke instans MySQL yang berjalan di luar Amazon RDS, Anda harus menentukan nilai untuk `replication_user_name` dan `replication_user_password`. Nilai-nilai ini harus menunjukkan pengguna replikasi yang memiliki izin `REPLICATION CLIENT` dan `REPLICATION SLAVE` pada instans external MySQL.

Untuk mengonfigurasi instans eksternal MySQL sebagai instans basis data sumber

1. Dengan menggunakan klien MySQL pilihan Anda, hubungkan ke instans eksternal MySQL dan buat akun pengguna yang akan digunakan untuk replikasi. Berikut adalah contohnya.

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'SomePassW0rd'
   ```

2. Pada instans eksternal MySQL, berikan hak akses `REPLICATION CLIENT` dan `REPLICATION SLAVE` kepada pengguna replikasi Anda. Contoh berikut memberikan hak akses `REPLICATION CLIENT` dan `REPLICATION SLAVE` pada semua basis data untuk pengguna `'repl_user'` domain Anda.

   ```
   GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com'
   IDENTIFIED BY 'SomePassW0rd'
   ```

Untuk informasi selengkapnya, lihat Mengonfigurasi replikasi posisi file log biner dengan instans sumber eksternal.

> **Note**
>
> Kami menyarankan Anda menggunakan replikasi baca untuk mengelola replikasi antara dua instans DB Amazon RDS jika memungkinkan. Saat Anda melakukannya, sebaiknya

hanya gunakan replika baca ini dan prosedur tersimpan terkait replikasi lainnya. Praktik ini memungkinkan topologi replikasi yang lebih kompleks antara instans DB Amazon RDS. Kami menawarkan prosedur tersimpan ini terutama untuk mengaktifkan replikasi dengan instans MySQL yang berjalan di luar Amazon RDS. Untuk informasi tentang mengelola replikasi di antara instans DB Amazon RDS, lihat Menggunakan replika baca instans DB.

Setelah memanggil `mysql.rds_set_external_master_with_auto_position` untuk mengonfigurasi instans DB Amazon RDS sebagai replika baca, Anda dapat memanggil mysql.rds_start_replication pada replika baca untuk memulai proses replikasi. Anda dapat memanggil mysql.rds_reset_external_master untuk menghapus konfigurasi replika baca.

Saat Anda memanggil `mysql.rds_set_external_master_with_auto_position`, Amazon RDS mencatat waktu, pengguna, dan tindakan `set master` di tabel `mysql.rds_history` dan `mysql.rds_replication_status`.

Untuk pemulihan bencana, Anda dapat menggunakan prosedur ini dengan prosedur tersimpan mysql.rds_start_replication_until atau mysql.rds_start_replication_until_gtid. Untuk meneruskan perubahan ke replika baca yang tertunda ke waktu sebelum bencana, Anda dapat menjalankan prosedur `mysql.rds_set_external_master_with_auto_position`. Setelah prosedur `mysql.rds_start_replication_until_gtid` menghentikan replikasi, Anda dapat mempromosikan replika baca menjadi instans DB primer baru dengan menggunakan petunjuk di Mempromosikan replika baca menjadi instans DB mandiri.

Untuk menggunakan prosedur `mysql.rds_rds_start_replication_until_gtid`, replikasi berbasis GTID harus diaktifkan. Untuk melewati transaksi berbasis GTID tertentu yang diketahui menyebabkan bencana, Anda dapat menggunakan prosedur tersimpan mysql.rds_skip_transaction_with_gtid. Untuk informasi selengkapnya tentang cara menggunakan replikasi berbasis GTID, lihat Menggunakan replikasi berbasis GTID untuk Amazon RDS for MySQL.

Contoh-contoh

Ketika dijalankan pada instans DB MySQL, contoh berikut mengonfigurasi instans DB menjadi replika baca dari instans MySQL yang berjalan di luar Amazon RDS. Ini menetapkan penundaan replikasi minimum menjadi satu jam (3.600 detik) pada instans DB MySQL. Perubahan dari instans basis data sumber MySQL yang berjalan di luar Amazon RDS tidak diterapkan pada replika baca instans DB MySQL selama setidaknya satu jam.

```
call mysql.rds_set_external_master_with_auto_position(
```

```
  'Externaldb.some.com',
  3306,
  'repl_user',
  'SomePassW0rd',
  0,
  3600);
```

mysql.rds_set_external_master_with_delay

Mengonfigurasi instans DB RDS for MySQL menjadi replika baca dari instans MySQL yang berjalan di luar Amazon RDS dan mengonfigurasi replikasi tertunda.

> **Important**
>
> Untuk menjalankan prosedur ini, `autocommit` harus diaktifkan. Untuk mengaktifkannya, atur parameter `autocommit` ke 1. Lihat informasi tentang cara mengubah parameter di Memodifikasi parameter dalam grup parameter DB.

Sintaksis

```
CALL mysql.rds_set_external_master_with_delay (
  host_name
  , host_port
  , replication_user_name
  , replication_user_password
  , mysql_binary_log_file_name
  , mysql_binary_log_file_location
  , ssl_encryption
  , delay
);
```

Parameter

host_name

Nama host atau alamat IP dari instans MySQL yang berjalan di luar Amazon RDS yang akan menjadi instans basis data sumber.

host_port

Port yang digunakan oleh instans MySQL yang berjalan di luar Amazon RDS untuk dikonfigurasi sebagai instans basis data sumber. Jika konfigurasi jaringan Anda mencakup replikasi port SSH yang mengubah nomor port, tentukan nomor port yang diekspos oleh SSH.

replication_user_name

ID pengguna dengan izin `REPLICATION CLIENT` dan `REPLICATION SLAVE` pada instans MySQL yang berjalan di luar Amazon RDS. Kami menyarankan Anda memberikan akun yang digunakan sepenuhnya untuk replikasi dengan instans eksternal.

replication_user_password

Kata sandi ID pengguna yang ditentukan di `replication_user_name`.

mysql_binary_log_file_name

Nama log biner pada Instans basis data sumber berisi informasi replikasi.

mysql_binary_log_file_location

Lokasi di log biner `mysql_binary_log_file_name` tempat replikasi akan mulai membaca informasi replikasi.

Anda dapat menentukan nama dan lokasi file binlog dengan menjalankan `SHOW MASTER STATUS` pada instans basis data sumber.

ssl_encryption

Nilai yang menentukan apakah enkripsi Lapisan Soket Aman (SSL) digunakan pada sambungan replikasi. 1 menentukan untuk menggunakan enkripsi SSL, 0 menentukan untuk tidak menggunakan enkripsi. Default-nya adalah 0.

> **Note**
>
> Opsi `MASTER_SSL_VERIFY_SERVER_CERT` tidak didukung. Opsi ini diatur ke 0, yang berarti koneksi dienkripsi, tetapi sertifikat tidak diverifikasi.

delay

Jumlah detik minimum untuk menunda replikasi dari instans basis data sumber.

Batas untuk parameter ini adalah satu hari (86.400 detik).

Catatan penggunaan

Pengguna utama harus menjalankan prosedur `mysql.rds_set_external_master_with_delay`. Prosedur ini harus dijalankan pada instans DB MySQL untuk dikonfigurasi sebagai replika baca dari instans MySQL yang berjalan di luar Amazon RDS.

Sebelum menjalankan `mysql.rds_set_external_master_with_delay`, Anda harus mengonfigurasi instans MySQL yang berjalan di luar Amazon RDS menjadi instans basis data sumber. Untuk terhubung ke instans MySQL yang berjalan di luar Amazon RDS, Anda harus menentukan nilai untuk `replication_user_name` dan `replication_user_password`. Nilai-nilai ini harus menunjukkan pengguna replikasi yang memiliki izin `REPLICATION CLIENT` dan `REPLICATION SLAVE` pada instans external MySQL.

Untuk mengonfigurasi instans eksternal MySQL sebagai instans basis data sumber

1. Dengan menggunakan klien MySQL pilihan Anda, hubungkan ke instans eksternal MySQL dan buat akun pengguna yang akan digunakan untuk replikasi. Berikut adalah contohnya.

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'SomePassW0rd'
   ```

2. Pada instans eksternal MySQL, berikan hak akses `REPLICATION CLIENT` dan `REPLICATION SLAVE` kepada pengguna replikasi Anda. Contoh berikut memberikan hak akses `REPLICATION CLIENT` dan `REPLICATION SLAVE` pada semua basis data untuk pengguna `'repl_user'` domain Anda.

   ```
   GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com'
   IDENTIFIED BY 'SomePassW0rd'
   ```

Untuk informasi selengkapnya, lihat Mengonfigurasi replikasi posisi file log biner dengan instans sumber eksternal.

> **Note**
>
> Kami menyarankan Anda menggunakan replikasi baca untuk mengelola replikasi antara dua instans DB Amazon RDS jika memungkinkan. Saat Anda melakukannya, sebaiknya hanya gunakan replika baca ini dan prosedur tersimpan terkait replikasi lainnya. Praktik ini memungkinkan topologi replikasi yang lebih kompleks antara instans DB Amazon RDS. Kami menawarkan prosedur tersimpan ini terutama untuk mengaktifkan replikasi dengan instans

MySQL yang berjalan di luar Amazon RDS. Untuk informasi tentang mengelola replikasi di antara instans DB Amazon RDS, lihat Menggunakan replika baca instans DB.

Setelah memanggil `mysql.rds_set_external_master_with_delay` untuk mengonfigurasi instans DB Amazon RDS sebagai replika baca, Anda dapat memanggil mysql.rds_start_replication pada replika baca untuk memulai proses replikasi. Anda dapat memanggil mysql.rds_reset_external_master untuk menghapus konfigurasi replika baca.

Saat Anda memanggil `mysql.rds_set_external_master_with_delay`, Amazon RDS mencatat waktu, pengguna, dan tindakan `set master` di tabel `mysql.rds_history` dan `mysql.rds_replication_status`.

Untuk pemulihan bencana, Anda dapat menggunakan prosedur ini dengan prosedur tersimpan mysql.rds_start_replication_until atau mysql.rds_start_replication_until_gtid. Untuk meneruskan perubahan ke replika baca yang tertunda ke waktu sebelum bencana, Anda dapat menjalankan prosedur `mysql.rds_set_external_master_with_delay`. Setelah prosedur `mysql.rds_start_replication_until` menghentikan replikasi, Anda dapat mempromosikan replika baca menjadi instans DB primer baru dengan menggunakan petunjuk di Mempromosikan replika baca menjadi instans DB mandiri.

Untuk menggunakan prosedur `mysql.rds_rds_start_replication_until_gtid`, replikasi berbasis GTID harus diaktifkan. Untuk melewati transaksi berbasis GTID tertentu yang diketahui menyebabkan bencana, Anda dapat menggunakan prosedur tersimpan mysql.rds_skip_transaction_with_gtid. Untuk informasi selengkapnya tentang cara menggunakan replikasi berbasis GTID, lihat Menggunakan replikasi berbasis GTID untuk Amazon RDS for MySQL.

Prosedur `mysql.rds_set_external_master_with_delay` tersedia dalam versi RDS for MySQL ini:

- MySQL 8.0.26 dan versi 8.0 yang lebih tinggi
- Semua versi 5.7

Contoh-contoh

Ketika dijalankan pada instans DB MySQL, contoh berikut mengonfigurasi instans DB menjadi replika baca dari instans MySQL yang berjalan di luar Amazon RDS. Ini menetapkan penundaan replikasi minimum menjadi satu jam (3.600 detik) pada instans DB MySQL. Perubahan dari instans basis data

sumber MySQL yang berjalan di luar Amazon RDS tidak diterapkan pada replika baca instans DB MySQL selama setidaknya satu jam.

```
call mysql.rds_set_external_master_with_delay(
  'Externaldb.some.com',
  3306,
  'repl_user',
  'SomePassW0rd',
  'mysql-bin-changelog.000777',
  120,
  0,
  3600);
```

mysql.rds_set_master_auto_position

Mengatur mode replikasi agar didasarkan pada posisi file log biner atau pengidentifikasi transaksi global (GTID).

Sintaksis

```
CALL mysql.rds_set_master_auto_position (
auto_position_mode
);
```

Parameter

auto_position_mode

Nilai yang menunjukkan apakah akan menggunakan replikasi posisi file log atau replikasi berbasis GTID:

- `0` – Gunakan metode replikasi berdasarkan posisi file log biner. Default-nya adalah `0`.
- 1 – Gunakan metode replikasi berbasis GTID.

Catatan penggunaan

Pengguna utama harus menjalankan prosedur `mysql.rds_set_master_auto_position`.

Prosedur ini didukung untuk semua versi RDS for MySQL 5.7, dan RDS for MySQL 8.0.26, dan versi 8.0 yang lebih tinggi.

mysql.rds_set_source_delay

Mengatur jumlah detik minimum untuk menunda replikasi dari instans basis data sumber ke replika baca saat ini. Gunakan prosedur ini saat Anda tersambung ke replika baca untuk menunda replikasi dari instans basis data sumbernya.

Sintaksis

```
CALL mysql.rds_set_source_delay(
delay
);
```

Parameter

delay

Jumlah detik minimum untuk menunda replikasi dari instans basis data sumber.

Batas untuk parameter ini adalah satu hari (86.400 detik).

Catatan penggunaan

Pengguna utama harus menjalankan prosedur `mysql.rds_set_source_delay`.

Untuk pemulihan bencana, Anda dapat menggunakan prosedur ini dengan prosedur tersimpan mysql.rds_start_replication_until atau mysql.rds_start_replication_until_gtid. Untuk meneruskan perubahan ke replika baca yang tertunda ke waktu sebelum bencana, Anda dapat menjalankan prosedur `mysql.rds_set_source_delay`. Setelah prosedur `mysql.rds_start_replication_until` atau `mysql.rds_start_replication_until_gtid` menghentikan replikasi, Anda dapat mempromosikan replika baca menjadi instans DB primer baru dengan menggunakan petunjuk di Mempromosikan replika baca menjadi instans DB mandiri.

Untuk menggunakan prosedur `mysql.rds_rds_start_replication_until_gtid`, replikasi berbasis GTID harus diaktifkan. Untuk melewati transaksi berbasis GTID tertentu yang diketahui menyebabkan bencana, Anda dapat menggunakan prosedur tersimpan mysql.rds_skip_transaction_with_gtid. Untuk informasi selengkapnya tentang replikasi berbasis GTID, lihat Menggunakan replikasi berbasis GTID untuk Amazon RDS for MySQL.

Prosedur `mysql.rds_set_source_delay` tersedia dalam versi RDS for MySQL ini:

- MySQL 8.0.26 dan versi 8.0 yang lebih tinggi
- Semua versi 5.7

Contoh-contoh

Untuk menunda replikasi dari instans basis data sumber ke replika baca saat ini setidaknya selama satu jam (3.600 detik), Anda dapat memanggil `mysql.rds_set_source_delay` dengan parameter berikut:

```
CALL mysql.rds_set_source_delay(3600);
```

mysql.rds_skip_transaction_with_gtid

Melewati replikasi transaksi dengan pengenal transaksi global (GTID) yang ditentukan pada instans DB MySQL.

Anda dapat menggunakan prosedur ini untuk pemulihan bencana ketika transaksi GTID tertentu diketahui menyebabkan masalah. Gunakan prosedur tersimpan ini untuk melewati transaksi bermasalah. Contoh transaksi bermasalah mencakup transaksi yang menonaktifkan replikasi, menghapus data penting, atau menyebabkan instans DB menjadi tidak tersedia.

Sintaksis

```
CALL mysql.rds_skip_transaction_with_gtid (
gtid_to_skip
);
```

Parameter

`gtid_to_skip`

GTID dari transaksi replikasi yang akan dilewati.

Catatan penggunaan

Pengguna utama harus menjalankan prosedur `mysql.rds_skip_transaction_with_gtid`.

Prosedur ini didukung untuk semua versi RDS for MySQL 5.7, dan RDS for MySQL 8.0.26, dan versi 8.0 yang lebih tinggi.

Contoh-contoh

Contoh berikut melewati replikasi transaksi dengan GTID `3E11FA47-71CA-11E1-9E33-C80AA9429562:23`.

```
CALL mysql.rds_skip_transaction_with_gtid('3E11FA47-71CA-11E1-9E33-C80AA9429562:23');
```

mysql.rds_skip_repl_error

Melewati dan menghapus kesalahan replikasi pada replika baca DB MySQL.

Sintaksis

```
CALL mysql.rds_skip_repl_error;
```

Catatan penggunaan

Pengguna utama harus menjalankan prosedur `mysql.rds_skip_repl_error` pada replika baca. Untuk informasi selengkapnya tentang prosedur ini, lihat Memanggil prosedur mysql.rds_skip_repl_error.

Untuk menentukan apakah ada kesalahan, jalankan perintah `SHOW REPLICA STATUS\G` MySQL. Jika kesalahan replikasi tidak parah, Anda dapat menjalankan `mysql.rds_skip_repl_error` untuk melewati kesalahan tersebut. Jika ada beberapa kesalahan, `mysql.rds_skip_repl_error` akan menghapus kesalahan pertama, lalu memberi peringatan bahwa ada kesalahan lain. Anda kemudian dapat menggunakan `SHOW REPLICA STATUS\G` untuk menentukan tindakan yang benar untuk kesalahan berikutnya. Untuk informasi tentang nilai yang ditampilkan, lihat SHOW REPLICA STATUS statement dalam dokumentasi MySQL.

> **Note**
>
> Versi MySQL sebelumnya menggunakan `SHOW SLAVE STATUS`, bukan `SHOW REPLICA STATUS`. Jika Anda menggunakan versi MySQL sebelum 8.0.23, gunakan `SHOW SLAVE STATUS`.

Untuk informasi selengkapnya tentang cara mengatasi kesalahan replikasi dengan Amazon RDS, lihat Pemecahan Masalah batasan replika baca MySQL.

Kesalahan replikasi terhenti

Ketika memanggil prosedur `mysql.rds_skip_repl_error`, Anda mungkin menerima pesan kesalahan yang menyatakan bahwa replika tidak berfungsi atau dinonaktifkan.

Pesan kesalahan ini muncul jika Anda menjalankan prosedur pada instans primer, bukan replika baca. Anda harus menjalankan prosedur ini pada replika baca agar prosedur berfungsi.

Pesan kesalahan ini mungkin juga muncul jika Anda menjalankan prosedur pada replika baca, tetapi replikasi tidak berhasil dimulai ulang.

Jika Anda perlu melewati sejumlah besar kesalahan, lag replikasi dapat meningkat hingga melampaui periode retensi default untuk file log biner (binlog). Dalam kasus ini, Anda mungkin mengalami kesalahan fatal karena file binlog dihapus sebelum diputar ulang di replika baca. Penghapusan ini menyebabkan replikasi berhenti, dan Anda tidak dapat lagi memanggil perintah `mysql.rds_skip_repl_error` untuk melewati kesalahan replikasi.

Anda dapat memitigasi masalah ini dengan meningkatkan jumlah jam retensi file binlog tersebut pada instans basis data sumber Anda. Setelah meningkatkan waktu retensi binlog, Anda dapat memulai ulang replikasi dan memanggil perintah `mysql.rds_skip_repl_error` sesuai kebutuhan.

Untuk mengatur waktu retensi binlog, gunakan prosedur mysql.rds_set_configuration dan tentukan parameter konfigurasi `'binlog retention hours'` bersama dengan jumlah jam untuk mempertahankan file binlog di klaster DB. Contoh berikut menetapkan periode penyimpanan file binlog menjadi 48 jam.

```
CALL mysql.rds_set_configuration('binlog retention hours', 48);
```

mysql.rds_start_replication

Memulai replikasi dari instans DB RDS for MySQL.

> **Note**
>
> Anda dapat menggunakan prosedur tersimpan mysql.rds_start_replication_until atau mysql.rds_start_replication_until_gtid untuk memulai replikasi dari instans DB RDS for MySQL dan menghentikan replikasi di lokasi file log biner yang telah ditentukan.

Sintaksis

```
CALL mysql.rds_start_replication;
```

Catatan penggunaan

Pengguna utama harus menjalankan prosedur `mysql.rds_start_replication`.

Untuk mengimpor data dari instans MySQL yang berjalan di luar Amazon RDS, panggil `mysql.rds_start_replication` pada replika baca untuk memulai proses replikasi setelah Anda memanggil `mysql.rds_set_external_master` membangun konfigurasi replikasi. Untuk informasi selengkapnya, lihat Memulihkan cadangan ke instans DB MySQL.

Untuk mengekspor data ke instans MySQL yang berjalan di luar Amazon RDS, panggil `mysql.rds_start_replication` dan `mysql.rds_stop_replication` pada replika baca untuk mengontrol beberapa tindakan replikasi, seperti membersihkan log biner. Untuk informasi selengkapnya, lihat Mengekspor data dari instans DB MySQL dengan menggunakan replikasi.

Anda juga dapat memanggil `mysql.rds_start_replication` pada replika baca untuk memulai kembali proses replikasi apa pun yang sebelumnya Anda hentikan dengan memanggil `mysql.rds_stop_replication`. Untuk informasi selengkapnya, lihat Menggunakan replika baca instans DB.

mysql.rds_start_replication_until

Memulai replikasi dari instans DB RDS for MySQL dan menghentikan replikasi di lokasi file log biner yang telah ditentukan.

Sintaksis

```
CALL mysql.rds_start_replication_until (
replication_log_file
  , replication_stop_point
);
```

Parameter

`replication_log_file`

Nama log biner pada instans basis data sumber yang berisi informasi replikasi.

replication_stop_point

Lokasi di log biner `replication_log_file` tempat replikasi akan berhenti.

Catatan penggunaan

Pengguna utama harus menjalankan prosedur `mysql.rds_start_replication_until`.

Prosedur `mysql.rds_start_replication_until` tersedia dalam versi RDS for MySQL ini:

- MySQL 8.0.26 dan versi 8.0 yang lebih tinggi
- Semua versi 5.7

Anda dapat menggunakan prosedur ini dengan replikasi tertunda untuk pemulihan bencana. Jika Anda telah mengonfigurasi replikasi tertunda, Anda dapat menggunakan prosedur ini untuk meneruskan perubahan ke replika baca tertunda ke waktu sebelum bencana terjadi. Setelah prosedur ini menghentikan replikasi, Anda dapat mempromosikan replika baca menjadi instans DB primer baru dengan menggunakan petunjuk di Mempromosikan replika baca menjadi instans DB mandiri.

Anda dapat mengonfigurasi replikasi tertunda menggunakan prosedur tersimpan berikut ini:

- mysql.rds_set_configuration
- mysql.rds_set_external_master_with_delay
- mysql.rds_set_source_delay

Nama file yang ditentukan untuk parameter `replication_log_file` harus cocok dengan nama file binlog instans basis data sumber.

Jika parameter `replication_stop_point` menentukan lokasi perhentian di masa lalu, replikasi akan segera dihentikan.

Contoh-contoh

Contoh berikut memulai replikasi dan mereplikasi perubahan hingga mencapai lokasi `120` di file log biner `mysql-bin-changelog.000777`.

```
call mysql.rds_start_replication_until(
  'mysql-bin-changelog.000777',
```

```
  120);
```

mysql.rds_start_replication_until_gtid

Memulai replikasi dari instans DB RDS for MySQL dan menghentikan replikasi segera setelah pengidentifikasi transaksi global (GTID) yang ditentukan.

Sintaksis

```
CALL mysql.rds_start_replication_until_gtid(gtid);
```

Parameter

gtid

GTID setelah replikasi dihentikan.

Catatan penggunaan

Pengguna utama harus menjalankan prosedur `mysql.rds_start_replication_until_gtid`.

Prosedur ini didukung untuk semua versi RDS for MySQL 5.7, dan RDS for MySQL 8.0.26, dan versi 8.0 yang lebih tinggi.

Anda dapat menggunakan prosedur ini dengan replikasi tertunda untuk pemulihan bencana. Jika Anda telah mengonfigurasi replikasi tertunda, Anda dapat menggunakan prosedur ini untuk meneruskan perubahan ke replika baca tertunda ke waktu sebelum bencana terjadi. Setelah prosedur ini menghentikan replikasi, Anda dapat mempromosikan replika baca menjadi instans DB primer baru dengan menggunakan petunjuk di Mempromosikan replika baca menjadi instans DB mandiri.

Anda dapat mengonfigurasi replikasi tertunda menggunakan prosedur tersimpan berikut ini:

- mysql.rds_set_configuration
- mysql.rds_set_external_master_with_delay
- mysql.rds_set_source_delay

Saat parameter `gtid` menentukan transaksi yang telah dijalankan oleh replika, replikasi akan segera dihentikan.

Contoh-contoh

Contoh berikut memulai replikasi dan mereplikasi perubahan hingga mencapai GTID `3E11FA47-71CA-11E1-9E33-C80AA9429562:23`.

```
call mysql.rds_start_replication_until_gtid('3E11FA47-71CA-11E1-9E33-C80AA9429562:23');
```

mysql.rds_stop_replication

Menghentikan replikasi dari instans DB MySQL.

Sintaksis

```
CALL mysql.rds_stop_replication;
```

Catatan penggunaan

Pengguna utama harus menjalankan prosedur `mysql.rds_stop_replication`.

Jika Anda mengonfigurasi replikasi untuk mengimpor data dari instans MySQL yang berjalan di luar Amazon RDS, Anda memanggil `mysql.rds_stop_replication` pada replika baca untuk menghentikan proses replikasi setelah impor selesai. Untuk informasi selengkapnya, lihat Memulihkan cadangan ke instans DB MySQL.

Jika Anda mengonfigurasi replikasi untuk mengekspor data ke instans MySQL yang berjalan di luar Amazon RDS, Anda memanggil `mysql.rds_start_replication` dan `mysql.rds_stop_replication` pada replika baca untuk mengontrol beberapa tindakan replikasi, seperti membersihkan log biner. Untuk informasi selengkapnya, lihat Mengekspor data dari instans DB MySQL dengan menggunakan replikasi.

Anda juga dapat menggunakan `mysql.rds_stop_replication` untuk menghentikan replikasi antara dua instans DB Amazon RDS. Anda biasanya menghentikan replikasi untuk menjalankan operasi berjangka panjang pada replika baca, seperti membuat indeks besar pada replika baca. Anda dapat memulai kembali proses replikasi apa pun yang sebelumnya Anda hentikan dengan memanggil mysql.rds_start_replication pada replika baca. Untuk informasi selengkapnya, lihat Menggunakan replika baca instans DB.

Pemanasan cache InnoDB

Prosedur tersimpan berikut menyimpan, memuat, atau membatalkan pemuatan kumpulan buffer InnoDB di instans DB RDS for MySQL. Untuk informasi selengkapnya, lihat Pemanasan cache InnoDB untuk MySQL di Amazon RDS.

Topik

mysql.rds_innodb_buffer_pool_dump_now

Membuang status kumpulan buffer saat ini ke disk.

Sintaksis

```
CALL mysql.rds_innodb_buffer_pool_dump_now();
```

Catatan penggunaan

Pengguna master harus menjalankan prosedur `mysql.rds_innodb_buffer_pool_dump_now`.

mysql.rds_innodb_buffer_pool_load_abort

Membatalkan pemuatan status kumpulan buffer yang disimpan saat sedang berlangsung.

Sintaksis

```
CALL mysql.rds_innodb_buffer_pool_load_abort();
```

Catatan penggunaan

Pengguna master harus menjalankan prosedur `mysql.rds_innodb_buffer_pool_load_abort`.

mysql.rds_innodb_buffer_pool_load_now

Memuat status kumpulan buffer yang disimpan dari disk.

Sintaksis

```
CALL mysql.rds_innodb_buffer_pool_load_now();
```

Catatan penggunaan

Pengguna master harus menjalankan prosedur `mysql.rds_innodb_buffer_pool_load_now`.

Amazon RDS for Oracle

Amazon RDS mendukung instans DB yang menjalankan versi dan edisi Oracle Database berikut ini:

- Oracle Database 21c (21.0.0.0)
- Oracle Database 19c (19.0.0.0)

> **Note**
>
> Oracle Database 11g, Oracle Database 12c, dan Oracle Database 18c adalah versi lama yang tidak lagi didukung di Amazon RDS.

Sebelum membuat instans DB, selesaikan langkah-langkah di bagian Menyiapkan Amazon RDS panduan ini. Saat membuat instans DB menggunakan akun master Anda, akun tersebut mendapatkan hak istimewa DBA, dengan beberapa batasan. Gunakan akun ini untuk tugas administratif seperti membuat akun basis data tambahan. Anda tidak dapat menggunakan SYS, SYSTEM, atau akun administratif lain yang diberikan oleh Oracle.

Anda dapat membuat berikut ini:

- Instans DB
- Snapshot DB
- Pemulihan titik waktu
- Pencadangan otomatis
- Pencadangan manual

Anda dapat menggunakan instans DB yang menjalankan Oracle di dalam VPC. Anda juga dapat menambahkan fitur ke instans DB Oracle Anda dengan mengaktifkan berbagai opsi. Amazon RDS mendukung deployment Multi-AZ untuk Oracle sebagai solusi failover dengan ketersediaan tinggi.

> **Important**
>
> Untuk memberikan pengalaman layanan terkelola, Amazon RDS tidak memberikan akses shell ke instans DB. Hal tersebut juga membatasi akses ke prosedur dan tabel sistem tertentu yang membutuhkan hak istimewa tingkat lanjut. Anda dapat mengakses basis data

Anda menggunakan klien SQL standar seperti Oracle SQL*Plus. Namun, Anda tidak dapat mengakses host secara langsung dengan menggunakan Telnet atau Secure Shell (SSH).

Topik

Ikhtisar Oracle di Amazon RDS

Anda dapat membaca bagian-bagian berikut untuk mengetahui ikhtisar RDS for Oracle.

Topik

- Batasan RDS for Oracle

Fitur-fitur RDS for Oracle

Amazon RDS for Oracle mendukung sebagian besar fitur dan kemampuan Oracle Database. Beberapa fitur mungkin memiliki dukungan terbatas atau hak istimewa yang dibatasi. Beberapa fitur hanya tersedia di Enterprise Edition, dan beberapa fitur memerlukan lisensi tambahan. Untuk informasi lebih lanjut tentang fitur Oracle Database untuk Oracle Database versi tertentu, lihat Panduan Pengguna Informasi Lisensi Oracle Database untuk versi yang Anda gunakan.

Anda dapat memfilter fitur-fitur Amazon RDS baru pada halaman Apa yang Baru di Database?. Untuk Produk, pilih Amazon RDS. Lalu, cari dengan menggunakan kata kunci seperti **`Oracle 2022`**.

> **Note**
>
> Daftar-daftar berikut tidak lengkap.

Topik

- Fitur-fitur baru di RDS for Oracle
- Fitur-fitur yang didukung di RDS for Oracle
- Fitur-fitur yang tidak didukung di RDS for Oracle

Fitur-fitur baru di RDS for Oracle

Untuk melihat fitur baru di RDS untuk Oracle, gunakan teknik berikut:

- Cari Riwayat dokumen dengan kata kunci **`Oracle`**.
- Filter fitur Amazon RDS baru di Apa yang Baru dengan Database? halaman. Untuk Produk, pilih Amazon RDS. Kemudian cari **`Oracle`** ***`YYYY`***, di mana *YYYY* merupakan tahun, misalnya **`2024`**.

Fitur-fitur yang didukung di RDS for Oracle

Amazon RDS for Oracle mendukung fitur-fitur Oracle Database berikut ini:

- Advanced Compression

- Application Express (APEX)

  Untuk informasi selengkapnya, lihat Oracle Application Express (APEX).
- Automatic Memory Management
- Automatic Undo Management
- Automatic Workload Repository (AWR)

  Untuk informasi selengkapnya, lihat Membuat laporan performa dengan Automatic Workload Repository (AWR).
- Penjaga Data Aktif dengan Kinerja Maksimum di AWS Wilayah yang sama atau di seluruh AWS Wilayah

  Untuk informasi selengkapnya, lihat Menggunakan replika baca untuk Amazon RDS for Oracle.
- Tabel Blockchain (Oracle Database 21c dan yang lebih tinggi)

  Untuk informasi selengkapnya, lihat Mengelola Tabel Blockchain dalam dokumentasi Database Oracle.
- Continuous Query Notification (versi 12.1.0.2.v7 dan yang lebih tinggi)

  Untuk informasi selengkapnya, lihat Menggunakan Continuous Query Notification (CQN) dalam dokumentasi Oracle.
- Data Redaction
- Database Change Notification

  Untuk informasi lebih lanjut, lihat Database Change Notification dalam dokumentasi Oracle.

  > **Note**
  >
  > Fitur ini berubah menjadi Continuous Query Notification di Oracle Database 12c Rilis 1 (12.1) dan yang lebih tinggi.
- Database In-Memory (Oracle Database 12c dan yang lebih tinggi)
- Distributed Queries and Transactions
- Edition-Based Redefinition

  Untuk informasi selengkapnya, lihat Mengatur edisi default untuk instans DB.
- EM Express (12c dan yang lebih tinggi)

Untuk informasi selengkapnya, lihat Oracle Enterprise Manager.

- Fine-Grained Auditing
- Flashback Table, Flashback Query, Flashback Transaction Query
- Rollover kata sandi bertahap untuk aplikasi (Oracle Database 21c dan yang lebih tinggi)

  Untuk informasi selengkapnya, lihat Managing Gradual Database Password Rollover for Applications dalam dokumentasi Oracle Database.
- HugePages

  Untuk informasi selengkapnya, lihat Mengaktifkan HugePages untuk instans RDS for Oracle.
- Impor/ekspor (warisan dan Data Pump) serta SQL*Loader

  Untuk informasi selengkapnya, lihat Mengimpor data ke Oracle di Amazon RDS.
- Java Virtual Machine (JVM)

  Untuk informasi selengkapnya, lihat Mesin virtual Oracle Java.
- JavaScript (Oracle Database 21c dan lebih tinggi)

  Untuk informasi selengkapnya, lihat DBMS_MLE dalam dokumentasi Oracle Database.
- Keamanan Label (Oracle Database 12c dan yang lebih tinggi)

  Untuk informasi selengkapnya, lihat Keamanan Label Oracle.
- Locator

  Untuk informasi selengkapnya, lihat Oracle Locator.
- Tampilan Terwujud
- Multimedia

  Untuk informasi selengkapnya, lihat Oracle Multimedia.
- Multipenghuni

  Arsitektur multipenghuni Oracle didukung untuk semua Oracle Database 19c dan rilis yang lebih tinggi. Untuk informasi selengkapnya, lihat Bekerja dengan CDB di RDS for Oracle.
- Enkripsi jaringan

  Untuk informasi lebih lanjut, lihat Enkripsi jaringan asli Oracle dan Lapisan Soket Aman Oracle.

- Partitioning
- Real Application Testing

  Untuk menggunakan kemampuan pengambilan dan pemutaran ulang penuh, Anda harus menggunakan Amazon Elastic File System (Amazon EFS) untuk mengakses file yang dihasilkan oleh Oracle Real Application Testing. Untuk informasi lebih lanjut, lihat Integrasi Amazon EFS dan posting blog Gunakan fitur Oracle Real Application Testing dengan Amazon RDS for Oracle.
- Sharding di tingkat aplikasi (tetapi bukan fitur Oracle Sharding)
- Spasial dan Grafik

  Untuk informasi selengkapnya, lihat Oracle Spatial.
- Star Query Optimization
- Streams dan Advanced Queuing
- Summary Management – Penulisan Ulang Kueri Tampilan Terwujud
- Teks (Jenis penyimpanan data File dan URL tidak didukung)
- Total Recall
- Transparent Data Encryption (TDE)

  Untuk informasi selengkapnya, lihat Enkripsi Data Transparan Oracle.
- Unified Auditing, Mixed Mode

  Untuk informasi lebih lanjut, lihat Mixed mode auditing dalam dokumentasi Oracle.
- XML DB (tanpa XML DB Protocol Server)

  Untuk informasi selengkapnya, lihat DB XML Oracle.
- Virtual Private Database

Fitur-fitur yang tidak didukung di RDS for Oracle

Amazon RDS for Oracle tidak mendukung fitur-fitur Oracle Database berikut ini:

- Automatic Storage Management (ASM)
- Database Vault
- Flashback Database

> **Note**
>
> Untuk solusi alternatif, lihat Alternatif entri Blog AWS Database ke fitur database kilas balik Oracle di Amazon RDS for Oracle.

- FTP dan SFTP
- Tabel partisi hibrida
- Messaging Gateway
- Repositori Manajemen Oracle Enterprise Manager Cloud Control
- Real Application Clusters (Oracle RAC)
- Real Application Security (RAS)
- Unified Auditing, Pure Mode
- Skela Workspace Manager (WMSYS)

> **Note**
>
> Daftar sebelumnya tidak lengkap.

> **Warning**
>
> Secara umum, Amazon RDS tidak mencegah Anda membuat skema untuk fitur yang tidak didukung. Namun, jika Anda membuat skema untuk fitur dan komponen Oracle yang memerlukan hak istimewa SYSDBA, Anda dapat merusak kamus data dan memengaruhi ketersediaan instans DB Anda. Hanya gunakan fitur yang didukung dan skema yang tersedia di Menambahkan opsi untuk instans DB Oracle.

Rilis RDS for Oracle

Amazon RDS for Oracle mendukung beberapa rilis Oracle Database.

> **Note**
>
> Untuk informasi selengkapnya tentang peningkatan rilis, lihat Meng-upgrade mesin DB Oracle.

Topik

- Oracle Database 21c dengan Amazon RDS
- Oracle Database 19c dengan Amazon RDS
- Oracle Database 12c dengan Amazon RDS

Oracle Database 21c dengan Amazon RDS

Amazon RDS mendukung Oracle Database 21c, yang mencakup Oracle Enterprise Edition dan Oracle Standard Edition 2. Oracle Database 21c (21.0.0.0) mencakup banyak fitur baru dan pembaruan dari versi sebelumnya. Perubahan utamanya adalah Oracle Database 21c hanya mendukung arsitektur multi-penghuni: Anda tidak dapat lagi membuat basis data sebagai non-CDB tradisional. Untuk mempelajari lebih lanjut tentang perbedaan antara CDB dan non-CDB, lihat Batasan CDB RDS for Oracle.

Di bagian ini, Anda dapat menemukan fitur dan perubahan yang penting dalam penggunaan Oracle Database 21c (21.0.0.0) di Amazon RDS. Untuk daftar perubahan lengkapnya, lihat dokumentasi Oracle Database 21c. Untuk daftar lengkap fitur yang didukung oleh setiap edisi Oracle Database 21c, lihat Fitur, opsi, dan paket manajemen yang diizinkan oleh penawaran database Oracle dalam dokumentasi Oracle.

Perubahan parameter Amazon RDS for Oracle Database 21c (21.0.0.0)

Oracle Database 21c (21.0.0.0) menyertakan beberapa parameter baru dan parameter dengan rentang baru dan nilai default baru.

Topik

- Parameter baru
- Perubahan untuk parameter yang kompatibel
- Parameter yang dihapus

Parameter baru

Tabel berikut ini menunjukkan parameter Amazon RDS yang baru untuk Oracle Database 21c (21.0.0.0).

Nama	Rentang nilai	Nilai default	Dapat diubah	Deskripsi
blockchain_table_max_no_drop	NONE | 0	NONE	Y	Memungkinkan Anda mengontrol jumlah maksimum waktu siaga yang dapat ditentukan saat membuat tabel blockchain.
dbnest_enable	NONE | CDB_RESOURCE_PDB_ALL	NONE	T	Memungkinkan Anda mengaktifkan atau menonaktifkan dbNest. DbNest menyediakan isolasi dan manajemen sumber daya sistem operasi, isolasi sistem file, dan komputasi aman untuk PDB.
dbnest_pdb_fs_conf	NONE | *pathname*	NONE	T	Menentukan file konfigurasi sistem file dbNest untuk PDB.
diagnostics_control	ERROR | WARNING | IGNORE	IGNORE	Y	Memungkinkan Anda mengontrol dan memantau pengguna yang melakukan operasi diagnostik basis data yang berpotensi tidak aman.
drcp_dedicated_opt	YES | NO	YES	Y	Mengaktifkan atau menonaktifkan optimasi khusus dengan Database Resident Connection Pooling (DRCP).
enable_per_pdb_drcp	true | false	true	T	Mengontrol apakah Database Resident Connection Pooling

Nama	Rentang nilai	Nilai default	Dapat diubah	Deskripsi
				(DRCP) mengonfigurasi satu kumpulan koneksi untuk seluruh CDB atau satu kumpulan koneksi terisolasi untuk setiap PDB.
inmemory_deep_vectorization	`true | false`	`true`	Y	Mengaktifkan atau menonaktifkan kerangka vektorisasi mendalam.
mandatory_user_profile	*`profile_name`*	N/A	T	Menentukan profil pengguna wajib untuk CDB atau PDB.
optimizer_capture_sql_quarantine	`true | false`	`false`	Y	Mengaktifkan atau menonaktifkan kerangka vektorisasi mendalam.
optimizer_use_sql_quarantine	`true | false`	`false`	Y	Mengaktifkan atau menonaktifkan pembuatan otomatis konfigurasi SQL Quarantine.
result_cache_execution_threshold	`0` ke `68719476736`	`2`	Y	Menentukan jumlah maksimum berapa kali fungsi PL/SQL dapat dieksekusi sebelum hasilnya disimpan dalam cache hasil.
result_cache_max_temp_result	`0` ke `100`	`5`	Y	Menentukan persentase `RESULT_CACHE_MAX_TEMP_SIZE` yang dapat dipakai oleh setiap hasil kueri cache.

Nama	Rentang nilai	Nilai default	Dapat diubah	Deskripsi
result_cache_max_temp_size	0 ke 2199023255552	RESULT_CACHE_SIZE * 10	Y	Menentukan jumlah maksimum tablespace sementara (dalam byte) yang dapat dipakai oleh cache hasil.
sga_min_size	0 ke 2199023255552 (nilai maksimumnya adalah 50% dari sga_target)	0	Y	Menunjukkan kemungkinan nilai minimum untuk penggunaan SGA dari basis data pluggable (PDB).
tablespace_encryption_default_algorithm	GOST256 | SEED128 | ARIA256 | ARIA192 | ARIA128 | 3DES168 | AES256 | AES192 | AES128	AES128	Y	Menentukan algoritma default yang digunakan basis data ketika mengenkripsi tablespace.

Perubahan untuk parameter yang kompatibel

Parameter `compatible` memiliki nilai maksimum baru untuk Oracle Database 21c (21.0.0.0) di Amazon RDS. Tabel berikut ini menunjukkan nilai default baru.

Nama parameter	Nilai maksimum Oracle Database 21c (21.0.0.0)
kompatibel	21.0.0

Parameter yang dihapus

Parameter berikut dihapus di Oracle Database 21c (21.0.0.0):

- `remote_os_authent`
- `sec_case_sensitive_logon`
- `unified_audit_sga_queue_size`

Oracle Database 19c dengan Amazon RDS

Amazon RDS mendukung Oracle Database 19c, yang mencakup Oracle Enterprise Editon dan Oracle Standard Edition Two.

Oracle Database 19c (19.0.0.0) mencakup banyak fitur baru dan pembaruan dari versi sebelumnya. Di bagian ini, Anda dapat menemukan fitur dan perubahan yang penting dalam penggunaan Oracle Database 19c (19.0.0.0) di Amazon RDS. Untuk daftar perubahan lengkapnya, lihat dokumentasi Oracle Database 19c. Untuk daftar lengkap fitur yang didukung oleh setiap edisi Oracle Database 19c, lihat Permitted features, options, and management packs by Oracle database offering dalam dokumentasi Oracle.

Perubahan parameter Amazon RDS for Oracle Database 19c (19.0.0.0)

Oracle Database 19c (19.0.0.0) menyertakan beberapa parameter baru dan parameter dengan rentang baru dan nilai default baru.

Topik

- Parameter baru
- Perubahan pada parameter yang kompatibel
- Parameter yang dihapus

Parameter baru

Tabel berikut ini menunjukkan parameter Amazon RDS yang baru untuk Oracle Database 19c (19.0.0.0).

Nama	Nilai	Dapat diubah	Deskripsi
lob_signature_enable	TRUE, FALSE (default)	Y	Mengaktifkan atau menonaktifkan fitur khas lokator LOB.
max_datapump_parallel_per_job	1 hingga 1024, atau AUTO	Y	Menentukan jumlah maksimum proses paralel yang diizinkan untuk setiap pekerjaan Data Pump Oracle.

Perubahan pada parameter yang kompatibel

Parameter `compatible` memiliki nilai maksimum baru untuk Oracle Database 19c (19.0.0.0) di Amazon RDS. Tabel berikut ini menunjukkan nilai default baru.

Nama parameter	Nilai maksimum Oracle Database 19c (19.0.0.0)
kompatibel	19.0.0

Parameter yang dihapus

Parameter berikut dihapus di Oracle Database 19c (19.0.0.0):

- `exafusion_enabled`
- `max_connections`
- `o7_dictionary_access`

Oracle Database 12c dengan Amazon RDS

Amazon RDS telah menghentikan dukungan untuk Oracle Database 12c di Oracle Enterprise Edition dan Oracle Standard Edition 2.

Topik

- Oracle Database 12c Rilis 2 (12.2.0.1) dengan Amazon RDS

- Oracle Database 12c Rilis 1 (12.1.0.2) dengan Amazon RDS

Oracle Database 12c Rilis 2 (12.2.0.1) dengan Amazon RDS

Pada 31 Maret 2022, Oracle Corporation menghentikan dukungan untuk Oracle Database 12c Rilis 2 (12.2.0.1) untuk BYOL dan LI. Pada tanggal tersebut, rilis dipindahkan dari Oracle Extended Support ke Oracle Sustaining Support, yang menunjukkan akhir dukungan untuk rilis ini. Untuk informasi selengkapnya, lihat jadwal akhir dukungan di AWS re:Post.

Tanggal	Tindakan
1 April 2022	Amazon RDS memulai peningkatan otomatis instans Oracle Database 12c Rilis 2 (12.2.0.1) ke Oracle Database 19c.
1 April 2022	Amazon RDS memulai peningkatan otomatis ke setiap instans DB Oracle Database 12c Rilis 2 (12.2.0.1) yang dipulihkan dari snapshot. Peningkatan otomatis terjadi dalam periode pemeliharaan. Jika periode pemeliharaan tidak tersedia saat peningkatan perlu dilakukan, Amazon RDS segera meningkatkan mesin tersebut.

Oracle Database 12c Rilis 1 (12.1.0.2) dengan Amazon RDS

Pada 31 Juli 2022, Amazon RDS menghentikan dukungan untuk Oracle Database 12c Rilis 1 (12.1.0.2) untuk BYOL dan LI. Rilis ini dipindahkan dari Oracle Extended Support ke Oracle Sustaining Support, yang menunjukkan bahwa Oracle Support tidak akan lagi merilis update patch penting untuk rilis ini. Untuk informasi selengkapnya, lihat jadwal akhir dukungan di AWS re:Post.

Tanggal	Tindakan
1 Agustus 2022	Amazon RDS memulai peningkatan otomatis instans Oracle Database 12c Rilis 1 (12.1.0.2) ke Release Update (RU) terbaru untuk Oracle Database 19c. Peningkatan otomatis terjadi dalam periode pemeliharaan. Jika periode pemeliharaan tidak tersedia saat peningkatan perlu dilakukan, Amazon RDS segera meningkatkan mesin tersebut.
1 Agustus 2022	Amazon RDS memulai peningkatan otomatis ke setiap instans DB Oracle Database 12c Rilis 1 (12.1.0.2) yang dipulihkan dari snapshot.

Opsi lisensi RDS for Oracle

Amazon RDS for Oracle memiliki dua opsi lisensi: Termasuk Lisensi (LI) dan Bawa Lisensi Sendiri (BYOL). Setelah Anda membuat instans DB Oracle di Amazon RDS, Anda dapat mengubah model lisensi dengan memodifikasi instans DB. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

> **Important**
>
> Pastikan Anda memiliki lisensi Oracle Database yang sesuai, dengan Dukungan dan Lisensi Pembaruan Perangkat Lunak, untuk kelas instans DB dan edisi Oracle Database. Pastikan juga Anda memiliki lisensi untuk setiap fitur Oracle Database berlisensi terpisah.

Topik

Termasuk Lisensi

Dalam model Lisensi Termasuk, Anda tidak perlu membeli lisensi Oracle Database secara terpisah. AWS memegang lisensi untuk perangkat lunak database Oracle. Dalam model ini, jika Anda memiliki AWS Support akun dengan dukungan kasus, hubungi AWS Support untuk permintaan layanan Amazon RDS dan Oracle Database. Model License Included hanya didukung di Amazon RDS for Oracle Database Standard Edition 2 (SE2). Penggunaan Anda atas RDS for Oracle, opsi LI, tunduk pada Bagian 10.3.1 dari Ketentuan Layanan AWS.

Bawa Lisensi Sendiri (BYOL)

Dalam model BYOL, Anda dapat menggunakan lisensi Oracle Database milik Anda yang ada untuk melakukan deployment basis data di Amazon RDS. Pastikan Anda memiliki lisensi Oracle Database yang sesuai (dengan Dukungan dan Lisensi Pembaruan Perangkat Lunak) untuk kelas instans DB dan edisi Oracle Database yang ingin Anda jalankan. Anda juga harus mengikuti kebijakan Oracle untuk lisensi perangkat lunak Oracle Database di lingkungan komputasi cloud. Untuk informasi lebih lanjut tentang kebijakan lisensi Oracle untuk Amazon EC2, lihat Lisensi perangkat lunak Oracle di lingkungan komputasi cloud.

Dalam model ini, Anda terus menggunakan akun dukungan Oracle aktif Anda, dan menghubungi Oracle secara langsung untuk permintaan layanan Oracle Database. Jika Anda memiliki AWS Support akun dengan dukungan kasus, Anda dapat menghubungi AWS Support untuk masalah Amazon RDS. Amazon Web Services dan Oracle memiliki proses dukungan multivendor untuk kasus-kasus yang memerlukan bantuan dari kedua organisasi.

Amazon RDS mendukung model BYOL hanya untuk Oracle Database Enterprise Edition (EE) dan Oracle Database Standard Edition 2 (SE2).

Integrasi dengan AWS License Manager

Untuk memudahkan pemantauan penggunaan lisensi Oracle dalam model BYOL, AWS License Manager berintegrasi dengan Amazon RDS for Oracle. License Manager mendukung pelacakan edisi mesin dan paket lisensi RDS for Oracle berdasarkan virtual core (vCPU). Anda juga dapat menggunakan License Manager AWS Organizations untuk mengelola semua akun organisasi Anda secara terpusat.

Tabel berikut menunjukkan filter informasi produk untuk RDS for Oracle.

Filter	Nama	Deskripsi
Edisi Mesin	`oracle-ee`	Oracle Database Enterprise Edition (EE)
	`oracle-se2`	Oracle Database Edisi Standar 2 (SE2)
Paket Lisensi	`data guard`	Lihat Menggunakan replika baca untuk Amazon RDS for Oracle (Oracle Active Data Guard)
	`olap`	Lihat Oracle OLAP
	`ols`	Lihat Keamanan Label Oracle
	`diagnostic pack sqlt`	Lihat Oracle SQLT
	`tuning pack sqlt`	Lihat Oracle SQLT

Untuk melacak penggunaan lisensi instans Oracle DB Anda, Anda dapat membuat lisensi yang dikelola sendiri. Dalam hal ini, RDS untuk sumber daya Oracle yang cocok dengan filter informasi

produk secara otomatis dikaitkan dengan lisensi yang dikelola sendiri. Penemuan instans DB Oracle dapat memakan waktu hingga 24 jam.

Konsol

Untuk membuat lisensi yang dikelola sendiri untuk melacak penggunaan lisensi instans Oracle DB Anda

1. Kunjungi https://console.aws.amazon.com/license-manager/.
2. Buat lisensi yang dikelola sendiri.

   Untuk petunjuk, lihat Membuat lisensi yang dikelola sendiri di Panduan AWS License Manager Pengguna.

   Tambahkan aturan untuk Filter Informasi Produk RDS dalam panel Informasi Produk.

   Untuk informasi selengkapnya, lihat ProductInformationdi Referensi AWS License Manager API.

AWS CLI

Untuk membuat lisensi yang dikelola sendiri dengan menggunakan AWS CLI, panggil create-license-configurationperintah. Gunakan parameter `--cli-input-json` atau `--cli-input-yaml` untuk meneruskan parameter ke perintah.

Example

Contoh berikut membuat lisensi yang dikelola sendiri untuk Oracle Enterprise Edition.

```
aws license-manager create-license-configuration --cli-input-json file://rds-oracle-ee.json
```

Berikut adalah sampel file `rds-oracle-ee.json` yang digunakan dalam contoh.

```
{
    "Name": "rds-oracle-ee",
    "Description": "RDS Oracle Enterprise Edition",
    "LicenseCountingType": "vCPU",
    "LicenseCountHardLimit": false,
    "ProductInformationList": [
        {
```

```
            "ResourceType": "RDS",
            "ProductInformationFilterList": [
                {
                    "ProductInformationFilterName": "Engine Edition",
                    "ProductInformationFilterValue": ["oracle-ee"],
                    "ProductInformationFilterComparator": "EQUALS"
                }
            ]
        }
    ]
}
```

Untuk informasi selengkapnya tentang informasi produk, lihat Penemuan otomatis inventaris sumber daya dalam Panduan Pengguna AWS License Manager .

Untuk informasi selengkapnya tentang *--cli-input* parameter, lihat Menghasilkan AWS CLI kerangka dan parameter input dari file input JSON atau YAMAL di Panduan Pengguna.AWS CLI

Bermigrasi antar edisi Oracle

Jika Anda memiliki lisensi Oracle BYOL tidak terpakai yang sesuai untuk edisi dan kelas instans DB yang rencananya akan Anda jalankan, Anda dapat bermigrasi dari Standard Edition 2 (SE2) ke Enterprise Edition (EE). Anda tidak dapat bermigrasi dari Enterprise Edition ke edisi lain.

Untuk mengubah edisi dan mempertahankan data Anda

1. Buat snapshot instans DB.

   Untuk informasi selengkapnya, lihat Membuat snapshot DB untuk instans DB Single-AZ.
2. Pulihkan snapshot ke instans DB baru, dan pilih edisi basis data Oracle yang ingin Anda gunakan.

   Untuk informasi selengkapnya, lihat Memulihkan dari snapshot DB.
3. (Opsional) Hapus instans DB lama, kecuali jika Anda ingin terus menjalankannya dan memiliki lisensi Oracle Database yang sesuai untuknya.

   Untuk informasi selengkapnya, lihat Menghapus instans DB.

Lisensi deployment Multi-AZ untuk Oracle

Amazon RDS mendukung deployment Multi-AZ untuk Oracle sebagai solusi failover dengan ketersediaan tinggi. Sebaiknya gunakan Multi-AZ untuk beban kerja produksi. Untuk informasi selengkapnya, lihat Mengonfigurasi dan mengelola deployment Multi-AZ.

Jika Anda menggunakan model Bawa Lisensi Sendiri, Anda harus memiliki lisensi untuk instans DB utama dan instans DB siaga dalam deployment Multi-AZ.

Pengguna dan hak istimewa RDS for Oracle

Saat Anda membuat instans DB Amazon RDS for Oracle, pengguna master default memiliki izin pengguna maksimum paling besar pada instans DB. Gunakan akun pengguna master untuk tugas administratif apa pun, seperti membuat akun pengguna tambahan dalam basis data Anda. Karena RDS adalah layanan terkelola, Anda tidak diizinkan masuk sebagai `SYS` dan`SYSTEM`, dan karenanya tidak memiliki hak istimewa `SYSDBA`.

Topik

- Keterbatasan untuk hak istimewa Oracle DBA
- Cara mengelola hak istimewa pada objek SYS

Keterbatasan untuk hak istimewa Oracle DBA

Dalam basis data, peran adalah sekumpulan hak istimewa yang dapat Anda berikan atau cabut dari pengguna. Basis data Oracle menggunakan peran untuk memberikan keamanan. Untuk informasi selengkapnya, lihat Configuring Privilege and Role Authorization dalam dokumentasi Oracle Database.

Peran `DBA` yang telah ditetapkan sebelumnya biasanya mengizinkan semua hak istimewa administratif di basis data Oracle. Saat Anda membuat instans DB, akun pengguna master Anda mendapatkan hak istimewa DBA (dengan beberapa batasan). Untuk memberikan pengalaman terkelola, basis data RDS for Oracle tidak menyediakan hak istimewa berikut untuk peran `DBA`:

- `ALTER DATABASE`
- `ALTER SYSTEM`
- `CREATE ANY DIRECTORY`
- `DROP ANY DIRECTORY`
- `GRANT ANY PRIVILEGE`

- `GRANT ANY ROLE`

Untuk informasi lebih lanjut tentang informasi peran dan hak istimewa sistem Oracle, lihat Hak akses akun pengguna master.

Cara mengelola hak istimewa pada objek SYS

Anda dapat mengelola hak istimewa pada objek `SYS` menggunakan paket `rdsadmin.rdsadmin_util`. Misalnya, jika Anda membuat pengguna basis data `myuser`, Anda dapat menggunakan prosedur `rdsadmin.rdsadmin_util.grant_sys_object` untuk memberikan hak istimewa `SELECT` di `V_$SQLAREA` kepada `myuser`. Untuk informasi selengkapnya, lihat topik berikut:

- Memberikan hak istimewa SELECT atau EXECUTE pada objek SYS
- Mencabut hak istimewa SELECT atau EXECUTE pada objek SYS
- Memberikan hak istimewa kepada pengguna non-master

Kelas instans RDS for Oracle

Kapasitas komputasi dan memori RDS untuk instans Oracle DB ditentukan oleh kelas instance-nya. Kelas instans basis data yang Anda butuhkan bergantung pada daya pemrosesan dan kebutuhan memori Anda.

Kelas instans RDS for Oracle yang didukung

Kelas instans RDS for Oracle yang didukung merupakan subset kelas instans basis data Amazon RDS. Lihat daftar lengkap kelas instans RDS di Kelas instans DB .

RDS untuk kelas instans yang dioptimalkan memori Oracle

RDS for Oracle juga menawarkan kelas instans yang dioptimalkan untuk beban kerja yang memerlukan memori, penyimpanan, dan I/O tambahan per vCPU. Kelas instans ini menggunakan konvensi penamaan berikut:

```
db.r5b.instance_size.tpcthreads_per_core.memratio
db.r5.instance_size.tpcthreads_per_core.memratio
```

Berikut ini contoh kelas instans yang didukung:

```
db.r5b.4xlarge.tpc2.mem2x
```

Komponen nama kelas instans sebelumnya adalah sebagai berikut:

- `db.r5b.4xlarge` – Nama kelas instans.
- `tpc2` – Thread per core. Nilai 2 berarti multithread diaktifkan. Jika nilainya 1, multithread dinonaktifkan.
- `mem2x` – Rasio memori tambahan pada memori standar untuk kelas instans. Dalam contoh ini, optimasi menyediakan memori dua kali lebih banyak sebagai instans db.r5.4xlarge standar.

Edisi yang didukung, kelas instance, dan kombinasi lisensi dalam RDS untuk Oracle

Jika Anda menggunakan konsol RDS, Anda dapat mengetahui apakah edisi tertentu, kelas instance, dan kombinasi lisensi didukung dengan memilih Buat database dan menentukan opsi yang berbeda. Di AWS CLI, Anda dapat menjalankan perintah berikut:

```
aws rds describe-orderable-db-instance-options --engine engine-type --license-
model license-type
```

Tabel berikut mencantumkan semua edisi, kelas instance, dan jenis lisensi yang didukung untuk RDS untuk Oracle. Rilis Oracle Database 12c Rilis 1 (12.1.0.2) dan Oracle Database 12c Rilis 2 (12.2.0.2) tercantum dalam tabel, tetapi dukungan untuk rilis sudah dihentikan. Untuk informasi tentang atribut memori tiap-tiap jenis, lihat Jenis instans RDS for Oracle. Untuk informasi tentang harga, lihat model harga Amazon RDS for Oracle.

Edisi Oracle	Oracle Database 19c dan yang lebih tinggi, Oracle Database 12c Rilis 2 (12.2.0.1) (sudah dihentikan)	Oracle Database 12c Rilis 1 (12.1.0.2) (sudah dihentikan)
Enterprise Edition (EE) Bawa Lisensi Sendiri (BYOL)	Kelas instans standar	
	db.m6i.large–db.m6i.32xlarge (khusus 19c) db.m5d.large–db.m5d.24xlarge db.m5.large–db.m5.24xlarge	db.m5.large–db.m5.24xlarge

Edisi Oracle	Oracle Database 19c dan yang lebih tinggi, Oracle Database 12c Rilis 2 (12.2.0.1) (sudah dihentikan)	Oracle Database 12c Rilis 1 (12.1.0.2) (sudah dihentikan)
	Kelas instans memori yang dioptimalkan	

Edisi Oracle	Oracle Database 19c dan yang lebih tinggi, Oracle Database 12c Rilis 2 (12.2.0.1) (sudah dihentikan)	Oracle Database 12c Rilis 1 (12.1.0.2) (sudah dihentikan)
	db.r6i.large–db.r6i.32xlarge (khusus 19c) db.r5d.large–db.r5d.24xlarge db.r5b.8xlarge.tpc2.mem3x db.r5b.6xlarge.tpc2.mem4x db.r5b.4xlarge.tpc2.mem4x db.r5b.4xlarge.tpc2.mem3x db.r5b.4xlarge.tpc2.mem2x db.r5b.2xlarge.tpc2.mem8x db.r5b.2xlarge.tpc2.mem4x db.r5b.2xlarge.tpc1.mem2x db.r5b.xlarge.tpc2.mem4x db.r5b.xlarge.tpc2.mem2x db.r5b.large.tpc1.mem2x db.r5b.large–db.r5b.24xlarge db.r5.12xlarge.tpc2.mem2x db.r5.8xlarge.tpc2.mem3x db.r5.6xlarge.tpc2.mem4x db.r5.4xlarge.tpc2.mem4x db.r5.4xlarge.tpc2.mem3x	db.r5.12xlarge.tpc2.mem2x db.r5b.large–db.r5b.24xlarge db.r5.8xlarge.tpc2.mem3x db.r5.6xlarge.tpc2.mem4x db.r5.4xlarge.tpc2.mem4x db.r5.4xlarge.tpc2.mem3x db.r5.4xlarge.tpc2.mem2x db.r5.2xlarge.tpc2.mem8x db.r5.2xlarge.tpc2.mem4x db.r5.2xlarge.tpc1.mem2x db.r5.xlarge.tpc2.mem4x db.r5.xlarge.tpc2.mem2x db.r5.large.tpc1.mem2x db.r5.large–db.r5.24xlarge db.x1e.xlarge–db.x1e.32xlarge db.x1.16xlarge–db.x1.32xlarge db.z1d.large–db.z1d.12xlarge

Edisi Oracle	Oracle Database 19c dan yang lebih tinggi, Oracle Database 12c Rilis 2 (12.2.0.1) (sudah dihentikan)	Oracle Database 12c Rilis 1 (12.1.0.2) (sudah dihentikan)
	db.r5.4xlarge.tpc2.mem2x db.r5.2xlarge.tpc2.mem8x db.r5.2xlarge.tpc2.mem4x db.r5.2xlarge.tpc1.mem2x db.r5.xlarge.tpc2.mem4x db.r5.xlarge.tpc2.mem2x db.r5.large.tpc1.mem2x db.r5.large–db.r5.24xlarge db.x2iedn.xlarge–db.x2iedn.32xlarge db.x2iezn.2xlarge–db.x2iezn.12xlarge db.x2idn.16xlarge–db.x2idn.32xlarge db.x1e.xlarge–db.x1e.32xlarge db.x1.16xlarge–db.x1.32xlarge db.z1d.large–db.z1d.12xlarge	
	Kelas instans performa yang dapat melonjak	
	db.t3.small–db.t3.2xlarge	db.t3.micro–db.t3.2xlarge

Edisi Oracle	Oracle Database 19c dan yang lebih tinggi, Oracle Database 12c Rilis 2 (12.2.0.1) (sudah dihentikan)	Oracle Database 12c Rilis 1 (12.1.0.2) (sudah dihentikan)
Standard Edition 2 (SE2) Bawa Lisensi Sendiri (BYOL)	Kelas instans standar	
	db.m6i.besar—db.m6i.4xlarge db.m5d.large–db.m5d.4xlarge db.m5.large–db.m5.4xlarge	db.m5.large–db.m5.4xlarge
	Kelas instans memori yang dioptimalkan	

Edisi Oracle	Oracle Database 19c dan yang lebih tinggi, Oracle Database 12c Rilis 2 (12.2.0.1) (sudah dihentikan)	Oracle Database 12c Rilis 1 (12.1.0.2) (sudah dihentikan)
	db.r6i.large–db.r6i.4xlarge (khusus 19c) db.r5d.large–db.r5d.4xlarge db.r5.4xlarge.tpc2.mem4x db.r5.4xlarge.tpc2.mem3x db.r5.4xlarge.tpc2.mem2x db.r5.2xlarge.tpc2.mem8x db.r5.2xlarge.tpc2.mem4x db.r5.2xlarge.tpc1.mem2x db.r5.xlarge.tpc2.mem4x db.r5.xlarge.tpc2.mem2x db.r5.large.tpc1.mem2x db.r5.large–db.r5.4xlarge db.r5b.large–db.r5b.4xlarge db.x2iedn.xlarge–db.x2iedn.4xlarge db.x2iezn.2xlarge–db.x2iezn.4xlarge db.z1d.large–db.z1d.3xlarge	db.r5.4xlarge.tpc2.mem4x db.r5.4xlarge.tpc2.mem3x db.r5.4xlarge.tpc2.mem2x db.r5.2xlarge.tpc2.mem8x db.r5.2xlarge.tpc2.mem4x db.r5.2xlarge.tpc1.mem2x db.r5.xlarge.tpc2.mem4x db.r5.xlarge.tpc2.mem2x db.r5.large.tpc1.mem2x db.r5.large–db.r5.4xlarge db.r5b.large–db.r5b.4xlarge db.z1d.large–db.z1d.3xlarge
	Kelas-kelas instans dengan performa yang dapat melonjak	
	db.t3.small–db.t3.2xlarge	db.t3.micro–db.t3.2xlarge

Edisi Oracle	Oracle Database 19c dan yang lebih tinggi, Oracle Database 12c Rilis 2 (12.2.0.1) (sudah dihentikan)	Oracle Database 12c Rilis 1 (12.1.0.2) (sudah dihentikan)
Standard Edition 2 (SE2) Termasuk Lisensi	Kelas instans standar	
	db.m5.large–db.m5.4xlarge	db.m5.large–db.m5.4xlarge
	Kelas instans memori yang dioptimalkan	
	db.r6i.large–db.r6i.4xlarge (khusus 19c) db.r5.large–db.r5.4xlarge	db.r5.large–db.r5.4xlarge
	Kelas-kelas instans dengan performa yang dapat melonjak	
	db.t3.small–db.t3.2xlarge	db.t3.micro–db.t3.2xlarge

> **Note**
>
> Kami mendorong semua pelanggan BYOL untuk mempelajari perjanjian lisensi mereka guna menilai dampak dari penghentian Amazon RDS for Oracle. Untuk informasi selengkapnya tentang kapasitas komputasi kelas instans yang DB didukung oleh RDS for Oracle, lihat Kelas instans DB dan Mengonfigurasi prosesor untuk kelas instans DB di RDS for Oracle.

> **Note**
>
> Jika Anda memiliki snapshot DB dari instans DB yang menggunakan kelas instans DB yang sudah dihentikan, Anda dapat memilih kelas instans DB yang belum dihentikan saat Anda memulihkan snapshot DB. Untuk informasi selengkapnya, lihat Memulihkan dari snapshot DB.

RDS usang untuk kelas instans Oracle DB

Kelas instans DB berikut sudah dihentikan untuk RDS for Oracle:

- db.m1, db.m2, db.m3, db.m4

- db.t3.micro (hanya didukung pada 12.1.0.2, yang sudah dihentikan)
- db.t1, db.t2
- db.r1, db.r2, db.r3, db.r4

Pemrosesan kelas instans DB ini telah diganti dengan kelas instans DB yang berperforma lebih baik, yang umumnya tersedia dengan biaya lebih rendah. Jika Anda memiliki instans DB yang menggunakan kelas instans DB yang sudah dihentikan, Anda memiliki opsi berikut:

- Izinkan Amazon RDS memodifikasi setiap instans DB secara otomatis untuk menggunakan kelas instans DB yang sebanding dan belum dihentikan. Lihat jadwal penghentian di Jenis kelas instans DB.
- Ubah kelas instans DB sendiri dengan memodifikasi instans DB. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Jika Anda memiliki snapshot DB dari instans DB yang menggunakan kelas instans DB yang sudah dihentikan, Anda dapat memilih kelas instans DB yang belum dihentikan saat Anda memulihkan snapshot DB. Untuk informasi selengkapnya, lihat Memulihkan dari snapshot DB.

Arsitektur basis data RDS for Oracle

Arsitektur multi-penghuni Oracle, yang juga dikenal sebagai arsitektur CDB, memungkinkan basis data Oracle berfungsi sebagai basis data kontainer multi-penghuni (CDB). CDB dapat mencakup basis data pluggable (PDB) yang dibuat oleh pelanggan. Non-CDB adalah basis data Oracle yang menggunakan arsitektur tradisional, yang tidak dapat berisi PDB. Untuk informasi selengkapnya tentang arsitektur multi-penghuni, lihat Oracle Multitenant Administrator's Guide.

Untuk Oracle Database 19c dan yang lebih tinggi, Anda dapat membuat instans DB RDS for Oracle yang menggunakan arsitektur CDB. Aplikasi klien Anda terhubung pada tingkat PDB, bukan CDB. RDS for Oracle mendukung konfigurasi arsitektur CDB sebagai berikut:

Konfigurasi multi-penghuni

Fitur platform RDS ini memungkinkan RDS untuk instance CDB Oracle berisi antara 1-30 database penyewa, tergantung pada edisi database dan opsi apa pun yang diperlukan lisensi database penyewa (PDB). Konfigurasi multi-penghuni tidak mendukung PDB aplikasi atau proksi PDB. Anda dapat menggunakan API RDS untuk menambah, memodifikasi, dan menghapus basis data penghuni.

> **Note**
>
> Fitur Amazon RDS disebut "multi-penghuni" dan bukannya "multipenghuni" karena merupakan kemampuan platform RDS, bukan hanya mesin Oracle DB. Istilah "Oracle multipenghuni" mengacu secara eksklusif ke arsitektur basis data Oracle, yang kompatibel dengan deployment RDS dan on-premise.

Konfigurasi satu penghuni

Fitur platform RDS ini membatasi RDS untuk instance CDB Oracle ke 1 database penyewa (PDB). Anda tidak dapat menambahkan lebih banyak PDB menggunakan RDS API. Konfigurasi satu penghuni menggunakan RDS API yang sama dengan arsitektur non-CDB. Oleh karena itu, pengalaman penggunaan CDB dalam konfigurasi satu penghuni hampir sama dengan pengalaman penggunaan non-CDB.

Anda dapat mengonversi CDB yang menggunakan konfigurasi penyewa tunggal ke konfigurasi multi-penyewa, sehingga memungkinkan Anda menambahkan PDB ke CDB Anda. Perubahan arsitektur ini bersifat permanen dan tidak dapat diubah. Untuk informasi selengkapnya, lihat Mengonversi konfigurasi satu penghuni menjadi multi-penghuni.

> **Note**
>
> Anda tidak dapat mengakses CDB itu sendiri.

Di Oracle Database 21c dan yang lebih tinggi, semua basis data adalah CDB. Di sisi lain, Anda dapat membuat instans DB Oracle Database 19c sebagai CDB atau non-CDB. Anda tidak dapat melakukan peningkatan dari non-CDB ke CDB, tetapi Anda dapat mengonversi Oracle Database 19c non-CDB ke CDB, kemudian meningkatkannya. Anda tidak dapat mengonversi CDB ke non-CDB.

Untuk informasi selengkapnya, lihat sumber daya berikut:

- Bekerja dengan CDB di RDS for Oracle
- Batasan CDB RDS for Oracle
- Membuat instans DB Amazon RDS

Parameter RDS for Oracle

Grup parameter DB

Di Amazon RDS, Anda mengelola parameter menggunakan grup parameter DB. Untuk informasi selengkapnya, lihat Bekerja dengan grup parameter. Untuk melihat parameter inisialisasi yang didukung untuk edisi dan versi Oracle Database tertentu, jalankan perintah. AWS CLI `describe-engine-default-parameters`

Misalnya, untuk melihat parameter inisialisasi yang didukung untuk Enterprise Edition dari Oracle Database 19c, jalankan perintah berikut.

```
aws rds describe-engine-default-parameters \
    --db-parameter-group-family oracle-ee-19
```

Parameter inisialisasi database Oracle

Untuk menemukan dokumentasi untuk parameter inisialisasi, lihat Parameter Inisialisasi dalam dokumentasi Oracle Database. Parameter inisialisasi berikut memiliki pertimbangan khusus:

- `ARCHIVE_LAG_TARGET`

  Parameter ini memaksa tombol redo log setelah waktu yang ditentukan berlalu. Dalam RDS untuk Oracle, `ARCHIVE_LAG_TARGET` diatur ke `300` karena tujuan titik pemulihan (RPO) adalah 5 menit. Untuk menghormati tujuan ini, RDS untuk Oracle mengganti log pengulangan online setiap 5 menit dan menyimpannya dalam ember Amazon S3. Jika frekuensi sakelar log menyebabkan masalah kinerja untuk database RDS for Oracle, Anda dapat menskalakan instans dan penyimpanan DB Anda ke yang memiliki IOPS dan throughput yang lebih tinggi. Atau, jika Anda menggunakan RDS Custom for Oracle atau menyebarkan database Oracle di Amazon EC2, Anda dapat menyesuaikan pengaturan parameter inisialisasi. `ARCHIVE_LAG_TARGET`

Set karakter RDS for Oracle

RDS for Oracle mendukung dua jenis set karakter: set karakter DB dan set karakter nasional.

Set karakter DB

Set karakter basis data Oracle digunakan di jenis data `CHAR`, `VARCHAR2`, and `CLOB`. Basis data juga menggunakan set karakter untuk metadata seperti nama tabel, nama kolom, dan pernyataan SQL. Set karakter basis data Oracle biasanya disebut sebagai set karakter DB.

Anda menetapkan set karakter saat Anda membuat instans DB. Anda tidak dapat mengubah set karakter DB setelah Anda membuat basis data.

Set karakter DB yang didukung

Tabel berikut mencantumkan set karakter Oracle DB yang didukung di Amazon RDS. Anda dapat menggunakan nilai dari tabel ini dengan parameter `--character-set-name` dari perintah AWS CLI create-db-instance atau dengan parameter `CharacterSetName` dari operasi API Amazon RDS CreateDBInstance.

> **Note**
>
> Set karakter untuk CDB selalu AL32UTF8. Anda dapat menetapkan set karakter yang berbeda untuk PDB saja.

Nilai	Deskripsi
AL32UTF8	Unicode 5.0 UTF-8 Set karakter universal (default)
AR8ISO8859P6	ISO 8859-6 Latin/Arab
AR8MSWIN1256	Microsoft Windows Code Page 1256 8-bit Latin/Arab
BLT8ISO8859P13	ISO 8859-13 Baltik
BLT8MSWIN1257	Microsoft Windows Code Page 1257 8-bit Baltik
CL8ISO8859P5	ISO 88559-5 Latin/Sirilik
CL8MSWIN1251	Microsoft Windows Code Page 1251 8-bit Latin/Sirilik

Nilai	Deskripsi
EE8ISO8859P2	ISO 8859-2 Eropa Timur
EL8ISO8859P7	ISO 8859-7 Latin/Yunani
EE8MSWIN1250	Microsoft Windows Code Page 1250 8-bit Eropa Timur
EL8MSWIN1253	Microsoft Windows Code Page 1253 8-bit Latin/ Yunani
IW8ISO8859P8	ISO 8859-8 Latin/Ibrani
IW8MSWIN1255	Microsoft Windows Code Page 1255 8-bit Latin/ Ibrani
JA16EUC	EUC 24-bit Jepang
JA16EUCTILDE	Sama seperti JA16EUC kecuali untuk pemetaan tanda hubung ombak dan tanda gelombang ke dan dari Unicode
JA16SJIS	Shift-JIS 16-bit Jepang
JA16SJISTILDE	Sama seperti JA16SJIS kecuali untuk pemetaan tanda hubung ombak dan tanda gelombang ke dan dari Unicode
KO16MSWIN949	Microsoft Windows Code Page 949 Korea
NE8ISO8859P10	ISO 8859-10 Eropa Utara
NEE8ISO8859P4	ISO 8859-4 Eropa Utara dan Timur Laut
TH8TISASCII	Thai Industrial Standard 620-2533-ASCII 8-bit
TR8MSWIN1254	Microsoft Windows Code Page 1254 8-bit Turki
US7ASCII	ASCII 7-bit Amerika

Nilai	Deskripsi
UTF8	Unicode 3.0 UTF-8 Set karakter universal, mematuhi CESU-8
VN8MSWIN1258	Microsoft Windows Code Page 1258 8-bit Vietnam
WE8ISO8859P1	Eropa Barat 8-bit ISO 8859 Part 1
WE8ISO8859P15	ISO 8859-15 Eropa Barat
WE8ISO8859P9	ISO 8859-9 Eropa Barat dan Turki
WE8MSWIN1252	Microsoft Windows Code Page 1252 8-bit Eropa Barat
ZHS16GBK	GBK 16-bit Tiongkok Aksara Sederhana
ZHT16HKSCS	Micro Windows Code Halaman 950 dengan Set Karakter Tambahan Hong Kong HKSCS-2001. Konversi set karakter didasarkan pada Unicode 3.0.
ZHT16MSWIN950	Microsoft Windows Code Page 950 Tiongkok Aksara Tradisional
ZHT32EUC	EUC 32-bit Tiongkok Aksara Tradisional

Variabel lingkungan NLS_LANG

Lokal adalah serangkaian informasi yang membahas persyaratan bahasa dan budaya yang sesuai dengan bahasa dan negara tertentu. Mengatur variabel lingkungan NLS_LANG di lingkungan klien Anda adalah cara paling sederhana untuk menentukan perilaku lokal untuk Oracle. Variabel ini mengatur bahasa dan wilayah yang digunakan oleh aplikasi klien dan server basis data. Variabel ini juga menunjukkan set karakter klien, yang sesuai dengan set karakter untuk data yang dimasukkan atau ditampilkan oleh aplikasi klien. Untuk informasi lebih lanjut tentang NLS_LANG dan set karakter, lihat What is a character set or code page? dalam dokumentasi Oracle.

Parameter inisialisasi NLS

Anda juga dapat mengatur parameter inisialisasi National Language Support (NLS) berikut di tingkat instans untuk instans DB Oracle di Amazon RDS:

- NLS_DATE_FORMAT
- NLS_LENGTH_SEMANTICS
- NLS_NCHAR_CONV_EXCP
- NLS_TIME_FORMAT
- NLS_TIME_TZ_FORMAT
- NLS_TIMESTAMP_FORMAT
- NLS_TIMESTAMP_TZ_FORMAT

Untuk informasi tentang cara mengubah parameter instans, lihat Bekerja dengan grup parameter.

Anda dapat menetapkan parameter inisialisasi NLS lain di klien SQL Anda. Misalnya, pernyataan berikut menetapkan parameter inisialisasi NLS_LANGUAGE ke GERMAN di klien SQL yang tersambung ke instans DB Oracle:

```
ALTER SESSION SET NLS_LANGUAGE=GERMAN;
```

Untuk informasi tentang cara menghubungkan ke instans DB Oracle dengan klien SQL, lihat Menghubungkan ke instans RDS for Oracle DB.

Set karakter nasional

Set karakter nasional digunakan dalam jenis data `NCHAR`, `NVARCHAR2`, dan `NCLOB`. Set karakter nasional biasanya disebut sebagai Set karakter NCHAR. Tidak seperti set karakter DB, set karakter NCHAR tidak memengaruhi metadata basis data.

Set karakter NCHAR mendukung set karakter berikut:

- AL16UTF16 (default)
- UTF8

Anda dapat menentukan nilai dengan parameter `--nchar-character-set-name` dari perintah create-db-instance (versi AWS CLI 2 saja). Jika Anda menggunakan Amazon RDS API, tentukan

parameter `NcharCharacterSetName` dari operasi CreateDBInstance. Anda tidak dapat mengubah set karakter nasional setelah Anda membuat basis data.

Untuk informasi lebih lanjut tentang Unicode dalam basis data Oracle, lihat Supporting multilingual databases with unicode dalam dokumentasi Oracle.

Batasan RDS for Oracle

Pada bagian berikut, Anda dapat menemukan batasan penting dalam menggunakan RDS for Oracle. Batasan spesifik untuk CDB dapat dilihat di Batasan CDB RDS for Oracle.

> **Note**
>
> Daftar ini tidak lengkap.

Topik

- Batas ukuran file Oracle di Amazon RDS
- Sinonim publik untuk skema yang disediakan Oracle
- Skema untuk fitur yang tidak didukung
- Keterbatasan untuk hak istimewa Oracle DBA
- Penghentian TLS 1.0 dan Keamanan Lapisan Pengangkutan 1.1

Batas ukuran file Oracle di Amazon RDS

Ukuran maksimum dari satu file di instans DB RDS for Oracle adalah 16 TiB (tebibyte). Batas ini diberlakukan oleh sistem file ext4 yang digunakan oleh instans. Dengan demikian, file data Oracle bigfile dibatasi hingga 16 TiB. Jika Anda mencoba mengubah ukuran file data di tablespace bigfile ke nilai yang melebihi batas, Anda menerima pesan kesalahan seperti berikut.

```
ORA-01237: cannot extend datafile 6
ORA-01110: data file 6: '/rdsdbdata/db/mydir/datafile/myfile.dbf'
ORA-27059: could not reduce file size
Linux-x86_64 Error: 27: File too large
Additional information: 2
```

Sinonim publik untuk skema yang disediakan Oracle

Jangan membuat atau memodifikasi sinonim publik untuk skema yang disediakan Oracle, termasuk SYS, SYSTEM, dan RDSADMIN. Tindakan tersebut dapat mengakibatkan invalidasi komponen basis data inti dan memengaruhi ketersediaan instans DB Anda.

Anda dapat membuat sinonim publik yang merujuk pada objek dalam skema Anda sendiri.

Skema untuk fitur yang tidak didukung

Secara umum, Amazon RDS tidak mencegah Anda membuat skema untuk fitur yang tidak didukung. Namun, jika Anda membuat skema untuk fitur dan komponen Oracle yang memerlukan hak istimewa SYS, Anda dapat merusak kamus data dan memengaruhi ketersediaan instans Anda. Hanya gunakan fitur yang didukung dan skema yang tersedia di Menambahkan opsi untuk instans DB Oracle.

Keterbatasan untuk hak istimewa Oracle DBA

Dalam basis data, peran adalah sekumpulan hak istimewa yang dapat Anda berikan atau cabut dari pengguna. Basis data Oracle menggunakan peran untuk memberikan keamanan.

Peran DBA yang telah ditetapkan sebelumnya biasanya mengizinkan semua hak istimewa administratif di basis data Oracle. Saat Anda membuat instans DB, akun pengguna master Anda mendapatkan hak istimewa DBA (dengan beberapa batasan). Untuk memberikan pengalaman terkelola, basis data RDS for Oracle database tidak menyediakan hak istimewa berikut untuk peran DBA:

- ALTER DATABASE
- ALTER SYSTEM
- CREATE ANY DIRECTORY
- DROP ANY DIRECTORY
- GRANT ANY PRIVILEGE
- GRANT ANY ROLE

Gunakan akun pengguna master untuk tugas administratif, seperti membuat akun pengguna tambahan dalam basis data. Anda tidak dapat menggunakan SYS, SYSTEM, dan akun administratif lain yang diberikan oleh Oracle.

Penghentian TLS 1.0 dan Keamanan Lapisan Pengangkutan 1.1

Protokol Keamanan Lapisan Pengangkutan versi 1.0 dan 1.1 (TLS 1.0 dan TLS 1.1) sudah dihentikan. Sesuai dengan praktik keamanan terbaik, Oracle sudah menghentikan penggunaan TLS 1.0 dan TLS 1.1. Untuk memenuhi persyaratan keamanan Anda, RDS for Oracle sangat menganjurkan agar Anda menggunakan TLS 1.2 sebagai gantinya.

Menghubungkan ke instans RDS for Oracle DB

Setelah Amazon RDS menyediakan instans Oracle DB, Anda dapat menggunakan aplikasi klien SQL standar untuk masuk ke instans DB. Karena RDS adalah layanan terkelola, Anda tidak dapat masuk sebagai SYS atau SYSTEM. Untuk mengetahui informasi selengkapnya, lihat Pengguna dan hak istimewa RDS for Oracle.

Dalam topik ini, Anda akan mempelajari cara menggunakan Oracle SQL Developer atau SQL*Plus untuk terhubung ke instans RDS for Oracle DB. Sebagai contoh yang akan memandu Anda melalui proses membuat dan menghubungkan ke instans DB sampel, lihat Membuat dan menghubungkan ke instans DB Oracle.

Topik

Menemukan titik akhir instans DB RDS for Oracle

Setiap instans basis data Amazon RDS memiliki titik akhir, dan setiap titik akhir memiliki nama dan nomor porta DNS untuk instans basis data. Untuk menghubungkan instans basis Anda menggunakan aplikasi klien SQL, Anda memerlukan nama dan nomor porta DNS untuk instans tersebut.

Anda dapat menemukan titik akhir instans DB menggunakan konsol Amazon RDS atau AWS CLI.

> **Note**
>
> Jika Anda menggunakan autentikasi Kerberos, lihat Menghubungkan ke Oracle dengan autentikasi Kerberos.

Konsol

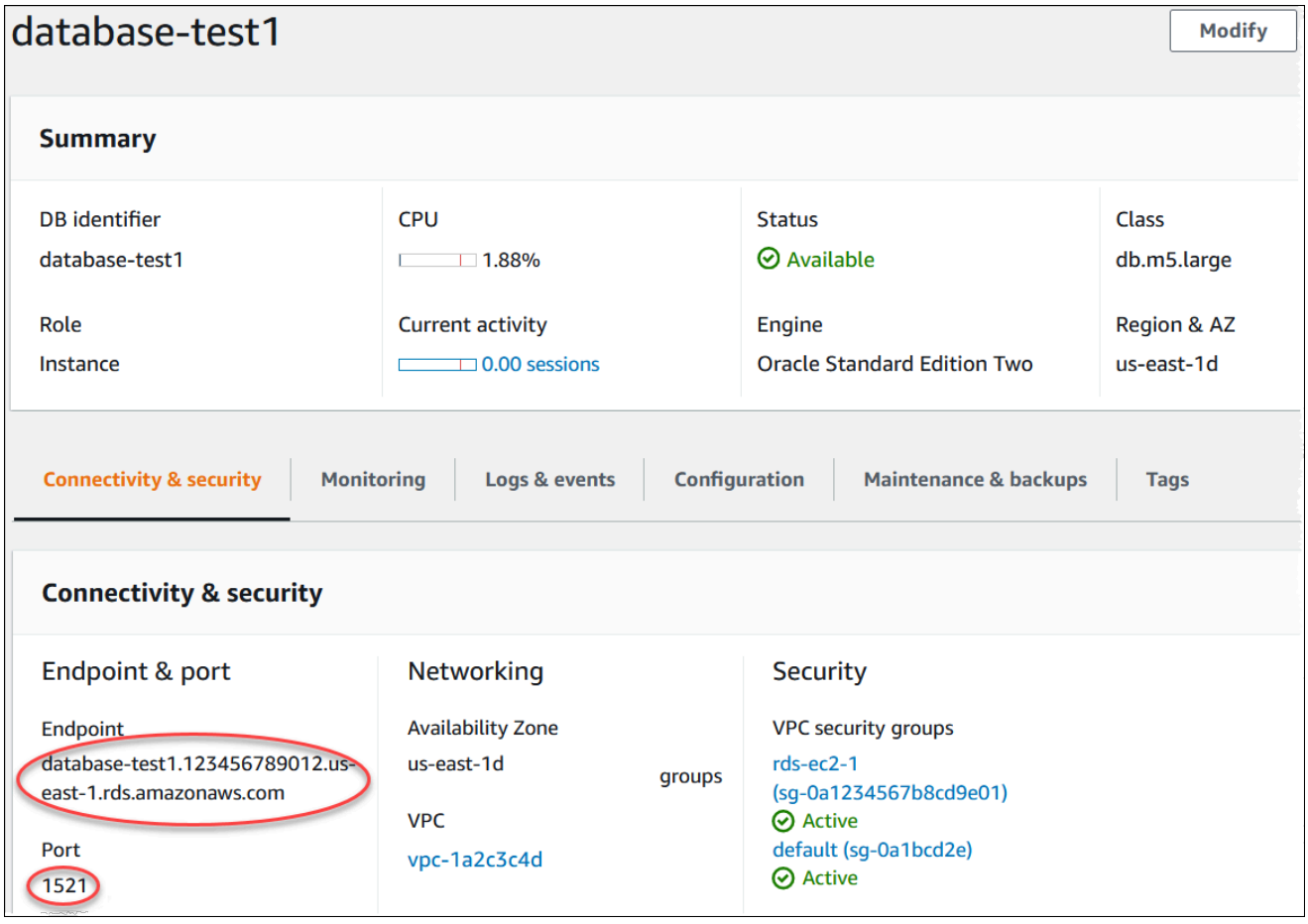
Untuk menemukan titik akhir menggunakan konsol

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di sudut kanan atas konsol, pilih Wilayah AWS untuk instans DB Anda.
3. Temukan nama DNS dan nomor port untuk instans DB Anda.

   a. Pilih Basis data untuk menampilkan daftar instans basis data Anda.

   b. Pilih nama instans DB Oracle untuk menampilkan detail instans.

   c. Di tab Konektivitas & keamanan, salin titik akhir. Selain itu, catat nomor port. Anda memerlukan titik akhir dan nomor port untuk terhubung ke instans DB.



AWS CLI

Untuk menemukan titik akhir instans DB Oracle menggunakan AWS CLI, panggil perintah describe-db-instances.

Example Untuk menemukan titik akhir menggunakan AWS CLI

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

Cari `Endpoint` dalam output untuk menemukan nama DNS dan nomor port instans DB Anda. Baris `Address` di dalam output berisi nama DNS. Berikut ini adalah contoh output titik akhir JSON.

```
"Endpoint": {
    "HostedZoneId": "Z1PVIF0B656C1W",
    "Port": 3306,
    "Address": "myinstance.123456789012.us-west-2.rds.amazonaws.com"
},
```

> **Note**
>
> Output dapat berisi informasi untuk beberapa instans DB.

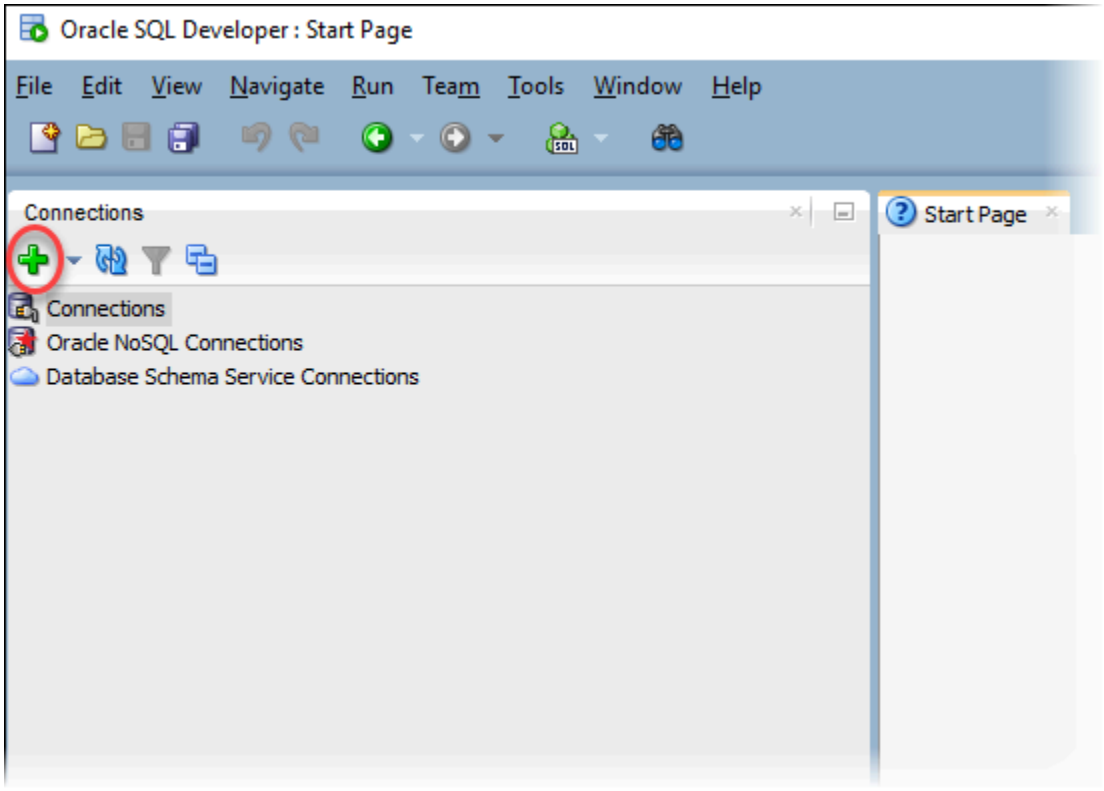
Menghubungkan ke instans DB menggunakan pengembang Oracle SQL

Dalam prosedur ini, Anda terhubung ke instans DB menggunakan Oracle SQL Developer. Untuk mengunduh versi mandiri utilitas ini, lihat Halaman unduhan pengembang Oracle SQL.

Untuk terhubung ke instans DB, Anda memerlukan nama DNS dan nomor port. Untuk mengetahui informasi tentang cara menemukan nama DNS dan nomor port untuk instans DB, lihat Menemukan titik akhir instans DB RDS for Oracle.

Untuk terhubung ke instans DB menggunakan pengembang SQL

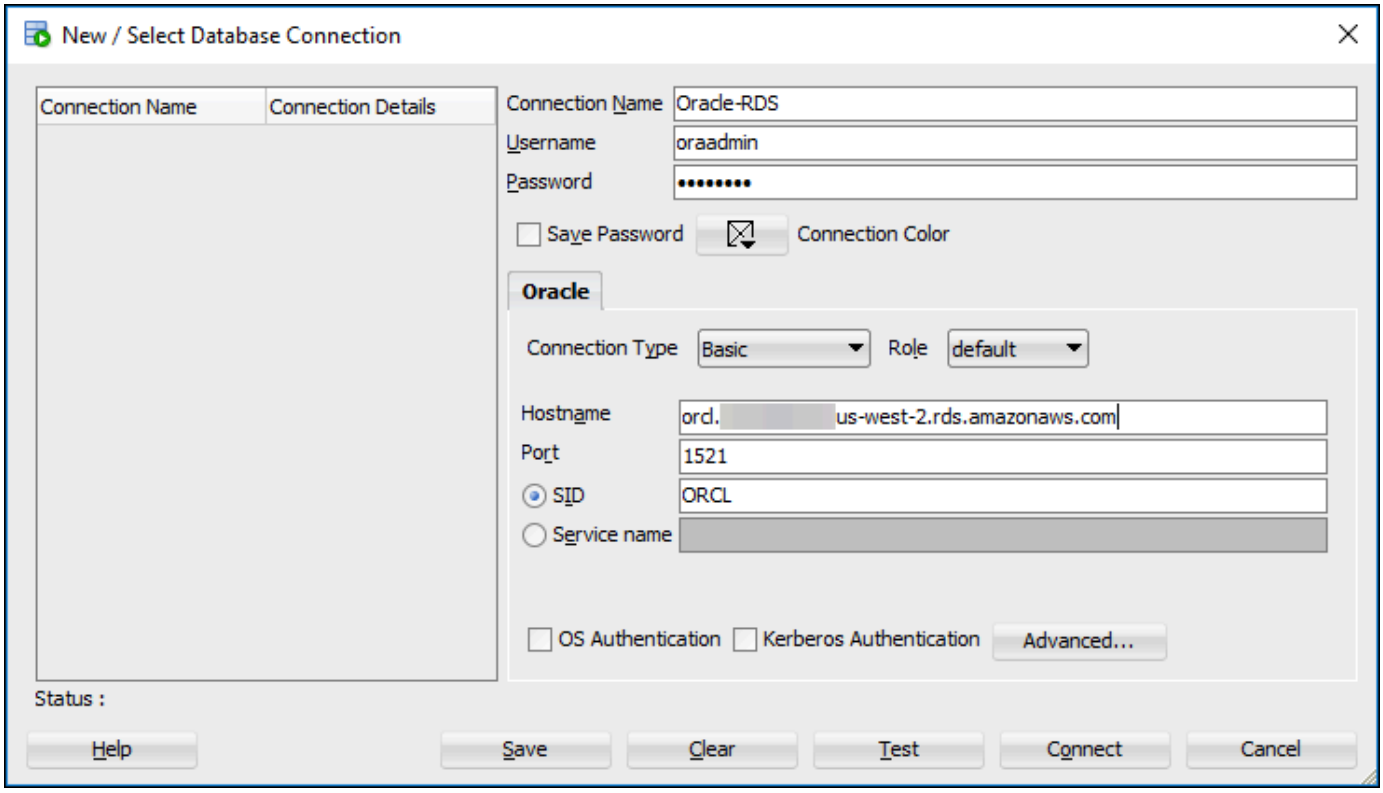
1. Mulai Oracle SQL Developer.
2. Di tab Koneksi, pilih ikon tambahkan (+).



3. Berikan informasi untuk instans DB Anda di kotak dialog Koneksi Basis Data Baru/Pilihan:

   - Untuk Nama Koneksi, masukkan nama yang menjelaskan koneksi, seperti `Oracle-RDS`.
   - Untuk Nama Pengguna, masukkan nama administrator basis data untuk instans DB.
   - Untuk Kata Sandi, masukkan kata sandi untuk administrator basis data.
   - Untuk Nama Host, masukkan nama DNS instans DB.
   - Untuk Port, masukkan nomor port.
   - Untuk SID, masukkan nama DB. Anda dapat menemukan nama DB pada tab Konfigurasi halaman detail basis data Anda.

   Kotak dialog yang lengkap akan terlihat mirip dengan yang berikut ini.

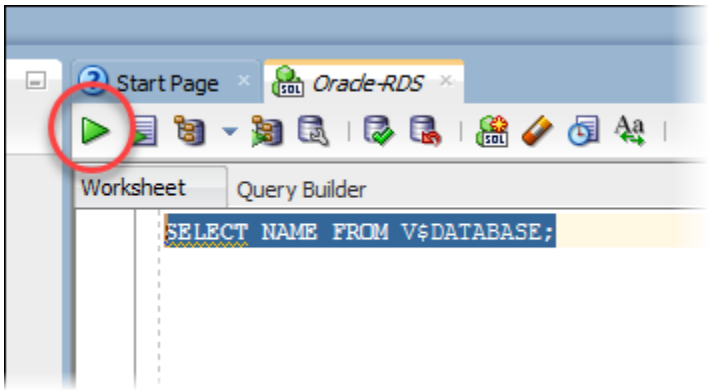


4. Pilih Hubungkan.
5. Sekarang Anda dapat mulai membuat basis data sendiri dan menjalankan kueri pada instans DB dan basis data Anda seperti biasa. Untuk menjalankan kueri pengujian pada instans DB Anda, lakukan tindakan berikut:

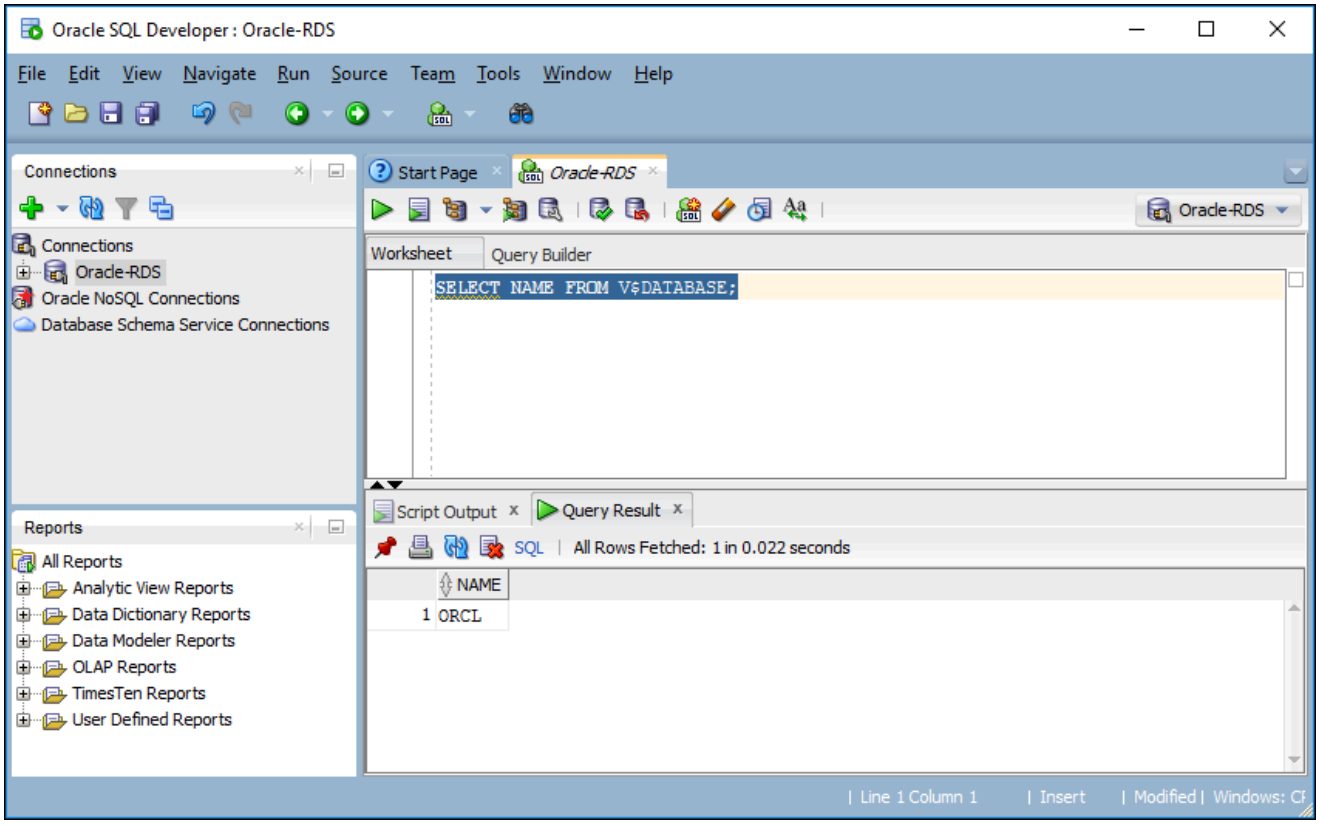
   a. Di tab Lembar Kerja untuk koneksi Anda, masukkan kueri SQL berikut.

   ```
   SELECT NAME FROM V$DATABASE;
   ```

   b. Pilih ikon jalankan untuk menjalankan kueri.

   SQL Developer menampilkan nama basis data.



Menghubungkan ke instans DB menggunakan SQL*Plus

Anda dapat menggunakan utilitas seperti SQL*Plus untuk terhubung ke instans Amazon RDS DB yang menjalankan Oracle. Untuk mengunduh Oracle Instant Client, yang mencakup versi mandiri SQL*Plus, lihat Unduhan Oracle Instant Client.

Untuk terhubung ke instans DB, Anda memerlukan nama DNS dan nomor port. Untuk mengetahui informasi tentang cara menemukan nama DNS dan nomor port untuk instans DB, lihat Menemukan titik akhir instans DB RDS for Oracle.

Example Untuk terhubung ke instans Oracle DB menggunakan SQL*Plus

Dalam contoh berikut, ganti nama pengguna administrator instans DB Anda. Selain itu, ganti titik akhir instans DB Anda, lalu sertakan nomor port dan SID Oracle. Nilai SID adalah nama basis data instans DB yang Anda tentukan saat membuat instans DB, bukan nama instans DB.

Untuk Linux, macOS, atau Unix:

```
sqlplus 'user_name@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=dns_name)(PORT=port))
(CONNECT_DATA=(SID=database_name)))'
```

Untuk Windows:

```
sqlplus user_name@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=dns_name)(PORT=port))
(CONNECT_DATA=(SID=database_name)))
```

Output Anda akan terlihat seperti berikut ini.

```
SQL*Plus: Release 12.1.0.2.0 Production on Mon Aug 21 09:42:20 2017
```

Perintah SQL akan muncul setelah Anda memasukkan kata sandi pengguna.

```
SQL>
```

> **Note**
>
> String koneksi format yang lebih pendek (EZ Connect), seperti `sqlplus USER/PASSWORD@longer-than-63-chars-rds-endpoint-here:1521/database-identifier`, mungkin mengalami batas karakter maksimum, jadi sebaiknya Anda tidak menggunakannya untuk menghubungkan.

Pertimbangan untuk grup keamanan

Agar Anda dapat terhubung ke instans DB, instans DB tersebut harus ditautkan dengan grup keamanan yang berisi alamat IP dan konfigurasi jaringan yang diperlukan. Instans DB Anda mungkin menggunakan grup keamanan default. Jika Anda menetapkan grup keamanan yang tidak dikonfigurasi secara default saat membuat instans DB, firewall instans DB akan mencegah koneksi. Untuk mengetahui informasi tentang cara membuat grup keamanan baru, lihat Mengontrol akses dengan grup keamanan.

Setelah Anda membuat grup keamanan baru, ubah instans DB Anda untuk dikaitkan dengan grup keamanan. Untuk mengetahui informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Anda dapat meningkatkan keamanan menggunakan SSL untuk mengenkripsi koneksi ke instans DB Anda. Untuk mengetahui informasi selengkapnya, lihat Lapisan Soket Aman Oracle.

Pertimbangan untuk arsitektur proses

Proses server menangani koneksi pengguna ke instans Oracle DB. Secara default, instans Oracle DB menggunakan proses server khusus. Dengan proses server khusus, setiap proses server hanya melayani satu proses pengguna. Anda dapat memilih untuk mengonfigurasi proses server bersama. Dengan proses server bersama, setiap proses server dapat melayani beberapa proses pengguna.

Anda dapat mempertimbangkan untuk menggunakan proses server bersama saat sejumlah besar sesi pengguna menggunakan terlalu banyak memori pada server. Anda juga dapat mempertimbangkan proses server bersama ketika sesi sangat sering terhubung dan terputus, sehingga mengakibatkan masalah performa. Penggunakan proses server bersama juga memiliki kekurangan. Sebagai contoh, sumber daya CPU dapat disaring, serta lebih rumit untuk dikonfigurasi dan dikelola.

Untuk mengetahui informasi selengkapnya tentang proses server khusus dan bersama, lihat Tentang proses server khusus dan bersama dalam dokumentasi Oracle. Untuk mengetahui informasi selengkapnya tentang cara mengonfigurasi proses server bersama di instans RDS for Oracle DB, lihat Bagaimana cara mengonfigurasi basis data Amazon RDS for Oracle untuk digunakan bersama dengan server bersama? di Pusat Pengetahuan.

Memecahkan masalah koneksi ke instans Oracle DB Anda

Berikut adalah masalah yang mungkin Anda hadapi saat mencoba menghubungkan ke instans Oracle DB.

Masalah	Saran pemecahan masalah
Tidak dapat terhubung ke instans DB Anda.	Untuk instans DB yang baru dibuat, instans DB tersebut memiliki status membuat hingga siap digunakan. Saat statusnya berubah menjadi tersedia, Anda dapat terhubung ke instans DB. Tergantung pada kelas instans DB dan jumlah penyimpanan, diperlukan waktu hingga 20 menit sebelum instans DB baru tersedia.
Tidak dapat terhubung ke instans DB Anda.	Jika Anda tidak dapat mengirim atau menerima komunikasi melalui port yang Anda tentukan saat membuat instans DB, Anda tidak dapat terhubung ke instans DB. Periksa dengan administrator jaringan Anda untuk memverifikasi bahwa port yang Anda tentukan untuk instans DB memungkinkan komunikasi masuk dan keluar.

Masalah	Saran pemecahan masalah
Tidak dapat terhubung ke instans DB Anda.	Aturan akses yang diterapkan oleh firewall lokal dan alamat IP yang Anda izinkan untuk mengakses instans DB dalam grup keamanan untuk instans DB mungkin tidak cocok. Masalahnya kemungkinan besar adalah aturan masuk atau keluar pada firewall Anda. Anda dapat menambahkan atau mengedit aturan masuk di grup keamanan. Untuk Sumber, pilih IP Saya. Pilihan ini akan mengizinkan akses ke instans DB dari alamat IP yang terdeteksi di browser Anda. Untuk mengetahui informasi selengkapnya, lihat Amazon VPC dan Amazon RDS. Untuk mengetahui informasi selengkapnya tentang grup keamanan, lihat Mengontrol akses dengan grup keamanan. Untuk mengetahui lebih lanjut tentang proses penyiapan aturan untuk grup keamanan Anda, lihat Tutorial: Membuat VPC untuk digunakan dengan instans DB (khusus IPv4).
Gagal menghubungkan karena host atau objek target tidak ada – Oracle, Kesalahan: QRA-12545	Pastikan bahwa Anda menentukan nama server dan nomor port dengan benar. Untuk Nama server, masukkan nama DNS dari konsol. Untuk mengetahui informasi tentang cara menemukan nama DNS dan nomor port untuk instans DB, lihat Menemukan titik akhir instans DB RDS for Oracle.
Nama pengguna/kata sandi tidak valid; logon ditolak – Oracle, Kesalahan : ORA-01017	Anda dapat menghubungi instans DB, tetapi koneksinya ditolak. Masalah ini biasanya disebabkan oleh pemberian nama pengguna atau kata sandi yang tidak benar. Verifikasi nama pengguna dan kata sandi, lalu coba lagi.

Masalah	Saran pemecahan masalah
TNS:listener saat ini tidak mengetahui SID yang diberikan dalam deskriptor koneksi - Oracle, KESALAHAN: ORA-12505	Pastikan SID yang benar sudah dimasukkan. SID sama dengan nama DB Anda. Temukan nama DB di tab Konfigurasi pada halaman Basis data untuk instans Anda. Anda juga dapat menemukan nama DB menggunakan AWS CLI: `aws rds describe-db-instances --query 'DBInstances[*].[DBInstanceIdentifier,DBName]' --output text`

Untuk mengetahui informasi selengkapnya tentang masalah koneksi, lihat Tidak dapat terhubung ke instans DB Amazon RDS.

Memodifikasi properti koneksi menggunakan parameter sqlnet.ora

File sqlnet.ora mencakup parameter yang mengonfigurasi fitur Oracle Net pada server dan klien basis data Oracle. Dengan menggunakan parameter dalam file sqlnet.ora, Anda dapat memodifikasi properti untuk koneksi masuk dan keluar basis data.

Untuk informasi selengkapnya tentang mengapa Anda mungkin menetapkan parameter sqlnet.ora, lihat Mengonfigurasi parameter profil dalam dokumentasi Oracle.

Menetapkan parameter sqlnet.ora

Grup parameter Amazon RDS for Oracle mencakup subset parameter sqlnet.ora. Anda menetapkannya dengan cara yang sama seperti Anda mengatur parameter Oracle lainnya. Awalan `sqlnetora.` mengidentifikasi parameter mana yang merupakan parameter sqlnet.ora. Misalnya, dalam grup parameter Oracle di Amazon RDS, parameter sqlnet.ora `default_sdu_size` adalah `sqlnetora.default_sdu_size`.

Untuk informasi tentang cara mengelola grup parameter dan mengatur nilai parameter, lihat Bekerja dengan grup parameter.

Parameter sqlnet.ora yang didukung

Amazon RDS mendukung parameter sqlnet.ora berikut. Perubahan pada parameter sqlnet dinamis langsung berpengaruh.

Parameter	Nilai valid	Statis/ Dinamis	Deskripsi
sqlnetora.default_sdu_size	Oracle 12c – 512 ke 20971	Dinamis	Ukuran unit data sesi (SDU), dalam byte. SDU adalah jumlah data yang dimasukkan ke dalam penyangga dan dikirim ke seluruh jaringan pada satu waktu.
sqlnetora.diag_adr_enabled	ON, OFF	Dinamis	Nilai yang mengaktifkan atau menonaktifkan pelacakan Automatic Diagnostic Repository (ADR). ON menentukan bahwa pelacakan file ADR digunakan. OFF menentukan bahwa pelacakan file non-ADR digunakan.
sqlnetora.recv_buf_size	8192 ke 26843	Dinamis	Batas ruang buffer untuk menerima operasi sesi, didukung oleh protokol TCP/IP, TCP/IP dengan SSL, dan SDP.
sqlnetora.send_buf_size	8192 ke 26843	Dinamis	Batas ruang buffer untuk mengirim operasi sesi, didukung oleh protokol TCP/IP, TCP/IP dengan SSL, dan SDP.

Parameter	Nilai valid	Statis/ Dinamis	Deskripsi
sqlnetora.sqlnet.allowed_logon_version_client	8, 10, 11, 12	Dinamis	Versi protokol autentikasi minimum yang diizinkan untuk klien, dan server yang bertindak sebagai klien, untuk membangun koneksi ke instans Oracle DB.
sqlnetora.sqlnet.allowed_logon_version_server	8, 9, 10, 11, 12, 12a	Dinamis	Versi protokol autentikasi minimum yang diizinkan untuk membuat koneksi ke instans Oracle DB.
sqlnetora.sqlnet.expire_time	0 ke 1440	Dinamis	Interval waktu, dalam menit, untuk mengirim cek untuk memverifikasi bahwa koneksi server-klien aktif.
sqlnetora.sqlnet.inbound_connect_timeout	0 atau 10 ke 7200	Dinamis	Waktu, dalam detik, untuk klien untuk terhubung dengan server basis data dan memberikan informasi autentikasi yang diperlukan.
sqlnetora.sqlnet.outbound_connect_timeout	0 atau 10 ke 7200	Dinamis	Waktu, dalam hitungan detik, bagi klien untuk membuat koneksi Oracle Net ke instans DB.
sqlnetora.sqlnet.recv_timeout	0 atau 10 ke 7200	Dinamis	Waktu, dalam detik, untuk server basis data menunggu data klien setelah membuat koneksi.

Parameter	Nilai valid	Statis/ Dinamis	Deskripsi
`sqlnetora.sqlnet.send_timeout`	`0` atau `10` ke `7200`	Dinamis	Waktu, dalam hitungan detik, untuk server basis data untuk menyelesaikan operasi pengiriman ke klien setelah membuat koneksi.
`sqlnetora.tcp.connect_timeout`	`0` atau `10` ke `7200`	Dinamis	Waktu, dalam hitungan detik, agar klien dapat membuat koneksi TCP ke server basis data.
`sqlnetora.trace_level_server`	`0`, `4`, `10`, `16`, `OFF`, `USER`, `ADMIN`, `SUPPOF`	Dinamis	Untuk pelacakan non-ADR, aktifkan pelacakan server pada tingkat tertentu atau nonaktifkan pelacakannya.

Nilai default untuk setiap parameter sqlnet.ora yang didukung adalah default Oracle untuk rilis tersebut. Untuk informasi tentang nilai default Basis Data Oracle 12c, lihat Parameter untuk file sqlnet.ora dalam dokumentasi Basis Data Oracle 12c.

Melihat parameter sqlnet.ora

Anda dapat melihat parameter sqlnet.ora dan pengaturannya menggunakan AWS Management Console, AWS CLI, atau klien SQL.

Melihat parameter sqlnet.ora menggunakan konsol

Untuk informasi tentang cara melihat parameter dalam grup parameter, lihat Bekerja dengan grup parameter.

Dalam grup parameter Oracle, awalan `sqlnetora.` mengidentifikasi parameter mana yang merupakan parameter sqlnet.ora.

Melihat parameter sqlnet.ora menggunakan AWS CLI

Untuk melihat parameter sqlnet.ora yang dikonfigurasi dalam grup parameter Oracle, gunakan perintah. AWS CLI describe-db-parameters

Untuk melihat semua parameter sqlnet.ora untuk instance Oracle DB, panggil perintah -portion. AWS CLI download-db-log-file Tentukan pengidentifikasi instans DB, nama file log, dan jenis output.

Example

Kode berikut mencantumkan semua parameter sqlnet.ora untuk `mydbinstance`.

Untuk Linux, macOS, atau Unix:

```
aws rds download-db-log-file-portion \
    --db-instance-identifier mydbinstance \
    --log-file-name trace/sqlnet-parameters \
    --output text
```

Untuk Windows:

```
aws rds download-db-log-file-portion ^
    --db-instance-identifier mydbinstance ^
    --log-file-name trace/sqlnet-parameters ^
    --output text
```

Melihat parameter sqlnet.ora menggunakan klien SQL

Setelah Anda terhubung ke instans Oracle DB dalam klien SQL, kueri berikut mencantumkan parameter sqlnet.ora.

```
SELECT * FROM TABLE
   (rdsadmin.rds_file_util.read_text_file(
        p_directory => 'BDUMP',
        p_filename  => 'sqlnet-parameters'));
```

Untuk informasi tentang cara menghubungkan ke instans DB Oracle di klien SQL, lihat Menghubungkan ke instans RDS for Oracle DB.

Mengamankan koneksi instans DB Oracle

Amazon RDS for Oracle mendukung koneksi yang dienkripsi SSL/TLS dan juga opsi Oracle Native Network Encryption (NNE) untuk mengenkripsi koneksi antara aplikasi Anda dan instans DB Oracle Anda. Untuk informasi selengkapnya tentang opsi Oracle Native Network Encryption, lihat Enkripsi jaringan asli Oracle.

Topik

- Menggunakan SSL dengan instans DB RDS for Oracle
- Memperbarui aplikasi untuk terhubung ke instans DB Oracle menggunakan sertifikat SSL/TLS baru
- Menggunakan enkripsi jaringan native dengan instans DB RDS for Oracle
- Mengonfigurasi autentikasi Kerberos untuk Amazon RDS for Oracle
- Mengonfigurasi akses UTL_HTTP menggunakan sertifikat dan dompet Oracle

Menggunakan SSL dengan instans DB RDS for Oracle

Lapisan Soket Aman (Secure Sockets Layer, SSL) adalah protokol standar industri untuk mengamankan koneksi jaringan antara klien dan server. Setelah SSL versi 3.0, namanya diubah menjadi Keamanan Lapisan Pengangkutan (TLS), tetapi kami masih sering menyebut protokol ini sebagai SSL. Amazon RDS mendukung enkripsi SSL untuk instans DB Oracle. Dengan SSL, Anda dapat mengenkripsi koneksi antara klien aplikasi dan instans DB Oracle. Dukungan SSL tersedia di semua Wilayah AWS untuk Oracle.

Untuk mengaktifkan enkripsi SSL untuk instans DB Oracle, tambahkan opsi Oracle SSL ke kelompok opsi yang terkait dengan instans DB. Amazon RDS menggunakan port kedua, sebagaimana diperlukan oleh Oracle, untuk koneksi SSL. Melakukan hal ini memungkinkan komunikasi baik teks jelas maupun berenkripsi SSL terjadi serentak antara instans basis data dan klien Oracle. Misalnya, Anda dapat menggunakan port dengan komunikasi teks jelas untuk berkomunikasi dengan sumber daya lain di dalam VPC sambil menggunakan porta komunikasi berenkripsi SSL untuk berkomunikasi dengan sumber daya di luar VPC.

Untuk informasi selengkapnya, lihat Lapisan Soket Aman Oracle.

> **Note**
>
> Anda tidak dapat menggunakan SSL dan Native Network Enrypction (NNE) Oracle pada instans DB yang sama. Sebelum Anda dapat menggunakan enkripsi SSL, Anda harus menonaktifkan enkripsi koneksi lainnya.

Memperbarui aplikasi untuk terhubung ke instans DB Oracle menggunakan sertifikat SSL/TLS baru

Sejak 13 Januari 2023, Amazon RDS telah menerbitkan serifikat Otoritas Sertifikat (CA) baru untuk terhubung ke instans DB RDS menggunakan Lapisan Soket Aman atau Keamanan Lapisan Pengangkutan (SSL/TLS). Setelah itu, Anda dapat menemukan informasi tentang pembaruan aplikasi untuk menggunakan sertifikat baru.

Topik ini dapat membantu Anda menentukan apakah aplikasi klien menggunakan SSL/TLS untuk terhubung ke instans DB Anda.

> **Important**
>
> Ketika Anda mengubah sertifikat untuk instans DB Amazon RDS for Oracle, hanya pendengar basis data yang dinyalakan ulang. Instans basis data tidak dimulai ulang. Koneksi basis data yang ada tidak terpengaruh, tetapi koneksi baru akan mengalami kesalahan selama periode waktu yang singkat sementara pendengar dimulai ulang.

> **Note**
>
> Untuk aplikasi klien yang menggunakan SSL/TLS untuk terhubung ke instans DB, Anda harus memperbarui penyimpanan kepercayaan aplikasi klien untuk menyertakan sertifikat CA baru.

Setelah Anda memperbarui sertifikat CA di penyimpanan kepercayaan aplikasi klien, Anda dapat merotasi sertifikat di instans DB Anda. Sebaiknya Anda menguji prosedur ini di lingkungan pengembangan dan penahapan sebelum menerapkannya di lingkungan produksi Anda.

Untuk informasi selengkapnya tentang rotasi sertifikat, lihat Merotasi sertifikat SSL/TLS. Untuk informasi selengkapnya tentang cara mengunduh sertifikat, lihat . Untuk informasi tentang menggunakan SSL/TLS dengan instans DB Oracle, lihat Lapisan Soket Aman Oracle.

Topik

- Mencari tahu apakah aplikasi terhubung menggunakan SSL
- Memperbarui penyimpanan kepercayaan aplikasi Anda
- Contoh kode Java untuk membangun koneksi SSL

Mencari tahu apakah aplikasi terhubung menggunakan SSL

Jika instans DB Oracle Anda menggunakan opsi SSL yang ditambahkan, Anda mungkin menggunakan SSL. Periksa ini dengan mengikuti petunjuk di Menampilkan daftar opsi dan pengaturan opsi untuk grup opsi. Untuk informasi tentang opsi SSL, lihat Lapisan Soket Aman Oracle.

Periksa log pendengar untuk menentukan apakah ada koneksi SSL. Berikut ini adalah output contoh di log pendengar.

```
date time * (CONNECT_DATA=(CID=(PROGRAM=program)
(HOST=host)(USER=user))(SID=sid)) *
(ADDRESS=(PROTOCOL=tcps)(HOST=host)(PORT=port)) * establish * ORCL * 0
```

Saat PROTOCOL memiliki nilai `tcps` untuk entri, itu menunjukkan koneksi SSL. Namun, saat HOST adalah `127.0.0.1`, Anda dapat mengabaikan entri tersebut. Koneksi dari `127.0.0.1` adalah agen manajemen lokal di instans DB. Koneksi ini bukan koneksi SSL eksternal. Oleh karena itu, Anda memiliki aplikasi yang terhubung menggunakan SSL jika Anda melihat entri log pendengar ketika PROTOCOL adalah `tcps` dan HOST adalah bukan `127.0.0.1`.

Untuk memeriksa log pendengar, Anda dapat menerbitkan log ke Log Amazon CloudWatch. Untuk informasi selengkapnya, lihat Menerbitkan log Oracle ke Amazon CloudWatch Logs.

Memperbarui penyimpanan kepercayaan aplikasi Anda

Anda dapat memperbarui penyimpanan kepercayaan untuk aplikasi yang menggunakan SQL*Plus atau JDBC untuk koneksi SSL/TLS.

Memperbarui penyimpanan kepercayaan aplikasi Anda untuk SQL*Plus

Anda dapat memperbarui penyimpanan kepercayaan untuk aplikasi yang menggunakan SQL*Plus untuk koneksi SSL/TLS.

> **Note**
>
> Saat memperbarui penyimpanan kepercayaan, Anda dapat mempertahankan sertifikat lama selain menambahkan sertifikat baru.

Untuk memperbarui penyimpanan kepercayaan untuk aplikasi SQL*Plus

1. Unduh sertifikat root baru yang berfungsi untuk semua Wilayah AWS dan letakkan file di direktori `ssl_wallet`.

   Untuk informasi tentang cara mengunduh sertifikat root, lihat .
2. Jalankan perintah berikut untuk memperbarui dompet Oracle.

   ```
   prompt>orapki wallet add -wallet $ORACLE_HOME/ssl_wallet -trusted_cert -cert
         $ORACLE_HOME/ssl_wallet/ssl-cert.pem -auto_login_only
   ```

   Ganti nama file dengan nama yang Anda unduh.
3. Jalankan perintah berikut untuk mengonfirmasi bahwa dompet berhasil diperbarui.

   ```
   prompt>orapki wallet display -wallet $ORACLE_HOME/ssl_wallet
   ```

   Output Anda harus berisi berikut ini.

   ```
   Trusted Certificates:
   Subject: CN=Amazon RDS Root 2019 CA,OU=Amazon RDS,O=Amazon Web Services\,
    Inc.,L=Seattle,ST=Washington,C=US
   ```

Memperbarui penyimpanan kepercayaan aplikasi Anda untuk JDBC

Anda dapat memperbarui penyimpanan kepercayaan untuk aplikasi yang menggunakan JDBC untuk koneksi SSL/TLS.

Untuk informasi tentang cara mengunduh sertifikat root, lihat .

Untuk contoh skrip yang mengimpor sertifikat, lihat Contoh skrip untuk mengimpor sertifikat ke trust store Anda.

Contoh kode Java untuk membangun koneksi SSL

Contoh kode berikut menunjukkan cara menyiapkan koneksi SSL menggunakan JDBC.

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Properties;

public class OracleSslConnectionTest {
    private static final String DB_SERVER_NAME = "<dns-name-provided-by-amazon-rds>";
    private static final Integer SSL_PORT = "<ssl-option-port-configured-in-option-
group>";
    private static final String DB_SID = "<oracle-sid>";
    private static final String DB_USER = "<user name>";
    private static final String DB_PASSWORD = "<password>";
    // This key store has only the prod root ca.
    private static final String KEY_STORE_FILE_PATH = "<file-path-to-keystore>";
    private static final String KEY_STORE_PASS = "<keystore-password>";

    public static void main(String[] args) throws SQLException {
        final Properties properties = new Properties();
        final String connectionString = String.format(
                "jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCPS)(HOST=%s)(PORT=
%d))(CONNECT_DATA=(SID=%s)))",
                DB_SERVER_NAME, SSL_PORT, DB_SID);
        properties.put("user", DB_USER);
        properties.put("password", DB_PASSWORD);
        properties.put("oracle.jdbc.J2EE13Compliant", "true");
        properties.put("javax.net.ssl.trustStore", KEY_STORE_FILE_PATH);
        properties.put("javax.net.ssl.trustStoreType", "JKS");
        properties.put("javax.net.ssl.trustStorePassword", KEY_STORE_PASS);
        final Connection connection = DriverManager.getConnection(connectionString,
 properties);
        // If no exception, that means handshake has passed, and an SSL connection can
 be opened
    }
}
```

> **Important**
>
> Setelah Anda menentukan bahwa koneksi basis data Anda menggunakan SSL/TLS dan telah memperbarui penyimpanan kepercayaan aplikasi, Anda dapat memperbarui basis data untuk menggunakan sertifikat rds-ca-rsa2048-g1. Untuk mengetahui petunjuknya, lihat langkah 3 dalam Memperbarui sertifikat CA Anda dengan memodifikasi instans atau cluster DB.

Menggunakan enkripsi jaringan native dengan instans DB RDS for Oracle

Oracle Database menawarkan dua cara untuk mengenkripsi data melalui jaringan: enkripsi jaringan native (NNE) dan Keamanan Lapisan Pengangkutan (TLS). NNE adalah fitur keamanan eksklusif Oracle, sedangkan TLS adalah standar industri. RDS for Oracle mendukung NNE untuk semua edisi Oracle Database.

NNE memiliki keuntungan sebagai berikut dibandingkan TLS:

- Anda dapat mengontrol NNE pada klien dan server menggunakan pengaturan di opsi NNE:
  - `SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS` dan `SQLNET.ALLOW_WEAK_CRYPTO`
  - `SQLNET.CRYPTO_CHECKSUM_CLIENT` dan `SQLNET.CRYPTO_CHECKSUM_SERVER`
  - `SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT` dan `SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER`
  - `SQLNET.ENCRYPTION_CLIENT` dan `SQLNET.ENCRYPTION_SERVER`
  - `SQLNET.ENCRYPTION_TYPES_CLIENT` dan `SQLNET.ENCRYPTION_TYPES_SERVER`
- Dalam kebanyakan kasus, Anda tidak perlu mengonfigurasi klien atau server Anda. Sebaliknya, TLS mengharuskan Anda untuk mengonfigurasi klien dan server.
- Tidak memerlukan sertifikat apa pun. Di TLS, server memerlukan sertifikat (yang nantinya kedaluwarsa), dan klien memerlukan sertifikat root tepercaya untuk otoritas sertifikat yang mengeluarkan sertifikat server tersebut.

Untuk mengaktifkan enkripsi NNE untuk instans DB Oracle, tambahkan opsi Oracle NNE ke kelompok opsi yang terkait dengan instans DB. Untuk informasi selengkapnya, lihat Enkripsi jaringan asli Oracle.

> **Note**
>
> Anda tidak dapat menggunakan NNE dan TLS pada instans DB yang sama.

Mengonfigurasi autentikasi Kerberos untuk Amazon RDS for Oracle

Sekarang Anda dapat menggunakan autentikasi Kerberos untuk mengautentikasi pengguna saat mereka terhubung ke instans DB Amazon RDS for Oracle Anda. Dalam konfigurasi ini, instans DB Anda bekerja dengan AWS Directory Service for Microsoft Active Directory, juga disebut AWS Managed Microsoft AD. Saat pengguna mengautentikasi dengan instans DB RDS for Oracle digabungkan ke domain terpercaya, permintaan autentikasi diteruskan ke direktori yang Anda buat dengan AWS Directory Service.

Menyimpan semua kredensial di direktori yang sama dapat menghemat waktu serta tenaga Anda. Anda memiliki tempat terpusat untuk menyimpan dan mengelola kredensial untuk beberapa instans basis data. Direktori juga dapat meningkatkan profil keamanan keseluruhan Anda.

Ketersediaan wilayah dan versi

Ketersediaan dan dukungan fitur bervariasi di berbagai versi khusus dari setiap mesin basis data, dan di seluruh Wilayah AWS. Untuk informasi lebih lanjut tentang versi dan ketersediaan Wilayah RDS for Oracle dengan autentikasi Kerberos, lihat Autentikasi Kerberos.

> **Note**
>
> Autentikasi Kerberos tidak didukung untuk kelas instans DB yang sudah tidak digunakan lagi untuk instans DB Oracle. Untuk informasi selengkapnya, lihat Kelas instans RDS for Oracle.

Topik

- Menyiapkan autentikasi Kerberos untuk instans DB Oracle
- Mengelola instans DB dalam domain
- Menghubungkan ke Oracle dengan autentikasi Kerberos

Menyiapkan autentikasi Kerberos untuk instans DB Oracle

Gunakan AWS Directory Service for Microsoft Active Directory, juga disebut AWS Managed Microsoft AD, untuk menyiapkan autentikasi Kerberos untuk instans DB Oracle. Untuk menyiapkan autentikasi Kerberos, selesaikan langkah berikut:

- Langkah 1: Buat direktori menggunakan AWS Managed Microsoft AD
- Langkah 2: Buat kepercayaan
- Langkah 3: Konfigurasi izin IAM untuk Amazon RDS
- Langkah 4: Buat dan konfigurasikan pengguna
- Langkah 5: Aktifkan lalu lintas antar-VPC antara direktori dan instans DB
- Langkah 6: Buat atau modifikasi instans Oracle DB
- Langkah 7: Buat login Oracle untuk autentikasi Kerberos
- Langkah 8: Konfigurasikan klien Oracle

> **Note**
>
> Selama penyiapan, RDS membuat pengguna basis data Oracle bernama *managed_service_user*@*example.com* dengan hak istimewa `CREATE SESSION`, dan *example.com* sebagai nama domain Anda. Pengguna ini sesuai dengan pengguna yang dibuat Directory Service di dalam Managed Active Directory Anda. Secara berkala, RDS menggunakan kredensial yang disediakan oleh Directory Service untuk masuk ke basis data Oracle Anda. Setelah itu, RDS langsung menghancurkan cache tiket.

Langkah 1: Buat direktori menggunakan AWS Managed Microsoft AD

AWS Directory Service membuat Active Directory yang dikelola penuh di AWS Cloud. Saat Anda membuat direktori AWS Managed Microsoft AD, AWS Directory Service membuat dua server pengendali domain dan Sistem Nama Domain (DNS) atas nama Anda. Server direktori dibuat di subnet berbeda di VPC. Redundansi ini membantu memastikan bahwa direktori Anda tetap dapat diakses meski terjadi kegagalan.

Saat Anda membuat direktori AWS Managed Microsoft AD, AWS Directory Service melakukan tugas berikut ini atas nama Anda:

- Menyiapkan Active Directory di dalam VPC.

- Membuat akun administrator direktori dengan nama pengguna Admin dan kata sandi yang ditentukan. Anda menggunakan akun ini untuk mengelola direktori Anda.

> **Note**
>
> Pastikan untuk menyimpan kata sandi ini. AWS Directory Service tidak menyimpannya. Anda dapat mengaturnya ulang, tetapi tidak dapat mengambilnya.

- Membuat grup keamanan untuk pengendali direktori.

Saat Anda meluncurkan sebuah AWS Managed Microsoft AD, AWS membuat Unit Organisasi (OU) yang berisi semua objek direktori Anda. OU ini memiliki nama NetBIOS yang Anda masukkan saat membuat direktori, dan terletak di root domain. Root domain dimiliki dan dikelola oleh AWS.

Akun Admin yang dibuat dengan direktori AWS Managed Microsoft AD Anda memiliki izin untuk melakukan berbagai tugas administratif paling umum bagi OU Anda:

- Membuat, memperbarui, atau menghapus pengguna
- Menambahkan sumber daya ke domain Anda seperti server file atau cetak, lalu menetapkan izin untuk sumber daya tersebut kepada pengguna di OU Anda
- Membuat OU dan kontainer tambahan
- Mendelegasikan kewenangan
- Memulihkan objek yang dihapus dari Keranjang Sampah Active Directory
- Jalankan PowerShell modul AD dan DNS Windows pada Layanan Web Direktori Aktif

Akun Admin juga berhak untuk melakukan aktivitas di seluruh domain berikut:

- Mengelola konfigurasi DNS (menambahkan, menghapus, atau memperbarui catatan, zona, dan penerus)
- Melihat log peristiwa DNS
- Melihat log peristiwa keamanan

Untuk membuat direktori, gunakan AWS Management Console, AWS CLI, atau API AWS Directory Service. Pastikan untuk membuka port keluar yang relevan pada grup keamanan direktori sehingga direktori dapat berkomunikasi dengan instans DB Oracle.

Untuk membuat direktori dengan AWS Managed Microsoft AD

1. Masuk ke AWS Management Console dan buka konsol AWS Directory Service di https://console.aws.amazon.com/directoryservicev2/.
2. Di panel navigasi, pilih Direktori, lalu pilih Siapkan Direktori.
3. Pilih AWS Managed Microsoft AD. AWS Managed Microsoft AD adalah satu-satunya opsi yang saat ini dapat Anda gunakan dengan Amazon RDS.
4. Masukkan informasi berikut:

   Nama DNS Direktori

   Nama yang sepenuhnya memenuhi syarat direktori, seperti **`corp.example.com`**.

   Nama NetBIOS direktori

   Nama singkat direktori, seperti **`CORP`**.

   Deskripsi direktori

   (Opsional) Deskripsi direktori.

   Kata sandi admin

   Kata sandi administrator direktori. Proses pembuatan direktori menciptakan akun administrator dengan nama pengguna Admin dan kata sandi ini.

   Kata sandi administrator direktori dan tidak boleh menyertakan kata "admin". Kata sandi peka terhadap huruf besar/kecil dan harus terdiri dari 8-64 karakter. Kata sandi juga harus berisi setidaknya satu karakter dari tiga di antara empat kategori berikut:

   - Huruf kecil (a-z)
   - Huruf besar (A-Z)
   - Angka (0–9)
   - Karakter non-alfanumerik (~!@#$%^&*_-+=`|\(){}[]:;"'<>,.?/)

   Konfirmasi kata sandi

   Kata sandi administrator diketik ulang.
5. Pilih Berikutnya.
6. Masukkan informasi berikut di bagian Jaringan, lalu pilih Berikutnya:

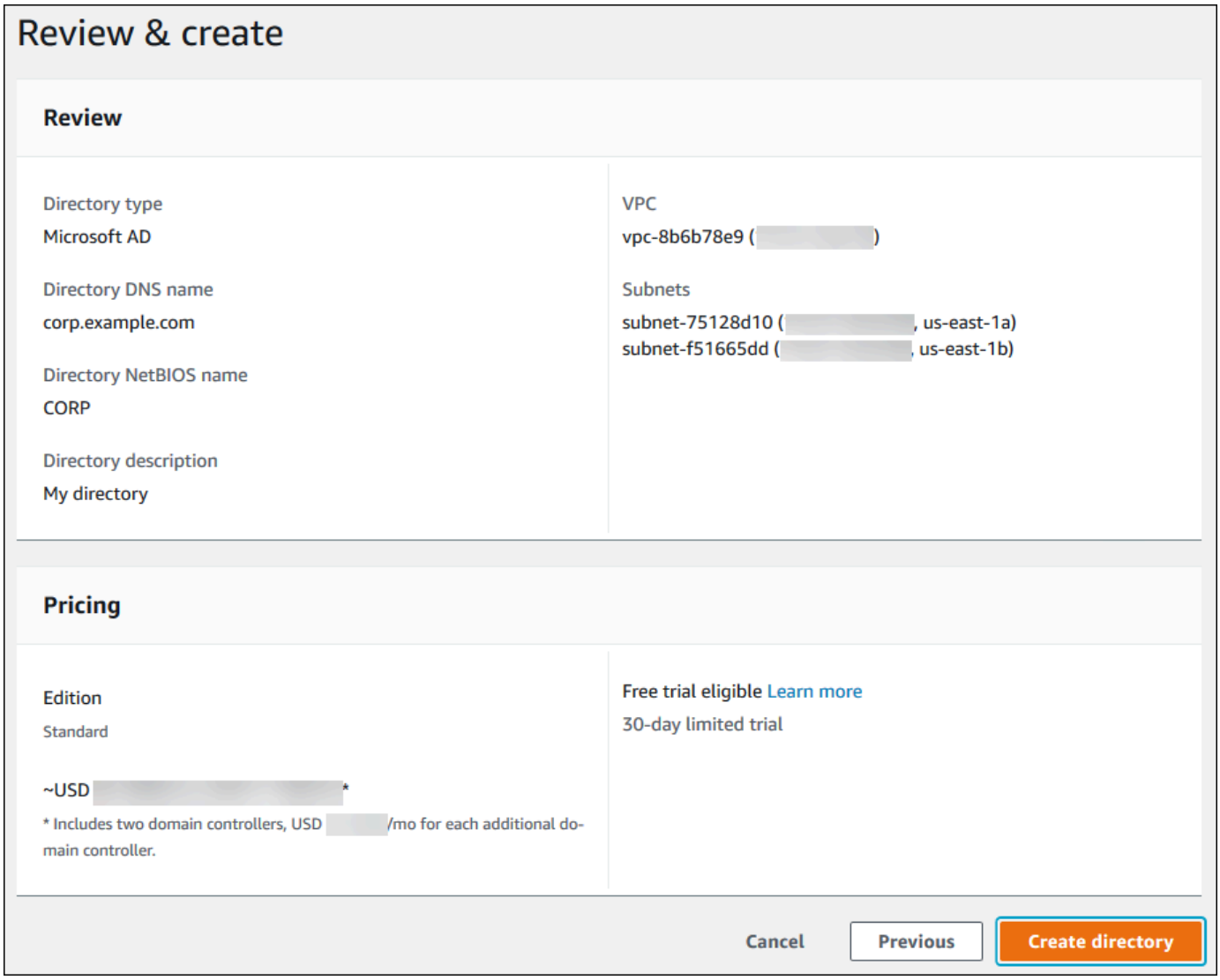
VPC

VPC untuk direktori. Buat instans DB Oracle dalam VPC yang sama ini.

Subnet

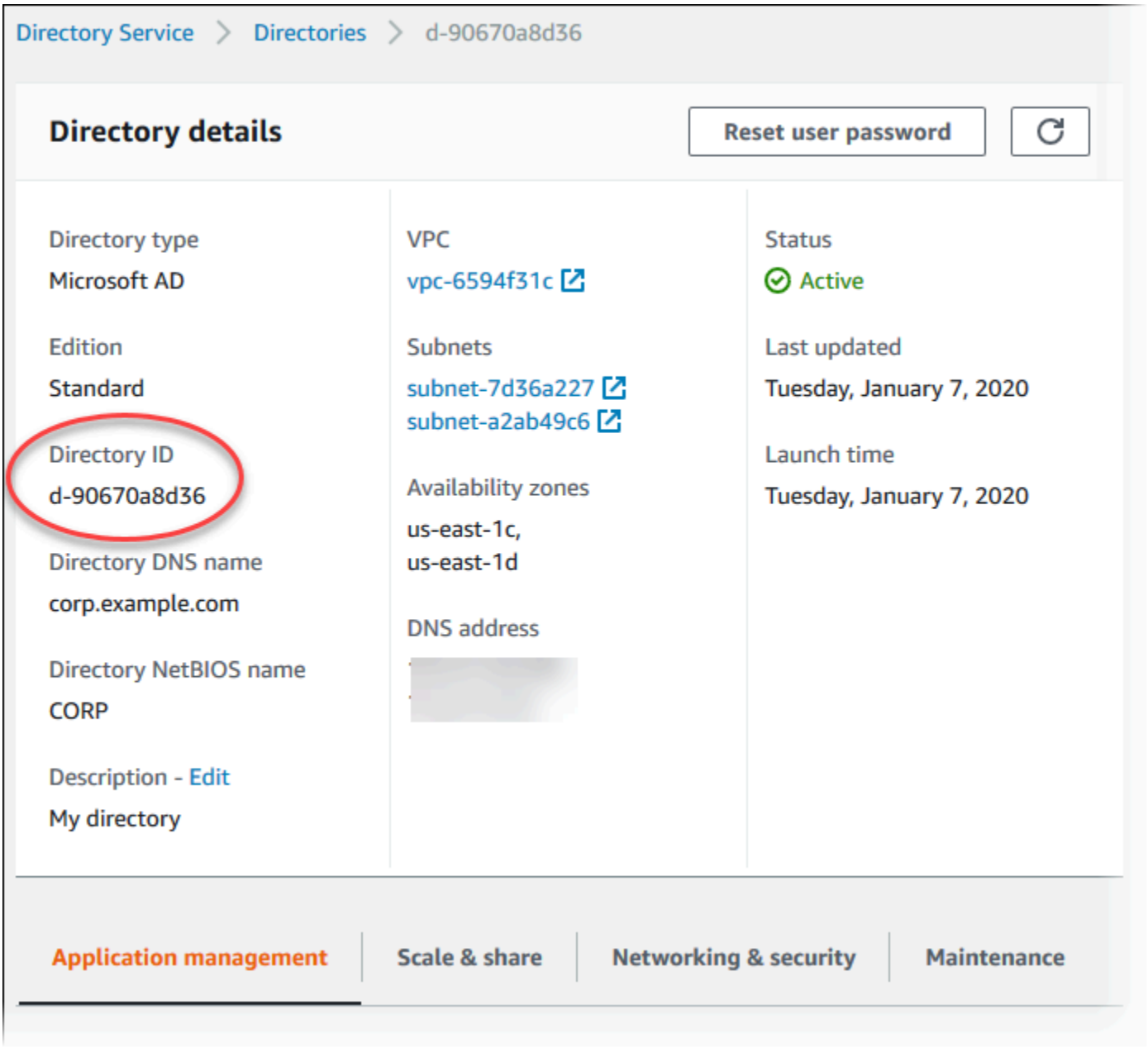
Subnet untuk server direktori. Kedua subnet harus berada di Zona Ketersediaan yang berbeda.

7. Tinjau informasi direktori dan buat perubahan yang diperlukan. Jika informasi sudah benar, pilih Buat direktori.



Pembuatan direktori memerlukan waktu beberapa menit. Setelah direktori berhasil dibuat, nilai Status berubah menjadi Aktif.

Untuk melihat informasi tentang direktori Anda, pilih nama direktori di daftar direktori. Catat nilai ID Direktori karena Anda memerlukan nilai ini saat membuat atau mengubah instans DB Oracle.



Langkah 2: Buat kepercayaan

Jika Anda berencana untuk menggunakan AWS Managed Microsoft AD saja, lanjutkan ke Langkah 3: Konfigurasi izin IAM untuk Amazon RDS.

Untuk mendapatkan autentikasi Kerberos menggunakan Microsoft Active Directory on-premise atau yang di-host sendiri, buat kepercayaan forest atau kepercayaan eksternal. Kepercayaan bisa satu atau dua arah. Untuk informasi selengkapnya tentang menyiapkan kepercayaan forest menggunakan

AWS Directory Service, lihat Waktu yang tepat untuk membuat hubungan kepercayaan dalam Panduan Administrasi AWS Directory Service.

Langkah 3: Konfigurasi izin IAM untuk Amazon RDS

Untuk memanggil AWS Directory Service untuk Anda, Amazon RDS memerlukan peran IAM yang menggunakan kebijakan IAM terkelola `AmazonRDSDirectoryServiceAccess`. Peran ini memungkinkan Amazon RDS melakukan panggilan ke AWS Directory Service.

> **Note**
>
> Agar peran ini mengizinkan akses, titik akhir AWS Security Token Service (AWS STS) harus diaktifkan dalam Wilayah AWS yang benar untuk Akun AWS Anda. Titik akhir AWS STS akan aktif secara default di semua Wilayah AWS, dan Anda dapat menggunakannya tanpa tindakan lebih lanjut. Untuk informasi selengkapnya, lihat Mengaktifkan dan menonaktifkan AWS STS di Wilayah AWS dalam Panduan Pengguna IAM.

Membuat peran IAM

Saat membuat instans DB menggunakan AWS Management Console, dan pengguna konsol memiliki izin `iam:CreateRole`, konsol akan secara otomatis membuat `rds-directoryservice-kerberos-access-role`. Jika tidak, Anda harus membuat peran IAM secara manual.
Saat membuat peran IAM secara manual, pilih `Directory Service`, dan lampirkan `AmazonRDSDirectoryServiceAccess` kebijakan terkelola AWS ke layanan ini.

Untuk informasi selengkapnya tentang pembuatan peran IAM untuk sebuah layanan, lihat Membuat peran untuk mendelegasikan izin ke layanan AWS dalam Panduan Pengguna IAM.

> **Note**
>
> Peran IAM yang digunakan untuk Autentikasi Windows untuk RDS for Microsoft SQL Server tidak dapat digunakan untuk RDS for Oracle.

Membuat kebijakan kepercayaan IAM secara manual

Secara opsional, Anda dapat membuat kebijakan sumber daya dengan izin yang diperlukan alih-alih menggunakan kebijakan IAM terkelola `AmazonRDSDirectoryServiceAccess`. Pilih `directoryservice.rds.amazonaws.com` dan `rds.amazonaws.com` sebagai pengguna utama.

Untuk membatasi izin yang diberikan Amazon RDS kepada layanan lain untuk sumber daya tertentu, sebaiknya gunakan kunci konteks kondisi global `aws:SourceArn` dan `aws:SourceAccount` dalam kebijakan sumber daya. Cara paling efektif untuk melindungi dari masalah confused deputy adalah menggunakan kunci konteks kondisi global `aws:SourceArn` dengan ARN lengkap sumber daya Amazon RDS. Untuk informasi selengkapnya, lihat Pencegahan masalah confused deputy lintas layanan.

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan `aws:SourceArn` dan kunci konteks kondisi global `aws:SourceAccount` di Amazon RDS untuk mencegah masalah confused deputy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "directoryservice.rds.amazonaws.com",
          "rds.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:rds:us-east-1:123456789012:db:mydbinstance"
        },
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        }
      }
    }
  ]
}
```

Peran ini juga harus memiliki kebijakan IAM berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
      "Action": [
        "ds:DescribeDirectories",
        "ds:AuthorizeApplication",
        "ds:UnauthorizeApplication",
        "ds:GetAuthorizedApplicationDetails"
      ],
    "Effect": "Allow",
    "Resource": "*"
    }
  ]
}
```

Langkah 4: Buat dan konfigurasikan pengguna

Anda dapat membuat pengguna dengan alat Active Directory Users and Computers, yang merupakan salah satu alat Active Directory Domain Services dan Active Directory Lightweight Directory Services. Dalam hal ini, pengguna adalah perorangan atau entitas yang memiliki akses ke direktori Anda.

Untuk membuat pengguna di direktori AWS Directory Service, Anda harus terhubung ke instans Amazon EC2 berbasis Windows yang merupakan anggota direktori AWS Directory Service. Pada saat yang sama, Anda harus masuk sebagai pengguna yang memiliki hak untuk membuat pengguna. Untuk informasi selengkapnya tentang pembuatan pengguna di Active Directory, lihat Mengelola pengguna dan grup di AWS Managed Microsoft AD dalam Panduan Administrasi AWS Directory Service.

Langkah 5: Aktifkan lalu lintas antar-VPC antara direktori dan instans DB

Jika Anda ingin mencari direktori dan instans DB dalam VPC yang sama, lewati langkah ini dan lanjutkan ke Langkah 6: Buat atau modifikasi instans Oracle DB.

Jika Anda ingin mencari direktori dan instans DB di VPC atau akun AWS yang berbeda, konfigurasikan lalu lintas antar-VPC menggunakan peering VPC atau AWS Transit Gateway. Prosedur berikut mengaktifkan lalu lintas antar-VPC menggunakan peering VPC. Ikuti petunjuk di Apa yang dimaksud dengan peering VPC? dalam Panduan Peering Amazon Virtual Private Cloud.

Untuk mengaktifkan lalu lintas VPC menggunakan peering VPC

1. Siapkan aturan perutean VPC yang sesuai untuk memastikan lalu lintas jaringan dapat berjalan dua arah.

2. Pastikan grup keamanan instans DB dapat menerima lalu lintas masuk dari grup keamanan direktori. Untuk informasi selengkapnya, lihat Praktik terbaik untuk AWS Managed Microsoft AD dalam Panduan Administrasi AWS Directory Service.
3. Pastikan tidak ada aturan daftar kontrol akses (ACL) jaringan yang memblokir lalu lintas.

Jika akun AWS lain memiliki direktori, Anda harus berbagi direktori.

Untuk berbagi direktori antara akun AWS

1. Mulai berbagi direktori dengan akun AWS yang akan digunakan untuk membuat instans DB dengan mengikuti petunjuk di Tutorial: Berbagi direktori AWS Managed Microsoft AD untuk kemudahan penggabungan Domain EC2 dalam Panduan Administrasi AWS Directory Service.
2. Masuk ke konsol AWS Directory Service menggunakan akun untuk instans DB, dan pastikan bahwa domain memiliki status SHARED sebelum melanjutkan.
3. Saat masuk ke konsol AWS Directory Service menggunakan akun untuk instans DB, catat nilai ID Direktori. Anda menggunakan ID direktori ini untuk menggabungkan instans DB ke domain.

Langkah 6: Buat atau modifikasi instans Oracle DB

Buat atau modifikasi instans Oracle DB untuk digunakan dengan direktori Anda. Anda dapat menggunakan konsol, CLI, atau RDS API untuk mengaitkan suatu instans DB dengan direktori. Anda dapat menyesuaikan waktu ini dengan cara berikut:

- Buat instans Oracle DB baru menggunakan konsol, perintah create-db-instanceCLI, atau operasi CreatedBInstance RDS API.

  Untuk petunjuk, lihat Membuat instans DB Amazon RDS.
- Ubah instans Oracle DB yang ada menggunakan konsol, perintah modify-db-instanceCLI, atau operasi ModifyDBInstance RDS API.

  Untuk petunjuk, lihat Memodifikasi instans DB Amazon RDS.
- Kembalikan instans Oracle DB dari snapshot DB menggunakan konsol, perintah CLI restore-db-instance-from-db-snapshot, atau operasi RestoreDB DBSnapshot RDS API. InstanceFrom

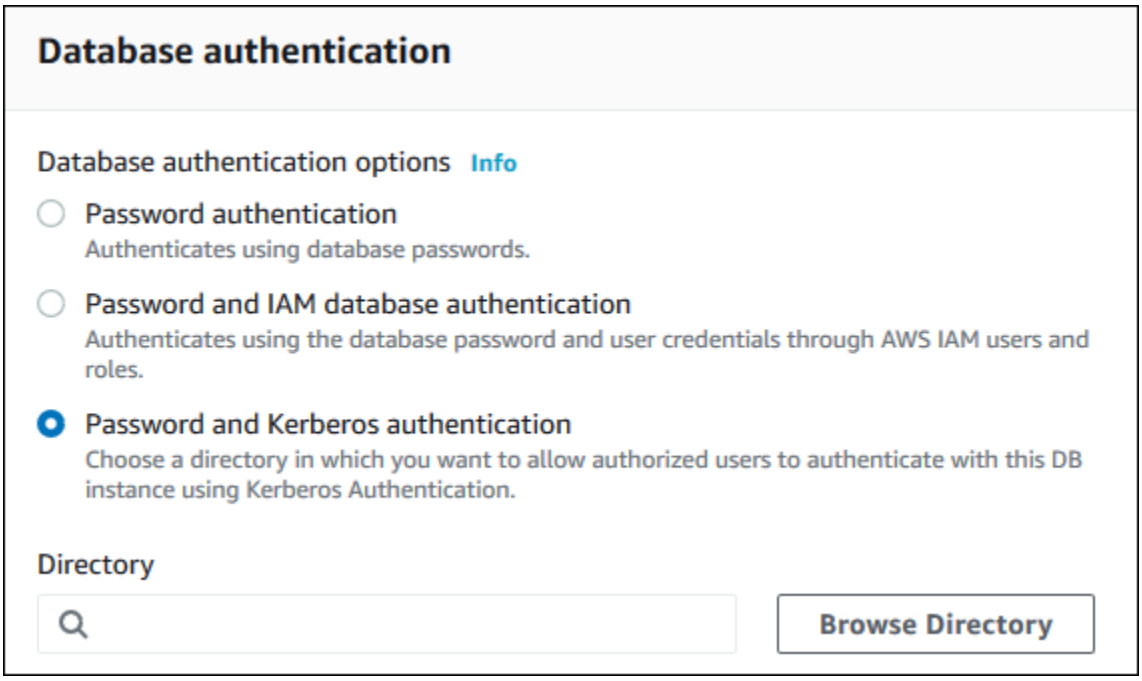
  Untuk petunjuk, lihat Memulihkan dari snapshot DB.
- Kembalikan instance Oracle DB ke point-in-time menggunakan konsol, perintah restore-db-instance-to- point-in-time CLI, atau operasi API RDS InstanceToPointInTimeRestoreDB.

Untuk petunjuknya, lihat Memulihkan instans DB dengan waktu yang ditentukan.

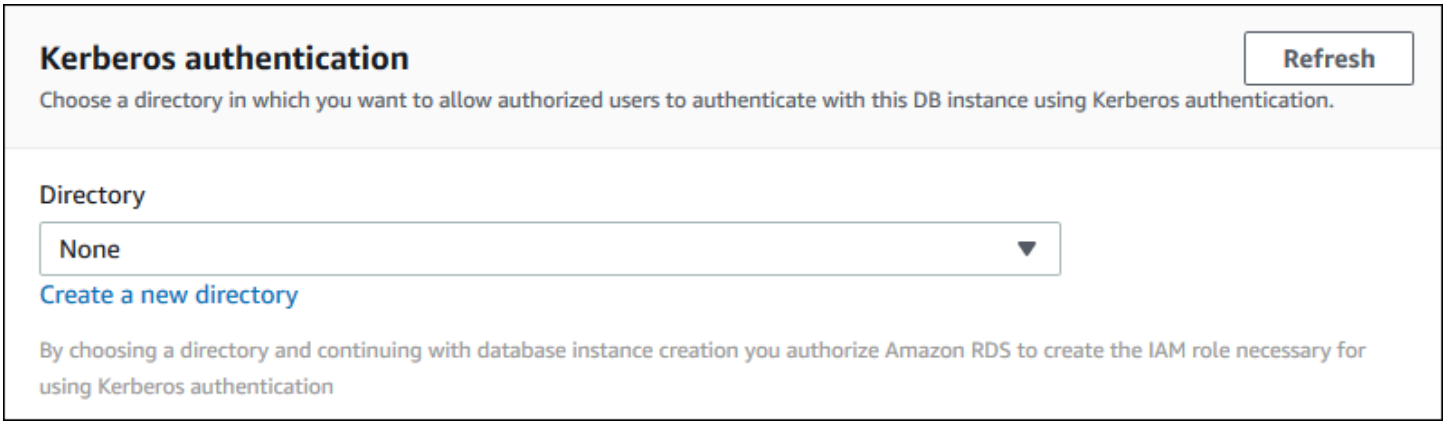
Autentikasi Kerberos hanya didukung untuk instans DB Oracle dalam VPC. Instans DB dapat berada dalam VPC yang sama dengan direktori, atau dalam VPC yang berbeda. Saat Anda membuat atau mengubah instans DB, lakukan tugas berikut:

- Sediakan pengidentifikasi domain (pengidentifikasi `d-*`) yang dihasilkan saat Anda membuat direktori.
- Beri nama peran IAM yang Anda buat.
- Pastikan bahwa grup keamanan instans DB dapat menerima lalu lintas masuk dari grup keamanan direktori dan mengirim lalu lintas keluar ke direktori.

Saat Anda menggunakan konsol untuk membuat instans DB, pilih Kata sandi dan autentikasi Kerberos di bagian Autentikasi basis data. Pilih Jelajah Direktori lalu pilih direktori, atau pilih Buat direktori baru.



Saat Anda menggunakan konsol untuk memodifikasi atau memulihkan instans DB, pilih direktori di bagian Autentikasi Kerberos atau pilih Buat direktori baru.



Saat Anda menggunakan AWS CLI, parameter berikut diperlukan agar instans DB dapat menggunakan direktori yang Anda buat:

- Untuk parameter `--domain`, gunakan pengidentifikasi domain (pengidentifikasi "d-*") yang dihasilkan saat Anda membuat direktori.
- Untuk parameter `--domain-iam-role-name`, gunakan peran yang Anda buat yang menggunakan kebijakan IAM terkelola `AmazonRDSDirectoryServiceAccess`.

Misalnya, perintah CLI berikut memodifikasi instans DB untuk menggunakan direktori.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --domain d-ID \
    --domain-iam-role-name role-name
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --domain d-ID ^
    --domain-iam-role-name role-name
```

> **Important**
>
> Jika Anda memodifikasi instans DB untuk mengaktifkan autentikasi Kerberos, reboot instans basis data setelah membuat perubahan.

> **Note**
>
> *MANAGED_SERVICE_USER* adalah akun layanan yang namanya dihasilkan secara acak oleh Directory Service for RDS. Selama penyiapan autentikasi Kerberos, RDS for Oracle membuat pengguna dengan nama yang sama dan menetapkan hak istimewa `CREATE SESSION`. Pengguna DB Oracle diidentifikasi secara eksternal sebagai *MANAGED_SERVICE_USER@EXAMPLE.COM*, dan nama domain Anda adalah *EXAMPLE.COM*. Secara berkala, RDS menggunakan kredensial yang disediakan oleh Directory Service untuk masuk ke basis data Oracle Anda. Setelah itu, RDS langsung menghancurkan cache tiket.

Langkah 7: Buat login Oracle untuk autentikasi Kerberos

Gunakan kredensial pengguna utama Amazon RDS untuk terhubung ke instans DB Oracle saat Anda mengerjakan instans DB lainnya. Instans DB digabungkan ke domain AWS Managed Microsoft AD. Dengan demikian, Anda bisa menyediakan login dan pengguna Oracle dari pengguna dan grup Microsoft Active Directory di domain Anda. Untuk mengelola izin basis data, Anda memberikan dan mencabut izin Oracle standar untuk login ini.

Untuk mengizinkan pengguna Microsoft Active Directory melakukan autentikasi dengan Oracle

1. Hubungkan ke instans DB Oracle menggunakan kredensial pengguna utama Amazon RDS Anda.
2. Buat pengguna yang diautentikasi secara eksternal di basis data Oracle.

   Pada contoh berikut, ganti *KRBUSER@CORP.EXAMPLE.COM* dengan nama pengguna dan nama domain.

   ```
   CREATE USER "KRBUSER@CORP.EXAMPLE.COM" IDENTIFIED EXTERNALLY;
   GRANT CREATE SESSION TO "KRBUSER@CORP.EXAMPLE.COM";
   ```

   Pengguna (baik manusia maupun aplikasi) dari domain Anda sekarang dapat terhubung ke instans DB Oracle dari mesin klien yang tergabung dalam domain menggunakan autentikasi Kerberos.

Langkah 8: Konfigurasikan klien Oracle

Untuk mengonfigurasi klien Oracle, penuhi persyaratan berikut:

- Buat file konfigurasi bernama krb5.conf (Linux) atau krb5.ini (Windows) untuk menunjuk ke domain. Konfigurasikan klien Oracle untuk menggunakan konfigurasi file ini.
- Pastikan lalu lintas dapat mengalir di antara host klien dan AWS Directory Service melalui DNS port 53 melalui TCP/UDP, port Kerberos (88 dan 464 untuk AWS Directory Service terkelola) melalui TCP, dan LDAP port 389 melalui TCP.
- Pastikan lalu lintas dapat mengalir di antara host klien dan instans DB melalui port basis data.

Berikut adalah contoh konten untuk AWS Managed Microsoft AD.

```
[libdefaults]
 default_realm = EXAMPLE.COM
[realms]
 EXAMPLE.COM = {
  kdc = example.com
  admin_server = example.com
 }
[domain_realm]
 .example.com = CORP.EXAMPLE.COM
 example.com = CORP.EXAMPLE.COM
```

Berikut adalah contoh konten untuk Microsoft AD on-premise. Di file krb5.conf atau krb5.ini Anda, ganti *on-prem-ad-server-name dengan nama* server AD lokal Anda.

```
[libdefaults]
 default_realm = ONPREM.COM
[realms]
 AWSAD.COM = {
  kdc = awsad.com
  admin_server = awsad.com
 }
 ONPREM.COM = {
  kdc = on-prem-ad-server-name
  admin_server = on-prem-ad-server-name
 }
[domain_realm]
 .awsad.com = AWSAD.COM
 awsad.com= AWSAD.COM
 .onprem.com = ONPREM.COM
 onprem.com= ONPREM.COM
```

> **Note**
>
> Setelah mengonfigurasi file krb5.ini atau krb5.conf, sebaiknya reboot server.

Berikut adalah contoh konten sqlnet.ora untuk konfigurasi SQL*Plus:

```
SQLNET.AUTHENTICATION_SERVICES=(KERBEROS5PRE,KERBEROS5)
SQLNET.KERBEROS5_CONF=path_to_krb5.conf_file
```

Untuk contoh konfigurasi SQL Developer, lihat Dokumen 1609359.1 dari Dukungan Oracle.

Mengelola instans DB dalam domain

Anda bisa menggunakan konsol, CLI, atau RDS API untuk mengelola instans DB dan hubungannya dengan Microsoft Active Directory. Misalnya, Anda dapat mengaitkan Microsoft Active Directory untuk mengaktifkan autentikasi Kerberos. Anda juga dapat membatalkan pengaitan Microsoft Active Directory untuk menonaktifkan autentikasi Kerberos. Anda juga dapat memindahkan instans DB untuk diautentikasi secara eksternal oleh satu Microsoft Active Directory ke yang lain.

Misalnya, dengan CLI, Anda dapat melakukan hal berikut:

- Untuk mengulangi upaya aktivasi autentikasi Kerberos untuk keanggotaan yang gagal, gunakan perintah CLI modify-db-instance dan tentukan ID direktori keanggotaan saat ini untuk opsi `--domain`.
- Untuk menonaktifkan autentikasi Kerberos pada instans DB, gunakan perintah CLI modify-db-instance dan tentukan `none` untuk opsi `--domain`.
- Untuk memindahkan instans DB dari satu domain ke domain lain, gunakan perintah CLI modify-db-instance dan tentukan pengidentifikasi domain pada domain baru untuk opsi `--domain`.

Melihat status keanggotaan domain

Instans DB yang telah Anda buat atau modifikasi akan menjadi anggota domain. Anda dapat melihat status keanggotaan domain untuk instans DB di konsol atau dengan menjalankan perintah CLI describe-db-instances. Status instans DB dapat berupa salah satu dari daftar berikut:

- `kerberos-enabled` – Instans DB mengaktifkan autentikasi Kerberos.
- `enabling-kerberos` – AWS sedang mengaktifkan autentikasi Kerberos pada instans DB ini.

- `pending-enable-kerberos` – Aktivasi autentikasi Kerberos pada instans DB ini tertunda.
- `pending-maintenance-enable-kerberos` – AWS akan mencoba mengaktifkan autentikasi Kerberos pada instans DB pada jendela pemeliharaan terjadwal berikutnya.
- `pending-disable-kerberos` – Penonaktifan autentikasi Kerberos pada instans DB ini tertunda.
- `pending-maintenance-disable-kerberos` – AWS akan mencoba menonaktifkan autentikasi Kerberos pada instans DB pada jendela pemeliharaan terjadwal berikutnya.
- `enable-kerberos-failed` – Masalah konfigurasi membuat AWS tidak dapat mengaktifkan autentikasi Kerberos pada instans DB. Perbaiki masalah konfigurasi tersebut sebelum menerbitkan ulang perintah untuk memodifikasi instans DB.
- `disabling-kerberos` – AWS sedang menonaktifkan autentikasi Kerberos pada instans DB ini.

Permintaan untuk mengaktifkan autentikasi Kerberos dapat gagal karena masalah koneksi jaringan atau kesalahan peran IAM. Jika upaya aktivasi autentikasi Kerberos gagal saat Anda membuat atau memodifikasi instans DB, pastikan peran IAM yang digunakan sudah benar. Kemudian modifikasi instans DB untuk menggabungkan domain.

> **Note**
>
> Hanya autentikasi Kerberos dengan Amazon RDS for Oracle yang mengirimkan lalu lintas ke server DNS domain. Semua permintaan DNS lainnya diperlakukan sebagai akses jaringan keluar pada instans DB Anda yang menjalankan Oracle. Untuk informasi selengkapnya tentang akses jaringan keluar dengan Amazon RDS for Oracle, lihat Menyiapkan server DNS kustom.

Rotasi paksa kunci Kerberos

Kunci rahasia dibagikan di antara instans DB Amazon RDS for Oracle dan AWS Managed Microsoft AD. Kunci ini dirotasi secara otomatis setiap 45 hari. Anda dapat menggunakan prosedur Amazon RDS berikut untuk memaksa rotasi kunci ini.

```
SELECT rdsadmin.rdsadmin_kerberos_auth_tasks.rotate_kerberos_keytab AS TASK_ID FROM
 DUAL;
```

> **Note**
>
> Dalam konfigurasi replika baca, prosedur ini hanya tersedia pada instans DB sumber dan tidak tersedia pada replika baca.

Pernyataan `SELECT` mengembalikan ID tugas dalam jenis data `VARCHAR2`. Anda dapat melihat status tugas yang sedang berlangsung di file bdump. File bdump terletak di direktori `/rdsdbdata/log/trace`. Setiap nama file bdump memiliki format berikut.

```
dbtask-task-id.log
```

Anda dapat melihat hasilnya dengan menampilkan file output tugas.

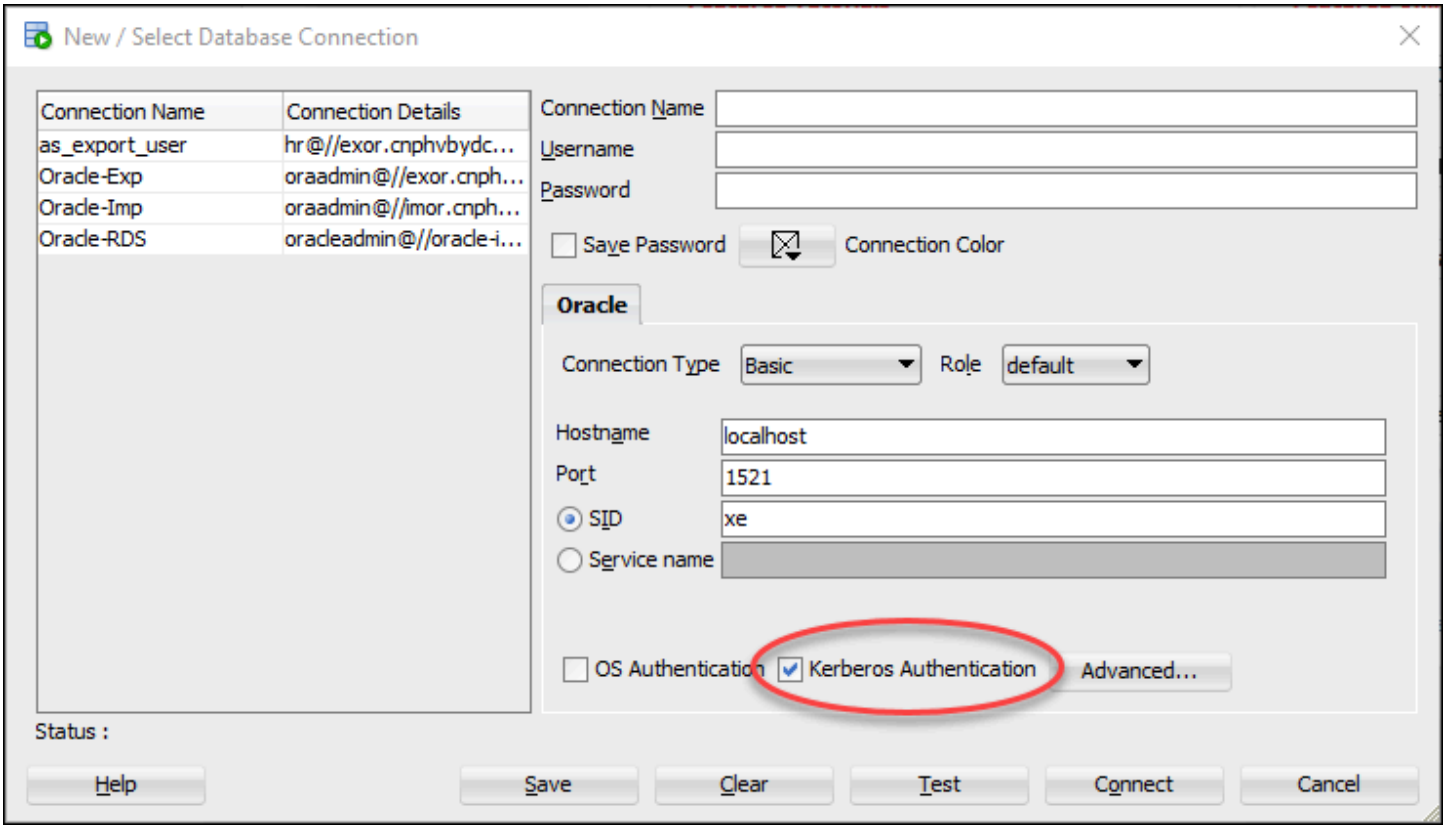
```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-task-id.log'));
```

Ganti *`task-id`* dengan ID tugas yang dikembalikan oleh prosedur.

> **Note**
>
> Tugas dijalankan secara asinkron.

Menghubungkan ke Oracle dengan autentikasi Kerberos

Bagian ini menganggap bahwa klien Oracle Anda telah disiapkan sebagaimana dijelaskan dalam Langkah 8: Konfigurasikan klien Oracle. Untuk terhubung ke DB Oracle dengan autentikasi Kerberos, masuk menggunakan jenis autentikasi Kerberos. Misalnya, setelah meluncurkan Oracle SQL Developer, pilih Autentikasi Kerberos sebagai jenis autentikasi, seperti yang ditunjukkan berikut ini.



Untuk terhubung ke Oracle dengan autentikasi Kerberos dengan SQL*Plus:

1. Pada jendela perintah, jalankan perintah berikut:

   ```
   kinit username
   ```

   Ganti *username* dengan nama pengguna dan, saat diminta, masukkan kata sandi yang disimpan dalam Microsoft Active Directory untuk pengguna.

2. Buka SQL*Plus dan sambungkan menggunakan nama DNS dan nomor port untuk instans DB Oracle.

   Untuk informasi selengkapnya tentang menghubungkan ke instans DB Oracle di SQL*Plus, lihat Menghubungkan ke instans DB menggunakan SQL*Plus.

Mengonfigurasi akses UTL_HTTP menggunakan sertifikat dan dompet Oracle

Amazon RDS mendukung akses jaringan keluar pada instans DB RDS for Oracle Anda. Untuk menghubungkan instans DB ke jaringan, Anda dapat menggunakan paket PL/SQL berikut:

`UTL_HTTP`

Paket ini membuat panggilan HTTP dari SQL dan PL/SQL. Anda dapat menggunakannya untuk mengakses data di Internet melalui HTTP. Untuk informasi lebih lanjut, lihat UTL_HTTP dalam dokumentasi Oracle.

`UTL_TCP`

Paket ini menyediakan fungsi akses sisi klien TCP/IP di PL/SQL. Paket ini berguna untuk aplikasi PL/SQL yang menggunakan protokol Internet dan email. Untuk informasi lebih lanjut, lihat UTL_TCP dalam dokumentasi Oracle.

`UTL_SMTP`

Paket ini menyediakan antarmuka untuk perintah SMTP yang memungkinkan klien mengirim email ke server SMTP. Untuk informasi lebih lanjut, lihat UTL_SMTP dalam dokumentasi Oracle.

Dengan menyelesaikan tugas-tugas berikut, Anda dapat mengonfigurasi `UTL_HTTP.REQUEST` agar bisa berfungsi dengan situs web yang memerlukan sertifikat autentikasi klien selama SSL handshake. Anda juga dapat mengonfigurasi autentikasi kata sandi untuk akses `UTL_HTTP` ke situs web dengan memodifikasi perintah pembuatan dompet Oracle dan prosedur `DBMS_NETWORK_ACL_ADMIN.APPEND_WALLET_ACE`. Untuk informasi selengkapnya, lihat DBMS_NETWORK_ACL_ADMIN dalam dokumentasi Oracle Database.

> **Note**
>
> Anda dapat mengadaptasi tugas berikut untuk `UTL_SMTP`, yang memungkinkan Anda mengirim email melalui SSL/TLS (termasuk Amazon Simple Email Service).

Topik

- Pertimbangan saat mengonfigurasi akses UTL_HTTP
- Langkah 1: Mendapatkan sertifikat root untuk situs web

- Langkah 2: Buat dompet Oracle
- Langkah 3: Unduh dompet Oracle Anda ke RDS Anda untuk instans Oracle
- Langkah 4: Berikan izin pengguna untuk dompet Oracle
- Langkah 5: Konfigurasikan akses ke situs web dari instans DB Anda
- Langkah 6: Uji koneksi dari instans DB Anda ke situs web

Pertimbangan saat mengonfigurasi akses UTL_HTTP

Sebelum mengonfigurasi akses, pertimbangkan hal berikut:

- Anda dapat menggunakan SMTP dengan opsi UTL_MAIL. Untuk informasi selengkapnya, lihat Oracle UTL_MAIL.
- Nama Domain Name Server (DNS) dari host jarak jauh dapat berupa:
  - Dapat diatasi secara publik.
  - Titik akhir instans DB Amazon RDS.
  - Dapat diatasi melalui server DNS kustom. Untuk informasi selengkapnya, lihat Menyiapkan server DNS kustom.
  - Nama DNS privat dari instans Amazon EC2 dalam VPC yang sama atau VPC tersambung. Dalam hal ini, pastikan bahwa nama dapat diatasi melalui server DNS kustom. Sebagai alternatif, untuk menggunakan DNS yang disediakan oleh Amazon, Anda dapat mengaktifkan atribut `enableDnsSupport` dalam pengaturan VPC dan mengaktifkan dukungan resolusi DNS untuk koneksi peering VPC. Untuk informasi lebih lanjut, lihat Dukungan DNS dalam VPC Anda dan Memodifikasi koneksi peering VPC Anda.
  - Untuk terkoneksi dengan aman ke sumber daya SSL/TLS jarak jauh, kami sarankan Anda membuat dan mengunggah dompet Oracle kustom. Dengan menggunakan integrasi Amazon S3 dengan fitur Amazon RDS for Oracle, Anda dapat mengunduh dompet dari Amazon S3 ke instans DB Oracle. Untuk informasi tentang integrasi Amazon S3 untuk Oracle, lihat Integrasi Amazon S3.
- Anda dapat membuat tautan basis data antara instans DB Oracle melalui titik akhir SSL/TLS jika opsi Oracle SSL dikonfigurasi untuk setiap instans. Tidak diperlukan konfigurasi lebih lanjut. Untuk informasi selengkapnya, lihat Lapisan Soket Aman Oracle.

Langkah 1: Mendapatkan sertifikat root untuk situs web

Agar instans DB RDS for Oracle dapat membuat koneksi aman ke situs web, tambahkan sertifikat root CA. Amazon RDS menggunakan sertifikat root untuk menandatangani sertifikat situs web ke dompet Oracle.

Anda bisa mendapatkan sertifikat root dengan berbagai cara. Misalnya, Anda dapat melakukan hal berikut:

1. Gunakan server web untuk mengunjungi situs web yang diamankan oleh sertifikat tersebut.
2. Unduh sertifikat root yang digunakan untuk penandatanganan.

Untuk layanan AWS, sertifikat root biasanya berada di Repositori layanan kepercayaan Amazon.

Langkah 2: Buat dompet Oracle

Buat dompet Oracle yang berisi sertifikat server web dan sertifikat autentikasi klien. Instans RDS Oracle menggunakan sertifikat server web untuk membuat koneksi aman ke situs web. Situs web tersebut membutuhkan sertifikat klien untuk mengautentikasi pengguna basis data Oracle.

Anda sebaiknya mengonfigurasi koneksi aman tanpa menggunakan sertifikat klien untuk autentikasi. Dalam kasus ini, Anda dapat melewati langkah-langkah keystore Java dalam prosedur berikut.

Untuk membuat dompet Oracle

1. Tempatkan sertifikat root dan klien dalam satu direktori, dan kemudian ubah ke direktori ini.
2. Ubah sertifikat klien.p12 ke keystore Java.

> **Note**
>
> Jika Anda tidak menggunakan sertifikat klien untuk autentikasi, Anda dapat melewati langkah ini.

Contoh berikut mengonversi sertifikat klien bernama *client_certificate.p12* menjadi keystore Java bernama *client_keystore.jks*. Keystore tersebut kemudian disertakan dalam dompet Oracle. Kata sandi keystore adalah *P12PASSWORD*.

```
orapki wallet pkcs12_to_jks -wallet ./client_certificate.p12 -
jksKeyStoreLoc ./client_keystore.jks -jksKeyStorepwd P12PASSWORD
```

3. Buat direktori untuk dompet Oracle Anda yang berbeda dari direktori sertifikat.

   Contoh berikut membuat direktori `/tmp/wallet`.

   ```
   mkdir -p /tmp/wallet
   ```

4. Buat dompet Oracle di direktori dompet Anda.

   Contoh berikut menetapkan kata sandi dompet Oracle menjadi *P12PASSWORD*, yaitu kata sandi yang juga digunakan oleh keystore Java pada langkah sebelumnya. Penggunaan kata sandi yang sama memang memudahkan, tetapi tidak wajib. Parameter `-auto_login` mengaktifkan fitur masuk otomatis, sehingga Anda tidak perlu menentukan kata sandi setiap kali Anda ingin mengaksesnya.

   > **Note**
   >
   > Tetapkan kata sandi selain perintah yang ditampilkan di sini sebagai praktik terbaik keamanan.

   ```
   orapki wallet create -wallet /tmp/wallet -pwd P12PASSWORD -auto_login
   ```

5. Tambahkan keystore Java ke dompet Oracle Anda.

   > **Note**
   >
   > Jika Anda tidak menggunakan sertifikat klien untuk autentikasi, Anda dapat melewati langkah ini.

   Contoh berikut menambahkan keystore *client_keystore.jks* ke dompet Oracle bernama */tmp/wallet*. Dalam contoh ini, Anda menentukan kata sandi yang sama untuk keystore Java dan dompet Oracle.

```
orapki wallet jks_to_pkcs12 -wallet /tmp/wallet -pwd P12PASSWORD -
keystore ./client_keystore.jks -jkspwd P12PASSWORD
```

6. Tambahkan sertifikat root untuk situs web target Anda ke dompet Oracle.

   Contoh berikut menambahkan sertifikat bernama *Root_CA.cer*.

   ```
   orapki wallet add -wallet /tmp/wallet -trusted_cert -cert ./Root_CA.cer -
   pwd P12PASSWORD
   ```

7. Tambahkan sertifikat perantara apa pun.

   Contoh berikut menambahkan sertifikat bernama *Intermediate.cer*. Ulangi langkah ini seperlunya untuk memuat semua sertifikat perantara.

   ```
   orapki wallet add -wallet /tmp/wallet -trusted_cert -cert ./Intermediate.cer -
   pwd P12PASSWORD
   ```

8. Konfirmasikan bahwa dompet Oracle Anda yang baru dibuat memiliki sertifikat yang diperlukan.

   ```
   orapki wallet display -wallet /tmp/wallet -pwd P12PASSWORD
   ```

Langkah 3: Unduh dompet Oracle Anda ke RDS Anda untuk instans Oracle

Pada langkah ini, Anda mengunggah dompet Oracle Anda ke Amazon S3, lalu mengunduh dompet tersebut dari Amazon S3 ke instans RDS for Oracle Anda.

Untuk mengunduh dompet Oracle Anda ke instans DB RDS for Oracle Anda

1. Lengkapi persyaratan untuk integrasi Amazon S3 dengan Oracle, dan tambahkan opsi `S3_INTEGRATION` untuk instans DB Oracle Anda. Pastikan bahwa peran IAM untuk opsi tersebut memiliki akses ke bucket Amazon S3 yang Anda gunakan.

   Untuk informasi selengkapnya, lihat Integrasi Amazon S3.

2. Masuk ke instans DB Anda sebagai pengguna master, lalu buat direktori Oracle untuk menyimpan dompet Oracle.

   Contoh berikut membuat direktori Oracle bernama *WALLET_DIR*.

```
EXEC rdsadmin.rdsadmin_util.create_directory('WALLET_DIR');
```

Untuk informasi selengkapnya, lihat Membuat dan menghapus direktori di ruang penyimpanan data utama.

3. Unggah dompet Oracle ke bucket Amazon S3.

   Anda dapat menggunakan teknik unggahan apa pun yang didukung.

4. Jika Anda mengunggah ulang dompet Oracle, hapus dompet yang ada. Jika tidak, lewati ke langkah berikutnya.

   Contoh berikut menghapus dompet yang ada, yang bernama *cwallet.sso*.

   ```
   EXEC UTL_FILE.FREMOVE ('WALLET_DIR','cwallet.sso');
   ```

5. Unduh dompet Oracle dari bucket Amazon S3 ke instans DB Oracle.

   Contoh berikut mengunduh dompet bernama *cwallet.sso* dari bucket Amazon S3 bernama *my_s3_bucket* ke direktori instans DB bernama *WALLET_DIR*.

   ```
   SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
         p_bucket_name    =>  'my_s3_bucket',
         p_s3_prefix      =>  'cwallet.sso',
         p_directory_name =>  'WALLET_DIR')
      AS TASK_ID FROM DUAL;
   ```

6. (Opsional) Unduh dompet Oracle yang dilindungi kata sandi.

   Unduh dompet ini hanya jika Anda ingin mewajibkan kata sandi untuk setiap penggunaan dompet. Contoh berikut mengunduh dompet *ewallet.p12* yang dilindungi kata sandi.

   ```
   SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
         p_bucket_name    =>  'my_s3_bucket',
         p_s3_prefix      =>  'ewallet.p12',
         p_directory_name =>  'WALLET_DIR')
      AS TASK_ID FROM DUAL;
   ```

7. Periksa status tugas DB Anda.

   Ganti ID tugas yang dikembalikan dari langkah-langkah sebelumnya untuk *dbtask-1234567890123-4567.log* dalam contoh berikut.

```
SELECT TEXT FROM
 TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-1234567890123-4567.log'));
```

8. Periksa isi direktori yang Anda gunakan untuk menyimpan dompet Oracle.

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir(p_directory => 'WALLET_DIR'));
```

Untuk informasi selengkapnya, lihat Membuat daftar file di direktori instans DB.

Langkah 4: Berikan izin pengguna untuk dompet Oracle

Anda dapat membuat pengguna basis data baru atau mengonfigurasi pengguna yang sudah ada. Dalam kedua kasus tersebut, Anda harus mengonfigurasi pengguna untuk mengakses dompet Oracle untuk koneksi yang aman dan autentikasi klien menggunakan sertifikat.

Untuk memberikan izin pengguna untuk dompet Oracle

1. Masuk ke instans DB RDS for Oracle sebagai pengguna master.
2. Jika Anda tidak ingin mengonfigurasi pengguna yang sudah ada, buat pengguna baru. Jika tidak, lewati ke langkah berikutnya.

   Contoh berikut membuat pengguna basis data bernama *my-user*.

```
CREATE USER my-user IDENTIFIED BY my-user-pwd;
GRANT CONNECT TO my-user;
```

3. Berikan izin kepada pengguna basis data Anda di direktori yang berisi dompet Oracle Anda.

   Contoh berikut memberikan akses baca ke pengguna *my-user* di direktori *WALLET_DIR*.

```
GRANT READ ON DIRECTORY WALLET_DIR TO my-user;
```

4. Berikan izin kepada pengguna basis data Anda untuk menggunakan paket UTL_HTTP.

   Program PL/SQL berikut memberikan akses UTL_HTTP ke pengguna *my-user*.

```
BEGIN
  rdsadmin.rdsadmin_util.grant_sys_object('UTL_HTTP', UPPER('my-user'));
  END;
```

```
/
```

5. Berikan izin kepada pengguna basis data Anda untuk menggunakan paket `UTL_FILE`.

   Program PL/SQL berikut memberikan akses `UTL_FILE` ke pengguna *`my-user`*.

```
BEGIN
  rdsadmin.rdsadmin_util.grant_sys_object('UTL_FILE', UPPER('my-user'));
  END;
/
```

Langkah 5: Konfigurasikan akses ke situs web dari instans DB Anda

Pada langkah ini, Anda mengonfigurasi agar pengguna basis data Oracle Anda dapat terhubung ke situs web target Anda menggunakan UTL_HTTP, dompet Oracle yang Anda unggah, dan sertifikat klien. Untuk informasi selengkapnya, lihat Configuring Access Control to an Oracle Wallet dalam dokumentasi Oracle Database.

Untuk mengonfigurasikan akses ke situs web dari instans DB RDS for Oracle Anda

1. Masuk ke instans DB RDS for Oracle sebagai pengguna master.
2. Buat Entri Kontrol Akses Host (ACE) untuk pengguna Anda dan situs web target pada port yang aman.

   Contoh berikut mengonfigurasi *`my-user`* untuk mengakses *`secret.encrypted-website.com`* pada port aman 443.

```
BEGIN
  DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE(
    host       => 'secret.encrypted-website.com',
    lower_port => 443,
    upper_port => 443,
    ace        => xs$ace_type(privilege_list => xs$name_list('http'),
                              principal_name => 'my-user',
                              principal_type => xs_acl.ptype_db));
                           -- If the program unit results in PLS-00201, set
                           -- the principal_type parameter to 2 as follows:
                           -- principal_type => 2));
END;
/
```

> **Important**
>
> Unit program sebelumnya dapat mengakibatkan kesalahan berikut: `PLS-00201: identifier 'XS_ACL' must be declared`. Jika kesalahan ini dikembalikan, ganti baris yang menetapkan nilai ke `principal_type` dengan baris berikut, lalu jalankan kembali unit program tersebut:
>
> ```
> principal_type => 2));
> ```
>
> Untuk informasi selengkapnya tentang konstanta dalam `XS_ACL` paket PL/SQL, lihat Real Application Security Administrator's and Developer's Guide dalam dokumentasi Oracle Database.

Untuk informasi selengkapnya, lihat Configuring Access Control for External Network Services dalam dokumentasi Oracle Database.

3. (Opsional) Buat ACE untuk pengguna Anda dan situs web target pada port standar.

   Anda mungkin perlu menggunakan port standar jika beberapa halaman web disajikan dari port server web standar (80), bukan port aman (443).

   ```
   BEGIN
     DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE(
       host       => 'secret.encrypted-website.com',
       lower_port => 80,
       upper_port => 80,
       ace        => xs$ace_type(privilege_list => xs$name_list('http'),
                                 principal_name => 'my-user',
                                 principal_type => xs_acl.ptype_db));
                               -- If the program unit results in PLS-00201, set
                               -- the principal_type parameter to 2 as follows:
                               -- principal_type => 2));
   END;
   /
   ```

4. Konfirmasikan bahwa entri kontrol akses sudah ada.

   ```
   SET LINESIZE 150
   COLUMN HOST FORMAT A40
   ```

```
COLUMN ACL FORMAT A50

SELECT HOST, LOWER_PORT, UPPER_PORT, ACL
  FROM DBA_NETWORK_ACLS
ORDER BY HOST;
```

5. Berikan izin kepada pengguna basis data Anda untuk menggunakan paket UTL_HTTP.

   Program PL/SQL berikut memberikan akses UTL_HTTP ke pengguna *my-user*.

```
BEGIN
  rdsadmin.rdsadmin_util.grant_sys_object('UTL_HTTP', UPPER('my-user'));
  END;
/
```

6. Konfirmasikan bahwa daftar kontrol akses terkait sudah ada.

```
SET LINESIZE 150
COLUMN ACL FORMAT A50
COLUMN PRINCIPAL FORMAT A20
COLUMN PRIVILEGE FORMAT A10

SELECT ACL, PRINCIPAL, PRIVILEGE, IS_GRANT,
       TO_CHAR(START_DATE, 'DD-MON-YYYY') AS START_DATE,
       TO_CHAR(END_DATE, 'DD-MON-YYYY') AS END_DATE
  FROM DBA_NETWORK_ACL_PRIVILEGES
ORDER BY ACL, PRINCIPAL, PRIVILEGE;
```

7. Berikan izin kepada pengguna basis data Anda untuk menggunakan sertifikat untuk autentikasi klien dan dompet Oracle Anda untuk koneksi.

   > **Note**
   >
   > Jika Anda tidak menggunakan sertifikat klien untuk autentikasi, Anda dapat melewati langkah ini.

```
DECLARE
  l_wallet_path all_directories.directory_path%type;
BEGIN
  SELECT DIRECTORY_PATH
    INTO l_wallet_path
```

```
    FROM ALL_DIRECTORIES
   WHERE UPPER(DIRECTORY_NAME)='WALLET_DIR';
  DBMS_NETWORK_ACL_ADMIN.APPEND_WALLET_ACE(
    wallet_path => 'file:/' || l_wallet_path,
    ace         =>  xs$ace_type(privilege_list => xs
$name_list('use_client_certificates'),
                                principal_name => 'my-user',
                                principal_type => xs_acl.ptype_db));
END;
/
```

Langkah 6: Uji koneksi dari instans DB Anda ke situs web

Pada langkah ini, Anda mengonfigurasi agar pengguna basis data Anda dapat terhubung ke situs web tersebut menggunakan UTL_HTTP, dompet Oracle yang Anda unggah, dan sertifikat klien.

Untuk mengonfigurasikan akses ke situs web dari instans DB RDS for Oracle Anda

1. Masuk ke instans DB RDS for Oracle Anda sebagai pengguna basis data dengan izin UTL_HTTP.
2. Konfirmasikan bahwa koneksi ke situs web target Anda dapat menyelesaikan alamat host.

   Contoh berikut mendapatkan alamat host dari *secret.encrypted-website.com*.

   ```
   SELECT UTL_INADDR.GET_HOST_ADDRESS(host => 'secret.encrypted-website.com')
     FROM DUAL;
   ```
3. Uji koneksi yang gagal.

   Kueri berikut gagal karena UTL_HTTP memerlukan lokasi dompet Oracle dengan sertifikat.

   ```
   SELECT UTL_HTTP.REQUEST('secret.encrypted-website.com') FROM DUAL;
   ```
4. Uji akses situs web menggunakan UTL_HTTP.SET_WALLET dan memilih dari DUAL.

   ```
   DECLARE
     l_wallet_path all_directories.directory_path%type;
   BEGIN
     SELECT DIRECTORY_PATH
       INTO l_wallet_path
   ```

```
    FROM ALL_DIRECTORIES
   WHERE UPPER(DIRECTORY_NAME)='WALLET_DIR';
  UTL_HTTP.SET_WALLET('file:/' || l_wallet_path);
END;
/

SELECT UTL_HTTP.REQUEST('secret.encrypted-website.com') FROM DUAL;
```

5. (Opsional) Uji akses situs web dengan menyimpan kueri Anda dalam variabel dan menggunakan `EXECUTE IMMEDIATE`.

```
DECLARE
  l_wallet_path all_directories.directory_path%type;
  v_webpage_sql VARCHAR2(1000);
  v_results     VARCHAR2(32767);
BEGIN
  SELECT DIRECTORY_PATH
    INTO l_wallet_path
    FROM ALL_DIRECTORIES
   WHERE UPPER(DIRECTORY_NAME)='WALLET_DIR';
  v_webpage_sql := 'SELECT UTL_HTTP.REQUEST(''secret.encrypted-website.com'', ''''',
 ''file:/' ||l_wallet_path||''''') FROM DUAL';
  DBMS_OUTPUT.PUT_LINE(v_webpage_sql);
  EXECUTE IMMEDIATE v_webpage_sql INTO v_results;
  DBMS_OUTPUT.PUT_LINE(v_results);
END;
/
```

6. (Opsional) Temukan lokasi sistem file direktori dompet Oracle Anda.

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir(p_directory => 'WALLET_DIR'));
```

Gunakan output dari perintah sebelumnya untuk membuat permintaan HTTP. Misalnya, jika direktori adalah *rdsdbdata/userdirs/01*, jalankan kueri berikut.

```
SELECT UTL_HTTP.REQUEST('https://secret.encrypted-website.com/', '',
 'file://rdsdbdata/userdirs/01')
FROM   DUAL;
```

Bekerja dengan CDB di RDS for Oracle

Dalam arsitektur multi-penghuni Oracle, basis data kontainer (CDB) dapat menyertakan basis data pluggable (PDB) buatan pelanggan. Untuk informasi selengkapnya tentang CDB, lihat Introduction to the Multitenant Architecture dalam dokumentasi Basis Data Oracle.

Topik

Ikhtisar CDB RDS for Oracle

Anda dapat membuat instans DB RDS for Oracle sebagai basis data kontainer (CDB) ketika Anda menjalankan Oracle Database 19c atau yang lebih tinggi. Mulai di Oracle Database 21c, semua basis data adalah CDB. CDB berbeda dari non-CDB karena dapat berisi pluggable database (PDB), yang disebut database penyewa di RDS untuk Oracle. PDB adalah kumpulan portabel skema dan objek yang muncul ke aplikasi sebagai basis data terpisah.

Anda membuat basis data penghuni awal (PDB) saat membuat instans CDB Anda. Di RDS for Oracle, aplikasi klien Anda berinteraksi dengan PDB, bukan CDB. Pengalaman pengguna dengan PDB sebagian besar identik dengan pengalaman pengguna dengan non-CDB.

Topik

- Akun pengguna dan hak istimewa dalam CDB
- Kelompok grup parameter dalam CDB
- Batasan CDB RDS for Oracle

Konfigurasi multi-penghuni pada arsitektur CDB

RDS for Oracle mendukung konfigurasi multi-penyewa dari arsitektur multipenyewa Oracle, juga disebut arsitektur CDB. Dalam konfigurasi ini, instans RDS for Oracle CDB Anda dapat berisi 1–30 basis data penghuni, bergantung pada edisi basis data dan lisensi opsi apa pun yang diperlukan. Dalam basis data Oracle, basis data penghuni adalah PDB. Instans DB Anda harus menggunakan basis data Oracle rilis 19.0.0.0.ru-2022-01.rur-2022.r1 atau yang lebih tinggi.

> **Note**
>
> Fitur Amazon RDS disebut "multi-penghuni" dan bukannya "multipenghuni" karena merupakan kemampuan platform RDS, bukan hanya mesin Oracle DB. Istilah “Oracle multipenghuni” mengacu secara eksklusif ke arsitektur basis data Oracle, yang kompatibel dengan deployment RDS dan on-premise.

Anda dapat mengonfigurasi pengaturan berikut:

- Nama basis data penghuni
- Nama pengguna master basis data penghuni
- Kata sandi master basis data penghuni
- Set karakter basis data penghuni
- Set karakter nasional basis data penghuni

Set karakter basis data penghuni dapat berbeda dengan set karakter CDB. Hal yang sama berlaku untuk set karakter nasional. Setelah membuat basis data penghuni awal, Anda dapat membuat, memodifikasi, atau menghapus basis data penghuni menggunakan RDS API. `RDSCDB` adalah nama bawaan untuk CDB dan tidak dapat diubah. Lihat informasi yang lebih lengkap di Pengaturan untuk instans DB dan Memodifikasi RDS untuk basis data penghuni Oracle.

Konfigurasi satu penghuni pada arsitektur CDB

RDS for Oracle mendukung konfigurasi warisan dari arsitektur multipenghuni Oracle yang disebut sebagai konfigurasi penghuni tunggal. Dalam konfigurasi ini, instans CDB RDS for Oracle hanya dapat berisi satu penghuni (PDB). Anda tidak dapat membuat PDB lainnya nanti.

Opsi pembuatan dan konversi untuk CDB

Oracle Database 21c hanya mendukung CDB, sedangkan Oracle Database 19c mendukung CDB dan non-CDB. Semua instans CDB RDS for Oracle mendukung konfigurasi multi-penghuni dan satu penghuni.

Opsi pembuatan, konversi, dan peningkatan untuk arsitektur basis data Oracle

Tabel berikut menunjukkan opsi arsitektur yang berbeda untuk membuat dan meningkatkan basis data RDS for Oracle.

Rilis	Opsi pembuatan basis data	Opsi konversi arsitektur	Target peningkatan versi utama
Oracle Database 21c	Khusus arsitektur CDB	N/A	N/A
Oracle Database 19c	Arsitektur CDB atau non-CDB	Arsitektur non-CDB ke CDB (April 2021 RU atau yang lebih tinggi)	21c CDB
Oracle Database 12c (dihentikan)	Khusus arsitektur non-CDB	N/A	19c non-CDB

Seperti yang ditunjukkan pada tabel sebelumnya, Anda tidak dapat langsung meningkatkan non-CDB ke CDB dalam versi basis data utama yang baru. Akan tetapi, Anda dapat mengonversi Oracle Database 19c non-CDB ke Oracle Database 19c CDB, lalu meningkatkan Oracle Database 19c CDB ke Oracle Database 21c CDB. Untuk informasi selengkapnya, lihat Mengonversi non-CDB RDS for Oracle ke CDB.

Opsi konversi untuk konfigurasi arsitektur CDB

Tabel berikut menunjukkan opsi yang berbeda untuk mengonversi konfigurasi arsitektur instans DB RDS for Oracle.

Arsitektur dan konfigurasi saat ini	Konversi ke konfigurasi satu penghuni pada arsitektur CDB	Konversi ke konfigurasi multi-penghuni pada arsitektur CDB	Konversi ke arsitektur non-CDB
Non-CDB	Didukung	Didukung*	N/A
CDB menggunakan konfigurasi satu penghuni	N/A	Didukung	Tidak Support
CDB menggunakan konfigurasi multi-penghuni	Tidak didukung	N/A	Tidak didukung

* Anda tidak dapat mengonversi non-CDB ke konfigurasi multi-penghuni dalam satu operasi. Saat Anda mengonversi non-CDB ke CDB, CDB berada dalam konfigurasi satu penghuni. Anda kemudian dapat mengonversi satu penghuni menjadi konfigurasi multi-penghuni dalam operasi terpisah.

Akun pengguna dan hak istimewa dalam CDB

Dalam arsitektur multipenghuni Oracle, semua akun pengguna baik pengguna umum atau pengguna lokal. Pengguna umum CDB adalah pengguna basis data yang identitas tunggal dan kata sandinya dikenal di root CDB dan di setiap PDB yang ada saat ini dan ke depannya. Sebaliknya, pengguna lokal hanya ada dalam satu PDB.

Pengguna master RDS adalah akun pengguna lokal di PDB, yang Anda beri nama saat Anda membuat instans DB Anda. Jika Anda membuat akun pengguna baru, pengguna ini juga akan menjadi pengguna lokal yang berada di PDB. Anda tidak dapat menggunakan akun pengguna untuk membuat PDB baru atau mengubah status PDB yang ada.

Pengguna `rdsadmin` adalah akun pengguna umum. Anda dapat menjalankan paket RDS for Oracle yang ada di akun ini, tetapi Anda tidak dapat masuk sebagai `rdsadmin`. Untuk informasi selengkapnya, lihat About Common Users and Local Users dalam dokumentasi Oracle.

Kelompok grup parameter dalam CDB

CDB memiliki kelompok grup parameternya sendiri dan nilai parameter default. Kelompok grup parameter CDB adalah sebagai berikut:

- oracle-ee-cdb-21
- oracle-se2-cdb-21
- oracle-ee-cdb-19
- oracle-se2-cdb-19

Batasan CDB RDS for Oracle

RDS for Oracle mendukung subset fitur yang tersedia dalam CDB on-premise.

Batasan CDB

Batasan berikut berlaku untuk CDB RDS for Oracle:

- Anda tidak dapat terhubung ke CDB. Anda selalu terhubung ke basis data penghuni (PDB), bukan CDB. Menentukan titik akhir untuk PDB, sama seperti halnya dengan non-CDB. Satu-satunya perbedaan adalah Anda menentukan pdb_name untuk nama basis data, di mana pdb_name adalah nama yang Anda pilih untuk PDB Anda.
- Anda tidak dapat mengonversi CDB dalam konfigurasi multi-penghuni ke CDB dalam konversi satu penghuni. Konversi ke konfigurasi multi-penghuni bersifat satu arah dan tidak dapat diubah.
- Anda tidak dapat mengaktifkan atau mengonversi ke konfigurasi multi-penghuni jika instans DB Anda menggunakan basis data Oracle rilisan yang lebih rendah dari 19.0.0.0.ru-2022-01.rur-2022.r1.
- Anda tidak dapat menggunakan CBD RDS for Oracle dengan ORDS v22 dan versi yang lebih baru. Sebagai solusinya, Anda dapat menggunakan versi ORDS yang lebih lama atau menggunakan Basis Data Oracle 19c non-CDB.
- Anda tidak dapat menggunakan CBD RDS for Oracle dengan ORDS v22 dan versi yang lebih baru. Sebagai solusinya, Anda dapat menggunakan versi ORDS yang lebih lama atau menggunakan Basis Data Oracle 19c non-CDB.

Dukungan untuk fitur-fitur berikut bergantung pada konfigurasi arsitektur.

Fitur	Didukung dalam satu penghuni	Didukung dalam multi-penghuni
Oracle Data Guard	Ya	Tidak
Keamanan Label Oracle	Tidak	Tidak
Oracle Enterprise Manager (OEM)	Tidak	Tidak
OEM Agent	Tidak	Tidak
Stream Aktivitas Basis Data	Ya	Tidak

Batasan basis data penghuni (PDB)

Batasan berikut berlaku untuk basis data penghuni dalam konfigurasi multi-penghuni RDS for Oracle:

- Anda tidak dapat menunda operasi basis data penyewa ke jendela pemeliharaan. Semua perubahan langsung terjadi.
- Anda tidak dapat menambahkan basis data penghuni ke CDB yang menggunakan konfigurasi satu penghuni.
- Anda tidak dapat menambahkan atau memodifikasi beberapa basis data penghuni dalam satu operasi. Anda hanya dapat menambahkan atau memodifikasinya satu per satu.
- Anda tidak dapat memodifikasi basis data penghuni untuk diberi nama `CDB$ROOT` atau `PDB$SEED`.
- Anda tidak dapat menghapus basis data penghuni jika itu adalah satu-satunya penghuni di CDB.
- Tidak semua jenis kelas instans DB memiliki sumber daya yang cukup untuk mendukung beberapa PDB dalam instans CDB RDS for Oracle. Jumlah PDB yang meningkat memengaruhi performa dan stabilitas kelas instans yang lebih kecil dan meningkatkan waktu sebagian besar operasi tingkat instans, misalnya, peningkatan basis data.
- Anda tidak dapat menggunakan beberapa Akun AWS untuk membuat PDB di CDB yang sama. PDB harus dimiliki oleh akun yang sama dengan instans DB yang menjadi host PDB.
- Semua PDB dalam CDB menggunakan titik akhir dan pendengar basis data yang sama.
- Operasi berikut tidak didukung di tingkat PDB, tetapi didukung di tingkat CDB:
  - Pencadangan dan pemulihan

  - Peningkatan basis data
  - Tindakan pemeliharaan
- Fitur-fitur berikut tidak didukung di tingkat PDB, tetapi didukung di tingkat CDB:
  - Wawasan Performa
  - Grup opsi (opsi diinstal di semua PDB pada instans CDB Anda)
  - Grup parameter (semua parameter berasal dari grup parameter yang terkait dengan instans CDB Anda)
- Operasi tingkat PDB yang didukung dalam arsitektur CDB on-premise tetapi tidak didukung dalam CDB RDS for Oracle mencakup:

> **Note**
>
> Daftar berikut tidak lengkap.

  - PDB Aplikasi
  - PDB Proksi
  - Memulai dan menghentikan PDB
  - Mencabut dan memasukkan PDB

    Untuk memindahkan data ke dalam atau keluar dari CDB Anda, gunakan teknik yang sama seperti untuk non-CDB. Untuk informasi selengkapnya tentang memigrasikan data, lihat Mengimpor data ke Oracle di Amazon RDS.
  - Opsi pengaturan di tingkat PDB

    PDB mewarisi pengaturan opsi dari grup opsi CDB. Untuk informasi selengkapnya tentang opsi ini, lihat Bekerja dengan grup parameter. Untuk praktik terbaik, lihat Menggunakan grup parameter DB.
  - Mengonfigurasi parameter dalam PDB

    PDB mewarisi pengaturan parameter dari CDB. Untuk informasi selengkapnya tentang opsi ini, lihat Menambahkan opsi untuk instans DB Oracle.
  - Menkonfigurasi pendengar yang berbeda untuk PDB dalam CDB yang sama
  - Fitur Oracle Flashback
  - Mengaudit informasi dari dalam PDB

Mengonfigurasi CDB RDS for Oracle

Konfigurasi CDB hampir sama dengan non-CDB.

Topik

- Membuat instans CBD RDS for Oracle
- Menghubungkan ke PDB di CDB RDS for Oracle Anda

Membuat instans CBD RDS for Oracle

Dalam RDS for Oracle, pembuatan CDB hampir sama dengan non-CDB. Perbedaannya adalah Anda memilih arsitektur multi-penghungi Oracle ketika membuat instans DB dan juga memilih konfigurasi arsitektur: multi-penghuni atau satu penghuni. Jika Anda membuat tag saat membuat CDB dalam konfigurasi multi-penghuni, RDS menyebarkan tag ke basis data penghuni awal. Untuk membuat CDB, gunakan AWS Management Console, AWS CLI, atau RDS API.

Konsol

Untuk membuat instans CDB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di sudut kanan atas konsol Amazon RDS, pilih Wilayah AWS yang akan digunakan untuk membuat instans CDB.
3. Di panel navigasi, pilih Basis Data.
4. Pilih Buat basis data.
5. Di Pilih metode pembuatan basis data, pilih Pembuatan Standar.
6. Di Opsi mesin, pilih Oracle.
7. Untuk Jenis manajemen basis data, pilih Amazon RDS.
8. Untuk Pengaturan arsitektur, pilih Arsitektur multi-penghuni Oracle.
9. Untuk Konfigurasi arsitektur, lakukan salah satu cara berikut:
   - Pilih Konfigurasi multi-penghuni dan lanjutkan ke langkah berikutnya.
   - Pilih Konfigurasi satu penghuni dan langsung ke Langkah 11.
10. (Konfigurasi multi-penghuni) Untuk Pengaturan basis data penghuni, buat perubahan berikut:

- Untuk Nama basis data penghuni, masukkan nama PDB awal Anda. Nama PDB harus berbeda dari nama CDB, yang defaultnya `RDSCDB`.
- Untuk Nama pengguna utama basis data penghuni, masukkan nama pengguna utama PDB Anda. Anda tidak dapat menggunakan nama pengguna utama basis data penghuni untuk masuk ke CDB itu sendiri.
- Masukkan kata sandi di Kata sandi utama basis data penghuni atau pilih Buat kata sandi secara otomatis.
- Untuk Set karakter basis data penghuni, pilih set karakter untuk PDB. Anda dapat memilih set karakter basis data penghuni yang berbeda dari set karakter CDB.

  Set karakter PDB defaultnya adalah AL32UTF8. Jika Anda memilih set karakter PDB nondefault, pembuatan CDB mungkin lebih lambat.

> **Note**
>
> Anda tidak dapat membuat beberapa basis data penghuni sebagai bagian dari proses pembuatan CDB. Anda hanya dapat menambahkan PDB ke CDB yang sudah ada.

11. (Konfigurasi satu penghuni) Pilih pengaturan yang Anda inginkan berdasarkan opsi yang tercantum di Pengaturan untuk instans DB. Perhatikan hal berikut:
    - Untuk Nama pengguna utama, masukkan nama pengguna lokal di PDB Anda. Anda tidak dapat menggunakan nama pengguna utama untuk masuk ke root CDB.
    - Untuk Nama basis data awal, masukkan nama PDB Anda. Anda tidak dapat memberi nama CDB yang memiliki nama default `RDSCDB`.
12. Pilih Buat basis data.

AWS CLI

Untuk membuat CDB dalam konfigurasi multi-tenant, gunakan create-db-instanceperintah dengan parameter berikut:

- `--db-instance-identifier`
- `--db-instance-class`
- `--engine { oracle-ee-cdb | oracle-se2-cdb }`

- `--master-username`
- `--master-user-password`
- `--multi-tenant` (untuk konfigurasi satu penghuni, jangan tentukan `multi-tenant` atau `--no-multi-tenant`)
- `--allocated-storage`
- `--backup-retention-period`

Lihat informasi tentang setiap setelan di Pengaturan untuk instans DB.

Contoh berikut ini menciptakan RDS untuk Oracle DB instance bernama *`my-cdb-inst`*dalam konfigurasi multi-tenant. Jika Anda menentukan `--no-multi-tenant` atau tidak menentukan`--multi-tenant`, CDB akan secara default menggunakan konfigurasi satu penghuni. Mesinnya adalah `oracle-ee-cdb`: perintah yang menentukan `oracle-ee` dan `--multi-tenant` gagal dengan kesalahan. Basis data penghuni awal bernama *`mypdb`*.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-instance \
    --engine oracle-ee-cdb \
    --db-instance-identifier my-cdb-inst \
    --multi-tenant \
    --db-name mypdb \
    --allocated-storage 250 \
    --db-instance-class db.t3.large \
    --master-username pdb_admin \
    --master-user-password pdb_admin_password \
    --backup-retention-period 3
```

Untuk Windows:

```
aws rds create-db-instance ^
    --engine oracle-ee-cdb ^
    --db-instance-identifier my-cdb-inst ^
    --multi-tenant ^
    --db-name mypdb ^
    --allocated-storage 250 ^
    --db-instance-class db.t3.large ^
    --master-username pdb_admin ^
```

```
    --master-user-password pdb_admin_password ^
    --backup-retention-period 3
```

> **Note**
>
> Tentukan kata sandi selain prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

Perintah ini menghasilkan output seperti berikut. Nama basis data, set karakter, set karakter nasional, dan pengguna utama tidak disertakan dalam output. Anda dapat melihat informasi ini menggunakan perintah CLI `describe-tenant-databases`.

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "my-cdb-inst",
        "DBInstanceClass": "db.t3.large",
        "MultiTenant": true,
        "Engine": "oracle-ee-cdb",
        "DBResourceId": "db-ABCDEFGJIJKLMNOPQRSTUVWXYZ",
        "DBInstanceStatus": "creating",
        "AllocatedStorage": 250,
        "PreferredBackupWindow": "04:59-05:29",
        "BackupRetentionPeriod": 3,
        "DBSecurityGroups": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-0a1bcd2e",
                "Status": "active"
            }
        ],
        "DBParameterGroups": [
            {
                "DBParameterGroupName": "default.oracle-ee-cdb-19",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "DBSubnetGroup": {
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default",
            "VpcId": "vpc-1234567a",
            "SubnetGroupStatus": "Complete",
            ...
```

RDS API

Untuk membuat instans DB menggunakan Amazon RDS API, panggil operasi createDBInstance.

Untuk informasi tentang setiap pengaturan, lihat Pengaturan untuk instans DB.

Menghubungkan ke PDB di CDB RDS for Oracle Anda

Anda dapat menggunakan utilitas seperti SQL*Plus untuk terhubung ke PDB. Untuk mengunduh Oracle Instant Client, yang mencakup versi mandiri SQL*Plus, lihat Unduhan Oracle Instant Client.

Untuk menghubungkan SQL*Plus ke PDB Anda, diperlukan informasi berikut:

- Nama PDB
- Nama pengguna dan kata sandi basis data
- Titik akhir untuk instans DB Anda
- Nomor port

Untuk informasi tentang penemuan informasi tersebut, lihat Menemukan titik akhir instans DB RDS for Oracle.

Example Untuk terhubung ke PDB Anda menggunakan SQL*Plus

Dalam contoh berikut, ganti pengguna utama Anda untuk *master_user_name*. Selain itu, ganti titik akhir instans DB Anda, lalu sertakan nomor port dan SID Oracle. Nilai SID adalah nama PDB yang Anda tentukan saat membuat instans DB, bukan pengidentifikasi instans DB.

Untuk Linux, macOS, atau Unix:

```
sqlplus 'master_user_name@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=endpoint)
(PORT=port))(CONNECT_DATA=(SID=pdb_name)))'
```

Untuk Windows:

```
sqlplus master_user_name@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=endpoint)
(PORT=port))(CONNECT_DATA=(SID=pdb_name)))
```

Output Anda akan terlihat seperti berikut ini.

```
SQL*Plus: Release 19.0.0.0.0 Production on Mon Aug 21 09:42:20 2021
```

Perintah SQL akan muncul setelah Anda memasukkan kata sandi pengguna.

```
SQL>
```

> **Note**
>
> String koneksi format yang lebih pendek (Easy connect atau EZCONNECT), seperti `sqlplus username/password@LONGER-THAN-63-CHARS-RDS-ENDPOINT-HERE:1521/database-identifier`, mungkin mencapai batas karakter maksimum dan tidak boleh digunakan untuk menghubungkan.

Mencadangkan dan memulihkan CDB

Anda dapat mencadangkan dan memulihkan CDB Anda menggunakan snapshot DB RDS atau Recovery Manager (RMAN).

Mencadangkan dan memulihkan CDB menggunakan snapshot DB

Snapshot DB bekerja dengan cara yang sama dalam arsitektur CDB dan non-CDB. Berikut ini perbedaan utamanya:

- Saat mengembalikan snapshot DB dari CDB, Anda tidak dapat mengganti nama CDB. CDB bernama `RDSCDB` dan tidak dapat diubah.
- Saat mengembalikan snapshot DB dari CDB, Anda tidak dapat mengganti nama PDB. Anda dapat mengubah nama PDB menggunakan perintah modify-tenant-database.
- Untuk menemukan basis data penghuni penyewa dalam snapshot, gunakan perintah CLI describe-db-snapshot-tenant-databases.
- Anda tidak dapat langsung berinteraksi dengan basis data penghuni dalam snapshot CDB yang menggunakan konfigurasi arsitektur multi-penghuni. Jika Anda memulihkan snapshot DB, Anda memulihkan semua basis data penghuni.
- RDS for Oracle secara implisit menyalin tag pada basis data penghuni ke basis data penghuni dalam snapshot DB. Saat Anda memulihkan basis data penghuni, tag muncul di basis data yang dipulihkan.
- Jika Anda mengembalikan snapshot DB dan menentukan tag baru menggunakan parameter `--tags`, tag baru menimpa semua tag yang ada.

- Jika Anda mengambil snapshot DB dari instans CDB yang memiliki tag, dan Anda menentukan `--copy-tags-to-snapshot`, RDS for Oracle menyalin tag dari basis data penghuni ke basis data penghuni dalam snapshot.

Untuk informasi selengkapnya, lihat Pertimbangan Oracle Database.

Mencadangkan dan memulihkan CDB menggunakan RMAN

Untuk mempelajari cara mencadangkan dan memulihkan CDB atau basis data penghuni individu menggunakan RMAN, lihat Melakukan tugas RMAN umum untuk instans DB Oracle.

Mengonversi non-CDB RDS for Oracle ke CDB

Anda dapat mengubah arsitektur database Oracle dari arsitektur non-CDB ke arsitektur multitenant Oracle, juga disebut arsitektur CDB, dengan perintah. `modify-db-instance` Dalam kebanyakan kasus, teknik ini lebih disukai daripada membuat CDB baru dan mengimpor data. Operasi konversi menimbulkan downtime.

Ketika Anda meningkatkan versi mesin basis data, Anda tidak dapat mengubah arsitektur basis data dalam operasi yang sama. Oleh karena itu, untuk meningkatkan non-CDB Oracle Database 19c ke CBD Oracle Database 21c, pertama-tama Anda perlu mengonversi non-CDB menjadi CDB dalam satu langkah, lalu meningkatkan CDB 19c ke CDB 21c dalam langkah terpisah.

Operasi konversi non-CDB memiliki persyaratan sebagai berikut:

- Anda harus menentukan `oracle-ee-cdb` atau `oracle-se2-cdb` untuk jenis mesin DB. Hanya nilai-nilai tersebut yang didukung.
- Mesin DB Anda harus menggunakan Oracle Database 19c dengan pembaruan rilis (RU) April 2021 atau yang lebih baru.

Operasi ini memiliki batasan sebagai berikut:

- Anda tidak dapat mengonversi CDB ke non-CDB. Anda tidak dapat mengonversi non-CDB ke CDB.
- Anda tidak dapat mengonversi non-CDB ke konfigurasi multi-penghuni dalam satu panggilan `modify-db-instance`. Setelah Anda mengonversi non-CDB ke CDB, CDB Anda berada dalam konfigurasi satu penghuni. Untuk mengonversi konfigurasi satu penghuni ke konfigurasi multi-

penghuni, jalankan `modify-db-instance` lagi. Untuk informasi selengkapnya, lihat Mengonversi konfigurasi satu penghuni menjadi multi-penghuni.

- Basis data primer atau replika yang mengaktifkan Oracle Data Guard tidak dapat dikonversi. Untuk mengonversi non-CDB yang memiliki replika baca, hapus terlebih dahulu semua replika baca.
- Anda tidak dapat meningkatkan versi mesin DB dan mengonversi non-CBD ke CBD dalam operasi yang sama.
- Pertimbangan untuk grup opsi dan parameter sama seperti ketika meningkatkan mesin DB. Untuk informasi selengkapnya, lihat Pertimbangan untuk upgrade DB Oracle.

Konsol

Cara mengonversi non-CDB ke CDB

1. Masuk ke AWS Management Console, lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di sudut kanan atas konsol Amazon RDS, pilih Wilayah AWS tempat instans DB Anda berada.
3. Di panel navigasi, pilih Basis data, lalu pilih instans non-CDB yang ingin Anda konversi menjadi instans CDB.
4. Pilih Ubah.
5. Untuk Pengaturan arsitektur, pilih Arsitektur multipenghuni Oracle. Setelah konversi, CDB Anda akan berada dalam konfigurasi satu penghuni.
6. (Opsional) Untuk Grup parameter DB, pilih grup parameter baru untuk instans CDB Anda. Pertimbangan grup parameter yang berlaku saat mengonversi instans DB sama seperti saat meningkatkan instans DB. Untuk informasi selengkapnya, lihat Pertimbangan grup parameter.
7. (Opsional) Untuk Grup opsi, pilih grup opsi baru untuk instans CDB Anda. Pertimbangan grup opsi yang berlaku saat mengonversi instans DB sama seperti saat meningkatkan instans DB. Untuk informasi selengkapnya, lihat Pertimbangan grup opsi.
8. Jika semua perubahan sudah sesuai dengan keinginan Anda, pilih Lanjutkan dan periksa ringkasan modifikasi.
9. (Opsional) Pilih Terapkan seketika untuk menerapkan perubahan dengan serta-merta. Memilih opsi ini dapat menyebabkan waktu henti dalam beberapa kasus. Untuk informasi selengkapnya, lihat Menggunakan pengaturan Terapkan Segera.
10. Di halaman konfirmasi, tinjau perubahan Anda. Jika sudah benar, pilih Modifikasi instans DB.

    Atau pilih Kembali untuk mengedit perubahan atau Batal untuk membatalkan perubahan.

AWS CLI

Untuk mengonversi non-CDB pada instans DB Anda ke CDB dalam konfigurasi penyewa tunggal, setel `--engine` ke `oracle-ee-cdb` atau dalam perintah. `oracle-se2-cdb` AWS CLI modify-db-instance Untuk informasi selengkapnya, lihat Pengaturan untuk instans DB.

Contoh berikut mengkonversi contoh DB bernama *my-non-cdb*dan menentukan kelompok pilihan kustom dan kelompok parameter.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-non-cdb \
    --engine oracle-ee-cdb \
    --option-group-name custom-option-group \
    --db-parameter-group-name custom-parameter-group
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-non-cdb ^
    --engine oracle-ee-cdb ^
    --option-group-name custom-option-group ^
    --db-parameter-group-name custom-parameter-group
```

API RDS

Untuk mengonversi non-CDB ke CDB, tentukan `Engine` dalam operasi RDS API ModifyDBInstance.

Mengonversi konfigurasi satu penghuni menjadi multi-penghuni

Anda dapat memodifikasi arsitektur CDB RDS for Oracle dari konfigurasi satu penghuni menjadi multi-penghuni. Sebelum dan sesudah konversi, CDB Anda berisi basis data satu penghuni (PDB).

Selama konversi, RDS for Oracle memigrasikan metadata berikut ke basis data penghuni baru:

- Nama pengguna utama
- Nama basis data
- Set karakter

- Set karakter nasional

Sebelum konversi, Anda dapat melihat informasi sebelumnya menggunakan perintah `describe-db-instances`. Setelah konversi, Anda melihat informasi menggunakan perintah `describe-tenant-database`.

Konversi memiliki persyaratan dan batasan sebagai berikut:

- Setelah mengonversi konfigurasi arsitektur satu penghuni menjadi multi-penghuni, Anda tidak dapat mengembalikan arsitektur ke konfigurasi satu penghuni. Operasi ini tidak dapat dibatalkan.
- Tag untuk instans DB disebarkan ke DB penghuni awal yang dibuat selama konversi.
- Basis data primer atau replika yang mengaktifkan Oracle Data Guard tidak dapat dikonversi.
- Anda tidak dapat meningkatkan versi mesin DB dan melakukan konversi ke konfigurasi multi-penghuni dalam operasi yang sama.
- Kebijakan IAM Anda harus memiliki izin untuk membuat basis data penghuni.

Konsol

Untuk mengonversi CDB yang menggunakan konfigurasi satu penghuni ke mulit-penghuni

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di sudut kanan atas konsol Amazon RDS, pilih Wilayah AWS tempat instans DB Anda berada.
3. Di panel navigasi, pilih Basis data, lalu pilih instans non-CDB yang ingin Anda konversi menjadi instans CDB.
4. Pilih Ubah.
5. Untuk Pengaturan arsitektur, pilih Arsitektur multi-penghuni Oracle.
6. Untuk Konfigurasi arsitektur, pilih Konfigurasi multi-penghuni.
7. (Opsional) Untuk Grup parameter DB, pilih grup parameter baru untuk instans CDB Anda. Pertimbangan grup parameter yang berlaku saat mengonversi instans DB sama seperti saat meningkatkan instans DB.
8. (Opsional) Untuk Grup opsi, pilih grup opsi baru untuk instans CDB Anda. Pertimbangan grup opsi yang berlaku saat mengonversi instans DB sama seperti saat meningkatkan instans DB.
9. Jika semua perubahan sudah sesuai dengan keinginan Anda, pilih Lanjutkan dan periksa ringkasan modifikasi.

10. Pilih Terapkan langsung. Opsi ini diperlukan saat Anda beralih ke konfigurasi multi-penghuni. Perlu diketahui bahwa opsi ini dapat menyebabkan waktu henti dalam beberapa kasus.
11. Di halaman konfirmasi, tinjau perubahan Anda. Jika sudah benar, pilih Modifikasi instans DB.

    Atau pilih Kembali untuk mengedit perubahan atau Batal untuk membatalkan perubahan.

AWS CLI

Untuk mengonversi CDB menggunakan konfigurasi penyewa tunggal ke konfigurasi multi-penyewa, tentukan dalam perintah. `--multi-tenant` AWS CLI modify-db-instance

Contoh berikut mengonversi instans DB bernama `my-st-cdb` dari konfigurasi satu penghuni ke multi-penghuni. Opsi `--apply-immediately` diperlukan.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance --region us-east-1\
    --db-instance-identifier my-st-cdb \
    --multi-tenant \
    --apply-immediately
```

Untuk Windows:

```
aws rds modify-db-instance --region us-east-1 ^
    --db-instance-identifier my-st-cdb ^
    --multi-tenant ^
    --apply-immediately
```

Output-nya akan terlihat seperti berikut.

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "my-st-cdb",
        "DBInstanceClass": "db.r5.large",
        "MultiTenant": false,
        "Engine": "oracle-ee-cdb",
        "DBResourceId": "db-AB1CDE2FGHIJK34LMNOPRLXTXU",
        "DBInstanceStatus": "modifying",
        "MasterUsername": "admin",
```

```
        "DBName": "ORCL",
        ...
        "EngineVersion": "19.0.0.0.ru-2022-01.rur-2022-01.r1",
        "AutoMinorVersionUpgrade": true,
        "ReadReplicaDBInstanceIdentifiers": [],
        "LicenseModel": "bring-your-own-license",
        "OptionGroupMemberships": [
            {
                "OptionGroupName": "default:oracle-ee-cdb-19",
                "Status": "in-sync"
            }
        ],
        ...
        "PendingModifiedValues": {
            "MultiTenant": "true"
        }
    }
}
```

Menambahkan basis data RDS for Oracle ke instans CDB Anda

Dalam konfigurasi multi-penghuni RDS for Oracle, basis data penghuni adalah PDB. Untuk menambahkan basis data penyewa, pastikan Anda memenuhi prasyarat berikut:

- CDB Anda mengaktifkan konfigurasi multi-penghuni. Untuk informasi selengkapnya, lihat Konfigurasi multi-penghuni pada arsitektur CDB.
- Anda memiliki izin IAM yang diperlukan untuk membuat basis data penghuni.

Anda dapat menambahkan basis data penghuni menggunakan AWS Management Console, AWS CLI, atau RDS API. Anda tidak dapat menambahkan beberapa basis data penghuni dalam satu operasi: Anda harus menambahkannya satu per satu. Jika retensi cadangan diaktifkan di CDB, Amazon RDS mencadangkan instans DB sebelum dan sesudah basis data penghuni baru ditambahkan.

Konsol

Untuk menambahkan basis data penghuni ke instans DB Anda

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.

2. Di sudut kanan atas konsol Amazon RDS, pilih Wilayah AWS untuk membuat basis data penghuni.
3. Di panel navigasi, pilih Basis Data.
4. Pilih instans CDB tempat basis data penghuni akan ditambahkan. Instans DB Anda harus menggunakan konfigurasi multi-penghuni arsitektur CDB.
5. Pilih Tindakan lalu Tambahkan basis data penghuni.
6. Untuk Pengaturan basis data penghuni, lakukan hal berikut:

   - Untuk Nama basis data penghuni, masukkan nama PDB baru Anda.
   - Untuk Nama pengguna master basis data penghuni, masukkan nama pengguna master untuk PDB Anda. Pengguna master ini berbeda dari pengguna master CDB.
   - Masukkan kata sandi di Kata sandi master basis data penghuni atau pilih Buat kata sandi secara otomatis.
   - Untuk Set karakter basis data penghuni, pilih set karakter untuk PDB. Default-nya adalah AL32UTF8. Anda dapat memilih set karakter PDB yang berbeda dari set karakter CDB.
   - Untuk Set karakter nasional basis data penghuni, pilih set karakter nasional untuk PDB. Default-nya adalah AL32UTF8. Set karakter nasional menentukan pengodean hanya untuk kolom yang menggunakan jenis data `NCHAR` (`NCHAR`, `NVARCHAR2`, dan `NCLOB`) serta tidak memengaruhi metadata basis data.

   Untuk informasi selengkapnya tentang pengaturan sebelumnya, lihat Pengaturan untuk instans DB.
7. Pilih Tambahkan penghuni.

AWS CLI

Untuk menambahkan database penyewa ke CDB Anda denganAWS CLI, gunakan perintah create-tenant-databasedengan parameter yang diperlukan berikut:

- `--db-instance-identifier`
- `--tenant-db-name`
- `--master-username`
- `--master-user-password`

Contoh berikut ini menciptakan database penyewa bernama *mypdb2* di RDS untuk contoh Oracle CDB bernama. *my-cdb-inst* Set karakter PDB adalah UTF-16.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds create-tenant-database --region us-east-1 \
    --db-instance-identifier my-cdb-inst \
    --tenant-db-name mypdb2 \
    --master-username mypdb2-admin \
    --master-user-password mypdb2-pwd \
    --character-set-name UTF-16
```

Untuk Windows:

```
aws rds create-tenant-database --region us-east-1 ^
    --db-instance-identifier my-cdb-inst ^
    --tenant-db-name mypdb2 ^
    --master-username mypdb2-admin ^
    --master-user-password mypdb2-pwd ^
    --character-set-name UTF-16
```

Output-nya akan terlihat serupa dengan yang berikut ini.

```
...}
    "TenantDatabase" :
         {
            "DbiResourceId" : "db-abc123",
            "TenantDatabaseResourceId" : "tdb-bac567",
            "TenantDatabaseArn" : "arn:aws:rds:us-east-1:123456789012:db:my-cdb-
inst:mypdb2",
            "DBInstanceIdentifier" : "my-cdb-inst",
            "TenantDBName" : "mypdb2",
            "Status" : "creating",
            "MasterUsername" : "mypdb2",
            "CharacterSetName" : "UTF-16",
            ...
        }
}...
```

Memodifikasi RDS untuk basis data penghuni Oracle

Anda hanya dapat memodifikasi nama PDB dan kata sandi pengguna master dari basis data penghuni di CDB Anda. Perhatikan persyaratan dan batasan berikut:

- Untuk memodifikasi pengaturan basis data penghuni di instans DB Anda, basis data penghuni harus ada.
- Anda tidak dapat memodifikasi beberapa basis data penghuni dalam satu operasi. Anda hanya dapat memodifikasi satu basis data penghuni dalam satu waktu.
- Anda tidak dapat mengubah nama basis data penghuni menjadi `CDB$ROOT` atau `PDB$SEED`.

Anda dapat memodifikasi PDB menggunakan AWS Management Console, AWS CLI, atau RDS API.

Konsol

Untuk memodifikasi nama PDB atau kata sandi master dari basis data penghuni

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di sudut kanan atas konsol Amazon RDS, pilih Wilayah AWS untuk membuat basis data penghuni.
3. Di panel navigasi, pilih Basis Data.
4. Pilih basis data penghuni yang nama basis datanya atau kata sandi pengguna masternya ingin Anda modifikasi.
5. Pilih Ubah.
6. Untuk Pengaturan basis data penghuni, lakukan salah satu hal berikut:
   - Untuk Nama basis data penghuni, masukkan nama baru untuk PDB baru Anda.
   - Untuk Kata sandi master basis data penghuni, masukkan kata sandi baru.
7. Pilih Modifikasi penghuni.

AWS CLI

Untuk memodifikasi database penyewa menggunakanAWS CLI, panggil modify-tenant-databaseperintah dengan parameter berikut:

- `--db-instance-identifier` *`value`*

- `--tenant-db-name value`
- `[--new-tenant-db-name value]`
- `[--master-user-password value]`

Contoh berikut mengganti nama basis data penghuni `pdb1` menjadi `pdb-hr` pada instans DB `my-cdb-inst`.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds modify-tenant-database --region us-east-1 \
    --db-instance-identifier my-cdb-inst \
    --tenant-db-name pdb1 \
    --new-tenant-db-name pdb-hr
```

Untuk Windows:

```
aws rds modify-tenant-database --region us-east-1 ^
    --db-instance-identifier my-cdb-inst ^
    --tenant-db-name pdb1 ^
    --new-tenant-db-name pdb-hr
```

Perintah ini menghasilkan output seperti berikut.

```
{
    "TenantDatabase" : {
        "DbiResourceId" : "db-abc123",
        "TenantDatabaseResourceId" : "tdb-bac567",
        "TenantDatabaseArn" : "arn:aws:rds:us-east-1:123456789012:db:my-cdb-inst:pdb1",
        "DBInstanceIdentifier" : "my-cdb-inst",
        "TenantDBName" : "pdb1",
        "Status" : "modifying",
        "MasterUsername" : "tenant-admin-user"
        "Port" : "6555",
        "CharacterSetName" : "UTF-16",
        "MaxAllocatedStorage" : "1000",
        "ParameterGroups": [
            {
                "ParameterGroupName": "pdb1-params",
```

```
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "OptionGroupMemberships": [
            {
                "OptionGroupName": "pdb1-options",
                "Status": "in-sync"
            }
        ],
        "PendingModifiedValues": {
            "TenantDBName": "pdb-hr"
        }
    }
}
```

Menghapus basis data penghuni RDS for Oracle dari CDB Anda

Anda dapat menghapus basis data penghuni (PDB) menggunakan AWS Management Console, AWS CLI, atau RDS API. Pertimbangkan prasyarat dan batasan berikut:

- Basis data penghuni dan instans DB harus ada.
- Agar penghapusan berhasil, salah satu situasi berikut harus ada:
  - Basis data penghuni dan instans DB tersedia.

    > **Note**
    >
    > Anda dapat mengambil snapshot final, tetapi hanya jika basis data penghuni dan instans DB berada dalam status tersedia sebelum Anda mengeluarkan perintah `delete-tenant-database`.

  - Basis data penghuni sedang dibuat.
  - Instans DB memodifikasi basis data penghuni.
- Anda tidak dapat menghapus beberapa basis data penghuni dalam satu operasi.
- Anda tidak dapat menghapus basis data penghuni jika itu adalah satu-satunya penghuni di CDB.

Konsol

Untuk menghapus basis data penghuni

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data, lalu pilih basis data penghuni yang akan dihapus.
3. Untuk Tindakan, pilih Hapus.
4. Untuk membuat snapshot DB akhir untuk instans DB, pilih Buat snapshot akhir?.
5. Jika Anda memilih untuk membuat snapshot akhir, masukkan Nama snapshot akhir.
6. Masukkan **`delete me`** di kotak.
7. Pilih Hapus.

AWS CLI

Untuk menghapus database penyewa menggunakanAWS CLI, panggil delete-tenant-databaseperintah dengan parameter berikut:

- `--db-instance-identifier` *`value`*
- `--tenant-db-name` *`value`*
- `[--skip-final-snapshot | --no-skip-final-snapshot]`
- `[--final-snapshot-identifier` *`value`*`]`

Contoh berikut ini menghapus database penyewa bernama *`pdb-test`* dari CDB bernama. *`my-cdb-inst`* Secara default, operasi ini membuat snapshot akhir.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds delete-tenant-database --region us-east-1 \
    --db-instance-identifier my-cdb-inst \
    --tenant-db-name pdb-test \
    --final-snapshot-identifier final-snap-pdb-test
```

Untuk Windows:

```
aws rds delete-tenant-database --region us-east-1 ^
    --db-instance-identifier my-cdb-inst ^
    --tenant-db-name pdb-test ^
    --final-snapshot-identifier final-snap-pdb-test
```

Perintah ini menghasilkan output seperti berikut.

```
{
    "TenantDatabase" : {
        "DbiResourceId" : "db-abc123",
        "TenantDatabaseResourceId" : "tdb-bac456",
        "TenantDatabaseArn" : "arn:aws:rds:us-east-1:123456789012:db:my-cdb-inst:pdb-
test",
        "DBInstanceIdentifier" : "my-cdb-inst",
        "TenantDBName" : "pdb-test",
        "Status" : "deleting",
        "MasterUsername" : "pdb-test-admin"
        "Port" : "6555",
        "CharacterSetName" : "UTF-16",
        "MaxAllocatedStorage" : "1000",
        "ParameterGroups": [
            {
                "ParameterGroupName": "tenant-1-params",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "OptionGroupMemberships": [
            {
                "OptionGroupName": "tenant-1-options",
                "Status": "in-sync"
            }
        ]
    }
}
```

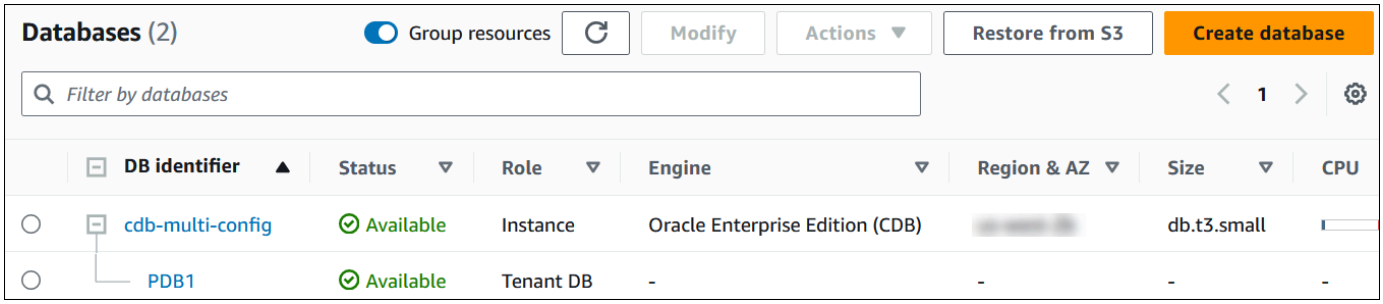
Melihat detail basis data penghuni

Cara untuk melihat detail tentang basis data penghuni sama seperti cara untuk melihat detail non-CDB atau CDB.

Konsol

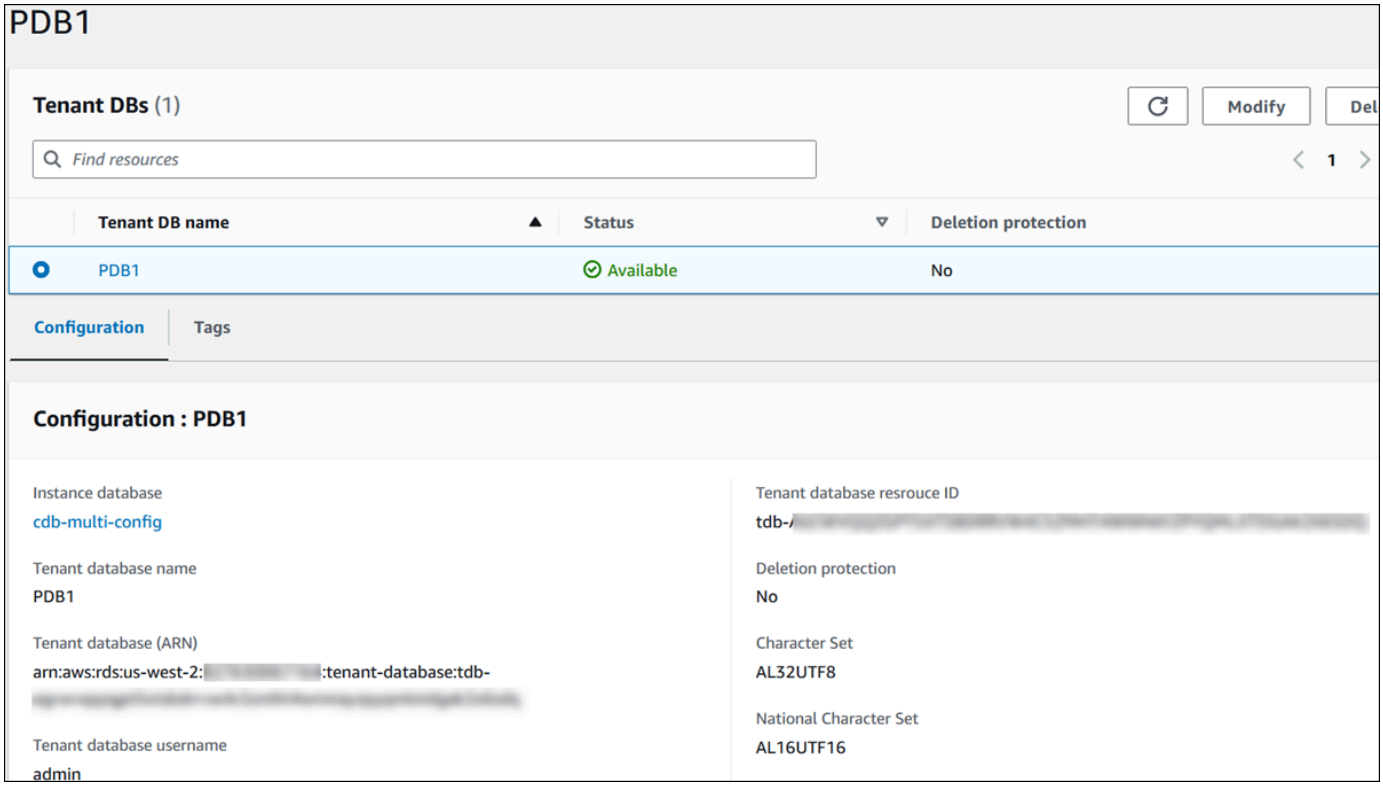
Untuk melihat detail tentang basis data penghuni

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di sudut kanan atas konsol Amazon RDS, pilih Wilayah AWS tempat instans DB Anda berada.
3. Di panel navigasi, pilih Basis Data.



Pada gambar sebelumnya, basis data penghuni tunggal (PDB) muncul sebagai turunan dari instans DB.

4. Pilih nama basis data penghuni.



AWS CLI

Untuk melihat detail tentang PDB Anda, gunakan AWS CLI perintah describe-tenant-databases.

Contoh berikut ini menjelaskan semua basis data penghuni di Wilayah tertentu.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds describe-tenant-databases --region us-east-1
```

Untuk Windows:

```
aws rds describe-tenant-databases --region us-east-1
```

Perintah ini menghasilkan output seperti berikut.

```
    "TenantDatabases" : [
         {
            "DBInstanceIdentifier" : "my-cdb-inst",
            "TenantDBName" : "pdb-test",
            "Status" : "available",
            "MasterUsername" : "pdb-test-admin",
            "DbiResourceId" : "db-abc123",
            "TenantDatabaseResourceId" : "tdb-bac456",
            "TenantDatabaseArn" : "arn:aws:rds:us-east-1:123456789012:db:my-cdb-
inst:pdb-test",
            "CharacterSetName": "AL32UTF8",
            "NcharCharacterSetName": "AL16UTF16",
            "DeletionProtection": false,
            "PendingModifiedValues": {
                 "MasterUserPassword": "****"
            },
            "TagList": []
         },
         {

            "DBInstanceIdentifier" : "my-cdb-inst2",
            "TenantDBName" : "pdb-dev",
            "Status" : "modifying",
            "MasterUsername" : "masterrdsuser"
```

```
            "DbiResourceId" : "db-xyz789",
            "TenantDatabaseResourceId" : "tdb-ghp890",
            "TenantDatabaseArn" : "arn:aws:rds:us-east-1:123456789012:db:my-cdb-
inst2:pdb-dev",
            "CharacterSetName": "AL32UTF8",
            "NcharCharacterSetName": "AL16UTF16",
            "DeletionProtection": false,
            "PendingModifiedValues": {
                 "MasterUserPassword": "****"
            },
            "TagList": []
         },
         ... other truncated data
```

Contoh berikut ini menjelaskan basis data penghuni di instans DB `my-cdb-inst` di Wilayah tertentu.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds describe-tenant-databases --region us-east-1 \
    --db-instance-identifier my-cdb-inst
```

Untuk Windows:

```
aws rds describe-tenant-databases --region us-east-1 ^
    --db-instance-identifier my-cdb-inst
```

Perintah ini menghasilkan output seperti berikut.

```
{
    "TenantDatabase": {
        "TenantDatabaseCreateTime": "2023-10-19T23:55:30.046Z",
        "DBInstanceIdentifier": "my-cdb-inst",
        "TenantDBName": "pdb-hr",
        "Status": "creating",
        "MasterUsername": "tenant-admin-user",
        "DbiResourceId": "db-abc123",
        "TenantDatabaseResourceId": "tdb-bac567",
        "TenantDatabaseARN": "arn:aws:rds:us-west-2:579508833180:pdb-hr:tdb-
abcdefghi1jklmno2p3qrst4uvw5xy6zabc7defghi8jklmn90op",
        "CharacterSetName": "AL32UTF8",
```

```
        "NcharCharacterSetName": "AL16UTF16",
        "DeletionProtection": false,
        "PendingModifiedValues": {
            "MasterUserPassword": "****"
        },
        "TagList": [
            {
                "Key": "TEST",
                "Value": "testValue"
            }
        ]
    }
}
```

Contoh berikut ini menjelaskan basis data penghuni pdb1 di instans DB `my-cdb-inst` di Wilayah AS Timur (Virginia Utara).

Example

Untuk Linux, macOS, atau Unix:

```
aws rds describe-tenant-databases --region us-east-1 \
--db-instance-identifier my-cdb-inst \
--tenant-db-name pdb1
```

Untuk Windows:

```
aws rds describe-tenant-databases --region us-east-1 ^
--db-instance-identifier my-cdb-inst ^
--tenant-db-name pdb1
```

Perintah ini menghasilkan output seperti berikut.

```
{
    "TenantDatabases" : [
        {
            "DbiResourceId" : "db-abc123",
            "TenantDatabaseResourceId" : "tdb-bac567",
            "TenantDatabaseArn" : "arn:aws:rds:us-east-1:123456789012:db:my-cdb-
inst:pdb1"
            "DBInstanceIdentifier" : "my-cdb-inst",
            "TenantDBName" : "pdb1",
```

```
            "Status" : "ACTIVE",
            "MasterUsername" : "masterawsuser"
            "Port" : "1234",
            "CharacterSetName": "UTF-8",
            "ParameterGroups": [
                {
                    "ParameterGroupName": "tenant-custom-pg",
                    "ParameterApplyStatus": "in-sync"
                }
            ],
            {
            "OptionGroupMemberships": [
                {
                    "OptionGroupName": "tenant-custom-og",
                    "Status": "in-sync"
                }
            ]
        }
    ]
}
```

Meningkatkan CDB Anda

Anda dapat meningkatkan CDB ke rilis Oracle Database yang berbeda. Misalnya, Anda dapat meningkatkan CBD Oracle Database 19c ke CBD Oracle Database 21c. Anda tidak dapat mengubah arsitektur basis data selama peningkatan. Dengan demikian, Anda tidak dapat meningkatkan non-CDB ke CDB atau meningkatkan CDB ke non-CDB.

Prosedur untuk meningkatkan CDB ke CDB sama dengan meningkatkan non-CDB ke non-CDB. Untuk informasi selengkapnya, lihat Meng-upgrade mesin DB Oracle.

Mengelola instans DB RDS for Oracle

Berikut ini adalah tugas pengelolaan umum yang Anda lakukan dengan instans DB RDS for Oracle. Beberapa tugas bersifat sama untuk semua instans DB RDS. Beberapa tugas lain bersifat khusus untuk RDS for Oracle.

Tugas-tugas berikut bersifat umum untuk semua basis data RDS, tetapi Basis Data Oracle memiliki pertimbangan khusus. Misalnya, Anda terhubung ke basis data Oracle menggunakan klien Oracle SQL* Plus dan SQL Developer.

Area tugas	Dokumentasi terkait
Kelas instans, penyimpanan, dan PIOPS Jika Anda membuat instans produksi, pelajari cara kerja kelas instans, tipe penyimpanan, dan IOPS yang tersedia di Amazon RDS.	Kelas instans RDS for Oracle Jenis penyimpanan Amazon RDS
Deployment Multi-AZ Instans DB produksi harus menggunakan deployment Multi-AZ. Deployment Multi-AZ memberikan peningkatan ketersediaan, ketahanan data, dan toleransi kesalahan untuk instans DB.	Mengonfigurasi dan mengelola deployment Multi-AZ
Amazon VPC Jika akun AWS Anda memiliki cloud privat virtual (VPC) default, instans DB Anda secara otomatis dibuat di dalam VPC default. Jika akun Anda tidak memiliki VPC default dan Anda ingin instans DB dalam VPC, buat VPC dan grup subnet sebelum Anda membuat instans.	Bekerja dengan klaster DB dalam VPC
Grup keamanan Secara default, instans DB menggunakan firewall yang mencegah akses. Pastikan Anda membuat grup keamanan dengan alamat IP dan konfigurasi jaringan yang benar untuk mengakses instans DB.	Mengontrol akses dengan grup keamanan

Area tugas	Dokumentasi terkait
Grup parameter Jika instans DB Anda akan membutuhkan parameter basis data tertentu, buat grup parameter sebelum Anda membuat instans DB.	Bekerja dengan grup parameter
Grup opsi Jika instans DB Anda membutuhkan opsi basis data tertentu, buat grup opsi sebelum Anda membuat instans DB.	Menambahkan opsi untuk instans DB Oracle
Menghubungkan ke instans DB Setelah membuat grup keamanan dan mengaitkannya ke instans DB, Anda dapat menghubungkan ke instans DB menggunakan aplikasi klien SQL standar seperti Oracle SQL*Plus.	Menghubungkan ke instans RDS for Oracle DB
Pencadangan dan pemulihan Anda dapat mengonfigurasi instans DB Anda untuk mengambil cadangan otomatis, atau mengambil snapshot manual, kemudian memulihkan instans dari cadangan atau snapshot.	Mencadangkan, memulihkan, dan mengekspor data
Pemantauan Anda dapat memantau instans Oracle DB dengan menggunak an metrik CloudWatch Amazon RDS, peristiwa, dan pemantaua n yang disempurnakan.	Melihat metrik di konsol Amazon RDS Melihat peristiwa Amazon RDS
File log Anda dapat mengakses file log untuk instans DB Oracle Anda.	Memantau file log Amazon RDS

Setelah itu, Anda dapat menemukan deskripsi untuk implementasi beberapa tugas DBA umum khusus Amazon RDS untuk RDS Oracle. Untuk memberikan pengalaman layanan terkelola, Amazon RDS tidak memberikan akses shell ke instans DB. Selain itu, RDS membatasi akses ke prosedur

dan tabel sistem tertentu yang membutuhkan hak istimewa tingkat lanjut. Dalam banyak tugas, Anda menjalankan paket `rdsadmin`, yang merupakan alat khusus Amazon RDS yang memungkinkan Anda untuk mengelola basis data.

Berikut adalah tugas DBA umum untuk instans DB yang menjalankan Oracle:

- Tugas sistem

Memutus koneksi sesi	Metode Amazon RDS: `rdsadmin.rdsadmin_util.disconnect` Metode Oracle: `alter system disconnect session`
Mengakhiri sesi	Metode Amazon RDS: `rdsadmin.rdsadmin_util.kill` Metode Oracle: `alter system kill session`
Membatalkan pernyataan SQL dalam sesi	Metode Amazon RDS: `rdsadmin.rdsadmin_util.cancel` Metode Oracle: `alter system cancel sql`
Mengaktifkan dan menonaktifkan sesi terbatas	Metode Amazon RDS: `rdsadmin.rdsadmin_util.restricted_session` Metode Oracle: `alter system enable restricted session`
Menghapus kolam bersama	Metode Amazon RDS: `rdsadmin.rdsadmin_util.flush_shared_pool` Metode Oracle: `alter system flush shared_pool`
Menghapus cache buffer	Metode Amazon RDS: `rdsadmin.rdsadmin_util.flush_buffer_cache` Metode Oracle: `alter system flush buffer_cache`
Memberikan hak istimewa SELECT atau EXECUTE pada objek SYS	Metode Amazon RDS: `rdsadmin.rdsadmin_util.grant_sys_object` Metode Oracle: `grant`

Mencabut hak istimewa SELECT atau EXECUTE pada objek SYS	Metode Amazon RDS: `rdsadmin.rdsadmin_util.revoke_sys_object` Metode Oracle: `revoke`
Mengelola tampilan RDS_X$ untuk instans DB Oracle	Metode Amazon RDS: `rdsadmin.rdsadmin_util.create_sys_x$_view` Metode Oracle: `CREATE VIEW`
Memberikan hak istimewa kepada pengguna non-master	Metode Amazon RDS: `grant`
Membuat fungsi kustom untuk memverifikasi kata sandi	Metode Amazon RDS: `rdsadmin.rdsadmin_password_verify.create_verify_function` Metode Amazon RDS: `rdsadmin.rdsadmin_password_verify.create_passthrough_verify_fcn`
Menyiapkan server DNS kustom	—
Mencantumkan peristiwa diagnostik sistem yang diizinkan	Metode Amazon RDS: `rdsadmin.rdsadmin_util.list_allowed_system_events` Metode Oracle: —
Menyetel peristiwa diagnostik sistem	Metode Amazon RDS: `rdsadmin.rdsadmin_util.set_allowed_system_events` Metode Oracle: `ALTER SYSTEM SET EVENTS 'set_event_clause'`

Mencantumkan peristiwa diagnostik sistem yang ditetapkan	Metode Amazon RDS: `rdsadmin.rdsadmin_util.list_set_system_events` Metode Oracle: `ALTER SESSION SET EVENTS 'IMMEDIATE EVENTDUMP(SYSTEM)'`
Membatalkan pengaturan peristiwa diagnostik sistem	Metode Amazon RDS: `rdsadmin.rdsadmin_util.unset_system_event` Metode Oracle: `ALTER SYSTEM SET EVENTS 'unset_event_clause'`

- Tugas basis data

Mengubah nama global basis data	Metode Amazon RDS: `rdsadmin.rdsadmin_util.rename_global_name` Metode Oracle: `alter database rename`
Membuat dan mengukur tablespace	Metode Amazon RDS: `create tablespace` Metode Oracle: `alter database`
Menetapkan tablespace default	Metode Amazon RDS: `rdsadmin.rdsadmin_util.alter_default_tablespace` Metode Oracle: `alter database default tablespace`
Menetapkan tablespace sementara default	Metode Amazon RDS: `rdsadmin.rdsadmin_util.alter_default_temp_tablespace` Metode Oracle: `alter database default temporary tablespace`
Membuat tablespace sementara di penyimpanan instans	Metode Amazon RDS: `rdsadmin.rdsadmin_util.create_inst_store_tmp_tblspace` Metode Oracle: `create temporary tablespace`

Membuat checkpoint basis data	Metode Amazon RDS: `rdsadmin.rdsadmin_util.checkpoint` Metode Oracle: `alter system checkpoint`
Mengatur pemulihan terdistribusi	Metode Amazon RDS: `rdsadmin.rdsadmin_util.enable_distr_recovery` Metode Oracle: `alter system enable distributed recovery`
Mengatur zona waktu basis data	Metode Amazon RDS: `rdsadmin.rdsadmin_util.alter_db_time_zone` Metode Oracle: `alter database set time_zone`
Bekerja dengan tabel eksternal Oracle	—
Membuat laporan performa dengan Automatic Workload Repository (AWR)	Metode Amazon RDS: Prosedur `rdsadmin.rdsadmin_diagnostic_util` Metode Oracle: Paket `dbms_workload_repository`
Menyesuaikan tautan basis data untuk penggunaan instans DB di VPC	—
Mengatur edisi default untuk instans DB	Metode Amazon RDS: `rdsadmin.rdsadmin_util.alter_default_edition` Metode Oracle: `alter database default edition`
Mengaktifkan audit untuk tabel SYS.AUD$	Metode Amazon RDS: `rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table` Metode Oracle: `audit`

Menonaktifkan pengauditan untuk tabel SYS.AUD$	Metode Amazon RDS: `rdsadmin.rdsadmin_master_util.noaudit_all_sys_aud_table` Metode Oracle: `noaudit`
Membersihkan pembuatan indeks online yang terganggu	Metode Amazon RDS: `rdsadmin.rdsadmin_dbms_repair.online_index_clean` Metode Oracle: `dbms_repair.online_index_clean`
Melewatkan blok yang rusak	Metode Amazon RDS: Beberapa prosedur `rdsadmin.rdsadmin_dbms_repair` Metode Oracle: Paket `dbms_repair`
Mengubah ukuran tablespace, file data, dan file temp	Metode Amazon RDS: Prosedur `rdsadmin.rdsadmin_util.resize_temp_tablespace`, `rdsadmin.rdsadmin_util.resize_tempfile`, atau `rdsadmin.rdsadmin_util.autoextend_tempfile` Prosedur `rdsadmin.rdsadmin_util.resize_datafile` atau `rdsadmin.rdsadmin_util.autoextend_datafile` Metode Oracle: —
Membersihkan keranjang sampah	Metode Amazon RDS: `EXEC rdsadmin.rdsadmin_util.purge_dba_recyclebin` Metode Oracle: `purge dba_recyclebin`
Mengatur nilai default yang ditampilkan untuk redaksi penuh	Metode Amazon RDS: `EXEC rdsadmin.rdsadmin_util.dbms_redact_upd_full_rdct_val` Metode Oracle: `exec dbms_redact.UPDATE_FULL_REDACTION_VALUES`

- Tugas log

Mengatur logging paksa	Metode Amazon RDS: `rdsadmin.rdsadmin_util.force_logging` Metode Oracle: `alter database force logging`
Mengatur logging tambahan	Metode Amazon RDS: `rdsadmin.rdsadmin_util.alter_supplemental_logging` Metode Oracle: `alter database add supplemental log`
Mengganti file log online	Metode Amazon RDS: `rdsadmin.rdsadmin_util.switch_logfile` Metode Oracle: `alter system switch logfile`
Menambahkan log pengulangan online	Metode Amazon RDS: `rdsadmin.rdsadmin_util.add_logfile`
Menghapus log pengulangan online	Metode Amazon RDS: `rdsadmin.rdsadmin_util.drop_logfile`
Mengubah ukuran log pengulangan online	—
Mempertahankan log pengulangan yang diarsipkan	Metode Amazon RDS: `rdsadmin.rdsadmin_util.set_configuration`

Mengunduh log pengulangan yang diarsipkan dari Amazon S3	Metode Amazon RDS: `rdsadmin.rdsadmin_archive_log_download.download_log_with_seqnum` Metode Amazon RDS: `rdsadmin.rdsadmin_archive_log_download.download_logs_in_seqnum_range`
Mengakses log pengulangan online dan yang diarsipkan	Metode Amazon RDS: `rdsadmin.rdsadmin_master_util.create_archivelog_dir` Metode Amazon RDS: `rdsadmin.rdsadmin_master_util.create_onlinelog_dir`

- Tugas RMAN

Memvalidasi file database dalam RDS untuk Oracle	Metode Amazon RDS: `rdsadmin_rman_util. procedure` Metode Oracle: `RMAN VALIDATE`

Mengaktifkan dan menonaktifkan pelacakan perubahan blok	Metode Amazon RDS: `rdsadmin_rman_util . procedure` Metode Oracle: `ALTER DATABASE`
Memeriksa ulang log pengulangan yang diarsipkan	Metode Amazon RDS: `rdsadmin_rman_util .crosscheck_archivelog` Metode Oracle: `RMAN BACKUP`
Mencadangkan file log redo yang diarsipkan	Metode Amazon RDS: `rdsadmin_rman_util . procedure` Metode Oracle: `RMAN BACKUP`
Melakukan pencadangan basis data penuh	Metode Amazon RDS: `rdsadmin_rman_util .backup_database_full` Metode Oracle: `RMAN BACKUP`
Melakukan pencadangan basis data inkremental	Metode Amazon RDS: `rdsadmin_rman_util .backup_database_incremental` Metode Oracle: `RMAN BACKUP`

Mencadangkan ruang tabel	Metode Amazon RDS: `rdsadmin_rman_util.backup_database_tablespace` Metode Oracle: `RMAN BACKUP`

- Tugas Scheduler Oracle

Memodifikasi pekerjaan DBMS_SCHEDULER	Metode Amazon RDS: `dbms_scheduler.set_attribute` Metode Oracle: `dbms_scheduler.set_attribute`
Memodifikasi AutoTask jendela pemeliharaan	Metode Amazon RDS: `dbms_scheduler.set_attribute` Metode Oracle: `dbms_scheduler.set_attribute`
Mengatur zona waktu untuk pekerjaan Oracle Scheduler	Metode Amazon RDS: `dbms_scheduler.set_scheduler_attribute` Metode Oracle: `dbms_scheduler.set_scheduler_attribute`

Menonaktifkan pekerjaan Oracle Scheduler yang dimiliki oleh SYS	Metode Amazon RDS: `rdsadmin.rdsadmin_dbms_scheduler.disable` Metode Oracle: `dbms_scheduler.disable`
Mengaktifkan pekerjaan Oracle Scheduler yang dimiliki oleh SYS	Metode Amazon RDS: `rdsadmin.rdsadmin_dbms_scheduler.enable` Metode Oracle: `dbms_scheduler.enable`
Memodifikasi interval pengulangan Oracle Scheduler untuk pekerjaan tipe CALENDAR	Metode Amazon RDS: `rdsadmin.rdsadmin_dbms_scheduler.set_attribute` Metode Oracle: `dbms_scheduler.set_attribute`
Memodifikasi interval pengulangan Oracle Scheduler untuk pekerjaan tipe NAMED	Metode Amazon RDS: `rdsadmin.rdsadmin_dbms_scheduler.set_attribute` Metode Oracle: `dbms_scheduler.set_attribute`

Menonaktifkan autocommit untuk pembuatan pekerjaan Oracle Scheduler	Metode Amazon RDS: `rdsadmin.rdsadmin_dbms_scheduler.set_no_commit_flag` Metode Oracle: `dbms_isched.set_no_commit_flag`

- Tugas diagnostik

Daftar insiden	Metode Amazon RDS: `rdsadmin.rdsadmin_adrci_util.list_adrci_incidents` Metode Oracle: Perintah ADRCI `show incident`
Mencantumkan masalah	Metode Amazon RDS: `rdsadmin.rdsadmin_adrci_util.list_adrci_problem` Metode Oracle: Perintah ADRCI `show problem`
Membuat paket insiden	Metode Amazon RDS: `rdsadmin.rdsadmin_adrci_util.create_adrci_package` Metode Oracle: Perintah ADRCI `ips create package`

Menampilkan file jejak	Metode Amazon RDS: `rdsadmin.rdsadmin_adrci_util.show_adrci_tracefile` Metode Oracle: Perintah ADRCI `show tracefile`

- Tugas lainnya

Membuat dan menghapus direktori di ruang penyimpanan data utama	Metode Amazon RDS: `rdsadmin.rdsadmin_util.create_directory` Metode Oracle: `CREATE DIRECTORY` Metode Amazon RDS: `rdsadmin.rdsadmin_util.drop_directory` Metode Oracle: `DROP DIRECTORY`
Membuat daftar file di direktori instans DB	Metode Amazon RDS: `rdsadmin.rds_file_util.listdir` Metode Oracle: —
Membaca file di direktori instans DB	Metode Amazon RDS: `rdsadmin.rds_file_util.read_text_file` Metode Oracle: —

Mengakses file Opatch	Metode Amazon RDS: `rdsadmin.rds_file_util.read_text_file` atau `rdsadmin.tracefile_listing` Metode Oracle: `opatch`
Mengatur parameter untuk tugas penasihat	Metode Amazon RDS: `rdsadmin.rdsadmin_util.advisor_task_set_parameter` Metode Oracle: Berbagai prosedur paket disimpan
Menonaktifkan AUTO_STATS_ADVISTOR_TASK	Metode Amazon RDS: `rdsadmin.rdsadmin_util.advisor_task_drop` Metode Oracle: —
Mengaktifkan kembali AUTO_STATS_ADVISTOR_TASK	Metode Amazon RDS: `rdsadmin.rdsadmin_util.dbms_stats_init` Metode Oracle: —

Anda juga dapat menggunakan prosedur Amazon RDS untuk integrasi Amazon S3 dengan Oracle dan untuk menjalankan tugas basis data OEM Management Agent. Lihat informasi yang lebih lengkap di Integrasi Amazon S3 dan Melakukan tugas basis data dengan Management Agent.

Melakukan tugas sistem umum untuk instans DB Oracle

Setelah itu, Anda dapat menemukan cara melakukan tugas DBA umum tertentu yang terkait dengan sistem di instans DB Amazon RDS yang menjalankan Oracle. Untuk memberikan pengalaman

layanan terkelola, Amazon RDS tidak memberikan akses shell ke instans DB, dan membatasi akses ke sejumlah prosedur dan tabel sistem tertentu yang memerlukan hak istimewa tingkat lanjut.

Topik

Memutus koneksi sesi

Untuk memutus koneksi sesi saat ini dengan mengakhiri proses server khusus, gunakan prosedur `rdsadmin.rdsadmin_util.disconnect` Amazon RDS. Prosedur `disconnect` memiliki parameter berikut.

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
`sid`	number	—	Ya	Pengidentifikasi sesi.
`serial`	number	—	Ya	Nomor seri sesi.
`method`	varchar	'IMMEDIATE'	Tidak	Nilai yang valid adalah `'IMMEDIATE'` atau

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
				`'POST_TRANSACTION'` .

Berikut ini adalah contoh cara memutus koneksi sesi.

```
begin
    rdsadmin.rdsadmin_util.disconnect(
        sid    => sid,
        serial => serial_number);
end;
/
```

Untuk mendapatkan pengidentifikasi sesi dan nomor sesi, buka tampilan `V$SESSION` . Contoh berikut mendapatkan semua sesi untuk pengguna `AWSUSER`.

```
SELECT SID, SERIAL#, STATUS FROM V$SESSION WHERE USERNAME = 'AWSUSER';
```

Basis data harus terbuka untuk menggunakan metode ini. Untuk informasi selengkapnya tentang cara memutuskan koneksi sesi, lihat ALTER SYSTEM di dokumentasi Oracle.

Mengakhiri sesi

Untuk mengakhiri sesi, gunakan prosedur `rdsadmin.rdsadmin_util.kill` Amazon RDS. Prosedur `kill` memiliki parameter berikut.

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
`sid`	number	—	Ya	Pengidentifikasi sesi.
`serial`	number	—	Ya	Nomor seri sesi.
`method`	varchar	null	Tidak	Nilai yang valid adalah `'IMMEDIATE'` atau `'PROCESS'` . Jika Anda menentukan `IMMEDIATE` , efeknya akan sama

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
				seperti menjalankan pernyataan berikut: `ALTER SYSTEM KILL SESSION 'sid,serial#' IMMEDIATE` Jika Anda menentukan `PROCESS`, Anda akan mengakhiri proses yang terkait dengan sesi. Hanya tentukan `PROCESS` jika mengakhiri sesi menggunakan `IMMEDIATE` tidak berhasil.

Untuk mendapatkan pengidentifikasi sesi dan nomor seri sesi, buat kueri tampilan `V$SESSION`. Contoh berikut mendapatkan semua sesi untuk pengguna *`AWSUSER`*.

```
SELECT SID, SERIAL#, STATUS FROM V$SESSION WHERE USERNAME = 'AWSUSER';
```

Berikut ini adalah contoh cara mengakhiri sesi.

```
BEGIN
    rdsadmin.rdsadmin_util.kill(
        sid    => sid,
        serial => serial_number,
        method => 'IMMEDIATE');
END;
/
```

Contoh berikut akan mengakhiri proses yang terkait dengan sesi.

```
BEGIN
    rdsadmin.rdsadmin_util.kill(
```

```
        sid    => sid,
        serial => serial_number,
        method => 'PROCESS');
END;
/
```

Membatalkan pernyataan SQL dalam sesi

Untuk membatalkan pernyataan SQL dalam sesi, gunakan prosedur `rdsadmin.rdsadmin_util.cancel` Amazon RDS.

> **Note**
>
> Prosedur ini mendukung Oracle Database 19c (19.0.0) dan semua versi utama dan minor yang lebih tinggi dari RDS for Oracle.

Prosedur `cancel` memiliki parameter berikut.

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
`sid`	number	—	Ya	Pengidentifikasi sesi.
`serial`	number	—	Ya	Nomor seri sesi.
`sql_id`	varchar2	null	Tidak	Pengidentifikasi SQL dari pernyataan SQL.

Contoh berikut membatalkan pernyataan SQL dalam sesi.

```
begin
    rdsadmin.rdsadmin_util.cancel(
        sid    => sid,
        serial => serial_number,
        sql_id => sql_id);
end;
/
```

Untuk mendapatkan pengidentifikasi sesi, nomor seri sesi, dan pengidentifikasi SQL dari pernyataan SQL, buat kueri tampilan `V$SESSION`. Contoh berikut mendapatkan semua sesi untuk pengidentifikasi SQL untuk pengguna `AWSUSER`.

```
select SID, SERIAL#, SQL_ID, STATUS from V$SESSION where USERNAME = 'AWSUSER';
```

Mengaktifkan dan menonaktifkan sesi terbatas

Untuk mengaktifkan dan menonaktifkan sesi terbatas, gunakan prosedur `rdsadmin.rdsadmin_util.restricted_session` Amazon RDS. Prosedur `restricted_session` memiliki parameter berikut.

Nama parameter	Tipe data	Default	Ya	Deskripsi
`p_enable`	boolean	true	Tidak	Tetapkan ke `true` untuk memungkinkan sesi terbatas, `false` untuk menonaktifkan sesi terbatas.

Contoh berikut menunjukkan cara mengaktifkan dan menonaktifkan sesi terbatas.

```
/* Verify that the database is currently unrestricted. */

SELECT LOGINS FROM V$INSTANCE;

LOGINS
-------
ALLOWED

/* Enable restricted sessions */

EXEC rdsadmin.rdsadmin_util.restricted_session(p_enable => true);


/* Verify that the database is now restricted. */

SELECT LOGINS FROM V$INSTANCE;
```

```
LOGINS
----------
RESTRICTED


/* Disable restricted sessions */

EXEC rdsadmin.rdsadmin_util.restricted_session(p_enable => false);


/* Verify that the database is now unrestricted again. */

SELECT LOGINS FROM V$INSTANCE;

LOGINS
-------
ALLOWED
```

Menghapus kolam bersama

Untuk menghapus kolam bersama, terapkan prosedur `rdsadmin.rdsadmin_util.flush_shared_pool` Amazon RDS. Prosedur `flush_shared_pool` tidak memiliki parameter.

Contoh berikut menghapus kolam bersama.

```
EXEC rdsadmin.rdsadmin_util.flush_shared_pool;
```

Menghapus cache buffer

Untuk menghapus cache buffer, terapkan prosedur `rdsadmin.rdsadmin_util.flush_buffer_cache` Amazon RDS. Prosedur `flush_buffer_cache` tidak memiliki parameter.

Contoh berikut akan menghapus cache buffer.

```
EXEC rdsadmin.rdsadmin_util.flush_buffer_cache;
```

Menghapus cache flash smart flash basis data

Untuk menghapus cache smart flash basis data, terapkan prosedur `rdsadmin.rdsadmin_util.flush_flash_cache` Amazon RDS. Prosedur

`flush_flash_cache` tidak memiliki parameter. Contoh berikut akan menghapus cache smart flash basis data.

```
EXEC rdsadmin.rdsadmin_util.flush_flash_cache;
```

Untuk informasi selengkapnya tentang penggunaan cache smart flash basis data dengan RDS for Oracle, lihat Menyimpan data sementara di penyimpanan instans RDS for Oracle.

Memberikan hak istimewa SELECT atau EXECUTE pada objek SYS

Biasanya Anda mentransfer hak istimewa dengan menggunakan peran, yang dapat berisi banyak objek. Untuk memberikan hak istimewa kepada satu objek, gunakan prosedur `rdsadmin.rdsadmin_util.grant_sys_object` Amazon RDS. Prosedur ini memberikan hanya hak istimewa yang telah diberikan kepada pengguna master melalui peran atau pemberian langsung.

Prosedur `grant_sys_object` memiliki parameter berikut.

> **Important**
>
> Untuk semua nilai parameter, gunakan huruf besar kecuali jika Anda membuat pengguna dengan pengidentifikasi yang peka huruf besar. Misalnya, jika Anda menjalankan `CREATE USER myuser` atau `CREATE USER MYUSER`, kamus data menyimpan `MYUSER`. Namun, jika Anda menggunakan tanda kutip ganda di `CREATE USER "MyUser"`, kamus data menyimpan `MyUser`.

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
p_obj_name	varchar2	—	Ya	Nama objek untuk menerima pemberian hak istimewa. Objek dapat berupa direktori, fungsi, paket, prosedur, urutan, tabel, atau tampilan. Nama objek harus dieja persis seperti yang muncul di DBA_OBJECTS. Sebagian besar

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
				objek sistem didefinis ikan dalam huruf besar, jadi kami menyarankan Anda untuk mencobanya terlebih dahulu.
`p_grantee`	varchar2	—	Ya	Nama objek untuk menerima pemberian hak istimewa. Objek dapat berupa skema atau peran.
`p_privilege`	varchar2	null	Ya	—
`p_grant_option`	boolean	false	Tidak	Tetapkan ke `true` untuk menggunakan opsi pemberian. Parameter `p_grant_option` mendukung 12.1.0.2.v4 dan versi yang lebih baru, semua versi 12.2.0.1, dan semua versi 19.0.0.

Contoh berikut memberikan hak istimewa tertentu pada objek yang diberi nama `V_$SESSION` ke pengguna bernama `USER1`.

```
begin
    rdsadmin.rdsadmin_util.grant_sys_object(
        p_obj_name  => 'V_$SESSION',
        p_grantee   => 'USER1',
        p_privilege => 'SELECT');
end;
/
```

Contoh berikut memberikan hak istimewa tertentu pada objek yang diberi nama `V_$SESSION` ke pengguna bernama `USER1` dengan opsi pemberian.

```
begin
    rdsadmin.rdsadmin_util.grant_sys_object(
        p_obj_name     => 'V_$SESSION',
        p_grantee      => 'USER1',
        p_privilege    => 'SELECT',
        p_grant_option => true);
end;
/
```

Agar dapat memberikan hak istimewa kepada suatu objek, akun Anda harus memiliki hak istimewa yang diberikan kepadanya secara langsung dengan opsi pemberian, atau melalui peran yang diberikan menggunakan `with admin option`. Dalam kasus yang paling umum, Anda mungkin ingin memberikan `SELECT` pada tampilan DBA yang telah diberikan kepada peran `SELECT_CATALOG_ROLE`. Jika peran tersebut belum diberikan secara langsung kepada pengguna Anda menggunakan `with admin option`, Anda tidak dapat mentransfer hak istimewa tersebut. Jika Anda memiliki hak istimewa DBA, Anda dapat memberikan peran tersebut secara langsung ke pengguna lain.

Contoh berikut memberikan `SELECT_CATALOG_ROLE` dan `EXECUTE_CATALOG_ROLE` kepada `USER1`. Sejak `with admin option` digunakan, `USER1` sekarang dapat memberikan akses ke objek SYS yang telah diberikan kepada `SELECT_CATALOG_ROLE`.

```
GRANT SELECT_CATALOG_ROLE TO USER1 WITH ADMIN OPTION;
GRANT EXECUTE_CATALOG_ROLE to USER1 WITH ADMIN OPTION;
```

Objek yang telah diberikan kepada `PUBLIC` tidak perlu diberikan ulang. Jika Anda menggunakan prosedur `grant_sys_object` untuk memberikan ulang akses, pemanggilan prosedur berhasil.

Mencabut hak istimewa SELECT atau EXECUTE pada objek SYS

Untuk mencabut hak pada satu objek, gunakan prosedur `rdsadmin.rdsadmin_util.revoke_sys_object` Amazon RDS. Prosedur ini membatalkan hanya hak istimewa yang telah diberikan kepada pengguna master melalui peran atau pemberian langsung.

Prosedur `revoke_sys_object` memiliki parameter berikut.

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
p_obj_name	varchar2	—	Ya	Nama objek yang akan dicabut hak istimewan ya. Objek dapat berupa direktori, fungsi, paket, prosedur, urutan, tabel, atau tampilan. Nama objek harus dieja persis seperti yang muncul di DBA_OBJECTS . Sebagian besar objek sistem didefinisikan dalam huruf besar, jadi kami menyarankan Anda untuk mencobanya terlebih dahulu.
p_revokee	varchar2	—	Ya	Nama objek yang akan dicabut hak istimewan ya. Objek dapat berupa skema atau peran.
p_privilege	varchar2	null	Ya	—

Contoh berikut mencabut hak istimewa tertentu pada objek yang diberi nama V_$SESSION ke pengguna bernama USER1.

```
begin
    rdsadmin.rdsadmin_util.revoke_sys_object(
        p_obj_name  => 'V_$SESSION',
        p_revokee   => 'USER1',
        p_privilege => 'SELECT');
end;
/
```

Mengelola tampilan RDS_X$ untuk instans DB Oracle

Anda mungkin perlu mengakses tabel tetap `SYS.X$`, yang hanya dapat diakses oleh `SYS`. Untuk membuat tampilan `SYS.RDS_X$` pada tabel `X$` yang memenuhi syarat, gunakan prosedur dalam paket `rdsadmin.rdsadmin_util`. Pengguna master Anda secara otomatis diberikan hak istimewa `SELECT … WITH GRANT OPTION` pada tampilan `RDS_X$`.

Prosedur `rdsadmin.rdsadmin_util` tersedia dalam versi mesin basis data berikut:

- `21.0.0.0.ru-2023-10.rur-2023-10.r1` dan versi Oracle Database 21c yang lebih baru
- `19.0.0.0.ru-2023-10.rur-2023-10.r1` dan versi Oracle Database 19c yang lebih baru

> **Important**
>
> Secara internal, paket `rdsadmin.rdsadmin_util` membuat tampilan pada tabel `X$`. Tabel `X$` adalah objek sistem internal yang tidak dijelaskan dalam dokumentasi Oracle Database. Sebaiknya uji tampilan spesifik dalam basis data non-produksi dan hanya membuat tampilan di basis data produksi Anda berdasarkan panduan Dukungan Oracle.

Mencantumkan tabel tetap X$ yang memenuhi syarat untuk digunakan dalam tampilan RDS_X$

Untuk mencantumkan tabel `X$` yang memenuhi syarat untuk digunakan dalam tampilan `RDS_X$`, gunakan prosedur `rdsadmin.rdsadmin_util.list_allowed_sys_x$_views` RDS. Prosedur ini tidak menerima parameter. Pernyataan berikut mencantumkan semua tabel `X$` yang memenuhi syarat (sampel output disertakan).

```
SQL> SET SERVEROUTPUT ON
SQL> SELECT * FROM TABLE(rdsadmin.rdsadmin_util.list_allowed_sys_x$_views);

'X$BH'
'X$K2GTE'
'X$KCBWBPD'
'X$KCBWDS'
'X$KGLLK'
'X$KGLOB'
'X$KGLPN'
'X$KSLHOT'
'X$KSMSP'
'X$KSPPCV'
```

```
'X$KSPPI'
'X$KSPPSV'
'X$KSQEQ'
'X$KSQRS'
'X$KTUXE'
'X$KQRFP'
```

Daftar tabel X$ yang memenuhi syarat dapat berubah seiring waktu. Untuk memastikan bahwa daftar tabel tetap X$ yang memenuhi syarat Anda selalu terbaru, jalankan kembali `list_allowed_sys_x$_views` secara berkala.

Membuat tampilan SYS.RDS_X$

Untuk membuat tampilan RDS_X$ pada tabel X$ yang memenuhi syarat, gunakan prosedur `rdsadmin.rdsadmin_util.create_sys_x$_view` RDS. Anda hanya dapat membuat tampilan untuk tabel yang tercantum dalam output `rdsadmin.rdsadmin_util.list_allowed_sys_x$_views`. Prosedur `create_sys_x$_view` menerima parameter berikut.

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
`p_x$_tbl`	varchar2	Null	Ya	Nama tabel X$ yang valid. Nilai harus menjadi salah satu tabel X$ yang dilaporkan oleh `list_allowed_sys_x$_views`.
`p_force_creation`	Boolean	`FALSE`	Tidak	Nilai yang menunjukkan apakah akan memaksa pembuatan tampilan RDS_X$ yang sudah ada untuk tabel X$. Secara default, RDS tidak akan membuat tampilan jika sudah ada. Untuk memaksa pembuatan, tetapkan parameter ini ke `TRUE`.

Contoh berikut membuat tampilan SYS.RDS_X$KGLOB di tabel X$KGLOB. Format untuk nama tampilan adalah RDS_*X$tablename*.

```
SQL> SET SERVEROUTPUT ON
SQL> EXEC rdsadmin.rdsadmin_util.create_sys_x$_view('X$KGLOB');

PL/SQL procedure successfully completed.
```

Kueri kamus data berikut mencantumkan tampilan SYS.RDS_X$KGLOB dan menunjukkan statusnya. Pengguna master Anda secara otomatis diberikan hak istimewa SELECT ... WITH GRANT OPTION pada tampilan ini.

```
SQL> SET SERVEROUTPUT ON
SQL> COL OWNER FORMAT A30
SQL> COL OBJECT_NAME FORMAT A30
SQL> COL STATUS FORMAT A30
SQL> SET LINESIZE 200
SQL> SELECT OWNER, OBJECT_NAME, STATUS
FROM DBA_OBJECTS
WHERE OWNER = 'SYS' AND OBJECT_NAME = 'RDS_X$KGLOB';

OWNER                          OBJECT_NAME                    STATUS
------------------------------ ------------------------------
 ------------------------------
SYS                            RDS_X$KGLOB                    VALID
```

> **Important**
>
> Tabel X$ tidak dijamin untuk tetap sama sebelum dan sesudah pemutakhiran. RDS for Oracle menghapus dan membuat ulang tampilan RDS_X$ pada tabel X$ selama pemutakhiran mesin. Kemudian, hak istimewa SELECT ... WITH GRANT OPTION diberikan kepada pengguna master. Setelah pemutakhiran, berikan hak istimewa kepada pengguna basis data sesuai kebutuhan pada tampilan RDS_X$ yang sesuai.

Mencantumkan tampilan SYS.RDS_X$

Untuk mencantumkan tampilan RDS_X$ yang ada, gunakan prosedur rdsadmin.rdsadmin_util.list_created_sys_x$_views RDS. Prosedur hanya

mencantumkan tampilan yang dibuat oleh prosedur `create_sys_x$_view`. Contoh berikut mencantumkan tabel X$ yang memiliki tampilan RDS_X$ yang sesuai (sampel output disertakan).

```
SQL> SET SERVEROUTPUT ON
SQL> COL XD_TBL_NAME FORMAT A30
SQL> COL STATUS FORMAT A30
SQL> SET LINESIZE 200
SQL> SELECT * FROM TABLE(rdsadmin.rdsadmin_util.list_created_sys_x$_views);

XD_TBL_NAME                    STATUS
------------------------------ ------------------------------
X$BH                           VALID
X$K2GTE                        VALID
X$KCBWBPD                      VALID

3 rows selected.
```

Menghapus tampilan RDS_X$

Untuk menghapus tampilan SYS.RDS_X$, gunakan prosedur `rdsadmin.rdsadmin_util.drop_sys_x$_view` RDS. Anda hanya dapat menghapus tampilan yang tercantum dalam output `rdsadmin.rdsadmin_util.list_allowed_sys_x$_views`. Prosedur `drop_sys_x$_view` menerima parameter berikut.

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
p_x$_tbl	varchar2	Null	Ya	Nama tabel tetap X$ yang valid. Nilai harus menjadi salah satu tabel tetap X$ yang dilaporkan oleh `list_created_sys_x$_views` .

Contoh berikut menghapus tampilan RDS_X$KGLOB, yang dibuat di tabel X$KGLOB.

```
SQL> SET SERVEROUTPUT ON
SQL> EXEC rdsadmin.rdsadmin_util.drop_sys_x$_view('X$KGLOB');

PL/SQL procedure successfully completed.
```

Contoh berikut menunjukkan bahwa tampilan SYS.RDS_X$KGLOB telah dihapus (sampel output disertakan).

```
SQL> SET SERVEROUTPUT ON
SQL> COL OWNER FORMAT A30
SQL> COL OBJECT_NAME FORMAT A30
SQL> COL STATUS FORMAT A30
SQL> SET LINESIZE 200
SQL> SELECT OWNER, OBJECT_NAME, STATUS
FROM DBA_OBJECTS
WHERE OWNER = 'SYS' AND OBJECT_NAME = 'RDS_X$KGLOB';

no rows selected
```

Memberikan hak istimewa kepada pengguna non-master

Anda dapat memberikan hak istimewa pilihan untuk banyak objek dalam skema SYS dengan menggunakan peran SELECT_CATALOG_ROLE. Peran SELECT_CATALOG_ROLE memberi pengguna hak istimewa SELECT pada tampilan kamus data. Contoh berikut memberikan peran SELECT_CATALOG_ROLE kepada pengguna dengan nama user1.

```
GRANT SELECT_CATALOG_ROLE TO user1;
```

Anda dapat memberikan hak istimewa EXECUTE untuk banyak objek dalam skema SYS menggunakan peran EXECUTE_CATALOG_ROLE. Peran EXECUTE_CATALOG_ROLE memberi pengguna hak istimewa EXECUTE untuk paket dan prosedur dalam kamus data. Contoh berikut memberikan peran EXECUTE_CATALOG_ROLE kepada pengguna dengan nama user1.

```
GRANT EXECUTE_CATALOG_ROLE TO user1;
```

Contoh berikut mendapatkan izin yang peran SELECT_CATALOG_ROLE dan EXECUTE_CATALOG_ROLE izinkan.

```
  SELECT *
    FROM ROLE_TAB_PRIVS
   WHERE ROLE IN ('SELECT_CATALOG_ROLE','EXECUTE_CATALOG_ROLE')
ORDER BY ROLE, TABLE_NAME ASC;
```

Contoh berikut membuat pengguna non-master bernama user1, memberikan hak istimewa CREATE SESSION, dan memberikan hak istimewa SELECT pada basis data yang diberi nama sh.sales.

```
CREATE USER user1 IDENTIFIED BY PASSWORD;
GRANT CREATE SESSION TO user1;
GRANT SELECT ON sh.sales TO user1;
```

Membuat fungsi kustom untuk memverifikasi kata sandi

Anda dapat membuat fungsi verifikasi kata sandi kustom dengan cara berikut:

- Untuk menggunakan logika verifikasi standar dan untuk menyimpan fungsi Anda di skema SYS, gunakan prosedur `create_verify_function`.
- Untuk menggunakan logika verifikasi kustom atau untuk menghindari menyimpan fungsi Anda dalam skema SYS, gunakan prosedur `create_passthrough_verify_fcn`.

Prosedur create_verify_function

Anda dapat membuat fungsi kustom untuk memverifikasi kata sandi dengan menerapkan prosedur `rdsadmin.rdsadmin_password_verify.create_verify_function` Amazon RDS. Prosedur `create_verify_function` mendukung versi 12.1.0.2.v5 dan semua versi utama dan minor yang lebih baru dari RDS for Oracle.

Prosedur `create_verify_function` memiliki parameter berikut.

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
`p_verify_function_name`	varchar2	—	Ya	Nama untuk fungsi kustom Anda. Fungsi ini dibuat untuk Anda dalam skema SYS. Anda menetapkan fungsi ini ke profil pengguna.
`p_min_length`	number	8	Tidak	Jumlah karakter minimum wajib diisi.
`p_max_length`	number	256	Tidak	Jumlah maksimum karakter diperbolehkan.

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
p_min_letters	number	1	Tidak	Jumlah huruf minimum wajib diisi.
p_min_uppercase	number	0	Tidak	Jumlah huruf besar minimum wajib diisi.
p_min_lowercase	number	0	Tidak	Jumlah huruf kecil minimum wajib diisi.
p_min_digits	number	1	Tidak	Jumlah digit minimum wajib diisi.
p_min_special	number	0	Tidak	Jumlah karakter khusus minimum wajib diisi.
p_min_different_chars	number	3	Tidak	Jumlah minimum karakter berbeda wajib diisi antara kata sandi lama dan baru.
p_disallow_username	boolean	true	Tidak	Tetapkan ke `true` untuk tidak mengizinkan nama pengguna dalam kata sandi.
p_disallow_reverse	boolean	true	Tidak	Tetapkan ke `true` untuk tidak mengizinkan pembalikan nama pengguna dalam kata sandi.
p_disallow_db_name	boolean	true	Tidak	Tetapkan ke `true` agar tidak mengizinkan nama basis data atau server ke dalam kata sandi.

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
p_disallow_simple_strings	boolean	true	Tidak	Tetapkan ke true untuk tidak mengizinkan string simpel ke dalam kata sandi.
p_disallow_whitespace	boolean	false	Tidak	Tetapkan ke true untuk tidak mengizinkan karakter dengan spasi dalam kata sandi.
p_disallow_at_sign	boolean	false	Tidak	Tetapkan ke true untuk tidak mengizinkan karakter @ dalam kata sandi.

Anda dapat membuat beberapa fungsi verifikasi kata sandi.

Ada pembatasan pada nama fungsi kustom Anda. Fungsi kustom Anda tidak dapat memiliki nama yang sama dengan objek sistem yang sudah ada. Panjang nama tidak boleh lebih dari 30 karakter. Selain itu, nama harus menyertakan salah satu dari string berikut: PASSWORD, VERIFY, COMPLEXITY, ENFORCE, atau STRENGTH.

Contoh berikut membuat fungsi dengan nama CUSTOM_PASSWORD_FUNCTION. Fungsi tersebut mengharuskan agar kata sandi memiliki setidaknya 12 karakter, 2 karakter huruf besar, 1 digit, dan 1 karakter khusus, dan kata sandi tidak boleh berisi karakter @.

```
begin
    rdsadmin.rdsadmin_password_verify.create_verify_function(
        p_verify_function_name => 'CUSTOM_PASSWORD_FUNCTION',
        p_min_length           => 12,
        p_min_uppercase        => 2,
        p_min_digits           => 1,
        p_min_special          => 1,
        p_disallow_at_sign     => true);
end;
/
```

Untuk melihat teks fungsi verifikasi Anda, kueri DBA_SOURCE. Contoh berikut mendapatkan teks fungsi kata sandi kustom bernama CUSTOM_PASSWORD_FUNCTION.

```
COL TEXT FORMAT a150

  SELECT TEXT
    FROM DBA_SOURCE
   WHERE OWNER = 'SYS'
     AND NAME = 'CUSTOM_PASSWORD_FUNCTION'
ORDER BY LINE;
```

Untuk mengaitkan fungsi verifikasi Anda dengan profil pengguna, gunakan `alter profile`. Contoh berikut mengaitkan fungsi verifikasi dengan profil pengguna DEFAULT.

```
ALTER PROFILE DEFAULT LIMIT PASSWORD_VERIFY_FUNCTION CUSTOM_PASSWORD_FUNCTION;
```

Untuk melihat keterkaitan profil pengguna dengan fungsi verifikasi, kueri DBA_PROFILES. Contoh berikut mendapatkan profil yang terkait dengan fungsi verifikasi kustom bernama CUSTOM_PASSWORD_FUNCTION.

```
SELECT * FROM DBA_PROFILES WHERE RESOURCE_NAME = 'PASSWORD' AND LIMIT =
 'CUSTOM_PASSWORD_FUNCTION';


PROFILE                    RESOURCE_NAME                    RESOURCE  LIMIT
-------------------------  -------------------------------  --------
 ------------------------
DEFAULT                    PASSWORD_VERIFY_FUNCTION         PASSWORD
 CUSTOM_PASSWORD_FUNCTION
```

Contoh berikut mendapatkan semua profil dan fungsi verifikasi kata sandi yang terkait dengan profil.

```
SELECT * FROM DBA_PROFILES WHERE RESOURCE_NAME = 'PASSWORD_VERIFY_FUNCTION';

PROFILE                    RESOURCE_NAME                    RESOURCE  LIMIT
-------------------------  -------------------------------  --------
 ------------------------
DEFAULT                    PASSWORD_VERIFY_FUNCTION         PASSWORD
 CUSTOM_PASSWORD_FUNCTION
RDSADMIN                   PASSWORD_VERIFY_FUNCTION         PASSWORD  NULL
```

Prosedur create_passthrough_verify_fcn

Prosedur `create_passthrough_verify_fcn` mendukung versi 12.1.0.2.v7 dan semua versi utama dan minor yang lebih baru dari RDS for Oracle.

Anda dapat membuat fungsi kustom untuk memverifikasi kata sandi dengan menerapkan prosedur `rdsadmin.rdsadmin_password_verify.create_passthrough_verify_fcn` Amazon RDS. Prosedur `create_passthrough_verify_fcn` memiliki parameter berikut.

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
`p_verify_function_name`	varchar2	—	Ya	Nama untuk fungsi verifikasi kustom Anda. Ini adalah fungsi pembungkus yang dibuat untuk Anda dalam skema SYS, dan tidak berisi logika verifikasi apa pun. Anda menetapkan fungsi ini ke profil pengguna.
`p_target_owner`	varchar2	—	Ya	Pemilik skema untuk fungsi verifikasi kustom Anda.
`p_target_function_name`	varchar2	—	Ya	Nama fungsi kustom Anda saat ini yang berisi logika verifikasi. Fungsi kustom Anda harus menampilkan boolean. Fungsi Anda harus menampilkan `true` jika kata sandi valid dan `false` jika kata sandi tidak valid.

Contoh berikut membuat fungsi verifikasi kata sandi yang menggunakan logika dari fungsi yang disebut `PASSWORD_LOGIC_EXTRA_STRONG`.

```
begin
    rdsadmin.rdsadmin_password_verify.create_passthrough_verify_fcn(
        p_verify_function_name => 'CUSTOM_PASSWORD_FUNCTION',
        p_target_owner         => 'TEST_USER',
        p_target_function_name => 'PASSWORD_LOGIC_EXTRA_STRONG');
end;
/
```

Untuk mengaitkan fungsi verifikasi dengan profil pengguna, gunakan `alter profile`. Contoh berikut mengaitkan fungsi verifikasi dengan profil pengguna `DEFAULT`.

```
ALTER PROFILE DEFAULT LIMIT PASSWORD_VERIFY_FUNCTION CUSTOM_PASSWORD_FUNCTION;
```

Menyiapkan server DNS kustom

Amazon RDS mendukung akses jaringan keluar di instans DB Anda yang menjalankan Oracle. Untuk informasi selengkapnya tentang akses jaringan keluar, termasuk prasyarat, lihat Mengonfigurasi akses UTL_HTTP menggunakan sertifikat dan dompet Oracle.

Amazon RDS Oracle memungkinkan resolusi Layanan Nama Domain (DNS) dari server DNS kustom yang dimiliki oleh pelanggan. Anda hanya dapat menyelesaikan nama domain yang benar-benar memenuhi syarat dari instans DB Amazon RDS Anda melalui server DNS khusus.

Setelah Anda mengatur nama server DNS Anda, perlu waktu hingga 30 menit untuk menyebarkan perubahan ke instans DB Anda. Setelah perubahan diterapkan ke instans DB Anda, semua lalu lintas jaringan keluar mengharuskan pencarian DNS mengkueri server DNS melalui port 53.

Untuk membuat server DNS khusus untuk instans DB Amazon RDS for Oracle, lakukan hal berikut:

- Dari set opsi DHCP yang terlampir pada cloud privat virtual (VPC) Anda, tetapkan opsi `domain-name-servers` ke alamat IP server nama DNS Anda. Untuk informasi selengkapnya, lihat Set opsi DHCP.

  > **Note**
  >
  > Opsi `domain-name-servers` menerima hingga empat nilai, tetapi instans DB Amazon RDS Anda hanya menggunakan nilai pertama.

- Pastikan bahwa server DNS Anda dapat menyelesaikan semua kueri pencarian, termasuk nama DNS publik, nama DNS privat Amazon EC2, dan nama DNS khusus pelanggan. Jika lalu lintas jaringan keluar memuat setiap pencarian DNS yang tidak dapat ditangani server DNS Anda, server DNS Anda harus dikonfigurasikan oleh penyedia DNS hulu yang sesuai.
- Konfigurasikan server DNS Anda untuk menghasilkan respons Protokol Datagram Pengguna (UDP) sebesar 512 byte atau kurang.
- Konfigurasikan server DNS Anda untuk menghasilkan respons Protokol Kontrol Transmisi (TCP) sebesar 1024 byte atau kurang.
- Konfigurasikan server DNS Anda untuk mengizinkan lalu lintas masuk dari instans DB Amazon RDS Anda melalui port 53. Jika server DNS Anda berada di Amazon VPC, maka VPC harus memiliki grup keamanan yang berisi aturan masuk yang mengizinkan lalu lintas UDP dan TCP di port 53. Jika server DNS Anda tidak berada di dalam VPC Amazon, server harus memiliki daftar izin firewall yang sesuai untuk mengizinkan lalu lintas masuk UDP dan TCP di port 53.

  Untuk informasi lebih lanjut, lihat Grup keamanan untuk VPC Anda dan Menambahkan dan menghapus aturan.
- Konfigurasikan VPC instans DB Amazon RDS Anda untuk mengizinkan lalu lintas keluar melalui port 53. VPC Anda harus memiliki grup keamanan yang berisi aturan keluar yang mengizinkan lalu lintas UDP dan TCP di port 53.

  Untuk informasi lebih lanjut, lihat Grup keamanan untuk VPC Anda serta Menambahkan dan menghapus aturan.
- Jalur perutean antara instans DB Amazon RDS dan server DNS harus dikonfigurasi dengan benar untuk memungkinkan lalu lintas DNS.
  - Jika instans DB Amazon RDS dan server DNS tidak berada dalam VPC yang sama, koneksi peering harus diatur di antara keduanya. Untuk informasi selengkapnya, lihat Apa yang dimaksud peering VPC?

Mengatur dan membatalkan pengaturan peristiwa diagnostik sistem

Untuk mengatur dan membatalkan pengaturan peristiwa diagnostik pada tingkat sesi, Anda dapat menggunakan pernyataan `ALTER SESSION SET EVENTS` Oracle SQL. Namun, untuk menyetel peristiwa di tingkat sistem, Anda tidak dapat menggunakan Oracle SQL. Sebaliknya, gunakan prosedur peristiwa sistem dalam paket `rdsadmin.rdsadmin_util`. Prosedur peristiwa sistem tersedia dalam versi mesin berikut:

- Semua versi Oracle Database 21c
- Versi 19.0.0.ru-2020-10.rur-2020-10.rur-2020-10.r1 dan Oracle Database 19c yang lebih baru

  Untuk informasi selengkapnya, lihat Versi 19.0.0.0.ru-2020-10.rur-2020-10.r1 di Catatan Rilis Amazon RDS for Oracle.
- Versi 12.2.0.1.ru-2020-10.rur-2020-10.rur-2020-10.r1 dan Oracle Database 12c Rilis 2 (12.2.0.1)

  Untuk informasi selengkapnya, lihat Versi 12.2.0.1.ru-2020-10.rur-2020-10.r1 di Catatan Rilis Amazon RDS for Oracle.
- Versi 12.1.0.2.V22 dan Oracle Database 12c Rilis 1 (12.1.0.2) yang lebih baru

  Untuk informasi selengkapnya, lihat Versi 12.1.0.2.v22 di Catatan Rilis Amazon RDS for Oracle.

para

> **Important**
>
> Secara internal, paket `rdsadmin.rdsadmin_util` menetapkan peristiwa menggunakan pernyataan `ALTER SYSTEM SET EVENTS`. Pernyataan `ALTER SYSTEM` tidak didokumentasikan di dokumentasi Oracle Database. Beberapa peristiwa diagnostik sistem dapat menghasilkan informasi pelacakan dalam jumlah besar, menyebabkan pertentangan, atau memengaruhi ketersediaan basis data. Sebaiknya uji peristiwa diagnostik tertentu dalam basis data nonproduksi, dan hanya tetapkan persitiwa dalam basis data produksi Anda berdasarkan panduan Dukungan Oracle.

Mencantumkan peristiwa diagnostik sistem yang diizinkan

Untuk mencantumkan peristiwa sistem yang dapat Anda tetapkan, gunakan prosedur `rdsadmin.rdsadmin_util.list_allowed_system_events` Amazon RDS. Prosedur ini tidak menerima parameter.

Contoh berikut mencantumkan semua peristiwa sistem yang dapat Anda tetapkan.

```
SET SERVEROUTPUT ON
EXEC rdsadmin.rdsadmin_util.list_allowed_system_events;
```

Output sampel berikut mencantumkan nomor peristiwa dan deskripsinya. Gunakan prosedur `set_system_event` Amazon RDS untuk mengatur peristiwa dan `unset_system_event` untuk membatalkan pengaturannya.

```
604   - error occurred at recursive SQL level
942   - table or view does not exist
1401  - inserted value too large for column
1403  - no data found
1410  - invalid ROWID
1422  - exact fetch returns more than requested number of rows
1426  - numeric overflow
1427  - single-row subquery returns more than one row
1476  - divisor is equal to zero
1483  - invalid length for DATE or NUMBER bind variable
1489  - result of string concatenation is too long
1652  - unable to extend temp segment by  in tablespace
1858  - a non-numeric character was found where a numeric was expected
4031  - unable to allocate  bytes of shared memory ("","","","")
6502  - PL/SQL: numeric or value error
10027 - Specify Deadlock Trace Information to be Dumped
10046 - enable SQL statement timing
10053 - CBO Enable optimizer trace
10173 - Dynamic Sampling time-out error
10442 - enable trace of kst for ORA-01555 diagnostics
12008 - error in materialized view refresh path
12012 - error on auto execute of job
12504 - TNS:listener was not given the SERVICE_NAME in CONNECT_DATA
14400 - inserted partition key does not map to any partition
31693 - Table data object  failed to load/unload and is being skipped due to error:
```

> **Note**
>
> Daftar peristiwa sistem yang diizinkan dapat berubah seiring waktu. Untuk memastikan bahwa Anda memiliki daftar terbaru peristiwa yang memenuhi syarat, gunakan `rdsadmin.rdsadmin_util.list_allowed_system_events`.

Menyetel peristiwa diagnostik sistem

Untuk menyetel peristiwa sistem, gunakan prosedur `rdsadmin.rdsadmin_util.set_system_event` Amazon RDS.

Anda hanya dapat menyetel peristiwa yang tercantum dalam output dari `rdsadmin.rdsadmin_util.list_allowed_system_events`. Prosedur `set_system_event` menerima parameter berikut.

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
`p_event`	number	—	Ya	Nomor peristiwa sistem. Nilai harus salah satu nomor event yang dilaporkan oleh `list_allowed_system_events` .
`p_level`	number	—	Ya	Level peristiwa. Lihat dokumentasi Oracle Database atau Dukungan Oracle untuk deskripsi berbagai nilai level.

Prosedur `set_system_event` membangun dan menjalankan pernyataan wajib `ALTER SYSTEM SET EVENTS` yang sesuai dengan prinsip berikut:

- Tipe peristiwa (`context` atau `errorstack`) ditentukan secara otomatis.
- Pernyataan dalam formulir `ALTER SYSTEM SET EVENTS 'event LEVEL event_level'` menetapkan peristiwa konteks. Notasi ini setara dengan `ALTER SYSTEM SET EVENTS 'event TRACE NAME CONTEXT FOREVER, LEVEL event_level'`.
- Pernyataan dalam formulir `ALTER SYSTEM SET EVENTS 'event ERRORSTACK (event_level)'` mengatur peristiwa tumpukan kesalahan. Notasi ini setara dengan `ALTER SYSTEM SET EVENTS 'event TRACE NAME ERRORSTACK LEVEL event_level'`.

Contoh berikut menetapkan peristiwa 942 di level 3, dan peristiwa 10442 di level 10. Contoh output disertakan.

```
SQL> SET SERVEROUTPUT ON
SQL> EXEC rdsadmin.rdsadmin_util.set_system_event(942,3);
Setting system event 942 with: alter system set events '942 errorstack (3)'
```

```
PL/SQL procedure successfully completed.

SQL> EXEC rdsadmin.rdsadmin_util.set_system_event(10442,10);
Setting system event 10442 with: alter system set events '10442 level 10'

PL/SQL procedure successfully completed.
```

Mencantumkan peristiwa diagnostik sistem yang ditetapkan

Untuk mencantumkan peristiwa sistem yang sudah ditetapkan saat ini, gunakan prosedur `rdsadmin.rdsadmin_util.list_set_system_events` Amazon RDS. Prosedur ini melaporkan hanya peristiwa yang ditetapkan pada tingkat sistem oleh `set_system_event`.

Contoh berikut mencantumkan peristiwa sistem aktif.

```
SET SERVEROUTPUT ON
EXEC rdsadmin.rdsadmin_util.list_set_system_events;
```

Output sampel berikut menunjukkan daftar peristiwa, tipe peristiwa, level peristiwa yang saat ini ditetapkan, dan waktu saat peristiwa ditetapkan.

```
942 errorstack (3) - set at 2020-11-03 11:42:27
10442 level 10 - set at 2020-11-03 11:42:41

PL/SQL procedure successfully completed.
```

Membatalkan pengaturan peristiwa diagnostik sistem

Untuk membatalkan pengaturan peristiwa sistem, gunakan prosedur `rdsadmin.rdsadmin_util.unset_system_event` Amazon RDS. Anda hanya dapat membatalkan pengaturan peristiwa yang tercantum dalam output dari `rdsadmin.rdsadmin_util.list_allowed_system_events`. Prosedur `unset_system_event` menerima parameter berikut.

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
`p_event`	number	—	Ya	Nomor peristiwa sistem. Nilai harus salah

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
				satu nomor peristiwa yang dilaporkan oleh `list_allowed_system_events` .

Contoh berikut akan membatalkan pengaturan peristiwa 942 dan 10442. Contoh output disertakan.

```
SQL> SET SERVEROUTPUT ON
SQL> EXEC rdsadmin.rdsadmin_util.unset_system_event(942);
Unsetting system event 942 with: alter system set events '942 off'

PL/SQL procedure successfully completed.

SQL> EXEC rdsadmin.rdsadmin_util.unset_system_event(10442);
Unsetting system event 10442 with: alter system set events '10442 off'

PL/SQL procedure successfully completed.
```

Melakukan tugas basis data umum untuk instans DB Oracle

Pada bagian berikut, Anda dapat menemukan cara melakukan tugas DBA umum tertentu terkait basis data di instans DB Amazon RDS Anda yang menjalankan Oracle. Untuk memberikan pengalaman layanan terkelola, Amazon RDS tidak memberikan akses shell ke instans DB. Amazon RDS juga membatasi akses ke prosedur dan tabel sistem tertentu yang memerlukan hak istimewa tingkat lanjut.

Topik

- Mengubah nama global basis data
- Membuat dan mengukur tablespace
- Menetapkan tablespace default
- Menetapkan tablespace sementara default
- Membuat tablespace sementara di penyimpanan instans
- Menambahkan tempfile ke penyimpanan instans di replika baca
- Meletakkan tempfile di replika baca

- Membuat checkpoint basis data
- Mengatur pemulihan terdistribusi
- Mengatur zona waktu basis data
- Bekerja dengan tabel eksternal Oracle
- Membuat laporan performa dengan Automatic Workload Repository (AWR)
- Menyesuaikan tautan basis data untuk penggunaan instans DB di VPC
- Mengatur edisi default untuk instans DB
- Mengaktifkan audit untuk tabel SYS.AUD$
- Menonaktifkan pengauditan untuk tabel SYS.AUD$
- Membersihkan pembuatan indeks online yang terganggu
- Melewatkan blok yang rusak
- Mengubah ukuran tablespace, file data, dan file temp
- Membersihkan keranjang sampah
- Mengatur nilai default yang ditampilkan untuk redaksi penuh

Mengubah nama global basis data

Untuk mengubah nama global basis data, gunakan prosedur `rdsadmin.rdsadmin_util.rename_global_name` Amazon RDS. Prosedur `rename_global_name` memiliki parameter berikut.

Nama parameter	Jenis data	Default	Wajib	Deskripsi
`p_new_global_name`	varchar2	—	Ya	Nama global baru untuk basis data.

Basis data harus terbuka agar namanya dapat diubah. Untuk informasi selengkapnya tentang cara mengubah nama global basis data, lihat ALTER DATABASE di dokumentasi Oracle.

Contoh berikut mengubah nama global basis data menjadi `new_global_name`.

```
EXEC rdsadmin.rdsadmin_util.rename_global_name(p_new_global_name => 'new_global_name');
```

Membuat dan mengukur tablespace

Amazon RDS hanya mendukung Oracle Managed Files (OMF) untuk file data, file log, dan file kontrol. Saat Anda membuat file data dan file log, Anda tidak dapat menyebutkan nama file fisik.

Jika Anda tidak menentukan ukuran file data, tablespace dibuat dengan pengaturan default `AUTOEXTEND ON`, dan ukuran maksimum tidak ditentukan. Dalam contoh berikut, tablespace *`users1`* dapat diperluas secara otomatis.

```
CREATE TABLESPACE users1;
```

Karena pengaturan default ini, ukuran tablespace dapat bertambah untuk menggunakan semua penyimpanan yang dialokasikan. Sebaiknya tentukan ukuran maksimum yang sesuai pada tablespace permanen dan sementara, serta pantau penggunaan ruang dengan cermat.

Contoh berikut membuat tablespace bernama *`users2`* dengan ukuran awal 1 gigabyte. Karena ukuran file data ditentukan, tetapi `AUTOEXTEND ON` tidak ditentukan, tablespace tidak dapat diperluas secara otomatis.

```
CREATE TABLESPACE users2 DATAFILE SIZE 1G;
```

Contoh berikut membuat tablespace bernama *`users3`* dengan ukuran awal 1 gigabyte, autoextend diaktifkan, dan ukuran maksimum 10 gigabyte.

```
CREATE TABLESPACE users3 DATAFILE SIZE 1G AUTOEXTEND ON MAXSIZE 10G;
```

Contoh berikut membuat tablespace sementara bernama *`temp01`*.

```
CREATE TEMPORARY TABLESPACE temp01;
```

Anda dapat mengubah ukuran tablespace bigfile dengan menggunakan `ALTER TABLESPACE`. Anda dapat menentukan ukuran dalam kilobyte (K), megabyte (M), atau gigabyte (G). Contoh berikut mengubah ukuran tablespace bigfile dengan nama *`users_bf`* hingga 200 MB.

```
ALTER TABLESPACE users_bf RESIZE 200M;
```

Contoh berikut menambahkan file data tambahan ke tablespace smallfile dengan nama *`users_sf`*.

```
ALTER TABLESPACE users_sf ADD DATAFILE SIZE 100000M AUTOEXTEND ON NEXT 250m
 MAXSIZE UNLIMITED;
```

Menetapkan tablespace default

Untuk menetapkan tablespace default, gunakan prosedur `rdsadmin.rdsadmin_util.alter_default_tablespace` Amazon RDS. Prosedur `alter_default_tablespace` memiliki parameter berikut.

Nama parameter	Jenis data	Default	Wajib	Deskripsi
`tablespace_name`	varchar	—	Ya	Nama tablespace default.

Contoh berikut menetapkan *`users2`* sebagai tablespace default:

```
EXEC rdsadmin.rdsadmin_util.alter_default_tablespace(tablespace_name => 'users2');
```

Menetapkan tablespace sementara default

Untuk menetapkan tablespace sementara default, gunakan prosedur `rdsadmin.rdsadmin_util.alter_default_temp_tablespace` Amazon RDS. Prosedur `alter_default_temp_tablespace` memiliki parameter berikut.

Nama parameter	Jenis data	Default	Wajib	Deskripsi
`tablespace_name`	varchar	—	Ya	Nama tablespace sementara default.

Contoh berikut menetapkan *`temp01`* sebagai tablespace sementara default.

```
EXEC rdsadmin.rdsadmin_util.alter_default_temp_tablespace(tablespace_name => 'temp01');
```

Membuat tablespace sementara di penyimpanan instans

Untuk membuat tablespace sementara di penyimpanan instans, gunakan prosedur `rdsadmin.rdsadmin_util.create_inst_store_tmp_tblspace` Amazon RDS. Prosedur `create_inst_store_tmp_tblspace` memiliki parameter berikut.

Nama parameter	Jenis data	Default	Wajib	Deskripsi
p_tablespace_name	varchar	—	Ya	Nama tablespace sementara.

Contoh berikut membuat tablespace sementara *temp01* di penyimpanan instans.

```
EXEC rdsadmin.rdsadmin_util.create_inst_store_tmp_tblspace(p_tablespace_name =>
 'temp01');
```

> **Important**
>
> Saat Anda menjalankan rdsadmin_util.create_inst_store_tmp_tblspace, tablespace sementara yang baru dibuat tidak akan otomatis ditetapkan sebagai tablespace sementara default. Untuk menetapkannya sebagai default, lihat Menetapkan tablespace sementara default.

Untuk informasi selengkapnya, lihat Menyimpan data sementara di penyimpanan instans RDS for Oracle.

Menambahkan tempfile ke penyimpanan instans di replika baca

Saat Anda membuat tablespace sementara di instans DB utama, replika baca tidak membuat file tempfile. Misalkan tablespace sementara kosong ada di replika baca Anda karena salah satu alasan berikut:

- Anda meletakkan tempfile dari tablespace di replika baca Anda. Untuk informasi selengkapnya, lihat Meletakkan tempfile di replika baca.
- Anda membuat tablespace sementara baru di instans DB utama. Dalam kasus ini, RDS for Oracle menyinkronkan metadata ke replika baca.

Anda dapat menambahkan tempfile ke tablespace sementara yang kosong dan menyimpan tempfile di penyimpanan instans. Untuk membuat tempfile di penyimpanan instans, gunakan prosedur rdsadmin.rdsadmin_util.add_inst_store_tempfile Amazon RDS. Anda dapat menggunakan prosedur ini hanya di replika baca. Prosedur ini memiliki parameter berikut.

Nama parameter	Jenis data	Default	Wajib	Deskripsi
p_tablespace_name	varchar	—	Ya	Nama tablespace sementara di replika baca Anda.

Dalam contoh berikut, tablespace sementara kosong *temp01* ada di replika baca Anda. Jalankan perintah berikut guna membuat tempfile untuk tablespace ini dan menyimpannya di penyimpanan instans.

```
EXEC rdsadmin.rdsadmin_util.add_inst_store_tempfile(p_tablespace_name => 'temp01');
```

Untuk informasi selengkapnya, lihat Menyimpan data sementara di penyimpanan instans RDS for Oracle.

Meletakkan tempfile di replika baca

Anda tidak dapat meletakkan tablespace sementara yang ada di replika baca. Anda dapat mengubah penyimpanan tempfile di replika baca dari Amazon EBS ke penyimpanan instans atau dari penyimpanan instans ke Amazon EBS. Untuk mencapai tujuan ini, lakukan hal berikut:

1. Letakkan tempfile saat ini di tablespace sementara di replika baca.
2. Buat tempfile baru di penyimpanan yang berbeda.

Untuk meletakkan tempfile, gunakan prosedur `rdsadmin.rdsadmin_util.drop_replica_tempfiles` Amazon RDS. Anda dapat menggunakan prosedur ini hanya di replika baca. Prosedur `drop_replica_tempfiles` memiliki parameter berikut.

Nama parameter	Jenis data	Default	Wajib	Deskripsi
p_tablespace_name	varchar	—	Ya	Nama tablespace sementara di replika baca Anda.

Misalkan tablespace sementara bernama *temp01* berada di penyimpanan instans pada replika baca Anda. Letakkan semua tempfile di tablespace ini dengan menjalankan perintah berikut.

```
EXEC rdsadmin.rdsadmin_util.drop_replica_tempfiles(p_tablespace_name => 'temp01');
```

Untuk informasi selengkapnya, lihat Menyimpan data sementara di penyimpanan instans RDS for Oracle.

Membuat checkpoint basis data

Untuk membuat checkpoint basis data, gunakan prosedur `rdsadmin.rdsadmin_util.checkpoint` Amazon RDS. Prosedur `checkpoint` tidak memiliki parameter.

Contoh berikut membuat checkpoint basis data.

```
EXEC rdsadmin.rdsadmin_util.checkpoint;
```

Mengatur pemulihan terdistribusi

Untuk mengatur pemulihan terdistribusi, gunakan prosedur `rdsadmin.rdsadmin_util.enable_distr_recovery` dan `disable_distr_recovery` Amazon RDS. Prosedur tersebut tidak memiliki parameter.

Contoh berikut mengaktifkan pemulihan terdistribusi.

```
EXEC rdsadmin.rdsadmin_util.enable_distr_recovery;
```

Contoh berikut menonaktifkan pemulihan terdistribusi.

```
EXEC rdsadmin.rdsadmin_util.disable_distr_recovery;
```

Mengatur zona waktu basis data

Anda dapat mengatur zona waktu basis data Amazon RDS Oracle Anda dengan cara berikut:

- Opsi `Timezone`

  Opsi `Timezone` mengubah zona waktu di tingkat host dan memengaruhi semua kolom tanggal dan nilai seperti `SYSDATE`. Untuk informasi selengkapnya, lihat Zona waktu Oracle.
- Prosedur `rdsadmin.rdsadmin_util.alter_db_time_zone` Amazon RDS

Prosedur `alter_db_time_zone` mengubah zona waktu hanya untuk tipe data tertentu dan tidak mengubah `SYSDATE`. Ada batas tambahan pada pengaturan zona waktu yang tercantum dalam dokumentasi Oracle.

> **Note**
>
> Anda juga dapat mengatur zona waktu default untuk Oracle Scheduler. Untuk informasi selengkapnya, lihat Mengatur zona waktu untuk pekerjaan Oracle Scheduler.

Prosedur `alter_db_time_zone` memiliki parameter berikut.

Nama parameter	Jenis data	Default	Wajib	Deskripsi
`p_new_tz`	varchar2	—	Ya	Zona waktu baru sebagai wilayah yang disebutkan atau selisih waktu mutlak dari Waktu Universal Terkoordinasi (UTC). Rentang selisih waktu yang valid berkisar dari -12.00 hingga +14.00.

Contoh berikut mengubah zona waktu menjadi UTC+3.

```
EXEC rdsadmin.rdsadmin_util.alter_db_time_zone(p_new_tz => '+3:00');
```

Contoh berikut mengubah zona waktu ke zona waktu Afrika/Aljazair.

```
EXEC rdsadmin.rdsadmin_util.alter_db_time_zone(p_new_tz => 'Africa/Algiers');
```

Setelah Anda mengubah zona waktu dengan menggunakan prosedur `alter_db_time_zone`, boot ulang instans DB Anda agar perubahan dapat diterapkan. Untuk informasi selengkapnya, lihat Mem-boot ulang instans DB. Untuk informasi selengkapnya tentang peningkatan zona waktu, lihat Pertimbangan zona waktu.

Bekerja dengan tabel eksternal Oracle

Tabel eksternal Oracle adalah tabel dengan data yang tidak ada di basis data. Sebagai gantinya, data tersebut ada di file eksternal yang dapat diakses basis data. Dengan menggunakan tabel eksternal, Anda dapat mengakses data dengan memuatnya ke dalam basis data. Untuk informasi selengkapnya tentang tabel eksternal, lihat Managing external tables di dokumentasi Oracle.

Dengan Amazon RDS, Anda dapat menyimpan file tabel eksternal di objek direktori. Anda dapat membuat objek direktori atau menggunakan yang sudah ditentukan sebelumnya di basis data Oracle, seperti direktori DATA_PUMP_DIR. Untuk informasi tentang cara membuat objek direktori, lihat Membuat dan menghapus direktori di ruang penyimpanan data utama. Anda dapat mengueri tampilan ALL_DIRECTORIES guna menampilkan daftar objek direktori untuk instans basis data Amazon RDS Oracle Anda.

> **Note**
>
> Direktori objek mengarah ke ruang penyimpanan data utama (volume EBS Amazon) yang digunakan oleh instans Anda. Ruang yang digunakan, beserta dengan file data, log pengulangan, audit, jejak, dan file lainnya, diperhitungkan terhadap penyimpanan yang dialokasikan.

Anda dapat memindahkan file data eksternal dari satu basis data Oracle ke basis data lainnya dengan menggunakan paket DBMS_FILE_TRANSFER atau paket UTL_FILE. File data eksternal dipindahkan dari direktori di basis data sumber ke direktori tertentu di basis data tujuan. Untuk informasi tentang cara menggunakan DBMS_FILE_TRANSFER, lihat Mengimpor menggunakan Oracle Data Pump.

Setelah memindahkan file data eksternal, Anda dapat membuat tabel eksternal dengannya. Contoh berikut membuat tabel eksternal yang menggunakan file `emp_xt_file1.txt` di direktori USER_DIR1.

```
CREATE TABLE emp_xt (
  emp_id      NUMBER,
  first_name  VARCHAR2(50),
  last_name   VARCHAR2(50),
  user_name   VARCHAR2(20)
)
ORGANIZATION EXTERNAL (
```

```
  TYPE ORACLE_LOADER
  DEFAULT DIRECTORY USER_DIR1
  ACCESS PARAMETERS (
    RECORDS DELIMITED BY NEWLINE
    FIELDS TERMINATED BY ','
    MISSING FIELD VALUES ARE NULL
    (emp_id,first_name,last_name,user_name)
  )
  LOCATION ('emp_xt_file1.txt')
)
PARALLEL
REJECT LIMIT UNLIMITED;
```

Misalkan Anda ingin memindahkan data yang ada di instans basis data Amazon RDS Oracle ke file data eksternal. Dalam kasus ini, Anda dapat mengisi file data eksternal dengan membuat tabel eksternal dan memilih data dari tabel di basis data. Misalnya, pernyataan SQL berikut membuat tabel eksternal `orders_xt` dengan mengueri tabel `orders` di basis data.

```
CREATE TABLE orders_xt
  ORGANIZATION EXTERNAL
   (
     TYPE ORACLE_DATAPUMP
     DEFAULT DIRECTORY DATA_PUMP_DIR
     LOCATION ('orders_xt.dmp')
   )
   AS SELECT * FROM orders;
```

Dalam contoh ini, data diisi dalam file `orders_xt.dmp` di direktori DATA_PUMP_DIR.

Membuat laporan performa dengan Automatic Workload Repository (AWR)

Untuk mengumpulkan data performa dan membuat laporan, Oracle merekomendasikan Automatic Workload Repository (AWR). AWR memerlukan Oracle Database Enterprise Edition dan lisensi untuk paket Diagnostics and Tuning. Untuk mengaktifkan AWR, atur parameter inisiasi `CONTROL_MANAGEMENT_PACK_ACCESS` ke `DIAGNOSTIC` atau `DIAGNOSTIC+TUNING`.

Bekerja dengan laporan AWR di RDS

Untuk membuat laporan AWR, Anda dapat menjalankan skrip seperti `awrrpt.sql`. Skrip ini diinstal di server host basis data. Di Amazon RDS, Anda tidak memiliki akses langsung ke host. Namun, Anda bisa mendapatkan salinan skrip SQL dari penginstalan Oracle Database lain.

Anda juga dapat menggunakan AWR dengan menjalankan prosedur di paket `SYS.DBMS_WORKLOAD_REPOSITORY` PL/SQL. Anda dapat menggunakan paket ini untuk mengelola baseline dan snapshot, dan juga untuk menampilkan laporan ASH dan AWR. Misalnya, untuk membuat laporan AWR dalam format teks, jalankan prosedur `DBMS_WORKLOAD_REPOSITORY.AWR_REPORT_TEXT`. Namun, Anda tidak dapat mengakses laporan AWR ini dari AWS Management Console.

Saat bekerja dengan AWR, sebaiknya gunakan prosedur `rdsadmin.rdsadmin_diagnostic_util`. Anda dapat menggunakan prosedur ini untuk membuat:

- Laporan AWR
- Laporan Active Session History (ASH)
- Laporan Database Diagnostic Monitor (ADDM)
- File dump Oracle Data Pump Export dari data AWR

Prosedur `rdsadmin_diagnostic_util` menyimpan laporan ke sistem file instans DB. Anda dapat mengakses laporan ini dari konsol. Anda juga dapat mengakses laporan menggunakan prosedur `rdsadmin.rds_file_util` dan Anda dapat mengakses laporan yang disalin ke Amazon S3 menggunakan opsi Integrasi S3. Untuk informasi lebih lanjut, lihat Membaca file di direktori instans DB dan Integrasi Amazon S3.

Anda dapat menggunakan prosedur `rdsadmin_diagnostic_util` di versi mesin DB Amazon RDS for Oracle berikut:

- Semua versi Oracle Database 21c
- 19.0.0.0.ru-2020-04.rur-2020-04.r1 dan versi Oracle Database 19c yang lebih baru
- 12.2.0.1.ru-2020-04.rur-2020-04.r1 dan versi Oracle Database 12c Rilis 2 (12.2) yang lebih baru
- 12.1.0.2.v20 dan versi Oracle Database 12c Rilis 1 (12.1) yang lebih baru

Untuk blog yang menjelaskan cara bekerja dengan laporan diagnostik dalam skenario replikasi, lihat Membuat laporan AWR untuk replika baca Amazon RDS for Oracle.

Parameter umum untuk paket utilitas diagnostik

Biasanya, Anda dapat menggunakan parameter berikut saat mengelola AWR dan ADDM dengan paket `rdsadmin_diagnostic_util`.

Parameter	Tipe data	Defaul	Wajib	Deskripsi
`begin_snap_id`	`NUMBEI`	—	Ya	ID snapshot awal.
`end_snap_id`	`NUMBEI`	—	Ya	ID snapshot akhir.
`dump_directory`	`VARCHA`	`BDUMI`	Tidak	Direktori untuk menulis laporan dan mengekspor file. Jika Anda menentukan direktori nondefault, pengguna yang menjalankan prosedur `rdsadmin_diagnostic_util` harus memiliki izin tulis untuk direktori tersebut.
`p_tag`	`VARCHA`	—	Tidak	String yang dapat digunakan untuk membedakan antara berbagai cadangan guna menunjukkan tujuan atau penggunaan backup, seperti `incremental` atau `daily`. Anda dapat menentukan hingga 30 karakter. Karakter yang valid adalah `a-z`, `A-Z`, `0-9` garis bawah (_), tanda hubung (-), dan titik (.). Tag tidak peka huruf besar/kecil. RMAN selalu menyimpan tanda dalam huruf besar, terlepas dari huruf yang digunakan saat memasukkannya. Tanda tidak harus unik. Jadi, beberapa cadangan dapat memiliki tanda yang sama. Jika Anda tidak menentukan tanda, RMAN akan menetapkan tanda default secara otomatis menggunakan format `TAG`*`YYYYMMDDTHHMMSS`*, dengan *`YYYY`* adalah tahun, *`MM`* adalah bulan, *`DD`* adalah hari, *`HH`* adalah jam (dalam format 24 jam), *`MM`* adalah menit, dan *`SS`* adalah detik. Tanggal dan waktu menunjukkan kapan RMAN memulai pencadangan. Misalnya, cadangan dengan tanda default `TAG20190927T214517` menunjukkan cadangan yang dimulai pada 27-09-2019 pukul 21.45.17.

Parameter	Tipe data	Defaul	Wajib	Deskripsi
				Parameter `p_tag` didukung untuk versi mesin DB RDS for Oracle berikut: • Oracle Database 21c (21.0.0) • Oracle Database 19c (19.0.0), menggunakan 19.0.0.0.ru-2021-10.rur-2021-10.r1 dan yang lebih baru • Oracle Database 12c Rilis 2 (12.2), menggunakan 12.2.0.1.ru-2021-10.r1 dan yang lebih baru • Oracle Database 12c Rilis 1 (12.1), menggunakan 12.1.0.2.V26 dan yang lebih baru
`report_type`	VARCHA	`HTML`	Tidak	Format laporan. Nilai yang valid adalah `TEXT` dan `HTML`.
`dbid`	NUMBEF	—	Tidak	Pengidentifikasi basis data (DBID) valid yang ditunjukkan di tampilan `DBA_HIST_DATABASE_INSTANCE` untuk Oracle. Jika parameter ini tidak ditentukan, RDS akan menggunakan DBID saat ini, yang ditunjukkan di tampilan `V$DATABASE.DBID` .

Anda dapat menggunakan parameter berikut saat mengelola ASH dengan paket rdsadmin_diagnostic_util.

Parameter	Tipe data	Default	Wajib	Deskripsi
`begin_time`	`DATE`	—	Ya	Waktu mulai analisis ASH.
`end_time`	`DATE`	—	Ya	Waktu akhir analisis ASH.
`slot_width`	`NUMBER`	`0`	Tidak	Durasi slot (dalam detik) yang digunakan di bagian "Top Activity" pada laporan ASH. Jika parameter ini tidak ditentukan, interval waktu antara `begin_tim`

Parameter	Tipe data	Default	Wajib	Deskripsi
				e dan end_time menggunakan tidak lebih dari 10 slot.
sid	NUMBER	Null	Tidak	ID sesi.
sql_id	VARCHAR2	Null	Tidak	ID SQL.
wait_class	VARCHAR2	Null	Tidak	Nama kelas tunggu.
service_hash	NUMBER	Null	Tidak	Hash nama layanan.
module_name	VARCHAR2	Null	Tidak	Nama modul.
action_name	VARCHAR2	Null	Tidak	Nama tindakan.
client_id	VARCHAR2	Null	Tidak	ID khusus aplikasi untuk sesi basis data.
plsql_entry	VARCHAR2	Null	Tidak	Titik entri PL/SQL.

Membuat laporan AWR

Untuk membuat laporan AWR, gunakan prosedur `rdsadmin.rdsadmin_diagnostic_util.awr_report`.

Contoh berikut membuat laporan AWR untuk rentang snapshot 101–106. File teks output diberi nama `awrrpt_101_106.txt`. Anda dapat mengakses laporan ini dari AWS Management Console.

```
EXEC rdsadmin.rdsadmin_diagnostic_util.awr_report(101,106,'TEXT');
```

Contoh berikut membuat laporan HTML untuk rentang snapshot 63-65. File HTML output diberi nama `awrrpt_63_65.html`. Prosedur menulis laporan ke direktori basis data nondefault bernama `AWR_RPT_DUMP`.

```
EXEC rdsadmin.rdsadmin_diagnostic_util.awr_report(63,65,'HTML','AWR_RPT_DUMP');
```

Mengekstrak data AWR ke dalam file dump

Untuk mengekstrak data AWR ke file dump, gunakan prosedur `rdsadmin.rdsadmin_diagnostic_util.awr_extract`.

Contoh berikut mengekstrak rentang snapshot 101–106. File dump output diberi nama `awrextract_101_106.dmp`. Anda dapat mengakses file ini melalui konsol.

```
EXEC rdsadmin.rdsadmin_diagnostic_util.awr_extract(101,106);
```

Contoh berikut mengekstrak rentang snapshot 63-65. File dump output diberi nama `awrextract_63_65.dmp`. File disimpan dalam direktori basis data nondefault bernama `AWR_RPT_DUMP`.

```
EXEC rdsadmin.rdsadmin_diagnostic_util.awr_extract(63,65,'AWR_RPT_DUMP');
```

Membuat laporan ADDM

Untuk membuat laporan ADDM, gunakan prosedur `rdsadmin.rdsadmin_diagnostic_util.addm_report`.

Contoh berikut membuat laporan ADDM untuk rentang snapshot 101-106. File teks output diberi nama `addmrpt_101_106.txt`. Anda dapat mengakses laporan ini melalui konsol.

```
EXEC rdsadmin.rdsadmin_diagnostic_util.addm_report(101,106);
```

Contoh berikut membuat laporan ADDM untuk rentang snapshot 63-65. File teks output diberi nama `addmrpt_63_65.txt`. File disimpan dalam direktori basis data nondefault bernama `ADDM_RPT_DUMP`.

```
EXEC rdsadmin.rdsadmin_diagnostic_util.addm_report(63,65,'ADDM_RPT_DUMP');
```

Membuat laporan ASH

Untuk membuat laporan ASH, gunakan prosedur `rdsadmin.rdsadmin_diagnostic_util.ash_report`.

Contoh berikut membuat laporan ASH yang mencakup data dari 14 menit yang lalu hingga saat ini. Nama file output menggunakan format `ashrpt`*`begin_time`*`end_time`.`txt`, dengan *`begin_time`* dan *`end_time`* menggunakan format `YYYYMMDDHH24MISS`. Anda dapat mengakses file ini melalui konsol.

```
BEGIN
    rdsadmin.rdsadmin_diagnostic_util.ash_report(
        begin_time     =>     SYSDATE-14/1440,
        end_time       =>     SYSDATE,
        report_type    =>     'TEXT');
END;
/
```

Contoh berikut membuat laporan ASH yang mencakup data dari 18 November 2019, mulai pukul 18.07 hingga 18.15. Nama laporan output HTML adalah `ashrpt_20190918180700_20190918181500.html`. Laporan disimpan dalam direktori basis data nondefault bernama `AWR_RPT_DUMP`.

```
BEGIN
    rdsadmin.rdsadmin_diagnostic_util.ash_report(
        begin_time     =>    TO_DATE('2019-09-18 18:07:00', 'YYYY-MM-DD HH24:MI:SS'),
        end_time       =>    TO_DATE('2019-09-18 18:15:00', 'YYYY-MM-DD HH24:MI:SS'),
        report_type    =>    'html',
        dump_directory =>    'AWR_RPT_DUMP');
END;
/
```

Mengakses laporan AWR dari konsol atau CLI

Untuk mengakses laporan AWR atau mengekspor file dump, Anda dapat menggunakan atau. AWS Management Console AWS CLI Untuk informasi selengkapnya, lihat Mengunduh file log basis data.

Menyesuaikan tautan basis data untuk penggunaan instans DB di VPC

Untuk menggunakan tautan basis data Oracle dengan instans DB Amazon RDS dalam cloud privat virtual (VPC) yang sama atau di-peering, dua instans DB tersebut harus memiliki rute yang valid di antara keduanya. Pastikan rute yang valid antara instans DB dengan menggunakan tabel perutean VPC Anda dan Daftar Kontrol Akses (ACL).

Grup keamanan dari setiap instans DB harus mengizinkan ingress dan egress dari instans DB lainnya. Aturan masuk dan keluar dapat merujuk ke grup keamanan dari VPC yang sama atau

VPC yang di-peering. Untuk informasi selengkapnya, lihat Memperbarui grup keamanan untuk mereferensikan grup keamanan VPC yang di-peering.

Jika Anda telah mengonfigurasi server DNS kustom menggunakan Set Opsi DHCP di VPC Anda, server DNS kustom Anda harus dapat menyelesaikan nama target tautan basis data. Untuk informasi selengkapnya, lihat Menyiapkan server DNS kustom.

Untuk informasi selengkapnya tentang cara menggunakan tautan basis data dengan Oracle Data Pump, lihat Mengimpor menggunakan Oracle Data Pump.

Mengatur edisi default untuk instans DB

Anda dapat mendefinisikan ulang objek basis data dalam lingkungan privat yang disebut edisi. Anda dapat menggunakan redefenisi berbasis edisi untuk meningkatkan objek basis data aplikasi dengan waktu henti minimal.

Anda dapat mengatur edisi default instans DB Amazon RDS Oracle menggunakan prosedur `rdsadmin.rdsadmin_util.alter_default_edition` Amazon RDS.

Contoh berikut menetapkan edisi default untuk instans DB Amazon RDS Oracle menjadi `RELEASE_V1`.

```
EXEC rdsadmin.rdsadmin_util.alter_default_edition('RELEASE_V1');
```

Contoh berikut menetapkan edisi default untuk laporan instans DB Amazon RDS Oracle kembali ke default Oracle.

```
EXEC rdsadmin.rdsadmin_util.alter_default_edition('ORA$BASE');
```

Untuk informasi lebih lanjut tentang redefinisi berbasis edisi Oracle, lihat About editions and edition-based redefinition di dokumentasi Oracle.

Mengaktifkan audit untuk tabel SYS.AUD$

Untuk mengaktifkan audit di tabel jejak audit basis data `SYS.AUD$`, gunakan prosedur `rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table` Amazon RDS. Properti audit yang didukung hanya `ALL`. Anda tidak dapat melakukan audit atau tidak dapat mengaudit pernyataan atau operasi individual.

Pengaktifan audit didukung untuk instans DB Oracle yang menjalankan versi berikut:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)
- Oracle Database 12c Rilis 2 (12.2)
- Oracle Database 12c Rilis 1 (12.1.0.2.v14) dan yang lebih baru

Prosedur `audit_all_sys_aud_table` memiliki parameter berikut.

Nama parameter	Jenis data	Default	Wajib	Deskripsi
`p_by_access`	boolean	true	Tidak	Atur ke `true` untuk mengaudit `BY ACCESS`. Atur ke `false` untuk mengaudit `BY SESSION`.

> **Note**
>
> Di CDB penghuni tunggal, operasi berikut dapat dijalankan, tetapi tidak ada mekanisme yang dapat dilihat pelanggan yang dapat mendeteksi status operasi saat ini. Informasi audit tidak tersedia dari dalam PDB. Untuk informasi selengkapnya, lihat Batasan CDB RDS for Oracle.

Kueri berikut mengembalikan konfigurasi audit `SYS.AUD$` saat ini untuk basis data.

```
SELECT * FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER='SYS' AND OBJECT_NAME='AUD$';
```

Perintah berikut mengaktifkan audit `ALL` di `SYS.AUD$ BY ACCESS`.

```
EXEC rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table;

EXEC rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table(p_by_access => true);
```

Perintah berikut mengaktifkan audit `ALL` di `SYS.AUD$ BY SESSION`.

```
EXEC rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table(p_by_access => false);
```

Untuk informasi selengkapnya, lihat AUDIT (traditional auditing) di dokumentasi Oracle.

Menonaktifkan pengauditan untuk tabel SYS.AUD$

Untuk menonaktifkan pengauditan di tabel jejak audit basis data `SYS.AUD$`, gunakan prosedur `rdsadmin.rdsadmin_master_util.noaudit_all_sys_aud_table` Amazon RDS. Prosedur ini tidak menggunakan parameter.

Kueri berikut mengembalikan konfigurasi audit `SYS.AUD$` saat ini untuk basis data:

```
SELECT * FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER='SYS' AND OBJECT_NAME='AUD$';
```

Perintah berikut menonaktifkan audit `ALL` di `SYS.AUD$`.

```
EXEC rdsadmin.rdsadmin_master_util.noaudit_all_sys_aud_table;
```

Untuk informasi selengkapnya, lihat NOAUDIT (traditional auditing) di dokumentasi Oracle.

Membersihkan pembuatan indeks online yang terganggu

Untuk membersihkan pembuatan indeks online yang gagal, gunakan prosedur `rdsadmin.rdsadmin_dbms_repair.online_index_clean` Amazon RDS.

Prosedur `online_index_clean` memiliki parameter berikut.

Nama parameter	Jenis data	Default	Wajib	Deskripsi
`object_id`	binary_integer	`ALL_INDEX_ID`	Tidak	ID objek indeks. Biasanya, Anda dapat menggunakan ID objek dari teks kesalahan ORA-08104.
`wait_for_lock`	binary_integer	`rdsadmin.rdsadmin_dbms_repair.lock_wait`	Tidak	Tentukan `rdsadmin.rdsadmin_dbms_repair.lock_wait` default, untuk mencoba mendapatkan kunci pada objek yang mendasari dan mencoba lagi hingga

Nama parameter	Jenis data	Default	Wajib	Deskripsi
				batas internal tercapai jika penguncian gagal. Tentukan `rdsadmin.rdsadmin_dbms_repair.lock_nowait` untuk mencoba mendapatkan penguncian di objek yang mendasari tetapi tidak mencoba lagi jika penguncian gagal.

Contoh berikut membersihkan pembuatan indeks online yang gagal:

```
declare
  is_clean boolean;
begin
  is_clean := rdsadmin.rdsadmin_dbms_repair.online_index_clean(
    object_id     => 1234567890,
    wait_for_lock => rdsadmin.rdsadmin_dbms_repair.lock_nowait
  );
end;
/
```

Untuk informasi selengkapnya, lihat ONLINE_INDEX_CLEAN function di dokumentasi Oracle.

Melewatkan blok yang rusak

Untuk melewati blok yang rusak selama pemindaian indeks dan tabel, gunakan paket `rdsadmin.rdsadmin_dbms_repair`.

Prosedur berikut mencakup fungsionalitas prosedur `sys.dbms_repair.admin_table` dan tidak menggunakan parameter:

- `rdsadmin.rdsadmin_dbms_repair.create_repair_table`
- `rdsadmin.rdsadmin_dbms_repair.create_orphan_keys_table`

- `rdsadmin.rdsadmin_dbms_repair.drop_repair_table`
- `rdsadmin.rdsadmin_dbms_repair.drop_orphan_keys_table`
- `rdsadmin.rdsadmin_dbms_repair.purge_repair_table`
- `rdsadmin.rdsadmin_dbms_repair.purge_orphan_keys_table`

Prosedur berikut menggunakan parameter yang sama seperti parameter dalam paket `DBMS_REPAIR` untuk basis data Oracle:

- `rdsadmin.rdsadmin_dbms_repair.check_object`
- `rdsadmin.rdsadmin_dbms_repair.dump_orphan_keys`
- `rdsadmin.rdsadmin_dbms_repair.fix_corrupt_blocks`
- `rdsadmin.rdsadmin_dbms_repair.rebuild_freelists`
- `rdsadmin.rdsadmin_dbms_repair.segment_fix_status`
- `rdsadmin.rdsadmin_dbms_repair.skip_corrupt_blocks`

Untuk informasi selengkapnya tentang cara menangani kerusakan basis data, lihat DBMS_REPAIR di dokumentasi Oracle.

Example Merespons blok yang rusak

Contoh ini menunjukkan alur kerja dasar untuk merespons blok yang rusak. Langkah-langkah yang diperlukan bergantung pada lokasi dan sifat kerusakan blok Anda.

> **Important**
>
> Sebelum mencoba memperbaiki blok yang rusak, tinjau dokumentasi DBMS_REPAIR dengan cermat.

Untuk melewati blok yang rusak selama pemindaian indeks dan tabel

1. Jalankan prosedur berikut untuk membuat tabel perbaikan jika belum ada.

```
EXEC rdsadmin.rdsadmin_dbms_repair.create_repair_table;
EXEC rdsadmin.rdsadmin_dbms_repair.create_orphan_keys_table;
```

2. Jalankan prosedur berikut untuk memeriksa catatan yang ada dan menghapusnya jika sesuai.

```
SELECT COUNT(*) FROM SYS.REPAIR_TABLE;
SELECT COUNT(*) FROM SYS.ORPHAN_KEY_TABLE;
SELECT COUNT(*) FROM SYS.DBA_REPAIR_TABLE;
SELECT COUNT(*) FROM SYS.DBA_ORPHAN_KEY_TABLE;

EXEC rdsadmin.rdsadmin_dbms_repair.purge_repair_table;
EXEC rdsadmin.rdsadmin_dbms_repair.purge_orphan_keys_table;
```

3. Jalankan prosedur berikut untuk memeriksa adanya blok yang rusak.

```
SET SERVEROUTPUT ON
DECLARE v_num_corrupt INT;
BEGIN
  v_num_corrupt := 0;
  rdsadmin.rdsadmin_dbms_repair.check_object (
    schema_name => '&corruptionOwner',
    object_name => '&corruptionTable',
    corrupt_count =>  v_num_corrupt
  );
  dbms_output.put_line('number corrupt: '||to_char(v_num_corrupt));
END;
/

COL CORRUPT_DESCRIPTION FORMAT a30
COL REPAIR_DESCRIPTION FORMAT a30

SELECT OBJECT_NAME, BLOCK_ID, CORRUPT_TYPE, MARKED_CORRUPT,
       CORRUPT_DESCRIPTION, REPAIR_DESCRIPTION
FROM   SYS.REPAIR_TABLE;

SELECT SKIP_CORRUPT
FROM   DBA_TABLES
WHERE  OWNER = '&corruptionOwner'
AND    TABLE_NAME = '&corruptionTable';
```

4. Gunakan prosedur `skip_corrupt_blocks` untuk mengaktifkan atau menonaktifkan pelompatan kerusakan untuk tabel yang terkena dampak. Bergantung pada situasi, Anda mungkin juga perlu mengekstrak data ke tabel baru, lalu meletakkan tabel yang berisi blok yang rusak.

   Gunakan prosedur berikut untuk mengaktifkan pelompatan kerusakan untuk tabel yang terkena dampak.

```
begin
  rdsadmin.rdsadmin_dbms_repair.skip_corrupt_blocks (
    schema_name => '&corruptionOwner',
    object_name => '&corruptionTable',
    object_type => rdsadmin.rdsadmin_dbms_repair.table_object,
    flags => rdsadmin.rdsadmin_dbms_repair.skip_flag);
end;
/
select skip_corrupt from dba_tables where owner = '&corruptionOwner' and table_name
 = '&corruptionTable';
```

Gunakan prosedur berikut untuk menonaktifkan pelompatan kerusakan.

```
begin
  rdsadmin.rdsadmin_dbms_repair.skip_corrupt_blocks (
    schema_name => '&corruptionOwner',
    object_name => '&corruptionTable',
    object_type => rdsadmin.rdsadmin_dbms_repair.table_object,
    flags => rdsadmin.rdsadmin_dbms_repair.noskip_flag);
end;
/

select skip_corrupt from dba_tables where owner = '&corruptionOwner' and table_name
 = '&corruptionTable';
```

5. Setelah menyelesaikan semua pekerjaan perbaikan, jalankan prosedur berikut untuk melakukan tabel perbaikan.

```
EXEC rdsadmin.rdsadmin_dbms_repair.drop_repair_table;
EXEC rdsadmin.rdsadmin_dbms_repair.drop_orphan_keys_table;
```

Mengubah ukuran tablespace, file data, dan file temp

Secara default, tablespace Oracle dibuat dengan mengaktifkan ekstensi otomatis dan tidak ada ukuran maksimum. Karena pengaturan default ini, tablespace terkadang dapat bertambah terlalu besar. Sebaiknya tentukan ukuran maksimum yang sesuai pada tablespace permanen dan sementara, serta pantau penggunaan ruang dengan cermat.

Mengubah ukuran tablespace permanen

Untuk mengubah ukuran tablespace permanen dalam instans DB RDS for Oracle, gunakan salah satu prosedur Amazon RDS berikut:

- `rdsadmin.rdsadmin_util.resize_datafile`
- `rdsadmin.rdsadmin_util.autoextend_datafile`

Prosedur `resize_datafile` memiliki parameter berikut.

Nama parameter	Jenis data	Default	Wajib	Deskripsi
`p_data_file_id`	number	—	Ya	Pengidentifikasi file data yang ukurannya akan diubah.
`p_size`	varchar2	—	Ya	Ukuran file data. Tentukan ukuran dalam byte (default), kilobyte (K), megabyte (M), atau gigabyte (G).

Prosedur `autoextend_datafile` memiliki parameter berikut.

Nama parameter	Jenis data	Default	Wajib	Deskripsi
`p_data_file_id`	number	—	Ya	Pengidentifikasi file data yang ukurannya akan diubah.
`p_autoextend_state`	varchar2	—	Ya	Status fitur ekstensi otomatis. Tentukan `ON` untuk memperluas file data secara otomatis dan `OFF` untuk menonaktifkan ekstensi otomatis.

Nama parameter	Jenis data	Default	Wajib	Deskripsi
p_next	varchar2	—	Tidak	Ukuran penambahan file data berikutnya. Tentukan ukuran dalam byte (default), kilobyte (K), megabyte (M), atau gigabyte (G).
p_maxsize	varchar2	—	Tidak	Ruang disk maksimum yang diizinkan untuk ekstensi otomatis. Tentukan ukuran dalam byte (default), kilobyte (K), megabyte (M), atau gigabyte (G). Anda dapat menentukan UNLIMITED untuk menghapus batas ukuran file.

Contoh berikut mengubah ukuran file data 4 hingga 500 MB.

```
EXEC rdsadmin.rdsadmin_util.resize_datafile(4,'500M');
```

Contoh berikut menonaktifkan ekstensi otomatis untuk file data 4. Hal ini juga mengaktifkan ekstensi otomatis untuk file data 5, dengan penambahan 128 MB dan tidak ada ukuran maksimum.

```
EXEC rdsadmin.rdsadmin_util.autoextend_datafile(4,'OFF');
EXEC rdsadmin.rdsadmin_util.autoextend_datafile(5,'ON','128M','UNLIMITED');
```

Mengubah ukuran tablespace sementara

Untuk mengubah ukuran tablespace sementara dalam instans DB RDS for Oracle, termasuk replika baca, gunakan salah satu prosedur Amazon RDS berikut:

- rdsadmin.rdsadmin_util.resize_temp_tablespace
- rdsadmin.rdsadmin_util.resize_tempfile

- `rdsadmin.rdsadmin_util.autoextend_tempfile`

Prosedur `resize_temp_tablespace` memiliki parameter berikut.

Nama parameter	Jenis data	Default	Wajib	Deskripsi
`p_temp_tablespace_name`	varchar2	—	Ya	Nama tablespace sementara yang ukurannya akan diubah.
`p_size`	varchar2	—	Ya	Ukuran tablespace. Tentukan ukuran dalam byte (default), kilobyte (K), megabyte (M), atau gigabyte (G).

Prosedur `resize_tempfile` memiliki parameter berikut.

Nama parameter	Jenis data	Default	Wajib	Deskripsi
`p_temp_file_id`	number	—	Ya	Pengidentifikasi file temp yang ukurannya akan diubah.
`p_size`	varchar2	—	Ya	Ukuran file temp. Tentukan ukuran dalam byte (default), kilobyte (K), megabyte (M), atau gigabyte (G).

Prosedur `autoextend_tempfile` memiliki parameter berikut.

Nama parameter	Jenis data	Default	Wajib	Deskripsi
p_temp_file_id	number	—	Ya	Pengidentifikasi file temp yang ukurannya akan diubah.
p_autoextend_state	varchar2	—	Ya	Status fitur ekstensi otomatis. Tentukan ON untuk memperluas file temp secara otomatis dan OFF untuk menonaktifkan ekstensi otomatis.
p_next	varchar2	—	Tidak	Ukuran penambahan file temp berikutnya. Tentukan ukuran dalam byte (default), kilobyte (K), megabyte (M), atau gigabyte (G).
p_maxsize	varchar2	—	Tidak	Ruang disk maksimum yang diizinkan untuk ekstensi otomatis. Tentukan ukuran dalam byte (default), kilobyte (K), megabyte (M), atau gigabyte (G). Anda dapat menentukan UNLIMITED untuk menghapus batas ukuran file.

Contoh berikut mengubah ukuran tablespace sementara bernama TEMP menjadi 4 GB.

```
EXEC rdsadmin.rdsadmin_util.resize_temp_tablespace('TEMP','4G');
```

```
EXEC rdsadmin.rdsadmin_util.resize_temp_tablespace('TEMP','4096000000');
```

Contoh berikut mengubah ukuran tablespace sementara berdasarkan file temp dengan pengidentifikasi file 1 menjadi 2 MB.

```
EXEC rdsadmin.rdsadmin_util.resize_tempfile(1,'2M');
```

Contoh berikut menonaktifkan ekstensi otomatis untuk file temp 1. Contoh ini juga menetapkan ukuran ekstensi otomatis maksimum file temp 2 hingga 10 GB, dengan penambahan 100 MB.

```
EXEC rdsadmin.rdsadmin_util.autoextend_tempfile(1,'OFF');
EXEC rdsadmin.rdsadmin_util.autoextend_tempfile(2,'ON','100M','10G');
```

Untuk informasi selengkapnya tentang replika baca untuk instans DB Oracle, lihat Menggunakan replika baca untuk Amazon RDS for Oracle.

Membersihkan keranjang sampah

Ketika Anda menghapus tabel, Oracle Database Anda tidak langsung menghapus ruang penyimpanan. Basis data mengubah nama tabel dan menempatkannya dan objek terkait apa pun di keranjang sampah. Membersihkan keranjang sampah akan menghapus item ini dan membersihkan ruang penyimpanannya.

Untuk membersihkan keranjang sampah sepenuhnya, gunakan prosedur `rdsadmin.rdsadmin_util.purge_dba_recyclebin` Amazon RDS. Namun, prosedur ini tidak dapat membersihkan keranjang sampah objek `SYS` dan `RDSADMIN`. Jika Anda perlu membersihkan objek ini, hubungi Dukungan AWS .

Contoh berikut membersihkan keranjang sampah sepenuhnya.

```
EXEC rdsadmin.rdsadmin_util.purge_dba_recyclebin;
```

Mengatur nilai default yang ditampilkan untuk redaksi penuh

Untuk mengubah nilai default yang ditampilkan untuk redaksi penuh di instans Amazon RDS Oracle Anda, gunakan prosedur `rdsadmin.rdsadmin_util.dbms_redact_upd_full_rdct_val` Amazon RDS. Perhatikan bahwa Anda membuat kebijakan redaksi dengan paket `DBMS_REDACT` PL/SQL, seperti yang dijelaskan di dokumentasi Oracle Database. Prosedur `dbms_redact_upd_full_rdct_val` menentukan karakter yang akan ditampilkan untuk berbagai tipe data yang dipengaruhi oleh kebijakan yang ada.

Prosedur dbms_redact_upd_full_rdct_val memiliki parameter berikut.

Nama parameter	Jenis data	Default	Wajib	Deskripsi
p_number_val	number	Null	Tidak	Memodifikasi nilai default untuk kolom tipe data NUMBER.
p_binfloat_val	binary_float	Null	Tidak	Memodifikasi nilai default untuk kolom tipe data BINARY_FLOAT .
p_bindouble_val	binary_double	Null	Tidak	Memodifikasi nilai default untuk kolom tipe data BINARY_DOUBLE .
p_char_val	char	Null	Tidak	Memodifikasi nilai default untuk kolom tipe data CHAR.
p_varchar_val	varchar2	Null	Tidak	Memodifikasi nilai default untuk kolom tipe data VARCHAR2.
p_nchar_val	nchar	Null	Tidak	Memodifikasi nilai default untuk kolom tipe data NCHAR.
p_nvarchar_val	nvarchar2	Null	Tidak	Memodifikasi nilai default untuk kolom tipe data NVARCHAR2 .
p_date_val	date	Null	Tidak	Memodifikasi nilai default untuk kolom tipe data DATE.
p_ts_val	timestamp	Null	Tidak	Memodifikasi nilai default untuk kolom tipe data TIMESTAMP .

Nama parameter	Jenis data	Default	Wajib	Deskripsi
p_tswtz_val	timestamp with time zone	Null	Tidak	Memodifikasi nilai default untuk kolom tipe data TIMESTAMP WITH TIME ZONE.
p_blob_val	blob	Null	Tidak	Memodifikasi nilai default untuk kolom tipe data BLOB.
p_clob_val	clob	Null	Tidak	Memodifikasi nilai default untuk kolom tipe data CLOB.
p_nclob_val	nclob	Null	Tidak	Memodifikasi nilai default untuk kolom tipe data NCLOB.

Contoh berikut mengubah nilai default yang disunting menjadi * untuk tipe data CHAR:

```
EXEC rdsadmin.rdsadmin_util.dbms_redact_upd_full_rdct_val(p_char_val => '*');
```

Contoh berikut mengubah nilai default yang disunting untuk NUMBER, DATE, dan tipe data CHAR:

```
BEGIN
rdsadmin.rdsadmin_util.dbms_redact_upd_full_rdct_val(
    p_number_val=>1,
    p_date_val=>to_date('1900-01-01','YYYY-MM-DD'),
    p_varchar_val=>'X');
END;
/
```

Setelah Anda mengubah nilai default untuk redaksi penuh dengan prosedur dbms_redact_upd_full_rdct_val, boot ulang instans DB Anda agar perubahan dapat diterapkan. Untuk informasi selengkapnya, lihat Mem-boot ulang instans DB.

Melakukan tugas umum terkait log untuk instans DB Oracle

Setelah itu, Anda dapat menemukan cara melakukan tugas DBA umum tertentu terkait proses logging di instans DB Amazon RDS Anda yang menjalankan Oracle. Untuk memberikan pengalaman layanan terkelola, Amazon RDS tidak memberikan akses shell ke instans DB, dan membatasi akses ke sejumlah prosedur dan tabel sistem tertentu yang memerlukan hak istimewa tingkat lanjut.

Untuk informasi selengkapnya, lihat File log basis data Oracle.

Topik

Mengatur logging paksa

Dalam mode logging paksa, Oracle menge-log semua perubahan pada database kecuali perubahan dalam ruang tabel sementara dan segmen sementara (`NOLOGGING` klausul diabaikan). Untuk informasi selengkapnya, lihat Menentukan mode LOGGING PAKSA di dokumentasi Oracle.

Untuk mengatur logging paksa, gunakan prosedur `rdsadmin.rdsadmin_util.force_logging` Amazon RDS. Prosedur `force_logging` memiliki parameter berikut.

Nama parameter	Tipe data	Default	Ya	Deskripsi
`p_enable`	boolean	true	Tidak	Atur ke `true` untuk menempatkan basis data dalam mode logging paksa, `false` untuk

Nama parameter	Tipe data	Default	Ya	Deskripsi
				menghapus basis data dari mode logging paksa.

Contoh berikut menempatkan basis data dalam mode logging paksa.

```
EXEC rdsadmin.rdsadmin_util.force_logging(p_enable => true);
```

Mengatur logging tambahan

Jika Anda mengaktifkan logging tambahan, LogMiner memiliki informasi yang diperlukan untuk mendukung baris berantai dan tabel klaster. Untuk informasi selengkapnya, lihat Logging tambahan di dokumentasi Oracle.

Oracle Database tidak mengaktifkan logging tambahan secara default. Untuk mengaktifkan dan menonaktifkan logging tambahan, gunakan prosedur `rdsadmin.rdsadmin_util.alter_supplemental_logging` Amazon RDS. Untuk informasi selengkapnya tentang cara Amazon RDS mengelola retensi log pengulangan yang diarsipkan untuk instans DB Oracle, lihat Mempertahankan log pengulangan yang diarsipkan.

Prosedur `alter_supplemental_logging` memiliki parameter berikut.

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
`p_action`	varchar2	—	Ya	`'ADD'` untuk menambahkan logging tambahan, `'DROP'` untuk menghapus logging tambahan.
`p_type`	varchar2	null	Tidak	Tipe logging tambahan. Nilai yang valid adalah `'ALL'`, `'FOREIGN KEY'`, `'PRIMARY KEY'`, `'UNIQUE'`, atau `PROCEDURAL` .

Contoh berikut mengaktifkan logging tambahan.

```
begin
    rdsadmin.rdsadmin_util.alter_supplemental_logging(
        p_action => 'ADD');
end;
/
```

Contoh berikut mengaktifkan logging tambahan untuk semua kolom ukuran maksimum dengan panjang tetap.

```
begin
    rdsadmin.rdsadmin_util.alter_supplemental_logging(
        p_action => 'ADD',
        p_type   => 'ALL');
end;
/
```

Contoh berikut memungkinkan logging tambahan untuk kolom kunci primer.

```
begin
    rdsadmin.rdsadmin_util.alter_supplemental_logging(
        p_action => 'ADD',
        p_type   => 'PRIMARY KEY');
end;
/
```

Mengganti file log online

Untuk beralih file log, gunakan prosedur `rdsadmin.rdsadmin_util.switch_logfile` Amazon RDS. Prosedur `switch_logfile` tidak memiliki parameter.

Contoh berikut mengganti file log.

```
EXEC rdsadmin.rdsadmin_util.switch_logfile;
```

Menambahkan log pengulangan online

Instans DB Amazon RDS yang menjalankan Oracle dimulai dengan empat log pengulangan online, yang masing-masing berukuran 128 MB. Untuk menambahkan log pengulangan tambahan, gunakan prosedur `rdsadmin.rdsadmin_util.add_logfile` Amazon RDS.

Prosedur `add_logfile` memiliki parameter berikut.

> **Note**
>
> Parameter ini tidak bisa ada pada saat yang sama.

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
`bytes`	positif	null	Tidak	Ukuran file log dalam byte.
`p_size`	varchar2	—	Ya	Ukuran file log. Anda dapat menentukan ukuran dalam kilobyte (K), megabyte (M), atau gigabyte (G).

Perintah berikut menambahkan file log 100 MB.

```
EXEC rdsadmin.rdsadmin_util.add_logfile(p_size => '100M');
```

Menghapus log pengulangan online

Untuk menghapus log pengulangan, gunakan prosedur `rdsadmin.rdsadmin_util.drop_logfile` Amazon RDS. Prosedur `drop_logfile` memiliki parameter berikut.

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
`grp`	positif	—	Ya	Nomor grup log.

Contoh berikut menghapus log dengan grup nomor 3.

```
EXEC rdsadmin.rdsadmin_util.drop_logfile(grp => 3);
```

Anda hanya dapat menghapus log yang memiliki status tidak digunakan atau tidak aktif. Contoh berikut mendapatkan status log.

```
SELECT GROUP#, STATUS FROM V$LOG;

GROUP#     STATUS
---------- ----------------
1          CURRENT
2          INACTIVE
3          INACTIVE
4          UNUSED
```

Mengubah ukuran log pengulangan online

Instans DB Amazon RDS yang menjalankan Oracle dimulai dengan empat log pengulangan online, yang masing-masing berukuran 128 MB. Contoh berikut menunjukkan cara menggunakan prosedur Amazon RDS untuk mengubah ukuran log Anda dari 128 MB menjadi 512 MB untuk masing-masing.

```
/* Query V$LOG to see the logs.          */
/* You start with 4 logs of 128 MB each. */

SELECT GROUP#, BYTES, STATUS FROM V$LOG;

GROUP#     BYTES      STATUS
---------- ---------- ----------------
1          134217728  INACTIVE
2          134217728  CURRENT
3          134217728  INACTIVE
4          134217728  INACTIVE


/* Add four new logs that are each 512 MB */

EXEC rdsadmin.rdsadmin_util.add_logfile(bytes => 536870912);
EXEC rdsadmin.rdsadmin_util.add_logfile(bytes => 536870912);
EXEC rdsadmin.rdsadmin_util.add_logfile(bytes => 536870912);
EXEC rdsadmin.rdsadmin_util.add_logfile(bytes => 536870912);


/* Query V$LOG to see the logs. */
/* Now there are 8 logs.        */
```

```
SELECT GROUP#, BYTES, STATUS FROM V$LOG;

GROUP#     BYTES      STATUS
---------- ---------- ----------------
1          134217728  INACTIVE
2          134217728  CURRENT
3          134217728  INACTIVE
4          134217728  INACTIVE
5          536870912  UNUSED
6          536870912  UNUSED
7          536870912  UNUSED
8          536870912  UNUSED


/* Drop each inactive log using the group number. */

EXEC rdsadmin.rdsadmin_util.drop_logfile(grp => 1);
EXEC rdsadmin.rdsadmin_util.drop_logfile(grp => 3);
EXEC rdsadmin.rdsadmin_util.drop_logfile(grp => 4);


/* Query V$LOG to see the logs. */
/* Now there are 5 logs.        */

select GROUP#, BYTES, STATUS from V$LOG;

GROUP#     BYTES      STATUS
---------- ---------- ----------------
2          134217728  CURRENT
5          536870912  UNUSED
6          536870912  UNUSED
7          536870912  UNUSED
8          536870912  UNUSED


/* Switch logs so that group 2 is no longer current. */

EXEC rdsadmin.rdsadmin_util.switch_logfile;


/* Query V$LOG to see the logs.        */
/* Now one of the new logs is current. */

SQL>SELECT GROUP#, BYTES, STATUS FROM V$LOG;
```

```
GROUP#     BYTES      STATUS
---------- ---------- ----------------
2          134217728  ACTIVE
5          536870912  CURRENT
6          536870912  UNUSED
7          536870912  UNUSED
8          536870912  UNUSED


/* If the status of log 2 is still "ACTIVE", issue a checkpoint to clear it to
 "INACTIVE".  */

EXEC rdsadmin.rdsadmin_util.checkpoint;


/* Query V$LOG to see the logs.            */
/* Now the final original log is inactive. */

select GROUP#, BYTES, STATUS from V$LOG;

GROUP#     BYTES      STATUS
---------- ---------- ----------------
2          134217728  INACTIVE
5          536870912  CURRENT
6          536870912  UNUSED
7          536870912  UNUSED
8          536870912  UNUSED


# Drop the final inactive log.

EXEC rdsadmin.rdsadmin_util.drop_logfile(grp => 2);


/* Query V$LOG to see the logs.    */
/* Now there are four 512 MB logs. */

SELECT GROUP#, BYTES, STATUS FROM V$LOG;

GROUP#     BYTES      STATUS
---------- ---------- ----------------
5          536870912  CURRENT
6          536870912  UNUSED
```

```
7          536870912  UNUSED
8          536870912  UNUSED
```

Mempertahankan log pengulangan yang diarsipkan

Anda dapat mempertahankan log pengulangan yang diarsipkan secara lokal di instans DB untuk digunakan dengan produk seperti Oracle LogMiner (`DBMS_LOGMNR`). Setelah mempertahankan log pengulangan, Anda dapat menggunakan LogMiner untuk menganalisis log. Untuk informasi selengkapnya, lihat Menggunakan LogMiner untuk menganalisis file log pengulangan di dokumentasi Oracle.

Untuk mempertahankan log pengulangan yang diarsipkan, terapkan prosedur `rdsadmin.rdsadmin_util.set_configuration` Amazon RDS. Prosedur `set_configuration` memiliki parameter berikut.

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
`name`	varchar	—	Ya	Nama konfigurasi yang akan diperbarui.
`value`	varchar	—	Ya	Nilai untuk konfigurasi.

Contoh berikut mempertahankan log pengulangan 24 jam.

```
begin
    rdsadmin.rdsadmin_util.set_configuration(
        name  => 'archivelog retention hours',
        value => '24');
end;
/
commit;
```

> **Note**
>
> Komitmen tersebut diperlukan agar perubahan dapat berlaku.

Untuk melihat berapa lama log pengulangan yang diarsipkan dipertahankan untuk instans DB Anda, gunakan prosedur `rdsadmin.rdsadmin_util.show_configuration` Amazon RDS.

Contoh berikut menunjukkan waktu retensi log.

```
set serveroutput on
EXEC rdsadmin.rdsadmin_util.show_configuration;
```

Output menunjukkan pengaturan saat ini untuk `archivelog retention hours`. Output berikut menunjukkan bahwa log pengulangan yang diarsipkan dipertahankan selama 48 jam.

```
NAME:archivelog retention hours
VALUE:48
DESCRIPTION:ArchiveLog expiration specifies the duration in hours before archive/redo
 log files are automatically deleted.
```

Karena log pengulangan yang diarsipkan dipertahankan dalam instans DB Anda, pastikan bahwa instans DB Anda memiliki penyimpanan yang cukup untuk log yang dipertahankan. Untuk menentukan seberapa besar ruang yang digunakan oleh instans DB Anda dalam X jam terakhir, Anda dapat menjalankan kueri berikut, mengganti X dengan jumlah jam.

```
SELECT SUM(BLOCKS * BLOCK_SIZE) bytes
  FROM V$ARCHIVED_LOG
 WHERE FIRST_TIME >= SYSDATE-(X/24) AND DEST_ID=1;
```

RDS for Oracle hanya menghasilkan log pengulangan yang diarsipkan jika periode retensi cadangan instans DB Anda lebih besar dari nol. Secara default periode retensi cadangan lebih besar dari nol.

Ketika periode retensi log yang diarsipkan berakhir, RDS for Oracle akan menghapus log pengulangan yang diarsipkan dari instans DB Anda. Untuk mendukung pemulihan instans DB Anda ke titik waktu tertentu, Amazon RDS menyimpan log pengulangan yang diarsipkan di luar instans DB Anda berdasarkan periode retensi cadangan. Untuk mengubah periode retensi cadangan, lihat Memodifikasi instans DB Amazon RDS.

> **Note**
>
> Dalam beberapa kasus, Anda mungkin menggunakan JDBC di Linux untuk mengunduh log pengulangan yang diarsipkan dan mengalami waktu latensi panjang dan pengaturan ulang koneksi. Dalam kasus seperti itu, masalah mungkin disebabkan oleh pengaturan generator nomor acak default pada klien Java Anda. Sebaiknya setel driver JDBC Anda untuk menggunakan generator nomor acak yang tidak blokir.

Mengakses log pengulangan online dan yang diarsipkan

Anda mungkin ingin mengakses file log pengulangan online dan yang diarsipkan untuk melakukan penambangan dengan alat eksternal seperti GoldenGate, Attunity, Informatica, dan lainnya. Untuk mengakses file ini, lakukan hal berikut:

1. Buat objek direktori yang memberikan akses hanya baca ke jalur file fisik.

   Gunakan `rdsadmin.rdsadmin_master_util.create_archivelog_dir` dan `rdsadmin.rdsadmin_master_util.create_onlinelog_dir`.
2. Baca file menggunakan PL/SQL.

   Anda dapat membaca file menggunakan PL/SQL. Untuk informasi selengkapnya tentang cara membaca file dari objek direktori, lihat Membuat daftar file di direktori instans DB dan Membaca file di direktori instans DB.

Mengakses log transaksi didukung untuk rilis berikut ini:

- Oracle Database 21c
- Oracle Database 19c
- Oracle Database 12c Rilis 2 (12.2.0.1)
- Oracle Database 12c Rilis 1 (12.1)

Kode berikut membuat direktori yang memberikan akses baca saja ke file log pengulangan online dan yang diarsipkan.

> **Important**
>
> Kode ini juga mencabut hak istimewa `DROP ANY DIRECTORY`.

```
EXEC rdsadmin.rdsadmin_master_util.create_archivelog_dir;
EXEC rdsadmin.rdsadmin_master_util.create_onlinelog_dir;
```

Kode berikut menghapus direktori untuk file log pengulangan online dan yang diarsipkan.

```
EXEC rdsadmin.rdsadmin_master_util.drop_archivelog_dir;
```

```
EXEC rdsadmin.rdsadmin_master_util.drop_onlinelog_dir;
```

Kode berikut memberikan dan mencabut hak istimewa `DROP ANY DIRECTORY`.

```
EXEC rdsadmin.rdsadmin_master_util.revoke_drop_any_directory;
EXEC rdsadmin.rdsadmin_master_util.grant_drop_any_directory;
```

Mengunduh log pengulangan yang diarsipkan dari Amazon S3

Anda dapat mengunduh log pengulangan yang diarsipkan pada instans DB Anda menggunakan paket `rdsadmin.rdsadmin_archive_log_download`. Jika log pengulangan yang diarsipkan tidak lagi tersedia di instans DB, Anda dapat mengunduhnya lagi dari Amazon S3. Kemudian Anda dapat menambang log atau menggunakannya untuk memulihkan atau mereplikasi basis data Anda.

> **Note**
>
> Anda tidak dapat mengunduh log pengulangan yang diarsipkan pada instans replika baca.

Mengunduh log pengulangan yang diarsipkan: langkah-langkah dasar

Ketersediaan log pengulangan yang diarsipkan bergantung pada kebijakan retensi berikut:

- Kebijakan retensi cadangan - Log dalam kebijakan ini tersedia di Amazon S3. Log di luar kebijakan ini akan dihapus.
- Kebijakan penyimpanan log yang diarsipkan - Log dalam kebijakan ini tersedia pada instans DB Anda. Log di luar kebijakan ini akan dihapus.

Jika log tidak tersedia di instans Anda tetapi dilindungi oleh periode retensi cadangan, gunakan `rdsadmin.rdsadmin_archive_log_download` untuk mengunduhnya lagi. RDS for Oracle menyimpan log ke direktori `/rdsdbdata/log/arch` pada instans DB Anda.

Untuk mengunduh log pengulangan yang diarsipkan dari Amazon S3

1. Konfigurasikan periode retensi Anda untuk memastikan unduhan log pengulangan yang diarsipkan dipertahankan selama yang Anda butuhkan. Pastikan untuk `COMMIT` perubahan Anda.

RDS mempertahankan log yang Anda unduh sesuai dengan kebijakan retensi log yang diarsipkan, mulai dari saat log diunduh. Untuk mempelajari cara menetapkan kebijakan retensi, lihat Mempertahankan log pengulangan yang diarsipkan.

2. Tunggu hingga 5 menit agar perubahan kebijakan retensi log yang diarsipkan diterapkan.
3. Unduh log pengulangan yang diarsipkan dari Amazon S3 menggunakan `rdsadmin.rdsadmin_archive_log_download`.

   Untuk informasi selengkapnya, lihat Mengunduh satu log pengulangan yang diarsipkan dan Mengunduh serangkaian log pengulangan yang diarsipkan.

   > **Note**
   >
   > RDS secara otomatis memeriksa penyimpanan yang tersedia sebelum mengunduh. Jika log yang diminta memakan ruang dengan persentase tinggi, Anda akan menerima peringatan.

4. Konfirmasi bahwa log berhasil diunduh dari Amazon S3.

   Anda dapat melihat status tugas unduhan Anda di file bdump. File bdump memiliki nama jalur `/rdsdbdata/log/trace/dbtask-`*`task-id`*`.log`. Pada langkah unduhan sebelumnya, Anda menjalankan pernyataan `SELECT` yang menampilkan ID tugas dalam tipe data `VARCHAR2`. Untuk informasi selengkapnya, lihat contoh serupa di Memantau status transfer file.

Mengunduh satu log pengulangan yang diarsipkan

Untuk mengunduh satu log pengulangan yang diarsipkan ke direktori `/rdsdbdata/log/arch`, gunakan `rdsadmin.rdsadmin_archive_log_download.download_log_with_seqnum`. Prosedur ini memiliki parameter berikut.

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
`seqnum`	number	—	Ya	Nomor urut log pengulangan yang diarsipkan.

Contoh berikut mengunduh log dengan nomor urut 20.

```
SELECT rdsadmin.rdsadmin_archive_log_download.download_log_with_seqnum(seqnum => 20)
       AS TASK_ID
FROM   DUAL;
```

Mengunduh serangkaian log pengulangan yang diarsipkan

Untuk mengunduh serangkaian log pengulangan yang diarsipkan ke direktori `/rdsdbdata/log/arch`, gunakan `download_logs_in_seqnum_range`. Unduhan Anda dibatasi hingga 300 log per permintaan. Prosedur `download_logs_in_seqnum_range` memiliki parameter berikut.

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
`start_seq`	number	—	Ya	Nomor urut awal untuk rangkaian.
`end_seq`	number	—	Ya	Nomor urut akhir untuk rangkaian.

Contoh berikut mengunduh log dengan nomor urut 50 sampai 100.

```
SELECT rdsadmin.rdsadmin_archive_log_download.download_logs_in_seqnum_range(start_seq
 => 50, end_seq => 100)
       AS TASK_ID
FROM   DUAL;
```

Melakukan tugas RMAN umum untuk instans DB Oracle

Di bagian berikut ini, Anda dapat menemukan cara melakukan tugas DBA Oracle Recovery Manager (RMAN) di instans DB Amazon RDS yang menjalankan Oracle. Untuk memberikan pengalaman layanan terkelola, Amazon RDS tidak memberikan akses shell ke instans DB. Layanan ini juga membatasi akses ke prosedur dan tabel sistem tertentu yang memerlukan hak istimewa tingkat lanjut.

Gunakan paket Amazon RDS `rdsadmin.rdsadmin_rman_util` untuk melakukan pencadangan RMAN untuk basis data Amazon RDS for Oracle ke disk. Paket `rdsadmin.rdsadmin_rman_util` mendukung pencadangan file basis data secara penuh dan bertahap, pencadangan ruang tabel, dan pencadangan log pengulangan yang diarsipkan.

Setelah pencadangan RMAN selesai, Anda dapat menyalin file cadangan dari host instans DB Amazon RDS for Oracle. Anda dapat melakukan tindakan ini dengan tujuan melakukan pemulihan ke

host non-RDS atau untuk penyimpanan cadangan jangka panjang. Misalnya, Anda dapat menyalin file cadangan ke bucket Amazon S3. Untuk informasi selengkapnya, lihat cara menggunakan Integrasi Amazon S3.

File cadangan untuk pencadangan RMAN tetap berada di host instans DB Amazon RDS hingga Anda menghapusnya secara manual. Anda dapat menggunakan prosedur Oracle `UTL_FILE.FREMOVE` untuk menghapus file dari direktori. Untuk informasi selengkapnya, lihat FREMOVE procedure di dokumentasi Oracle Database.

Anda tidak dapat menggunakan RMAN untuk memulihkan instans DB RDS for Oracle. Namun, Anda dapat menggunakan RMAN untuk memulihkan cadangan ke instans EC2 Amazon atau on-premise. Untuk informasi selengkapnya, lihat artikel blog Memulihkan instans Amazon RDS for Oracle ke instans yang dikelola sendiri.

> **Note**
>
> Untuk membuat cadangan dan memulihkan ke instans DB Amazon RDS for Oracle, Anda dapat terus menggunakan fitur pencadangan dan pemulihan Amazon RDS. Untuk informasi selengkapnya, lihat Mencadangkan, memulihkan, dan mengekspor data.

Topik

Prasyarat untuk pencadangan RMAN

Sebelum membuat cadangan basis data menggunakan paket `rdsadmin.rdsadmin_rman_util`, pastikan Anda memenuhi prasyarat berikut:

- Pastikan bahwa basis data RDS for Oracle Anda berada dalam mode `ARCHIVELOG`. Untuk mengaktifkan mode ini, atur periode retensi pencadangan ke nilai bukan nol.
- Saat membuat cadangan log pengulangan yang diarsipkan atau menjalankan pencadangan penuh atau bertahap yang mencakup log pengulangan yang diarsipkan, dan saat membuat cadangan basis data, pastikan retensi log pengulangan diatur ke nilai bukan nol. Log pengulangan yang diarsipkan diperlukan untuk membuat file basis data konsisten selama pemulihan. Untuk informasi selengkapnya, lihat Mempertahankan log pengulangan yang diarsipkan.
- Pastikan instans DB Anda memiliki ruang kosong yang cukup untuk menyimpan cadangan. Saat membuat cadangan basis data, Anda menentukan objek direktori Oracle sebagai parameter dalam panggilan prosedur. RMAN menempatkan file di direktori yang ditentukan. Anda dapat menggunakan direktori default, seperti `DATA_PUMP_DIR`, atau membuat direktori baru. Untuk informasi selengkapnya, lihat Membuat dan menghapus direktori di ruang penyimpanan data utama.

  Anda dapat memantau ruang kosong saat ini dalam RDS untuk instance Oracle menggunakan metrik. CloudWatch `FreeStorageSpace` Sebaiknya siapkan ruang kosong yang melebihi ukuran basis data saat ini, meskipun RMAN hanya mencadangkan blok yang diformat dan mendukung kompresi.

Parameter umum untuk prosedur RMAN

Anda dapat menggunakan prosedur di paket `rdsadmin.rdsadmin_rman_util` Amazon RDS untuk melakukan tugas dengan RMAN. Beberapa parameter umum untuk prosedur dalam paket. Paket memiliki parameter umum berikut.

Nama parameter	Tipe data	Nilai valid	Defaul	Diperlu n	Deskripsi
`p_directory_name`	varchar	Nama direktori basis data yang valid.	—	Ya	Nama direktori yang memuat file cadangan.

Nama parameter	Tipe data	Nilai valid	Defaul	Diperlu n	Deskripsi
`p_label`	varchar	`a-z`, `A-Z`, `0-9`, `'_'`, `'-'`, `'.'`	—	Tidak	String unik yang disertakan dalam nama file cadangan. **Note** Batasnya adalah 30 karakter.
`p_owner`	varchar	Pemilik sah direktori yang ditetapka n di `p_directo ry_name` .	—	Ya	Pemilik direktori yang memuat file cadangan.

Nama parameter	Tipe data	Nilai valid	Defaul	Diperlu n	Deskripsi
p_tag	varchar	a-z, A-Z, 0-9, '_', '-', '.'	NULL	Tidak	String yang dapat digunakan untuk membedakan antara cadangan untuk menunjukkan tujuan atau penggunaan cadangan, seperti pencadangan harian, mingguan, atau inkremental sesuai tingkat tertentu. Batasnya adalah 30 karakter. Tanda tidak peka huruf besar/kecil. Tanda selalu disimpan dalam huruf besar, terlepas dari huruf yang digunakan saat memasukkan tanda. Tanda tidak harus unik. Jadi, beberapa cadangan dapat memiliki tanda yang sama. Jika Anda tidak menentukan tanda, RMAN akan menetapkan tanda default secara otomatis menggunakan format TAG*YYYYMMDDTHHMMSS* , di mana *YYYY* adalah tahun, *MM* adalah bulan, *DD* adalah hari, *HH* adalah jam (dalam format 24 jam), *MM* adalah menit, dan *SS* adalah detik. Tanggal dan waktu mengacu pada kapan RMAN memulai pencadangan. Misalnya, cadangan mungkin menerima tanda TAG20190927T214517 untuk pencadangan yang dimulai pada 2019-09-27 pukul 21:45:17.

Nama parameter	Tipe data	Nilai valid	Defaul	Diperlu n	Deskripsi
					Parameter `p_tag` didukung untuk versi mesin DB Amazon RDS for Oracle berikut: • Oracle Database 21c (21.0.0) • Oracle Database 19c (19.0.0), menggunakan 19.0.0.0.ru-2021-10.rur-2021-10.r1 atau yang lebih baru • Oracle Database 12c Rilis 2 (12.2), menggunakan 12.2.0.1.ru-2021-10.rur-2021-10.r1 atau yang lebih baru • Oracle Database 12c Rilis 1 (12.1), menggunakan 12.1.0.2.V26 atau yang lebih tinggi
`p_compress`	boolear	`TRUE`, `FALSE`	`FALSE`	Tidak	Tentukan `TRUE` untuk mengaktifkan kompresi pencadangan BASIC. Tentukan `FALSE` untuk menonaktifkan kompresi pencadangan BASIC.

Nama parameter	Tipe data	Nilai valid	Defaul	Diperlu n	Deskripsi
p_include_archive_logs	boolear	TRUE, FALSE	FALSE	Tidak	Tentukan TRUE untuk menyertakan log pengulangan yang diarsipkan dalam pencadangan. Tentukan FALSE untuk tidak menyertakan log pengulangan yang diarsipkan dari pencadangan. Jika Anda menyertakan log pengulangan yang diarsipkan dalam pencadangan, atur retensi ke satu jam atau lebih menggunakan prosedur rdsadmin.rdsadmin_util.set_configuration . Selain itu, panggil prosedur rdsadmin.rdsadmin_rman_util.crosscheck_archivelog sebelum menjalankan pencadangan. Jika tidak, pencadangan dapat gagal karena file log pengulangan yang diarsipkan tidak ada. File ini telah dihapus oleh prosedur pengelolaan Amazon RDS.
p_include_controlfile	boolear	TRUE, FALSE	FALSE	Tidak	Tentukan TRUE untuk menyertakan file kontrol di pencadangan. Tentukan FALSE untuk tidak menyertakan file kontrol dari pencadangan.

Nama parameter	Tipe data	Nilai valid	Defaul	Diperlu n	Deskripsi
p_optimize	boolear	TRUE, FALSE	TRUE	Tidak	Tentukan TRUE untuk mengaktifkan pengoptimalan pencadangan, jika log pengulangan yang diarsipkan disertaka n, untuk mengurangi ukuran cadangan. Tentukan FALSE untuk menonaktifkan pengoptimalan pencadangan.
p_parallel	number	Suatu bilangan bulat yang valid antara 1 dan 254 untuk Oracle Database Enterpris e Edition (EE) 1 untuk Oracle Database edisi lain	1	Tidak	Jumlah saluran.
p_rman_to _dbms_out put	boolear	TRUE, FALSE	FALSE	Tidak	Jika TRUE, artinya output RMAN dikirimkan ke paket DBMS_OUTPUT dengan tambahan ke file di direktori BDUMP. Di SQL*Plus, gunakan SET SERVEROUTPUT ON untuk melihat outputnya. Jika FALSE, artinya output RMAN hanya dikirimkan ke file di direktori BDUMP.

Nama parameter	Tipe data	Nilai valid	Defaul	Diperlu n	Deskripsi
`p_section_size_mb`	number	Bilangan bulat yang valid	`NULL`	Tidak	Ukuran bagian dalam megabyte (MB). Memvalidasi secara paralel dengan membagi setiap file ke dalam ukuran bagian yang ditentukan. Jika `NULL`, artinya parameter akan diabaikan.
`p_validation_type`	varchar	`'PHYSICAL'` , `'PHYSICAL+LOGICAL'`	`'PHYS'`	Tidak	Level deteksi kerusakan. Tentukan `'PHYSICAL'` untuk memeriksa kerusakan fisik. Contoh kerusakan fisik adalah kerusakan blok yang tidak sesuai pada header dan footer. Tentukan `'PHYSICAL+LOGICAL'` untuk memeriksa inkonsistensi logis selain kerusakan fisik. Contoh kerusakan logis adalah blok yang rusak.

Memvalidasi file database dalam RDS untuk Oracle

Anda dapat menggunakan paket Amazon RDS `rdsadmin.rdsadmin_rman_util` untuk memvalidasi file database Amazon RDS for Oracle, seperti file data, tablespaces, file kontrol, dan file parameter server (SPFiles).

Untuk informasi selengkapnya tentang validasi RMAN, lihat Validating database files and backups dan VALIDATE di dokumentasi Oracle.

Topik

- Memvalidasi database
- Memvalidasi basis data penyewa
- Memvalidasi ruang tabel

- Memvalidasi file kontrol
- Memvalidasi SPFILE
- Memvalidasi file data Oracle

Memvalidasi database

Untuk memvalidasi semua file yang relevan yang digunakan oleh database Oracle di RDS untuk Oracle, gunakan prosedur Amazon RDS. `rdsadmin.rdsadmin_rman_util.validate_database`

Prosedur ini menggunakan parameter umum berikut untuk tugas RMAN:

- `p_validation_type`
- `p_parallel`
- `p_section_size_mb`
- `p_rman_to_dbms_output`

Untuk informasi selengkapnya, lihat Parameter umum untuk prosedur RMAN.

Contoh berikut memvalidasi database menggunakan nilai default untuk parameter.

```
EXEC rdsadmin.rdsadmin_rman_util.validate_database;
```

Contoh berikut memvalidasi database menggunakan nilai yang ditentukan untuk parameter.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.validate_database(
        p_validation_type     => 'PHYSICAL+LOGICAL',
        p_parallel            => 4,
        p_section_size_mb     => 10,
        p_rman_to_dbms_output => FALSE);
END;
/
```

Saat parameter `p_rman_to_dbms_output` ditetapkan ke `FALSE`, output RMAN ditulis ke file di direktori `BDUMP`.

Untuk melihat file di direktori `BDUMP`, jalankan pernyataan `SELECT` berikut.

```
SELECT * FROM table(rdsadmin.rds_file_util.listdir('BDUMP')) order by mtime;
```

Untuk melihat konten file di direktori BDUMP, jalankan pernyataan SELECT berikut.

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','rds-rman-
validate-nnn.txt'));
```

Ganti nama file dengan nama file yang ingin Anda lihat.

Memvalidasi basis data penyewa

Untuk memvalidasi file data basis data penyewa dalam basis data kontainer (CDB), gunakan prosedur `rdsadmin.rdsadmin_rman_util.validate_tenant` Amazon RDS.

Prosedur ini hanya berlaku untuk basis data penyewa saat ini dan menggunakan parameter umum berikut untuk tugas RMAN:

- `p_validation_type`
- `p_parallel`
- `p_section_size_mb`
- `p_rman_to_dbms_output`

Untuk informasi selengkapnya, lihat Parameter umum untuk prosedur RMAN. Prosedur ini didukung untuk versi mesin DB berikut:

- Oracle Database 21c (21.0.0) CDB
- Oracle Database 19c (19.0.0) CDB

Contoh berikut memvalidasi basis data penyewa saat ini menggunakan nilai default untuk parameter.

```
EXEC rdsadmin.rdsadmin_rman_util.validate_tenant;
```

Contoh berikut memvalidasi basis data penyewa saat ini menggunakan nilai yang ditentukan untuk parameter.

```
BEGIN
```

```
    rdsadmin.rdsadmin_rman_util.validate_tenant(
        p_validation_type     => 'PHYSICAL+LOGICAL',
        p_parallel            => 4,
        p_section_size_mb     => 10,
        p_rman_to_dbms_output => FALSE);
END;
/
```

Saat parameter `p_rman_to_dbms_output` ditetapkan ke `FALSE`, output RMAN ditulis ke file di direktori `BDUMP`.

Untuk melihat file di direktori `BDUMP`, jalankan pernyataan `SELECT` berikut.

```
SELECT * FROM table(rdsadmin.rds_file_util.listdir('BDUMP')) order by mtime;
```

Untuk melihat konten file di direktori `BDUMP`, jalankan pernyataan `SELECT` berikut.

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','rds-rman-validate-nnn.txt'));
```

Ganti nama file dengan nama file yang ingin Anda lihat.

Memvalidasi ruang tabel

Untuk memvalidasi file terkait dengan ruang tabel, gunakan prosedur `rdsadmin.rdsadmin_rman_util.validate_tablespace` Amazon RDS.

Prosedur ini menggunakan parameter umum berikut untuk tugas RMAN:

- `p_validation_type`
- `p_parallel`
- `p_section_size_mb`
- `p_rman_to_dbms_output`

Untuk informasi selengkapnya, lihat Parameter umum untuk prosedur RMAN.

Prosedur ini juga menggunakan parameter tambahan berikut.

Nama parameter	Tipe data	Nilai valid	Default	Diperlukan	Deskripsi
`p_tablespace_name`	varchar2	Nama ruang tabel yang valid	—	Ya	Nama ruang tabel.

Memvalidasi file kontrol

Untuk memvalidasi hanya file kontrol yang digunakan instans DB Amazon RDS Oracle, gunakan prosedur `rdsadmin.rdsadmin_rman_util.validate_current_controlfile` Amazon RDS.

Prosedur ini menggunakan parameter umum berikut untuk tugas RMAN:

- `p_validation_type`
- `p_rman_to_dbms_output`

Untuk informasi selengkapnya, lihat Parameter umum untuk prosedur RMAN.

Memvalidasi SPFILE

Untuk memvalidasi hanya file parameter server (SPFILE) yang digunakan instans DB Amazon RDS Oracle, gunakan prosedur `rdsadmin.rdsadmin_rman_util.validate_spfile` Amazon RDS.

Prosedur ini menggunakan parameter umum berikut untuk tugas RMAN:

- `p_validation_type`
- `p_rman_to_dbms_output`

Untuk informasi selengkapnya, lihat Parameter umum untuk prosedur RMAN.

Memvalidasi file data Oracle

Untuk memvalidasi file data, gunakan prosedur `rdsadmin.rdsadmin_rman_util.validate_datafile` Amazon RDS.

Prosedur ini menggunakan parameter umum berikut untuk tugas RMAN:

- p_validation_type
- p_parallel
- p_section_size_mb
- p_rman_to_dbms_output

Untuk informasi selengkapnya, lihat Parameter umum untuk prosedur RMAN.

Prosedur ini juga menggunakan parameter tambahan berikut.

Nama parameter	Tipe data	Nilai valid	Default	Diperlukan	Deskripsi
p_datafile	varchar2	Nomor ID file data yang valid atau nama file data yang valid termasuk jalur lengkap	—	Ya	Nomor ID file data (dari v$datafile.file#) atau nama file data lengkap termasuk jalur (dari v$datafile.name).
p_from_block	number	Bilangan bulat yang valid	NULL	Tidak	Jumlah blok tempat validasi mulai dalam file data. Jika ditetapkan ke NULL, 1 akan digunakan.
p_to_block	number	Bilangan bulat yang valid	NULL	Tidak	Jumlah blok tempat validasi berakhir dalam file data. Jika ditetapkan ke NULL, blok maksimum dalam file data digunakan.

Mengaktifkan dan menonaktifkan pelacakan perubahan blok

Pelacakan perubahan blok mencatat blok yang berubah dalam file pelacakan. Teknik ini dapat meningkatkan performa pencadangan inkremental RMAN. Untuk informasi selengkapnya, lihat Using Block Change Tracking to Improve Incremental Backup Performance dalam dokumentasi Oracle Database.

Fitur RMAN tidak didukung dalam replika baca. Namun, sebagai bagian dari strategi ketersediaan tinggi, Anda dapat memilih untuk mengaktifkan pelacakan blok dalam replika hanya-baca menggunakan prosedur `rdsadmin.rdsadmin_rman_util.enable_block_change_tracking`. Jika Anda mempromosikan replika hanya-baca ini ke instans DB sumber, pelacakan perubahan blok akan diaktifkan untuk instans sumber baru. Dengan demikian, instans Anda dapat memperoleh manfaat dari pencadangan inkremental yang cepat.

Prosedur pelacakan perubahan blok didukung di Edisi Perusahaan hanya untuk versi mesin DB berikut:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)
- Oracle Database 12c Rilis 2 (12.2), menggunakan 12.2.0.1.ru-2019-01.rur-2019-01.r1 atau yang lebih tinggi (tidak digunakan lagi)
- Oracle Database 12c Rilis 1 (12.1), menggunakan 12.1.0.2.v15 atau yang lebih tinggi (tidak digunakan lagi)

> **Note**
>
> Di CDB penyewa tunggal, operasi berikut dapat dijalankan, tetapi tidak ada mekanisme yang dapat dilihat pelanggan yang dapat mendeteksi status operasi saat ini. Lihat juga Batasan CDB RDS for Oracle.

Untuk mengaktifkan pelacakan perubahan blok untuk instans DB, gunakan prosedur `rdsadmin.rdsadmin_rman_util.enable_block_change_tracking` Amazon RDS. Untuk menonaktifkan pelacakan perubahan blok, gunakan `disable_block_change_tracking`. Prosedur ini tidak menggunakan parameter.

Untuk menentukan apakah pelacakan perubahan blok diaktifkan untuk instans DB Anda, jalankan kueri berikut.

```
SELECT STATUS, FILENAME FROM V$BLOCK_CHANGE_TRACKING;
```

Contoh berikut memungkinkan pelacakan perubahan blok untuk instans DB.

```
EXEC rdsadmin.rdsadmin_rman_util.enable_block_change_tracking;
```

Contoh berikut menonaktifkan pelacakan perubahan blok untuk instans DB.

```
EXEC rdsadmin.rdsadmin_rman_util.disable_block_change_tracking;
```

Memeriksa ulang log pengulangan yang diarsipkan

Anda dapat memeriksa ulang log pengulangan yang diarsipkan menggunakan prosedur `rdsadmin.rdsadmin_rman_util.crosscheck_archivelog` Amazon RDS.

Anda dapat menggunakan prosedur ini untuk memeriksa ulang log pengulangan yang diarsipkan yang terdaftar dalam file kontrol. Anda juga dapat menghapus catatan log yang sudah tidak berlaku. Saat RMAN membuat cadangan, RMAN akan membuat catatan di file kontrol. Seiring berjalannya waktu, catatan ini meningkatkan ukuran file kontrol. Sebaiknya hapus catatan yang sudah tidak berlaku secara berkala.

> **Note**
>
> Pencadangan Amazon RDS standar tidak menggunakan RMAN sehingga tidak membuat catatan di file kontrol.

Prosedur ini menggunakan parameter `p_rman_to_dbms_output` umum untuk tugas RMAN.

Untuk informasi selengkapnya, lihat Parameter umum untuk prosedur RMAN.

Prosedur ini juga menggunakan parameter tambahan berikut.

Nama parameter	Tipe data	Nilai valid	Default	Diperlukan	Deskripsi
`p_delete_expired`	boolean	`TRUE`, `FALSE`	`TRUE`	Tidak	Saat ditetapkan ke `TRUE`, hapus riwayat

Nama parameter	Tipe data	Nilai valid	Default	Diperlukan	Deskripsi
					log pengulangan yang diarsipkan dari file kontrol. Saat ditetapkan ke FALSE, simpan riwayat log pengulangan yang diarsipkan di file kontrol.

Prosedur ini didukung untuk versi mesin DB Amazon RDS for Oracle berikut:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)
- Oracle Database 12c Rilis 2 (12.2), menggunakan 12.2.0.1.ru-2019-01.rur-2019-01.r1 atau yang lebih tinggi
- Oracle Database 12c Rilis 1 (12.1), menggunakan 12.1.0.2.v15 atau yang lebih tinggi

Contoh berikut menandai catatan log pengulangan yang diarsipkan di file kontrol sebagai kedaluwarsa, tetapi tidak menghapus catatan tersebut.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.crosscheck_archivelog(
        p_delete_expired      => FALSE,
        p_rman_to_dbms_output => FALSE);
END;
/
```

Contoh berikut menghapus catatan log pengulangan yang diarsipkan yang telah kedaluwarsa dari file kontrol.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.crosscheck_archivelog(
        p_delete_expired      => TRUE,
        p_rman_to_dbms_output => FALSE);
```

```
END;
/
```

Mencadangkan file log redo yang diarsipkan

Anda dapat menggunakan paket `rdsadmin.rdsadmin_rman_util` Amazon RDS untuk mencadangkan log pengulangan yang diarsipkan untuk instans DB Amazon RDS Oracle.

Prosedur untuk mencadangkan log pengulangan yang diarsipkan mendukung versi mesin DB Amazon RDS for Oracle berikut:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)
- Oracle Database 12c Rilis 2 (12.2), menggunakan 12.2.0.1.ru-2019-01.rur-2019-01.r1 atau yang lebih tinggi
- Oracle Database 12c Rilis 1 (12.1), menggunakan 12.1.0.2.v15 atau yang lebih tinggi

Topik

Mencadangkan semua log pengulangan yang diarsipkan

Untuk mencadangkan semua log pengulangan yang diarsipkan untuk instans DB Amazon RDS Oracle, gunakan prosedur `rdsadmin.rdsadmin_rman_util.backup_archivelog_all` Amazon RDS.

Prosedur ini menggunakan parameter umum berikut untuk tugas RMAN:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_compress`

- p_rman_to_dbms_output
- p_tag

Untuk informasi selengkapnya, lihat Parameter umum untuk prosedur RMAN.

Contoh berikut mencadangkan semua log pengulangan yang diarsipkan untuk instans DB.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_archivelog_all(
        p_owner               => 'SYS',
        p_directory_name      => 'MYDIRECTORY',
        p_parallel            => 4,
        p_tag                 => 'MY_LOG_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Mencadangkan log pengulangan yang diarsipkan dari rentang tanggal

Untuk mencadangkan log pengulangan tertentu yang diarsipkan untuk instans DB Amazon RDS Oracle dengan menentukan rentang tanggal, gunakan prosedur `rdsadmin.rdsadmin_rman_util.backup_archivelog_date` Amazon RDS. Rentang tanggal menentukan log pengulangan yang diarsipkan mana yang akan dicadangkan.

Prosedur ini menggunakan parameter umum berikut untuk tugas RMAN:

- p_owner
- p_directory_name
- p_label
- p_parallel
- p_compress
- p_rman_to_dbms_output
- p_tag

Untuk informasi selengkapnya, lihat Parameter umum untuk prosedur RMAN.

Prosedur ini juga menggunakan parameter tambahan berikut.

Nama parameter	Tipe data	Nilai valid	Default	Diperlukan	Deskripsi
p_from_date	date	Tanggal antara start_date dan next_date dari log pengulangan yang diarsipkan yang ada di disk. Nilai harus kurang dari atau sama dengan nilai yang ditentukan untuk p_to_date.	—	Ya	Tanggal mulai untuk pencadangan log yang diarsipkan.
p_to_date	date	Tanggal antara start_date dan next_date dari log pengulangan yang diarsipka	—	Ya	Tanggal akhir untuk pencadangan log yang diarsipkan.

Nama parameter	Tipe data	Nilai valid	Default	Diperlukan	Deskripsi
		n yang ada di disk. Nilai harus lebih besar dari atau sama dengan nilai yang ditentukan untuk `p_from_date` .			

Contoh berikut mencadangkan log pengulangan yang diarsipkan dalam rentang tanggal tertentu untuk instans DB.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_archivelog_date(
        p_owner                => 'SYS',
        p_directory_name       => 'MYDIRECTORY',
        p_from_date            => '03/01/2019 00:00:00',
        p_to_date              => '03/02/2019 00:00:00',
        p_parallel             => 4,
        p_tag                  => 'MY_LOG_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Mencadangkan log pengulangan yang diarsipkan dari rentang SCN

Untuk mencadangkan log pengulangan tertentu yang diarsipkan untuk instans DB Amazon RDS Oracle dengan menentukan rentang nomor perubahan sistem (SCN), gunakan prosedur

`rdsadmin.rdsadmin_rman_util.backup_archivelog_scn` Amazon RDS. Rentang SCN menentukan log pengulangan yang diarsipkan mana yang akan dicadangkan.

Prosedur ini menggunakan parameter umum berikut untuk tugas RMAN:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_compress`
- `p_rman_to_dbms_output`
- `p_tag`

Untuk informasi selengkapnya, lihat Parameter umum untuk prosedur RMAN.

Prosedur ini juga menggunakan parameter tambahan berikut.

Nama parameter	Tipe data	Nilai valid	Default	Diperlukan	Deskripsi
`p_from_scn`	number	SCN dari log pengulangan yang diarsipkan yang ada di disk. Nilai harus kurang dari atau sama dengan nilai yang ditentuka	—	Ya	SCN awal untuk pencadangan log yang diarsipkan.

Nama parameter	Tipe data	Nilai valid	Default	Diperlukan	Deskripsi
		n untuk p_to_scn.			
p_to_scn	number	SCN dari log pengulangan yang diarsipkan yang ada di disk. Nilai harus lebih besar dari atau sama dengan nilai yang ditentukan untuk p_from_scn .	—	Ya	SCN akhir untuk pencadangan log yang diarsipkan.

Contoh berikut membuat pencadangan log pengulangan yang diarsipkan dalam rentang SCN untuk instans DB.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_archivelog_scn(
        p_owner                => 'SYS',
        p_directory_name       => 'MYDIRECTORY',
        p_from_scn             => 1533835,
        p_to_scn               => 1892447,
        p_parallel             => 4,
        p_tag                  => 'MY_LOG_BACKUP',
```

```
        p_rman_to_dbms_output => FALSE);
END;
/
```

Mencadangkan log pengulangan yang diarsipkan dari rentang nomor urut

Untuk mencadangkan log pengulangan tertentu yang diarsipkan untuk instans DB Amazon RDS Oracle dengan menentukan rentang nomor urut, gunakan prosedur `rdsadmin.rdsadmin_rman_util.backup_archivelog_sequence` Amazon RDS. Rentang nomor urut menentukan log pengulangan yang diarsipkan mana yang akan dicadangkan.

Prosedur ini menggunakan parameter umum berikut untuk tugas RMAN:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_compress`
- `p_rman_to_dbms_output`
- `p_tag`

Untuk informasi selengkapnya, lihat Parameter umum untuk prosedur RMAN.

Prosedur ini juga menggunakan parameter tambahan berikut.

Nama parameter	Tipe data	Nilai valid	Default	Diperlukan	Deskripsi
`p_from_sequence`	number	Nomor urut log pengulangan yang diarsipkan yang ada di disk. Nilai	—	Ya	Nomor urut awal untuk pencadangan log yang diarsipkan.

Nama parameter	Tipe data	Nilai valid	Default	Diperlukan	Deskripsi
		harus kurang dari atau sama dengan nilai yang ditentukan untuk `p_to_sequence` .			
`p_to_sequence`	number	Nomor urut log pengulangan yang diarsipkan yang ada di disk. Nilai harus lebih besar dari atau sama dengan nilai yang ditentukan untuk `p_from_sequence` .	—	Ya	Nomor urut akhir untuk pencadangan log yang diarsipkan.

Contoh berikut mencadangkan log pengulangan yang diarsipkan dalam rentang nomor urut untuk instans DB.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_archivelog_sequence(
        p_owner               => 'SYS',
        p_directory_name      => 'MYDIRECTORY',
        p_from_sequence       => 11160,
        p_to_sequence         => 11160,
        p_parallel            => 4,
        p_tag                 => 'MY_LOG_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Melakukan pencadangan basis data penuh

Anda dapat melakukan pencadangan pada semua blok file data yang disertakan dalam pencadangan menggunakan prosedur `rdsadmin.rdsadmin_rman_util.backup_database_full` Amazon RDS.

Prosedur ini menggunakan parameter umum berikut untuk tugas RMAN:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_section_size_mb`
- `p_include_archive_logs`
- `p_optimize`
- `p_compress`
- `p_rman_to_dbms_output`
- `p_tag`

Untuk informasi selengkapnya, lihat Parameter umum untuk prosedur RMAN.

Prosedur ini didukung untuk versi mesin DB Amazon RDS for Oracle berikut:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)

Contoh berikut melakukan pencadangan penuh instans DB menggunakan nilai yang ditentukan untuk parameter.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_database_full(
        p_owner               => 'SYS',
        p_directory_name      => 'MYDIRECTORY',
        p_parallel            => 4,
        p_section_size_mb     => 10,
        p_tag                 => 'FULL_DB_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Melakukan pencadangan penuh untuk basis data penyewa

Anda dapat melakukan pencadangan untuk semua blok data termasuk basis data penyewa dalam basis data kontainer (CDB). Gunakan prosedur `rdsadmin.rdsadmin_rman_util.backup_tenant_full` Amazon RDS. Prosedur ini hanya berlaku untuk pencadangan basis data saat ini dan menggunakan parameter umum berikut untuk tugas RMAN:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_section_size_mb`
- `p_include_archive_logs`
- `p_optimize`
- `p_compress`
- `p_rman_to_dbms_output`
- `p_tag`

Untuk informasi selengkapnya, lihat Parameter umum untuk prosedur RMAN.

Prosedur `rdsadmin_rman_util.backup_tenant_full` didukung untuk versi mesin DB RDS for Oracle berikut:

- Oracle Database 21c (21.0.0) CDB
- Oracle Database 19c (19.0.0) CDB

Contoh berikut melakukan pencadangan penuh untuk basis data penyewa saat ini menggunakan nilai yang ditentukan untuk parameter.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_tenant_full(
        p_owner               => 'SYS',
        p_directory_name      => 'MYDIRECTORY',
        p_parallel            => 4,
        p_section_size_mb     => 10,
        p_tag                 => 'FULL_TENANT_DB_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Melakukan pencadangan basis data inkremental

Anda dapat melakukan pencadangan inkremental untuk instans DB Anda menggunakan prosedur `rdsadmin.rdsadmin_rman_util.backup_database_incremental` Amazon RDS.

Untuk informasi selengkapnya tentang pencadangan inkremental, lihat Incremental backups di dokumentasi Oracle.

Prosedur ini menggunakan parameter umum berikut untuk tugas RMAN:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_section_size_mb`
- `p_include_archive_logs`
- `p_include_controlfile`
- `p_optimize`
- `p_compress`
- `p_rman_to_dbms_output`

- `p_tag`

Untuk informasi selengkapnya, lihat Parameter umum untuk prosedur RMAN.

Prosedur ini didukung untuk versi mesin DB Amazon RDS for Oracle berikut:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)
- Oracle Database 12c Rilis 2 (12.2), menggunakan 12.2.0.1.ru-2019-01.rur-2019-01.r1 atau yang lebih tinggi
- Oracle Database 12c Rilis 1 (12.1), menggunakan 12.1.0.2.v15 atau yang lebih tinggi

Prosedur ini juga menggunakan parameter tambahan berikut.

Nama parameter	Tipe data	Nilai valid	Default	Diperlukan	Deskripsi
`p_level`	number	`0`, `1`	`0`	Tidak	Tentukan `0` untuk mengaktifkan pencadangan inkremental. Tentukan 1 untuk mengaktifkan pencadangan inkremental non-kumulatif.

Contoh berikut melakukan pencadangan inkremental instans DB menggunakan nilai yang ditentukan untuk parameter.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_database_incremental(
        p_owner                 => 'SYS',
        p_directory_name        => 'MYDIRECTORY',
        p_level                 => 1,
        p_parallel              => 4,
```

```
        p_section_size_mb     => 10,
        p_tag                 => 'MY_INCREMENTAL_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Melakukan pencadangan inkremental untuk basis data penyewa

Anda dapat melakukan pencadangan inkremental untuk basis data penyewa saat ini di CDB Anda. Gunakan prosedur `rdsadmin.rdsadmin_rman_util.backup_tenant_incremental` Amazon RDS.

Untuk informasi selengkapnya tentang pencadangan inkremental, lihat Incremental backups di dokumentasi Oracle Database.

Prosedur ini hanya berlaku untuk basis data penyewa saat ini dan menggunakan parameter umum berikut untuk tugas RMAN:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_section_size_mb`
- `p_include_archive_logs`
- `p_include_controlfile`
- `p_optimize`
- `p_compress`
- `p_rman_to_dbms_output`
- `p_tag`

Untuk informasi selengkapnya, lihat Parameter umum untuk prosedur RMAN.

Prosedur ini didukung untuk versi mesin DB Amazon RDS for Oracle berikut:

- Oracle Database 21c (21.0.0) CDB
- Oracle Database 19c (19.0.0) CDB

Prosedur ini juga menggunakan parameter tambahan berikut.

Nama parameter	Tipe data	Nilai valid	Default	Diperlukan	Deskripsi
`p_level`	number	`0`, `1`	`0`	Tidak	Tentukan `0` untuk mengaktifkan pencadangan inkremental. Tentukan 1 untuk mengaktifkan pencadangan inkremental non-kumulatif.

Contoh berikut melakukan pencadangan inkremental untuk basis data penyewa saat ini menggunakan nilai yang ditentukan untuk parameter.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_tenant_incremental(
        p_owner               => 'SYS',
        p_directory_name      => 'MYDIRECTORY',
        p_level               => 1,
        p_parallel            => 4,
        p_section_size_mb     => 10,
        p_tag                 => 'MY_INCREMENTAL_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Mencadangkan ruang tabel

Anda dapat mencadangkan ruang tabel menggunakan prosedur `rdsadmin.rdsadmin_rman_util.backup_tablespace` Amazon RDS.

Prosedur ini menggunakan parameter umum berikut untuk tugas RMAN:

- `p_owner`

- p_directory_name
- p_label
- p_parallel
- p_section_size_mb
- p_include_archive_logs
- p_include_controlfile
- p_optimize
- p_compress
- p_rman_to_dbms_output
- p_tag

Untuk informasi selengkapnya, lihat Parameter umum untuk prosedur RMAN.

Prosedur ini juga menggunakan parameter tambahan berikut.

Nama parameter	Tipe data	Nilai valid	Default	Diperlukan	Deskripsi
p_tablespace_name	varchar2	Nama ruang tabel yang valid.	—	Ya	Nama ruang tabel untuk dicadangkan.

Prosedur ini didukung untuk versi mesin DB Amazon RDS for Oracle berikut:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)
- Oracle Database 12c Rilis 2 (12.2), menggunakan 12.2.0.1.ru-2019-01.rur-2019-01.r1 atau yang lebih tinggi
- Oracle Database 12c Rilis 1 (12.1), menggunakan 12.1.0.2.v15 atau yang lebih tinggi

Contoh berikut menjalankan pencadangan ruang tabel menggunakan nilai yang ditentukan untuk parameter.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_tablespace(
        p_owner               => 'SYS',
        p_directory_name      => 'MYDIRECTORY',
        p_tablespace_name     => MYTABLESPACE,
        p_parallel            => 4,
        p_section_size_mb     => 10,
        p_tag                 => 'MYTABLESPACE_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Mencadangkan file kontrol

Anda dapat mencadangkan file kontrol menggunakan prosedur `rdsadmin.rdsadmin_rman_util.backup_current_controlfile` Amazon RDS.

Prosedur ini menggunakan parameter umum berikut untuk tugas RMAN:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_compress`
- `p_rman_to_dbms_output`
- `p_tag`

Untuk informasi selengkapnya, lihat Parameter umum untuk prosedur RMAN.

Prosedur ini didukung untuk versi mesin DB Amazon RDS for Oracle berikut:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)
- Oracle Database 12c Rilis 2 (12.2), menggunakan 12.2.0.1.ru-2019-01.rur-2019-01.r1 atau yang lebih tinggi
- Oracle Database 12c Rilis 1 (12.1), menggunakan 12.1.0.2.v15 atau yang lebih tinggi

Contoh berikut mencadangkan file kontrol menggunakan nilai yang ditentukan untuk parameter.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_current_controlfile(
        p_owner               => 'SYS',
        p_directory_name      => 'MYDIRECTORY',
        p_tag                 => 'CONTROL_FILE_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Melakukan pemulihan media blok

Anda dapat memulihkan blok data individual, yang dikenal sebagai pemulihan media blok, menggunakan prosedur `rdsadmin.rdsadmin_rman_util.recover_datafile_block` Amazon RDS. Anda dapat menggunakan prosedur kelebihan beban ini untuk memulihkan baik blok data individu atau berbagai blok data.

Prosedur ini menggunakan parameter umum berikut untuk tugas RMAN:

- `p_rman_to_dbms_output`

Untuk informasi selengkapnya, lihat Parameter umum untuk prosedur RMAN.

Prosedur ini menggunakan parameter tambahan berikut.

Nama parameter	Tipe data	Nilai valid	Default	Diperlukan	Deskripsi
`p_datafile`	`NUMBER`	Nomor ID file data yang valid.	—	Ya	File data yang berisi blok korup. Tentukan file data dengan salah satu cara berikut: • Nomor ID file data, yang terletak di `V$DATAFILE.FILE#` • Nama file data lengkap, termasuk

Nama parameter	Tipe data	Nilai valid	Default	Diperlukan	Deskripsi
					jalur, terletak di V $DATAFILE.NAME
p_block	NUMBER	Sebuah integer yang valid.	—	Ya	Jumlah blok individu yang akan dipulihkan. Parameter berikut saling eksklusif: • p_block • p_from_block dan p_to_block
p_from_block	NUMBER	Sebuah integer yang valid.	—	Ya	Nomor blok pertama dalam berbagai blok yang akan dipulihkan. Parameter berikut saling eksklusif: • p_block • p_from_block dan p_to_block
p_to_block	NUMBER	Sebuah integer yang valid.	—	Ya	Nomor blok terakhir dalam berbagai blok yang akan dipulihkan. Parameter berikut saling eksklusif: • p_block • p_from_block dan p_to_block

Prosedur ini didukung untuk versi mesin DB Amazon RDS for Oracle berikut:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)

Contoh berikut memulihkan blok 100 dalam file data 5.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.recover_datafile_block(
        p_datafile            => 5,
        p_block               => 100,
        p_rman_to_dbms_output => TRUE);
END;
/
```

Contoh berikut memulihkan blok 100 hingga 150 dalam file data 5.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.recover_datafile_block(
        p_datafile            => 5,
        p_from_block          => 100,
        p_to_block            => 150,
        p_rman_to_dbms_output => TRUE);
END;
/
```

Melakukan tugas penjadwalan umum untuk instans DB Oracle

Beberapa pekerjaan penjadwal yang dimiliki SYS dapat mengganggu operasi basis data normal. Dukungan Oracle merekomendasikan Anda menonaktifkan pekerjaan ini atau memodifikasi jadwal. Untuk melakukan tugas untuk pekerjaan Oracle Scheduler yang dimiliki oleh SYS, gunakan paket `rdsadmin.rdsadmin_dbms_scheduler` Amazon RDS.

Prosedur `rdsadmin.rdsadmin_dbms_scheduler` mendukung versi mesin DB Amazon RDS for Oracle berikut:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c

- Oracle Database 12c Rilis 2 (12.2) di 12.2.0.2.ru-2019-07.rur-2019-07.r1 atau versi 12.2 yang lebih baru
- Oracle Database 12c Rilis 1 (12.1) di 12.1.0.2.v17 atau versi 12.1 yang lebih baru

Parameter umum untuk prosedur Oracle Scheduler

Untuk melakukan tugas dengan Oracle Scheduler, gunakan prosedur di paket `rdsadmin.rdsadmin_dbms_scheduler` Amazon RDS. Beberapa parameter umum untuk prosedur dalam paket. Paket memiliki parameter umum berikut.

Nama parameter	Tipe data	Nilai valid	Default	Diperlukan	Deskripsi
`name`	varchar2	`'SYS.BSLN_MAINTAIN_STATS_JOB' ,'SYS.CLEANUP_ONLINE_IND_BUILD'`	—	Ya	Nama pekerjaan yang akan dimodifikasi. **Note** Saat ini, Anda hanya dapat memodifikasi pekerjaan `SYS.CLEANUP_ONLINE_IND_BUILD` dan `SYS.BSLN_MAINTAIN_STATS_JOB`.
`attribute`	varchar2	`'REPEAT_INTERVAL' _NAME'`	–	Ya	Atribut untuk diubah. Untuk memodifikasi interval pengulangan untuk pekerjaan,

Nama parameter	Tipe data	Nilai valid	Default	Diperlukan	Deskripsi
					tentukan `'REPEAT_INTERVAL'` . Untuk memodifikasi nama jadwal untuk pekerjaan, tentukan `'SCHEDULE_NAME'` .
`value`	varchar2	Interval jadwal atau nama jadwal yang valid, tergantung atribut yang digunakan .	–	Ya	Nilai baru dari atribut.

Memodifikasi pekerjaan DBMS_SCHEDULER

Untuk memodifikasi komponen tertentu dari Oracle Scheduler, gunakan prosedur `dbms_scheduler.set_attribute` Oracle. Untuk informasi selengkapnya, lihat DBMS_SCHEDULER dan SET_ATTRIBUTE procedure di dokumentasi Oracle.

Saat bekerja dengan instans DB Amazon RDS, siapkan nama skema SYS ke nama objek. Contoh berikut mengatur atribut rencana sumber daya untuk objek periode hari Senin.

```
BEGIN
    DBMS_SCHEDULER.SET_ATTRIBUTE(
        name      => 'SYS.MONDAY_WINDOW',
        attribute => 'RESOURCE_PLAN',
        value     => 'resource_plan_1');
END;
```

```
/
```

Memodifikasi AutoTask jendela pemeliharaan

Instans Amazon RDS for Oracle dibuat dengan pengaturan default untuk masa pemeliharaan. Tugas pemeliharaan otomatis seperti pengumpulan statistik pengoptimal berjalan selama periode ini. Secara default, masa pemeliharaan mengaktifkan Oracle Database Resource Manager.

Untuk memodifikasi periode, gunakan paket `DBMS_SCHEDULER`. Anda mungkin perlu memodifikasi pengaturan masa pemeliharaan karena alasan berikut ini:

- Anda ingin pekerjaan pemeliharaan berjalan pada waktu yang berbeda, dengan pengaturan yang berbeda, atau tidak sama sekali. Misalnya, Anda mungkin ingin mengubah durasi periode atau mengubah waktu dan interval pengulangan.
- Anda ingin menghindari dampak kinerja dari mengaktifkan Resource Manager selama pemeliharaan. Misalnya, jika rencana pemeliharaan default ditentukan dan jika masa pemeliharaan terbuka saat basis data sedang dimuat, Anda mungkin melihat peristiwa tunggu seperti `resmgr:cpu quantum`. Peristiwa tunggu ini terkait dengan Database Resource Manager. Anda memiliki opsi berikut:
  - Pastikan masa pemeliharaan aktif selama waktu tidak sibuk untuk instans DB Anda.
  - Nonaktifkan rencana pemeliharaan default dengan mengatur atribut `resource_plan` ke string kosong.
  - Tetapkan parameter `resource_manager_plan` ke `FORCE:` dalam grup parameter Anda. Jika instans Anda menggunakan Enterprise Edition, pengaturan ini mencegah paket Database Resource Manager diaktifkan.

Untuk memodifikasi pengaturan masa pemeliharaan

1. Terhubung ke basis data Anda menggunakan klien Oracle SQL.
2. Kueri konfigurasi saat ini untuk periode penjadwal.

   Contoh berikut mengkueri konfigurasi untuk `MONDAY_WINDOW`.

   ```
   SELECT ENABLED, RESOURCE_PLAN, DURATION, REPEAT_INTERVAL
   FROM   DBA_SCHEDULER_WINDOWS
   WHERE  WINDOW_NAME='MONDAY_WINDOW';
   ```

Output berikut menunjukkan bahwa periode menggunakan nilai default.

```
ENABLED         RESOURCE_PLAN                  DURATION         REPEAT_INTERVAL
--------------- ------------------------------ ----------------
 ------------------------------
TRUE            DEFAULT_MAINTENANCE_PLAN       +000 04:00:00
 freq=daily;byday=MON;byhour=22
                                                                 ;byminute=0;
 bysecond=0
```

3. Modifikasi periode menggunakan paket DBMS_SCHEDULER.

   Contoh berikut menetapkan rencana sumber daya ke null sehingga Resource Manager tidak akan berjalan selama masa pemeliharaan.

```
BEGIN
  -- disable the window to make changes
  DBMS_SCHEDULER.DISABLE(name=>'"SYS"."MONDAY_WINDOW"',force=>TRUE);

  -- specify the empty string to use no plan
  DBMS_SCHEDULER.SET_ATTRIBUTE(name=>'"SYS"."MONDAY_WINDOW"',
 attribute=>'RESOURCE_PLAN', value=>'');

  -- re-enable the window
  DBMS_SCHEDULER.ENABLE(name=>'"SYS"."MONDAY_WINDOW"');
END;
/
```

   Contoh berikut mengatur durasi maksimum periode menjadi 2 jam.

```
BEGIN
  DBMS_SCHEDULER.DISABLE(name=>'"SYS"."MONDAY_WINDOW"',force=>TRUE);
  DBMS_SCHEDULER.SET_ATTRIBUTE(name=>'"SYS"."MONDAY_WINDOW"',
 attribute=>'DURATION', value=>'0 2:00:00');
  DBMS_SCHEDULER.ENABLE(name=>'"SYS"."MONDAY_WINDOW"');
END;
/
```

   Contoh berikut menetapkan interval pengulangan untuk setiap hari Senin pukul 10 pagi.

```
BEGIN
```

```
  DBMS_SCHEDULER.DISABLE(name=>'"SYS"."MONDAY_WINDOW"',force=>TRUE);
  DBMS_SCHEDULER.SET_ATTRIBUTE(name=>'"SYS"."MONDAY_WINDOW"',
 attribute=>'REPEAT_INTERVAL',
 value=>'freq=daily;byday=MON;byhour=10;byminute=0;bysecond=0');
  DBMS_SCHEDULER.ENABLE(name=>'"SYS"."MONDAY_WINDOW"');
END;
/
```

Mengatur zona waktu untuk pekerjaan Oracle Scheduler

Untuk memodifikasi zona waktu Oracle Scheduler, Anda dapat menggunakan prosedur `dbms_scheduler.set_scheduler_attribute` Oracle. Untuk informasi selengkapnya tentang paket `dbms_scheduler`, lihat DBMS_SCHEDULER dan SET_SCHEDULER_ATTRIBUTE dalam dokumentasi Oracle.

Untuk memodifikasi pengaturan zona waktu saat ini

1. Terhubung ke basis data menggunakan klien seperti SQL Developer. Untuk informasi selengkapnya, lihat Menghubungkan ke instans DB menggunakan pengembang Oracle SQL.
2. Atur zona waktu default sebagai berikut, menggantikan zona waktu untuk *`time_zone_name`*.

```
BEGIN
  DBMS_SCHEDULER.SET_SCHEDULER_ATTRIBUTE(
    attribute => 'default_timezone',
    value => 'time_zone_name'
  );
END;
/
```

Dalam contoh berikut, Anda mengubah zona waktu ke Asia/Shanghai.

Mulai dengan melakukan kueri zona waktu saat ini, seperti yang ditunjukkan berikut.

```
SELECT VALUE FROM DBA_SCHEDULER_GLOBAL_ATTRIBUTE WHERE
 ATTRIBUTE_NAME='DEFAULT_TIMEZONE';
```

Output menunjukkan bahwa zona waktu saat ini adalah ETC/UTC.

```
VALUE
```

```
-------
Etc/UTC
```

Kemudian Anda mengubah zona waktu ke Asia/Shanghai.

```
BEGIN
  DBMS_SCHEDULER.SET_SCHEDULER_ATTRIBUTE(
    attribute => 'default_timezone',
    value => 'Asia/Shanghai'
  );
END;
/
```

Untuk informasi selengkapnya tentang perubahan zona waktu sistem, lihat Zona waktu Oracle.

Menonaktifkan pekerjaan Oracle Scheduler yang dimiliki oleh SYS

Untuk menonaktifkan pekerjaan Oracle Scheduler yang dimiliki oleh pengguna SYS, gunakan prosedur `rdsadmin.rdsadmin_dbms_scheduler.disable`.

Prosedur ini menggunakan parameter umum `name` untuk tugas Oracle Scheduler. Untuk informasi selengkapnya, lihat Parameter umum untuk prosedur Oracle Scheduler.

Contoh berikut menonaktifkan pekerjaan `SYS.CLEANUP_ONLINE_IND_BUILD` Oracle Scheduler.

```
BEGIN
   rdsadmin.rdsadmin_dbms_scheduler.disable('SYS.CLEANUP_ONLINE_IND_BUILD');
END;
/
```

Mengaktifkan pekerjaan Oracle Scheduler yang dimiliki oleh SYS

Untuk mengaktifkan pekerjaan Oracle Scheduler yang dimiliki oleh SYS, gunakan prosedur `rdsadmin.rdsadmin_dbms_scheduler.enable`.

Prosedur ini menggunakan parameter umum `name` untuk tugas Oracle Scheduler. Untuk informasi selengkapnya, lihat Parameter umum untuk prosedur Oracle Scheduler.

Contoh berikut memungkinkan pekerjaan `SYS.CLEANUP_ONLINE_IND_BUILD` Oracle Scheduler.

```
BEGIN
   rdsadmin.rdsadmin_dbms_scheduler.enable('SYS.CLEANUP_ONLINE_IND_BUILD');
```

```
END;
/
```

Memodifikasi interval pengulangan Oracle Scheduler untuk pekerjaan tipe CALENDAR

Untuk memodifikasi interval pengulangan untuk memodifikasi pekerjaan Oracle Scheduler yang dimiliki SYS dari tipe `CALENDAR`, gunakan prosedur `rdsadmin.rdsadmin_dbms_scheduler.disable`.

Prosedur ini menggunakan parameter umum berikut untuk tugas Oracle Scheduler:

- `name`
- `attribute`
- `value`

Untuk informasi selengkapnya, lihat Parameter umum untuk prosedur Oracle Scheduler.

Contoh berikut memodifikasi interval pengulangan dari pekerjaan `SYS.CLEANUP_ONLINE_IND_BUILD` Oracle Scheduler.

```
BEGIN
     rdsadmin.rdsadmin_dbms_scheduler.set_attribute(
          name      => 'SYS.CLEANUP_ONLINE_IND_BUILD',
          attribute => 'repeat_interval',
          value     => 'freq=daily;byday=FRI,SAT;byhour=20;byminute=0;bysecond=0');
END;
/
```

Memodifikasi interval pengulangan Oracle Scheduler untuk pekerjaan tipe NAMED

Beberapa pekerjaan Oracle Scheduler menggunakan nama jadwal, bukan interval. Untuk jenis pekerjaan ini, Anda harus membuat jadwal baru yang sudah dinamai di skema pengguna master. Gunakan standar prosedur `sys.dbms_scheduler.create_schedule` Oracle untuk melakukan tindakan ini. Selain itu, gunakan `rdsadmin.rdsadmin_dbms_scheduler.set_attribute procedure` untuk menetapkan jadwal baru yang dinamai pada pekerjaan.

Prosedur ini menggunakan parameter umum berikut untuk tugas Oracle Scheduler:

- `name`
- `attribute`

- `value`

Untuk informasi selengkapnya, lihat Parameter umum untuk prosedur Oracle Scheduler.

Contoh berikut memodifikasi interval pengulangan dari pekerjaan `SYS.BSLN_MAINTAIN_STATS_JOB` Oracle Scheduler.

```
BEGIN
     DBMS_SCHEDULER.CREATE_SCHEDULE (
          schedule_name   => 'rds_master_user.new_schedule',
          start_date      => SYSTIMESTAMP,
          repeat_interval =>
 'freq=daily;byday=MON,TUE,WED,THU,FRI;byhour=0;byminute=0;bysecond=0',
          end_date        => NULL,
          comments        => 'Repeats daily forever');
END;
/

BEGIN
     rdsadmin.rdsadmin_dbms_scheduler.set_attribute (
          name      => 'SYS.BSLN_MAINTAIN_STATS_JOB',
          attribute => 'schedule_name',
          value     => 'rds_master_user.new_schedule');
END;
/
```

Menonaktifkan autocommit untuk pembuatan pekerjaan Oracle Scheduler

Ketika `DBMS_SCHEDULER.CREATE_JOB` membuat pekerjaan Oracle Scheduler, tindakan ini langsung menciptakan pekerjaan dan melakukan perubahan. Anda mungkin perlu memasukkan pembuatan pekerjaan Oracle Scheduler dalam transaksi pengguna untuk melakukan hal berikut:

- Lakukan roll-back pada pekerjaan Oracle Schedule saat roll-back diterapkan pada transaksi pengguna.
- Buat pekerjaan Oracle Scheduler saat transaksi pengguna utama dilakukan.

Anda dapat menggunakan prosedur `rdsadmin.rdsadmin_dbms_scheduler.set_no_commit_flag` untuk mengaktifkan perilaku ini. Prosedur ini tidak menggunakan parameter. Anda dapat menggunakan prosedur ini dalam rilis RDS for Oracle berikut:

- 21.0.0.0.ru-2022-07.rur-2022-07.r1 dan yang lebih baru
- 19.0.0.0.ru-2022-07.rur-2022-07.r1 dan yang lebih baru

Contoh berikut menonaktifkan autocommit untuk Oracle Scheduler, membuat pekerjaan Oracle Scheduler, dan kemudian melakukan roll-back pada transaksi. Karena autocommit dinonaktifkan, basis data juga melakukan roll-back pada pembuatan pekerjaan Oracle Scheduler.

```
BEGIN
  rdsadmin.rdsadmin_dbms_scheduler.set_no_commit_flag;
  DBMS_SCHEDULER.CREATE_JOB(job_name   => 'EMPTY_JOB',
                            job_type   => 'PLSQL_BLOCK',
                            job_action => 'begin null; end;',
                            auto_drop  => false);
  ROLLBACK;
END;
/

PL/SQL procedure successfully completed.

SELECT * FROM DBA_SCHEDULER_JOBS WHERE JOB_NAME='EMPTY_JOB';

no rows selected
```

Melakukan tugas diagnostik umum untuk instans DB Oracle

Oracle Database mencakup infrastruktur kemampuan diagnosis kesalahan yang dapat Anda gunakan untuk mengatasi masalah basis data. Dalam terminologi Oracle, masalah adalah kesalahan kritis seperti bug kode atau kerusakan data. Insiden adalah kondisi saat terjadi masalah. Jika kesalahan yang sama terjadi tiga kali, infrastruktur akan menunjukkan tiga insiden masalah ini. Untuk informasi selengkapnya, lihat Diagnosing and resolving problems di dokumentasi Oracle Database.

Utilitas Automatic Diagnostic Repository Command Interpreter (ADRCI) merupakan alat baris perintah Oracle yang Anda gunakan untuk mengelola data diagnostik. Misalnya, Anda dapat menggunakan alat ini untuk menyelidiki masalah dan memaketkan data diagnostik. Paket insiden mencakup data diagnostik insiden atau semua insiden yang mengacu pada masalah tertentu. Anda dapat mengunggah paket insiden, yang diimplementasikan sebagai file .zip, ke Oracle Support.

Untuk memberikan pengalaman layanan terkelola, Amazon RDS tidak menyediakan akses shell ADRCI. Untuk melakukan tugas diagnostik untuk instans Oracle Anda, gunakan paket `rdsadmin.rdsadmin_adrci_util` Amazon RD.

Dengan menggunakan fungsi di `rdsadmin_adrci_util`, Anda dapat membuat daftar serta mengemas masalah dan insiden, serta menunjukkan file jejak. Semua fungsi mengembalikan ID tugas. ID ini merupakan bagian dari nama file log yang berisi output ADRCI, seperti di `dbtask-`*`task_id`*`.log`. File log terletak di direktori BDUMP. Anda dapat mengunduh file log dengan mengikuti prosedur yang dijelaskan di Mengunduh file log basis data.

Parameter umum untuk prosedur diagnostik

Untuk melakukan tugas diagnostik, gunakan fungsi dalam paket `rdsadmin.rdsadmin_adrci_util` Amazon RDS. Paket memiliki parameter umum berikut.

Nama parameter	Tipe data	Nilai valid	Default	Diperlukan	Deskripsi
`incident_id`	number	ID insiden yang valid atau null	Null	Tidak	Jika nilainya null, fungsi akan menunjukkan semua insiden. Jika nilainya tidak null dan mewakili ID insiden yang valid, fungsi akan menampilkan insiden yang ditentukan.
`problem_id`	number	ID masalah yang valid atau null	Null	Tidak	Jika nilainya null, fungsinya akan menunjukkan semua masalah. Jika nilainya tidak null dan mewakili ID masalah yang valid, fungsi akan menampilkan masalah yang ditentukan.
`last`	number	Bilangan bulat	Null	Tidak	Jika nilainya null, fungsi akan

Nama parameter	Tipe data	Nilai valid	Default	Diperlukan	Deskripsi
		valid yang lebih besar dari 0 atau null			menampilkan maksimal 50 item. Jika nilainya tidak null, fungsi akan menampilkan jumlah yang ditentukan.

Daftar insiden

Untuk membuat daftar masalah diagnostik bagi Oracle, gunakan fungsi `rdsadmin.rdsadmin_adrci_util.list_adrci_incidents` Amazon RDS. Anda dapat membuat daftar insiden dalam mode dasar atau terperinci. Secara default, fungsi ini mencantumkan 50 insiden terbaru.

Fungsi ini menggunakan parameter umum berikut:

- `incident_id`
- `problem_id`
- `last`

Jika Anda menentukan `incident_id` dan`problem_id`, `incident_id` akan mengganti `problem_id`. Untuk informasi selengkapnya, lihat Parameter umum untuk prosedur diagnostik.

Fungsi ini menggunakan parameter tambahan berikut.

Nama parameter	Tipe data	Nilai valid	Default	Diperlukan	Deskripsi
`detail`	boolean	`TRUE` atau `FALSE`	`FALSE`	Tidak	Jika `TRUE`, fungsi ini mencantumkan insiden dalam mode terperinci. Jika `FALSE`, fungsi ini mencantumkan daftar

Nama parameter	Tipe data	Nilai valid	Default	Diperlukan	Deskripsi
					insiden dalam mode dasar.

Untuk mencantumkan semua insiden, kueri fungsi `rdsadmin.rdsadmin_adrci_util.list_adrci_incidents` tanpa argumen apa pun. Kueri akan mengembalikan ID tugas.

```
SQL> SELECT rdsadmin.rdsadmin_adrci_util.list_adrci_incidents AS task_id FROM DUAL;

TASK_ID
------------------
1590786706158-3126
```

Anda juga dapat memanggil fungsi `rdsadmin.rdsadmin_adrci_util.list_adrci_incidents` tanpa argumen dan menyimpan output di variabel klien SQL. Anda dapat menggunakan variabel di pernyataan lainnya.

```
SQL> VAR task_id VARCHAR2(80);
SQL> EXEC :task_id := rdsadmin.rdsadmin_adrci_util.list_adrci_incidents;

PL/SQL procedure successfully completed.
```

Untuk membaca file log, panggil prosedur `rdsadmin.rds_file_util.read_text_file` Amazon RDS. Berikan ID tugas sebagai bagian dari nama file. Output berikut menunjukkan tiga insiden: 53523, 53522, dan 53521.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||:task_id||'.log'));

TEXT
--------------------------------------------------------------------------------
2020-05-29 21:11:46.193 UTC [INFO ] Listing ADRCI incidents.
2020-05-29 21:11:46.256 UTC [INFO ]
ADR Home = /rdsdbdata/log/diag/rdbms/orcl_a/ORCL:
*************************************************************************
INCIDENT_ID PROBLEM_KEY                                                 CREATE_TIME
```

```
---------- ------------------------------------------------------------
 ---------------------------------------
53523       ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_003 2020-05-29
 20:15:20.928000 +00:00
53522       ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_002 2020-05-29
 20:15:15.247000 +00:00
53521       ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_001 2020-05-29
 20:15:06.047000 +00:00
3 rows fetched


2020-05-29 21:11:46.256 UTC [INFO ] The ADRCI incidents were successfully listed.
2020-05-29 21:11:46.256 UTC [INFO ] The task finished successfully.

14 rows selected.
```

Untuk membuat daftar insiden tertentu, tentukan ID menggunakan parameter `incident_id`. Dalam contoh berikut, Anda mengueri file log untuk insiden 53523 saja.

```
SQL> EXEC :task_id :=
 rdsadmin.rdsadmin_adrci_util.list_adrci_incidents(incident_id=>53523);

PL/SQL procedure successfully completed.

SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||:task_id||'.log'));

TEXT
--------------------------------------------------------------------------------
2020-05-29 21:15:25.358 UTC [INFO ] Listing ADRCI incidents.
2020-05-29 21:15:25.426 UTC [INFO ]
ADR Home = /rdsdbdata/log/diag/rdbms/orcl_a/ORCL:
*************************************************************************
INCIDENT_ID          PROBLEM_KEY
 CREATE_TIME
-------------------- ------------------------------------------------------------
 ---------------------------------
53523                ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_003
 2020-05-29 20:15:20.928000 +00:00
1 rows fetched


2020-05-29 21:15:25.427 UTC [INFO ] The ADRCI incidents were successfully listed.
```

```
2020-05-29 21:15:25.427 UTC [INFO ] The task finished successfully.

12 rows selected.
```

Mencantumkan masalah

Untuk mencantumkan masalah diagnostik bagi Oracle, gunakan fungsi `rdsadmin.rdsadmin_adrci_util.list_adrci_problems` Amazon RDS.

Secara default, fungsi ini akan mencantumkan 50 masalah terbaru.

Fungsi ini menggunakan parameter umum `problem_id` dan `last`. Untuk informasi selengkapnya, lihat Parameter umum untuk prosedur diagnostik.

Guna mendapatkan ID tugas untuk semua masalah, panggil fungsi `rdsadmin.rdsadmin_adrci_util.list_adrci_problems` tanpa argumen apa pun, dan simpan hasilnya dalam variabel klien SQL.

```
SQL> EXEC :task_id := rdsadmin.rdsadmin_adrci_util.list_adrci_problems;

PL/SQL procedure successfully completed.
```

Untuk membaca file log, panggil fungsi `rdsadmin.rds_file_util.read_text_file`, berikan ID tugas sebagai bagian dari nama file. Di output berikut, file log menampilkan tiga masalah: 1, 2, dan 3.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||:task_id||'.log'));

TEXT
--------------------------------------------------------------------------------
2020-05-29 21:18:50.764 UTC [INFO ] Listing ADRCI problems.
2020-05-29 21:18:50.829 UTC [INFO ]
ADR Home = /rdsdbdata/log/diag/rdbms/orcl_a/ORCL:
*************************************************************************
PROBLEM_ID   PROBLEM_KEY                                                 LAST_INCIDENT
       LASTINC_TIME
---------- ------------------------------------------------------------ -------------
 -------------------------------
2          ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_003 53523
 2020-05-29 20:15:20.928000 +00:00
3          ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_002 53522
 2020-05-29 20:15:15.247000 +00:00
```

```
1          ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_001 53521
 2020-05-29 20:15:06.047000 +00:00
3 rows fetched


2020-05-29 21:18:50.829 UTC [INFO ] The ADRCI problems were successfully listed.
2020-05-29 21:18:50.829 UTC [INFO ] The task finished successfully.

14 rows selected.
```

Dalam contoh berikut, Anda hanya mencantumkan masalah nomor 3.

```
SQL> EXEC :task_id := rdsadmin.rdsadmin_adrci_util.list_adrci_problems(problem_id=>3);

PL/SQL procedure successfully completed.
```

Untuk membaca file log masalah 3, panggil `rdsadmin.rds_file_util.read_text_file`. Berikan ID tugas sebagai bagian dari nama file.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||:task_id||'.log'));

TEXT
--------------------------------------------------------------------------------
2020-05-29 21:19:42.533 UTC [INFO ] Listing ADRCI problems.
2020-05-29 21:19:42.599 UTC [INFO ]
ADR Home = /rdsdbdata/log/diag/rdbms/orcl_a/ORCL:
*************************************************************************
PROBLEM_ID PROBLEM_KEY                                                     LAST_INCIDENT
 LASTINC_TIME
---------- ------------------------------------------------------------ -------------
 -------------------------------
3          ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_002 53522
 2020-05-29 20:15:15.247000 +00:00
1 rows fetched


2020-05-29 21:19:42.599 UTC [INFO ] The ADRCI problems were successfully listed.
2020-05-29 21:19:42.599 UTC [INFO ] The task finished successfully.

12 rows selected.
```

Membuat paket insiden

Anda dapat membuat paket insiden menggunakan fungsi `rdsadmin.rdsadmin_adrci_util.create_adrci_package` Amazon RDS. Output adalah file .zip yang dapat Anda berikan ke Oracle Support.

Fungsi ini menggunakan parameter umum berikut:

- `problem_id`
- `incident_id`

Pastikan untuk menentukan salah satu parameter sebelumnya. Jika Anda menentukan kedua parameter, `incident_id` akan menimpa `problem_id`. Untuk informasi selengkapnya, lihat Parameter umum untuk prosedur diagnostik.

Untuk membuat paket insiden tertentu, panggil fungsi Amazon RDS `rdsadmin.rdsadmin_adrci_util.create_adrci_package` dengan parameter `incident_id`. Contoh berikut membuat paket untuk insiden 53523.

```
SQL> EXEC :task_id :=
 rdsadmin.rdsadmin_adrci_util.create_adrci_package(incident_id=>53523);

PL/SQL procedure successfully completed.
```

Untuk membaca file log, panggil `rdsadmin.rds_file_util.read_text_file`. Anda dapat memberikan ID tugas sebagai bagian dari nama file. Output menunjukkan bahwa Anda menghasilkan paket insiden `ORA700EVE_20200529212043_COM_1.zip`.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||:task_id||'.log'));

TEXT
--------------------------------------------------------------------------------
2020-05-29 21:20:43.031 UTC [INFO ] The ADRCI package is being created.
2020-05-29 21:20:47.641 UTC [INFO ] Generated package 1 in file /rdsdbdata/log/trace/
ORA700EVE_20200529212043_COM_1.zip, mode complete
2020-05-29 21:20:47.642 UTC [INFO ] The ADRCI package was successfully created.
2020-05-29 21:20:47.642 UTC [INFO ] The task finished successfully.
```

Untuk membuat paket data diagnostik masalah tertentu, tentukan ID menggunakan parameter `problem_id`. Dalam contoh berikut, Anda hanya memaketkan data untuk masalah nomor 3.

```
SQL> EXEC :task_id := rdsadmin.rdsadmin_adrci_util.create_adrci_package(problem_id=>3);

PL/SQL procedure successfully completed.
```

Untuk membaca output tugas, panggil `rdsadmin.rds_file_util.read_text_file`, berikan ID tugas sebagai bagian dari nama file. Output menunjukkan bahwa Anda menghasilkan paket insiden `ORA700EVE_20200529212111_COM_1.zip`.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||:task_id||'.log'));

TEXT
--------------------------------------------------------------------------------
2020-05-29 21:21:11.050 UTC [INFO ] The ADRCI package is being created.
2020-05-29 21:21:15.646 UTC [INFO ] Generated package 2 in file /rdsdbdata/log/trace/
ORA700EVE_20200529212111_COM_1.zip, mode complete
2020-05-29 21:21:15.646 UTC [INFO ] The ADRCI package was successfully created.
2020-05-29 21:21:15.646 UTC [INFO ] The task finished successfully.
```

Anda juga dapat mengunduh file log. Untuk informasi selengkapnya, lihat Mengunduh file log basis data.

Menampilkan file jejak

Anda dapat menggunakan fungsi Amazon RDS `rdsadmin.rdsadmin_adrci_util.show_adrci_tracefile` untuk membuat daftar file jejak di bagian direktori jejak dan semua direktori insiden di beranda ADR saat ini. Anda juga dapat menampilkan konten file jejak dan file jejak insiden.

Fungsi ini menggunakan parameter berikut.

Nama parameter	Tipe data	Nilai valid	Default	Diperlukan	Deskripsi
`filename`	varchar2	Nama file jejak	Null	Tidak	Jika nilainya null, fungsi akan menunjukkan semua

Nama parameter	Tipe data	Nilai valid	Default	Diperlukan	Deskripsi
		yang valid			file jejak. Jika nilainya tidak null, fungsi akan menampilkan file yang ditentukan.

Untuk menampilkan file jejak, panggil fungsi `rdsadmin.rdsadmin_adrci_util.show_adrci_tracefile` Amazon RDS.

```
SQL> EXEC :task_id := rdsadmin.rdsadmin_adrci_util.show_adrci_tracefile;

PL/SQL procedure successfully completed.
```

Untuk mencantumkan nama file jejak, panggil prosedur `rdsadmin.rds_file_util.read_text_file` Amazon RDS, dengan memberikan ID tugas sebagai bagian dari nama file.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||:task_id||'.log')) WHERE TEXT LIKE '%/alert_%';

TEXT
--------------------------------------------------------------------
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-28
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-27
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-26
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-25
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-24
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-23
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-22
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-21
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log

9 rows selected.
```

Dalam contoh berikut, Anda membuat output untuk `alert_ORCL.log`.

```
SQL> EXEC :task_id := rdsadmin.rdsadmin_adrci_util.show_adrci_tracefile('diag/rdbms/
orcl_a/ORCL/trace/alert_ORCL.log');
```

```
PL/SQL procedure successfully completed.
```

Untuk membaca file log, panggil `rdsadmin.rds_file_util.read_text_file`. Berikan ID tugas sebagai bagian dari nama file. Output menampilkan 10 baris pertama alert_ORCL.log.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||:task_id||'.log')) WHERE ROWNUM <= 10;

TEXT
--------------------------------------------------------------------------------
2020-05-29 21:24:02.083 UTC [INFO ] The trace files are being displayed.
2020-05-29 21:24:02.128 UTC [INFO ] Thu May 28 23:59:10 2020
Thread 1 advanced to log sequence 2048 (LGWR switch)
  Current log# 3 seq# 2048 mem# 0: /rdsdbdata/db/ORCL_A/onlinelog/o1_mf_3_hbl2p8xs_.log
Thu May 28 23:59:10 2020
Archived Log entry 2037 added for thread 1 sequence 2047 ID 0x5d62ce43 dest 1:
Fri May 29 00:04:10 2020
Thread 1 advanced to log sequence 2049 (LGWR switch)
  Current log# 4 seq# 2049 mem# 0: /rdsdbdata/db/ORCL_A/onlinelog/o1_mf_4_hbl2qgmh_.log
Fri May 29 00:04:10 2020

10 rows selected.
```

Anda juga dapat mengunduh file log. Untuk informasi selengkapnya, lihat Mengunduh file log basis data.

Melakukan berbagai tugas untuk instans DB Oracle

Setelah itu, Anda dapat menemukan cara melakukan berbagai tugas DBA di instans DB Amazon RDS Anda yang menjalankan Oracle. Untuk memberikan pengalaman layanan terkelola, Amazon RDS tidak memberikan akses shell ke instans DB, dan membatasi akses ke sejumlah prosedur dan tabel sistem tertentu yang memerlukan hak istimewa tingkat lanjut.

Topik

- Membuat dan menghapus direktori di ruang penyimpanan data utama
- Membuat daftar file di direktori instans DB
- Membaca file di direktori instans DB
- Mengakses file Opatch

- Mengelola tugas penasihat
- Mengangkut tablespace

Membuat dan menghapus direktori di ruang penyimpanan data utama

Untuk membuat direktori, gunakan prosedur `rdsadmin.rdsadmin_util.create_directory` Amazon RDS. Anda dapat membuat hingga 10.000 direktori, semuanya berada di ruang penyimpanan data utama Anda. Untuk menghapus direktori gunakan prosedur `rdsadmin.rdsadmin_util.drop_directory` Amazon RDS.

Prosedur `create_directory` dan `drop_directory` memiliki parameter yang diperlukan berikut.

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
`p_directory_name`	varchar2	—	Ya	Nama direktori.

Contoh berikut membuat direktori baru dengan nama `PRODUCT_DESCRIPTIONS`.

```
EXEC rdsadmin.rdsadmin_util.create_directory(p_directory_name =>
 'product_descriptions');
```

Direktori data menyimpan nama direktori dengan huruf besar. Anda dapat membuat daftar direktori dengan membuat kueri `DBA_DIRECTORIES`. Sistem memilih nama jalur host aktual secara otomatis. Contoh berikut memperoleh jalur direktori untuk direktori dengan nama `PRODUCT_DESCRIPTIONS`:

```
SELECT DIRECTORY_PATH
  FROM DBA_DIRECTORIES
 WHERE DIRECTORY_NAME='PRODUCT_DESCRIPTIONS';

DIRECTORY_PATH
----------------------------------------
/rdsdbdata/userdirs/01
```

Nama pengguna master untuk instans DB memiliki hak istimewa baca dan tulis di direktori baru, dan dapat memberikan akses ke pengguna lain. Hak istimewa `EXECUTE` tidak tersedia untuk direktori pada instans DB. Direktori dibuat di ruang penyimpanan data utama Anda dan akan mengonsumsi ruang dan bandwidth I/O.

Contoh berikut menghapus direktori dengan nama PRODUCT_DESCRIPTIONS.

```
EXEC rdsadmin.rdsadmin_util.drop_directory(p_directory_name => 'product_descriptions');
```

> **Note**
>
> Anda juga dapat menghapus direktori menggunakan perintah DROP DIRECTORY Oracle SQL.

Menghapus direktori tidak menghapus kontennya. Karena prosedur `rdsadmin.rdsadmin_util.create_directory` dapat menggunakan ulang nama jalur, file di direktori yang dihapus dapat muncul di direktori yang baru dibuat. Sebelum menghapus direktori, sebaiknya Anda menggunakan `UTL_FILE.FREMOVE` untuk menghapus file dari direktori. Untuk informasi lebih lanjut, lihat FREMOVE procedure di dokumentasi Oracle.

Membuat daftar file di direktori instans DB

Untuk membuat daftar file di direktori, gunakan prosedur `rdsadmin.rds_file_util.listdir` Amazon RDS. Prosedur `listdir` memiliki parameter berikut.

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
`p_directory`	varchar2	—	Ya	Nama direktori yang dibuat daftarnya.

Contoh berikut memberikan hak baca/tulis pada direktori PRODUCT_DESCRIPTIONS ke penggunardsadmin, lalu membuat daftar file dalam direktori ini.

```
GRANT READ,WRITE ON DIRECTORY PRODUCT_DESCRIPTIONS TO rdsadmin;
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir(p_directory =>
 'PRODUCT_DESCRIPTIONS'));
```

Membaca file di direktori instans DB

Untuk membaca file teks, gunakan prosedur `rdsadmin.rds_file_util.read_text_file` Amazon RDS. Prosedur `read_text_file` memiliki parameter berikut.

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
p_directory	varchar2	—	Ya	Nama direktori yang memuat file.
p_filename	varchar2	—	Ya	Nama file yang akan dibaca.

Contoh berikut membuat file rice.txt di direktori PRODUCT_DESCRIPTIONS.

```
declare
  fh sys.utl_file.file_type;
begin
  fh := utl_file.fopen(location=>'PRODUCT_DESCRIPTIONS', filename=>'rice.txt',
 open_mode=>'w');
  utl_file.put(file=>fh, buffer=>'AnyCompany brown rice, 15 lbs');
  utl_file.fclose(file=>fh);
end;
/
```

Contoh berikut membaca file rice.txt dari direktori PRODUCT_DESCRIPTIONS.

```
SELECT * FROM TABLE
    (rdsadmin.rds_file_util.read_text_file(
        p_directory => 'PRODUCT_DESCRIPTIONS',
        p_filename  => 'rice.txt'));
```

Mengakses file Opatch

Opatch adalah utilitas Oracle yang mendukung aplikasi dan rollback patch untuk perangkat lunak Oracle. Mekanisme Oracle untuk menentukan patch mana yang telah diterapkan ke basis data adalah perintah opatch lsinventory. Untuk membuka permintaan layanan bagi pelanggan Bring Your Own Licence (BYOL), Oracle mendukung permintaan lsinventory dan terkadang file lsinventory_detail yang dihasilkan oleh Opatch.

Untuk memberikan pengalaman layanan terkelola, Amazon RDS tidak menyediakan akses shell ke Opatch. Sebagai gantinya, lsinventory-*dbv*.txt di direktori BDUMP berisi informasi patch yang terkait dengan versi mesin Anda saat ini. Saat Anda melakukan peningkatan kecil atau

besar, Amazon RDS memperbarui `lsinventory-dbv.txt` dalam waktu satu jam setelah patch diterapkan. Untuk memverifikasi patch yang diterapkan, baca `lsinventory-dbv.txt`. Tindakan ini mirip dengan menjalankan perintah `opatch lsinventory`.

> **Note**
>
> Contoh di bagian ini mengasumsikan bahwa direktori BDUMP diberi nama `BDUMP`. Pada replika baca, nama direktori BDUMP berbeda. Untuk mempelajari cara mendapatkan nama BDUMP dengan mengkueri `V$DATABASE.DB_UNIQUE_NAME` pada replika baca, lihat Membuat daftar file.

File inventaris menggunakan konvensi penamaan Amazon RDS `lsinventory-dbv.txt` dan `lsinventory_detail-dbv.txt`, di mana *dbv* adalah nama lengkap versi DB Anda. File `lsinventory-dbv.txt` tersedia di semua versi DB. `lsinventory_detail-dbv.txt` yang sesuai tersedia pada DB version berikut:

- 19.0.0.0, ru-2020-01.rur-2020-01.r1 atau yang lebih baru
- 12.2.0.1, ru-2020-01.rur-2020-01.r1 atau yang lebih baru
- 12.1.0.2, v19 atau yang lebih baru

Misalnya, jika versi DB Anda adalah 19.0.0.0.ru-2021-07.rur-2021-07.r1, inventaris file Anda memiliki nama berikut.

```
lsinventory-19.0.0.0.ru-2021-07.rur-2021-07.r1.txt
lsinventory_detail-19.0.0.0.ru-2021-07.rur-2021-07.r1.txt
```

Pastikan Anda mengunduh file yang sesuai dengan versi mesin DB terkini milik Anda.

Konsol

Untuk mengunduh file inventaris menggunakan konsol

1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data.
3. Pilih nama instans DB yang memiliki file log yang ingin dilihat.
4. Pilih tab Log & peristiwa.

5. Gulir ke bawah hingga bagian Log.
6. Di bagian Log, cari `lsinventory`.
7. Pilih file yang ingin Anda akses, lalu pilih Unduh.

SQL

Untuk membaca `lsinventory-dbv.txt` di klien SQL, Anda dapat menggunakan pernyataan `SELECT`. Untuk teknik ini, gunakan fungsi `rdsadmin` berikut: `rdsadmin.rds_file_util.read_text_file` atau `rdsadmin.tracefile_listing`.

Pada kueri sampel berikut, ganti *dbv* dengan versi DB Oracle Anda. Misalnya, versi DB Anda mungkin 19.0.0.0.ru-2020-04.rur-2020-04.r1.

```
SELECT text
FROM   TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP', 'lsinventory-dbv.txt'));
```

PL/SQL

Untuk membaca `lsinventory-dbv.txt` di klien SQL, Anda dapat menulis program PL/SQL. Program ini menggunakan `utl_file` untuk membaca file, dan `dbms_output` untuk mencetaknya. Ini adalah paket yang diberikan oleh Oracle.

Pada program sampel berikut, ganti *dbv* dengan versi DB Oracle Anda. Misalnya, versi DB Anda mungkin 19.0.0.0.ru-2020-04.rur-2020-04.r1.

```
SET SERVEROUTPUT ON
DECLARE
  v_file              SYS.UTL_FILE.FILE_TYPE;
  v_line              VARCHAR2(1000);
  v_oracle_home_type  VARCHAR2(1000);
  c_directory         VARCHAR2(30) := 'BDUMP';
  c_output_file       VARCHAR2(30) := 'lsinventory-dbv.txt';
BEGIN
  v_file := SYS.UTL_FILE.FOPEN(c_directory, c_output_file, 'r');
  LOOP
    BEGIN
      SYS.UTL_FILE.GET_LINE(v_file, v_line,1000);
      DBMS_OUTPUT.PUT_LINE(v_line);
    EXCEPTION
```

```
      WHEN no_data_found THEN
        EXIT;
    END;
  END LOOP;
END;
/
```

Atau kueri `rdsadmin.tracefile_listing`, dan gulung output ke file. Contoh berikut menggulung output ke `/tmp/tracefile.txt`.

```
SPOOL /tmp/tracefile.txt
SELECT *
FROM   rdsadmin.tracefile_listing
WHERE  FILENAME LIKE 'lsinventory%';
SPOOL OFF;
```

Mengelola tugas penasihat

Oracle Database mencakup sejumlah penasihat. Setiap penasihat mendukung tugas otomatis dan manual. Anda dapat menggunakan prosedur di paket `rdsadmin.rdsadmin_util` untuk mengelola beberapa tugas penasihat.

Prosedur tugas penasihat tersedia dalam versi mesin berikut:

- Oracle Database 21c (21.0.0)
- Versi 19.0.0.0.ru-2021-01.rur-2021-01.r1 dan Oracle Database versi 19c yang lebih baru

  Untuk informasi selengkapnya, lihat Versi 19.0.0.0.ru-2021-01.rur-2021-01.r1 di Catatan Rilis Amazon RDS for Oracle.
- Versi 12.2.0.1.ru-2021-01.rur-2021-01.rur-2021-01.r1 dan versi Oracle Database 12c (Rilis 2) 12.2.0.1 yang lebih baru

  Untuk informasi selengkapnya, lihat Versi 12.2.0.1.ru-2021-01.rur-2021-01.r1 di Catatan Rilis Amazon RDS for Oracle.

Topik

- Mengatur parameter untuk tugas penasihat
- Menonaktifkan AUTO_STATS_ADVISTOR_TASK

- Mengaktifkan kembali AUTO_STATS_ADVISTOR_TASK

Mengatur parameter untuk tugas penasihat

Untuk mengatur parameter untuk beberapa tugas penasihat, gunakan prosedur `rdsadmin.rdsadmin_util.advisor_task_set_parameter` Amazon RDS. Prosedur `advisor_task_set_parameter` memiliki parameter berikut.

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
`p_task_name`	varchar2	—	Ya	Nama tugas penasihat yang parameternya ingin Anda ubah. Nilai berikut ini valid: • `AUTO_STATS_ADVISOR_TASK` • `INDIVIDUAL_STATS_ADVISOR_TASK` • `SYS_AUTO_SPM_EVOLVE_TASK` • `SYS_AUTO_SQL_TUNING_TASK`

`p_parameter`	varchar2	—	Ya	Nama parameter tugas. Untuk menemukan parameter yang valid untuk tugas penasihat, jalankan kueri berikut. Ganti *`p_task_name`* dengan nilai yang valid untuk `p_task_name` :
				`COL PARAMETER_NAME FORMAT a30`
				`COL PARAMETER_VALUE FORMAT a30`
				`SELECT PARAMETER_NAME, PARAMETER_VALUE`
				`FROM DBA_ADVISOR_PARAMETERS`
				`WHERE TASK_NAME='` *`p_task_name`* `'`
				`AND PARAMETER_VALUE != 'UNUSED'`
				`ORDER BY PARAMETER_NAME;`

`p_value`	varchar2	—	Ya	Nilai untuk parameter tugas. Untuk menemukan nilai yang valid untuk parameter tugas, jalankan kueri berikut. Ganti *`p_task_name`* dengan nilai yang valid untuk `p_task_name` :

Nama parameter	Tipe data	Default	Diperluk n	Deskripsi

				`COL PARAMETER_NAME FORMAT a30`
				`COL PARAMETER_VALUE FORMAT a30`
				`SELECT PARAMETER_NAME, PARAMETER _VALUE`
				`FROM DBA_ADVISOR_PARAMETERS`
				`WHERE TASK_NAME='  p_task_name '`
				`AND PARAMETER_VALUE != 'UNUSED'`
				`ORDER BY PARAMETER_NAME;`

Program PL/SQL berikut menetapkan `ACCEPT_PLANS` menjadi `FALSE` untuk `SYS_AUTO_SPM_EVOLVE_TASK`. Tugas otomatis SQL Plan Management memverifikasi rencana dan membuat laporan temuan, tetapi tidak mengembangkan rencana secara otomatis. Anda dapat menggunakan laporan untuk mengidentifikasi garis dasar rencana SQL baru dan menerimanya secara manual.

```
BEGIN
  rdsadmin.rdsadmin_util.advisor_task_set_parameter(
    p_task_name => 'SYS_AUTO_SPM_EVOLVE_TASK',
    p_parameter => 'ACCEPT_PLANS',
    p_value     => 'FALSE');
END;
```

Program PL/SQL berikut menetapkan `EXECUTION_DAYS_TO_EXPIRE` menjadi `10` untuk `AUTO_STATS_ADVISOR_TASK`. Tugas `AUTO_STATS_ADVISOR_TASK` yang telah ditetapkan berjalan secara otomatis pada masa pemeliharaan sekali per hari. Contoh yang ada menetapkan periode retensi untuk eksekusi tugas sampai 10 hari.

```
BEGIN
  rdsadmin.rdsadmin_util.advisor_task_set_parameter(
    p_task_name => 'AUTO_STATS_ADVISOR_TASK',
    p_parameter => 'EXECUTION_DAYS_TO_EXPIRE',
    p_value     => '10');
END;
```

Menonaktifkan AUTO_STATS_ADVISTOR_TASK

Untuk menonaktifkan `AUTO_STATS_ADVISOR_TASK`, gunakan prosedur `rdsadmin.rdsadmin_util.advisor_task_drop` Amazon RDS. Prosedur `advisor_task_drop` menerima parameter berikut.

> **Note**
>
> Prosedur ini tersedia di Oracle Database 12c Rilis 2 (12.2.0.1) dan yang lebih baru.

Nama parameter	Tipe data	Default	Diperluk n	Deskripsi
`p_task_name`	varchar2	—	Ya	Nama tugas penasihat yang akan dinonakti fkan. Satu-satunya nilai yang valid adalah `AUTO_STATS_ADVISOR_TASK` .

Perintah berikut menghapus `AUTO_STATS_ADVISOR_TASK`.

```
EXEC rdsadmin.rdsadmin_util.advisor_task_drop('AUTO_STATS_ADVISOR_TASK')
```

Anda dapat mengaktifkan kembali `AUTO_STATS_ADVISOR_TASK` menggunakan `rdsadmin.rdsadmin_util.dbms_stats_init`.

Mengaktifkan kembali AUTO_STATS_ADVISTOR_TASK

Untuk mengaktifkan kembali `AUTO_STATS_ADVISOR_TASK`, gunakan prosedur `rdsadmin.rdsadmin_util.dbms_stats_init` Amazon RDS. Prosedur `dbms_stats_init` tidak menggunakan parameter.

Perintah berikut mengaktifkan kembali `AUTO_STATS_ADVISOR_TASK`.

```
EXEC rdsadmin.rdsadmin_util.dbms_stats_init()
```

Mengangkut tablespace

Gunakan paket Amazon RDS `rdsadmin.rdsadmin_transport_util` untuk menyalin satu set tablespace dari basis data Oracle on-premise ke instans DB RDS for Oracle. Pada tingkat fisik, fitur

tablespace yang dapat diangkut secara bertahap menyalin file data sumber dan file metadata ke instans target Anda. Anda dapat mentransfer file menggunakan Amazon EFS atau Amazon S3. Untuk informasi selengkapnya, lihat Bermigrasi menggunakan tablespace yang dapat dipindahkan Oracle.

Topik

- Mengimpor tablespace yang diangkut ke instans DB Anda
- Mengimpor metadata tablespace yang dapat diangkut ke instans DB Anda
- Mencantumkan file tanpa induk setelah impor tablespace
- Menghapus file data tanpa induk setelah impor tablespace

Mengimpor tablespace yang diangkut ke instans DB Anda

Gunakan prosedur `rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces` untuk memulihkan tablespace yang sebelumnya telah Anda ekspor dari instans DB sumber. Pada fase pengangkutan, Anda mencadangkan tablespace hanya baca, mengekspor metadata Data Pump, mentransfer file-file ini ke instans DB target Anda, kemudian mengimpor tablespace. Untuk informasi selengkapnya, lihat Tahap 4: Pindahkan tablespace.

Sintaks

```
FUNCTION import_xtts_tablespaces(
    p_tablespace_list IN CLOB,
    p_directory_name  IN VARCHAR2,
    p_platform_id     IN NUMBER DEFAULT 13,
    p_parallel        IN INTEGER DEFAULT 0) RETURN VARCHAR2;
```

Parameter

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
`p_tablespace_list`	`CLOB`	—	Ya	Daftar tablespace yang akan diimpor.
`p_directory_name`	`VARCHAR2`	—	Ya	Direktori yang berisi cadangan tablespace.

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
p_platform_id	NUMBER	13	Tidak	Berikan ID platform yang cocok dengan yang ditentukan selama fase pencadangan. Untuk menemukan daftar platform, lakukan kueri V$TRANSPORTABLE_PLATFORM . Platform default adalah Linux x86 64-bit, yang sedikit endian.
p_parallel	INTEGER	0	Tidak	Tingkat paralelisme. Secara default, paralelisme dinonaktifkan.

Contoh-contoh

Contoh berikut mengimpor *TBS1*, *TBS2*, dan *TBS3* tablespace dari direktori *DATA_PUMP_DIR*. Platform sumbernya adalah Sistem berbasis AIX (64-bit), yang memiliki ID platform. 6 Anda dapat menemukan ID platform dengan melakukan kueriV$TRANSPORTABLE_PLATFORM.

```
VAR task_id CLOB

BEGIN
  :task_id:=rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces(
        'TBS1,TBS2,TBS3',
        'DATA_PUMP_DIR',
        p_platform_id => 6);
END;
/

PRINT task_id
```

Mengimpor metadata tablespace yang dapat diangkut ke instans DB Anda

Gunakan prosedur `rdsadmin.rdsadmin_transport_util.import_xtts_metadata` untuk mengimpor metadata tablespace yang dapat diangkut ke RDS Anda untuk instans DB Oracle. Selama operasi, status impor metadata ditunjukkan pada tabel `rdsadmin.rds_xtts_operation_info`. Untuk informasi selengkapnya, lihat Langkah 5: Impor metadata tablespace pada instans DB target Anda.

Sintaks

```
PROCEDURE import_xtts_metadata(
    p_datapump_metadata_file IN SYS.DBA_DATA_FILES.FILE_NAME%TYPE,
    p_directory_name         IN VARCHAR2,
    p_exclude_stats          IN BOOLEAN DEFAULT FALSE,
    p_remap_tablespace_list  IN CLOB DEFAULT NULL,
    p_remap_user_list        IN CLOB DEFAULT NULL);
```

Parameter

Nama parameter	Tipe data	Default	Diperluka n	Deskripsi
`p_datapum p_metadat a_file`	`SYS.DBA_DATA_FILES .FILE_NAME%TYPE`	—	Ya	Nama file Oracle Data Pump yang berisi metadata untuk tablespace yang dapat diangkut.
`p_directo ry_name`	`VARCHAR2`	—	Ya	Direktori yang berisi file Data Pump.
`p_exclude_stats`	`BOOLEAN`	`FALSE`	Tidak	Bendera yang menunjukkan apakah akan mengecualikan statistik.
`p_remap_t ablespace_list`	`CLOB`	`NULL`	Tidak	Daftar tablespace yang akan dipetakan ulang selama impor metadata.

Nama parameter	Tipe data	Default	Diperluka n	Deskripsi
				Gunakan format *from_tbs*:*to_tbs*. Sebagai contoh, tentukan `users:user_data` .
`p_remap_user_list`	`CLOB`	NULL	Tidak	Daftar skema pengguna yang akan dipetakan ulang selama impor metadata. Gunakan format *from_schema_name* :*to_schema_name* . Sebagai contoh, tentukan `hr:human_resources` .

Contoh-contoh

Contoh mengimpor metadata tablespace dari file *xttdump.dmp*, yang terletak di direktori *DATA_PUMP_DIR*.

```
BEGIN
  rdsadmin.rdsadmin_transport_util.import_xtts_metadata('xttdump.dmp','DATA_PUMP_DIR');
END;
/
```

Mencantumkan file tanpa induk setelah impor tablespace

Gunakan prosedur `rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files` untuk mencantumkan file data tanpa induk setelah impor tablespace. Setelah Anda mengidentifikasi file data, Anda dapat menghapusnya dengan memanggil `rdsadmin.rdsadmin_transport_util.cleanup_incomplete_xtts_import`.

Sintaks

```
FUNCTION list_xtts_orphan_files RETURN xtts_orphan_files_list_t PIPELINED;
```

Contoh-contoh

Contoh berikut menjalankan prosedur `rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files`. Output menunjukkan dua file data tanpa induk.

```
SQL> SELECT * FROM TABLE(rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files);

FILENAME       FILESIZE
-------------- ---------
datafile_7.dbf 104865792
datafile_8.dbf 104865792
```

Menghapus file data tanpa induk setelah impor tablespace

Gunakan prosedur `rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files` untuk menghapus file data tanpa induk setelah impor tablespace.
Menjalankan perintah ini akan menghasilkan file log yang menggunakan format nama `rds-xtts-delete_xtts_orphaned_files-`*`YYYY-MM-DD.HH24-MI-SS.FF`*`.log` dalam direktori BDUMP. Gunakan prosedur `rdsadmin.rdsadmin_transport_util.cleanup_incomplete_xtts_import` untuk menemukan file tanpa induk. Anda dapat membaca file log dengan memanggil prosedur `rdsadmin.rds_file_util.read_text_file`. Untuk informasi selengkapnya, lihat Tahap 6: Bersihkan file sisa.

Sintaks

```
PROCEDURE cleanup_incomplete_xtts_import(
    p_directory_name IN VARCHAR2);
```

Parameter

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
p_directory_name	VARCHAR2	—	Ya	Direktori yang berisi file data tanpa induk.

Contoh-contoh

Contoh berikut menghapus file data tanpa induk di *DATA_PUMP_DIR.*

```
BEGIN
  rdsadmin.rdsadmin_transport_util.cleanup_incomplete_xtts_import('DATA_PUMP_DIR');
END;
/
```

Contoh berikut membaca file log yang dihasilkan oleh perintah sebelumnya.

```
SELECT *
FROM TABLE(rdsadmin.rds_file_util.read_text_file(
       p_directory => 'BDUMP',
       p_filename  => 'rds-xtts-
delete_xtts_orphaned_files-2023-06-01.09-33-11.868894000.log'));

TEXT
--------------------------------------------------------------------------------
orphan transported datafile datafile_7.dbf deleted.
orphan transported datafile datafile_8.dbf deleted.
```

Mengonfigurasi fitur RDS for Oracle

RDS for Oracle mendukung berbagai fitur canggih, termasuk HugePages, penyimpanan instans, dan jenis data yang diperluas.

Topik

Menyimpan data sementara di penyimpanan instans RDS for Oracle

Gunakan penyimpanan instans untuk ruang tabel sementara dan Database Smart Flash Cache (cache flash) pada kelas instans DB RDS for Oracle yang didukung.

Topik

Gambaran umum penyimpanan instans RDS for Oracle

Penyimpanan instans menyediakan penyimpanan tingkat blok sementara untuk instans DB RDS for Oracle. Anda dapat menggunakan penyimpanan instans sebagai penyimpanan sementara informasi yang sering berubah.

Penyimpanan instans didasarkan pada perangkat Non-Volatile Memory Express (NVMe) yang secara fisik terpasang pada komputer host. Penyimpanan tersebut dioptimalkan untuk latensi rendah, performa I/O acak, dan throughput baca berurutan.

Ukuran penyimpanan instans bervariasi menurut jenis instans DB. Untuk informasi selengkapnya tentang penyimpanan instans, lihat penyimpanan instans Amazon EC2 di Panduan Pengguna Amazon Elastic Compute Cloud untuk Instans Linux.

Topik

- Jenis data di penyimpanan instans RDS for Oracle
- Manfaat penyimpanan instans RDS for Oracle
- Kelas instans yang didukung untuk penyimpanan instans RDS for Oracle
- Versi mesin yang didukung untuk penyimpanan instans RDS for Oracle
- Wilayah AWS yang didukung untuk penyimpanan instans RDS for Oracle
- Biaya penyimpanan instans RDS for Oracle

Jenis data di penyimpanan instans RDS for Oracle

Anda dapat menempatkan jenis data sementara RDS for Oracle berikut di penyimpanan instans:

Ruang tabel sementara

Oracle Database menggunakan ruang tabel sementara untuk menyimpan hasil kueri perantara yang tidak sesuai dengan memori. Kueri yang lebih besar dapat menghasilkan data perantara dalam jumlah besar yang perlu di-cache sementara, tetapi tidak perlu disimpan. Secara khusus, ruang tabel sementara berguna untuk pengurutan, agregasi hash, dan gabungan. Jika instans DB RDS for Oracle menggunakan Enterprise Edition atau Standard Edition 2, Anda dapat menempatkan ruang tabel sementara di penyimpanan instans.

Cache flash

Cache flash meningkatkan performa pembacaan acak blok tunggal di jalur konvensional. Praktik terbaiknya adalah mengukur cache untuk mengakomodasi sebagian besar kumpulan data aktif Anda. Jika instans DB RDS for Oracle menggunakan Enterprise Edition, Anda dapat menempatkan cache flash di penyimpanan instans.

Secara default, penyimpanan instans dikonfigurasi untuk ruang tabel sementara tetapi tidak untuk cache flash. Anda tidak dapat menempatkan file data Oracle dan file log basis data di penyimpanan instans.

Manfaat penyimpanan instans RDS for Oracle

Anda dapat mempertimbangkan menggunakan penyimpanan instans untuk menyimpan file sementara dan cache yang dapat Anda tanggung jika hilang. Jika Anda ingin meningkatkan performa DB, atau jika beban kerja yang meningkat menyebabkan masalah performa untuk penyimpanan Amazon EBS Anda, pertimbangkan untuk menskalakan ke kelas instans yang mendukung penyimpanan instans.

Dengan menempatkan ruang tabel sementara dan cache flash di penyimpanan instans, Anda mendapatkan manfaat berikut:

- Latensi baca lebih rendah
- Throuput lebih tinggi
- Berkurangnya beban pada volume Amazon EBS Anda
- Biaya penyimpanan dan snapshot lebih rendah karena pengurangan beban Amazon EBS
- Lebih sedikit kebutuhan untuk menyediakan IOPS tinggi, sehingga dapat menurunkan biaya keseluruhan

Dengan menempatkan ruang tabel sementara di penyimpanan instans, Anda menghadirkan peningkatan performa langsung ke kueri yang menggunakan ruang sementara. Saat Anda menempatkan cache flash di penyimpanan instans, pembacaan blok yang di-cache biasanya memiliki latensi yang jauh lebih rendah daripada pembacaan Amazon EBS. Cache flash perlu “dipanaskan” sebelum memberikan manfaat performa. Cache memanas dengan sendirinya karena basis data menulis blok ke cache flash seiring bertambahnya usia cache buffer basis data.

> **Note**
>
> Dalam beberapa kasus, cache flash menyebabkan overhead performa karena manajemen cache. Sebelum mengaktifkan cache flash di lingkungan produksi, sebaiknya Anda menganalisis beban kerja dan menguji cache di lingkungan pengujian.

Kelas instans yang didukung untuk penyimpanan instans RDS for Oracle

Amazon RDS mendukung penyimpanan instans untuk kelas instans DB berikut:

- db.m5d
- db.r5d

- db.x2idn
- db.x2iedn

RDS for Oracle mendukung kelas instans DB sebelumnya hanya untuk model lisensi BYOL. Untuk informasi selengkapnya, lihat Kelas instans RDS for Oracle yang didukung dan Bawa Lisensi Sendiri (BYOL).

Untuk melihat total penyimpanan instans untuk jenis instans DB yang didukung, jalankan perintah berikut di AWS CLI.

Example

```
aws ec2 describe-instance-types \
  --filters "Name=instance-type,Values=*5d.*large*" \
  --query "InstanceTypes[?contains(InstanceType,'m5d')||contains(InstanceType,'r5d')]
[InstanceType, InstanceStorageInfo.TotalSizeInGB]" \
  --output table
```

Perintah sebelumnya mengembalikan ukuran perangkat mentah untuk penyimpanan instans. RDS for Oracle menggunakan sebagian kecil ruang ini untuk konfigurasi. Ruang di penyimpanan instans yang tersedia untuk ruang tabel sementara atau cache flash sedikit lebih kecil.

Versi mesin yang didukung untuk penyimpanan instans RDS for Oracle

Penyimpanan instans didukung untuk versi mesin RDS for Oracle:

- 21.0.0.0.ru-2022-01.rur-2022-01.r1 atau versi Oracle Database 21c yang lebih tinggi
- 19.0.0.0.ru-2021-10.rur-2021-10.r1 atau versi Oracle Database 19c yang lebih tinggi

Wilayah AWS yang didukung untuk penyimpanan instans RDS for Oracle

Penyimpanan instans tersedia di semua Wilayah AWS tempat satu atau beberapa jenis instans ini didukung. Untuk informasi selengkapnya tentang kelas instans db.m5d dan db.r5d, lihat Kelas instans DB . Untuk informasi selengkapnya tentang kelas instans yang didukung oleh Amazon RDS for Oracle, lihat Kelas instans RDS for Oracle.

Biaya penyimpanan instans RDS for Oracle

Biaya penyimpanan instans dimasukkan ke dalam biaya penyimpanan instans yang mengaktifkan instans. Anda tidak dikenakan biaya tambahan dengan mengaktifkan penyimpanan instans pada

instans DB RDS for Oracle. Untuk informasi selengkapnya tentang penyimpanan instans yang mengaktifkan instans, lihat Kelas instans yang didukung untuk penyimpanan instans RDS for Oracle.

Mengaktifkan penyimpanan instans RDS for Oracle

Untuk mengaktifkan penyimpanan instans data sementara RDS for Oracle, lakukan salah satu tindakan berikut:

- Buat instans DB RDS for Oracle menggunakan kelas instans yang didukung. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.
- Ubah instans DB RDS for Oracle yang ada untuk menggunakan kelas instans yang didukung. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Mengonfigurasi penyimpanan instans RDS for Oracle

Secara default, 100% ruang penyimpanan instans dialokasikan ke ruang tabel sementara. Untuk mengonfigurasi penyimpanan instans guna mengalokasikan ruang ke cache flash dan ruang tabel sementara, atur parameter berikut dalam grup parameter untuk instans Anda:

db_flash_cache_size={DBInstanceStore*{0,2,4,6,8,10}/10}

Parameter ini menentukan jumlah ruang penyimpanan yang dialokasikan untuk cache flash. Parameter ini hanya berlaku untuk Oracle Database Enterprise Edition. Nilai defaultnya adalah `{DBInstanceStore*0/10}`. Jika Anda menetapkan nilai selain nol untuk `db_flash_cache_size`, instans RDS for Oracle Anda mengaktifkan cache flash setelah Anda memulai ulang instans.

rds.instance_store_temp_size={DBInstanceStore*{0,2,4,6,8,10}/10}

Parameter ini menentukan jumlah ruang penyimpanan yang dialokasikan untuk ruang tabel sementara. Nilai defaultnya adalah `{DBInstanceStore*10/10}`. Parameter ini dapat diubah untuk Oracle Database Enterprise Edition dan baca-saja untuk Standard Edition 2. Jika Anda menetapkan nilai bukan nol untuk `rds.instance_store_temp_size`, Amazon RDS mengalokasikan ruang di penyimpanan instans untuk ruang tabel sementara.

Anda dapat mengatur parameter `db_flash_cache_size` dan `rds.instance_store_temp_size` untuk instans DB yang tidak menggunakan penyimpanan instans. Dalam hal ini, kedua pengaturan mengevaluasi `0`, yang menonaktifkan fitur. Dalam kasus ini, Anda dapat menggunakan grup parameter yang sama untuk ukuran instans yang berbeda dan

untuk instans yang tidak menggunakan penyimpanan instans. Jika Anda mengubah parameter ini, pastikan untuk mem-boot ulang instans terkait sehingga perubahan dapat diterapkan.

> **Important**
>
> Jika Anda mengalokasikan ruang untuk ruang tabel sementara, Amazon RDS tidak membuat ruang tabel sementara secara otomatis. Untuk mempelajari cara membuat ruang tabel sementara di penyimpanan instans, lihat Membuat tablespace sementara di penyimpanan instans.

Nilai gabungan dari parameter sebelumnya tidak boleh melebihi 10/10, atau 100%. Tabel berikut mengilustrasikan pengaturan parameter yang valid dan tidak valid.

Pengaturan db_flash_cache_size	Pengaturan rds.instance_store_temp_size	Penjelasan
db_flash_cache_size={DBInstanceStore*0/10}	rds.instance_store_temp_size={DBInstanceStore*10/10}	Ini adalah konfigurasi yang valid untuk semua edisi Oracle Database. Amazon RDS mengalokasikan 100% ruang penyimpanan instans ke ruang tabel sementara. Ini adalah default.
db_flash_cache_size={DBInstanceStore*10/10}	rds.instance_store_temp_size={DBInstanceStore*0/10}	Parameter ini hanya berlaku untuk Oracle Database Enterpris

Pengaturan db_flash_cache_size	Pengaturan rds.instance_store_temp_size	Penjelasan
		e Edition. Amazon RDS mengalokasikan 100% ruang penyimpanan instans ke cache flash.
db_flash_cache_size={DBInstanceStore*2/10}	rds.instance_store_temp_size={DBInstanceStore*8/10}	Parameter ini hanya berlaku untuk Oracle Database Enterprise Edition. Amazon RDS mengalokasikan 20% ruang penyimpanan instans ke cache flash, dan 80% ruang penyimpanan instans ke ruang tabel sementara.

Pengaturan db_flash_cache_size	Pengaturan rds.instance_store_temp_size	Penjelasan
db_flash_cache_size={DBInstanceStore*6/10}	rds.instance_store_temp_size={DBInstanceStore*4/10}	Parameter ini hanya berlaku untuk Oracle Database Enterprise Edition. Amazon RDS mengalokasikan 60% ruang penyimpanan instans ke cache flash, dan 40% ruang penyimpanan instans ke ruang tabel sementara.

Pengaturan db_flash_cache_size	Pengaturan rds.instance_store_temp_size	Penjelasan
db_flash_cache_size={DBInstanceStore*2/10}	rds.instance_store_temp_size={DBInstanceStore*4/10}	Parameter ini hanya berlaku untuk Oracle Database Enterprise Edition. Amazon RDS mengalokasikan 20% ruang penyimpanan instans ke cache flash, dan 40% ruang penyimpanan instans ke ruang tabel sementara.
db_flash_cache_size={DBInstanceStore*8/10}	rds.instance_store_temp_size={DBInstanceStore*8/10}	Ini adalah konfigurasi yang tidak valid karena persentase gabungan ruang penyimpanan instans melebihi 100%. Dalam kasus demikian, Amazon RDS gagal dalam upaya tersebut.

Pertimbangan saat mengubah jenis instans DB

Jika Anda mengubah jenis instans DB Anda, hal tersebut dapat memengaruhi konfigurasi cache flash atau ruang tabel sementara di penyimpanan instans. Pertimbangkan pengubahan berikut serta efeknya:

Anda meningkatkan atau menurunkan skala instans DB yang mendukung penyimpanan instans.

Nilai berikut bertambah atau berkurang secara proporsional dengan ukuran penyimpanan instans baru:

- Ukuran cache flash baru.
- Ruang tersebut dialokasikan ke ruang tabel sementara yang berada di penyimpanan instans.

Misalnya, pengaturan `db_flash_cache_size={DBInstanceStore*6/10}` pada instans db.m5d.4xlarge menyediakan sekitar 340 GB ruang cache flash. Jika Anda meningkatkan jenis instans ke db.m5d.8xlarge, ruang cache flash meningkat menjadi sekitar 680 GB.

Anda mengubah instans DB yang tidak menggunakan penyimpanan instans menjadi instans yang menggunakan penyimpanan instans.

Jika `db_flash_cache_size` diatur ke nilai yang lebih besar dari `0`, cache flash dikonfigurasi. Jika `rds.instance_store_temp_size` diatur ke nilai yang lebih besar dari `0`, ruang penyimpanan instans dialokasikan untuk digunakan oleh ruang tabel sementara. RDS for Oracle tidak memindahkan tempfile ke penyimpanan instans secara otomatis. Untuk informasi tentang menggunakan ruang yang dialokasikan, lihat Membuat tablespace sementara di penyimpanan instans atau Menambahkan tempfile ke penyimpanan instans di replika baca.

Anda mengubah instans DB yang menggunakan penyimpanan instans menjadi instans yang tidak menggunakan penyimpanan instans.

Dalam hal ini, RDS for Oracle menghapus cache flash. RDS membuat ulang tempfile yang saat ini berada di penyimpanan instans di volume Amazon EBS. Ukuran maksimum tempfile baru adalah ukuran parameter `rds.instance_store_temp_size` sebelumnya.

Menggunakan penyimpanan instans pada replika baca Oracle

Replika baca mendukung cache flash dan ruang tabel sementara di penyimpanan instans. Sementara cache flash berfungsi dengan cara yang sama seperti pada instans DB primer, perhatikan perbedaan berikut untuk ruang tabel sementara:

- Anda tidak dapat membuat ruang tabel sementara yang ada di replika baca. Jika Anda membuat ruang tabel sementara baru pada instans primer, RDS for Oracle mereplikasi informasi ruang tabel tanpa tempfile. Untuk menambahkan tempfile baru, gunakan salah satu teknik berikut:
  - Gunakan prosedur Amazon RDS `rdsadmin.rdsadmin_util.add_inst_store_tempfile`. RDS for Oracle membuat tempfile di penyimpanan instans pada replika baca Anda, dan menambahkannya ke ruang tabel sementara yang ditentukan.
  - Jalankan perintah `ALTER TABLESPACE … ADD TEMPFILE`. RDS for Oracle menempatkan tempfile di penyimpanan Amazon EBS.

  > **Note**
  >
  > Jenis penyimpanan dan ukuran tempfile dapat berbeda pada instans DB primer dan replika baca.

- Anda dapat mengelola pengaturan ruang tabel sementara default hanya pada instans DB primer. RDS for Oracle mereplikasi pengaturan ke semua replika baca.
- Anda dapat mengonfigurasi grup ruang tabel sementara hanya pada instans DB primer. RDS for Oracle mereplikasi pengaturan ke semua replika baca.

Mengonfigurasi grup ruang tabel sementara di penyimpanan instans dan Amazon EBS

Anda dapat mengonfigurasi grup ruang tabel sementara untuk menyertakan ruang tabel sementara pada penyimpanan instans dan Amazon EBS. Teknik ini berguna ketika Anda menginginkan penyimpanan sementara lebih banyak daripada yang diperbolehkan oleh pengaturan maksimal `rds.instance_store_temp_size`.

Saat Anda mengonfigurasi grup ruang tabel sementara di penyimpanan instans dan Amazon EBS, kedua ruang tabel memiliki karakteristik performa yang sangat berbeda. Oracle Database memilih ruang tabel untuk melayani kueri berdasarkan algoritma internal. Oleh karena itu, performa kueri yang serupa dapat berbeda.

Biasanya, Anda membuat ruang tabel sementara di penyimpanan instans sebagai berikut:

1. Buat ruang tabel sementara di penyimpanan instans.
2. Tetapkan ruang tabel baru sebagai ruang tabel sementara default basis data.

Jika ukuran raung tabel di penyimpanan instans tidak mencukupi, Anda dapat membuat penyimpanan sementara tambahan sebagai berikut:

1. Tetapkan ruang tabel sementara di penyimpanan instans ke grup ruang tabel sementara.
2. Buat ruang tabel sementara baru di Amazon EBS jika belum ada.
3. Tetapkan ruang tabel sementara di Amazon EBS ke grup ruang tabel yang sama yang menyertakan ruang tabel penyimpanan instans.
4. Tetapkan grup ruang tabel sebagai ruang tabel sementara default.

Contoh berikut mengasumsikan bahwa ukuran ruang tabel sementara di penyimpanan instans tidak memenuhi persyaratan aplikasi Anda. Contoh tersebut membuat ruang tabel sementara `temp_in_inst_store` di penyimpanan instans, menetapkannya ke grup ruang tabel `temp_group`, menambahkan ruang tabel Amazon EBS yang sudah ada yang bernama `temp_in_ebs` ke grup ini, dan menetapkan grup ini sebagai ruang tabel sementara default.

```
SQL> EXEC rdsadmin.rdsadmin_util.create_inst_store_tmp_tblspace('temp_in_inst_store');

PL/SQL procedure successfully completed.

SQL> ALTER TABLESPACE temp_in_inst_store TABLESPACE GROUP temp_group;

Tablespace altered.

SQL> ALTER TABLESPACE temp_in_ebs TABLESPACE GROUP temp_group;

Tablespace altered.

SQL> EXEC rdsadmin.rdsadmin_util.alter_default_temp_tablespace('temp_group');

PL/SQL procedure successfully completed.

SQL> SELECT * FROM DBA_TABLESPACE_GROUPS;

GROUP_NAME                     TABLESPACE_NAME
------------------------------ ------------------------------
TEMP_GROUP                     TEMP_IN_EBS
TEMP_GROUP                     TEMP_IN_INST_STORE

SQL> SELECT PROPERTY_VALUE FROM DATABASE_PROPERTIES WHERE
 PROPERTY_NAME='DEFAULT_TEMP_TABLESPACE';
```

```
PROPERTY_VALUE
--------------
TEMP_GROUP
```

Menghapus penyimpanan instans RDS for Oracle

Untuk menghapus penyimpanan instans, ubah instans DB RDS for Oracle Anda untuk menggunakan jenis instans yang tidak mendukung penyimpanan instans, seperti db.m5 atau db.r5.

Mengaktifkan HugePages untuk instans RDS for Oracle

Amazon RDS for Oracle mendukung HugePages kernel Linux untuk meningkatkan skalabilitas basis data. HugePages menghasilkan tabel halaman yang lebih kecil dan memakan lebih sedikit waktu CPU untuk manajemen memori, sehingga meningkatkan performa instans basis data yang besar. Untuk informasi lebih lanjut, lihat Overview of HugePages dalam dokumentasi Oracle.

Anda dapat menggunakan HugePages dengan semua versi dan edisi RDS for Oracle yang didukung.

Parameter `use_large_pages` mengontrol apakah HugePages diaktifkan untuk suatu instans DB. Kemungkinan pengaturan untuk parameter ini adalah `ONLY`, `FALSE`, dan `{DBInstanceClassHugePagesDefault}`. Parameter `use_large_pages` diatur menjadi `{DBInstanceClassHugePagesDefault}` dalam grup parameter DB default untuk Oracle.

Untuk mengontrol apakah HugePages diaktifkan secara otomatis untuk instans DB, Anda dapat menggunakan variabel rumus `DBInstanceClassHugePagesDefault` dalam grup parameter. Nilainya ditentukan sebagai berikut:

- Untuk kelas instans DB yang disebutkan dalam tabel berikut, `DBInstanceClassHugePagesDefault` selalu mengevaluasi menjadi `FALSE` secara default, dan `use_large_pages` mengevaluasi menjadi `FALSE`. Anda dapat mengaktifkan HugePages secara manual untuk kelas instans DB ini jika kelas instans DB memiliki memori setidaknya 14 GiB.
- Untuk kelas instans DB yang tidak disebutkan dalam tabel berikut, jika kelas instans DB memiliki memori kurang dari 14 GiB, `DBInstanceClassHugePagesDefault` selalu mengevaluasi menjadi `FALSE`. Selain itu, `use_large_pages` mengevaluasi menjadi `FALSE`.
- Untuk kelas instans DB yang tidak disebutkan dalam tabel berikut, jika kelas instans memiliki memori setidaknya 14 GiB dan kurang dari 100 GiB, `DBInstanceClassHugePagesDefault` mengevaluasi menjadi `TRUE` secara default. Selain itu, `use_large_pages` mengevaluasi menjadi

ONLY. Anda dapat mematikan HugePages secara manual dengan mengatur `use_large_pages` ke `FALSE`.

- Untuk kelas instans DB yang tidak disebutkan dalam tabel berikut, jika kelas instans memiliki memori kurang dari 100 GiB, `DBInstanceClassHugePagesDefault` selalu mengevaluasi menjadi `TRUE`. Selain itu ,`use_large_pages` mengevaluasi menjadi `ONLY` dan HugePages tidak dapat dinonaktifkan.

HugePages tidak diaktifkan secara default untuk kelas instans DB berikut.

Jajaran kelas instans DB	Kelas instans DB dengan HugePages tidak diaktifkan secara default
db.m5	db.m5.large
db.m4	db.m4.large, db.m4.xlarge, db.m4.2xlarge, db.m4.4xlarge, db.m4.10xlarge
db.t3	db.t3.micro, db.t3.small, db.t3.medium, db.t3.large

Lihat informasi selengkapnya tentang kelas instans DB di Spesifikasi perangkat keras kelas instans DB .

Untuk secara manual mengaktifkan HugePages untuk instans DB baru atau yang sudah ada, atur parameter `use_large_pages` ke `ONLY`. Anda tidak dapat menggunakan HugePages dengan Oracle Automatic Memory Management (AMM). Jika Anda mengatur parameter `use_large_pages` ke `ONLY`, Anda juga harus mengatur `memory_target` dan `memory_max_target` ke `0`. Untuk informasi selengkapnya tentang mengatur parameter DB untuk instans DB Anda, lihat Bekerja dengan grup parameter.

Anda juga dapat mengatur parameter `sga_target`, `sga_max_size`, dan `pga_aggregate_target`. Saat Anda menetapkan parameter memori system global area (GA) dan program global area (PGA), tambahkan nilai-nilainya secara bersamaan. Kurangi total ini dari memori instans Anda yang tersedia (`DBInstanceClassMemory`) untuk menentukan memori kosong di luar alokasi HugePages. Anda harus menyisakan memori kosong setidaknya 2 GiB, atau 10 persen dari total memori instans yang tersedia, mana saja yang lebih kecil.

Setelah mengonfigurasi parameter, Anda harus melakukan reboot instans DB Anda untuk menerapkan perubahan. Untuk informasi selengkapnya, lihat Mem-boot ulang instans DB.

> **Note**
>
> Instans DB Oracle menolak perubahan pada parameter inisialisasi terkait SGA sampai Anda melakukan reboot instans tersebut tanpa failover. Di konsol Amazon RDS, pilih Reboot tetapi jangan pilih Reboot dengan failover. Dalam AWS CLI, panggil perintah `reboot-db-instance` dengan parameter `--no-force-failover`. Instans DB tidak memproses parameter terkait SGA selama failover atau selama operasi pemeliharaan lain yang menyebabkan instans dimulai ulang.

Berikut ini adalah contoh konfigurasi parameter untuk HugePages yang mengaktifkan HugePages secara manual. Anda harus mengatur nilai sesuai kebutuhan Anda.

```
memory_target            = 0
memory_max_target        = 0
pga_aggregate_target     = {DBInstanceClassMemory*1/8}
sga_target               = {DBInstanceClassMemory*3/4}
sga_max_size             = {DBInstanceClassMemory*3/4}
use_large_pages          = ONLY
```

Asumsikan nilai parameter berikut diatur dalam suatu grup parameter.

```
memory_target            = IF({DBInstanceClassHugePagesDefault}, 0,
 {DBInstanceClassMemory*3/4})
memory_max_target        = IF({DBInstanceClassHugePagesDefault}, 0,
 {DBInstanceClassMemory*3/4})
pga_aggregate_target     = IF({DBInstanceClassHugePagesDefault},
 {DBInstanceClassMemory*1/8}, 0)
sga_target               = IF({DBInstanceClassHugePagesDefault},
 {DBInstanceClassMemory*3/4}, 0)
sga_max_size             = IF({DBInstanceClassHugePagesDefault},
 {DBInstanceClassMemory*3/4}, 0)
use_large_pages          = {DBInstanceClassHugePagesDefault}
```

Grup parameter tersebut digunakan oleh kelas instans DB db.r4 dengan memori kurang dari 100 GiB. Dengan pengaturan parameter ini dan `use_large_pages` diatur menjadi `{DBInstanceClassHugePagesDefault}`, HugePages diaktifkan untuk instans db.r4.

Pertimbangkan contoh lain dengan mengikuti nilai parameter yang ditetapkan di grup parameter.

```
memory_target           = IF({DBInstanceClassHugePagesDefault}, 0,
 {DBInstanceClassMemory*3/4})
memory_max_target       = IF({DBInstanceClassHugePagesDefault}, 0,
 {DBInstanceClassMemory*3/4})
pga_aggregate_target    = IF({DBInstanceClassHugePagesDefault},
 {DBInstanceClassMemory*1/8}, 0)
sga_target              = IF({DBInstanceClassHugePagesDefault},
 {DBInstanceClassMemory*3/4}, 0)
sga_max_size            = IF({DBInstanceClassHugePagesDefault},
 {DBInstanceClassMemory*3/4}, 0)
use_large_pages         = FALSE
```

Grup parameter tersebut digunakan oleh kelas instans DB db.r4 dan kelas instans DB db.r5, keduanya dengan memori kurang dari 100 GiB. Dengan pengaturan parameter ini, HugePages dinonaktifkan untuk instans db.r4 dan db.r5.

> **Note**
>
> Jika grup parameter ini digunakan oleh kelas instans DB db.r4 atau kelas instans DB db.r5 dengan memori setidaknya 100 GiB, pengaturan `FALSE` untuk `use_large_pages` ditimpa dan diatur menjadi `ONLY`. Dalam hal ini, pemberitahuan pelanggan tentang penimpaan ini dikirim.

Setelah HugePages aktif di instans DB Anda, Anda dapat melihat informasi HugePages dengan mengaktifkan pemauntauan ditingkatkan. Untuk informasi selengkapnya, lihat Memantau metrik OS dengan Pemantauan yang Disempurnakan.

Mengaktifkan jenis extended data di RDS for Oracle

Amazon RDS for Oracle mendukung jenis extended data Pada jenis extended data, ukuran maksimumnya adalah 32.767 byte untuk jenis data `VARCHAR2`, `NVARCHAR2`, dan `RAW`. Untuk menggunakan jenis extended data, atur parameter `MAX_STRING_SIZE` ke `EXTENDED`. Untuk informasi lebih lanjut, lihat Extended data types dalam dokumentasi Oracle.

Jika Anda tidak ingin menggunakan jenis extended data, biarkan parameter MAX_STRING_SIZE diatur ke STANDARD (default). Dalam hal ini, batasan ukurannya adalah 4.000 byte untuk jenis data VARCHAR2 dan NVARCHAR2, serta 2.000 byte untuk jenis data RAW.

Anda dapat mengaktifkan jenis extended data pada instans DB baru atau yang sudah ada. Untuk instans DB baru, waktu pembuatan instans DB biasanya lebih lama saat Anda mengaktifkan jenis extended data. Instans DB yang ada tidak tersedia selama proses konversi.

Pertimbangan untuk jenis extended data

Pertimbangkan hal berikut saat Anda mengaktifkan jenis extended data untuk instans DB Anda:

- Saat Anda mengaktifkan jenis extended data, Anda tidak dapat mengubah instans DB agar kembali menggunakan ukuran standar untuk jenis extended data. Setelah instans DB dikonversi untuk menggunakan jenis extended data, jika Anda mengatur parameter MAX_STRING_SIZE kembali ke STANDARD, status incompatible-parameters akan muncul.
- Ketika Anda memulihkan instans DB yang menggunakan jenis extended data, Anda harus menentukan grup parameter mengatur parameter MAX_STRING_SIZE ke EXTENDED. Selama pemulihan, jika Anda menentukan grup parameter default atau grup parameter lainnya dengan mengatur MAX_STRING_SIZE ke STANDARD, hasilnya adalah status incompatible-parameters.
- Saat instans DB berstatus incompatible-parameters karena pengaturan MAX_STRING_SIZE, instans DB tetap tidak tersedia hingga Anda mengatur parameter MAX_STRING_SIZE ke EXTENDED dan melakukan reboot instans DB.
- Kami sarankan Anda tidak mengaktifkan jenis extended data untuk instans DB Oracle yang berjalan di kelas instans DB t2.micro.

Mengaktifkan jenis extended data untuk instans DB baru

Untuk mengaktifkan jenis extended data untuk instans DB baru

1. Atur parameter MAX_STRING_SIZE ke EXTENDED dalam grup parameter.

   Untuk mengatur parameter, Anda dapat membuat grup parameter baru atau memodifikasi grup parameter yang sudah ada.

   Untuk informasi selengkapnya, lihat Bekerja dengan grup parameter.
2. Buat instans DB RDS for Oracle baru

Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.

3. Hubungkan grup parameter baru dengan `MAX_STRING_SIZE` diatur ke `EXTENDED` dengan instans DB.

   Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.

Mengaktifkan jenis extended data untuk instans DB yang sudah ada

Saat Anda memodifikasi instans DB untuk mengaktifkan jenis extended data, RDS mengonversi data di basis data untuk menggunakan ukuran lebih besar. Konversi dan waktu henti terjadi ketika Anda melakukan reboot pada basis data setelah perubahan parameter. Instans DB tidak tersedia selama konversi.

Waktu yang dibutuhkan untuk mengonversi data bergantung pada kelas instans DB, ukuran basis data, dan waktu snapshot DB terakhir. Untuk mengurangi waktu henti, sebaiknya ambil snapshot segera sebelum melakukan reboot. Hal ini mempersingkat waktu pencadangan yang terjadi selama alur kerja konversi.

> **Note**
>
> Setelah Anda mengaktifkan jenis extended data, Anda tidak dapat melakukan pemulihan titik waktu ke suatu waktu selama konversi berlangsung. Anda dapat memulihkan ke waktu tertentu segera sebelum konversi atau setelah konversi.

Untuk mengaktifkan jenis extended data untuk instans DB yang sudah ada

1. Ambil snapshot basis data.

   Jika ada objek yang tidak valid di basis data, Amazon RDS mencoba mengompilasinya kembali. Konversi ke jenis extended data dapat gagal jika Amazon RDS tidak dapat mengompilasi ulang objek yang tidak valid. Snapshot memungkinkan Anda memulihkan basis data jika konversi bermasalah. Selalu periksa apakah ada objek tidak valid sebelum konversi dan perbaiki atau lepaskan objek yang tidak valid tersebut. Untuk basis data produksi, kami menyarankan untuk menguji proses konversi pada salinan instans DB Anda terlebih dahulu.

   Untuk informasi selengkapnya, lihat Membuat snapshot DB untuk instans DB Single-AZ.

2. Atur parameter `MAX_STRING_SIZE` ke `EXTENDED` dalam grup parameter.

Untuk mengatur parameter, Anda dapat membuat grup parameter baru atau memodifikasi grup parameter yang sudah ada.

Untuk informasi selengkapnya, lihat Bekerja dengan grup parameter.

3. Modifikasi instans DB untuk menghubungkannya dengan grup parameter yang mengatur `MAX_STRING_SIZE`-nya ke `EXTENDED`.

   Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

4. Reboot instans DB agar perubahan parameter berlaku.

   Untuk informasi selengkapnya, lihat Mem-boot ulang instans DB.

Mengimpor data ke Oracle di Amazon RDS

Cara pengimporan data ke instans DB Amazon RDS for Oracle bergantung pada hal berikut:

- Jumlah data yang Anda miliki
- Jumlah objek basis data di basis data Anda
- Variasi objek basis data di basis data Anda

Misalnya, alat berikut dapat Anda gunakan, tergantung kebutuhan Anda:

- Oracle SQL Developer – Mengimpor basis data sederhana berukuran 20 MB.
- Oracle Data Pump – Mengimpor basis data yang kompleks, atau basis data yang berukuran beberapa ratus megabyte atau beberapa terabyte. Misalnya, Anda dapat mengangkut tablespace dari basis data on-premise ke instans DB RDS for Oracle Anda. Anda dapat menggunakan Amazon S3 atau Amazon EFS untuk mentransfer file data dan metadata. Lihat informasi selengkapnya di Bermigrasi menggunakan tablespace yang dapat dipindahkan Oracle, Integrasi Amazon EFS, dan Integrasi Amazon S3.
- AWS Database Migration Service (AWS DMS) — Migrasikan database tanpa downtime. Untuk informasi selengkapnya AWS DMS, lihat Apa itu AWS Database Migration Service dan posting blog Migrasi database Oracle dengan downtime mendekati nol menggunakan DMS. AWS

> **Important**
>
> Sebelum Anda menggunakan teknik migrasi sebelumnya, sebaiknya buat cadangan basis data Anda. Setelah mengimpor data, Anda dapat membuat cadangan instans DB RDS for Oracle Anda dengan membuat snapshot. Selanjutnya, Anda dapat memulihkan snapshot tersebut. Untuk informasi selengkapnya, lihat Mencadangkan, memulihkan, dan mengekspor data.

Untuk sebagian besar mesin basis data, replikasi berkelanjutan dapat terus berlangsung hingga Anda siap untuk beralih ke basis data target. Anda dapat menggunakan AWS DMS untuk bermigrasi ke RDS untuk Oracle baik dari mesin database yang sama atau mesin yang berbeda. Jika Anda bermigrasi dari mesin database yang berbeda, Anda dapat menggunakan objek skema AWS Schema Conversion Tool untuk memigrasi yang AWS DMS tidak bermigrasi.

Topik

Mengimpor menggunakan Oracle SQL Developer

Oracle SQL Developer adalah alat Java grafis yang didistribusikan tanpa biaya oleh Oracle. SQL Developer menyediakan opsi untuk memigrasi data antara dua basis data Oracle, atau untuk memigrasi data dari basis data lain, seperti MySQL, ke basis data Oracle. Alat ini paling baik untuk memigrasi database kecil.

Anda dapat menginstal alat ini di komputer desktop (Windows, Linux, atau Mac) atau di salah satu server Anda. Setelah menginstal SQL Developer, Anda dapat menggunakannya untuk terhubung ke basis data sumber dan target. Gunakan perintah Database Copy pada menu Tools untuk menyalin data Anda ke RDS Anda untuk instans Oracle DB.

Untuk mengunduh SQL Developer, kunjungi http://www.oracle.com/technetwork/developer-tools/sql-developer.

Sebaiknya baca dokumentasi produk Oracle SQL Developer sebelum Anda mulai memigrasikan data Anda. Oracle juga memiliki dokumentasi terkait cara bermigrasi dari basis data lain, termasuk MySQL dan SQL Server. Untuk informasi selengkapnya, lihat http://www.oracle.com/technetwork/database/migration dalam dokumentasi Oracle.

Bermigrasi menggunakan tablespace yang dapat dipindahkan Oracle

Anda dapat menggunakan fitur tablespace yang dapat dipindahkan Oracle untuk menyalin satu set tablespace dari basis data Oracle on-premise ke instans DB RDS for Oracle. Pada tingkat fisik, Anda mentransfer file data sumber dan file metadata ke instans DB target menggunakan Amazon EFS atau Amazon S3. Fitur ruang meja yang dapat diangkut menggunakan paket. `rdsadmin.rdsadmin_transport_util` Untuk sintaks dan semantik paket ini, lihat. Mengangkut tablespace

Untuk posting blog yang menjelaskan cara mengangkut tablespace, lihat Memigrasi Database Oracle untuk AWS menggunakan ruang meja yang dapat diangkut dan Amazon RDS for Oracle Transportable Tablespaces menggunakan RMAN.

Topik

- Ikhtisar tablespace Oracle yang dapat dipindahkan
- Tahap 1: Siapkan host sumber
- Tahap 2: Siapkan pencadangan tablespace penuh
- Tahap 3: Buat dan transfer cadangan inkremental
- Tahap 4: Pindahkan tablespace
- Tahap 5: Validasi tablespace yang dipindahkan
- Tahap 6: Bersihkan file sisa

Ikhtisar tablespace Oracle yang dapat dipindahkan

Set tablespace yang dapat dipindahkan terdiri dari file data untuk set tablespace yang dipindahkan dan file dump ekspor yang berisi metadata tablespace. Dalam solusi migrasi fisik seperti tablespace yang dapat dipindahkan, Anda mentransfer file fisik: file data, file konfigurasi, dan file dump Data Pump.

Topik

- Kelebihan dan kekurangan tablespace yang dapat dipindahkan
- Batasan tablespace yang dapat dipindahkan
- Prasyarat untuk tablespace yang dapat dipindahkan

Kelebihan dan kekurangan tablespace yang dapat dipindahkan

Penggunaan tablespace yang dapat dipindahkan disarankan saat Anda perlu memigrasikan satu atau beberapa tablespace besar ke RDS dengan waktu henti minimum. Dibanding migrasi logis, tablespace yang dapat dipindahkan memiliki kelebihan sebagai berikut:

- Waktu henti lebih rendah dibandingkan solusi migrasi Oracle lainnya.
- Karena fitur tablespace yang dapat dipindahkan hanya menyalin file fisik, fitur ini mencegah kesalahan integritas data dan kerusakan logis yang dapat terjadi pada migrasi logis.
- Tidak perlu lisensi tambahan.

- Anda bisa memigrasikan set tablespace di berbagai platform dan jenis endian, misalnya, dari platform Oracle Solaris ke Linux. Namun, pemindahan tablespace ke dan dari server Windows tidak didukung.

> **Note**
>
> Linux sepenuhnya teruji dan didukung. Tidak semua variasi UNIX telah diuji.

Jika Anda menggunakan tablespace yang dapat dipindahkan, Anda dapat memindahkan data menggunakan Amazon S3 atau Amazon EFS:

- Saat menggunakan EFS, cadangan Anda tetap berada di sistem file EFS selama impor. Anda dapat menghapus file sesudahnya. Dalam teknik ini, Anda tidak perlu menyediakan penyimpanan EBS untuk instans DB. Karena alasan ini, sebaiknya gunakan Amazon EFS sebagai ganti S3. Untuk informasi selengkapnya, lihat Integrasi Amazon EFS.
- Ketika menggunakan S3, Anda mengunduh cadangan RMAN ke penyimpanan EBS yang terhubung ke instans DB Anda. File tetap berada di penyimpanan EBS Anda selama impor. Setelah impor, Anda dapat mengosongkan ruang ini, yang tetap dialokasikan untuk instans DB Anda.

Kekurangan utama dari tablespace yang dapat dipindahkan adalah perlunya pengetahuan yang cukup mendalam tentang Oracle Database. Untuk informasi selengkapnya, lihat Transporting Tablespaces Between Databases dalam Panduan Administrator Oracle Database.

Batasan tablespace yang dapat dipindahkan

Batasan Oracle Database untuk tablespace yang dapat dipindahkan berlaku ketika Anda menggunakan fitur ini di RDS for Oracle. Untuk informasi selengkapnya, lihat Limitations on Transportable Tablespaces dan General Limitations on Transporting Data dalam Panduan Administrator Oracle Database. Ketahui batasan tambahan untuk tablespace yang dapat dipindahkan di RDS for Oracle berikut:

- Baik basis data sumber maupun target tidak dapat menggunakan Standard Edition 2 (SE2). Hanya mendukung Enterprise Edition.
- Anda tidak dapat menggunakan basis data Oracle Database 11g sebagai sumber. Fitur tablespace lintas platform yang dapat dipindahkan RMAN bergantung pada mekanisme pemindahan RMAN, yang tidak didukung oleh Oracle Database 11g.

- Anda tidak dapat memigrasikan data dari instans DB RDS for Oracle menggunakan tablespace yang dapat dipindahkan. Anda hanya dapat menggunakan tablespace yang dapat dipindahkan untuk memigrasikan data ke instans DB RDS for Oracle.
- Tidak mendukung sistem operasi Windows.
- Anda tidak dapat memindahkan tablespace ke dalam basis data pada tingkat rilis yang lebih rendah. Basis data target harus berada pada tingkat rilis yang sama atau lebih baru dengan basis data sumber. Sebagai contoh, Anda tidak dapat memindahkan tablespace dari Oracle Database 21c ke Oracle Database 19c.
- Anda tidak dapat memindahkan tablespace administratif seperti `SYSTEM` dan `SYSAUX`.
- Anda tidak dapat memindahkan objek non-data seperti paket PL/SQL, kelas Java, tampilan, pemicu, urutan, pengguna, peran, dan tabel sementara. Untuk memindahkan objek non-data, buat secara manual atau gunakan ekspor dan impor metadata Data Pump. Untuk informasi selengkapnya, lihat My Oracle Support Note 1454872.1.
- Anda tidak dapat memindahkan tablespace terenkripsi atau menggunakan kolom terenkripsi.
- Jika Anda mentransfer file menggunakan Amazon S3, ukuran file maksimum yang didukung adalah 5 TiB.
- Jika basis data sumber menggunakan opsi Oracle seperti Spatial, Anda hanya dapat memindahkan tablespace jika basis data target mengonfigurasikan opsi yang sama.
- Anda tidak dapat memindahkan tablespace ke instans DB RDS for Oracle dalam konfigurasi replika Oracle. Sebagai solusinya, Anda dapat menghapus semua replika, memindahkan tablespace, lalu membuat ulang replika.

Prasyarat untuk tablespace yang dapat dipindahkan

Sebelum memulai, selesaikan tugas berikut:

- Tinjau persyaratan untuk tablespace yang dapat dipindahkan yang dijelaskan dalam dokumen berikut di Dukungan Oracle Saya:
  - Reduce Transportable Tablespace Downtime using Cross Platform Incremental Backup (Doc ID 2471245.1)
  - Transportable Tablespace (TTS) Restrictions and Limitations: Details, Reference, and Version Where Applicable (Doc ID 1454872.1)
  - Primary Note for Transportable Tablespaces (TTS) -- Common Questions and Issues (Doc ID 1166564.1)

- Rencanakan konversi endian. Jika Anda menentukan ID platform sumber, RDS for Oracle akan otomatis mengonversi endian. Untuk mempelajari cara menemukan ID platform, lihat Data Guard Support for Heterogeneous Primary and Physical Standbys in Same Data Guard Configuration (Doc ID 413484.1).
- Pastikan fitur tablespace yang dapat dipindahkan telah aktif pada instans DB target Anda. Fitur ini hanya diaktifkan jika tidak ada kesalahan `ORA-20304` saat Anda menjalankan kueri berikut:

  ```
  SELECT * FROM TABLE(rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files);
  ```

  Jika fitur tablespace yang dapat dipindahkan tidak diaktifkan, reboot instans DB Anda. Untuk informasi selengkapnya, lihat Mem-boot ulang instans DB.
- Jika Anda berencana untuk mentransfer file menggunakan Amazon S3, lakukan hal berikut:
  - Pastikan bucket Amazon S3 tersedia untuk transfer file, dan bucket Amazon S3 berada di Wilayah yang AWS sama dengan instans DB Anda. Untuk mengetahui petunjuknya, lihat Membuat bucket di Panduan Memulai Amazon Simple Storage Service.
  - Siapkan bucket Amazon S3 untuk integrasi Amazon RDS dengan mengikuti petunjuk di Mengonfigurasi izin IAM untuk integrasi RDS for Oracle dengan Amazon S3.
- Jika Anda berencana untuk mentransfer file menggunakan Amazon EFS, pastikan EFS telah dikonfigurasi sesuai dengan petunjuk di Integrasi Amazon EFS.
- Sangat disarankan untuk mengaktifkan pencadangan otomatis di instans DB target Anda. Karena langkah impor metadata berpotensi gagal, pastikan instans DB dapat dipulihkan ke kondisi sebelum impor, sehingga tablespace tidak perlu dicadangkan, ditransfer, dan diimpor kembali.

Tahap 1: Siapkan host sumber

Pada langkah ini, salin skrip tablespace pemindahan yang disediakan oleh Dukungan My Oracle dan siapkan file konfigurasi yang diperlukan. Pada langkah berikut, host sumber menjalankan basis data yang berisi tablespace yang akan dipindahkan ke instans target.

Untuk menyiapkan host sumber

1. Masuk ke host sumber sebagai pemilik beranda Oracle.
2. Pastikan variabel lingkungan `ORACLE_HOME` dan `ORACLE_SID` mengarah ke basis data sumber Anda.
3. Masuk ke basis data sebagai administrator, dan pastikan versi zona waktu, set karakter DB, dan set karakter nasional sama dengan yang ada di basis data target.

```
SELECT * FROM V$TIMEZONE_FILE;
SELECT * FROM NLS_DATABASE_PARAMETERS
  WHERE PARAMETER IN ('NLS_CHARACTERSET','NLS_NCHAR_CHARACTERSET');
```

4. Siapkan utilitas tablespace yang dapat dipindahkan seperti yang dijelaskan di Oracle Support note 2471245.1.

   Penyiapan termasuk mengedit file `xtt.properties` di host sumber Anda. Sampel file `xtt.properties` berikut menetapkan pencadangan tiga tablespace di dalam direktori `/dsk1/backups`. Ketiganya adalah tablespace yang akan Anda pindahkan ke instans DB target. Sampel tersebut juga menentukan ID platform sumber untuk mengonversi endian secara otomatis.

   > **Note**
   >
   > Untuk mengetahui ID platform yang valid, lihat ta Guard Support for Heterogeneous Primary and Physical Standbys in Same Data Guard Configuration (Doc ID 413484.1).

   ```
   #linux system
   platformid=13
   #list of tablespaces to transport
   tablespaces=TBS1,TBS2,TBS3
   #location where backup will be generated
   src_scratch_location=/dsk1/backups
   #RMAN command for performing backup
   usermantransport=1
   ```

Tahap 2: Siapkan pencadangan tablespace penuh

Pada tahap ini, Anda mencadangkan tablespace untuk pertama kalinya, mentransfer cadangan ke host target, lalu memulihkannya menggunakan prosedur `rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces`. Setelah tahap ini selesai, cadangan tablespace awal berada di instans DB target Anda dan dapat diperbarui dengan cadangan inkremental.

Topik

Langkah 1: Cadangkan tablespace pada host sumber

Pada langkah ini, gunakan skrip `xttdriver.pl` untuk mencadangkan tablespace Anda secara keseluruhan. Output `xttdriver.pl` disimpan dalam variabel lingkungan `TMPDIR`.

Untuk mencadangkan tablespace

1. Jika tablespace Anda dalam mode hanya baca, masuk ke basis data sumber Anda sebagai pengguna dengan hak akses `ALTER TABLESPACE`, dan ubah modenya menjadi baca/tulis. Jika tidak, lewati ke langkah berikutnya.

   Contoh berikut menempatkan `tbs1`, `tbs2`, dan `tbs3` dalam mode baca/tulis.

   ```
   ALTER TABLESPACE tbs1 READ WRITE;
   ALTER TABLESPACE tbs2 READ WRITE;
   ALTER TABLESPACE tbs3 READ WRITE;
   ```

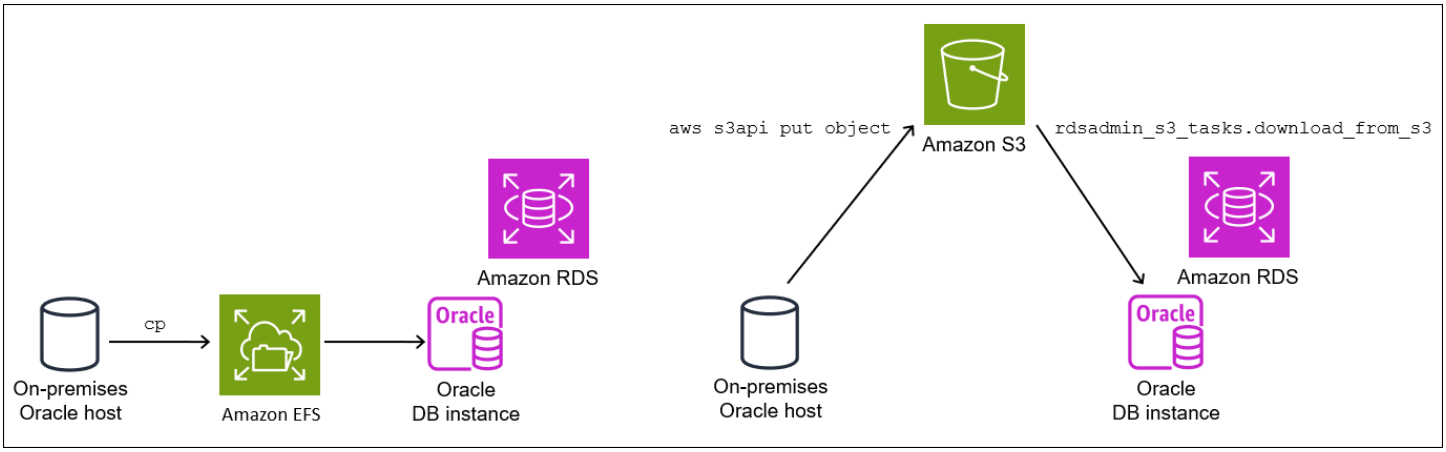
2. Cadangkan tablespace Anda menggunakan skrip `xttdriver.pl`. Secara opsional, Anda dapat menentukan `--debug` untuk menjalankan skrip dalam mode debug.

   ```
   export TMPDIR=location_of_log_files
   cd location_of_xttdriver.pl
   $ORACLE_HOME/perl/bin/perl xttdriver.pl --backup
   ```

Langkah 2: Transfer file cadangan ke instans DB target Anda

Pada langkah ini, salin file cadangan dan konfigurasi dari lokasi awal ke instans DB target. Pilih salah satu opsi berikut:

- Jika host sumber dan target menggunakan sistem file Amazon EFS yang sama, gunakan utilitas sistem operasi seperti `cp` untuk menyalin file cadangan dan file `res.txt` dari lokasi awal ke direktori bersama. Lalu, langsung ke Langkah 3: Impor tablespace pada instans DB target Anda.
- Jika Anda perlu melakukan pencadangan ke bucket Amazon S3, selesaikan langkah berikut.



Langkah 2.2: Unggah cadangan ke bucket Amazon S3 Anda

Unggah cadangan dan `res.txt` file Anda dari direktori awal ke bucket Amazon S3. Untuk informasi selengkapnya, lihat Mengunggah objek di Panduan Pengguna Amazon Simple Storage Service.

Langkah 2.3: Unduh cadangan dari bucket Amazon S3 ke instans DB target

Pada langkah ini, gunakan prosedur `rdsadmin.rdsadmin_s3_tasks.download_from_s3` untuk mengunduh cadangan ke instans DB RDS for Oracle.

Untuk mengunduh cadangan dari bucket Amazon S3

1. Mulai SQL*Plus atau Oracle SQL Developer dan masuk ke instans DB RDS for Oracle.
2. Unduh cadangan dari bucket Amazon S3 ke instans DB target Anda menggunakan prosedur Amazon RDS `rdsadmin.rdsadmin_s3_tasks.download_from_s3` untuk d. Contoh berikut ini mengunduh semua file dari bucket Amazon S3 bernama *`mys3bucket`* ke direktori *`DATA_PUMP_DIR`*.

   ```
   EXEC UTL_FILE.FREMOVE ('DATA_PUMP_DIR', 'res.txt');
   SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
     p_bucket_name    =>  'mys3bucket',
     p_directory_name =>  'DATA_PUMP_DIR')
   AS TASK_ID FROM DUAL;
   ```

   Pernyataan `SELECT` mengembalikan ID tugas dalam jenis data `VARCHAR2`. Untuk informasi selengkapnya, lihat Mengunduh file dari bucket Amazon S3 ke instans DB Oracle.

Langkah 3: Impor tablespace pada instans DB target Anda

Untuk mengembalikan tablespace Anda ke instans DB target Anda, gunakan prosedur`rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces`. Prosedur ini secara otomatis mengonversi file data ke format endian yang benar.

Jika Anda mengimpor dari platform selain Linux, tentukan platform sumber menggunakan parameter `p_platform_id` saat Anda meneleponi`mport_xtts_tablespaces`. Pastikan ID platform yang Anda tentukan cocok dengan yang ditentukan dalam `xtt.properties` file diLangkah 2: Ekspor metadata tablespace di host sumber Anda.

Impor tablespace pada instans DB target Anda

1. Mulai klien Oracle SQL dan masuk ke instans DB RDS for Oracle target Anda sebagai pengguna utama.
2. Jalankan prosedur `rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces`, tentukan tablespace yang akan diimpor dan direktori yang berisi cadangan.

   Contoh berikut ini mengimpor tablespace *`TBS1`*, *`TBS2`*, dan *`TBS3`* dari direktori *`DATA_PUMP_DIR`*. Platform sumbernya adalah Sistem berbasis AIX (64-bit), yang memiliki ID platform. 6 Anda dapat menemukan ID platform dengan melakukan kueri`V$TRANSPORTABLE_PLATFORM`.

   ```
   VAR task_id CLOB

   BEGIN
     :task_id:=rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces(
           'TBS1,TBS2,TBS3',
           'DATA_PUMP_DIR',
           p_platform_id => 6);
   END;
   /

   PRINT task_id
   ```

3. (Opsional) Pantau kemajuan dengan membuat kueri tabel `rdsadmin.rds_xtts_operation_info`. Kolom `xtts_operation_state` menampilkan nilai `EXECUTING`, `COMPLETED`, atau `FAILED`.

   ```
   SELECT * FROM rdsadmin.rds_xtts_operation_info;
   ```

> **Note**
>
> Untuk operasi berdurasi panjang, Anda juga dapat membuat kueri `V$SESSION_LONGOPS`, `V$RMAN_STATUS`, dan `V$RMAN_OUTPUT`.

4. Lihat log impor yang telah selesai menggunakan ID tugas dari langkah sebelumnya.

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||'&task_id'||'.log'));
```

Pastikan impor sudah berhasil sebelum melanjutkan ke langkah berikutnya.

Tahap 3: Buat dan transfer cadangan inkremental

Pada tahap ini, buat dan transfer cadangan inkremental secara berkala saat basis data sumber aktif. Teknik ini mengurangi ukuran pencadangan akhir tablespace Anda. Jika Anda mengambil beberapa cadangan inkremental, Anda harus menyalin file `res.txt` setelah pencadangan inkremental terakhir sebelum menerapkannya pada instans target.

Langkahnya sama seperti pada Tahap 2: Siapkan pencadangan tablespace penuh, kecuali langkah impor yang sifatnya opsional.

Tahap 4: Pindahkan tablespace

Pada tahap ini, cadangkan tablespace hanya baca dan ekspor metadata Data Pump, transfer file ini ke host target, dan impor tablespace dan metadata.

Topik

- Langkah 1: Cadangkan tablespace hanya baca Anda
- Langkah 2: Ekspor metadata tablespace di host sumber Anda
- Langkah 3: (Hanya Amazon S3) Transfer file cadangan dan ekspor ke instans DB target
- Langkah 4: Impor tablespace pada instans DB target Anda
- Langkah 5: Impor metadata tablespace pada instans DB target Anda

Langkah 1: Cadangkan tablespace hanya baca Anda

Langkah ini sama dengan Langkah 1: Cadangkan tablespace pada host sumber, dengan satu perbedaan utama: tablespace Anda diubah ke mode hanya baca sebelum dicadangkan untuk terakhir kalinya.

Contoh berikut menempatkan `tbs1`, `tbs2`, dan `tbs3` dalam mode hanya baca.

```
ALTER TABLESPACE tbs1 READ ONLY;
ALTER TABLESPACE tbs2 READ ONLY;
ALTER TABLESPACE tbs3 READ ONLY;
```

Langkah 2: Ekspor metadata tablespace di host sumber Anda

Ekspor metadata tablespace dengan menjalankan utilitas `expdb` di host sumber Anda. Contoh berikut mengekspor tablespace *TBS1*, *TBS2*, dan *TBS3* ke file dump *xttdump.dmp* di direktori *DATA_PUMP_DIR*.

```
expdp username/pwd \
dumpfile=xttdump.dmp \
directory=DATA_PUMP_DIR \
statistics=NONE \
transport_tablespaces=TBS1,TBS2,TBS3 \
transport_full_check=y \
logfile=tts_export.log
```

Jika *DATA_PUMP_DIR* adalah direktori bersama di Amazon EFS, langsung ke Langkah 4: Impor tablespace pada instans DB target Anda.

Langkah 3: (Hanya Amazon S3) Transfer file cadangan dan ekspor ke instans DB target

Jika Anda menggunakan Amazon S3 untuk melakukan pencadangan tablespace dan file ekspor Data Pump, selesaikan langkah berikut.

Langkah 3.1: Unggah cadangan dan file dump dari host sumber ke bucket Amazon S3 Anda

Unggah cadangan dan file dump dari host sumber ke bucket Amazon S3 Anda. Untuk informasi selengkapnya, lihat Mengunggah objek di Panduan Pengguna Amazon Simple Storage Service.

Langkah 3.2: Unduh cadangan dan file dump dari bucket Amazon S3 ke instans DB target Anda

Pada langkah ini, gunakan prosedur `rdsadmin.rdsadmin_s3_tasks.download_from_s3` untuk mengunduh cadangan dan file dump ke instans DB RDS for Oracle. Ikuti langkah-langkahnya di Langkah 2.3: Unduh cadangan dari bucket Amazon S3 ke instans DB target.

Langkah 4: Impor tablespace pada instans DB target Anda

Gunakan prosedur `rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces` untuk memulihkan tablespace. Untuk sintaks dan semantik prosedur ini, lihat Mengimpor tablespace yang diangkut ke instans DB Anda

> **Important**
>
> Setelah impor akhir tablespace selesai, kemudian impor metadata Oracle Data Pump. Jika impor gagal, instans DB harus dikembalikan ke kondisi sebelum terjadi kegagalan. Oleh karena itu, sebaiknya buat snapshot DB untuk instans DB Anda sesuai petunjuk di Membuat snapshot DB untuk instans DB Single-AZ. Snapshot akan berisi semua tablespace yang diimpor, jadi jika impor gagal, Anda tidak perlu mengulangi proses pencadangan dan impor. Jika instans DB target Anda mengaktifkan pencadangan otomatis, dan Amazon RDS tidak mendeteksi adanya snapshot yang valid sebelum impor metadata dilakukan, RDS akan mencoba membuat snapshot. Bergantung pada aktivitas instans Anda, snapshot ini kemungkinan dapat berhasil atau gagal. Jika tidak ada snapshot yang valid atau snapshot tidak dapat dimulai, maka impor metadata akan gagal.

Impor tablespace pada instans DB target Anda

1. Mulai klien Oracle SQL dan masuk ke instans DB RDS for Oracle target Anda sebagai pengguna utama.
2. Jalankan prosedur `rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces`, tentukan tablespace yang akan diimpor dan direktori yang berisi cadangan.

   Contoh berikut ini mengimpor tablespace *TBS1*, *TBS2*, dan *TBS3* dari direktori *DATA_PUMP_DIR*.

   ```
   BEGIN

    :task_id:=rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces('TBS1,TBS2,TBS3','DATA_
   END;
   ```

```
/
PRINT task_id
```

3. (Opsional) Pantau kemajuan dengan membuat kueri tabel `rdsadmin.rds_xtts_operation_info`. Kolom `xtts_operation_state` menampilkan nilai `EXECUTING`, `COMPLETED`, atau `FAILED`.

```
SELECT * FROM rdsadmin.rds_xtts_operation_info;
```

> **Note**
>
> Untuk operasi berdurasi panjang, Anda juga dapat membuat kueri V`$SESSION_LONGOPS`, V`$RMAN_STATUS`, dan V`$RMAN_OUTPUT`.

4. Lihat log impor yang telah selesai menggunakan ID tugas dari langkah sebelumnya.

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||'&task_id'||'.log'));
```

   Pastikan impor sudah berhasil sebelum melanjutkan ke langkah berikutnya.

5. Ambil snapshot DB manual dengan mengikuti petunjuk di Membuat snapshot DB untuk instans DB Single-AZ.

Langkah 5: Impor metadata tablespace pada instans DB target Anda

Pada langkah ini, Anda mengimpor metadata tablespace yang dapat dipindahkan ke dalam instans DB RDS for Oracle menggunakan prosedur `rdsadmin.rdsadmin_transport_util.import_xtts_metadata`. Untuk sintaks dan semantik prosedur ini, lihat Mengimpor metadata tablespace yang dapat diangkut ke instans DB Anda. Selama operasi, status impor ditunjukkan pada tabel `rdsadmin.rds_xtts_operation_info`.

> **Important**
>
> Sebelum mengimpor metadata, sangat disarankan untuk memastikan bahwa snapshot DB telah berhasil dibuat setelah Anda mengimpor tablespace. Jika langkah impor gagal, pulihkan instans DB, atasi kesalahan impor, lalu coba impor kembali.

Impor metadata Data Pump ke dalam instans DB RDS for Oracle

1. Mulai klien Oracle SQL dan masuk ke instans DB target Anda sebagai pengguna utama.
2. Buat pengguna yang memiliki skema di tablespace yang dipindahkan, jika pengguna tersebut belum ada.

```
CREATE USER tbs_owner IDENTIFIED BY password;
```

3. Impor metadata, tentukan nama file dump dan lokasi direktorinya.

```
BEGIN

 rdsadmin.rdsadmin_transport_util.import_xtts_metadata('xttdump.dmp','DATA_PUMP_DIR');
END;
/
```

4. (Opsional) Kueri tabel riwayat tablespace yang dapat dipindahkan untuk melihat status impor metadata.

```
SELECT * FROM rdsadmin.rds_xtts_operation_info;
```

   Setelah operasi selesai, tablespace Anda berada dalam mode hanya baca.

5. (Opsional) Lihat file log.

   Contoh berikut mencantumkan isi direktori BDUMP, kemudian membuat kueri log impor.

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir(p_directory => 'BDUMP'));

SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file(
  p_directory => 'BDUMP',
  p_filename => 'rds-xtts-
import_xtts_metadata-2023-05-22.01-52-35.560858000.log'));
```

Tahap 5: Validasi tablespace yang dipindahkan

Pada langkah opsional ini, validasi tablespace yang dipindahkan menggunakan prosedur `rdsadmin.rdsadmin_rman_util.validate_tablespace`, lalu ubah tablespace ke mode baca/tulis.

Untuk memvalidasi data yang dipindahkan

1. Mulai SQL*Plus atau SQL Developer dan masuk ke instans DB target Anda sebagai pengguna utama.
2. Validasi tablespace menggunakan prosedur `rdsadmin.rdsadmin_rman_util.validate_tablespace`.

```
SET SERVEROUTPUT ON
BEGIN
    rdsadmin.rdsadmin_rman_util.validate_tablespace(
        p_tablespace_name     => 'TBS1',
        p_validation_type     => 'PHYSICAL+LOGICAL',
        p_rman_to_dbms_output => TRUE);
    rdsadmin.rdsadmin_rman_util.validate_tablespace(
        p_tablespace_name     => 'TBS2',
        p_validation_type     => 'PHYSICAL+LOGICAL',
        p_rman_to_dbms_output => TRUE);
    rdsadmin.rdsadmin_rman_util.validate_tablespace(
        p_tablespace_name     => 'TBS3',
        p_validation_type     => 'PHYSICAL+LOGICAL',
        p_rman_to_dbms_output => TRUE);
END;
/
```

3. Posisikan tablespace Anda ke dalam mode baca/tulis.

```
ALTER TABLESPACE TBS1 READ WRITE;
ALTER TABLESPACE TBS2 READ WRITE;
ALTER TABLESPACE TBS3 READ WRITE;
```

Tahap 6: Bersihkan file sisa

Dalam langkah opsional ini, hapus file yang tidak dibutuhkan. Gunakan prosedur `rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files` untuk membuat daftar file data yang tidak memiliki induk setelah impor tablespace, lalu gunakan prosedur `rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files` untuk menghapusnya. Untuk sintaks dan semantik prosedur ini, lihat Mencantumkan file tanpa induk setelah impor tablespace dan Menghapus file data tanpa induk setelah impor tablespace.

Untuk membersihkan file sisa

1. Hapus cadangan lama di *DATA_PUMP_DIR* sebagai berikut:

   a. Buat daftar file cadangan dengan menjalankan `rdsadmin.rdsadmin_file_util.listdir`.

   ```
   SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir(p_directory =>
    'DATA_PUMP_DIR'));
   ```

   b. Hapus cadangan satu per satu dengan memanggil `UTL_FILE.FREMOVE`.

   ```
   EXEC UTL_FILE.FREMOVE ('DATA_PUMP_DIR', 'backup_filename');
   ```

2. Jika Anda mengimpor tablespace tetapi tidak mengimpor metadatanya, Anda dapat menghapus file data tanpa induk sebagai berikut:

   a. Buat daftar file data tanpa induk yang perlu dihapus. Contoh berikut menjalankan prosedur `rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files`.

   ```
   SQL> SELECT * FROM
    TABLE(rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files);

   FILENAME       FILESIZE
   -------------- ---------
   datafile_7.dbf 104865792
   datafile_8.dbf 104865792
   ```

   b. Hapus file tanpa induk dengan menjalankan prosedur `rdsadmin.rdsadmin_transport_util.cleanup_incomplete_xtts_import`.

   ```
   BEGIN

    rdsadmin.rdsadmin_transport_util.cleanup_incomplete_xtts_import('DATA_PUMP_DIR');
   END;
   /
   ```

   Operasi pembersihan menghasilkan file log yang menggunakan format nama `rds-xtts-delete_xtts_orphaned_files-`*YYYY-MM-DD.HH24-MI-SS.FF*`.log` dalam direktori BDUMP.

c. Baca file log yang dihasilkan pada langkah sebelumnya. Contoh berikut membaca log `rds-xtts-delete_xtts_orphaned_files-2023-06-01.09-33-11.868894000.log`.

```
SELECT *
FROM TABLE(rdsadmin.rds_file_util.read_text_file(
       p_directory => 'BDUMP',
       p_filename  => 'rds-xtts-
delete_xtts_orphaned_files-2023-06-01.09-33-11.868894000.log'));

TEXT
--------------------------------------------------------------------------------
orphan transported datafile datafile_7.dbf deleted.
orphan transported datafile datafile_8.dbf deleted.
```

3. Jika Anda mengimpor tablespace beserta metadatanya, tetapi terjadi kesalahan kompatibilitas atau masalah Oracle Data Pump lainnya, bersihkan file data yang dipindahkan sebagian sebagai berikut:

   a. Buat daftar tablespace yang berisi file data yang dipindahkan sebagian dengan kueri `DBA_TABLESPACES`.

   ```
   SQL> SELECT TABLESPACE_NAME FROM DBA_TABLESPACES WHERE PLUGGED_IN='YES';

   TABLESPACE_NAME
   --------------------------------------------------------------------------------
   TBS_3
   ```

   b. Hapus tabelspace dan file data yang dipindahkan sebagian.

   ```
   DROP TABLESPACE TBS_3 INCLUDING CONTENTS AND DATAFILES;
   ```

Mengimpor menggunakan Oracle Data Pump

Oracle Data Pump adalah utilitas yang memungkinkan Anda untuk mengekspor data Oracle ke file dump dan mengimpornya ke basis data Oracle lain. Ini adalah pengganti jangka panjang untuk utilitas Ekspor/Impor Oracle. Oracle Data Pump adalah cara yang disarankan untuk memindahkan sejumlah besar data dari basis data Oracle ke instans DB Amazon RDS.

Contoh dalam bagian ini menunjukkan satu cara untuk mengimpor data ke dalam basis data Oracle, tetapi Oracle Data Pump mendukung teknik lainnya. Lihat informasi yang lebih lengkap dalam dokumentasi Oracle Database.

Contoh dalam bagian ini menggunakan paket `DBMS_DATAPUMP`. Anda dapat menyelesaikan tugas yang sama menggunakan utilitas baris perintah `impdp` dan `expdp` Oracle Data Pump. Anda dapat menginstal utilitas ini pada host jarak jauh sebagai bagian dari instalasi Oracle Client, termasuk Oracle Instant Client. Untuk informasi selengkapnya, lihat Bagaimana cara menggunakan Oracle Instant Client untuk menjalankan Impor atau Ekspor Data Pump untuk instans DB Amazon RDS for Oracle saya?

Topik

- Gambaran umum Oracle Data Pump
- Mengimpor data dengan Oracle Data Pump dan bucket Amazon S3
- Mengimpor data dengan Oracle Data Pump dan tautan basis data

Gambaran umum Oracle Data Pump

Oracle Data Pump terdiri dari komponen-komponen berikut:

- Klien baris perintah dan `expdp` `impdp`
- Paket PL/SQL `DBMS_DATAPUMP`
- Paket PL/SQL `DBMS_METADATA`

Anda dapat menggunakan Oracle Data Pump untuk skenario berikut:

- Impor data dari basis data Oracle, baik on-premise maupun di sebuah instans Amazon EC2, ke instans DB RDS for Oracle.
- Impor data dari instans DB RDS for Oracle ke basis data Oracle, baik on-premise maupun di sebuah instans Amazon EC2.
- Impor data antara instans DB RDS for Oracle, misalnya, untuk memigrasi data dari EC2-Klasik ke VPC.

Untuk mengunduh utilitas Oracle Data Pump, lihat Unduhan perangkat lunak basis data Oracle di situs web Oracle Technology Network. Untuk pertimbangan kompatibilitas ketika memigrasikan antarversi Oracle Database, lihat dokumentasi Oracle Database.

Alur kerja Oracle Data Pump

Biasanya, Anda menggunakan Oracle Data Pump dalam tahapan berikut:

1. Ekspor data Anda ke dalam file dump pada basis data sumber.
2. Unggah file dump Anda ke instans DB RDS for Oracle tujuan Anda. Anda dapat mentransfer menggunakan bucket Amazon S3 atau menggunakan tautan basis data antara dua basis data tersebut.
3. Impor data dari file dump Anda ke instans DB RDS for Oracle.

Praktik terbaik Oracle Data Pump

Saat Anda menggunakan Oracle Data Pump untuk mengimpor data ke dalam instans RDS for Oracle, kami merekomendasikan praktik terbaik berikut:

- Lakukan impor dalam mode `schema` atau `table` untuk mengimpor skema dan objek tertentu.
- Hanya impor skema yang diperlukan oleh aplikasi Anda.
- Jangan mengimpor dalam mode `full` atau mengimpor skema untuk komponen yang dikelola sistem.

  Karena RDS for Oracle tidak mengizinkan akses untuk pengguna administratif `SYS` atau `SYSDBA`, tindakan ini dapat merusak kamus data Oracle dan memengaruhi stabilitas basis data Anda.
- Saat memuat data dalam jumlah besar, lakukan hal berikut:
  1. Transfer file dump ke instans DB RDS for Oracle target.
  2. Ambil snapshot DB dari instans Anda.
  3. Uji pengimporan untuk memastikan keberhasilannya.

  Jika komponen basis data tidak divalidasi, Anda dapat menghapus instans DB dan membuat ulang instans tersebut dari snapshot DB. Instans DB yang dipulihkan mencakup file dump yang ditetapkan pada instans DB saat Anda mengambil snapshot DB.
- Jangan mengimpor file dump yang dibuat menggunakan parameter ekspor Oracle Data Pump`TRANSPORT_TABLESPACES`, `TRANSPORTABLE`, atau `TRANSPORT_FULL_CHECK`. Instans DB RDS for Oracle tidak mendukung pengimporan file dump ini.
- Jangan mengimpor file dump yang berisi objek Oracle Scheduler di `SYS`, `SYSTEM` ,`RDSADMIN`, `RDSSEC`, serta `RDS_DATAGUARD`, dan yang termasuk dalam kategori berikut:
  - Tugas

- Program
- Jadwal
- Rantai
- Aturan
- Konteks evaluasi
- Set aturan

Instans DB RDS for Oracle tidak mendukung pengimporan file dump ini.

- Untuk mengecualikan objek Oracle Scheduler yang tidak didukung, gunakan arahan tambahan selama ekspor Data Pump. Jika Anda menggunakan`DBMS_DATAPUMP`, Anda dapat menambahkan `METADATA_FILTER` tambahan sebelum `DBMS_METADATA.START_JOB`:

```
DBMS_DATAPUMP.METADATA_FILTER(
  v_hdnl,
  'EXCLUDE_NAME_EXPR',
  q'[IN (SELECT NAME FROM SYS.OBJ$
         WHERE TYPE# IN (66,67,74,79,59,62,46)
         AND OWNER# IN
           (SELECT USER# FROM SYS.USER$
            WHERE NAME IN ('RDSADMIN','SYS','SYSTEM','RDS_DATAGUARD','RDSSEC')
            )
        )
  ]',
  'PROCOBJ'
);
```

Jika Anda menggunakan`expdp`, buat file parameter yang berisi arahan `exclude` sebagaimana ditunjukkan dalam contoh berikut. Kemudian gunakan `PARFILE=`*`parameter_file`* dengan perintah `expdp`.

```
exclude=procobj:"IN
  (SELECT NAME FROM sys.OBJ$
   WHERE TYPE# IN (66,67,74,79,59,62,46)
   AND OWNER# IN
     (SELECT USER# FROM SYS.USER$
      WHERE NAME IN ('RDSADMIN','SYS','SYSTEM','RDS_DATAGUARD','RDSSEC')
     )
  )"
```

Mengimpor data dengan Oracle Data Pump dan bucket Amazon S3

Proses impor berikut menggunakan Oracle Data Pump dan bucket Amazon S3. Langkah-langkahnya adalah sebagai berikut:

1. Ekspor data pada basis data sumber menggunakan paket DBMS_DATAPUMP Oracle.
2. Tempatkan file dump di bucket Amazon S3.
3. Unduh file dump dari bucket Amazon S3 ke direktori `DATA_PUMP_DIR` pada instans DB RDS for Oracle target.
4. Impor data dari file dump yang disalin ke dalam instans DB RDS for Oracle menggunakan paket `DBMS_DATAPUMP`.

Topik

Persyaratan untuk Mengimpor data dengan Oracle Data Pump dan bucket Amazon S3

Proses ini memiliki persyaratan sebagai berikut:

- Pastikan bucket Amazon S3 tersedia untuk transfer file, dan bucket Amazon S3 Wilayah AWS sama dengan instans DB. Untuk mengetahui petunjuknya, lihat Membuat bucket di Panduan Memulai Amazon Simple Storage Service.
- Objek yang Anda unggah ke dalam bucket Amazon S3 harus sebesar 5 TB atau kurang. Untuk mengetahui informasi selengkapnya tentang cara menggunakan objek di Amazon S3, lihat Panduan Pengguna Amazon Simple Storage Service.

> **Note**
>
> Jika file dump lebih besar dari 5 TB, Anda dapat menjalankan ekspor Oracle Data Pump dengan opsi paralel. Operasi ini menyebarkan data ke dalam banyak file dump sehingga setiap file tidak melebihi batas 5 TB.

- Anda harus menyiapkan bucket Amazon S3 untuk integrasi Amazon RDS dengan mengikuti petunjuk di Mengonfigurasi izin IAM untuk integrasi RDS for Oracle dengan Amazon S3.
- Anda harus memastikan bahwa ruang penyimpanan cukup untuk menyimpan file dump pada instans sumber dan instans DB target.

> **Note**
>
> Proses ini mengimpor file dump ke dalam direktori `DATA_PUMP_DIR`, direktori yang telah dikonfigurasi di semua instans DB Oracle. Direktori ini terletak di volume penyimpanan yang sama dengan file data Anda. Saat Anda mengimpor file dump, file data Oracle yang ada menggunakan lebih banyak ruang. Dengan demikian, Anda harus memastikan bahwa instans DB Anda dapat mengakomodasi penggunaan ruang tambahan. File dump yang diimpor tidak secara otomatis dihapus atau dihilangkan dari direktori `DATA_PUMP_DIR`. Untuk menghapus file dump yang diimpor, gunakan UTL_FILE.FREMOVE, yang ada di situs web Oracle.

Langkah 1: Berikan hak istimewa kepada pengguna basis data pada instans DB target RDS for Oracle

Pada langkah ini, Anda membuat skema yang akan menerima impor data dan memberikan hak istimewa yang diperlukan kepada pengguna.

Untuk membuat pengguna dan memberikan hak istimewa yang diperlukan pada instans target RDS for Oracle

1. Gunakan SQL*Plus atau Oracle SQL Developer untuk masuk sebagai pengguna master ke instans DB RDS for Oracle tempat data akan diimpor. Lihat informasi yang lebih lengkap tentang cara menghubungi instans Anda di Menghubungkan ke instans RDS for Oracle DB.
2. Buat ruang tabel yang diperlukan sebelum Anda mengimpor data. Untuk informasi selengkapnya, lihat Membuat dan mengukur tablespace.

3. Buat akun pengguna dan berikan izin serta peran yang diperlukan jika akun pengguna tempat impor data tidak ada. Jika Anda berencana mengimpor data ke dalam beberapa skema pengguna, buat setiap akun pengguna serta berikan hak istimewa dan peran yang diperlukan ke akun tersebut.

   Misalnya, pernyataan SQL berikut membuat pengguna baru serta memberikan izin dan peran yang diperlukan untuk mengimpor data ke dalam skema yang dimiliki oleh pengguna ini. Ganti *schema_1* dengan nama skema Anda pada langkah ini dan pada langkah selanjutnya.

   ```
   CREATE USER schema_1 IDENTIFIED BY my_password;
   GRANT CREATE SESSION, RESOURCE TO schema_1;
   ALTER USER schema_1 QUOTA 100M ON users;
   ```

   > **Note**
   >
   > Tentukan kata sandi selain perintah yang ditampilkan di sini sebagai praktik terbaik keamanan.

   Pernyataan sebelumnya memberi pengguna baru hak istimewa `CREATE SESSION` dan peran `RESOURCE`. Anda mungkin memerlukan hak istimewa dan peran tambahan, tergantung objek basis data yang Anda impor.

Langkah 2: Ekspor data ke file dump menggunakan DBMS_DATAPUMP

Untuk membuat file dump, gunakan paket `DBMS_DATAPUMP`.

Untuk mengekspor data Oracle ke dalam file dump

1. Gunakan SQL Plus atau Oracle SQL Developer untuk terhubung dengan instans DB Oracle sumber dengan pengguna administratif. Jika basis data sumber adalah instans DB RDS for Oracle, hubungkan dengan pengguna utama Amazon RDS.
2. Ekspor data dengan memanggil prosedur `DBMS_DATAPUMP`.

   Skrip berikut mengekspor skema *SCHEMA_1* ke file dump bernama `sample.dmp` dalam direktori `DATA_PUMP_DIR`. Ganti *SCHEMA_1* dengan nama skema yang ingin Anda ekspor.

   ```
   DECLARE
   ```

```
  v_hdnl NUMBER;
BEGIN
  v_hdnl := DBMS_DATAPUMP.OPEN(
    operation => 'EXPORT',
    job_mode  => 'SCHEMA',
    job_name  => null
  );
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl         ,
    filename  => 'sample.dmp'   ,
    directory => 'DATA_PUMP_DIR',
    filetype  => dbms_datapump.ku$_file_type_dump_file
  );
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'sample_exp.log',
    directory => 'DATA_PUMP_DIR' ,
    filetype  => dbms_datapump.ku$_file_type_log_file
  );
  DBMS_DATAPUMP.METADATA_FILTER(v_hdnl,'SCHEMA_EXPR','IN (''SCHEMA_1'')');
  DBMS_DATAPUMP.METADATA_FILTER(
    v_hdnl,
    'EXCLUDE_NAME_EXPR',
    q'[IN (SELECT NAME FROM SYS.OBJ$
           WHERE TYPE# IN (66,67,74,79,59,62,46)
           AND OWNER# IN
             (SELECT USER# FROM SYS.USER$
              WHERE NAME IN ('RDSADMIN','SYS','SYSTEM','RDS_DATAGUARD','RDSSEC')
             )
          )
    ]',
    'PROCOBJ'
  );
  DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/
```

> **Note**
>
> Data Pump memulai tugas secara asinkron. Untuk informasi tentang pemantauan tugas Data Pump, lihat Monitoring job status dalam dokumentasi Oracle.

3. (Opsional) Lihat konten log ekspor dengan memanggil prosedur `rdsadmin.rds_file_util.read_text_file`. Untuk informasi selengkapnya, lihat Membaca file di direktori instans DB.

Langkah 3: Unggah file dump ke bucket Amazon S3 Anda

Gunakan prosedur Amazon RDS `rdsadmin.rdsadmin_s3_tasks.upload_to_s3` untuk menyalin file dump ke bucket Amazon S3. Contoh berikut ini mengunggah semua file dari direktori `DATA_PUMP_DIR` ke bucket Amazon S3 bernama *`myS3bucket`*.

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
  p_bucket_name    =>  'myS3bucket',
  p_directory_name =>  'DATA_PUMP_DIR')
AS TASK_ID FROM DUAL;
```

Pernyataan `SELECT` mengembalikan ID tugas dalam jenis data `VARCHAR2`. Untuk informasi selengkapnya, lihat Mengunggah file dari instans DB RDS for Oracle ke bucket Amazon S3.

Langkah 4: Unduh file dump dari bucket Amazon S3 Anda ke instans DB target Anda

Lakukan langkah ini menggunakan prosedur `rdsadmin.rdsadmin_s3_tasks.download_from_s3` Amazon RDS. Saat Anda mengunduh file ke direktori, prosedur `download_from_s3` melewatkan unduhan jika file bernama identik sudah ada di direktori. Untuk menghapus file dari direktori unduhan, gunakan UTL_FILE.FREMOVE, yang ada di situs web Oracle.

Untuk mengunduh file dump Anda

1. Mulai SQL*Plus atau Oracle SQL Developer dan login sebagai master di instans DB Oracle target Amazon RDS Anda.
2. Unduh file dump menggunakan prosedur `rdsadmin.rdsadmin_s3_tasks.download_from_s3` Amazon RDS.

   Contoh berikut ini mengunduh semua file dari bucket Amazon S3 bernama *`myS3bucket`* ke direktori `DATA_PUMP_DIR`.

   ```
   SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
     p_bucket_name    =>  'myS3bucket',
     p_directory_name =>  'DATA_PUMP_DIR')
   ```

```
AS TASK_ID FROM DUAL;
```

Pernyataan `SELECT` mengembalikan ID tugas dalam jenis data `VARCHAR2`. Untuk informasi selengkapnya, lihat Mengunduh file dari bucket Amazon S3 ke instans DB Oracle.

Langkah 5: Impor file dump Anda ke instans DB target Anda menggunakan DBMS_DATAPUMP

Gunakan `DBMS_DATAPUMP` untuk mengimpor skema ke instans DB RDS for Oracle Anda. Opsi tambahan seperti `METADATA_REMAP` mungkin diperlukan.

Untuk mengimpor data ke instans DB target Anda

1. Mulai SQL*Plus atau SQL Developer dan login sebagai pengguna master ke instans DB RDS for Oracle Anda.
2. Impor data dengan memanggil `DBMS_DATAPUMP` prosedur.

   Contoh berikut mengimpor data *`SCHEMA_1`* dari `sample_copied.dmp` ke instans DB target Anda.

```
DECLARE
  v_hdnl NUMBER;
BEGIN
  v_hdnl := DBMS_DATAPUMP.OPEN(
    operation => 'IMPORT',
    job_mode  => 'SCHEMA',
    job_name  => null);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'sample_copied.dmp',
    directory => 'DATA_PUMP_DIR',
    filetype  => dbms_datapump.ku$_file_type_dump_file);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'sample_imp.log',
    directory => 'DATA_PUMP_DIR',
    filetype  => dbms_datapump.ku$_file_type_log_file);
  DBMS_DATAPUMP.METADATA_FILTER(v_hdnl,'SCHEMA_EXPR','IN (''SCHEMA_1'')');
  DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/
```

> **Note**
>
> Tugas Data Pump dimulai secara asinkron. Untuk informasi tentang pemantauan tugas Data Pump, lihat Monitoring job status dalam dokumentasi Oracle. Anda dapat melihat konten log impor menggunakan prosedur `rdsadmin.rds_file_util.read_text_file`. Untuk informasi selengkapnya, lihat Membaca file di direktori instans DB.

3. Verifikasi impor data dengan mencantumkan tabel skema pada instans DB target Anda.

   Misalnya, kueri berikut mengembalikan jumlah tabel untuk *SCHEMA_1*.

```
SELECT COUNT(*) FROM DBA_TABLES WHERE OWNER='SCHEMA_1';
```

Langkah 6: Bersihkan

Setelah data diimpor, Anda dapat menghapus file yang tidak ingin Anda simpan.

Untuk menghapus file yang tidak diperlukan

1. Mulai SQL*Plus atau SQL Developer dan login sebagai pengguna master ke instans DB RDS for Oracle Anda.
2. Cantumkan file di `DATA_PUMP_DIR` menggunakan perintah berikut.

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir('DATA_PUMP_DIR')) ORDER BY
 MTIME;
```

3. Hapus file di `DATA_PUMP_DIR` yang tidak lagi Anda perlukan menggunakan perintah berikut.

```
EXEC UTL_FILE.FREMOVE('DATA_PUMP_DIR','filename');
```

   Misalnya, perintah berikut menghapus file bernama `sample_copied.dmp`.

```
EXEC UTL_FILE.FREMOVE('DATA_PUMP_DIR','sample_copied.dmp');
```

Mengimpor data dengan Oracle Data Pump dan tautan basis data

Proses impor berikut menggunakan Oracle Data Pump dan paket DBMS_FILE_TRANSFER Oracle. Langkah-langkahnya adalah sebagai berikut:

1. Hubungkan basis data Oracle sumber, yang dapat berupa basis data on-premise, instans Amazon EC2, atau instans DB RDS for Oracle.
2. Ekspor data menggunakan paket DBMS_DATAPUMP.
3. Gunakan `DBMS_FILE_TRANSFER.PUT_FILE` untuk menyalin file dump dari basis data Oracle ke direktori `DATA_PUMP_DIR` pada instans DB RDS for Oracle yang terhubung menggunakan tautan basis data.
4. Impor data dari file dump yang disalin ke dalam instans DB RDS for Oracle menggunakan paket `DBMS_DATAPUMP`.

Proses impor yang menggunakan Oracle Data Pump dan paket `DBMS_FILE_TRANSFER` memiliki langkah-langkah berikut.

Topik

Persyaratan untuk mengimpor data dengan Oracle Data Pump dan tautan basis data

Proses ini memiliki persyaratan sebagai berikut:

- Anda harus memiliki hak istimewa eksekusi pada paket `DBMS_FILE_TRANSFER` dan `DBMS_DATAPUMP`.
- Anda harus memiliki hak istimewa tulis ke direktori `DATA_PUMP_DIR` pada instans DB sumber.

- Anda harus memastikan bahwa ruang penyimpanan cukup untuk menyimpan file dump pada instans sumber dan instans DB target.

> **Note**
>
> Proses ini mengimpor file dump ke dalam direktori `DATA_PUMP_DIR`, direktori yang telah dikonfigurasi di semua instans DB Oracle. Direktori ini terletak di volume penyimpanan yang sama dengan file data Anda. Saat Anda mengimpor file dump, file data Oracle yang ada menggunakan lebih banyak ruang. Dengan demikian, Anda harus memastikan bahwa instans DB Anda dapat mengakomodasi penggunaan ruang tambahan. File dump yang diimpor tidak secara otomatis dihapus atau dihilangkan dari direktori `DATA_PUMP_DIR`. Untuk menghapus file dump yang diimpor, gunakan UTL_FILE.FREMOVE, yang ada di situs web Oracle.

Langkah 1: Berikan hak istimewa kepada pengguna pada instans DB target RDS for Oracle

Untuk memberikan hak istimewa kepada pengguna pada instans DB target RDS for Oracle, lakukan langkah-langkah berikut:

1. Gunakan SQL Plus atau Oracle SQL Developer untuk terhubung dengan instans DB RDS for Oracle yang datanya akan diimpor. Terhubung sebagai pengguna master Amazon RDS. Lihat informasi yang lebih lengkap tentang cara menghubungkan ke instans DB di Menghubungkan ke instans RDS for Oracle DB.
2. Buat ruang tabel yang diperlukan sebelum Anda mengimpor data. Untuk informasi selengkapnya, lihat Membuat dan mengukur tablespace.
3. Jika akun pengguna untuk impor data tidak ada, buat akun pengguna serta berikan izin dan peran yang diperlukan. Jika Anda berencana mengimpor data ke dalam beberapa skema pengguna, buat setiap akun pengguna serta berikan hak istimewa dan peran yang diperlukan ke akun tersebut.

   Misalnya, perintah berikut membuat pengguna baru bernama *schema_1* dan memberikan izin dan peran yang diperlukan untuk mengimpor data ke dalam skema untuk pengguna ini.

```
CREATE USER schema_1 IDENTIFIED BY my-password;
GRANT CREATE SESSION, RESOURCE TO schema_1;
ALTER USER schema_1 QUOTA 100M ON users;
```

> **Note**
>
> Tentukan kata sandi selain perintah yang ditampilkan di sini sebagai praktik terbaik keamanan.

Contoh sebelumnya memberi pengguna baru hak istimewa `CREATE SESSION` dan peran `RESOURCE`. Hak istimewa dan peran tambahan mungkin diperlukan bergantung pada objek basis data yang Anda impor.

> **Note**
>
> Ganti *`schema_1`* dengan nama skema Anda pada langkah ini dan pada langkah selanjutnya.

Langkah 2: Berikan hak istimewa kepada pengguna pada basis data sumber

Gunakan SQL*Plus atau Oracle SQL Developer untuk terhubung ke instans DB RDS for Oracle yang berisi data untuk diimpor. Jika perlu, buat akun pengguna dan berikan izin yang diperlukan.

> **Note**
>
> Jika basis data sumber adalah instans Amazon RDS, Anda dapat melewati langkah ini. Anda menggunakan akun pengguna master Amazon RDS untuk melakukan ekspor.

Perintah berikut membuat pengguna baru dan memberikan izin yang diperlukan.

```
CREATE USER export_user IDENTIFIED BY my-password;
GRANT CREATE SESSION, CREATE TABLE, CREATE DATABASE LINK TO export_user;
ALTER USER export_user QUOTA 100M ON users;
GRANT READ, WRITE ON DIRECTORY data_pump_dir TO export_user;
GRANT SELECT_CATALOG_ROLE TO export_user;
GRANT EXECUTE ON DBMS_DATAPUMP TO export_user;
GRANT EXECUTE ON DBMS_FILE_TRANSFER TO export_user;
```

> **Note**
>
> Tentukan kata sandi selain perintah yang ditampilkan di sini sebagai praktik terbaik keamanan.

Langkah 3: Buat file dump menggunakan DBMS_DATAPUMP

Untuk membuat file dump, lakukan hal berikut:

1. Gunakan SQL*Plus atau Oracle SQL Developer untuk terhubung ke instans Oracle sumber dengan pengguna administratif atau pengguna yang Anda buat pada langkah 2. Jika basis data sumber adalah instans DB Amazon RDS for Oracle, hubungkan dengan pengguna utama Amazon RDS.
2. Buat file dump menggunakan utilitas Oracle Data Pump.

   Skrip berikut membuat file dump bernama sampel.dmp dalam direktori `DATA_PUMP_DIR`.

```
DECLARE
  v_hdnl NUMBER;
BEGIN
  v_hdnl := DBMS_DATAPUMP.OPEN(
    operation => 'EXPORT' ,
    job_mode  => 'SCHEMA' ,
    job_name  => null
  );
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'sample.dmp'    ,
    directory => 'DATA_PUMP_DIR' ,
    filetype  => dbms_datapump.ku$_file_type_dump_file
  );
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl           ,
    filename  => 'sample_exp.log' ,
    directory => 'DATA_PUMP_DIR'  ,
    filetype  => dbms_datapump.ku$_file_type_log_file
  );
  DBMS_DATAPUMP.METADATA_FILTER(
    v_hdnl              ,
    'SCHEMA_EXPR'       ,
    'IN (''SCHEMA_1'')'
```

```
  );
  DBMS_DATAPUMP.METADATA_FILTER(
    v_hdnl,
    'EXCLUDE_NAME_EXPR',
    q'[IN (SELECT NAME FROM sys.OBJ$
           WHERE TYPE# IN (66,67,74,79,59,62,46)
           AND OWNER# IN
             (SELECT USER# FROM SYS.USER$
              WHERE NAME IN ('RDSADMIN','SYS','SYSTEM','RDS_DATAGUARD','RDSSEC')
             )
          )
    ]',
    'PROCOBJ'
  );
  DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/
```

> **Note**
>
> Tugas Data Pump dimulai secara asinkron. Untuk informasi tentang pemantauan tugas Data Pump, lihat Monitoring job status dalam dokumentasi Oracle. Anda dapat melihat konten log ekspor menggunakan prosedur `rdsadmin.rds_file_util.read_text_file`. Untuk informasi selengkapnya, lihat Membaca file di direktori instans DB.

Langkah 4: Buat tautan basis data ke instans DB target

Buat tautan basis data antara instans sumber Anda dan instans DB target Anda. Instans Oracle lokal Anda harus memiliki konektivitas jaringan ke instans DB untuk membuat tautan basis data dan untuk mentransfer file dump ekspor Anda.

Lakukan langkah ini dengan terhubung ke akun pengguna yang sama dengan langkah sebelumnya.

Jika Anda membuat tautan basis data antara dua instans DB di dalam VPC yang sama atau VPC terhubung, kedua instans DB tersebut harus memiliki rute yang valid di antara keduanya. Grup keamanan dari setiap instans DB harus mengizinkan ingress dan egress dari instans DB lainnya. Aturan masuk dan keluar grup keamanan dapat merujuk ke grup keamanan dari VPC yang sama atau VPC terhubung. Untuk informasi selengkapnya, lihat Menyesuaikan tautan basis data untuk penggunaan instans DB di VPC.

Perintah berikut membuat tautan basis data bernama `to_rds` yang terhubung dengan pengguna master Amazon RDS pada instans DB target.

```
CREATE DATABASE LINK to_rds
  CONNECT TO <master_user_account> IDENTIFIED BY <password>
  USING '(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=<dns or ip address of remote db>)
         (PORT=<listener port>))(CONNECT_DATA=(SID=<remote SID>)))';
```

Langkah 5: Salin file dump yang diekspor ke instans DB target menggunakan DBMS_FILE_TRANSFER

Gunakan `DBMS_FILE_TRANSFER` untuk menyalin file dump dari instans basis data sumber ke instans DB target. Skrip berikut menyalin file dump bernama sample.dmp dari instans sumber ke tautan basis data target bernama to-rds (yang dibuat pada langkah sebelumnya).

```
BEGIN
  DBMS_FILE_TRANSFER.PUT_FILE(
    source_directory_object       => 'DATA_PUMP_DIR',
    source_file_name              => 'sample.dmp',
    destination_directory_object  => 'DATA_PUMP_DIR',
    destination_file_name         => 'sample_copied.dmp',
    destination_database          => 'to_rds' );
END;
/
```

Langkah 6: Impor file data ke instans DB target menggunakan DBMS_DATAPUMP

Gunakan Oracle Data Pump untuk mengimpor skema dalam instans DB. Opsi tambahan seperti METADATA_REMAP mungkin diperlukan.

Hubungkan ke instans DB dengan akun pengguna master Amazon RDS untuk melakukan impor.

```
DECLARE
  v_hdnl NUMBER;
BEGIN
  v_hdnl := DBMS_DATAPUMP.OPEN(
    operation => 'IMPORT',
    job_mode  => 'SCHEMA',
    job_name  => null);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'sample_copied.dmp',
```

```
    directory => 'DATA_PUMP_DIR',
    filetype  => dbms_datapump.ku$_file_type_dump_file );
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'sample_imp.log',
    directory => 'DATA_PUMP_DIR',
    filetype  => dbms_datapump.ku$_file_type_log_file);
  DBMS_DATAPUMP.METADATA_FILTER(v_hdnl,'SCHEMA_EXPR','IN (''SCHEMA_1'')');
  DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/
```

> **Note**
>
> Tugas Data Pump dimulai secara asinkron. Untuk informasi tentang pemantauan tugas Data Pump, lihat Monitoring job status dalam dokumentasi Oracle. Anda dapat melihat konten log impor menggunakan prosedur `rdsadmin.rds_file_util.read_text_file`. Untuk informasi selengkapnya, lihat Membaca file di direktori instans DB.

Anda dapat memverifikasi impor data dengan melihat tabel pengguna pada instans DB. Misalnya, kueri berikut mengembalikan jumlah tabel untuk *schema_1*.

```
SELECT COUNT(*) FROM DBA_TABLES WHERE OWNER='SCHEMA_1';
```

Langkah 7: Bersihkan

Setelah data diimpor, Anda dapat menghapus file yang tidak ingin Anda simpan. Anda dapat mencantumkan file di DATA_PUMP_DIR menggunakan perintah berikut.

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir('DATA_PUMP_DIR')) ORDER BY MTIME;
```

Untuk menghapus file di DATA_PUMP_DIR yang tidak lagi Anda perlukan, gunakan perintah berikut.

```
EXEC UTL_FILE.FREMOVE('DATA_PUMP_DIR','<file name>');
```

Misalnya, perintah berikut menghapus file bernama "sample_copied.dmp".

```
EXEC UTL_FILE.FREMOVE('DATA_PUMP_DIR','sample_copied.dmp');
```

Impor menggunakan Ekspor/Impor Oracle

Anda dapat mempertimbangkan utilitas Ekspor/Impor Oracle untuk migrasi dalam kondisi berikut:

- Ukuran data Anda kecil.
- Tidak memerlukan jenis data seperti binary float dan double.

Proses impor menciptakan objek skema yang diperlukan. Dengan demikian, Anda tidak perlu menjalankan skrip untuk membuat objek sebelumnya.

Cara termudah untuk menginstal utilitas ekspor dan impor Oracle adalah dengan menginstal Oracle Instant Client. Untuk mengunduh perangkat lunak tersebut, buka https://www.oracle.com/database/technologies/instant-client.html. Dokumentasinya bisa dilihat di Instant Client for SQL*Loader, Export, and Import dalam manual Oracle Database Utilities.

Cara mengekspor tabel lalu mengimpornya

1. Ekspor tabel dari basis data sumber menggunakan perintah `exp`.

   Perintah berikut mengekspor tabel bernama `tab1`, `tab2`, dan `tab3`. File dump-nya adalah `exp_file.dmp`.

   ```
   exp cust_dba@ORCL FILE=exp_file.dmp TABLES=(tab1,tab2,tab3) LOG=exp_file.log
   ```

   Ekspor menciptakan file dump biner yang berisi skema dan data untuk tabel yang ditentukan.
2. Impor skema dan data tersebut ke dalam basis data target menggunakan perintah `imp`.

   Perintah berikut mengimpor tabel `tab1`, `tab2`, dan `tab3` dari file dump `exp_file.dmp`.

   ```
   imp cust_dba@targetdb FROMUSER=cust_schema TOUSER=cust_schema \
   TABLES=(tab1,tab2,tab3) FILE=exp_file.dmp LOG=imp_file.log
   ```

Ekspor dan Impor memiliki variasi lain yang mungkin lebih sesuai dengan kebutuhan Anda. Lihat dokumentasi Oracle Database untuk mengetahui detail selengkapnya.

Mengimpor menggunakan Oracle SQL*Loader

Anda dapat mempertimbangkan Oracle SQL*Loader untuk basis data besar yang berisi objek dalam jumlah terbatas. Karena proses pengeksporan dari basis data sumber dan pemuatan ke basis data target spesifik untuk skema, contoh berikut membuat objek skema sampel, mengekspor dari sumber, lalu memuat data ke dalam basis data target.

Cara termudah untuk menginstal Oracle SQL*Loader adalah dengan menginstal Oracle Instant Client. Untuk mengunduh perangkat lunak tersebut, buka https://www.oracle.com/database/technologies/instant-client.html. Dokumentasinya bisa dilihat di Instant Client for SQL*Loader, Export, and Import dalam manual Oracle Database Utilities.

Cara mengimpor data menggunakan Oracle SQL*Loader

1. Buat tabel sumber sampel menggunakan pernyataan SQL berikut.

```
CREATE TABLE customer_0 TABLESPACE users
   AS (SELECT ROWNUM id, o.*
       FROM   ALL_OBJECTS o, ALL_OBJECTS x
       WHERE  ROWNUM <= 1000000);
```

2. Pada instans DB RDS for Oracle target, buat tabel tujuan untuk memuat data. Klausa `WHERE 1=2` memastikan bahwa Anda menyalin struktur `ALL_OBJECTS` tanpa menyalin baris apa pun.

```
CREATE TABLE customer_1 TABLESPACE users
  AS (SELECT 0 AS ID, OWNER, OBJECT_NAME, CREATED
      FROM   ALL_OBJECTS
      WHERE  1=2);
```

3. Ekspor data dari basis data sumber ke file teks. Contoh berikut menggunakan SQL*Plus. Untuk data Anda, Anda mungkin perlu membuat skrip yang melakukan ekspor untuk semua objek dalam basis data.

```
ALTER SESSION SET NLS_DATE_FORMAT = 'YYYY/MM/DD HH24:MI:SS'

SET LINESIZE 800 HEADING OFF FEEDBACK OFF ARRAY 5000 PAGESIZE 0
SPOOL customer_0.out
SET MARKUP HTML PREFORMAT ON
SET COLSEP ','

SELECT id, owner, object_name, created
```

```
FROM   customer_0;

SPOOL OFF
```

4. Buat file kontrol untuk mendeskripsikan data tersebut. Anda mungkin perlu menulis skrip untuk melakukan langkah ini.

```
cat << EOF > sqlldr_1.ctl
load data
infile customer_0.out
into table customer_1
APPEND
fields terminated by "," optionally enclosed by '"'
(
  id           POSITION(01:10)    INTEGER EXTERNAL,
  owner        POSITION(12:41)    CHAR,
  object_name  POSITION(43:72)    CHAR,
  created      POSITION(74:92)    date "YYYY/MM/DD HH24:MI:SS"
)
```

   Jika perlu, salin file yang dihasilkan oleh kode sebelumnya ke area penahapan, seperti instans Amazon EC2.

5. Impor data menggunakan SQL*Loader dengan nama pengguna dan kata sandi yang sesuai untuk basis data target.

```
sqlldr cust_dba@targetdb CONTROL=sqlldr_1.ctl BINDSIZE=10485760 READSIZE=10485760
 ROWS=1000
```

Bermigrasi dengan tampilan terwujud Oracle

Untuk memigrasikan set data berukuran besar secara efisien, Anda dapat menggunakan replikasi tampilan terwujud Oracle. Dengan replikasi, tabel target tetap dapat sinkron dengan tabel sumber. Dengan demikian, Anda dapat beralih ke Amazon RDS nantinya, jika perlu.

Sebelum Anda dapat bermigrasi menggunakan tampilan terwujud, pastikan Anda memenuhi persyaratan berikut:

- Konfigurasikan akses dari basis data target ke basis data sumber. Dalam contoh berikut, aturan akses diaktifkan pada basis data sumber agar basis data target RDS for Oracle dapat terhubung ke sumber melalui SQL*Net.
- Buat tautan basis data dari instans DB RDS for Oracle ke basis data sumber.

Cara memigrasikan data menggunakan tampilan terwujud

1. Buat akun pengguna pada instans sumber maupun instans target RDS for Oracle yang dapat diautentikasi dengan kata sandi yang sama. Contoh berikut membuat pengguna bernama `dblink_user`.

```
CREATE USER dblink_user IDENTIFIED BY my-password
  DEFAULT TABLESPACE users
  TEMPORARY TABLESPACE temp;

GRANT CREATE SESSION TO dblink_user;

GRANT SELECT ANY TABLE TO dblink_user;

GRANT SELECT ANY DICTIONARY TO dblink_user;
```

> **Note**
>
> Tentukan kata sandi selain perintah yang ditampilkan di sini sebagai praktik terbaik keamanan.

2. Buat tautan basis data dari instans target RDS for Oracle ke instans sumber menggunakan pengguna yang baru Anda buat.

```
CREATE DATABASE LINK remote_site
  CONNECT TO dblink_user IDENTIFIED BY my-password
  USING '(description=(address=(protocol=tcp) (host=my-host)
    (port=my-listener-port)) (connect_data=(sid=my-source-db-sid)))';
```

> **Note**
>
> Tentukan kata sandi selain perintah yang ditampilkan di sini sebagai praktik terbaik keamanan.

3. Uji tautan tersebut:

```
SELECT * FROM V$INSTANCE@remote_site;
```

4. Buat tabel sampel dengan kunci primer dan log tampilan terwujud pada instans sumber.

```
CREATE TABLE customer_0 TABLESPACE users
  AS (SELECT ROWNUM id, o.*
      FROM   ALL_OBJECTS o, ALL_OBJECTS x
      WHERE  ROWNUM <= 1000000);

ALTER TABLE customer_0 ADD CONSTRAINT pk_customer_0 PRIMARY KEY (id) USING INDEX;

CREATE MATERIALIZED VIEW LOG ON customer_0;
```

5. Pada instans DB RDS for Oracle target, buat tampilan terwujud.

```
CREATE MATERIALIZED VIEW customer_0
  BUILD IMMEDIATE REFRESH FAST
  AS (SELECT *
      FROM   cust_dba.customer_0@remote_site);
```

6. Pada instans DB RDS for Oracle target, refresh tampilan terwujud.

```
EXEC DBMS_MV.REFRESH('CUSTOMER_0', 'f');
```

7. Batalkan tampilan terwujud dan sertakan klausa `PRESERVE TABLE` untuk mempertahankan tabel kontainer tampilan terwujud beserta isinya.

```
DROP MATERIALIZED VIEW customer_0 PRESERVE TABLE;
```

   Tabel yang dipertahankan memiliki nama yang sama dengan tampilan terwujud yang dibatalkan.

Menggunakan replika baca untuk Amazon RDS for Oracle

Untuk mengonfigurasi replikasi di antara instans DB Oracle, Anda dapat membuat basis data replika. Untuk ikhtisar replika baca Amazon RDS, lihat Gambaran umum replika baca Amazon RDS. Untuk ringkasan perbedaan antara replika Oracle dan mesin DB lainnya, lihat Perbedaan di antara beberapa replika baca untuk mesin DB.

Topik

Ikhtisar replika RDS for Oracle

Basis data replika Oracle adalah salinan fisik dari basis data primer Anda. Replika Oracle dalam mode hanya baca disebut replika baca. Replika Oracle dalam mode terpasang disebut replika terpasang. Oracle Database tidak mengizinkan penulisan dalam replika, tetapi Anda dapat meningkatkan replika agar dapat ditulis. Replika baca yang ditingkatkan memiliki data yang direplikasi ke waktu pembuatan permintaan peningkatan.

Video berikut memberikan ikhtisar bermanfaat tentang pemulihan bencana RDS for Oracle.

Untuk informasi selengkapnya, lihat posting blog Pemulihan bencana terkelola dengan pencadangan otomatis lintas Wilayah Amazon RDS for Oracle - Bagian 1 dan Pemulihan bencana terkelola dengan pencadangan otomatis lintas Wilayah Amazon RDS for Oracle - Bagian 2.

Topik

- Gangguan selama replikasi

Replika hanya baca dan terpasang

Saat membuat atau memodifikasi replika Oracle, Anda dapat menggunakan salah satu mode berikut:

Hanya baca

Ini menjadi opsi default. Active Data Guard mentransmisikan dan menerapkan perubahan dari basis data sumber ke semua basis data replika baca.

Anda dapat membuat hingga lima replika baca dari satu instans DB sumber. Untuk informasi umum tentang replika baca yang berlaku untuk semua mesin DB, lihat Menggunakan replika baca instans DB. Untuk informasi tentang Oracle Data Guard, lihat Oracle Data Guard concepts and administration dalam dokumentasi Oracle.

Terpasang

Dalam kasus ini, replikasi menggunakan Oracle Data Guard, tetapi basis data replika tidak menerima koneksi pengguna. Penggunaan utama replika terpasang adalah untuk pemulihan bencana lintas Wilayah.

Replika terpasang tidak dapat memberikan beban kerja hanya baca. Replika terpasang menghapus file log pengulangan yang diarsipkan setelah menerapkannya, terlepas dari kebijakan penyimpanan log yang diarsipkan.

Anda dapat membuat kombinasi replika DB terpasang dan hanya baca untuk instans DB sumber yang sama. Anda dapat mengubah replika hanya baca ke mode terpasang, atau mengubah replika terpasang ke mode hanya baca. Dalam kedua kasus tersebut, basis data Oracle mempertahankan pengaturan retensi log yang diarsipkan.

Replika baca CDB

RDS for Oracle mendukung replika baca Data Guard untuk Oracle Database 19c dan 21c CDB hanya dalam konfigurasi penghuni tunggal. Anda dapat membuat, mengelola, dan meningkatkan replika baca di CDB seperti halnya di non-CDB. Replika terpasang juga didukung. Berikut manfaat yang Anda dapatkan:

- Pemulihan bencana terkelola, ketersediaan tinggi, dan akses hanya baca ke replika Anda

- Kemampuan untuk membuat replika baca di Wilayah AWS yang berbeda.
- Integrasi dengan API replika baca RDS yang sudah ada: CreateDBInstanceReadReplica, PromoteReadReplica, dan SwitchoverReadReplica

Untuk menggunakan fitur ini, Anda memerlukan lisensi Active Data Guard dan lisensi Oracle Database Enterprise Edition untuk instans DB primer dan replika. Penggunaan arsitektur CDB tidak menimbulkan biaya tambahan. Anda hanya perlu membayar instans DB.

Untuk informasi selengkapnya tentang konfigurasi penghuni tunggal dan multi-penghuni arsitektur CDB, lihat Ikhtisar CDB RDS for Oracle.

Retensi log pengulangan yang diarsipkan

Jika instans DB primer tidak memiliki replika baca lintas Wilayah, Amazon RDS for Oracle akan menyimpan log pengulangan yang diarsipkan selama minimal dua jam pada instans DB sumber. Hal ini berlaku tanpa mempertimbangkan pengaturan `archivelog retention hours` di `rdsadmin.rdsadmin_util.set_configuration`.

RDS membersihkan log dari instans DB sumber setelah dua jam atau setelah pengaturan jam retensi arsip log berakhir, mana pun yang lebih lama. RDS membersihkan log dari replika baca setelah pengaturan jam retensi log arsip berakhir hanya jika pengaturan tersebut berhasil diterapkan ke basis data.

Dalam beberapa kasus, instans DB primer utama mungkin memiliki satu replika baca lintas Wilayah atau lebih. Jika demikian, Amazon RDS for Oracle menyimpan log transaksi pada instans DB sumber sampai log transaksi tersebut dikirim dan diterapkan ke semua replika baca lintas Wilayah. Untuk informasi tentang `rdsadmin.rdsadmin_util.set_configuration`, lihat Menyimpan log pengulangan yang diarsipkan.

Gangguan selama replikasi

Saat Anda membuat replika Oracle, tidak terjadi gangguan pada instans DB sumber. Amazon RDS mengambil snapshot instans DB sumber. Snapshot ini menjadi replika. Amazon RDS menetapkan parameter dan izin yang diperlukan untuk DB sumber dan replika tanpa gangguan layanan. Demikian pula, jika Anda menghapus replika, tidak akan tejadi gangguan.

Persyaratan dan pertimbangan untuk replika RDS for Oracle

Sebelum membuat replika Oracle, pahami persyaratan dan pertimbangan berikut terlebih dahulu.

Topik

Persyaratan versi dan lisensi untuk replika RDS for Oracle

Sebelum Anda membuat replika RDS for Oracle, pertimbangkan hal berikut:

- Jika replika berada dalam mode hanya baca, pastikan bahwa Anda memiliki lisensi Active Data Guard. Jika replika berada dalam mode terpasang, Anda tidak memerlukan lisensi Active Data Guard. Hanya mesin DB Oracle yang mendukung replika terpasang.
- Replika Oracle hanya didukung untuk mesin Oracle Enterprise Edition (EE).
- Replika Oracle dari non-CDB hanya didukung untuk instans DB yang dibuat menggunakan versi Oracle Database 12c Rilis 1 (12.1.0.2.v10) dan rilis 12c yang lebih tinggi, serta untuk instans non-CDB dari Oracle Database 19c.
- Replika Oracle CDB hanya didukung untuk instans CDB yang dibuat menggunakan versi Oracle Database 19c dan yang lebih tinggi.
- Replika Oracle tersedia untuk instans DB yang berjalan hanya di kelas instans DB dengan dua vCPU atau lebih. Instans DB sumber tidak dapat menggunakan kelas instans db.t3.micro atau db.t3.small.
- Versi mesin DB Oracle dari instans DB sumber dan semua replikanya harus sama. Amazon RDS langsung memutakhirkan replika setelah instans DB sumber ditingkatkan, terlepas dari periode pemeliharaan replika. Untuk peningkatan versi utama replika lintas Wilayah, Amazon RDS secara otomatis melakukan hal berikut:
  - Membuat grup opsi untuk versi target.
  - Menyalin semua opsi dan pengaturan opsi dari grup opsi asli ke grup opsi baru.
  - Mengaitkan replika lintas Wilayah yang telah ditingkatkan dengan grup opsi baru.

  Untuk informasi selengkapnya tentang meningkatkan versi mesin DB, lihat Meng-upgrade mesin DB Oracle.

Pertimbangan grup opsi untuk replika RDS for Oracle

Sebelum Anda membuat replika RDS for Oracle, pertimbangkan hal berikut:

- Jika replika Oracle Anda berada di Wilayah AWS yang sama dengan lokasi instans DB sumbernya, pastikan replika tersebut berasal dari grup opsi yang sama dengan lokasi instans DB sumber. Perubahan pada grup opsi sumber atau keanggotaan grup opsi sumber menyebar ke replika. Perubahan ini diterapkan ke replika segera setelah diterapkan ke instans DB sumber, terlepas dari masa pemeliharaan replika.

  Untuk informasi selengkapnya tentang grup opsi, lihat Menggunakan grup opsi.
- Saat Anda membuat replika lintas Wilayah RDS for Oracle, Amazon RDS membuat grup opsi khusus untuk replika tersebut.

  Anda tidak dapat menghapus replika lintas Wilayah RDS for Oracle dari grup opsi khususnya. Tidak ada instans DB lain yang dapat menggunakan grup opsi khusus untuk replika lintas Wilayah RDS for Oracle.

  Anda hanya dapat menambahkan atau menghapus opsi nonreplikasi berikut dari grup opsi khusus:
  - `NATIVE_NETWORK_ENCRYPTION`
  - `OEM`
  - `OEM_AGENT`
  - `SSL`

  Untuk menambahkan opsi lain ke replika lintas Wilayah RDS for Oracle, tambahkan opsi tersebut ke grup opsi instans DB sumber. Opsi ini juga diinstal pada semua replika instans DB sumber. Untuk opsi berlisensi, pastikan terdapat lisensi yang cukup untuk replika.

  Saat Anda mempromosikan replika lintas Wilayah RDS for Oracle, replika yang dipromosikan memiliki perilaku yang sama seperti instans DB Oracle lainnya, termasuk manajemen opsinya. Anda dapat mempromosikan replika secara eksplisit atau implisit dengan menghapus instans DB sumbernya.

  Untuk informasi selengkapnya tentang grup opsi, lihat Menggunakan grup opsi.

Pertimbangan pencadangan dan pemulhan untuk replika RDS for Oracle

Sebelum Anda membuat replika RDS for Oracle, pertimbangkan hal berikut:

- Untuk membuat snapshot replika RDS for Oracle atau mengaktifkan pencadangan otomatis, atur periode retensi cadangan secara manual. Pencadangan otomatis tidak diaktifkan secara default.
- Saat memulihkan cadangan replika, Anda memulihkan ke waktu basis data, bukan ke waktu saat pencadangan dilakukan. Waktu basis data adalah waktu transaksi terakhir yang diterapkan pada data dalam cadangan. Perbedaannya cukup signifikan karena replika dapat tertinggal beberapa menit atau jam dari cadangan utama.

  Untuk menemukan perbedaannya, gunakan perintah `describe-db-snapshots`. Bandingkan `snapshotDatabaseTime`, yang merupakan waktu basis data cadangan replika, dan bidang `OriginalSnapshotCreateTime`, yang merupakan transaksi terakhir yang diterapkan pada basis data utama.

Persyaratan dan batasan Oracle Data Guard untuk replika RDS for Oracle

Sebelum Anda membuat replika RDS for Oracle, perhatikan persyaratan dan batasan berikut:

- Jika instans DB primer Anda menggunakan konfigurasi penghuni tunggal dari arsitektur multi-penghuni, pertimbangkan hal berikut:
  - Anda harus menggunakan Oracle Database 19c atau yang lebih tinggi dengan Enterprise Edition.
  - Instans CDB primer Anda harus dalam siklus hidup `ACTIVE`.
  - Anda tidak dapat mengonversi instans primer non-CDB menjadi instans CDB dan mengonversi replikanya dalam operasi yang sama. Jadi, hapus replika non-CDB, konversi instans DB primer menjadi CDB, lalu buat replika baru
- Pastikan bahwa pemicu logon pada instans DB primer memungkinkan akses ke pengguna `RDS_DATAGUARD` dan pengguna apa pun yang nilai `AUTHENTICATED_IDENTITY`-nya adalah `RDS_DATAGUARD` atau `rdsdb`. Selain itu, pemicu tidak boleh menetapkan skema saat ini untuk pengguna `RDS_DATAGUARD`.
- Agar koneksi dari proses broker Data Guard tidak terblokir, jangan mengaktifkan sesi terbatas. Untuk informasi selengkapnya tentang sesi terbatas, lihat Mengaktifkan dan menonaktifkan sesi terbatas.

Pertimbangan lainnya untuk replika RDS for Oracle

Sebelum Anda membuat replika RDS for Oracle, pertimbangkan hal berikut:

- Jika instans DB Anda adalah sumber untuk satu replika lintas Wilayah atau lebih, DB sumber mempertahankan log pengulangannya yang diarsipkan sampai diterapkan pada semua replika lintas Wilayah. Log pengulangan yang diarsipkan mungkin mengakibatkan peningkatan konsumsi penyimpanan.
- Agar tidak mengganggu otomatisasi RDS, pemicu sistem harus mengizinkan pengguna tertentu untuk masuk ke basis data primer dan replika. Pemicu sistem mencakup pemicu DDL, logon, dan peran basis data. Sebaiknya tambahkan kode pemicu Anda untuk mengecualikan pengguna yang tercantum dalam kode sampel berikut:

```
-- Determine who the user is
SELECT SYS_CONTEXT('USERENV','AUTHENTICATED_IDENTITY') INTO CURRENT_USER FROM DUAL;
-- The following users should always be able to login to either the Primary or
 Replica
IF CURRENT_USER IN ('master_user', 'SYS', 'SYSTEM', 'RDS_DATAGUARD', 'rdsdb') THEN
RETURN;
END IF;
```

- Pelacakan perubahan blok didukung untuk replika hanya baca, tetapi tidak untuk replika terpasang. Anda dapat mengubah replika terpasang menjadi replika hanya baca, lalu mengaktifkan pelacakan perubahan blok. Untuk informasi selengkapnya, lihat Mengaktifkan dan menonaktifkan pelacakan perubahan blok.

Bersiap membuat replika Oracle

Sebelum menggunakan replika, lakukan tugas berikut.

Topik

- Mengaktifkan pencadangan otomatis
- Mengaktifkan mode pencatatan log paksa
- Mengubah konfigurasi pencatatan log
- Mengatur parameter MAX_STRING_SIZE
- Merencanakan sumber daya komputasi dan penyimpanan

Mengaktifkan pencadangan otomatis

Sebelum instans DB dapat berfungsi sebagai instans DB sumber, pastikan untuk mengaktifkan pencadangan otomatis pada instans DB sumber. Pelajari cara melakukan prosedur ini di Mengaktifkan pencadangan otomatis.

Mengaktifkan mode pencatatan log paksa

Kami sarankan Anda mengaktifkan mode pencatatan log paksa. Dalam mode pencatatan log paksa, basis data Oracle menulis data pengulangan meski `NOLOGGING` digunakan dengan pernyataan bahasa definisi data (DDL).

Untuk mengaktifkan mode pencatatan log paksa

1. Masuk ke basis data Oracle Anda menggunakan alat klien seperti SQL Developer.
2. Aktifkan mode pencatatan log paksa dengan menjalankan prosedur berikut.

```
exec rdsadmin.rdsadmin_util.force_logging(p_enable => true);
```

Lihat informasi selengkapnya tentang prosedur ini di Mengatur logging paksa.

Mengubah konfigurasi pencatatan log

Untuk log pengulangan online n ukuran m, RDS secara otomatis membuat log siaga n+1 ukuran m pada instans DB primer dan semua replika. Setiap kali Anda mengubah konfigurasi pencatatan log pada instans primer, perubahan akan disebarkan secara otomatis ke replika.

Saat mengubah konfigurasi pencatatan log, pertimbangkan panduan berikut:

- Kami sarankan Anda menyelesaikan perubahan sebelum menjadikan instans DB sebagai sumber replika. RDS for Oracle juga mendukung pembaruan instans setelah menjadi sumber.
- Sebelum mengubah konfigurasi pencatatan log pada instans DB primer, periksa apakah penyimpanan pada setiap replika cukup untuk mengakomodasi konfigurasi baru.

Anda dapat mengubah konfigurasi pencatatan log instans DB menggunakan prosedur Amazon RDS `rdsadmin.rdsadmin_util.add_logfile` dan `rdsadmin.rdsadmin_util.drop_logfile`. Untuk informasi lebih lanjut, lihat Menambahkan log pengulangan online dan Menghapus log pengulangan online.

Mengatur parameter MAX_STRING_SIZE

Sebelum membuat replika Oracle, pastikan pengaturan parameter `MAX_STRING_SIZE` pada instans DB sumber dan replikanya sama. Anda dapat melakukannya dengan mengaitkan instans DB sumber dan replikanya dengan grup parameter yang sama. Jika grup parameter sumber dan replika berbeda, Anda dapat mengatur `MAX_STRING_SIZE` ke nilai yang sama. Untuk informasi selengkapnya tentang pengaturan parameter ini, lihat Mengaktifkan jenis extended data untuk instans DB baru.

Merencanakan sumber daya komputasi dan penyimpanan

Pastikan ukuran instans DB sumber dan replikanya sudah benar, dalam hal komputasi dan penyimpanan, sesuai dengan beban operasionalnya. Jika suatu replika mencapai kapasitas sumber daya komputasi, jaringan, atau penyimpanan, replika tersebut akan berhenti menerima atau menerapkan perubahan dari sumbernya. Amazon RDS for Oracle tidak melakukan intervensi untuk mengurangi keterlambatan replika yang tinggi antara instans DB sumber dan replikanya. Anda dapat memodifikasi sumber daya penyimpanan dan CPU replika secara terpisah dari sumbernya dan replika lainnya.

Membuat replika RDS for Oracle dalam mode terpasang

Secara default, replika Oracle berada dalam mode hanya baca. Untuk membuat replika dalam mode terpasang, gunakan konsol, AWS CLI, atau RDS API.

Konsol

Untuk membuat replika terpasang dari instans DB Oracle sumber

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Database.
3. Pilih instans DB Oracle yang akan digunakan sebagai sumber untuk replika terpasang.
4. Untuk Tindakan, pilih Buat replika.
5. Untuk Mode replika, pilih Terpasang.
6. Pilih pengaturan yang ingin Anda gunakan. Untuk Pengidentifikasi instans DB, masukkan nama replika baca. Sesuaikan pengaturan lain sesuai kebutuhan.
7. Untuk Wilayah, pilih Wilayah tempat replika terpasang akan diluncurkan.
8. Pilih ukuran dan jenis penyimpanan instans Anda. Sebaiknya gunakan kelas dan jenis penyimpanan instans DB yang sama seperti instans DB sumber untuk replika baca.

9. Untuk Deployment Multi-AZ, pilih Buat instans siaga untuk membuat replika Anda siaga di Zona Ketersediaan lain untuk dukungan failover bagi replika terpasang. Pembuatan replika terpasang sebagai instans Multi-AZ DB tidak bergantung pada apakah basis data sumber merupakan instans DB Multi-AZ.
10. Pilih pengaturan lain yang ingin Anda gunakan.
11. Pilih Buat replika.

Di halaman Basis data, replika terpasang memiliki peran sebagai Replika.

AWS CLI

Untuk membuat replika Oracle dalam mode mount, atur `--replica-mode` ke `mounted` dalam AWS CLI perintah create-db-instance-read -replica.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-instance-read-replica \
    --db-instance-identifier myreadreplica \
    --source-db-instance-identifier mydbinstance \
    --replica-mode mounted
```

Untuk Windows:

```
aws rds create-db-instance-read-replica ^
    --db-instance-identifier myreadreplica ^
    --source-db-instance-identifier mydbinstance ^
    --replica-mode mounted
```

Untuk mengubah replika hanya-baca ke status terpasang, atur `--replica-mode` ke `mounted` dalam perintah. AWS CLI modify-db-instance Untuk membuat replika terpasang dalam mode hanya baca, atur `--replica-mode` menjadi `open-read-only`.

RDS API

Untuk membuat replika Oracle dalam mode mount, tentukan `ReplicaMode=mounted` dalam operasi API RDS CreateDB. InstanceReadReplica

Mengubah mode replika RDS for Oracle

Untuk mengubah mode replika pada replika yang sudah ada, gunakan konsol, AWS CLI, atau RDS API. Saat Anda mengubah ke mode terpasang, replika memutuskan semua koneksi aktif. Saat Anda mengubah ke mode hanya baca, Amazon RDS akan menginisialisasi Active Data Guard.

Operasi perubahan dapat memakan waktu beberapa menit. Selama operasi, status instans DB berubah menjadi memodifikasi. Untuk informasi selengkapnya tentang perubahan status, lihat Melihat status instans DB Amazon RDS.

Konsol

Untuk mengubah mode replika Oracle dari terpasang menjadi hanya baca

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data.
3. Pilih basis data replika terpasang.
4. Pilih Ubah.
5. Untuk Mode replika, pilih Hanya baca.
6. Pilih pengaturan lain yang ingin Anda ubah.
7. Pilih Lanjutkan.
8. Untuk Penjadwalan perubahan, pilih Terapkan langsung.
9. Pilih Ubah instans DB.

AWS CLI

Untuk mengubah replika baca ke mode yang dipasang, atur `--replica-mode` ke `mounted` dalam AWS CLI perintah modify-db-instance. Untuk mengubah replika terpasang ke mode hanya baca, tetapkan `--replica-mode` menjadi `open-read-only`.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier myreadreplica \
```

```
    --replica-mode mode
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier myreadreplica ^
    --replica-mode mode
```

RDS API

Untuk mengubah replika hanya baca ke mode terpasang, tetapkan `ReplicaMode=mounted` di ModifyDBInstance. Untuk mengubah replika terpasang ke mode hanya baca, tetapkan `ReplicaMode=read-only`.

Bekerja dengan pencadangan replika RDS for Oracle

Anda dapat membuat dan memulihkan cadangan replika RDS for Oracle. Mendukung pencadangan otomatis dan snapshot manual. Untuk informasi selengkapnya, lihat Mencadangkan, memulihkan, dan mengekspor data. Bagian berikut menjelaskan perbedaan utama antara mengelola cadangan utama dan replika RDS for Oracle.

Mengaktifkan pencadangan replika RDS for Oracle

Pencadangan otomatis replika Oracle tidak diaktifkan secara default. Aktifkan pencadangan otomatis dengan mengatur periode retensi pencadangan ke nilai positif bukan nol.

Konsol

Untuk langsung mengaktifkan pencadangan otomatis

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data, lalu pilih instans DB atau klaster DB Multi-AZ yang ingin Anda ubah.
3. Pilih Ubah.
4. Untuk Periode retensi cadangan, pilih nilai positif bukan nol, misalnya 3 hari.
5. Pilih Lanjutkan.
6. Pilih Terapkan langsung.

7. Pilih Ubah instans DB atau Ubah klaster untuk menyimpan perubahan dan mengaktifkan pencadangan otomatis.

AWS CLI

Untuk mengaktifkan pencadangan otomatis, gunakan perintah AWS CLI `modify-db-instance` atau `modify-db-cluster`.

Sertakan parameter berikut:

- `--db-instance-identifier` (atau `--db-cluster-identifier` untuk klaster DB Multi-AZ)
- `--backup-retention-period`
- `--apply-immediately` atau `--no-apply-immediately`

Pada contoh berikut, kami mengaktifkan pencadangan otomatis dengan mengatur periode retensi cadangan menjadi tiga hari. Perubahan langsung diterapkan.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance  \
    --backup-retention-period 3 \
    --apply-immediately
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance  ^
    --backup-retention-period 3 ^
    --apply-immediately
```

RDS API

Untuk mengaktifkan pencadangan otomatis, gunakan operasi RDS API `ModifyDBInstance` atau `ModifyDBCluster` dengan parameter yang diperlukan sebagai berikut:

- `DBInstanceIdentifier` atau `DBClusterIdentifier`

- `BackupRetentionPeriod`

Memulihkan cadangan replika RDS for Oracle

Anda dapat memulihkan cadangan replika Oracle dengan cara yang sama seperti memulihkan cadangan instans utama. Untuk informasi selengkapnya, lihat hal berikut:

- Memulihkan dari snapshot DB
- Memulihkan instans DB dengan waktu yang ditentukan

Pertimbangan utama saat memulihkan cadangan replika adalah menentukan titik waktu yang akan dipulihkan. Waktu basis data adalah waktu transaksi terakhir yang diterapkan pada data dalam cadangan. Saat memulihkan cadangan replika, Anda memulihkan ke waktu basis data, bukan ke waktu saat pencadangan selesai. Perbedaannya cukup signifikan karena replika RDS for Oracle dapat tertinggal beberapa menit atau jam dari cadangan utama. Oleh karena itu, waktu basis data cadangan replika, dan juga titik waktu yang Anda pulihkan, mungkin jauh lebih awal daripada waktu pembuatan cadangan.

Untuk menemukan perbedaan antara waktu basis data dan waktu pembuatan, gunakan perintah `describe-db-snapshots`. Bandingkan `SnapshotDatabaseTime`, yang merupakan waktu basis data cadangan replika, dan bidang `OriginalSnapshotCreateTime`, yang merupakan transaksi terakhir yang diterapkan pada basis data utama. Contoh berikut menunjukkan perbedaan antara kedua waktu tersebut:

```
aws rds describe-db-snapshots \
    --db-instance-identifier my-oracle-replica
    --db-snapshot-identifier my-replica-snapshot

{
    "DBSnapshots": [
        {
            "DBSnapshotIdentifier": "my-replica-snapshot",
            "DBInstanceIdentifier": "my-oracle-replica",
            "SnapshotDatabaseTime": "2022-07-26T17:49:44Z",
            ...
            "OriginalSnapshotCreateTime": "2021-07-26T19:49:44Z"
        }
    ]
}
```

Melakukan switchover Oracle Data Guard

Switchover adalah pembalikan peran antara basis data primer dan basis data siaga. Selama switchover, basis data primer asli beralih ke peran siaga, sementara basis data siaga asli beralih ke peran utama.

Dalam lingkungan Oracle Data Guard, basis data primer mendukung satu atau beberapa basis data siaga. Anda dapat melakukan transisi peran terkelola berbasis switchover dari basis data primer ke basis data siaga. Switchover adalah pembalikan peran antara basis data primer dan basis data siaga. Selama switchover, basis data primer asli beralih ke peran siaga, sementara basis data siaga asli beralih ke peran utama.

Topik

Ikhtisar switchover Oracle Data Guard

Amazon RDS mendukung transisi peran berbasis switchover yang dikelola sepenuhnya untuk replika Oracle Database. Replika dapat berada di AWS Wilayah terpisah atau di Availability Zone (AZ) yang berbeda dari satu Wilayah. Anda hanya dapat memulai switchover ke basis data siaga yang terpasang atau hanya baca terbuka.

Switchover berbeda dengan promosi replika baca. Dalam peralihan, instance DB sumber dan replika mengubah peran. Dalam promosi, replika baca menjadi instance DB sumber, tetapi instance DB sumber tidak menjadi replika. Untuk informasi selengkapnya, lihat Mempromosikan replika baca menjadi instans DB mandiri.

Topik

- Cara kerja switchover Oracle Data Guard

Manfaat switchover Oracle Data Guard

Sama seperti replika baca RDS for Oracle, switchover terkelola bergantung pada Oracle Data Guard. Operasi ini dirancang untuk menghindari kehilangan data. Amazon RDS mengotomatiskan sejumlah aspek switchover berikut:

- Membalik peran basis data primer dan basis data siaga yang ditentukan akan membuat basis data siaga baru berada dalam kondisi yang sama (terpasang atau hanya baca) dengan basis data siaga asli
- Memastikan konsistensi data
- Mempertahankan konfigurasi replikasi Anda setelah transisi
- Mendukung pembalikan berulang, memungkinkan basis data siaga baru untuk kembali ke peran utama semula

Versi Oracle Database yang didukung

Switchover Oracle Data Guard didukung untuk rilis berikut:

- Oracle Database 19c
- Oracle Database 12c Rilis 2 (12.2)
- Oracle Database 12c Rilis 1 (12.1) menggunakan PSU 12.1.0.2.v10 atau yang lebih tinggi

AWS Dukungan wilayah

Peralihan Oracle Data Guard tersedia di Wilayah berikut: AWS

- Asia Pasifik (Mumbai)
- Asia Pasifik (Osaka)
- Asia Pasifik (Seoul)
- Asia Pasifik (Singapura)
- Asia Pasifik (Sydney)
- Asia Pasifik (Tokyo)
- Kanada (Pusat)

- Eropa (Frankfurt)
- Eropa (Irlandia)
- Eropa (London)
- Eropa (Paris)
- Eropa (Stockholm)
- Amerika Selatan (Sao Paulo)
- AS Timur (Virginia Utara)
- AS Timur (Ohio)
- AS Barat (California Utara)
- AS Barat (Oregon)
- AWS GovCloud (AS-Timur)
- AWS GovCloud (AS-Barat)

Biaya switchover Oracle Data Guard

Fitur switchover Oracle Data Guard tidak menimbulkan biaya tambahan. Oracle Database Enterprise Edition mencakup dukungan untuk basis data siaga dalam mode terpasang. Untuk membuka basis data siaga dalam mode hanya baca, Anda memerlukan opsi Oracle Data Guard.

Cara kerja switchover Oracle Data Guard

Switchover Oracle Data Guard adalah operasi yang dikelola sepenuhnya. Anda memulai switchover untuk basis data siaga dengan mengeluarkan perintah CLI `switchover-read-replica`. Kemudian Amazon RDS memodifikasi peran utama dan siaga dalam konfigurasi replikasi Anda.

Siaga asli dan utama asli adalah peran yang sudah ada sebelum switchover. Siaga baru dan utama baru adalah peran yang ada setelah switchover. Replika pengamat adalah basis data replika yang berfungsi sebagai basis data siaga di lingkungan Oracle Data Guard tetapi tidak berganti peran.

Topik

Tahapan switchover Oracle Data Guard

Untuk melakukan switchover, Amazon RDS harus melakukan langkah berikut:

1. Memblokir transaksi baru pada basis data primer asli. Selama switchover, Amazon RDS menginterupsi replikasi untuk semua basis data dalam konfigurasi Oracle Data Guard Anda. Selama switchover, basis data primer yang asli tidak dapat memproses permintaan penulisan.
2. Mengirimkan transaksi yang belum diterapkan ke basis data siaga asli, dan menerapkannya.
3. Memulai ulang basis data siaga baru dalam mode hanya baca atau mode terpasang. Mode bergantung pada status terbuka basis data siaga asli sebelum switchover.
4. Membuka basis data primer baru dalam mode baca/tulis.

Setelah switchover Oracle Data Guard

Amazon RDS mengubah peran basis data primer dan siaga. Anda bertanggung jawab untuk menghubungkan kembali aplikasi Anda dan melakukan konfigurasi lain yang diinginkan.

Topik

- Kriteria keberhasilan
- Koneksi ke basis data primer baru
- Konfigurasi basis data primer baru

Kriteria keberhasilan

Switchover Oracle Data Guard dinyatakan berhasil jika basis data siaga asli melakukan hal berikut:

- Melakukan peralihan peran sebagai basis data primer baru
- Menyelesaikan konfigurasi ulang

Untuk membatasi waktu henti, basis data primer baru Anda akan langsung diaktifkan. Karena Amazon RDS mengonfigurasi replika pengamat secara asinkron, replika ini mungkin akan aktif setelah basis data primer asli.

Koneksi ke basis data primer baru

Amazon RDS tidak akan menyebarkan koneksi basis data Anda saat ini ke basis data primer baru setelah switchover. Setelah switchover Oracle Data Guard selesai, hubungkan kembali aplikasi Anda ke basis data primer baru.

Konfigurasi basis data primer baru

Untuk melakukan switchover ke basis data primer baru, Amazon RDS mengubah mode basis data siaga asli menjadi terbuka. Yang berubah di dalam basis data hanya peran. Amazon RDS tidak menyiapkan fitur seperti replikasi Multi-AZ.

Jika Anda melakukan switchover ke replika lintas Wilayah dengan opsi yang berbeda, basis data primer baru akan mempertahankan opsinya sendiri. Amazon RDS tidak akan memigrasikan opsi pada basis data primer asli. Jika basis data primer asli memiliki opsi seperti SSL, NNE, OEM, dan OEM_AGENT, Amazon RDS tidak akan menyebarkannya ke basis data primer baru.

Mempersiapkan switchover Oracle Data Guard

Sebelum memulai switchover Oracle Data Guard, pastikan lingkungan replikasi Anda memenuhi persyaratan berikut:

- Basis data siaga asli dalam kondisi terpasang atau hanya baca terbuka.
- Pencadangan otomatis pada basis data siaga asli diaktifkan.
- Basis data utama asli dan basis data siaga asli tersedia.
- Tidak ada tindakan pemeliharaan yang tertunda pada basis data primer dan basis data siaga asli.
- Basis data siaga asli sedang direplikasi.
- Anda tidak memulai switchover ketika basis data primer atau basis data siaga sedang berada dalam siklus switchover. Jika basis data replika dikonfigurasi ulang setelah switchover, Amazon RDS tidak akan mengizinkan Anda untuk melakukan switchover lagi.

  > **Note**
  >
  > Replika pengamat adalah replika dalam konfigurasi Oracle Data Guard yang bukan merupakan target switchover. Replika pengamat dapat berada dalam kondisi apa pun selama switchover.

- Konfigurasi basis data siaga asli mirip dengan basis data primer yang diinginkan. Misalnya dalam sebuah skenario terdapat perbedaan opsi pada basis data primer dan basis data siaga. Setelah switchover selesai, Amazon RDS tidak secara otomatis mengonfigurasi ulang basis data primer baru agar memiliki opsi yang sama dengan basis data primer asli.
- Konfigurasikan deployment Multi-AZ yang Anda inginkan sebelum memulai switchover. Amazon RDS tidak mengelola Multi-AZ sebagai bagian dari switchover. Deployment Multi-AZ tidak berubah.

Anggap bahwa db_maz adalah basis data primer dalam deployment Multi-AZ, dan db_saz adalah replika Satu AZ. Anda memulai switchover dari db_maz ke db_saz. Setelah itu, db_maz adalah basis data replika Multi-AZ, dan db_saz adalah basis data primer Satu AZ. Basis data utama baru sekarang tidak dilindungi oleh deployment Multi-AZ.

- Dalam persiapan switchover lintas Wilayah, basis data primer tidak menggunakan grup opsi yang sama dengan instans DB di luar konfigurasi replikasi. Agar switchover lintas Wilayah berhasil, basis data primer saat ini dan replika bacanya harus menjadi satu-satunya instans DB yang menggunakan grup opsi basis data primer saat ini. Jika tidak, switchover tidak akan diizinkan oleh Amazon RDS.

Memulai switchover Oracle Data Guard

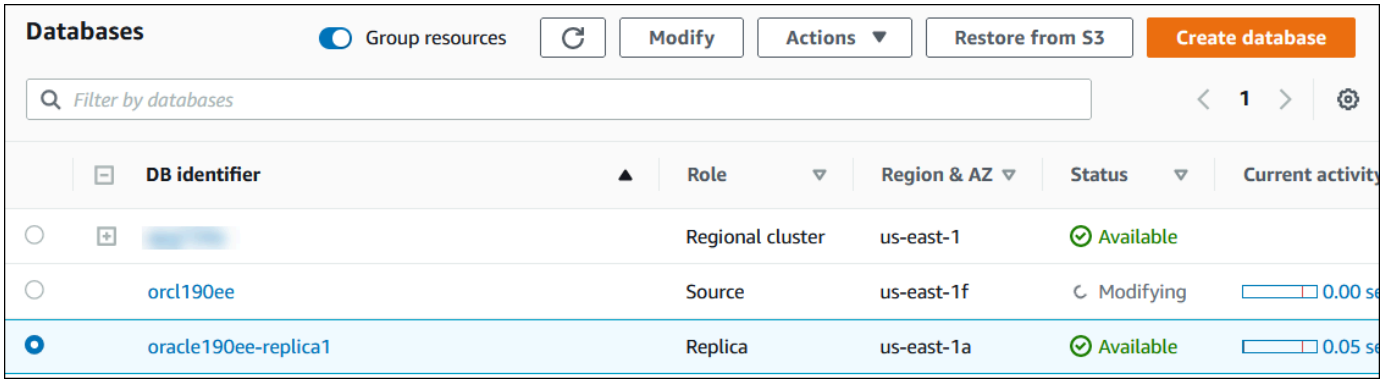
Anda dapat melakukan switchover replika baca RDS for Oracle menjadi peran utama, dan instans DB primer sebelumnya menjadi peran replika.

Konsol

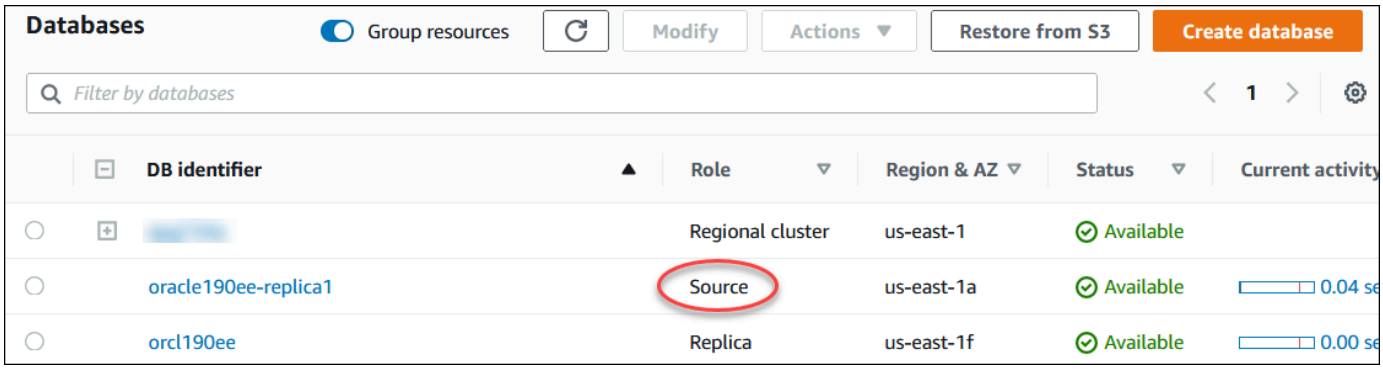
Untuk melakukan switchover replika baca Oracle ke peran DB primer

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di konsol Amazon RDS, pilih Basis Data.

   Panel Basis Data muncul. Setiap replika baca menampilkan Replika di kolom Peran.
3. Pilih replika baca yang ingin Anda alihkan ke peran utama.
4. Untuk Tindakan, pilih Alihkan replika.
5. Pilih Saya setuju. Kemudian pilih Alihkan replika.
6. Di halaman Basis Data, pantau progres switchover.



Setelah switchover selesai, peran yang menjadi target switchover akan berganti dari Replika menjadi Sumber.



AWS CLI

Untuk mengalihkan replika Oracle ke peran DB utama, gunakan perintah. AWS CLI [switchover-read-replica](#) Contoh berikut membuat replika Oracle bernama *replica-to-be-made-primary ke database primer* baru.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds switchover-read-replica \
    --db-instance-identifier replica-to-be-made-primary
```

Untuk Windows:

```
aws rds switchover-read-replica ^
    --db-instance-identifier replica-to-be-made-primary
```

API RDS

Untuk melakukan switchover replika Oracle ke peran DB primer, panggil operasi API Amazon RDS [SwitchoverReadReplica](#) dengan parameter yang diperlukan `DBInstanceIdentifier`. Parameter ini menentukan nama replika Oracle yang ingin Anda gunakan sebagai peran DB primer.

Memantau switchover Oracle Data Guard

Untuk memeriksa status instance Anda, gunakan perintah AWS `describe-db-instances` CLI. Perintah berikut memeriksa status instans DB *`orcl2`*. Sebelum switchover, basis data ini adalah basis data siaga, tetapi menjadi basis data primer baru setelah switchover.

```
aws rds describe-db-instances \
    --db-instance-identifier orcl2
```

Untuk mengonfirmasi bahwa switchover telah berhasil diselesaikan, buat kueri `V$DATABASE.OPEN_MODE`. Periksa apakah nilai untuk basis data primer baru adalah `READ WRITE`.

```
SELECT OPEN_MODE FROM V$DATABASE;
```

Untuk mencari peristiwa terkait switchover, gunakan perintah CLI. AWS `describe-events` Contoh berikut mencari peristiwa pada instans *`orcl2`*.

```
aws rds describe-events \
    --source-identifier orcl2 \
    --source-type db-instance
```

Pemecahan masalah replika RDS for Oracle

Bagian ini menjelaskan potensi masalah replikasi dan solusinya.

Topik

- Memantau keterlambatan replikasi Oracle
- Pemecahan masalah kegagalan replikasi Oracle setelah menambahkan atau memodifikasi pemicu

Memantau keterlambatan replikasi Oracle

Untuk memantau keterlambatan replikasi di Amazon CloudWatch, lihat metrik `ReplicaLag` Amazon RDS. Untuk informasi selengkapnya tentang waktu keterlambatan replikasi, lihat Memantau replikasi baca dan CloudWatch Metrik Amazon untuk Amazon RDS.

Untuk replika baca, jika waktu keterlambatan terlalu lama, kueri tampilan berikut:

- `V$ARCHIVED_LOG` – Menampilkan commit mana yang telah diterapkan ke replika baca.

- `V$DATAGUARD_STATS` – Menampilkan uraian mendetail tentang komponen penyusun metrik `ReplicaLag`.
- `V$DATAGUARD_STATUS` – Menampilkan hasil log dari proses replikasi internal Oracle.

Untuk replika terpasang, jika waktu keterlambatan terlalu lama, Anda tidak dapat melakukan kueri tampilan `V$`. Sebagai gantinya, lakukan hal berikut:

- Periksa metrik `ReplicaLag` di CloudWatch.
- Periksa file log peringatan untuk replika di konsol. Cari kesalahan dalam pesan pemulihan. Pesan termasuk nomor urutan log, yang dapat Anda bandingkan dengan nomor urutan utama. Untuk informasi selengkapnya, lihat File log basis data Oracle.

Pemecahan masalah kegagalan replikasi Oracle setelah menambahkan atau memodifikasi pemicu

Jika setelah Anda menambahkan atau memodifikasi pemicu terjadi kegagalan replikasi, permasalahannya mungkin terletak pada pemicu. Pastikan pemicu mengecualikan akun pengguna berikut, yang diperlukan oleh RDS untuk replikasi:

- Akun pengguna dengan hak istimewa administrator
- `SYS`
- `SYSTEM`
- `RDS_DATAGUARD`
- `rdsdb`

Untuk informasi selengkapnya, lihat Pertimbangan lainnya untuk replika RDS for Oracle.

Menambahkan opsi untuk instans DB Oracle

Di Amazon RDS, opsi berarti fitur tambahan. Pada bagian berikut ini, Anda dapat menemukan deskripsi opsi yang dapat Anda tambahkan ke instans Amazon RDS yang menjalankan mesin DB Oracle.

Topik

Ringkasan opsi DB Oracle

Untuk mengaktifkan opsi untuk basis data Oracle Anda, tambahkan semua opsi ke grup opsi, lalu kaitkan grup opsi dengan instans DB Anda. Untuk informasi selengkapnya, lihat Menggunakan grup opsi.

Topik

Ringkasan opsi Oracle Database

Anda dapat menambahkan opsi berikut untuk instans DB Oracle.

Opsi	ID Opsi
Integrasi Amazon S3	`S3_INTEGRATION`
Oracle Application Express (APEX)	`APEX` `APEX-DEV`
Oracle Enterprise Manager	`OEM` `OEM_AGENT`
Mesin virtual Oracle Java	`JVM`
Keamanan Label Oracle	`OLS`
Oracle Locator	`LOCATOR`
Oracle Multimedia	`MULTIMEDIA`
Enkripsi jaringan asli Oracle	`NATIVE_NETWORK_ENCRYPTION`
Oracle OLAP	`OLAP`
Lapisan Soket Aman Oracle	`SSL`
Oracle Spatial	`SPATIAL`
Oracle SQLT	`SQLT`

Opsi	ID Opsi
Oracle Statspack	STATSPACK
Zona waktu Oracle	Timezone
Pemutakhiran otomatis file zona waktu Oracle	TIMEZONE_FILE_AUTOUPGRADE
Enkripsi Data Transparan Oracle	TDE
Oracle UTL_MAIL	UTL_MAIL
DB XML Oracle	XMLDB

Opsi yang didukung untuk edisi yang berbeda

RDS for Oracle mencegah penambahan opsi ke edisi tertentu jika tidak didukung. Untuk mengetahui opsi RDS mana yang didukung dalam edisi Oracle Database yang berbeda, gunakan perintah `aws rds describe-option-group-options`. Contoh berikut mencantumkan opsi yang didukung untuk Basis Data Oracle 19c Edisi Perusahaan.

```
aws rds describe-option-group-options \
    --engine-name oracle-ee \
    --major-engine-version 19
```

Untuk informasi selengkapnya, lihat describe-option-group-optionsdi Referensi Perintah AWS CLI.

Persyaratan memori untuk opsi tertentu

Beberapa opsi memerlukan memori tambahan untuk dijalankan pada instans DB Anda. Misalnya, Oracle Enterprise Manager Database Control menggunakan sekitar 300 MB RAM. Jika Anda mengaktifkan opsi ini untuk instans DB berukuran kecil, Anda mungkin akan mengalami masalah kinerja karena kendala memori. Anda dapat menyesuaikan parameter Oracle sehingga basis datanya membutuhkan lebih sedikit RAM. Alternatifnya, Anda dapat menaikkan skala ke instans DB yang lebih besar.

Integrasi Amazon S3

Anda bisa mentransfer file di antara instans DB RDS for Oracle dan bucket Amazon S3. Anda dapat menggunakan integrasi Amazon S3 dengan fitur Oracle Database seperti Oracle Data Pump. Sebagai contoh, Anda dapat mengunduh file Data Pump dari Amazon S3 ke instans DB RDS for Oracle. Untuk informasi selengkapnya, lihat Mengimpor data ke Oracle di Amazon RDS.

> **Note**
>
> Instans DB dan bucket Amazon S3 Anda harus berada di Wilayah AWS yang sama.

Topik

Mengonfigurasi izin IAM untuk integrasi RDS for Oracle dengan Amazon S3

Agar RDS untuk Oracle dapat diintegrasikan dengan Amazon S3, instans DB Anda harus memiliki akses ke bucket Amazon S3. Amazon VPC yang digunakan oleh instans DB Anda tidak perlu memberikan akses ke titik akhir Amazon S3.

RDS untuk Oracle mendukung pengunggahan file dari instans DB di satu akun ke bucket Amazon S3 di akun yang berbeda. Langkah tambahan yang diperlukan akan disebutkan di bagian berikut.

Topik

Langkah 1: Buat kebijakan IAM untuk peran Amazon RDS Anda

Pada langkah ini, Anda membuat kebijakan AWS Identity and Access Management (IAM) dengan izin yang diperlukan untuk mentransfer file dari bucket Amazon S3 ke instans DB RDS. Pada langkah ini, Anda dianggap telah membuat bucket S3.

Sebelum membuat kebijakan, catat potongan informasi berikut:

- Amazon Resource Name (ARN) untuk bucket Anda
- ARN untuk kunci AWS KMS Anda, jika bucket Anda menggunakan enkripsi SSE-KMS atau SSE-S3

> **Note**
>
> Instans DB RDS for Oracle tidak dapat mengakses bucket Amazon S3 yang dienkripsi dengan SSE-C.

Untuk informasi selengkapnya, lihat Melindungi data menggunakan enkripsi sisi server dalam Panduan Pengguna Amazon Simple Storage Service.

Konsol

Untuk membuat kebijakan IAM untuk mengizinkan Amazon RDS mengakses bucket Amazon S3

1. Buka Konsol Manajemen IAM.
2. Di bagian Manajemen akses, pilih Kebijakan.
3. Pilih Buat Kebijakan.
4. Pada tab Editor visual, pilih Pilih layanan, lalu pilih S3.
5. Untuk Tindakan, pilih Perluas semua, lalu pilih izin bucket dan izin objek yang diperlukan untuk mentransfer file dari bucket Amazon S3 ke Amazon RDS. Sebagai contoh, lakukan hal berikut:
   - Perluas Daftar, lalu pilih ListBucket.
   - Perluas Baca, lalu pilih GetObject.
   - Perluas Tulis, lalu pilih PutObjectdan DeleteObject.
   - Perluas manajemen Izin, lalu pilih PutObjectAcl. Izin ini diperlukan jika Anda ingin mengunggah file ke bucket milik akun yang berbeda, dan akun ini memerlukan kontrol penuh terhadap isi bucket.

Izin objek adalah izin untuk operasi objek di Amazon S3. Izin ini harus diberikan untuk objek di dalam bucket, bukan untuk bucket itu sendiri. Untuk informasi selengkapnya, lihat Izin untuk operasi objek.

6. Pilih Sumber daya, lalu lakukan hal berikut:

   a. Pilih Spesifik.

   b. Untuk bucket, pilih Tambah ARN. Masukkan ARN bucket. Nama bucket terisi secara otomatis. Kemudian pilih Tambahkan.

   c. Jika muncul sumber daya objek, pilih Tambah ARN untuk menambahkan sumber daya secara manual atau pilih Apa pun.

   > **Note**
   >
   > Anda dapat mengatur Amazon Resource Name (ARN) ke nilai ARN yang lebih spesifik agar Amazon RDS hanya dapat mengakses file atau folder tertentu dalam bucket Amazon S3. Untuk informasi selengkapnya tentang cara penentuan kebijakan akses untuk Amazon S3, lihat Mengelola izin akses ke sumber daya Amazon S3 Anda.

7. (Opsional) Pilih Tambahkan izin tambahan untuk menambahkan sumber daya ke kebijakan. Sebagai contoh, lakukan hal berikut:

   a. Jika bucket Anda dienkripsi dengan kuci KMS kustom, pilih KMS untuk layanan.

   b. Untuk Tindakan manual, pilih opsi berikut:

      - Enkripsi
      - ReEncrypt dari dan ReEncrypt ke
      - Dekripsi
      - DescribeKey
      - GenerateDataKey

   c. Untuk Sumber daya, pilih Spesifik.

   d. Untuk kunci, pilih Tambahkan ARN. Masukkan ARN kunci kustom Anda sebagai sumber daya, lalu pilih Tambahkan.

Untuk informasi selengkapnya, lihat Melindungi Data Menggunakan Enkripsi Sisi Server dengan kunci KMS yang Disimpan di AWS Key Management Service (SSE-KMS) dalam Panduan Pengguna Amazon Simple Storage Service.

e. Agar Amazon RDS dapat mengakses bucket lain, tambahkan ARN untuk bucket yang bersangkutan. Jika mau, Anda juga dapat memberikan akses ke semua bucket dan objek di Amazon S3.

8. Pilih Berikutnya: Tag lalu Berikutnya: Tinjau.
9. Untuk Nama, masukkan nama kebijakan IAM Anda, misalnya `rds-s3-integration-policy`. Anda menggunakan nama ini saat membuat peran IAM untuk dikaitkan dengan instans DB Anda. Anda juga dapat menambahkan nilai Deskripsi opsional.
10. Pilih Buat kebijakan.

AWS CLI

Buat kebijakan AWS Identity and Access Management (IAM) yang memberi Amazon RDS akses ke bucket Amazon S3. Setelah membuat kebijakan, catat ARN-nya. ARN ini diperlukan pada langkah berikutnya.

Sertakan tindakan yang sesuai dalam kebijakan berdasarkan jenis akses yang diperlukan:

- `GetObject` – Diperlukan untuk mentransfer file dari bucket Amazon S3 ke Amazon RDS.
- `ListBucket` – Diperlukan untuk mentransfer file dari bucket Amazon S3 ke Amazon RDS.
- `PutObject` – Diperlukan untuk mentransfer file dari Amazon RDS ke bucket Amazon S3.

Perintah AWS CLI berikut membuat kebijakan IAM bernama *`rds-s3-integration-policy`* dengan opsi ini. Kebijakan ini memberikan akses ke bucket bernama *`your-s3-bucket-arn`*.

Example

Untuk Linux, macOS, atau Unix:

```
aws iam create-policy \
   --policy-name rds-s3-integration-policy \
   --policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
```

```
        "Sid": "s3integration",
        "Action": [
          "s3:GetObject",
          "s3:ListBucket",
          "s3:PutObject"
        ],
        "Effect": "Allow",
        "Resource": [
          "arn:aws:s3:::your-s3-bucket-arn",
          "arn:aws:s3:::your-s3-bucket-arn/*"
        ]
      }
    ]
  }'
```

Contoh berikut mencakup izin untuk kunci KMS kustom.

```
aws iam create-policy \
  --policy-name rds-s3-integration-policy \
  --policy-document '{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Sid": "s3integration",
        "Action": [
          "s3:GetObject",
          "s3:ListBucket",
          "s3:PutObject",
          "kms:Decrypt",
          "kms:Encrypt",
          "kms:ReEncrypt",
          "kms:GenerateDataKey",
          "kms:DescribeKey",
        ],
        "Effect": "Allow",
        "Resource": [
          "arn:aws:s3:::your-s3-bucket-arn",
          "arn:aws:s3:::your-s3-bucket-arn/*",
          "arn:aws:kms:::your-kms-arn"
        ]
      }
    ]
  }'
```

Untuk Windows:

```
aws iam create-policy ^
   --policy-name rds-s3-integration-policy ^
   --policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Sid": "s3integration",
         "Action": [
           "s3:GetObject",
           "s3:ListBucket",
           "s3:PutObject"
         ],
         "Effect": "Allow",
         "Resource": [
           "arn:aws:s3:::your-s3-bucket-arn",
           "arn:aws:s3:::your-s3-bucket-arn/*"
         ]
       }
     ]
   }'
```

Contoh berikut mencakup izin untuk kunci KMS kustom.

```
aws iam create-policy ^
   --policy-name rds-s3-integration-policy ^
   --policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Sid": "s3integration",
         "Action": [
           "s3:GetObject",
           "s3:ListBucket",
           "s3:PutObject",
           "kms:Decrypt",
           "kms:Encrypt",
           "kms:ReEncrypt",
           "kms:GenerateDataKey",
           "kms:DescribeKey",
         ],
         "Effect": "Allow",
```

```
        "Resource": [
          "arn:aws:s3:::your-s3-bucket-arn",
          "arn:aws:s3:::your-s3-bucket-arn/*",
          "arn:aws:kms:::your-kms-arn"
        ]
      }
    ]
  }'
```

Langkah 2: (Opsional) Buat kebijakan IAM untuk bucket Amazon S3

Langkah ini hanya diperlukan dalam kondisi berikut:

- Anda ingin mengunggah file ke bucket Amazon S3 dari satu akun (akun A) dan mengaksesnya dari akun lain (akun B).
- Akun B memiliki bucket.
- Akun B memerlukan kontrol penuh atas objek yang dimasukkan ke dalam bucket tersebut.

Jika kondisi sebelumnya tidak berlaku untuk Anda, lanjutkan ke Langkah 3: Buat peran IAM untuk instans DB Anda dan lampirkan kebijakan Anda.

Untuk membuat kebijakan bucket, pastikan Anda memiliki hal berikut:

- ID akun untuk akun A
- Nama pengguna untuk akun A
- Nilai ARN untuk bucket Amazon S3 di akun B

Konsol

Untuk membuat atau mengedit kebijakan bucket

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di https://console.aws.amazon.com/s3/.
2. Di daftar Bucket, pilih nama bucket yang ingin Anda buatkan kebijakan bucket atau yang kebijakan bucket-nya ingin Anda edit.
3. Pilih Izin.
4. Di bagian Kebijakan bucket, pilih Edit. Opsi ini akan membuka halaman Edit kebijakan bucket.

5. Di halaman Edit kebijakan bucket, jelajahi Contoh kebijakan dalam Panduan Pengguna Amazon S3, pilih Pembuat kebijakan untuk membuat kebijakan secara otomatis, atau edit JSON di bagian Kebijakan.

   Jika Anda memilih Pembuat kebijakan, Pembuat Kebijakan AWS akan terbuka di jendela baru:

   a. Di halaman Pembuat Kebijakan AWS, di bagian Pilih Jenis Kebijakan, pilih Kebijakan Bucket S3.

   b. Tambahkan pernyataan dengan memasukkan informasi di bidang yang tersedia, lalu pilih Tambahkan Pernyataan. Ulangi langkah ini sesuai jumlah pernyataan yang ingin Anda tambahkan. Untuk informasi selengkapnya tentang bidang ini, lihat Referensi elemen kebijakan JSON IAM dalam Panduan Pengguna IAM.

      > **Note**
      >
      > Untuk memudahkan penggunaan, halaman Edit kebijakan bucket menampilkan ARN Bucket (Amazon Resource Name) bucket saat ini di atas bidang teks Kebijakan. Anda dapat menyalin ARN ini untuk digunakan dalam pernyataan di halaman Pembuat Kebijakan AWS.

   c. Setelah Anda selesai menambah pernyataan, pilih Buat Kebijakan.

   d. Salin teks kebijakan yang dihasilkan, pilih Tutup, dan kembali ke halaman Edit kebijakan bucket di konsol Amazon S3.

6. Di kotak Kebijakan, edit kebijakan yang ada atau tempel kebijakan bucket dari Pembuat kebijakan. Pastikan peringatan keamanan, kesalahan, peringatan umum, dan saran telah ditangani sebelum menyimpan kebijakan.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Example permissions",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::account-A-ID:account-A-user"
      },
      "Action": [
        "s3:PutObject",
        "s3:PutObjectAcl"
```

```
      ],
      "Resource": [
        "arn:aws:s3:::account-B-bucket-arn",
        "arn:aws:s3:::account-B-bucket-arn/*"
      ]
    }
  ]
}
```

7. Pilih Simpan perubahan, yang akan membawa Anda kembali ke halaman Izin Bucket.

Langkah 3: Buat peran IAM untuk instans DB Anda dan lampirkan kebijakan Anda

Pada langkah ini, Anda dianggap telah membuat kebijakan IAM di Langkah 1: Buat kebijakan IAM untuk peran Amazon RDS Anda. Pada langkah ini, Anda membuat peran untuk instans DB RDS for Oracle, kemudian melampirkan kebijakan Anda ke peran tersebut.

Konsol

Untuk membuat peran IAM agar Amazon RDS dapat mengakses bucket Amazon S3

1. Buka Konsol Manajemen IAM.
2. Di panel navigasi, pilih Peran.
3. Pilih Buat peran.
4. Pilih Layanan AWS.
5. Untuk Kasus penggunaan untuk layanan AWS lainnya:, pilih RDS lalu RDS – Tambahkan Peran ke Basis Data. Lalu pilih Selanjutnya.
6. Untuk Cari di bagian Kebijakan izin, masukkan nama kebijakan IAM yang telah Anda buat di Langkah 1: Buat kebijakan IAM untuk peran Amazon RDS Anda, lalu pilih kebijakan tersebut saat muncul dalam daftar. Lalu pilih Selanjutnya.
7. Untuk Nama peran, masukkan nama untuk peran IAM Anda, misalnya, `rds-s3-integration-role`. Anda juga dapat menambahkan nilai Deskripsi opsional.
8. Pilih Buat peran.

AWS CLI

Untuk membuat peran dan melampirkan kebijakan ke peran ini

1. Buat peran IAM yang dapat digunakan oleh Amazon RDS atas nama Anda untuk mengakses bucket Amazon S3 Anda.

   Sebaiknya gunakan kunci konteks kondisi global `aws:SourceArn` dan `aws:SourceAccount` dalam hubungan kepercayaan berbasis sumber daya untuk membatasi izin layanan ke sumber daya tertentu. Ini adalah cara paling efektif untuk melindungi dari masalah confused deputy.

   Anda dapat menggunakan kedua kunci konteks kondisi global dan memiliki nilai `aws:SourceArn` yang berisi ID akun. Dalam hal ini, nilai `aws:SourceAccount` dan akun dalam nilai `aws:SourceArn` harus menggunakan ID akun yang sama ketika digunakan dalam pernyataan yang sama.

   - Gunakan `aws:SourceArn` jika Anda ingin akses lintas layanan untuk satu sumber daya.
   - Gunakan `aws:SourceAccount` jika Anda ingin mengizinkan sumber daya apa pun di akun tersebut dikaitkan dengan penggunaan lintas layanan.

   Dalam hubungan kepercayaan, pastikan untuk menggunakan kunci konteks kondisi global `aws:SourceArn` dengan Amazon Resource Name (ARN) penuh pada sumber daya yang mengakses peran.

   Perintah AWS CLI berikut membuat peran bernama *`rds-s3-integration-role`* untuk tujuan ini.

   Example

   Untuk Linux, macOS, atau Unix:

```
aws iam create-role \
   --role-name rds-s3-integration-role \
   --assume-role-policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
            "Service": "rds.amazonaws.com"
```

```
          },
          "Action": "sts:AssumeRole",
          "Condition": {
              "StringEquals": {
                  "aws:SourceAccount": my_account_ID,
                  "aws:SourceArn": "arn:aws:rds:Region:my_account_ID:db:dbname"
              }
          }
      }
    ]
  }'
```

Untuk Windows:

```
aws iam create-role ^
   --role-name rds-s3-integration-role ^
   --assume-role-policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
            "Service": "rds.amazonaws.com"
          },
         "Action": "sts:AssumeRole",
         "Condition": {
             "StringEquals": {
                 "aws:SourceAccount": my_account_ID,
                 "aws:SourceArn": "arn:aws:rds:Region:my_account_ID:db:dbname"
             }
         }
       }
     ]
   }'
```

Untuk informasi selengkapnya, lihat Membuat peran untuk mendelegasikan izin ke pengguna IAM dalam Panduan Pengguna IAM.

2. Setelah peran dibuat, catat ARN peran tersebut. ARN ini diperlukan pada langkah berikutnya.
3. Lampirkan kebijakan yang Anda buat ke peran yang Anda buat.

Perintah AWS CLI berikut melampirkan kebijakan ke peran bernama *rds-s3-integration-role*.

Example

Untuk Linux, macOS, atau Unix:

```
aws iam attach-role-policy \
   --policy-arn your-policy-arn \
   --role-name rds-s3-integration-role
```

Untuk Windows:

```
aws iam attach-role-policy ^
   --policy-arn your-policy-arn ^
   --role-name rds-s3-integration-role
```

Ganti *your-policy-arn* dengan ARN kebijakan yang Anda catat di langkah sebelumnya.

Langkah 4: Kaitkan peran IAM Anda dengan instans DB RDS for Oracle

Langkah terakhir dalam mengonfigurasi izin untuk integrasi Amazon S3 adalah mengaitkan peran IAM Anda dengan instans DB. Perhatikan persyaratan berikut:

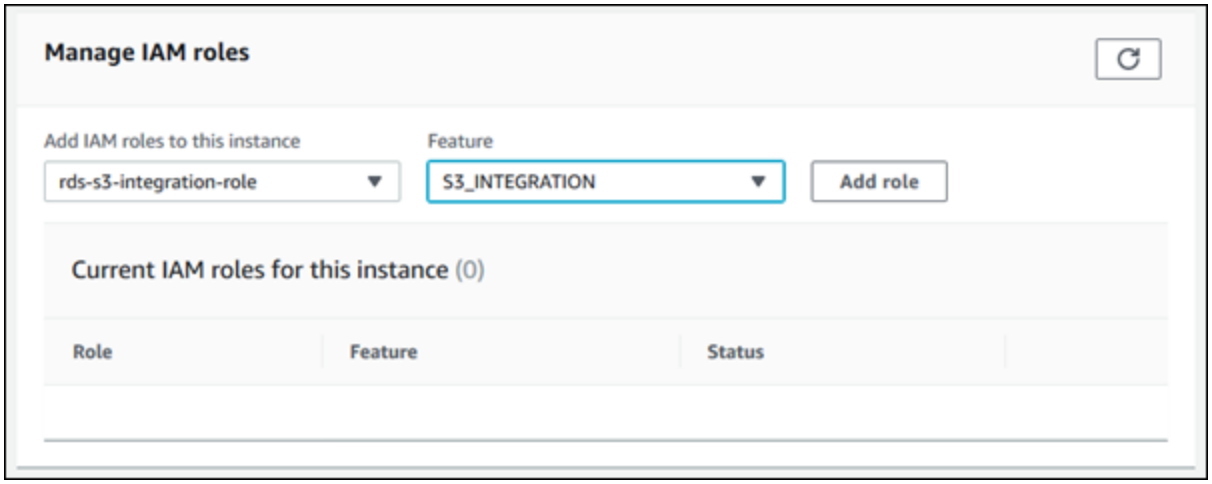
- Anda harus memiliki akses ke peran IAM yang telah memiliki kebijakan izin Amazon S3 yang diperlukan.
- Anda hanya dapat mengaitkan satu peran IAM dengan instans DB RDS for Oracle Anda dalam satu waktu.
- Instans DB Anda harus dalam status Tersedia.

Konsol

Untuk mengaitkan peran IAM Anda dengan instans DB RDS for Oracle Anda

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Pilih Basis data dari panel navigasi.

3. Pilih nama instans DB RDS for Oracle untuk menampilkan detailnya.
4. Pada tab Konektivitas & keamanan, gulir ke Kelola peran IAM di bagian bawah halaman.
5. Untuk Tambahkan peran IAM ke instans ini, pilih peran yang Anda buat di Langkah 3: Buat peran IAM untuk instans DB Anda dan lampirkan kebijakan Anda.
6. Untuk Fitur, pilih S3_INTEGRATION.



7. Pilih Tambahkan peran.

AWS CLI

Perintah AWS CLI berikut menambahkan peran ke instans DB Oracle bernama *mydbinstance*.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds add-role-to-db-instance \
   --db-instance-identifier mydbinstance \
   --feature-name S3_INTEGRATION \
   --role-arn your-role-arn
```

Untuk Windows:

```
aws rds add-role-to-db-instance ^
   --db-instance-identifier mydbinstance ^
   --feature-name S3_INTEGRATION ^
   --role-arn your-role-arn
```

Ganti *your-role-arn* dengan peran ARN yang Anda catat di langkah sebelumnya. `S3_INTEGRATION` harus ditentukan untuk opsi `--feature-name`.

Menambahkan opsi integrasi Amazon S3

Untuk mengintegrasikan Amazon RDS for Oracle dengan Amazon S3, instans DB Anda harus dikaitkan dengan grup opsi yang menyertakan opsi `S3_INTEGRATION`.

Konsol

Untuk mengonfigurasi grup opsi untuk integrasi Amazon S3

1. Buat grup opsi baru atau identifikasi grup opsi yang ada yang dapat ditambahi opsi `S3_INTEGRATION`.

   Untuk informasi tentang cara membuat grup opsi, lihat Membuat grup opsi.
2. Tambahkan opsi `S3_INTEGRATION` ke kelompok opsi.

   Untuk informasi tentang penambahan opsi ke grup opsi, lihat Menambahkan opsi ke grup opsi.
3. Buat instans DB RDS for Oracle yang baru dan kaitkan dengan grup opsi, atau modifikasi instans DB RDS for Oracle untuk dikaitkan dengan grup opsi.

   Untuk informasi tentang pembuatan instans DB, lihat Membuat instans DB Amazon RDS.

   Untuk informasi tentang modifikasi instans DB, lihat Memodifikasi instans DB Amazon RDS.

AWS CLI

Untuk mengonfigurasi grup opsi untuk integrasi Amazon S3

1. Buat grup opsi baru atau identifikasi grup opsi yang ada yang dapat ditambahi opsi `S3_INTEGRATION`.

   Untuk informasi tentang cara membuat grup opsi, lihat Membuat grup opsi.
2. Tambahkan opsi `S3_INTEGRATION` ke grup opsi.

   Misalnya, perintah AWS CLI berikut menambahkan opsi `S3_INTEGRATION` ke grup opsi bernama **`myoptiongroup`**.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds add-option-to-option-group \
   --option-group-name myoptiongroup \
   --options OptionName=S3_INTEGRATION,OptionVersion=1.0
```

Untuk Windows:

```
aws rds add-option-to-option-group ^
   --option-group-name myoptiongroup ^
   --options OptionName=S3_INTEGRATION,OptionVersion=1.0
```

3. Buat instans DB RDS for Oracle yang baru dan kaitkan dengan grup opsi, atau modifikasi instans DB RDS for Oracle untuk dikaitkan dengan grup opsi.

   Untuk informasi tentang pembuatan instans DB, lihat Membuat instans DB Amazon RDS.

   Untuk informasi tentang modifikasi instans DB RDS for Oracle, lihat Memodifikasi instans DB Amazon RDS.

Mentransfer file antara Amazon RDS for Oracle dan bucket Amazon S3

Untuk mentransfer file antara instans DB RDS for Oracle dan bucket Amazon S3, Anda dapat menggunakan paket `rdsadmin_s3_tasks` Amazon RDS. Anda dapat mengompres file dengan GZIP saat mengunggahnya, dan membuka kompresinya saat mengunduh.

Topik

- Persyaratan dan batasan transfer file
- Mengunggah file dari instans DB RDS for Oracle ke bucket Amazon S3
- Mengunduh file dari bucket Amazon S3 ke instans DB Oracle
- Memantau status transfer file

Persyaratan dan batasan transfer file

Sebelum mentransfer file antara instans DB Anda dan bucket Amazon S3, perhatikan hal-hal berikut:

- Paket `rdsadmin_s3_tasks` mentransfer file yang berada di dalam satu direktori. Anda tidak dapat menyertakan subdirektori dalam transfer.
- Ukuran objek maksimum dalam bucket Amazon S3 adalah 5 TB.
- Tugas yang dibuat oleh `rdsadmin_s3_tasks` dijalankan secara asinkron.
- Anda dapat mengunggah file dari direktori Data Pump, seperti `DATA_PUMP_DIR`, atau direktori lain yang dibuat oleh pengguna. Anda tidak dapat mengunggah file dari direktori yang digunakan oleh proses latar belakang Oracle, seperti direktori `adump`, `bdump`, atau `trace`.
- Batas pengunduhan adalah 2.000 file per panggilan prosedur untuk `download_from_s3`. Jika Anda perlu mengunduh lebih dari 2.000 file dari Amazon S3, bagi unduhan menjadi beberapa tindakan terpisah, dengan tidak lebih dari 2.000 file per panggilan prosedur.
- Jika terdapat file dengan nama yang sama di folder unduhan Anda, `download_from_s3` tidak akan memproses pengunduhan. Untuk menghapus file dari direktori unduhan, gunakan prosedur PL/SQL UTL_FILE.FREMOVE.

Mengunggah file dari instans DB RDS for Oracle ke bucket Amazon S3

Untuk mengunggah file dari instans DB Anda ke bucket Amazon S3, gunakan prosedur `rdsadmin.rdsadmin_s3_tasks.upload_to_s3`. Misalnya, Anda dapat mengunggah file cadangan Oracle Recovery Manager (RMAN) atau file Oracle Data Pump. Untuk informasi selengkapnya tentang penggunaan objek, lihat Panduan Pengguna Amazon Simple Storage Service. Untuk informasi selengkapnya tentang proses pencadangan RMAN, lihat Melakukan tugas RMAN umum untuk instans DB Oracle.

Prosedur `rdsadmin.rdsadmin_s3_tasks.upload_to_s3` memiliki parameter berikut.

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
`p_bucket_name`	VARCHAR2	–	wajib	Nama bucket Amazon S3 tempat file diunggah.
`p_directory_name`	VARCHAR2	–	wajib	Nama objek direktori Oracle asal file yang akan diunggah. Direktori dapat berupa objek direktori yang dibuat pengguna atau direktori

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
				Data Pump, seperti `DATA_PUMP_DIR` . Anda tidak dapat mengunggah file dari direktori yang digunakan oleh proses latar belakang, seperti `adump`, `bdump`, dan `trace`. **Note** Anda hanya dapat mengunggah file dari direktori yang ditentukan. Anda tidak dapat mengunggah file di subdirektori dalam direktori yang ditentukan.

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
`p_s3_prefix`	VARCHAR2	–	wajib	Awalan nama file Amazon S3 tempat file diunggah. Awalan kosong mengunggah semua file ke tingkat teratas di bucket Amazon S3 yang ditentukan dan tidak menambahkan awalan pada nama file. Misalnya, jika prefiksnya adalah `folder_1/oradb` , file akan diunggah ke `folder_1`. Dalam hal ini, prefiks `oradb` ditambahkan ke setiap file.
`p_prefix`	VARCHAR2	–	wajib	Prefiks nama file yang harus sama dengan nama file yang akan diunggah. Prefiks kosong akan mengunggah semua file dalam direktori yang ditentukan.
`p_compression_level`	ANGKA	`0`	opsional	Tingkat kompresi GZIP. Nilai yang valid berkisar dari `0` sampai `9`: • `0` – Tanpa kompresi • `1` – Kompresi tercepat • `9` – Kompresi tertinggi

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
`p_bucket_owner_full_control`	VARCHAR2	–	opsional	Pengaturan kontrol akses untuk bucket. Satu-satunya nilai yang valid adalah null atau `FULL_CONTROL` . Pengaturan ini diperlukan hanya jika Anda mengunggah file dari satu akun (akun A) ke dalam sebuah bucket milik akun yang berbeda (akun B), dan akun B memerlukan kontrol penuh atas file tersebut.

Nilai yang dikembalikan untuk prosedur `rdsadmin.rdsadmin_s3_tasks.upload_to_s3` adalah ID tugas.

Contoh berikut mengunggah semua file di direktori *`DATA_PUMP_DIR`* ke bucket Amazon S3 bernama *`mys3bucket`*. File tidak dikompresi.

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
      p_bucket_name    =>  'mys3bucket',
      p_prefix         =>  '',
      p_s3_prefix      =>  '',
      p_directory_name =>  'DATA_PUMP_DIR')
   AS TASK_ID FROM DUAL;
```

Contoh berikut mengunggah semua file dengan prefiks *`db`* dalam direktori *`DATA_PUMP_DIR`* ke bucket Amazon S3 bernama *`mys3bucket`*. Amazon RDS menerapkan tingkat kompresi GZIP tertinggi ke file.

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
      p_bucket_name       =>  'mys3bucket',
      p_prefix            =>  'db',
      p_s3_prefix         =>  '',
```

```
      p_directory_name    =>  'DATA_PUMP_DIR',
      p_compression_level =>  9)
   AS TASK_ID FROM DUAL;
```

Contoh berikut mengunggah semua file di direktori *DATA_PUMP_DIR* ke bucket Amazon S3 bernama *mys3bucket*. File tersebut diunggah ke folder `dbfiles`. Dalam contoh ini, tingkat kompresi GZIP adalah *1*, yang merupakan tingkat kompresi tercepat.

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
      p_bucket_name       =>  'mys3bucket',
      p_prefix            =>  '',
      p_s3_prefix         =>  'dbfiles/',
      p_directory_name    =>  'DATA_PUMP_DIR',
      p_compression_level =>  1)
   AS TASK_ID FROM DUAL;
```

Contoh berikut mengunggah semua file di direktori *DATA_PUMP_DIR* ke bucket Amazon S3 bernama *mys3bucket*. File tersebut diunggah ke folder `dbfiles` dan `ora` ditambahkan ke awal setiap nama file. Tidak ada kompresi yang diterapkan.

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
      p_bucket_name    =>  'mys3bucket',
      p_prefix         =>  '',
      p_s3_prefix      =>  'dbfiles/ora',
      p_directory_name =>  'DATA_PUMP_DIR')
   AS TASK_ID FROM DUAL;
```

Contoh berikut ini mengasumsikan bahwa perintah dijalankan di akun A, tetapi akun B memerlukan kontrol penuh atas isi bucket. Perintah `rdsadmin_s3_tasks.upload_to_s3` mentransfer semua file dalam direktori *DATA_PUMP_DIR* ke bucket bernama *s3bucket0wnedByAccountB*. Kontrol akses diatur ke `FULL_CONTROL` agar akun B dapat mengakses file di bucket. Tingkat kompresi GZIP adalah *6*, yang menyeimbangkan kecepatan dan ukuran file.

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
      p_bucket_name               =>  's3bucket0wnedByAccountB',
      p_prefix                    =>  '',
      p_s3_prefix                 =>  '',
      p_directory_name            =>  'DATA_PUMP_DIR',
      p_bucket_owner_full_control =>  'FULL_CONTROL',
      p_compression_level         =>  6)
```

```
   AS TASK_ID FROM DUAL;
```

Di setiap contoh, pernyataan `SELECT` mengembalikan ID tugas dalam jenis data `VARCHAR2`.

Anda dapat melihat hasilnya dengan menampilkan file output tugas.

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-task-id.log'));
```

Ganti *`task-id`* dengan ID tugas yang dikembalikan oleh prosedur.

> **Note**
>
> Tugas dijalankan secara asinkron.

Mengunduh file dari bucket Amazon S3 ke instans DB Oracle

Untuk mengunduh file dari bucket Amazon S3 ke instans RDS for Oracle, gunakan prosedur `rdsadmin.rdsadmin_s3_tasks.download_from_s3` Amazon RDS.

Prosedur `download_from_s3` memiliki parameter berikut.

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
`p_bucket_name`	VARCHAI	–	Wajib	Nama bucket Amazon S3 asal unduhan file.
`p_directory_name`	VARCHAI	–	Wajib	Nama objek direktori Oracle untuk menyimpan unduhan file. Direktori dapat berupa objek direktori yang dibuat pengguna atau direktori Data Pump, seperti `DATA_PUMP_DIR` .
`p_error_on_zero_downloads`	VARCHAI	SALAH	Opsional	Tanda yang menentukan apakah tugas akan memunculkan kesalahan jika tidak ada objek di bucket Amazon S3 yang cocok dengan prefiks. Jika

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
				parameter ini tidak diatur atau diatur ke SALAH (default), tugas akan mencetak pesan bahwa objek tidak ditemukan, tetapi tidak memunculkan pengecualian atau gagal. Jika parameter ini BENAR, tugas akan memunculkan pengecualian dan gagal. Contoh spesifikasi prefiks yang dapat menggagalkan uji kecocokan adalah spasi pada prefiks, seperti pada `' import/test9.log'` , dan ketidakcocokan huruf besar/kecil, seperti pada `test9.log` dan `test9.LOG` .

Nama parameter	Tipe data	Default	Diperlukan	Deskripsi
p_s3_prefix	VARCHAI	–	Wajib	Prefiks nama file yang harus sama dengan nama file yang akan diunduh. Prefiks kosong akan mengunduh semua file tingkat atas dalam bucket Amazon S3 yang ditentukan, tetapi tidak akan mengunduh file di folder di dalam bucket. Prosedur ini mengunduh objek Amazon S3 hanya dari folder tingkat pertama yang cocok dengan prefiks. Struktur direktori bersarang yang cocok dengan prefiks yang ditentukan tidak akan diunduh. Sebagai contoh, misal bucket Amazon S3 memiliki struktur folder `folder_1/folder_2/folder_3` . Prefiks yang Anda gunakan adalah `'folder_1/folder_2/'` . Dalam hal ini, hanya file di dalam `folder_2` yang akan diunduh, bukan file di `folder_1` atau `folder_3`. Sebaliknya, jika Anda menggunakan prefiks `'folder_1/folder_2'` , semua file di `folder_1` yang cocok dengan prefiks `'folder_2'` akan diunduh, dan file di `folder_2` tidak akan diunduh satu pun.
p_decompression_format	VARCHAI	–	Opsional	Format dekompresi. Nilai yang valid untuk tanpa dekompresi adalah `NONE`, dan `GZIP` untuk dekompresi.

Nilai yang dikembalikan untuk prosedur `rdsadmin.rdsadmin_s3_tasks.download_from_s3` adalah ID tugas.

Contoh berikut mengunduh semua file di bucket Amazon S3 bernama *`mys3bucket`* ke direktori *`DATA_PUMP_DIR`*. File tidak dikompresi, jadi tidak ada dekompresi yang diterapkan.

```
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
      p_bucket_name    =>  'mys3bucket',
      p_directory_name =>  'DATA_PUMP_DIR')
   AS TASK_ID FROM DUAL;
```

Contoh berikut mengunduh semua file dengan prefiks *`db`* di bucket Amazon S3 bernama *`mys3bucket`* ke direktori *`DATA_PUMP_DIR`*. File dikompresi dengan GZIP, sehingga dekompresi diterapkan. Parameter `p_error_on_zero_downloads` mengaktifkan pemeriksaan kesalahan prefiks, jadi jika prefiks tidak cocok dengan file di dalam bucket, tugas akan memunculkan pengecualian dan gagal.

```
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
      p_bucket_name               =>  'mys3bucket',
      p_s3_prefix                 =>  'db',
      p_directory_name            =>  'DATA_PUMP_DIR',
      p_decompression_format      =>  'GZIP',
      p_error_on_zero_downloads   =>  'TRUE')
   AS TASK_ID FROM DUAL;
```

Contoh berikut mengunduh semua file dalam folder *`myfolder/`* di bucket Amazon S3 bernama *`mys3bucket`* ke direktori *`DATA_PUMP_DIR`*. Gunakan parameter `p_s3_prefix` untuk menentukan folder Amazon S3. File yang diunggah akan dikompresi dengan GZIP, tetapi tidak didekompresi selama pengunduhan.

```
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
      p_bucket_name          =>  'mys3bucket',
      p_s3_prefix            =>  'myfolder/',
      p_directory_name       =>  'DATA_PUMP_DIR',
      p_decompression_format =>  'NONE')
   AS TASK_ID FROM DUAL;
```

Contoh berikut mengunduh file *`mydumpfile.dmp`* di bucket Amazon S3 bernama *`mys3bucket`* ke direktori *`DATA_PUMP_DIR`*. Tidak ada dekompresi yang diterapkan.

```
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
      p_bucket_name    =>  'mys3bucket',
      p_s3_prefix      =>  'mydumpfile.dmp',
      p_directory_name =>  'DATA_PUMP_DIR')
   AS TASK_ID FROM DUAL;
```

Di setiap contoh, pernyataan SELECT mengembalikan ID tugas dalam jenis data VARCHAR2.

Anda dapat melihat hasilnya dengan menampilkan file output tugas.

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-task-id.log'));
```

Ganti *task-id* dengan ID tugas yang dikembalikan oleh prosedur.

> **Note**
>
> Tugas dijalankan secara asinkron.
> Anda dapat menggunakan prosedur UTL_FILE.FREMOVE Oracle untuk menghapus file dari direktori. Untuk informasi selengkapnya, lihat Prosedur FREMOVE dalam dokumentasi Oracle.

Memantau status transfer file

Tugas transfer file menerbitkan peristiwa Amazon RDS saat dimulai dan selesai. Pesan peristiwa berisi ID tugas untuk transfer file. Untuk informasi tentang cara melihat peristiwa, lihat Melihat peristiwa Amazon RDS.

Anda dapat melihat status tugas yang sedang berlangsung di file bdump. File bdump terletak di direktori /rdsdbdata/log/trace. Setiap nama file bdump memiliki format berikut.

```
dbtask-task-id.log
```

Ganti *task-id* dengan ID tugas yang ingin Anda pantau.

> **Note**
>
> Tugas dijalankan secara asinkron.

Anda dapat menggunakan prosedur tersimpan `rdsadmin.rds_file_util.read_text_file` untuk melihat isi file bdump. Misalnya, kueri berikut mengembalikan isi file bdump *`dbtask-1234567890123-1234.log`*.

```
SELECT text FROM
 table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-1234567890123-1234.log'));
```

Contoh berikut menunjukkan file log untuk transfer yang gagal.

```
TASK_ID









--------------------------------------------------------------------------------------
1234567890123-1234


TEXT









--------------------------------------------------------------------------------------
2023-04-17 18:21:33.993 UTC [INFO ] File #1: Uploading the file /rdsdbdata/datapump/
A123B4CDEF567890G1234567890H1234/sample.dmp to Amazon S3 with bucket name mys3bucket
 and key sample.dmp.
2023-04-17 18:21:34.188 UTC [ERROR] RDS doesn't have permission to write to Amazon S3
 bucket name mys3bucket and key sample.dmp.
2023-04-17 18:21:34.189 UTC [INFO ] The task failed.
```

Pemecahan masalah integrasi Amazon S3

Untuk mendapatkan tip pemecahan masalah, lihat artikel AWS re:Post Bagaimana cara memecahkan masalah yang muncul saat mengintegrasikan Amazon RDS for Oracle dengan Amazon S3?.

Menghapus opsi integrasi Amazon S3

Anda dapat menghapus opsi integrasi Amazon S3 dari instans DB.

Untuk menghapus opsi integrasi Amazon S3 dari instans DB, lakukan salah satu hal berikut:

- Untuk menghapus opsi integrasi Amazon S3 dari beberapa instans DB, hapus opsi `S3_INTEGRATION` dari grup opsi tempat instans DB berada. Perubahan ini memengaruhi semua instans DB yang menggunakan grup opsi tersebut. Untuk informasi selengkapnya, lihat Menghapus opsi dari grup opsi.
- Untuk menghapus opsi integrasi Amazon S3 dari satu instans DB, modifikasi instans DB dan tentukan grup opsi lain yang tidak menyertakan opsi `S3_INTEGRATION`. Anda dapat menentukan grup opsi default (kosong) atau grup opsi kustom lain. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Oracle Application Express (APEX)

Amazon RDS mendukung Oracle Application Express (APEX) melalui penggunaan opsi `APEX` dan `APEX-DEV`. Anda dapat melakukan deployment Oracle APEX sebagai lingkungan run-time atau sebagai lingkungan pengembangan penuh untuk aplikasi berbasis web. Dengan menggunakan Oracle APEX, Anda dapat membangun aplikasi sepenuhnya di dalam browser web. Untuk informasi lebih lanjut, lihat Oracle Application Express di dokumentasi Oracle.

Topik

Komponen APEX

Oracle APEX terdiri dari komponen utama berikut:

- Repositori yang menyimpan metadata untuk aplikasi dan komponen APEX. Repositori terdiri dari tabel, indeks, dan objek lain yang diinstal di instans DB Amazon RDS Anda.
- Pendengar yang mengelola komunikasi HTTP dengan klien Oracle APEX. Pendengar berada di host terpisah seperti instans Amazon EC2, server on-premise di perusahaan, atau komputer desktop Anda. Pendengar menerima koneksi masuk dari browser web, meneruskannya ke instans DB Amazon RDS untuk diproses, lalu mengirimkan hasil dari repositori kembali ke browser. RDS for Oracle mendukung jenis pendengar berikut ini:
  - Untuk APEX versi 5.0 dan yang lebih baru, gunakan Oracle Rest Data Services (ORDS) versi 19.1 dan yang lebih baru. Sebaiknya gunakan Oracle APEX dan ORDS versi terbaru yang didukung. Dokumentasi ini menjelaskan versi lama hanya untuk kompatibilitas mundur.
  - Untuk APEX versi 4.1.1, Anda dapat menggunakan Oracle APEX Listener versi 1.1.4.

- Anda dapat menggunakan Oracle HTTP Server dan pendengar `mod_plsql`.

> **Note**
>
> Amazon RDS tidak mendukung server Oracle XML DB HTTP dengan gateway PL/SQL tersemat. Anda tidak dapat menggunakan ini sebagai pendengar untuk APEX. Secara umum, Oracle merekomendasikan untuk tidak menggunakan gateway PL/SQL tersemat untuk aplikasi yang berjalan di internet.

Untuk informasi lebih lanjut tentang jenis pendengar ini, lihat About choosing a web listener di dokumentasi Oracle.

Saat Anda menambahkan opsi Amazon RDS APEX ke instans DB RDS for Oracle, Amazon RDS hanya menginstal repositori Oracle APEX. Instal listener Anda di host terpisah.

Persyaratan versi APEX

Opsi APEX menggunakan penyimpanan pada kelas instans DB untuk instans DB Anda. Berikut adalah versi yang didukung dan perkiraan persyaratan penyimpanan untuk Oracle APEX.

Versi APEX	Persyaratan penyimpanan	Versi Basis Data Oracle yang didukung	Catatan
Oracle APEX versi 23.1.v1	106 MiB	19c dan non-CDB yang lebih tinggi	Versi ini mencakup patch 35283657: PAKETAN PSE UNTUK APEX 23.1 (PSES DI ATAS 23.1.0), PATCH_VERSION 2.
Oracle APEX versi 22.2.v1	106 MiB	Semua non-CDB	Versi ini mencakup patch 34628174: PAKETAN PSE UNTUK APEX 22.2 (PSES DI ATAS 22.2.0), PATCH_VERSION 4.
Oracle APEX versi 22.1.v1	124 MiB	Semua non-CDB	Versi ini mencakup patch 34020981: PAKETAN PSE UNTUK APEX 22.1

Versi APEX	Persyaratan penyimpanan	Versi Basis Data Oracle yang didukung	Catatan
			(PSES DI ATAS 22.1.0), PATCH_VERSION 6.
Oracle APEX versi 21.2.v1	125 MiB	Semua	Versi ini mencakup patch 33420059: PAKETAN PSE UNTUK APEX 21.2 (PSES DI ATAS 21.2.0), PATCH_VERSION 8.
Oracle APEX versi 21.1.v1	125 MiB	Semua	Versi ini mencakup patch 32598392: PAKETAN PSE UNTUK APEX 21.1, PATCH_VERSION 3.
Oracle APEX versi 20.2.v1	148 MiB	Semua kecuali 21c	Versi ini mencakup patch 32006852: PAKETAN PSE UNTUK APEX 20.2, PATCH_VERSION 2020.11.12. Anda dapat melihat nomor dan tanggal patch dengan menjalankan kueri berikut: `SELECT PATCH_VERSION, PATCH_NUMBER FROM APEX_PATCHES;`
Oracle APEX versi 20.1.v1	173 MiB	Semua kecuali 21c	Versi ini mencakup patch 30990551: PAKETAN PSE UNTUK APEX 20.1, PATCH_VERSION 2020.07.15.
Oracle APEX versi 19.2.v1	149 MiB	Semua kecuali 21c	
Oracle APEX versi 19.1.v1	148 MiB	Semua kecuali 21c	
Oracle APEX versi 18.2.v1	146 MiB	12.1 dan 12.2 saja	

Versi APEX	Persyaratan penyimpanan	Versi Basis Data Oracle yang didukung	Catatan
Oracle APEX versi 18.1.v1	145 MiB	12.1 dan 12.2 saja	
Oracle APEX versi 5.1.4.v1	220 MiB	12.1 dan 12.2 saja	
Oracle APEX versi 5.1.2.v1	150 MiB	12.1 dan 12.2 saja	
Oracle APEX versi 5.0.4.v1	140 MiB	12.1 dan 12.2 saja	
Oracle APEX versi 4.2.6.v1	160 MiB	12.1 saja	

Untuk file.zip APEX yang dapat diunduh, lihat Oracle APEX Prior Release Archives di situs web Oracle.

persyaratan dan batasan Oracle APEX dan ORDS

Perhatikan persyaratan berikut untuk APEX dan ORDS:

- Anda harus menggunakan Lingkungan Waktu Proses Java (JRE).
- Instalasi klien Oracle Anda harus mencakup komponen berikut:
  - SQL*Plus atau SQL Developer untuk tugas administrasi
  - Oracle Net Services untuk mengonfigurasi koneksi ke instans DB RDS for Oracle

Perhatikan batasan berikut untuk APEX dan ORDS:

- Anda tidak dapat menggunakan CBD RDS for Oracle dengan ORDS v22 dan versi yang lebih baru. Sebagai solusinya, Anda dapat menggunakan versi ORDS yang lebih lama atau menggunakan Basis Data Oracle 19c non-CDB.

Menambahkan opsi APEX dan APEX-DEV

Untuk menambahkan opsi `APEX-DEV` dan `APEX` ke instans DB, lakukan hal berikut:

1. Buat grup opsi baru, atau salin atau ubah grup opsi yang ada.
2. Tambahkan opsi `APEX` dan `APEX-DEV` opsi ke grup opsi.
3. Kaitkan grup opsi dengan instans DB.

Saat Anda menambahkan opsi Amazon RDS APEX, penonaktifan singkat akan terjadi saat instans DB Anda dimulai ulang secara otomatis.

> **Note**
>
> `APEX_MAIL` tersedia saat opsi `APEX` diinstal. Hak istimewa eksekusi untuk paket `APEX_MAIL` diberikan ke `PUBLIC` sehingga Anda tidak memerlukan akun administratif APEX untuk menggunakannya.

Untuk menambahkan opsi APEX ke instans DB

1. Tentukan grup opsi yang ingin Anda gunakan. Anda dapat membuat grup opsi baru atau menggunakan grup opsi yang ada. Jika Anda ingin menggunakan grup opsi yang ada, lanjutkan ke langkah berikutnya. Jika tidak, buat grup opsi DB kustom dengan pengaturan berikut:

   a. Untuk Mesin, pilih edisi Oracle yang ingin Anda gunakan. Opsi APEX didukung di semua edisi.

   b. Untuk Versi mesin utama, pilih versi instans DB Anda.

   Untuk informasi selengkapnya, lihat Membuat grup opsi.

2. Tambahkan opsi ke grup opsi. Jika Anda ingin melakukan deployment lingkungan run-time Oracle APEX saja, hanya tambahkan opsi `APEX`. Jika Anda ingin menerapkan lingkungan pengembangan penuh, tambahkan kedua opsi `APEX` dan `APEX-DEV`. Untuk Basis Data Oracle 12c, tambahkan opsi APEX dan APEX-DEV.

   Untuk Versi, pilih versi `APEX` yang ingin Anda gunakan. Jika Anda tidak memilih versi, versi 4.2.6.v1 adalah default untuk Basis Data Oracle 12c.

> **Important**
>
> Jika Anda menambahkan opsi APEX ke grup opsi yang sudah ada yang sudah terpasang ke satu instans DB atau lebih, penonaktifan singkat akan terjadi. Selama penonaktifan ini, semua instans DB secara otomatis dimulai ulang.

Untuk informasi selengkapnya tentang cara menambahkan opsi, lihat Menambahkan opsi ke grup opsi.

3. Terapkan grup opsi ke instans DB baru atau yang sudah ada:

   - Untuk instans DB baru, Anda menerapkan grup opsi saat Anda meluncurkan instans. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.
   - Untuk instans DB yang ada, Anda menerapkan grup opsi dengan memodifikasi instans dan melampirkan grup opsi baru. Ketika Anda menambahkan opsi APEX ke instans DB yang ada, penonaktifan singkat akan terjadi saat instans DB Anda dimulai ulang secara otomatis. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Membuka kunci akun pengguna publik

Setelah opsi Amazon RDS APEX diinstal, pastikan untuk melakukan hal berikut:

1. Ubah kata sandi untuk akun pengguna publik APEX.
2. Buka kunci akun.

Anda dapat melakukannya dengan menggunakan utilitas baris perintah Oracle SQL*Plus. Hubungkan ke instans DB Anda sebagai pengguna master dan berikan perintah berikut. Ganti `new_password` dengan kata sandi pilihan Anda.

```
ALTER USER APEX_PUBLIC_USER IDENTIFIED BY new_password;
ALTER USER APEX_PUBLIC_USER ACCOUNT UNLOCK;
```

Mengonfigurasi layanan RESTful untuk Oracle APEX

Untuk mengkonfigurasi layanan RESTful di APEX (tidak diperlukan untuk APEX 4.1.1.V1), gunakan SQL*Plus untuk menghubungkan ke instans DB sebagai pengguna master. Setelah Anda melakukan

tindakan ini, jalankan prosedur tersimpan `rdsadmin.rdsadmin_run_apex_rest_config`. Ketika Anda menjalankan prosedur tersimpan, Anda memberikan kata sandi untuk pengguna berikut ini:

- `APEX_LISTENER`
- `APEX_REST_PUBLIC_USER`

Prosedur tersimpan menjalankan skrip `apex_rest_config.sql`, yang membuat akun basis data baru untuk pengguna ini.

> **Note**
>
> Konfigurasi tidak diperlukan untuk Oracle APEX versi 4.1.1.v1. Untuk versi Oracle APEX ini saja, Anda tidak perlu menjalankan prosedur tersimpan.

Perintah berikut menjalankan prosedur tersimpan.

```
EXEC rdsadmin.rdsadmin_run_apex_rest_config('apex_listener_password',
 'apex_rest_public_user_password');
```

Menyiapkan ORDS for Oracle APEX

Anda sekarang siap untuk menginstal dan mengkonfigurasi Oracle Rest Data Services (ORDS) untuk digunakan dengan Oracle APEX. Untuk APEX versi 5.0 dan yang lebih baru, gunakan Oracle Rest Data Services (ORDS) versi 19.1 dan yang lebih baru.

Instal pendengar di host terpisah seperti instans Amazon EC2, server on-premise di perusahaan, atau komputer desktop Anda. Untuk contoh di bagian ini, kami berasumsi bahwa nama host Anda adalah `myapexhost.example.com`, dan bahwa host Anda menjalankan Linux.

Persiapan untuk menginstal ORDS

Sebelum dapat menginstal ORDS, Anda perlu membuat pengguna OS non-hak istimewa, lalu mengunduh dan mengekstrak file instalasi APEX.

Untuk mempersiapkan instalasi ORDS

1. Masuk ke`myapexhost.example.com` sebagai`root` .

2. Buat pengguna OS non-hak istimewa untuk memiliki instalasi pendengar. Perintah berikut membuat pengguna baru bernama apexuser.

```
useradd -d /home/apexuser apexuser
```

Perintah berikut memberikan kata sandi untuk pengguna baru.

```
passwd apexuser;
```

3. Masuk ke `myapexhost.example.com` sebagai `apexuser`, dan unduh file instalasi APEX dari Oracle ke direktori `/home/apexuser`:
   - http://www.oracle.com/technetwork/developer-tools/apex/downloads/index.html
   - Oracle application Express prior release archives
4. Ekstrak file di direktori `/home/apexuser`.

```
unzip apex_<version>.zip
```

Setelah Anda mengekstrak file, ada file`apex` direktori di`/home/apexuser` direktori.

5. Saat Anda masih masuk ke `myapexhost.example.com` sebagai `apexuser`, unduh file Oracle REST Data Services dari Oracle ke direktori `/home/apexuser`: http://www.oracle.com/technetwork/developer-tools/apex-listener/downloads/index.html.

Menginstal dan mengonfigurasi ORDS

Sebelum dapat menggunakan APEX, Anda perlu mengunduh file ords.war, menggunakan Java untuk menginstal ORDS, dan kemudian memulai pendengar.

Untuk menginstal dan mengonfigurasi ORDS untuk digunakan dengan Oracle APEX

1. Buat direktori baru berdasarkan ORDS, lalu ekstrak file pendengar.

```
mkdir /home/apexuser/ORDS
cd /home/apexuser/ORDS
```

2. Unduh file `ords.version.number.zip` dari layanan data Oracle REST.
3. Ekstrak file ke dalam direktori `/home/apexuser/ORDS`.

4. Jika Anda menginstal ORDS dalam basis data multi-penghuni, tambahkan baris berikut ke file `/home/apexuser/ORDS/params/ords_params.properties`:

```
pdb.disable.lockdown=false
```

5. Berikan pengguna master hak istimewa yang diperlukan untuk menginstal ORDS.

   Setelah opsi Amazon RDS APEX diinstal, berikan pengguna master hak istimewa yang diperlukan untuk menginstal skema ORDS. Anda dapat melakukannya dengan menghubungkan ke basis data dan menjalankan perintah berikut. Ganti *MASTER_USER* dengan nama pengguna master dengan huruf besar.

   > **Important**
   >
   > Jika Anda memasukkan nama pengguna, gunakan huruf besar kecuali Anda membuat pengguna dengan pengidentifikasi peka huruf besar/kecil. Misalnya, jika Anda menjalankan `CREATE USER myuser` atau `CREATE USER MYUSER`, kamus data akan menyimpan `MYUSER`. Namun, jika Anda menggunakan tanda kutip ganda di `CREATE USER "MyUser"`, kamus data akan menyimpan `MyUser`. Untuk informasi lebih lanjut, lihat Memberikan hak istimewa SELECT atau EXECUTE pada objek SYS.

```
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_OBJECTS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_ROLE_PRIVS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_TAB_COLUMNS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_CONS_COLUMNS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_CONSTRAINTS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_OBJECTS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_PROCEDURES', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_TAB_COLUMNS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_TABLES', 'MASTER_USER',
 'SELECT', true);
```

```
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_VIEWS', 'MASTER_USER', 'SELECT',
 true);
exec rdsadmin.rdsadmin_util.grant_sys_object('WPIUTL', 'MASTER_USER', 'EXECUTE',
 true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_SESSION', 'MASTER_USER',
 'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_UTILITY', 'MASTER_USER',
 'EXECUTE', true);
```

> **Note**
>
> Perintah ini berlaku untuk ORDS versi 19.1 dan yang lebih baru.

6. Instal skema ORDS menggunakan file ords.war yang diunduh.

```
java -jar ords.war install advanced
```

Program ini meminta Anda untuk memberikan informasi berikut. Nilai default-nya ada dalam tanda kurung. Untuk informasi lebih lanjut, lihat Introduction to Oracle REST data services di dokumentasi Oracle.

- Masukkan lokasi untuk menyimpan data konfigurasi:

  Masukkan */home/apexuser/ORDS*. Ini adalah lokasi file konfigurasi ORDS.
- Tentukan tipe koneksi basis data yang akan digunakan. Masukkan nomor untuk [1] Basic [2] TNS [3] Custom URL [1]:

  Pilih tipe koneksi yang diinginkan.
- Masukkan nama server basis data [localhost]: *DB_instance_endpoint*

  Pilih default atau masukkan nilai yang benar.
- Masukkan port pendengar basis data [1521]: *DB_instance_port*

  Pilih default atau masukkan nilai yang benar.
- Masukkan 1 untuk menentukan nama layanan basis data, atau 2 untuk menentukan SID basis data [1]:

  Pilih 2 untuk menentukan SID basis data.
- SID basis data [xe]

Pilih default atau masukkan nilai yang benar.

- Masukkan 1 jika Anda ingin memverifikasi/menginstal skema Oracle REST Data Services atau 2 untuk melewati langkah ini [1]:

  Pilih 1. Langkah ini membuat pengguna proxy Oracle REST Data Services bernama ORDS_PUBLIC_USER.

- Masukkan kata sandi basis data untuk ORDS_PUBLIC_USER:

  Masukkan kata sandi, lalu konfirmasi.

- Perlu login dengan hak administrator untuk memverifikasi skema Oracle REST Data Services.

  Masukkan nama pengguna administrator: *master_user*

  Masukkan kata sandi basis data untuk *master_user*: *master_user_password*

  Konfirmasi kata sandi: *master_user_password*

  > **Note**
  >
  > Tentukan kata sandi yang berbeda dengan contoh yang ditampilkan di sini sebagai praktik terbaik keamanan.

- Masukkan ruang tabel default untuk ORDS_METADATA [SYSAUX].

  Masukkan ruang tabel sementara untuk ORDS_METADATA [TEMP].

  Masukkan ruang tabel default untuk ORDS_PUBLIC_USER [USERS].

  Masukkan ruang tabel sementara untuk ORDS_PUBLIC_USER [TEMP].

- Masukkan 1 jika Anda ingin menggunakan PL/SQL Gateway atau 2 untuk melewati langkah ini. Jika Anda menggunakan Oracle Application Express atau bermigrasi dari mod_plsql, Anda harus memasukkan 1 [1].

  Pilih default.

- Masukkan nama pengguna basis data PL/SQL Gateway [APEX_PUBLIC_USER]

  Pilih default.

- Masukkan kata sandi basis data untuk APEX_PUBLIC_USER:

Masukkan kata sandi, lalu konfirmasi.

- Masukkan 1 untuk menentukan kata sandi untuk pengguna basis data Application Express RESTful Services (APEX_LISTENER, APEX_REST_PUBLIC_USER) atau 2 untuk melewati langkah ini [1]:

  Pilih 2 untuk APEX 4.1.1.V1; pilih 1 untuk semua versi APEX lainnya.

- [Tidak diperlukan untuk APEX 4.1.1.v1] Kata sandi basis data untuk APEX_LISTENER

  Masukkan kata sandi (jika diperlukan), lalu konfirmasi.

- [Tidak diperlukan untuk APEX 4.1.1.v1] Kata sandi basis data untuk APEX_REST_PUBLIC_USER

  Masukkan kata sandi (jika diperlukan), lalu konfirmasi.

- Masukkan nomor untuk memilih fitur yang akan diaktifkan:

  Masukkan 1 untuk mengaktifkan semua fitur: SQL Developer Web, REST Enabled SQL, dan Database API.

- Masukkan 1 jika Anda ingin memulai dalam mode mandiri atau 2 untuk keluar [1]:

  Masukkan 1.

- Masukkan lokasi sumber daya statis APEX:

  Jika Anda mengekstrak file instalasi APEX ke `/home/apexuser`, masukkan `/home/apexuser/apex/images`. Jika tidak, masukkan *`unzip_path`*`/apex/images`, dengan *`unzip_path`* adalah direktori tempat Anda mengekstrak file.

- Masukkan 1 jika menggunakan HTTP atau 2 jika menggunakan HTTPS [1]:

  Jika Anda memasukkan 1, tentukan port HTTP. Jika Anda memasukkan 2, tentukan port HTTPS dan nama host SSL. Opsi HTTPS meminta Anda untuk menentukan bagaimana Anda akan memberikan sertifikat:

  - Memasukkan 1 untuk menggunakan sertifikat yang ditandatangani sendiri.
  - Memasukkan 2 untuk memberikan sertifikat Anda sendiri. Jika Anda memasukkan 2, tentukan jalur untuk sertifikat SSL dan jalur untuk kunci privat sertifikat SSL.

7. Tetapkan kata sandi untuk pengguna `admin` APEX. Untuk melakukannya, gunakan SQL*Plus untuk menghubungkan ke instans DB Anda sebagai pengguna master, lalu jalankan perintah berikut.

```
EXEC rdsadmin.rdsadmin_util.grant_apex_admin_role;
grant APEX_ADMINISTRATOR_ROLE to master;
@/home/apexuser/apex/apxchpwd.sql
```

Ganti *master* dengan nama pengguna master Anda. Saat diminta oleh skrip `apxchpwd.sql`, masukkan kata sandi `admin` yang baru.

8. Mulai pendengar ORDS. Jalankan kode berikut.

```
java -jar ords.war
```

Pertama kali Anda memulai ORDS, Anda diminta untuk memberikan lokasi sumber daya statis APEX. Folder gambar ini terletak di direktori `/apex/images` di direktori instalasi untuk APEX.

9. Kembali ke jendela administrasi APEX di browser Anda dan pilih Administrasi. Selanjutnya, pilih Application Express Internal Administration. Saat Anda diminta untuk memberikan kredensial, masukkan informasi berikut ini:

   - Nama pengguna – `admin`
   - Kata sandi - kata sandi yang Anda tetapkan menggunakan skrip `apxchpwd.sql`

   Pilih Masuk, lalu tetapkan kata sandi baru untuk pengguna `admin`.

Pendengar Anda sekarang siap digunakan.

Menyiapkan pendengar Oracle APEX

> **Note**
>
> Pendengar Oracle APEX tidak digunakan lagi.

Amazon RDS for Oracle terus mendukung APEX versi 4.1.1 dan Oracle APEX Listener versi 1.1.4. Sebaiknya gunakan versi terbaru Oracle APEX dan ORDS yang didukung.

Instal Pendengar Oracle APEX pada host terpisah seperti instans Amazon EC2, server on-premise di perusahaan, atau komputer desktop Anda. Kami berasumsi bahwa nama host Anda adalah `myapexhost.example.com`, dan bahwa host Anda menjalankan Linux.

Bersiap untuk menginstal pendengar APEX Oracle

Sebelum dapat menginstal Oracle APEX Listener, Anda perlu membuat pengguna OS non-hak istimewa, lalu mengunduh dan mengekstrak file instalasi APEX.

Untuk mempersiapkan instalasi pendengar Oracle APEX

1. Masuk ke `myapexhost.example.com` sebagai `root`.
2. Buat pengguna OS non-hak istimewa untuk memiliki instalasi pendengar. Perintah berikut membuat pengguna baru bernama apexuser.

   ```
   useradd -d /home/apexuser apexuser
   ```

   Perintah berikut memberikan kata sandi untuk pengguna baru.

   ```
   passwd apexuser;
   ```
3. Masuk ke `myapexhost.example.com` sebagai `apexuser`, dan unduh file instalasi APEX dari Oracle ke direktori `/home/apexuser`:
   - http://www.oracle.com/technetwork/developer-tools/apex/downloads/index.html
   - Oracle application Express prior release archives
4. Ekstrak file di direktori `/home/apexuser`.

   ```
   unzip apex_<version>.zip
   ```

   Setelah Anda mengekstrak file, ada direktori `apex` di direktori `/home/apexuser`.
5. Saat Anda masih masuk ke `myapexhost.example.com` sebagai `apexuser`, unduh file Oracle APEX Listener dari Oracle ke direktori `/home/apexuser`.

Menginstal dan mengonfigurasi pendengar Oracle APEX

Sebelum dapat menggunakan APEX, Anda perlu mengunduh file apex.war, menggunakan Java untuk menginstal Oracle APEX Listener, lalu memulai pendengar.

Untuk menginstal dan mengonfigurasi pendengar APEX Oracle

1. Buat direktori baru berdasarkan Oracle APEX Listener dan buka file pendengar.

Jalankan kode berikut:

```
mkdir /home/apexuser/apexlistener
cd /home/apexuser/apexlistener
unzip ../apex_listener.version.zip
```

2. Jalankan kode berikut.

```
java -Dapex.home=./apex -Dapex.images=/home/apexuser/apex/images -Dapex.erase -
jar ./apex.war
```

3. Masukkan informasi untuk program yang meminta hal berikut:

   - Nama pengguna Administrator APEX Listerner. Default-nya adalah adminlistener.
   - Kata sandi untuk Administrator APEX Listener.
   - Nama pengguna Pengelola APEX Listener. Default-nya adalah managerlistener.
   - Kata sandi untuk Administrator APEX Listener.

   Program mencetak URL yang Anda perlukan untuk menyelesaikan konfigurasi, sebagai berikut.

```
INFO: Please complete configuration at: http://localhost:8080/apex/
listenerConfigure
Database is not yet configured
```

4. Biarkan Oracle APEX Listener berjalan sehingga Anda dapat menggunakan Oracle Application Express. Setelah Anda menyelesaikan prosedur konfigurasi ini, Anda dapat menjalankan pendengar di latar belakang.
5. Dari browser web Anda, buka URL yang disediakan oleh program APEX Listener. Jendela administrasi Oracle Application Express Listener akan muncul. Masukkan informasi berikut:

   - Nama pengguna – `APEX_PUBLIC_USER`
   - Kata sandi - kata sandi untuk APEX_PUBLIC_USER. Kata sandi ini adalah yang Anda tentukan sebelumnya saat Anda mengonfigurasi repositori APEX. Untuk informasi selengkapnya, lihat Membuka kunci akun pengguna publik.
   - Tipe koneksi - Basic
   - Nama host - titik akhir instans DB Amazon RDS Anda, seperti `mydb.f9rbfa893tft.us-east-1.rds.amazonaws.com`.

- Port - 1521
- SID - nama basis data di instans DB Amazon RDS Anda, seperti `mydb`.

6. Pilih Terapkan. Jendela administrasi APEX muncul.
7. Tetapkan kata sandi untuk pengguna `admin` APEX. Untuk melakukannya, gunakan SQL*Plus untuk menghubungkan ke instans DB Anda sebagai pengguna master, lalu jalankan perintah berikut.

```
EXEC rdsadmin.rdsadmin_util.grant_apex_admin_role;
grant APEX_ADMINISTRATOR_ROLE to master;
@/home/apexuser/apex/apxchpwd.sql
```

   Ganti *`master`* dengan nama pengguna master Anda. Ketika diminta oleh skrip `apxchpwd.sql`, masukkan kata sandi `admin` yang baru.
8. Kembali ke jendela administrasi APEX di browser Anda dan pilih Administrasi. Selanjutnya, pilih Application Express Internal Administration. Saat Anda diminta untuk memberikan kredensial, masukkan informasi berikut ini:

   - Nama pengguna – `admin`
   - Kata sandi - kata sandi yang Anda tetapkan menggunakan skrip `apxchpwd.sql`

   Pilih Masuk, lalu tetapkan kata sandi baru untuk pengguna `admin`.

Pendengar Anda sekarang siap digunakan.

Memutakhirkan versi APEX

> **Important**
>
> Cadangkan instans DB Anda sebelum memutakhirkan APEX. Untuk informasi selengkapnya, lihat Membuat snapshot DB untuk instans DB Single-AZ dan Menguji upgrade DB Oracle.

Untuk memutakhirkan APEX dengan instans DB Anda, lakukan hal berikut:

- Buat grup opsi baru untuk versi pemutakhiran instans DB Anda.

- Tambahkan versi APEX dan APEX-DEV yang dimutakhirkan ke grup opsi baru. Pastikan untuk menyertakan opsi lain yang digunakan instans DB Anda. Untuk informasi lebih lanjut, lihat Pertimbangan grup opsi.
- Saat Anda memutakhirkan instans DB Anda, tentukan grup opsi baru untuk instans DB yang dimutakhirkan.

Setelah memutakhirkan versi APEX, skema APEX untuk versi sebelumnya mungkin masih ada di basis data Anda. Jika Anda tidak membutuhkannya lagi, Anda dapat menghapus skema APEX lama dari basis data Anda setelah pemutakhiran.

Jika Anda memutakhirkan versi APEX dan layanan RESTful tidak dikonfigurasi dalam versi APEX sebelumnya, sebaiknya konfigurasi layanan RESTful. Untuk informasi selengkapnya, lihat Mengonfigurasi layanan RESTful untuk Oracle APEX.

Dalam beberapa kasus, ketika berencana untuk memutakhirkan versi utama instans DB, Anda mungkin menemukan bahwa Anda menggunakan versi APEX yang tidak kompatibel dengan versi basis data target. Dalam kasus ini, Anda dapat memutakhirkan versi APEX Anda sebelum memutakhirkan instans DB Anda. Memutakhirkan APEX terlebih dahulu dapat mengurangi jumlah waktu yang diperlukan untuk memutakhirkan instans DB Anda.

> **Note**
>
> Setelah memutakhirkan APEX, instal dan konfigurasi pendengar untuk digunakan dengan versi yang dimutakhirkan. Untuk petunjuk, lihat Menyiapkan pendengar Oracle APEX .

Menghapus opsi APEX

Anda dapat menghapus opsi Amazon RDS APEX dari instans DB. Untuk menghapus opsi APEX dari instans DB, lakukan salah satu hal berikut:

- Untuk menghapus opsi APEX dari beberapa instans DB, hapus opsi APEX dari grup opsi yang mencakupnya. Perubahan ini memengaruhi semua instans DB yang menggunakan grup opsi tersebut. Saat Anda menghapus opsi APEX dari grup opsi yang dilampirkan ke beberapa instans DB, penonaktifan singkat akan terjadi saat semua instans DB dimulai ulang.

  Untuk informasi lebih lanjut, lihat Menghapus opsi dari grup opsi.

- Untuk menghapus opsi APEX dari satu instans DB, ubah instans DB dan tentukan grup opsi berbeda yang tidak menyertakan opsi APEX. Anda dapat menentukan grup opsi default (kosong) atau grup opsi kustom yang berbeda. Saat Anda menghapus opsi APEX, penonaktifan singkat akan terjadi saat instans DB Anda dimulai ulang secara otomatis.

  Untuk informasi lebih lanjut, lihat Memodifikasi instans DB Amazon RDS.

Saat Anda menghapus opsi APEX dari instans DB, skema APEX akan dihapus dari basis data Anda.

Integrasi Amazon EFS

Amazon Elastic File System (Amazon EFS) menyediakan penyimpanan file nirserver dan sepenuhnya elastis sehingga Anda dapat berbagi data file tanpa perlu menyediakan atau mengelola kapasitas dan performa penyimpanan. Dengan Amazon EFS, Anda dapat membuat sistem file dan memasangnya di VPC melalui protokol NFS versi 4.0 dan 4.1 (NFSv4). Kemudian Anda dapat menggunakan sistem file EFS seperti sistem file sesuai POSIX lainnya. Untuk informasi umum, lihat Apa itu Amazon Elastic File System? dan blog AWS Mengintegrasikan Amazon RDS for Oracle dengan Amazon EFS.

Topik

Ikhtisar integrasi Amazon EFS

Dengan Amazon EFS, Anda dapat mentransfer file antara instans DB RDS for Oracle dan sistem file EFS. Sebagai contoh, Anda dapat menggunakan EFS untuk mendukung kasus penggunaan berikut:

- Membagikan sistem file antara aplikasi dan beberapa server basis data.
- Membuat direktori bersama untuk file terkait migrasi, termasuk file data tablespace yang dapat diangkut. Untuk informasi selengkapnya, lihat Bermigrasi menggunakan tablespace yang dapat dipindahkan Oracle.
- Menyimpan dan membagikan file log pengulangan yang diarsipkan tanpa mengalokasikan ruang penyimpanan tambahan di server.
- Menggunakan utilitas Oracle Database seperti `UTL_FILE` untuk membaca dan menulis file.

Keuntungan integrasi Amazon EFS

Ketika memilih sistem file EFS daripada solusi transfer data alternatif, Anda akan mendapatkan manfaat berikut:

- Anda dapat mentransfer file Oracle Data Pump antara Amazon EFS dan instans DB RDS for Oracle Anda. Anda tidak perlu menyalin file ini secara lokal karena Data Pump mengimpor langsung dari sistem file EFS. Untuk informasi selengkapnya, lihat Mengimpor data ke Oracle di Amazon RDS.
- Migrasi data lebih cepat daripada menggunakan tautan basis data.
- Tidak perlu mengalokasikan ruang penyimpanan file di instans DB RDS for Oracle Anda.
- Sistem file EFS dapat secara otomatis menskalakan penyimpanan tanpa harus Anda sediakan.
- Integrasi Amazon EFS tidak mengenakan biaya minimum atau harga tertentu. Pembayaran dilakukan sesuai penggunaan.

Persyaratan integrasi Amazon EFS

Pastikan Anda memenuhi persyaratan berikut:

- Basis data Anda menjalankan basis data versi 19.0.0.0.ru-2022-07.rur-2022-07.r1 atau yang lebih baru.
- Instans DB dan sistem file EFS Anda berada di Wilayah AWS dan VPC yang sama.
- Atribut `enableDnsSupport` di VPC Anda aktif. Untuk informasi selengkapnya, lihat Atribut DNS untuk VPC Anda dalam Panduan Pengguna Amazon Virtual Private Cloud.
- Sistem file EFS Anda menggunakan kelas penyimpanan Standar atau Standard-IA.
- Untuk dapat menggunakan nama DNS dalam perintah `mount`, pastikan hal berikut telah sesuai:
  - Instans DB penghubung berada di VPC dan dikonfigurasi untuk menggunakan server DNS yang disediakan oleh Amazon. Server DNS kustom tidak didukung.
  - VPC instans penghubung harus mengaktifkan Resolusi DNS dan Nama Host DNS.
  - Instans penghubung harus berada di dalam VPC yang sama dengan sistem file EFS.
- Anda menggunakan solusi non-RDS untuk mencadangkan sistem file EFS. RDS for Oracle tidak mendukung pencadangan otomatis atau snapshot DB manual untuk sistem file EFS. Untuk informasi selengkapnya, lihat Mencadangkan sistem file Amazon EFS Anda.

Mengonfigurasi izin jaringan untuk integrasi RDS for Oracle dengan Amazon EFS

Agar RDS for Oracle dapat diintegrasikan dengan Amazon EFS, pastikan instans DB Anda memiliki akses jaringan ke sistem file EFS. Untuk informasi selengkapnya, lihat Mengontrol akses jaringan ke sistem file Amazon EFS untuk klien NFS dalam Panduan Pengguna Amazon Elastic File System.

Topik

- Mengontrol akses jaringan dengan grup keamanan
- Mengontrol akses jaringan dengan kebijakan sistem file

Mengontrol akses jaringan dengan grup keamanan

Anda dapat mengontrol akses instans DB ke sistem file EFS menggunakan mekanisme keamanan lapisan jaringan seperti grup keamanan VPC. Untuk mengizinkan akses instans DB ke sistem file EFS, pastikan sistem file EFS Anda memenuhi persyaratan berikut:

- Terdapat target pemasangan EFS di setiap Zona Ketersediaan yang digunakan oleh instans DB RDS for Oracle.

  Target pemasangan EFS menyediakan alamat IP untuk titik akhir NFSv4 tempat pemasangan sistem file EFS. Sistem file dipasang menggunakan nama DNS-nya, yang diselesaikan ke alamat IP target pemasangan EFS yang digunakan oleh Zona Ketersediaan instans DB Anda.

  Anda dapat mengonfigurasi agar instans DB di AZ yang berbeda dapat menggunakan sistem file EFS yang sama. Untuk Multi-AZ, diperlukan titik pemasangan untuk setiap AZ dalam deployment Anda. Anda mungkin perlu memindahkan instans DB ke AZ yang berbeda. Karena alasan ini, sebaiknya buat titik pemasangan EFS di setiap AZ dalam VPC Anda. Secara default, saat Anda membuat sistem file EFS baru menggunakan konsol, RDS membuat target pemasangan untuk semua AZ.
- Grup keamanan terpasang pada target pemasangan.
- Grup keamanan memiliki aturan masuk untuk mengizinkan subnet jaringan atau grup keamanan instans DB RDS for Oracle pada TCP/2049 (Tipe NFS).

Untuk informasi selengkapnya, lihat Membuat sistem file Amazon EFS dan Membuat dan mengelola target pemasangan dan grup keamanan EFS dalam Panduan Pengguna Amazon Elastic File System.

Mengontrol akses jaringan dengan kebijakan sistem file

Integrasi Amazon EFS dengan RDS for Oracle menggunakan kebijakan sistem file EFS default (kosong). Kebijakan default tidak menggunakan IAM untuk autentikasi. Sebagai gantinya, kebijakan memberikan akses penuh ke klien anonim yang dapat terhubung ke sistem file menggunakan target pemasangan. Kebijakan default berlaku saat kebijakan sistem file yang dikonfigurasi pengguna tidak berlaku, termasuk saat pembuatan sistem file. Untuk informasi selengkapnya, lihat Kebijakan default sistem file EFS dalam Panduan Pengguna Amazon Elastic File System.

Guna memperkuat akses semua klien ke sistem file EFS Anda, termasuk RDS for Oracle, Anda dapat mengonfigurasi izin IAM. Dalam pendekatan ini, Anda membuat kebijakan sistem file. Untuk informasi selengkapnya, lihat Membuat kebijakan sistem file di Panduan Pengguna Amazon Elastic File System.

Mengonfigurasi izin IAM untuk integrasi RDS for Oracle dengan Amazon EFS

Secara default, fitur integrasi Amazon EFS tidak menggunakan peran IAM: pengaturan `USE_IAM_ROLE` opsi adalah`FALSE`. Untuk mengintegrasikan RDS untuk Oracle dengan Amazon EFS dan peran IAM, instans DB Anda harus memiliki izin IAM untuk mengakses sistem file Amazon EFS.

Topik

- Langkah 1: Buat peran IAM untuk instans DB Anda dan lampirkan kebijakan Anda
- Langkah 2: Buat kebijakan sistem file untuk Amazon EFS Anda
- Langkah 3: Kaitkan peran IAM Anda dengan instans DB RDS for Oracle

Langkah 1: Buat peran IAM untuk instans DB Anda dan lampirkan kebijakan Anda

Pada langkah ini, buat peran untuk instans DB RDS for Oracle agar Amazon RDS dapat mengakses sistem file EFS Anda.

Konsol

Untuk membuat peran IAM agar Amazon RDS dapat mengakses sistem file EFS

1. Buka Konsol Manajemen IAM.
2. Di panel navigasi, pilih Peran.
3. Pilih Buat peran.
4. Untuk Layanan AWS, pilih RDS.

5. Untuk Pilih kasus penggunaan, pilih RDS – Tambahkan Peran ke Basis Data.
6. Pilih Berikutnya.
7. Jangan tambahkan kebijakan izin apa pun. Pilih Berikutnya.
8. Tentukan Nama peran untuk nama peran IAM, misalnya `rds-efs-integration-role`. Anda juga dapat menambahkan nilai Deskripsi opsional.
9. Pilih Buat peran.

AWS CLI

Untuk membatasi izin layanan ke sumber daya tertentu, sebaiknya gunakan kunci konteks kondisi global `aws:SourceArn` dan `aws:SourceAccount` dalam hubungan kepercayaan berbasis sumber daya. Ini adalah perlindungan paling efektif dari masalah confused deputy.

Anda dapat menggunakan kedua kunci konteks kondisi global dan memiliki nilai `aws:SourceArn` yang berisi ID akun. Dalam hal ini, nilai `aws:SourceAccount` dan akun dalam nilai `aws:SourceArn` harus menggunakan ID akun yang sama ketika digunakan dalam pernyataan yang sama.

- Gunakan `aws:SourceArn` jika Anda ingin akses lintas layanan untuk satu sumber daya.
- Gunakan `aws:SourceAccount` jika Anda ingin mengizinkan sumber daya apa pun di akun tersebut dikaitkan dengan penggunaan lintas layanan.

Dalam hubungan kepercayaan, pastikan untuk menggunakan kunci konteks kondisi global `aws:SourceArn` dengan Amazon Resource Name (ARN) penuh pada sumber daya yang mengakses peran.

Perintah AWS CLI berikut membuat peran bernama *`rds-efs-integration-role`* untuk tujuan ini.

Example

Untuk Linux, macOS, atau Unix:

```
aws iam create-role \
   --role-name rds-efs-integration-role \
   --assume-role-policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
```

```
        "Principal": {
           "Service": "rds.amazonaws.com"
         },
        "Action": "sts:AssumeRole",
        "Condition": {
            "StringEquals": {
                "aws:SourceAccount": my_account_ID,
                "aws:SourceArn": "arn:aws:rds:Region:my_account_ID:db:dbname"
            }
        }
      }
    ]
  }'
```

Untuk Windows:

```
aws iam create-role ^
   --role-name rds-efs-integration-role ^
   --assume-role-policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
            "Service": "rds.amazonaws.com"
         },
         "Action": "sts:AssumeRole",
         "Condition": {
             "StringEquals": {
                 "aws:SourceAccount": my_account_ID,
                 "aws:SourceArn": "arn:aws:rds:Region:my_account_ID:db:dbname"
             }
         }
       }
     ]
   }'
```

Untuk informasi selengkapnya, lihat Membuat peran untuk mendelegasikan izin ke pengguna IAM dalam Panduan Pengguna IAM.

Langkah 2: Buat kebijakan sistem file untuk Amazon EFS Anda

Pada langkah ini, buat kebijakan sistem file untuk EFS Anda.

Untuk membuat atau mengedit kebijakan sistem file EFS

1. Buka Konsol Manajemen EFS.
2. Pilih Sistem File.
3. Pada halaman Sistem file, pilih sistem file yang akan diedit atau dibuatkan kebijakan sistem file. Halaman detail sistem file akan terbuka.
4. Pilih tab Kebijakan sistem file.

   Jika kebijakan kosong, kebijakan sistem file EFS default sedang digunakan. Untuk informasi selengkapnya, lihat Kebijakan sistem file EFS default dalam Panduan Pengguna Amazon Elastic File System.
5. Pilih Edit. Halaman Kebijakan sistem file muncul.
6. Di Editor kebijakan, masukkan kebijakan seperti berikut ini, lalu pilih Simpan.

```
{
    "Version": "2012-10-17",
    "Id": "ExamplePolicy01",
    "Statement": [
        {
            "Sid": "ExampleStatement01",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:role/rds-efs-integration-role"
            },
            "Action": [
                "elasticfilesystem:ClientMount",
                "elasticfilesystem:ClientWrite",
                "elasticfilesystem:ClientRootAccess"
            ],
            "Resource": "arn:aws:elasticfilesystem:us-east-1:123456789012:file-
system/fs-1234567890abcdef0"
        }
    ]
}
```

Langkah 3: Kaitkan peran IAM Anda dengan instans DB RDS for Oracle

Pada langkah ini, kaitkan peran IAM Anda dengan instans DB Anda. Perhatikan persyaratan berikut:

- Anda harus memiliki akses ke peran IAM dengan kebijakan izin Amazon EFS yang diperlukan.
- Anda hanya dapat mengaitkan satu peran IAM dengan instans DB RDS for Oracle Anda dalam satu waktu.
- Status instans Anda harus Tersedia.

Untuk informasi selengkapnya, lihat Manajemen identitas dan akses Amazon EFS dalam Panduan Pengguna Amazon Elastic File System.

Konsol

Untuk mengaitkan peran IAM Anda dengan instans DB RDS for Oracle Anda

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Pilih Basis data.
3. Jika instans basis data Anda tidak tersedia, pilih Tindakan lalu Mulai. Saat instans berstatus Dimulai, lanjutkan ke langkah berikutnya.
4. Pilih nama instans DB Oracle untuk menampilkan detailnya.
5. Pada tab Konektivitas & keamanan, gulir ke Kelola peran IAM di bagian bawah halaman.
6. Pilih peran yang akan ditambahkan di bagian Tambahkan peran IAM ke instans ini.
7. Untuk Fitur, pilih EFS_INTEGRATION.
8. Pilih Tambahkan peran.

AWS CLI

Perintah AWS CLI berikut menambahkan peran ke instans DB Oracle bernama *mydbinstance*.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds add-role-to-db-instance \
   --db-instance-identifier mydbinstance \
   --feature-name EFS_INTEGRATION \
   --role-arn your-role-arn
```

Untuk Windows:

```
aws rds add-role-to-db-instance ^
   --db-instance-identifier mydbinstance ^
   --feature-name EFS_INTEGRATION ^
   --role-arn your-role-arn
```

Ganti *your-role-arn* dengan peran ARN yang Anda catat di langkah sebelumnya. `EFS_INTEGRATION` harus ditentukan untuk opsi `--feature-name`.

Menambahkan opsi EFS_INTEGRATION

Untuk mengintegrasikan Amazon RDS for Oracle dengan Amazon EFS, instans DB Anda harus dikaitkan dengan grup opsi yang menyertakan opsi `EFS_INTEGRATION`.

Beberapa instans DB Oracle di dalam grup opsi yang sama memiliki sistem file EFS yang sama. Instans DB yang berbeda dapat mengakses data yang sama, tetapi akses dapat dibagi menggunakan direktori Oracle yang berbeda. Untuk informasi selengkapnya, lihat Mentransfer file antara RDS for Oracle dan sistem file Amazon EFS.

Konsol

Untuk mengonfigurasi grup opsi untuk integrasi Amazon EFS

1. Buat grup opsi baru atau identifikasi grup opsi yang ada yang dapat ditambahi opsi `EFS_INTEGRATION`.

   Untuk informasi tentang cara membuat grup opsi, lihat Membuat grup opsi.
2. Tambahkan opsi `EFS_INTEGRATION` ke grup opsi. Anda perlu menentukan ID sistem file `EFS_ID` dan mengatur bendera `USE_IAM_ROLE`.

   Untuk informasi selengkapnya, lihat Menambahkan opsi ke grup opsi.
3. Kaitkan grup opsi dengan instans DB Anda melalui salah satu cara berikut:

   - Buat instans DB Oracle baru dan kaitkan dengan grup opsi. Untuk informasi tentang pembuatan instans DB, lihat Membuat instans DB Amazon RDS.
   - Modifikasi instans DB Oracle untuk dikaitkan dengan grup opsi. Untuk informasi tentang memodifikasi instans DB Oracle, lihat Memodifikasi instans DB Amazon RDS.

AWS CLI

Untuk mengonfigurasi grup opsi untuk integrasi EFS

1. Buat grup opsi baru atau identifikasi grup opsi yang ada yang dapat ditambahi opsi `EFS_INTEGRATION`.

   Untuk informasi tentang cara membuat grup opsi, lihat Membuat grup opsi.
2. Tambahkan opsi `EFS_INTEGRATION` ke grup opsi.

   Misalnya, perintah AWS CLI berikut menambahkan opsi `EFS_INTEGRATION` ke grup opsi bernama ***myoptiongroup***.

   Example

   Untuk Linux, macOS, atau Unix:

   ```
   aws rds add-option-to-option-group \
      --option-group-name myoptiongroup \
      --options "OptionName=EFS_INTEGRATION,OptionSettings=\
      [{Name=EFS_ID,Value=fs-1234567890abcdef0},{Name=USE_IAM_ROLE,Value=TRUE}]"
   ```

   Untuk Windows:

   ```
   aws rds add-option-to-option-group ^
      --option-group-name myoptiongroup ^
      --options "OptionName=EFS_INTEGRATION,OptionSettings=^
      [{Name=EFS_ID,Value=fs-1234567890abcdef0},{Name=USE_IAM_ROLE,Value=TRUE}]"
   ```
3. Kaitkan grup opsi dengan instans DB Anda melalui salah satu cara berikut:

   - Buat instans DB Oracle baru dan kaitkan dengan grup opsi. Untuk informasi tentang pembuatan instans DB, lihat Membuat instans DB Amazon RDS.
   - Modifikasi instans DB Oracle untuk dikaitkan dengan grup opsi. Untuk informasi tentang modifikasi instans DB Oracle, lihat Memodifikasi instans DB Amazon RDS.

Mengonfigurasi izin sistem file Amazon EFS

Secara default, hanya pengguna root (UID `0`) yang memiliki izin baca, tulis, dan eksekusi untuk sistem file EFS yang baru dibuat. Pengguna lain yang ingin memodifikasi sistem file harus secara

eksplisit mendapatkan akses dari pengguna root. Pengguna untuk instans DB RDS for Oracle berada dalam kategori `others`. Untuk informasi selengkapnya, lihat Bekerja dengan pengguna, grup, dan izin di Tingkat Network File System (NFS) dalam Panduan Pengguna Amazon Elastic File System.

Agar instans DB RDS for Oracle dapat membaca dan menulis file di sistem file EFS, lakukan hal berikut:

- Pasang sistem file EFS secara lokal di Amazon EC2 atau instans on-premise.
- Konfigurasikan izin mendetail.

Misalnya, untuk memberikan izin penulisan pada sistem file EFS kepada pengguna `other`, jalankan `chmod 777` pada direktori ini. Untuk informasi selengkapnya, lihat Contoh kasus penggunaan dan izin sistem file Amazon EFS dalam Panduan Pengguna Amazon Elastic File System.

Mentransfer file antara RDS for Oracle dan sistem file Amazon EFS

Untuk mentransfer file antara instans RDS untuk Oracle dan sistem file Amazon EFS, buat setidaknya satu direktori Oracle dan konfigurasikan izin sistem file EFS untuk mengontrol akses instans DB.

Topik

- Membuat direktori Oracle
- Mentransfer data ke dan dari sistem file EFS: contoh

Membuat direktori Oracle

Untuk membuat direktori Oracle, gunakan prosedur `rdsadmin.rdsadmin_util.create_directory_efs`. Prosedur ini memiliki parameter berikut.

Nama parameter	Tipe data	Defaul	Diperlu n	Deskripsi
`p_directory_name`	VARCHAR	–	Ya	Nama direktori Oracle.
`p_path_on_efs`	VARCHAR	–	Ya	Jalur pada sistem file EFS. Prefiks nama jalur menggunakan pola `/rdsefs-`*`fsid`*`/`, dengan

Nama parameter	Tipe data	Defaul	Diperlu n	Deskripsi
				fsid sebagai placeholder untuk ID sistem file EFS Anda. Misalnya, jika sistem file EFS Anda bernama `fs-1234567890abcdef0` , dan Anda membuat subdirektori pada sistem file ini bernama `mydir`, Anda dapat menentukan nilai berikut: `/rdsefs-fs-1234567890abcdef0/mydir`

Misalnya Anda membuat subdirektori bernama `/datapump1` pada sistem file EFS `fs-1234567890abcdef0`. Contoh berikut membuat direktori Oracle `DATA_PUMP_DIR_EFS` yang mengarah ke direktori `/datapump1` pada sistem file EFS. Nilai jalur sistem file untuk parameter `p_path_on_efs` menggunakan prefiks string `/rdsefs-`.

```
BEGIN
  rdsadmin.rdsadmin_util.create_directory_efs(
    p_directory_name => 'DATA_PUMP_DIR_EFS',
    p_path_on_efs    => '/rdsefs-fs-1234567890abcdef0/datapump1');
END;
/
```

Mentransfer data ke dan dari sistem file EFS: contoh

Contoh berikut menggunakan Oracle Data Pump untuk mengekspor tabel bernama `MY_TABLE` ke file `datapump.dmp`. File ini berada di sistem file EFS.

```
DECLARE
  v_hdnl NUMBER;
BEGIN
  v_hdnl := DBMS_DATAPUMP.OPEN(operation => 'EXPORT', job_mode => 'TABLE',
 job_name=>null);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'datapump.dmp',
    directory => 'DATA_PUMP_DIR_EFS',
```

```
    filetype  => dbms_datapump.ku$_file_type_dump_file);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'datapump-exp.log',
    directory => 'DATA_PUMP_DIR_EFS',
    filetype  => dbms_datapump.ku$_file_type_log_file);
  DBMS_DATAPUMP.METADATA_FILTER(v_hdnl,'NAME_EXPR','IN (''MY_TABLE'')');
  DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/
```

Contoh berikut menggunakan Oracle Data Pump untuk mengimpor tabel bernama `MY_TABLE` dari file `datapump.dmp`. File ini berada di sistem file EFS.

```
DECLARE
  v_hdnl NUMBER;
BEGIN
  v_hdnl := DBMS_DATAPUMP.OPEN(
    operation => 'IMPORT',
    job_mode  => 'TABLE',
    job_name  => null);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'datapump.dmp',
    directory => 'DATA_PUMP_DIR_EFS',
    filetype  => dbms_datapump.ku$_file_type_dump_file );
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'datapump-imp.log',
    directory => 'DATA_PUMP_DIR_EFS',
    filetype  => dbms_datapump.ku$_file_type_log_file);
  DBMS_DATAPUMP.METADATA_FILTER(v_hdnl,'NAME_EXPR','IN (''MY_TABLE'')');
  DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/
```

Untuk informasi selengkapnya, lihat Mengimpor data ke Oracle di Amazon RDS.

Menghapus opsi EFS_INTEGRATION

Untuk menghapus opsi `EFS_INTEGRATION` dari instans DB RDS for Oracle, lakukan salah satu cara berikut:

- Untuk menghapus opsi `EFS_INTEGRATION` dari beberapa instans DB, hapus opsi `EFS_INTEGRATION` dari grup opsi asal instans DB. Perubahan ini memengaruhi semua instans DB yang menggunakan grup opsi tersebut. Untuk informasi selengkapnya, lihat Menghapus opsi dari grup opsi.

- Untuk menghapus opsi `EFS_INTEGRATION` dari satu instans DB, modifikasi instans dan tentukan grup opsi lain yang tidak menyertakan opsi `EFS_INTEGRATION`. Anda dapat menentukan grup opsi default (kosong) atau grup opsi kustom yang berbeda. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Pemecahan masalah integrasi Amazon EFS

Instans DB RDS for Oracle Anda memantau konektivitas ke sistem file Amazon EFS. Saat terdeteksi masalah, pemantau mungkin akan mencoba memperbaiki masalah tersebut dan menerbitkan peristiwa di konsol RDS. Untuk informasi selengkapnya, lihat Melihat peristiwa Amazon RDS.

Gunakan informasi di bagian ini untuk membantu Anda mendiagnosis dan memperbaiki masalah umum saat bekerja menggunakan integrasi Amazon EFS.

Notifikasi	Deskripsi	Tindakan
`The EFS for RDS Oracle instance` *`instance_name`* `isn't available on the primary host. NFS port 2049 of your EFS isn't reachable.`	Instans DB tidak dapat berkomunikasi dengan sistem file EFS.	Pastikan hal-hal berikut: • Sistem file EFS ada. • Grup keamanan yang dilampirkan ke target pemasangan EFS memiliki aturan masuk untuk mengizinkan grup keamanan atau subnet jaringan instans DB RDS for Oracle DB pada TCP/2049 (Tipe NFS).
`The EFS isn't reachable.`	Terjadi kesalahan selama instalasi opsi `EFS_INTEGRATION` .	Pastikan hal-hal berikut: • Sistem file EFS ada.

Notifikasi	Deskripsi	Tindakan
		• Grup keamanan yang dilampirkan ke target pemasangan EFS memiliki aturan masuk untuk mengizinkan grup keamanan atau subnet jaringan instans DB RDS for Oracle DB pada TCP/2049 (Tipe NFS). • Atribut `enableDnsSupport` untuk VPC Anda aktif. • Anda menggunakan server DNS yang disediakan Amazon di VPC Anda. Integrasi Amazon EFS tidak kompatibel dengan DNS DHCP kustom.
`The associated role with your DB instance wasn't found.`	Terjadi kesalahan selama penginstalan opsi `EFS_INTEGRATION` .	Pastikan Anda telah mengaitkan peran IAM dengan instans DB RDS for Oracle.
`The associated role with your DB instance wasn't found.`	Terjadi kesalahan selama penginstalan opsi `EFS_INTEGRATION` . RDS untuk Oracle dipulihkan dari snapshot DB dengan pengaturan `USE_IAM_ROLE` opsi. `TRUE`	Pastikan Anda telah mengaitkan peran IAM dengan instans DB RDS for Oracle.

Notifikasi	Deskripsi	Tindakan
`The associated role with your DB instance wasn't found.`	Terjadi kesalahan selama penginstalan opsi `EFS_INTEGRATION` . RDS untuk Oracle dibuat dari all-in-one CloudFormation template dengan pengaturan `USE_IAM_ROLE` opsi. `TRUE`	Sebagai solusinya, selesaikan langkah-langkah berikut: 1. Buat instance DB dengan peran IAM dan grup opsi default. 2. Pada pembaruan tumpukan berikutnya, tambahkan grup opsi khusus dengan `EFS_INTEGRATION` opsi.
`PLS-00302: component 'CREATE_DIRECTORY_EFS' must be declared`	Kesalahan ini dapat terjadi ketika Anda menggunakan versi RDS for Oracle yang tidak mendukung Amazon EFS.	Pastikan Anda menggunakan instans DB RDS for Oracle versi 19.0.0.0.ru-2022-07.rur-2022-07.r1 atau yang lebih tinggi.
`Read access of your EFS is denied. Check your file system policy.`	Instans DB Anda tidak dapat membaca sistem file EFS.	Pastikan sistem file EFS Anda mengizinkan akses baca melalui peran IAM atau pada tingkat sistem file EFS.
N/A	Instans DB Anda tidak dapat menulis ke sistem file EFS.	Lakukan langkah berikut: 1. Pastikan sistem file EFS Anda terpasang pada instans Amazon EC2. 2. Berikan akses tulis grup `others` ke pengguna RDS Anda. Teknik yang paling sederhana adalah dengan menjalankan perintah `chmod 777` pada direktori teratas sistem file EFS.

Notifikasi	Deskripsi	Tindakan
Perintah `host -s` mengembalikan *`hostname`* `not found: 3(NXDOMAIN)`	Anda menggunakan server DNS kustom.	Untuk menggunakan nama DNS dalam perintah `mount`, pastikan hal berikut sudah benar: • Instans DB penghubung berada di VPC dan dikonfigurasi untuk menggunakan server DNS yang disediakan oleh Amazon. Server DNS kustom tidak didukung. • VPC instans penghubung harus mengaktifkan Resolusi DNS dan Nama Host DNS. • Instans penghubung harus berada di dalam VPC yang sama dengan sistem file EFS.

Mesin virtual Oracle Java

Amazon RDS mendukung Oracle Java Virtual Machine (JVM) melalui penggunaan opsi `JVM`. Oracle Java menyediakan skema dan fungsi SQL yang memfasilitasi fitur Oracle Java dalam basis data Oracle. Untuk informasi selengkapnya, lihat Pengenalan Java dalam basis data Oracle di dokumentasi Oracle.

Anda dapat menggunakan Oracle JVM dengan versi Oracle Database berikut:

- Oracle Database 21c (21.0.0), semua versi
- Oracle Database 19c (19.0.0), semua versi
- Oracle Database 12c Rilis 2 (12.2), semua versi
- Oracle Database 12c Rilis 1 (12.1), versi 12.1.0.2.v13 dan yang lebih tinggi

Set izin untuk implementasi Java di Amazon RDS terbatas. Pengguna utama diberi peran `RDS_JAVA_ADMIN`, yang memberikan sebagian hak istimewa yang diberikan oleh peran `JAVA_ADMIN`. Untuk mencantumkan hak istimewa yang diberikan kepada peran `RDS_JAVA_ADMIN`, jalankan kueri berikut pada instans DB Anda:

```
SELECT * FROM dba_java_policy
   WHERE grantee IN ('RDS_JAVA_ADMIN', 'PUBLIC')
   AND enabled = 'ENABLED'
   ORDER BY type_name, name, grantee;
```

Prasyarat untuk Oracle JVM

Berikut adalah prasyarat untuk menggunakan Oracle Java:

- Instans DB harus dari kelas yang cukup besar. Oracle Java tidak didukung untuk kelas instans DB db.t3.micro atau db.t3.small. Untuk informasi selengkapnya, lihat Kelas instans DB .
- Instans DB Anda harus mengaktifkan Peningkatan Versi Minor Otomatis. Opsi ini memungkinkan instans DB Anda menerima peningkatan versi mesin DB minor secara otomatis saat tersedia. Amazon RDS menggunakan opsi ini untuk memperbarui instans DB Anda ke Oracle Patch Set Update (PSU) atau Release Update (RU) terbaru. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Praktik terbaik untuk Oracle JVM

Berikut adalah praktik terbaik untuk menggunakan Oracle Java:

- Untuk keamanan maksimal, gunakan opsi `JVM` dengan Secure Sockets Layer (SSL). Untuk informasi selengkapnya, lihat Lapisan Soket Aman Oracle.
- Konfigurasikan instans DB Anda untuk membatasi akses jaringan. Untuk informasi selengkapnya, lihat Skenario untuk mengakses instans DB di VPC dan Bekerja dengan klaster DB dalam VPC.
- Perbarui konfigurasi titik akhir HTTPS Anda untuk mendukung TLSv1.2 jika Anda memenuhi kondisi berikut:
  - Anda menggunakan Oracle Java Virtual Machine (JVM) untuk menghubungkan titik akhir HTTPS melalui protokol TLSv1 atau TLSv1.1.
  - Titik akhir Anda tidak mendukung protokol TLSv1.2.
  - Anda belum menerapkan pembaruan rilis April 2021 ke DB Oracle.

  Dengan memperbarui konfigurasi titik akhir, Anda memastikan bahwa konektivitas JVM ke titik akhir HTTPS akan terus berfungsi. Untuk informasi selengkapnya tentang perubahan TLS Oracle JRE dan JDK, lihat Oracle JRE and JDK Cryptographic Roadmap.

Menambahkan opsi Oracle JVM

Berikut adalah proses umum untuk menambahkan opsi `JVM` ke instans DB:

1. Buat grup opsi baru, atau salin atau ubah grup opsi yang ada.
2. Tambahkan opsi ke grup opsi.
3. Kaitkan grup opsi dengan instans DB.

Ada pemadaman singkat saat opsi `JVM` ditambahkan. Setelah opsi ditambahkan, instans DB tidak perlu dimulai ulang. Setelah grup opsi aktif, Oracle Java akan langsung tersedia.

> **Note**
>
> Selama terjadi pemadaman, fungsi verifikasi kata sandi akan dinonaktifkan sementara. Anda juga dapat melihat peristiwa terkait fungsi verifikasi kata sandi selama terjadi pemadaman. Fungsi verifikasi kata sandi akan diaktifkan kembali sebelum instans Oracle DB tersedia.

Untuk menambahkan opsi JVM ke instans DB

1. Tentukan grup opsi yang ingin Anda gunakan. Anda dapat membuat grup opsi baru atau menggunakan grup opsi yang ada. Jika Anda ingin menggunakan grup opsi yang ada, lanjutkan ke langkah berikutnya. Jika tidak, buat grup opsi DB kustom dengan pengaturan berikut:

   - Untuk Mesin, pilih mesin DB yang digunakan oleh instans DB (oracle-ee, oracle-se, oracle-se1, atau oracle-se2).
   - Untuk Versi mesin utama, pilih versi instans DB Anda.

   Untuk informasi selengkapnya, lihat Membuat grup opsi.
2. Tambahkan opsi JVM ke grup opsi. Untuk informasi selengkapnya tentang cara menambahkan opsi, lihat Menambahkan opsi ke grup opsi.
3. Terapkan grup opsi ke instans DB baru atau yang sudah ada:

   - Untuk instans DB baru, terapkan grup opsi saat Anda meluncurkan instans. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.
   - Untuk instans DB yang sudah ada, terapkan grup opsi dengan memodifikasi instans dan menambahkan grup opsi baru. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.
4. Berikan izin yang diperlukan kepada pengguna.

   Pengguna utama Amazon RDS memiliki izin untuk menggunakan opsi `JVM` secara default. Jika pengguna lain memerlukan izin ini, sambungkan ke instans DB sebagai pengguna utama di klien SQL dan berikan izin kepada pengguna.

   Contoh berikut memberikan izin untuk menggunakanopsi `JVM` ke pengguna `test_proc`.

```
create user test_proc identified by password;
CALL dbms_java.grant_permission('TEST_PROC',
 'oracle.aurora.security.JServerPermission', 'LoadClassInPackage.*', '');
```

> **Note**
>
> Tentukan kata sandi selain perintah yang ditampilkan di sini sebagai praktik terbaik keamanan.

Setelah pengguna diberi izin, kueri berikut harus mengembalikan output.

```
select * from dba_java_policy where grantee='TEST_PROC';
```

> **Note**
>
> Nama pengguna Oracle peka huruf besar/kecil, dan biasanya semua dalam huruf besar.

Menghapus opsi Oracle JVM

Anda dapat menghapus opsi `JVM` dari instans DB. Ada pemadaman singkat saat opsi dihapus. Setelah opsi `JVM` dihapus, instans DB tidak perlu dimulai ulang.

> **Warning**
>
> Menghapus opsi `JVM` dapat mengakibatkan kehilangan data jika instans DB menggunakan tipe data yang diaktifkan sebagai bagian dari opsi. Cadangkan data Anda sebelum melanjutkan. Untuk informasi selengkapnya, lihat Mencadangkan, memulihkan, dan mengekspor data.

Untuk menghapus opsi `JVM` dari instans DB, lakukan salah satu hal berikut:

- Hapus opsi `JVM` dari grup opsi asalnya. Perubahan ini memengaruhi semua instans DB yang menggunakan grup opsi tersebut. Untuk informasi selengkapnya, lihat Menghapus opsi dari grup opsi.
- Ubah instans DB dan tentukan grup opsi lain yang tidak menyertakan opsi `JVM`. Perubahan ini memengaruhi satu instans DB. Anda dapat menentukan grup opsi default (kosong) atau grup opsi kustom yang berbeda. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Oracle Enterprise Manager

Amazon RDS mendukung Oracle Enterprise Manager (OEM). OEM adalah lini produk Oracle untuk manajemen terintegrasi teknologi informasi perusahaan.

Amazon RDS mendukung OEM melalui opsi berikut.

Opsi	ID Opsi	Rilis OEM yang didukung	Rilis Oracle Database yang didukung
OEM Database Express	`OEM`	OEM Database Express 12c	Oracle Database 19c (khusus non-CDB) Oracle Database 12c
OEM Management Agent	`OEM_AGEN`	OEM Cloud Control untuk 13c OEM Cloud Control untuk 12c	Oracle Database 19c (khusus non-CDB) Oracle Database 12c

> **Note**
>
> Anda dapat menggunakan OEM Database atau OEM Management Agent, tetapi tidak keduanya.

> **Note**
>
> Opsi ini tidak didukung untuk arsitektur multipenghuni Oracle.

Oracle Enterprise Manager Database Express

Amazon RDS mendukung Oracle Enterprise Manager (OEM) Database Express melalui penggunaan opsi OEM. Amazon RDS mendukung Oracle Enterprise Database Express untuk rilis berikut:

- Oracle Database 19c (khusus non-CDB)
- Oracle Database 12c

OEM Database Express dan Database Control adalah alat serupa yang memiliki antarmuka berbasis web untuk administrasi basis data Oracle. Untuk informasi selengkapnya tentang alat ini, lihat Accessing Enterprise Manager database Express 18c dan Accessing Enterprise Manager Database Express 12c dalam dokumentasi Oracle.

Berikut ini adalah batasan untuk OEM Database Express:

- OEM Database Express tidak didukung pada kelas instans DB db.t3.micro atau db.t3.small.

  Untuk informasi lebih lanjut tentang kelas instans DB, lihat Kelas instans RDS for Oracle.

Pengaturan opsi OEM Database

Amazon RDS mendukung pengaturan berikut untuk opsi OEM.

Pengaturan opsi	Nilai valid	Deskripsi
Port	Nilai bilangan bulat	Port pada instans DB yang mendengarkan OEM Database. Default untuk OEM Database Express adalah 5500.
Grup Keamanan	—	Grup keamanan yang memiliki akses ke Port.

Menambahkan opsi OEM Database

Proses umum untuk menambahkan opsi OEM ke instans DB adalah sebagai berikut:

1. Buat grup opsi baru, atau salin atau ubah grup opsi yang ada.
2. Tambahkan opsi ke grup opsi.
3. Kaitkan grup opsi dengan instans DB.

Saat Anda menambahkan opsi OEM untuk instans Oracle Database 12c atau DB yang lebih baru, pemadaman singkat akan terjadi saat instans DB Anda dimulai ulang secara otomatis.

Untuk menambahkan opsi OEM ke instans DB

1. Tentukan grup opsi yang ingin Anda gunakan. Anda dapat membuat grup opsi baru atau menggunakan grup opsi yang ada. Jika Anda ingin menggunakan grup opsi yang ada, lanjutkan ke langkah berikutnya. Jika tidak, buat grup opsi DB kustom dengan pengaturan berikut:

   a. Untuk Mesin pilih edisi Oracle untuk instans DB Anda.

   b. Untuk Versi mesin utama, pilih versi instans DB Anda.

   Untuk informasi selengkapnya, lihat Membuat grup opsi.

2. Tambahkan opsi OEM ke grup opsi dan konfigurasikan pengaturan opsi. Untuk informasi selengkapnya tentang cara menambahkan opsi, lihat Menambahkan opsi ke grup opsi. Untuk informasi selengkapnya tentang setiap pengaturan, lihat Pengaturan opsi OEM Database.

   > **Note**
   >
   > Jika Anda menambahkan opsi OEM ke grup opsi yang sudah ada yang sudah terpasang ke satu instans DB Oracle Database 19c (khusus non-CDB) atau Oracle Database 12c, pemadaman singkat akan terjadi saat semua instans DB dimulai ulang secara otomatis.

3. Terapkan grup opsi ke instans DB baru atau yang sudah ada:

   - Untuk instans DB baru, Anda menerapkan grup opsi saat Anda meluncurkan instans. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.
   - Untuk instans DB yang ada, Anda menerapkan grup opsi dengan memodifikasi instans dan melampirkan grup opsi baru. Saat Anda menambahkan opsi OEM untuk instans DB Oracle Database 19c (khusus non-CDB) atau Oracle Database 12c, pemadaman singkat akan terjadi saat instans DB Anda dimulai ulang secara otomatis. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

> **Note**
>
> Anda juga dapat menggunakan AWS CLI untuk menambahkan opsi OEM. Sebagai contoh, lihat Menambahkan opsi ke grup opsi.

Mengakses OEM melalui browser Anda

Setelah Anda mengaktifkan opsi OEM, Anda dapat mulai menggunakan alat OEM Database dari browser web Anda.

Anda dapat mengakses OEM Database Control atau OEM Database Express dari browser web Anda. Misalnya, jika titik akhir untuk instans DB Amazon RDS Anda adalah `mydb.f9rbfa893tft.us-east-1.rds.amazonaws.com` dan port OEM Anda adalah 1158, maka URL untuk mengakses OEM Database Control adalah sebagai berikut.

```
https://mydb.f9rbfa893tft.us-east-1.rds.amazonaws.com:1158/em
```

Saat Anda mengakses salah satu alat dari browser web Anda, jendela login muncul yang meminta Anda memasukkan nama pengguna dan kata sandi. Ketikkan nama pengguna master dan kata sandi master untuk instans DB Anda. Anda sekarang siap untuk mengelola basis data Oracle Anda.

Mengubah pengaturan OEM Database

Setelah Anda mengaktifkan OEM Database, Anda dapat mengubah pengaturan Grup Keamanan untuk opsi tersebut.

Anda tidak dapat mengubah nomor port OEM setelah Anda mengaitkan grup opsi dengan instans DB. Untuk mengubah nomor port OEM untuk instans DB, lakukan hal berikut:

1. Buat grup opsi baru.
2. Tambahkan opsi OEM dengan nomor port baru ke grup opsi baru.
3. Hapus grup opsi yang ada dari instans DB.
4. Tambahkan grup opsi baru ke instans DB.

Untuk informasi selengkapnya tentang cara mengubah pengaturan opsi, lihat Memodifikasi pengaturan opsi. Untuk informasi selengkapnya tentang setiap pengaturan, lihat Pengaturan opsi OEM Database.

Menjalankan tugas OEM Database Express

Anda dapat menggunakan prosedur Amazon RDS untuk menjalankan tugas OEM Database Express tertentu. Dengan menjalankan prosedur ini, Anda dapat melakukan tugas berikut.

> **Note**
>
> Tugas OEM Database Express berjalan secara asinkron.

Tugas

- Mengalihkan frontend situs web untuk OEM Database Express ke Adobe Flash
- Mengalihkan frontend situs web untuk OEM Database Express ke Oracle JET

Mengalihkan frontend situs web untuk OEM Database Express ke Adobe Flash

> **Note**
>
> Tugas ini hanya tersedia untuk Oracle Database 19c non-CDB.

Dimulai dengan Oracle Dabase 19c, Oracle tidak lagi menggunakan antarmuka pengguna OEM Database Express lama, yang didasarkan pada Adobe Flash. Sebagai gantinya, OEM Database Express sekarang menggunakan antarmuka yang dibangun dengan Oracle JET. Jika Anda mengalami kesulitan dengan antarmuka baru ini, Anda dapat beralih kembali ke antarmuka berbasis Flash yang sudah tidak digunakan lagi. Kesulitan yang mungkin Anda alami dengan antarmuka baru termasuk mengalami macet di layar `Loading` setelah masuk ke OEM Database Express. Anda mungkin juga tidak mendapati fitur tertentu yang ada di versi berbasis Flash dari OEM Database Express.

Untuk mengalihkan frontend situs web OEM Database Express ke Adobe Flash, jalankan prosedur `rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_flash` Amazon RDS. Prosedur ini setara dengan perintah SQL `execemx emx`.

Praktik terbaik keamanan mencegah penggunaan Adobe Flash. Meskipun Anda dapat kembali ke OEM Database Express berbasis Flash, kami merekomendasikan penggunaan situs web OEM Database Express berbasis JET jika memungkinkan. Jika Anda kembali menggunakan Adobe Flash dan ingin beralih kembali menggunakan Oracle JET, gunakan prosedur

`rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_jet`. Setelah peningkatan basis data Oracle, versi terbaru dari Oracle JET mungkin dapat menyelesaikan masalah terkait JET di OEM Database Express. Untuk informasi lebih lanjut tentang cara beralih ke Oracle JET, lihat Mengalihkan frontend situs web untuk OEM Database Express ke Oracle JET.

> **Note**
>
> Menjalankan tugas ini dari instans DB sumber untuk replika baca juga menyebabkan replika baca mengalihkan frontend situs web OEM Database Express ke Adobe Flash.

Invokasi prosedur berikut membuat tugas untuk mengalihkan situs web OEM Database Express ke Adobe Flash dan menampilkan ID tugas.

```
SELECT rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_flash() as TASK_ID from DUAL;
```

Anda dapat melihat hasilnya dengan menampilkan file output tugas.

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-task-id.log'));
```

Ganti *`task-id`* dengan ID tugas yang ditampilkan oleh prosedur. Untuk informasi lebih lanjut tentang prosedur `rdsadmin.rds_file_util.read_text_file` Amazon RDS, lihat Membaca file di direktori instans DB

Anda juga dapat melihat konten file output tugas di AWS Management Console dengan mencari entri log di bagian Log & peristiwa untuk `task-id`.

Mengalihkan frontend situs web untuk OEM Database Express ke Oracle JET

> **Note**
>
> Tugas ini hanya tersedia untuk Oracle Database 19c non-CDB.

Untuk mengalihkan frontend situs web OEM Database Express ke Oracle JET, jalankan prosedur `rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_jet` Amazon RDS. Prosedur ini setara dengan perintah SQL `execemx omx`.

Secara default, situs web OEM Database Express untuk instans DB Oracle yang menjalankan 19c atau versi yang lebih baru menggunakan Oracle JET. Jika Anda telah menggunakan prosedur `rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_flash` untuk mengalihkan frontend situs web OEM Database Express ke Adobe Flash, Anda dapat beralih kembali ke Oracle JET. Untuk melakukan ini, gunakan prosedur `rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_jet`. Untuk informasi lebih lanjut tentang cara beralih ke Adobe Flash, lihat Mengalihkan frontend situs web untuk OEM Database Express ke Adobe Flash.

> **Note**
>
> Menjalankan tugas ini dari instans DB sumber untuk replika baca juga menyebabkan replika baca mengalihkan frontend situs web OEM Database Express ke Oracle JET.

Invokasi prosedur berikut membuat tugas untuk mengalihkan situs web OEM Database Express ke Oracle JET dan menampilkan ID tugas tersebut.

```
SELECT rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_jet() as TASK_ID from DUAL;
```

Anda dapat melihat hasilnya dengan menampilkan file output tugas.

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-task-id.log'));
```

Ganti *`task-id`* dengan ID tugas yang ditampilkan oleh prosedur. Untuk informasi lebih lanjut tentang prosedur `rdsadmin.rds_file_util.read_text_file` Amazon RDS, lihat Membaca file di direktori instans DB

Anda juga dapat melihat konten file output tugas di AWS Management Console dengan mencari entri log di bagian Log & peristiwa untuk `task-id`.

Menghapus opsi OEM Database

Anda dapat menghapus opsi OEM dari instans DB. Saat Anda menghapus opsi OEM untuk instans DB Oracle Database 12c atau yang lebih baru, pemadaman singkat akan terjadi saat instans DB Anda dimulai ulang secara otomatis. Oleh karena itu, setelah Anda menghapus opsi OEM, Anda tidak perlu memulai ulang instans DB Anda.

Untuk menghapus opsi OEM dari instans DB, lakukan salah satu hal berikut ini:

- Hapus opsi OEM dari grup opsi yang mencakupnya. Perubahan ini memengaruhi semua instans DB yang menggunakan grup opsi tersebut. Untuk informasi selengkapnya, lihat Menghapus opsi dari grup opsi.
- Ubah instans DB dan tentukan grup opsi berbeda yang tidak menyertakan opsi OEM. Perubahan ini memengaruhi instans DB tunggal. Anda dapat menentukan grup opsi default (kosong) atau grup opsi kustom yang berbeda. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Oracle Management Agent untuk Kontrol Cloud Enterprise Manager

Oracle Enterprise Manager (OEM) Management Agent adalah komponen perangkat lunak yang memantau target yang berjalan pada host dan mengomunikasikan informasi tersebut ke Oracle Management Service (OMS) tingkat menengah. Untuk informasi lebih lanjut, lihat Overview of Oracle Enterprise Manager cloud control 12c dan Overview of Oracle Enterprise Manager cloud control 13c dalam dokumentasi Oracle.

Amazon RDS mendukung Management Agent melalui penggunaan opsi `OEM_AGENT`. Management Agent memerlukan instans DB Amazon RDS yang menjalankan salah satu rilis berikut:

- Oracle Database 19c (19.0.0.0) yang menggunakan arsitektur non-CDB
- Oracle Database 12c Rilis 2 (12.2.0.1)
- Oracle Database 12c Rilis 1 (12.1.0.2)

Amazon RDS mendukung Management Agent untuk versi OEM berikut:

- Oracle Enterprise Manager Cloud Control untuk 13c
- Oracle Enterprise Manager Cloud Control untuk 12c

Topik

Prasyarat untuk Management Agent

Untuk menggunakan Management Agent, pastikan bahwa Anda memenuhi prasyarat berikut.

Prasyarat umum

Berikut ini adalah prasyarat umum penggunaan Management Agent:

- Anda memerlukan Oracle Management Service (OMS) yang dikonfigurasi untuk terhubung ke instans DB Amazon RDS Anda.
- Dalam kebanyakan kasus, Anda harus mengonfigurasi VPC Anda untuk mengizinkan koneksi dari OMS ke instans DB Anda. Jika Anda tidak familier dengan Amazon Virtual Private Cloud (Amazon VPC), kami menyarankan Anda menyelesaikan langkah-langkah di Tutorial: Membuat VPC untuk digunakan dengan instans DB (khusus IPv4) sebelum melanjutkan.
- Management Agent versi 13.5.0.0.v1 memerlukan OMS versi 13.5.0.0 atau yang lebih baru.
- Management Agent versi 13.4.0.9.v1 memerlukan OMS versi 13.4.0.9 atau yang lebih baru dan patch 32198287.
- Pastikan Anda memiliki ruang penyimpanan yang cukup untuk rilis OEM Anda:
  - Setidaknya 8,5 GiB untuk OEM 13c Rilis 5
  - Setidaknya 8,5 GiB untuk OEM 13c Rilis 4
  - Setidaknya 8,5 GiB untuk OEM 13c Rilis 3
  - Setidaknya 5,5 GiB untuk OEM 13c Rilis 2
  - Setidaknya 4,5 GiB OEM 13c Rilis 1
  - Setidaknya 2,5 GiB untuk OEM 12c
- Jika Anda menggunakan Management Agent versi `OEM_AGENT 13.2.0.0.v3` dan`13.3.0.0.v2`, dan jika Anda ingin menggunakan konektivitas TCPS, ikuti petunjuk di Configuring third party CA certificates for communication with target databases dalam dokumentasi Oracle. Selain itu, perbarui JDK di OMS Anda dengan mengikuti petunjuk di dokumen Oracle, yaitu Oracle Doc ID 2241358.1. Langkah ini memastikan bahwa OMS mendukung semua rangkaian penyandian yang didukung basis data tersebut.

> **Note**
>
> Konektivitas TCPS antara Management Agent dan instans DB didukung untuk Management Agent `OEM_AGENT 13.2.0.0.v3`, `13.3.0.0.v2`, `13.4.0.9.v1`, dan versi yang lebih tinggi.

Prasyarat rilis Oracle Database

Berikut adalah versi Oracle Database yang didukung untuk setiap versi Management Agent.

Versi Management Agent	Oracle Database 19c yang menggunakan arsitektur non-CDB	Oracle Database 12c Rilis 2 (12.2)	Oracle Database 12c Rilis 1 (12.1)
13.5.0.0.v1	Didukung	Didukung	Didukung
13.4.0.9.v1	Didukung	Didukung	Didukung
13.3.0.0.v2	Didukung	Didukung	Didukung
13.3.0.0.v1	Didukung	Didukung	Didukung
13.2.0.0.v3	Didukung	Didukung	Didukung
13.2.0.0.v2	Didukung	Didukung	Didukung
13.2.0.0.v1	Didukung	Didukung	Didukung
13.1.0.0.v1	Didukung	Didukung	Didukung
12.1.0.5.v1	Tidak didukung	Didukung	Didukung
12.1.0.4.v1	Tidak didukung	Didukung	Didukung

Berikut ini adalah prasyarat untuk versi basis data yang berbeda:

- Untuk instans DB Amazon RDS yang menjalankan Oracle Database 19c (19.0.0.0), `AGENT_VERSION` minimum adalah 13.1.0.0.v1.
- Untuk instans DB Amazon RDS yang menjalankan Oracle Database Release 2 (12.2.0.1) atau lebih rendah, penuhi persyaratan berikut:
  - Untuk OMS 13c Rilis 2 dengan Oracle patch 25163555 diterapkan, gunakan OEM Agent 13.2.0.0.v2 atau yang lebih baru.

    Gunakan OMSPatcher untuk menerapkan patch.
  - Untuk OMS 13c Rilis 2 yang tidak menerapkan patch, gunakan OEM Agent 13.2.0.0.v1.

Gunakan OMSPatcher untuk menerapkan patch.

Prasyarat komunikasi host OMS

Pastikan host OMS dan instans DB Amazon RDS Anda dapat berkomunikasi. Lakukan hal berikut:

- Untuk menghubungkan dari Management Agent ke OMS Anda, jika OMS Anda berada di belakang firewall, tambahkan alamat IP instans DB Anda ke OMS Anda.

  Pastikan firewall untuk OMS mengizinkan lalu lintas dari port pendengar DB (default 1521) dan port OEM Agent (default 3872), yang berasal dari alamat IP instans DB.
- Untuk menyambung dari OMS Anda ke Management Agent, jika OMS Anda memiliki nama host yang dapat diselesaikan secara publik, tambahkan alamat OMS ke grup keamanan. Grup keamanan Anda harus memiliki aturan masuk yang memungkinkan akses ke port pendengar DB dan port Management Agent. Untuk contoh pembuatan keamanan dan menambahkan aturan masuk, lihat Tutorial: Membuat VPC untuk digunakan dengan instans DB (khusus IPv4).
- Untuk menyambungkan dari OMS Anda ke Management Agent, jika OMS Anda tidak memiliki nama host yang dapat diselesaikan secara publik, gunakan salah satu dari yang berikut ini:
  - Jika OMS Anda di-hosting di instans Amazon Elastic Compute Cloud (Amazon EC2) di VPC privat, Anda dapat menyiapkan peering VPC untuk menghubungkan dari OMS ke Management Agent. Untuk informasi lebih lanjut, lihat Instans DB dalam VPC yang diakses oleh instans EC2 dalam VPC yang sama.
  - Jika OMS Anda di-hosting secara on-premise, Anda dapat menyiapkan koneksi VPN untuk mengizinkan akses dari OMS ke Management Agent. Untuk informasi lebih lanjut, lihat Instans DB dalam VPC yang diakses oleh aplikasi klien melalui internet atau koneksi VPN.

Batasan untuk Management Agent

Berikut adalah beberapa batasan dalam menggunakan Management Agent:

- Anda tidak dapat memberikan gambar Agen Manajemen Oracle kustom.
- Tugas administratif seperti pelaksanaan pekerjaan dan patch basis data, yang memerlukan kredensial host, tidak didukung.
- Metrik host dan daftar proses belum tentu mencerminkan status sistem yang sebenarnya. Dengan demikian, Anda tidak boleh menggunakan OEM untuk memantau sistem file root atau sistem file

titik pemasangan. Untuk informasi lebih lanjut tentang memantau sistem operasi, lihat Memantau metrik OS dengan Pemantauan yang Disempurnakan.

- Penemuan otomatis tidak didukung. Anda harus menambahkan target basis data secara manual.
- Ketersediaan modul OMS tergantung pada edisi basis data Anda. Misalnya, modul penyesuaian dan diagnosis performa basis data hanya tersedia untuk Oracle Database Enterprise Edition.
- Management Agent menggunakan memori tambahan dan sumber daya komputasi. Jika Anda mengalami masalah performa setelah mengaktifkan opsi `OEM_AGENT`, sebaiknya naikkan skala ke kelas instans DB yang lebih besar. Untuk informasi lebih lanjut, lihat Kelas instans DB dan Memodifikasi instans DB Amazon RDS.
- Pengguna yang menjalankan `OEM_AGENT` pada host Amazon RDS tidak memiliki akses sistem operasi ke log peringatan. Dengan demikian, Anda tidak dapat mengumpulkan metrik untuk `DB Alert Log` dan `DB Alert Log Error Status` di OEM.

Pengaturan opsi untuk Management Agent

Amazon RDS mendukung pengaturan berikut untuk opsi Management Agent.

Pengaturan opsi	Dibutuhkan	Nilai valid	Deskripsi

Versi (`AGENT_VERSION`)	Ya	`13.5.0.0.v1`
		`13.4.0.9.v1`
		`13.3.0.0.v2`
		`13.3.0.0.v1`
		`13.2.0.0.v3`
		`13.2.0.0.v2`
			Versi perangkat lunak Management Agent.
			Nama opsi AWS CLI adalah `OptionVersion`.
			Note
			Di Wilayah AWS GovCloud (US), versi 12.1 dan 13.1 tidak tersedia.

Pengaturan opsi	Dibutuhkan	Nilai valid	Deskripsi

		`13.2.0.0.v1`
		`13.1.0.0.v1`
		`12.1.0.5.v1`
		`12.1.0.4.v1`	

Port (`AGENT_PORT`)	Ya	Nilai integer	Port pada instans DB yang mendengarkan host OMS. Default-nya adalah 3872. Host OMS Anda harus tercakup dalam grup keamanan yang memiliki akses ke port ini. Nama opsi AWS CLI adalah `Port`.
Grup Keamanan	Ya	Grup keamanan yang ada	Grup keamanan yang memiliki akses ke Port. Host OMS Anda harus tercakup dalam grup keamanan ini. Nama opsi AWS CLI adalah `VpcSecurityGroupMemberships` atau `DBSecurityGroupMemberships`.
OMS_HOST	Ya	Nilai string, misalnya *`my.example.oms`*	Nama host atau alamat IP OMS yang dapat diakses publik. Nama opsi AWS CLI adalah `OMS_HOST`.

Pengaturan opsi	Dibutuhkan	Nilai valid	Deskripsi
OMS_PORT	Ya	Nilai integer	Port upload HTTPS di OMS Host yang mendengarkan Management Agent. Untuk menentukan port upload HTTPS, sambungkan ke host OMS, dan jalankan perintah berikut (yang memerlukan kata sandi `SYSMAN`): `emctl status oms -details` Nama opsi AWS CLI adalah `OMS_PORT`.
AGENT_REGISTRATION_PASSWORD	Ya	Nilai string.	Kata sandi yang digunakan Management Agent untuk mengautentikasi dirinya sendiri dengan OMS. Kami menyarankan Anda untuk membuat kata sandi yang persisten di OMS Anda sebelum mengaktifkan opsi `OEM_AGENT`. Dengan kata sandi yang persisten, Anda dapat berbagi satu grup opsi Management Agent dengan beberapa basis data Amazon RDS. Nama opsi AWS CLI adalah `AGENT_REGISTRATION_PASSWORD`.
ALLOW_TLS_ONLY	Tidak	`true`, `false` (default)	Nilai yang mengonfigurasi OEM Agent untuk hanya mendukung protokol `TLSv1` saat agen mendengarkan sebagai server. Pengaturan ini hanya didukung untuk agen versi 12.1. Versi agen yang lebih baru hanya mendukung Keamanan Lapisan Pengangkutan (TLS) secara default.

Pengaturan opsi	Dibutuhkan	Nilai valid	Deskripsi
MINIMUM_TLS_VERSION	Tidak	`TLSv1` (default), `TLSv1.2`	Nilai yang menentukan versi TLS minimum yang didukung oleh OEM Agent saat agen mendengarkan sebagai server. Pengaturan ini hanya didukung untuk agen versi 13.1.0.0.v1 dan yang lebih tinggi. Versi agen sebelumnya hanya mendukung pengaturan `TLSv1`.
TLS_CIPHER_SUITE	Tidak	Lihat Pengaturan TLS untuk opsi Management Agent.	Nilai yang menentukan rangkaian penyandian TLS yang digunakan oleh OEM Agent saat agen mendengarkan sebagai server.

Tabel berikut mencantumkan rangkaian penyandian TLS yang mendukung opsi Management Agent.

Pengaturan TLS untuk opsi Management Agent

Rangkaian Penyandian	Versi Agent yang didukung	Sesuai FedRAMP
TLS_RSA_WITH_AES_128_CBC_SHA	Semua	Tidak
TLS_RSA_WITH_AES_128_CBC_SHA256	13.1.0.0.v1 atau yang lebih tinggi	Tidak
TLS_RSA_WITH_AES_256_CBC_SHA	13.2.0.0.v3 atau yang lebih tinggi	Tidak
TLS_RSA_WITH_AES_256_CBC_SHA256	13.2.0.0.v3 atau yang lebih tinggi	Tidak
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	13.2.0.0.v3 atau yang lebih tinggi	Ya

Rangkaian Penyandian	Versi Agent yang didukung	Sesuai FedRAMP
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	13.2.0.0.v3 atau yang lebih tinggi	Ya
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	13.2.0.0.v3 atau yang lebih tinggi	Ya
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	13.2.0.0.v3 atau yang lebih tinggi	Ya

Menambahkan opsi Management Agent

Proses umum untuk menambahkan opsi Management Agent ke instans DB adalah sebagai berikut:

1. Buat grup opsi baru, atau salin atau ubah grup opsi yang ada.
2. Tambahkan opsi tersebut ke grup opsi.
3. Kaitkan grup opsi tersebut dengan instans DB.

Jika Anda mengalami kesalahan, periksa dokumen My Oracle Support untuk informasi tentang menyelesaikan masalah tertentu.

Setelah opsi Management Agent ditambahkan, instans DB tidak perlu dimulai ulang. Segera setelah grup opsi aktif, OEM Agent aktif.

Jika host OMS Anda menggunakan sertifikat pihak ketiga yang tidak tepercaya, Amazon RDS mengembalikan kesalahan berikut.

```
You successfully installed the OEM_AGENT option. Your OMS host is using an untrusted
 third party certificate.
Configure your OMS host with the trusted certificates from your third party.
```

Jika kesalahan ini dikembalikan, opsi Management Agent tidak diaktifkan hingga masalah diperbaiki. Untuk informasi tentang memperbaiki masalah tersebut, lihat dokumen My Oracle Support 2202569.1.

Menghibur

Untuk menambahkan opsi Management Agent ke instans DB

1. Tentukan grup opsi yang ingin Anda gunakan. Anda dapat membuat grup opsi baru atau menggunakan grup opsi yang ada. Jika Anda ingin menggunakan grup opsi yang ada, lanjutkan ke langkah berikutnya. Jika tidak, buat grup opsi DB kustom dengan pengaturan berikut:

   a. Untuk Mesin pilih edisi Oracle untuk instans DB Anda.

   b. Untuk Versi mesin utama, pilih versi instans DB Anda.

   Untuk informasi selengkapnya, lihat Membuat grup opsi.
2. Tambahkan opsi OEM_AGENT ke grup opsi dan konfigurasikan pengaturan opsi. Untuk informasi selengkapnya tentang cara menambahkan opsi, lihat Menambahkan opsi ke grup opsi. Untuk informasi selengkapnya tentang setiap pengaturan, lihat Pengaturan opsi untuk Management Agent.
3. Terapkan grup opsi ke instans DB baru atau yang sudah ada:

   - Untuk instans DB baru, Anda menerapkan grup opsi saat Anda meluncurkan instans. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.
   - Untuk instans DB yang ada, Anda menerapkan grup opsi dengan memodifikasi instans dan melampirkan grup opsi baru. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

AWS CLI

Contoh berikut menggunakan perintah AWS CLI add-option-to-option-group untuk menambahkan `OEM_AGENT` opsi ke grup opsi yang disebut`myoptiongroup`.

Untuk Linux, macOS, atau Unix:

```
aws rds add-option-to-option-group \
    --option-group-name "myoptiongroup" \
    --options
 OptionName=OEM_AGENT,OptionVersion=13.1.0.0.v1,Port=3872,VpcSecurityGroupMemberships=sg-123456
{Name=OMS_PORT,Value=4903},{Name=AGENT_REGISTRATION_PASSWORD,Value=password}] \
    --apply-immediately
```

Untuk Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name "myoptiongroup" ^
    --options
 OptionName=OEM_AGENT,OptionVersion=13.1.0.0.v1,Port=3872,VpcSecurityGroupMemberships=sg-123456
{Name=OMS_PORT,Value=4903},{Name=AGENT_REGISTRATION_PASSWORD,Value=password}] ^
    --apply-immediately
```

Menggunakan Management Agent

Setelah Anda mengaktifkan opsi Management Agent, lakukan langkah-langkah berikut untuk mulai menggunakannya.

Untuk menggunakan Management Agent

1. Buka kunci dan atur ulang kredensial akun DBSNMP. Lakukan hal ini dengan menjalankan kode berikut pada basis data target Anda di instans DB Anda dan menggunakan akun pengguna master Anda.

   ```
   ALTER USER dbsnmp IDENTIFIED BY new_password ACCOUNT UNLOCK;
   ```

2. Tambahkan target Anda ke konsol OMS secara manual:

   a. Di konsol OMS Anda, pilih Pengaturan, Tambahkan Target, Tambahkan Target Secara Manual.

   b. Pilih Tambahkan Target Secara Deklaratif dengan Menentukan Properti Pemantauan Target.

   c. Untuk Jenis Target, pilih Instans Basis Data.

   d. Untuk Agen Pemantau, pilih agen dengan pengidentifikasi yang sama dengan pengidentifikasi instans DB RDS Anda.

   e. Pilih Tambahkan Secara Manual.

   f. Masukkan titik akhir untuk instans DB Amazon RDS, atau pilih dari daftar nama host. Pastikan bahwa nama host yang ditentukan cocok dengan titik akhir dari instans DB Amazon RDS.

      Untuk informasi tentang cara menemukan titik akhir untuk instans DB Amazon RDS Anda, lihat Menemukan titik akhir instans DB RDS for Oracle.

   g. Tentukan properti basis data berikut:

- Untuk Nama target, masukkan nama.
- Untuk Nama sistem basis data, masukkan nama.
- Untuk Nama pengguna pemantauan, masukkan **`dbsnmp`**.
- Untuk Kata sandi pemantauan, masukkan kata sandi dari langkah 1.
- Untuk Peran, masukkan normal.
- Untuk Jalur beranda Oracle, masukkan **`/oracle`**.
- Untuk Nama Mesin Pendengar, pengidentifikasi agen sudah muncul.
- Untuk Port, masukkan port basis data. Port default RDS adalah 1521.
- Untuk Nama basis data, masukkan nama basis data Anda.

h. Pilih Uji Koneksi.

i. Pilih Selanjutnya. Basis data target muncul di daftar sumber daya yang dipantau.

Mengubah pengaturan Management Agent

Setelah Anda mengaktifkan Management Agent, Anda dapat mengubah pengaturan untuk opsi tersebut. Untuk informasi selengkapnya tentang cara mengubah pengaturan opsi, lihat Memodifikasi pengaturan opsi. Untuk informasi selengkapnya tentang setiap pengaturan, lihat Pengaturan opsi untuk Management Agent.

Melakukan tugas basis data dengan Management Agent

Anda dapat menggunakan prosedur Amazon RDS untuk menjalankan perintah EMCTL tertentu di Management Agent. Dengan menjalankan prosedur ini, Anda dapat melakukan tugas-tugas berikut ini.

> **Note**
>
> Tugas dijalankan secara asinkron.

Tugas

- Mendapatkan status Management Agent
- Memulai ulang Management Agent
- Membuat daftar target yang dipantau oleh Management Agent
- Membuat daftar utas koleksi yang dipantau oleh Management Agent

- Menghapus status Management Agent
- Membuat Management Agent mengunggah OMS-nya
- Mengirim ping pada OMS
- Melihat status tugas yang sedang berlangsung

Mendapatkan status Management Agent

Untuk mendapatkan status Management Agent, jalankan prosedur `rdsadmin.rdsadmin_oem_agent_tasks.get_status_oem_agent` Amazon RDS. Prosedur ini setara dengan perintah `emctl status agent`.

Prosedur berikut membuat tugas untuk mendapatkan status Management Agent dan mengembalikan ID tugas.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.get_status_oem_agent() as TASK_ID from DUAL;
```

Untuk melihat hasilnya dengan menampilkan file output tugas, lihat Melihat status tugas yang sedang berlangsung.

Memulai ulang Management Agent

Untuk memulai ulang Management Agent, jalankan prosedur `rdsadmin.rdsadmin_oem_agent_tasks.restart_oem_agent` Amazon RDS. Prosedur ini setara dengan menjalankan perintah `emctl stop agent` dan `emctl start agent`.

Prosedur berikut membuat tugas untuk memulai ulang Management Agent dan mengembalikan ID tugas.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.restart_oem_agent as TASK_ID from DUAL;
```

Untuk melihat hasilnya dengan menampilkan file output tugas, lihat Melihat status tugas yang sedang berlangsung.

Membuat daftar target yang dipantau oleh Management Agent

Untuk membuat daftar target yang dipantau oleh Management Agent, jalankan prosedur `rdsadmin.rdsadmin_oem_agent_tasks.list_targets_oem_agent` Amazon RDS. Prosedur ini setara dengan menjalankan perintah `emctl config agent listtargets`.

Prosedur berikut membuat tugas untuk membuat daftar target yang dipantau oleh Management Agent dan mengembalikan ID tugas.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.list_targets_oem_agent as TASK_ID from DUAL;
```

Untuk melihat hasilnya dengan menampilkan file output tugas, lihat Melihat status tugas yang sedang berlangsung.

Membuat daftar utas koleksi yang dipantau oleh Management Agent

Untuk membuat daftar semua utas koleksi yang sedang berjalan, sudah siap, dan terjadwal yang dipantau oleh Management Agent, jalankan prosedur `rdsadmin.rdsadmin_oem_agent_tasks.list_clxn_threads_oem_agent` Amazon RDS. Prosedur ini setara dengan perintah `emctl status agent scheduler`.

Prosedur berikut membuat tugas untuk membuat daftar utas koleksi dan mengembalikan ID tugas.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.list_clxn_threads_oem_agent() as TASK_ID from
 DUAL;
```

Untuk melihat hasilnya dengan menampilkan file output tugas, lihat Melihat status tugas yang sedang berlangsung.

Menghapus status Management Agent

Untuk menghapus Management Agent, jalankan prosedur `rdsadmin.rdsadmin_oem_agent_tasks.clearstate_oem_agent` Amazon RDS. Prosedur ini setara dengan menjalankan perintah `emctl clearstate agent`.

Prosedur berikut membuat tugas yang menghapus status Management Agent dan mengembalikan ID tugas.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.clearstate_oem_agent() as TASK_ID from DUAL;
```

Untuk melihat hasilnya dengan menampilkan file output tugas, lihat Melihat status tugas yang sedang berlangsung.

Membuat Management Agent mengunggah OMS-nya

Agar Management Agent mengunggah Oracle Management Server (OMS) yang terkait dengannya, jalankan prosedur `rdsadmin.rdsadmin_oem_agent_tasks.upload_oem_agent` Amazon RDS. Prosedur ini setara dengan menjalankan perintah `emclt upload agent`.

Prosedur berikut membuat tugas yang membuat Management Agent mengunggah OMS terkaitnya dan mengembalikan ID tugas.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.upload_oem_agent() as TASK_ID from DUAL;
```

Untuk melihat hasilnya dengan menampilkan file output tugas, lihat Melihat status tugas yang sedang berlangsung.

Mengirim ping pada OMS

Untuk mengirim ping ke OMS Management Agent, jalankan prosedur `rdsadmin.rdsadmin_oem_agent_tasks.ping_oms_oem_agent` Amazon RDS. Prosedur ini setara dengan menjalankan perintah `emctl pingOMS`.

Prosedur berikut membuat tugas yang mengirim ping ke OMS Management Agent dan mengembalikan ID tugas.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.ping_oms_oem_agent() as TASK_ID from DUAL;
```

Untuk melihat hasilnya dengan menampilkan file output tugas, lihat Melihat status tugas yang sedang berlangsung.

Melihat status tugas yang sedang berlangsung

Anda dapat melihat status tugas yang sedang berlangsung di file bdump. File bdump terletak di direktori `/rdsdbdata/log/trace`. Setiap nama file bdump memiliki format berikut.

```
dbtask-task-id.log
```

Saat Anda ingin memantau tugas, ganti *`task-id`* dengan ID tugas yang ingin Anda pantau.

Untuk melihat isi file bdump, jalankan prosedur `rdsadmin.rds_file_util.read_text_file` Amazon RDS. Misalnya, kueri berikut mengembalikan isi file bdump `dbtask-1546988886389-2444.log`.

```
SELECT text FROM
 table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-1546988886389-2444.log'));
```

Untuk informasi lebih lanjut tentang prosedur `rdsadmin.rds_file_util.read_text_file` Amazon RDS, lihat Membaca file di direktori instans DB.

Menambahkan opsi Management Agent

Anda dapat menghapus OEM Agent dari instans DB. Setelah OEM Agent dihapus, instans DB tidak perlu dimulai ulang.

Untuk menghapus OEM Agent dari instans DB, lakukan salah satu hal berikut ini:

- Hapus OEM Agent dari grup opsi tempatnya berada. Perubahan ini memengaruhi semua instans DB yang menggunakan grup opsi tersebut. Untuk informasi lebih lanjut, lihat Menghapus opsi dari grup opsi.
- Ubah instans DB dan tentukan grup opsi lain yang tidak menyertakan OEM Agent tersebut. Perubahan ini memengaruhi satu instans DB. Anda dapat menentukan grup opsi default (kosong) atau grup opsi kustom lain. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Keamanan Label Oracle

Amazon RDS mendukung Keamanan Label Oracle untuk Oracle Database Enterprise Edition melalui penggunaan opsi OLS.

Sebagian besar keamanan basis data mengontrol akses di tingkat objek. Keamanan Label Oracle menyediakan kontrol ketat untuk akses ke setiap baris tabel. Misalnya, Anda dapat menggunakan Keamanan Label untuk menerapkan kepatuhan peraturan dengan model administrasi berbasis kebijakan. Anda dapat menggunakan kebijakan Keamanan Label untuk mengontrol akses ke data sensitif, dan hanya memberikan akses kepada pengguna dengan tingkat izin yang sesuai. Untuk informasi selengkapnya, lihat Introduction to Oracle Label Security dalam dokumentasi Oracle.

Topik

Prasyarat untuk Keamanan Label Oracle

Pahami prasyarat berikut untuk Keamanan Label Oracle:

- Instans DB Anda harus menggunakan model Bawa Lisensi Anda Sendiri (BYOL). Untuk informasi selengkapnya, lihat Opsi lisensi RDS for Oracle.
- Anda harus memiliki lisensi Oracle Enterprise Edition yang valid dengan Lisensi dan Dukungan Pembaruan Perangkat Lunak.
- Lisensi Oracle Anda harus menyertakan opsi Keamanan Label.
- Anda harus menggunakan arsitektur basis data non-multi-penghuni (non-CDB). Untuk informasi selengkapnya, lihat Konfigurasi satu penghuni pada arsitektur CDB.

Menambahkan opsi Keamanan Label Oracle

Berikut adalah proses umum untuk menambahkan opsi Keamanan Label Oracle ke instans DB:

1. Buat grup opsi baru, atau salin atau ubah grup opsi yang ada.

2. Tambahkan opsi tersebut ke grup opsi.

> **Important**
>
> Keamanan Label Oracle adalah opsi permanen dan tetap.

3. Kaitkan grup opsi tersebut dengan instans DB.

Setelah Anda menambahkan opsi Keamanan Label, segera setelah grup opsi aktif, Keamanan Label juga aktif.

Untuk menambahkan opsi keamanan label ke instans DB

1. Tentukan grup opsi yang ingin Anda gunakan. Anda dapat membuat grup opsi baru atau menggunakan grup opsi yang ada. Jika Anda ingin menggunakan grup opsi yang ada, lanjutkan ke langkah berikutnya. Jika tidak, buat grup opsi DB kustom dengan pengaturan berikut:

   a. Untuk Mesin , pilih oracle-ee.

   b. Untuk Versi mesin utama, pilih versi instans DB Anda.

   Untuk informasi selengkapnya, lihat Membuat grup opsi.

2. Tambahkan opsi OLS ke grup opsi. Untuk informasi selengkapnya tentang cara menambahkan opsi, lihat Menambahkan opsi ke grup opsi.

> **Important**
>
> Jika Anda menambahkan Keamanan Label ke grup opsi yang sudah ada dan terpasang ke satu instans DB atau lebih, semua instans DB dimulai ulang.

3. Terapkan grup opsi ke instans DB baru atau yang sudah ada:

   - Untuk instans DB baru, Anda menerapkan grup opsi saat Anda meluncurkan instans. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.

   - Untuk instans DB yang ada, Anda menerapkan grup opsi dengan memodifikasi instans dan melampirkan grup opsi baru. Ketika Anda menambahkan opsi Label Keamanan ke instans DB yang ada, pemadaman singkat akan terjadi selagi instans DB Anda dimulai ulang secara otomatis. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Menggunakan Keamanan Label Oracle

Untuk menggunakan Keamanan Label Oracle, Anda membuat kebijakan yang mengontrol akses ke baris tertentu di tabel Anda. Untuk informasi lebih lanjut, lihat Creating an Oracle Label Security policy dalam dokumentasi Oracle.

Saat Anda menggunakan Keamanan Label, Anda melakukan semua tindakan sebagai peran LBAC_DBA. Pengguna utama untuk instans DB Anda diberi peran LBAC_DBA. Anda dapat memberikan peran LBAC_DBA kepada pengguna lain agar mereka dapat mengelola kebijakan Keamanan Label.

Untuk rilis berikut, berikan akses ke paket `OLS_ENFORCEMENT` untuk setiap pengguna baru yang memerlukan akses ke Keamanan Label Oracle:

- Oracle Database 19c yang menggunakan arsitektur non-CDB
- Oracle Database 12c Rilis 2 (12.2)

Untuk memberikan akses ke paket `OLS_ENFORCEMENT`, sambungkan ke instans DB sebagai pengguna utama dan jalankan pernyataan SQL berikut:

```
GRANT ALL ON LBACSYS.OLS_ENFORCEMENT TO username;
```

Anda dapat mengonfigurasi Keamanan Label melalui Oracle Enterprise Manager (OEM) Cloud Control. Amazon RDS mendukung Kontrol Cloud OEM melalui opsi Management Agent. Untuk informasi selengkapnya, lihat Oracle Management Agent untuk Kontrol Cloud Enterprise Manager.

Menghapus opsi Keamanan Label Oracle (tidak didukung)

Mulai dari Oracle Database 12c Rilis 2 (12.2), Keamanan Label Oracle adalah opsi permanen dan tetap. Karena opsi ini permanen, Anda tidak dapat menghapusnya dari grup opsi. Jika Anda menambahkan Keamanan Label Oracle ke grup opsi dan mengaitkannya dengan instans DB Anda, nantinya Anda dapat mengaitkan grup opsi yang berbeda dengan instans DB Anda, tetapi grup ini juga harus berisi opsi Keamanan Label Oracle.

Pemecahan Masalah

Masalah berikut ini mungkin Anda temui ketika menggunakan Keamanan Label Oracle.

Masalah	Saran pemecahan masalah
Ketika Anda mencoba membuat kebijakan, Anda melihat pesan kesalahan yang mirip dengan yang berikut ini: `insufficient authorization for the SYSDBA package` .	Masalah umum dengan fitur Keamanan Label Oracle membuat pengguna dengan nama pengguna berisi 16 atau 24 karakter tidak dapat menjalankan perintah Keamanan Label. Anda dapat membuat pengguna baru dengan jumlah karakter yang berbeda, memberikan LBAC_DBA kepada pengguna baru tersebut, masuk sebagai pengguna baru, dan menjalankan perintah OLS sebagai pengguna baru. Untuk informasi lainnya, silakan hubungi dukungan Oracle.

Oracle Locator

Amazon RDS mendukung Oracle Locator melalui penggunaan opsi `LOCATOR`. Oracle Locator menyediakan kemampuan yang biasanya dibutuhkan untuk mendukung aplikasi berbasis layanan nirkabel dan internet serta solusi GIS berbasis mitra. Oracle Locator adalah subset terbatas dari Oracle Spatial. Untuk informasi lebih lanjut, lihat Oracle Locator dalam dokumentasi Oracle.

> **Important**
>
> Jika Anda menggunakan Oracle Locator, Amazon RDS secara otomatis memperbarui instans DB Anda ke Oracle PSU terbaru jika ada kerentanan keamanan dengan skor Common Vulnerability Scoring System (CVSS) 9+ atau kerentanan keamanan lain yang diumumkan.

Amazon RDS mendukung Oracle Locator untuk rilis Oracle Database berikut ini:

- Oracle Database 19c (19.0.0.0)
- Oracle Database 12c Rilis 2 (12.2.0.1)
- Oracle Database 12c Rilis 1 (12.1), versi 12.1.0.2.v13 atau yang lebih baru

Oracle Locator tidak didukung untuk Oracle Database 21c, tetapi fungsinya tersedia dalam opsi Oracle Spatial. Sebelumnya, opsi Spatial memerlukan lisensi tambahan. Oracle Locator mewakili subset fitur Oracle Spatial dan tidak memerlukan lisensi tambahan. Pada tahun 2019, Oracle mengumumkan bahwa semua fitur Oracle Spatial dimasukkan dalam lisensi Enterprise Edition dan Standard Edition 2 tanpa biaya tambahan. Oleh karena itu, opsi Oracle Spatial tidak lagi memerlukan lisensi tambahan.

Dimulai dengan Oracle Database 21c, opsi Oracle Locator tidak lagi didukung. Untuk menggunakan fitur Oracle Locator di Oracle Database 21c, instal opsi Oracle Spatial sebagai gantinya. Untuk informasi selengkapnya, lihat Machine Learning, Spatial and Graph - No License Required! di blog Oracle Database Insider.

Prasyarat untuk Oracle Locator

Berikut adalah prasyarat dalam penggunaan Oracle Locator:

- Instans DB Anda harus memiliki kelas yang memadai. Oracle Java tidak didukung untuk kelas instans DB db.t3.micro atau db.t3.small. Untuk informasi selengkapnya, lihat Kelas instans RDS for Oracle.
- Instans DB Anda harus mengaktifkan Peningkatan Versi Minor Otomatis. Opsi ini memungkinkan instans DB Anda menerima peningkatan versi mesin DB minor secara otomatis ketika tersedia dan diperlukan untuk opsi apa pun yang menginstal Oracle Java Virtual Machine (JVM). Amazon RDS menggunakan opsi ini untuk memperbarui instans DB Anda ke Oracle Patch Set Update (PSU) atau Release Update (RU) terbaru. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Praktik terbaik untuk Oracle Locator

Berikut adalah praktik terbaik dalam penggunaan Oracle Locator:

- Untuk keamanan maksimal, gunakan opsi `LOCATOR` dengan Secure Sockets Layer (SSL). Untuk informasi selengkapnya, lihat Lapisan Soket Aman Oracle.
- Konfigurasikan instans DB Anda untuk membatasi akses ke instans DB Anda. Untuk informasi lebih lanjut, lihat Skenario untuk mengakses instans DB di VPC dan Bekerja dengan klaster DB dalam VPC.

Menambahkan opsi Oracle Locator

Berikut adalah proses umum untuk menambahkan opsi `LOCATOR` ke instans DB:

1. Buat grup opsi baru, atau salin atau ubah grup opsi yang ada.
2. Tambahkan opsi tersebut ke grup opsi.
3. Kaitkan grup opsi tersebut dengan instans DB.

Jika Oracle Java Virtual Machine (JVM) tidak diinstal pada instans DB, akan terjadi pemadaman singkat saat opsi `LOCATOR` ditambahkan. Pemadaman tidak terjadi jika Oracle Java Virtual Machine (JVM) sudah diinstal pada instans DB. Setelah opsi ditambahkan, instans DB tidak perlu dimulai ulang. Setelah grup opsi aktif, Oracle Locator akan langsung tersedia.

> **Note**
>
> Selama pemadaman ini, fungsi verifikasi kata sandi akan dinonaktifkan sementara. Anda juga dapat melihat peristiwa terkait fungsi verifikasi kata sandi selama terjadi pemadaman. Fungsi verifikasi kata sandi akan diaktifkan kembali sebelum instans Oracle DB tersedia.

Untuk menambahkan opsi **`LOCATOR`** ke instans DB

1. Tentukan grup opsi yang ingin Anda gunakan. Anda dapat membuat grup opsi baru atau menggunakan grup opsi yang ada. Jika Anda ingin menggunakan grup opsi yang ada, lanjutkan ke langkah berikutnya. Jika tidak, buat grup opsi DB kustom dengan pengaturan berikut:

   a. Untuk Mesin, pilih edisi Oracle untuk instans DB Anda.

   b. Untuk Versi mesin utama, pilih versi instans DB Anda.

   Untuk informasi lebih lanjut, lihat Membuat grup opsi.
2. Tambahkan opsi LOCATOR ke grup opsi. Untuk informasi selengkapnya tentang cara menambahkan opsi, lihat Menambahkan opsi ke grup opsi.
3. Terapkan grup opsi ke instans DB baru atau yang sudah ada:

   - Untuk instans DB baru, Anda menerapkan grup opsi saat Anda meluncurkan instans. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.
   - Untuk instans DB yang ada, Anda menerapkan grup opsi dengan memodifikasi instans dan melampirkan grup opsi baru. Untuk informasi lebih lanjut, lihat Memodifikasi instans DB Amazon RDS.

Menggunakan Oracle Locator

Setelah Anda mengaktifkan opsi Oracle Locator, Anda dapat mulai menggunakannya. Anda sebaiknya hanya menggunakan fitur Oracle Locator. Jangan menggunakan fitur Oracle Spatial kecuali Anda memiliki lisensi untuk Oracle Spatial.

Untuk daftar fitur yang didukung untuk Oracle Locator, lihat Features Included with Locator dalam dokumentasi Oracle.

Untuk daftar fitur yang tidak didukung untuk Oracle Locator, lihat Features Not Included with Locator dalam dokumentasi Oracle.

Menghapus opsi Oracle Locator

Setelah Anda membatalkan semua objek yang menggunakan jenis data yang disediakan oleh opsi LOCATOR, Anda dapat menghapus opsi tersebut dari instans DB. Jika Oracle Java Virtual Machine (JVM) tidak diinstal pada instans DB, akan terjadi pemadaman singkat saat opsi LOCATOR dihapus. Pemadaman tidak terjadi jika Oracle Java Virtual Machine (JVM) sudah diinstal pada instans DB. Setelah opsi LOCATOR dihapus, instans DB tidak perlu dimulai ulang.

Untuk membatalkan opsi **LOCATOR**

1. Cadangkan data Anda.

   > **Warning**
   >
   > Jika instans menggunakan jenis data yang diaktifkan sebagai bagian dari opsi, dan jika Anda menghapus opsi LOCATOR, data Anda bisa hilang. Untuk informasi selengkapnya, lihat Mencadangkan, memulihkan, dan mengekspor data.

2. Periksa apakah ada jenis data referensi objek atau fitur dari opsi LOCATOR.

   Jika opsi LOCATOR ada, instans bisa macet saat menerapkan grup opsi baru yang tidak memiliki opsi LOCATOR. Anda dapat mengidentifikasi objek menggunakan kueri berikut ini:

```
SELECT OWNER, SEGMENT_NAME, TABLESPACE_NAME, BYTES/1024/1024 mbytes
FROM   DBA_SEGMENTS
WHERE  SEGMENT_TYPE LIKE '%TABLE%'
AND    (OWNER, SEGMENT_NAME) IN
       (SELECT DISTINCT OWNER, TABLE_NAME
        FROM   DBA_TAB_COLUMNS
        WHERE  DATA_TYPE='SDO_GEOMETRY'
        AND    OWNER <> 'MDSYS')
ORDER BY 1,2,3,4;

SELECT OWNER, TABLE_NAME, COLUMN_NAME
FROM   DBA_TAB_COLUMNS
WHERE  DATA_TYPE = 'SDO_GEOMETRY'
AND    OWNER <> 'MDSYS'
ORDER BY 1,2,3;
```

3. Batalkan objek apa pun yang mereferensikan jenis data atau fitur opsi `LOCATOR`.
4. Lakukan salah satu dari berikut ini:

   - Hapus opsi `LOCATOR` dari grup opsi asalnya. Perubahan ini akan memengaruhi semua instans DB yang menggunakan grup opsi. Untuk informasi selengkapnya, lihat Menghapus opsi dari grup opsi.
   - Ubah instans DB dan tentukan grup opsi lain yang tidak menyertakan opsi `LOCATOR`. Perubahan ini memengaruhi instans DB tunggal. Anda dapat menentukan grup opsi default (kosong) atau grup opsi kustom lain. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Oracle Multimedia

Amazon RDS mendukung Oracle Multimedia melalui penggunaan opsi `MULTIMEDIA`. Anda dapat menggunakan Oracle Multimedia untuk menyimpan, mengelola, dan mengambil gambar, audio, video, serta data media heterogen lainnya. Untuk informasi lebih lanjut, lihat Oracle Multimedia dalam dokumentasi Oracle.

> **Important**
>
> Jika Anda menggunakan Oracle Multimedia, Amazon RDS secara otomatis memperbarui instans DB Anda ke Oracle PSU terbaru jika ada kerentanan keamanan dengan skor Common Vulnerability Scoring System (CVSS) 9+ atau kerentanan keamanan lain yang diumumkan.

Amazon RDS mendukung Oracle Multimedia untuk semua edisi versi berikut:

- Oracle Database 12c Rilis 2 (12.2)
- Oracle Database 12c Rilis 1 (12.1), versi 12.1.0.2.v13 atau yang lebih tinggi

> **Note**
>
> Oracle tidak mendukung Oracle Multimedia di Oracle Database 19c. Jadi, Oracle Multimedia tidak didukung untuk instans DB Oracle Database 19c. Untuk informasi lebih lanjut, lihat Desupport of Oracle Multimedia dalam dokumentasi Oracle.

Prasyarat untuk Oracle Multimedia

Berikut adalah prasyarat untuk menggunakan Oracle Multimedia:

- Instans DB Anda harus memiliki kelas yang memadai. Oracle Multimedia tidak didukung untuk kelas instans DB db.t3.micro atau db.t3.small. Untuk informasi selengkapnya, lihat Kelas instans RDS for Oracle.
- Instans DB Anda harus mengaktifkan Peningkatan Versi Minor Otomatis. Opsi ini memungkinkan instans DB Anda menerima peningkatan versi mesin DB minor secara otomatis ketika tersedia dan diperlukan untuk opsi apa pun yang menginstal Oracle Java Virtual Machine (JVM). Amazon RDS

menggunakan opsi ini untuk memperbarui instans DB Anda ke Oracle Patch Set Update (PSU) atau Release Update (RU) terbaru. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Praktik terbaik untuk Oracle Multimedia

Berikut adalah praktik terbaik dalam penggunaan Oracle Multimedia:

- Untuk keamanan maksimal, gunakan opsi `MULTIMEDIA` dengan Secure Sockets Layer (SSL). Untuk informasi selengkapnya, lihat Lapisan Soket Aman Oracle.
- Konfigurasikan instans DB Anda untuk membatasi akses ke instans DB Anda. Untuk informasi selengkapnya, lihat Skenario untuk mengakses instans DB di VPC dan Bekerja dengan klaster DB dalam VPC.

Menambahkan opsi Oracle Multimedia

Berikut adalah proses umum untuk menambahkan opsi `MULTIMEDIA` ke instans DB:

1. Buat grup opsi baru, atau salin atau ubah grup opsi yang ada.
2. Tambahkan opsi tersebut ke grup opsi.
3. Kaitkan grup opsi tersebut dengan instans DB.

Jika Oracle Java Virtual Machine (JVM) tidak diinstal pada instans DB, akan terjadi pemadaman singkat saat opsi `MULTIMEDIA` ditambahkan. Pemadaman tidak terjadi jika Oracle Java Virtual Machine (JVM) sudah diinstal pada instans DB. Setelah opsi ditambahkan, instans DB tidak perlu dimulai ulang. Setelah grup opsi aktif, Oracle Multimedia akan langsung tersedia.

> **Note**
>
> Selama pemadaman ini, fungsi verifikasi kata sandi akan dinonaktifkan sementara. Anda juga dapat melihat peristiwa terkait fungsi verifikasi kata sandi selama terjadi pemadaman. Fungsi verifikasi kata sandi akan diaktifkan kembali sebelum instans Oracle DB tersedia.

Untuk menambahkan opsi **`MULTIMEDIA`** ke instans DB

1. Tentukan grup opsi yang ingin Anda gunakan. Anda dapat membuat grup opsi baru atau menggunakan grup opsi yang ada. Jika Anda ingin menggunakan grup opsi yang ada, lanjutkan ke langkah berikutnya. Jika tidak, buat grup opsi DB kustom dengan pengaturan berikut:

   a. Untuk Mesin, pilih edisi untuk instans DB Oracle Anda.

   b. Untuk Versi mesin utama, pilih versi instans DB Anda.

   Untuk informasi selengkapnya, lihat Membuat grup opsi.
2. Tambahkan opsi MULTIMEDIA ke grup opsi. Untuk informasi selengkapnya tentang cara menambahkan opsi, lihat Menambahkan opsi ke grup opsi.
3. Terapkan grup opsi ke instans DB baru atau yang sudah ada:

   - Untuk instans DB baru, Anda menerapkan grup opsi saat Anda meluncurkan instans. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.
   - Untuk instans DB yang ada, Anda menerapkan grup opsi dengan memodifikasi instans dan melampirkan grup opsi baru. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Menghapus opsi Oracle Multimedia

Setelah Anda melepaskan semua objek yang menggunakan jenis data yang disediakan oleh opsi `MULTIMEDIA`, Anda dapat menghapus opsi tersebut dari instans DB. Jika Oracle Java Virtual Machine (JVM) tidak diinstal pada instans DB, akan terjadi gangguan singkat saat opsi `MULTIMEDIA` dihapus. Pemadaman tidak terjadi jika Oracle Java Virtual Machine (JVM) sudah diinstal pada instans DB. Setelah opsi `MULTIMEDIA` dihapus, instans DB tidak perlu dimulai ulang.

Untuk melepaskan opsi **`MULTIMEDIA`**

1. Cadangkan data Anda.

   > **Warning**
   >
   > Jika instans menggunakan jenis data yang diaktifkan sebagai bagian dari opsi, dan jika Anda menghapus opsi `MULTIMEDIA`, data Anda bisa hilang. Untuk informasi selengkapnya, lihat Mencadangkan, memulihkan, dan mengekspor data.

2. Periksa apakah ada jenis data referensi objek atau fitur dari opsi `MULTIMEDIA`.
3. Batalkan objek apa pun yang mereferensikan jenis data atau fitur opsi `MULTIMEDIA`.
4. Lakukan salah satu langkah berikut:

   - Hapus opsi `MULTIMEDIA` dari grup opsi asalnya. Perubahan ini memengaruhi semua instans DB yang menggunakan grup opsi tersebut. Untuk informasi selengkapnya, lihat Menghapus opsi dari grup opsi.
   - Ubah instans DB dan tentukan grup opsi lain yang tidak menyertakan opsi `MULTIMEDIA`. Perubahan ini memengaruhi instans DB tunggal. Anda dapat menentukan grup opsi default (kosong) atau grup opsi kustom yang berbeda. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Enkripsi jaringan asli Oracle

Amazon RDS mendukung enkripsi jaringan asli (NNE) Oracle. Dengan enkripsi jaringan asli, Anda dapat mengenkripsi data saat berpindah ke dan dari instans DB. Amazon RDS mendukung NNE untuk semua edisi Oracle Database.

Diskusi mendetail tentang enkripsi jaringan asli Oracle berada di luar cakupan panduan ini, tetapi sebaiknya pahami kekuatan dan kelemahan setiap algoritma dan kunci sebelum Anda memutuskan solusi untuk deployment Anda. Untuk informasi tentang algoritma dan kunci yang tersedia melalui enkripsi jaringan asli Oracle, lihat Configuring network data encryption di dokumentasi Oracle. Untuk informasi lebih lanjut tentang keamanan AWS, lihat pusat keamanan AWS.

> **Note**
>
> Anda dapat menggunakan Enkripsi Jaringan Asli atau Lapisan Soket Aman, tetapi tidak keduanya sekaligus. Untuk informasi selengkapnya, lihat Lapisan Soket Aman Oracle.

Pengaturan opsi NNE

Anda dapat menentukan persyaratan enkripsi pada server dan klien. Instans DB dapat bertindak sebagai klien ketika, misalnya, menggunakan tautan basis data untuk terhubung ke basis data lain. Sebaiknya hindari enkripsi paksa di sisi server. Misalnya, sebaiknya jangan memaksa semua komunikasi klien untuk menggunakan enkripsi karena server memerlukannya. Dalam hal ini, Anda dapat memaksa enkripsi di sisi klien untuk menggunakan opsi `SQLNET.*CLIENT`.

Amazon RDS mendukung pengaturan berikut untuk opsi NNE.

> **Note**
>
> Saat Anda menggunakan koma untuk memisahkan nilai untuk pengaturan opsi, jangan beri spasi setelah koma.

Pengaturan opsi	Nilai valid	Nilai default	Deskripsi
`SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS`	`TRUE`, `FALSE`	`TRUE`	Perilaku server ketika klien yang menggunakan cipher yang tidak aman

Pengaturan opsi	Nilai valid	Nilai default	Deskripsi
			mencoba untuk terhubung ke basis data. Jika TRUE, klien dapat terhubung meskipun klien tidak di-patch dengan PSU Juli 2021. Jika pengaturannya FALSE, klien dapat terhubung ke basis data hanya ketika klien di-patch dengan PSU Juli 2021. Sebelum Anda menetapkan SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS ke FALSE, pastikan bahwa kondisi berikut terpenuhi: • SQLNET.ENCRYPTION_TYPES_SERVER dan SQLNET.ENCRYPTION_TYPES_CLIENT memiliki satu metode enkripsi yang cocok yang bukan DES, 3DES, atau RC4 (semua panjang kunci). • SQLNET.CHECKSUM_TYPES_SERVER dan SQLNET.CHECKSUM_TYPES_CLIENT memiliki satu metode checksumming aman yang cocok yang bukan MD5. • Klien di-patch dengan PSU Juli 2021. Jika klien tidak di-patch, klien kehilangan koneksi dan menerima kesalahan ORA-12269 .

Pengaturan opsi	Nilai valid	Nilai default	Deskripsi
SQLNET.ALLOW_WEAK_CRYPTO	TRUE, FALSE	TRUE	Perilaku server ketika klien yang menggunakan cipher yang tidak aman mencoba untuk terhubung ke basis data. Cipher berikut dianggap tidak aman: • Metode enkripsi DES (semua panjang kunci) • Metode enkripsi 3DES (semua panjang kunci) • Metode enkripsi RC4 (semua panjang kunci) • Metode checksumming MD5 Jika pengaturannya TRUE, klien dapat terhubung ketika klien menggunakan cipher yang tidak aman sebelumnya. Jika pengaturannya FALSE, basis data mencegah klien terhubung ketika klien menggunakan cipher yang tidak aman sebelumnya. Sebelum Anda menetapkan SQLNET.ALLOW_WEAK_CRYPTO ke FALSE, pastikan bahwa kondisi berikut terpenuhi: • SQLNET.ENCRYPTION_TYPES_SERVER dan SQLNET.ENCRYPTION_TYPES_CLIENT memiliki satu metode enkripsi yang cocok yang bukan DES, 3DES, atau RC4 (semua panjang kunci).

Pengaturan opsi	Nilai valid	Nilai default	Deskripsi
			• SQLNET.CHECKSUM_TYPES_SERVER dan SQLNET.CHECKSUM_TYPES_CLIENT memiliki satu metode checksumming aman yang cocok yang bukan MD5. • Klien di-patch dengan PSU Juli 2021. Jika klien tidak di-patch, klien kehilangan koneksi dan menerima kesalahan ORA-12269 .
SQLNET.CRYPTO_CHECKSUM_CLIENT	Accepted Rejected Requested , Required	Requested	Perilaku integritas data ketika instans DB terhubung ke klien atau server yang bertindak sebagai klien. Ketika instans DB menggunakan tautan basis data, instans akan bertindak sebagai klien. Requested menunjukkan bahwa klien tidak memerlukan instans DB untuk melakukan checksum.
SQLNET.CRYPTO_CHECKSUM_SERVER	Accepted Rejected Requested , Required	Requested	Perilaku integritas data ketika klien atau server yang bertindak sebagai klien terhubung ke instans DB. Ketika instans DB menggunakan tautan basis data, instans akan bertindak sebagai klien. Requested menunjukkan bahwa instans DB tidak memerlukan klien untuk melakukan checksum.

Pengaturan opsi	Nilai valid	Nilai default	Deskripsi
SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT	SHA256, SHA384, SHA512, SHA1, MD5	SHA256, SHA384, SHA512	Daftar algoritma checksum. Anda dapat menentukan salah satu nilai atau daftar nilai yang dipisahkan koma. Jika Anda menggunakan koma, jangan masukkan spasi setelah koma; jika dilakukan, Anda akan menerima kesalahan InvalidParameterValue . Parameter ini dan SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER harus memiliki cipher umum.
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER	SHA256, SHA384, SHA512, SHA1, MD5	SHA256, SHA384, SHA512, SHA1, MD5	Daftar algoritma checksum. Anda dapat menentukan salah satu nilai atau daftar nilai yang dipisahkan koma. Jika Anda menggunakan koma, jangan masukkan spasi setelah koma; jika dilakukan, Anda akan menerima kesalahan InvalidParameterValue . Parameter ini dan SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT harus memiliki cipher umum.

Pengaturan opsi	Nilai valid	Nilai default	Deskripsi
SQLNET.ENCRYPTION_CLIENT	Accepted Rejected Requested , Required	Requested	Perilaku enkripsi klien saat klien atau server yang bertindak sebagai klien terhubung ke instans DB. Ketika instans DB menggunakan tautan basis data, instans akan bertindak sebagai klien. Requested menunjukkan bahwa klien tidak memerlukan lalu lintas dari server untuk dienkripsi.
SQLNET.ENCRYPTION_SERVER	Accepted Rejected Requested , Required	Requested	Perilaku enkripsi server ketika klien atau server yang bertindak sebagai klien terhubung ke instans DB. Ketika instans DB menggunakan tautan basis data, instans akan bertindak sebagai klien. Requested menunjukkan bahwa instans DB tidak memerlukan lalu lintas dari klien untuk dienkripsi.

Pengaturan opsi	Nilai valid	Nilai default	Deskripsi
SQLNET.ENCRYPTION_TYPES_CLIENT	RC4_256, AES256, AES192, 3DES168, RC4_128, AES128, 3DES112, RC4_56, DES, RC4_40, DES40	RC4_256, AES256, AES192, 3DES168, RC4_128, AES128, 3DES112, RC4_56, DES, RC4_40, DES40	Daftar algoritma enkripsi yang digunakan oleh klien. Klien mencoba mendekripsi input server dengan mencoba setiap algoritma secara berurutan, yang terus berlanjut hingga algoritma berhasil atau akhir daftar tercapai. Amazon RDS menggunakan daftar default berikut dari Oracle. RDS dimulai dengan RC4_256 dan berlanjut hingga akhir daftar secara berurutan . Anda dapat mengubah urutan atau membatasi algoritma yang akan diterima oleh instans DB. 1. RC4_256: RSA RC4 (ukuran kunci 256-bit) 2. AES256: AES (ukuran kunci 256-bit) 3. AES192: AES (ukuran kunci 192-bit) 4. 3DES168: 3 kunci Triple-DES (ukuran kunci efektif 112-bit) 5. RC4_128: RSA RC4 (ukuran kunci 128-bit) 6. AES128: AES (ukuran kunci 128-bit) 7. 3DES112: 2 kunci Triple-DES (ukuran kunci efektif 80-bit) 8. RC4_56: RSA RC4 (ukuran kunci 56-bit) 9. DES: DES Standar (ukuran kunci 56-bit)

Pengaturan opsi	Nilai valid	Nilai default	Deskripsi
			10`RC4_40`: RSA RC4 (ukuran kunci 40-bit) 11`DES40`: DES40 (ukuran kunci 40-bit) Anda dapat menentukan salah satu nilai atau daftar nilai yang dipisahkan koma. Jika Anda menggunakan koma, jangan masukkan spasi setelah koma; jika dilakukan, Anda akan menerima kesalahan `InvalidParameterValue` . Parameter ini dan `SQLNET.SQLNET.ENCRYPTION_TYPES_SERVER` harus memiliki cipher umum.

Pengaturan opsi	Nilai valid	Nilai default	Deskripsi
SQLNET.ENCRYPTION_TYPES_SERVER	RC4_256, AES256, AES192, 3DES168, RC4_128, AES128, 3DES112, RC4_56, DES, RC4_40, DES40	RC4_256, AES256, AES192, 3DES168, RC4_128, AES128, 3DES112, RC4_56, DES, RC4_40, DES40	Daftar algoritma enkripsi yang digunakan oleh instans DB. Instans DB menggunakan setiap algoritma untuk mencoba mendekripsi input klien hingga algoritma berhasil atau hingga akhir daftar tercapai. Amazon RDS menggunakan daftar default berikut dari Oracle. Anda dapat mengubah urutan atau membatasi algoritma yang akan diterima klien. 1. RC4_256: RSA RC4 (ukuran kunci 256-bit) 2. AES256: AES (ukuran kunci 256-bit) 3. AES192: AES (ukuran kunci 192-bit) 4. 3DES168: 3 kunci Triple-DES (ukuran kunci efektif 112-bit) 5. RC4_128: RSA RC4 (ukuran kunci 128-bit) 6. AES128: AES (ukuran kunci 128-bit) 7. 3DES112: 2 kunci Triple-DES (ukuran kunci efektif 80-bit) 8. RC4_56: RSA RC4 (ukuran kunci 56-bit) 9. DES: DES Standar (ukuran kunci 56-bit) 10. RC4_40: RSA RC4 (ukuran kunci 40-bit) 11. DES40: DES40 (ukuran kunci 40-bit)

Pengaturan opsi	Nilai valid	Nilai default	Deskripsi
			Anda dapat menentukan salah satu nilai atau daftar nilai yang dipisahkan koma. Jika Anda menggunakan koma, jangan masukkan spasi setelah koma; jika dilakukan, Anda akan menerima kesalahan `InvalidParameterValue` . Parameter ini dan `SQLNET.SQLNET.ENCRYPTION_TYPES_SERVER` harus memiliki cipher umum.

Menambahkan opsi NNE

Proses umum untuk menambahkan opsi NNE ke instans DB adalah sebagai berikut:

1. Buat grup opsi baru, atau salin atau ubah grup opsi yang ada.
2. Tambahkan opsi ke grup opsi.
3. Kaitkan grup opsi dengan instans DB.

Saat grup opsi aktif, NNE akan aktif.

Untuk menambahkan opsi NNE ke instans DB menggunakan AWS Management Console

1. Untuk Mesin, pilih edisi Oracle yang ingin Anda gunakan. NNE didukung di semua edisi.
2. Untuk Versi mesin utama, pilih versi instans DB Anda.

   Untuk informasi selengkapnya, lihat Membuat grup opsi.
3. Tambahkan opsi NNE ke grup opsi. Untuk informasi selengkapnya tentang cara menambahkan opsi, lihat Menambahkan opsi ke grup opsi.

> **Note**
>
> Setelah Anda menambahkan opsi NNE, Anda tidak perlu memulai ulang instans DB Anda. Begitu grup opsi aktif, NNE akan aktif.

4. Terapkan grup opsi ke instans DB baru atau yang sudah ada:

   - Untuk instans DB baru, Anda menerapkan grup opsi saat Anda meluncurkan instans. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.
   - Untuk instans DB yang ada, Anda menerapkan grup opsi dengan memodifikasi instans dan melampirkan grup opsi baru. Setelah Anda menambahkan opsi NNE, Anda tidak perlu memulai ulang instans DB Anda. Begitu grup opsi aktif, NNE akan aktif. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Mengatur nilai NNE di sqlnet.ora

Dengan enkripsi jaringan asli Oracle, Anda dapat mengatur enkripsi jaringan di sisi server dan sisi klien. Klien adalah komputer yang digunakan untuk terhubung ke instans DB. Anda dapat menentukan pengaturan klien berikut di sqlnet.ora:

- `SQLNET.ALLOW_WEAK_CRYPTO`
- `SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS`
- `SQLNET.CRYPTO_CHECKSUM_CLIENT`
- `SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT`
- `SQLNET.ENCRYPTION_CLIENT`
- `SQLNET.ENCRYPTION_TYPES_CLIENT`

Untuk mengetahui informasinya, lihat Configuring network data encryption and integrity for Oracle servers and clients di dokumentasi Oracle.

Terkadang, instans DB menolak permintaan koneksi dari aplikasi. Misalnya, penolakan dapat terjadi ketika algoritma enkripsi pada klien dan di server tidak cocok. Untuk menguji enkripsi jaringan asli Oracle, tambahkan baris berikut ke file sqlnet.ora di klien:

```
DIAG_ADR_ENABLED=off
```

```
TRACE_DIRECTORY_CLIENT=/tmp
TRACE_FILE_CLIENT=nettrace
TRACE_LEVEL_CLIENT=16
```

Ketika koneksi dicoba, baris sebelumnya menghasilkan file jejak pada klien yang disebut `/tmp/nettrace*`. File jejak berisi informasi tentang koneksi. Untuk informasi selengkapnya tentang masalah terkait koneksi saat Anda menggunakan Oracle Native Network Encryption, lihat About negotiating encryption and integrity dalam dokumentasi Basis Data Oracle.

Mengubah pengaturan opsi NNE

Setelah Anda mengaktifkan NNE, Anda dapat mengubah pengaturannya. Saat ini, Anda dapat mengubah pengaturan opsi NNE hanya dengan AWS CLI atau RDS API. Anda tidak dapat menggunakan konsol. Untuk mempelajari cara mengubah pengaturan opsi menggunakan CLI, lihat AWS CLI. Untuk informasi selengkapnya tentang setiap pengaturan, lihat Pengaturan opsi NNE.

Topik

- Mengubah nilai CRYPTO_CHECKSUM_*
- Mengubah pengaturan ALLOW_WEAK_CRYPTO*

Mengubah nilai CRYPTO_CHECKSUM_*

Jika Anda mengubah pengaturan opsi NNE, pastikan bahwa pengaturan opsi berikut memiliki setidaknya satu chiper umum:

- `SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER`
- `SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT`

Contoh berikut menunjukkan skenario di mana Anda mengubah `SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER`. Konfigurasi ini valid karena `CRYPTO_CHECKSUM_TYPES_CLIENT` dan `CRYPTO_CHECKSUM_TYPES_SERVER` sama-sama menggunakan `SHA256`.

Pengaturan opsi	Nilai sebelum modifikasi	Nilai setelah modifikasi
`SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT`	**`SHA256`**, `SHA384`, `SHA512`	Tidak ada perubahan

Pengaturan opsi	Nilai sebelum modifikasi	Nilai setelah modifikasi
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER	**SHA256**, SHA384, SHA512, SHA1, MD5	SHA1,MD5, **SHA256**

Untuk contoh lain, asumsikan bahwa Anda ingin mengubah SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER dari pengaturan defaultnya ke SHA1,MD5. Dalam hal ini, pastikan Anda menetapkan SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT ke SHA1 atau MD5. Algoritma ini tidak termasuk dalam nilai default untuk SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT.

Mengubah pengaturan ALLOW_WEAK_CRYPTO*

Untuk mengatur opsi SQLNET.ALLOW_WEAK_CRYPTO* dari nilai default ke FALSE, pastikan bahwa kondisi berikut terpenuhi:

- SQLNET.ENCRYPTION_TYPES_SERVER dan SQLNET.ENCRYPTION_TYPES_CLIENT memiliki satu metode enkripsi aman yang cocok. Sebuah metode dianggap aman jika bukan DES, 3DES, atau RC4 (semua panjang kunci).
- SQLNET.CHECKSUM_TYPES_SERVER dan SQLNET.CHECKSUM_TYPES_CLIENT memiliki satu metode checksumming aman yang cocok. Sebuah metode dianggap aman jika bukan MD5.
- Klien di-patch dengan PSU Juli 2021. Jika klien tidak di-patch, klien kehilangan koneksi dan menerima kesalahan ORA-12269.

Contoh berikut menunjukkan pengaturan NNE. Asumsikan bahwa Anda ingin menetapkan SQLNET.ENCRYPTION_TYPES_SERVER dan SQLNET.ENCRYPTION_TYPES_CLIENT ke FALSE, sehingga memblokir koneksi yang tidak aman. Pengaturan opsi checksum memenuhi prasyarat karena keduanya memiliki SHA256. Namun, SQLNET.ENCRYPTION_TYPES_CLIENT dan SQLNET.ENCRYPTION_TYPES_SERVER mengg gunakanDES, 3DES, dan metode enkripsi RC4, yang bersifat tidak aman. Oleh karena itu, untuk menetapkan opsi SQLNET.ALLOW_WEAK_CRYPTO* ke FALSE, pertama tetapkan SQLNET.ENCRYPTION_TYPES_SERVER dan SQLNET.ENCRYPTION_TYPES_CLIENT ke metode enkripsi yang aman seperti AES256.

Pengaturan opsi	Nilai
SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT	SHA256, SHA384, SHA512

Pengaturan opsi	Nilai
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER	SHA1,MD5,SHA256
SQLNET.ENCRYPTION_TYPES_CLIENT	RC4_256, 3DES168, DES40
SQLNET.ENCRYPTION_TYPES_SERVER	RC4_256, 3DES168, DES40

Menghapus opsi NNE

Anda dapat menghapus NNE dari instans DB.

Untuk menghapus NNE dari instans DB, lakukan salah satu tindakan berikut ini:

- Untuk menghapus NNE dari beberapa instans DB, hapus opsi NNE dari grup opsi yang mencakupnya. Perubahan ini memengaruhi semua instans DB yang menggunakan grup opsi tersebut. Setelah Anda menghapus opsi NNE, Anda tidak perlu memulai ulang instans DB Anda. Untuk informasi selengkapnya, lihat Menghapus opsi dari grup opsi.
- Untuk menghapus NNE dari satu instans DB, ubah instans DB dan tentukan grup opsi berbeda yang tidak menyertakan opsi NNE. Anda dapat menentukan grup opsi default (kosong) atau grup opsi kustom yang berbeda. Setelah Anda menghapus opsi NNE, Anda tidak perlu memulai ulang instans DB Anda. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Oracle OLAP

Amazon RDS mendukung Oracle OLAP melalui penggunaan opsi OLAP. Opsi ini menyediakan On-line Analytical Processing (OLAP) untuk instans DB Oracle. Anda dapat menggunakan Oracle OLAP untuk menganalisis data dalam jumlah besar dengan membuat objek dan kubus dimensi sesuai dengan standar OLAP. Lihat informasi yang lebih lengkap dalam dokumentasi Oracle.

> **Important**
>
> Jika Anda menggunakan Oracle OLAP, Amazon RDS secara otomatis memperbarui instans DB Anda ke Oracle PSU terbaru jika ada kerentanan keamanan dengan skor Common Vulnerability Scoring System (CVSS) 9+ atau kerentanan keamanan lain yang diumumkan.

Amazon RDS mendukung Oracle OLAP untuk edisi dan versi Oracle berikut:

- Oracle Database 21c Enterprise Edition, semua versi
- Oracle Database 19c Enterprise Edition, semua versi
- Oracle Database 12c Rilis 2 (12.2.0.1) Enterprise Edition, semua versi
- Oracle Database 12c Rilis 1 (12.1.0.2) Enterprise Edition, versi 12.1.0.2.v13 atau yang lebih baru

Prasyarat untuk Oracle OLAP

Berikut adalah prasyarat untuk menggunakan Oracle OLAP:

- Anda harus memiliki lisensi Oracle OLAP dari Oracle. Untuk informasi lebih lanjut, lihat Licensing Information dalam dokumentasi Oracle.
- Instans DB Anda harus memiliki kelas instans yang memadai. Oracle OLAP tidak didukung untuk kelas instans DB db.t3.micro atau db.t3.small. Untuk informasi selengkapnya, lihat Kelas instans RDS for Oracle.
- Instans DB Anda harus mengaktifkan Peningkatan Versi Minor Otomatis. Opsi ini memungkinkan instans DB Anda menerima peningkatan versi mesin DB minor secara otomatis ketika tersedia dan diperlukan untuk opsi apa pun yang menginstal Oracle Java Virtual Machine (JVM). Amazon RDS menggunakan opsi ini untuk memperbarui instans DB Anda ke Oracle Patch Set Update (PSU) atau Release Update (RU) terbaru. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

- Instans DB Anda tidak boleh memiliki pengguna bernama `OLAPSYS`. Jika ya, penginstalan opsi OLAP gagal.

Praktik terbaik untuk Oracle OLAP

Berikut adalah praktik terbaik dalam penggunaan Oracle OLAP:

- Untuk keamanan maksimal, gunakan opsi `OLAP` dengan Secure Sockets Layer (SSL). Untuk informasi selengkapnya, lihat Lapisan Soket Aman Oracle.
- Konfigurasikan instans DB Anda untuk membatasi akses ke instans DB Anda. Untuk informasi selengkapnya, lihat Skenario untuk mengakses instans DB di VPC dan Bekerja dengan klaster DB dalam VPC.

Menambahkan opsi Oracle OLAP

Berikut adalah proses umum untuk menambahkan opsi `OLAP` ke instans DB:

1. Buat grup opsi baru, atau salin atau ubah grup opsi yang ada.
2. Tambahkan opsi tersebut ke grup opsi.
3. Kaitkan grup opsi tersebut dengan instans DB.

Jika Oracle Java Virtual Machine (JVM) tidak diinstal pada instans DB, akan terjadi pemadaman singkat saat opsi `OLAP` ditambahkan. Pemadaman tidak terjadi jika Oracle Java Virtual Machine (JVM) sudah diinstal pada instans DB. Setelah opsi ditambahkan, instans DB tidak perlu dimulai ulang. Setelah grup opsi aktif, Oracle OLAP akan langsung tersedia.

Untuk menambahkan opsi OLAP ke instans DB

1. Tentukan grup opsi yang ingin Anda gunakan. Anda dapat membuat grup opsi baru atau menggunakan grup opsi yang ada. Jika Anda ingin menggunakan grup opsi yang ada, lanjutkan ke langkah berikutnya. Jika tidak, buat grup opsi DB kustom dengan pengaturan berikut:

   - Untuk Mesin, pilih edisi Oracle untuk instans DB Anda.
   - Untuk Versi mesin utama, pilih versi instans DB Anda.

   Untuk informasi selengkapnya, lihat Membuat grup opsi.

2. Tambahkan opsi OLAP ke grup opsi. Untuk informasi selengkapnya tentang cara menambahkan opsi, lihat Menambahkan opsi ke grup opsi.
3. Terapkan grup opsi ke instans DB baru atau yang sudah ada:
   - Untuk instans DB baru, terapkan grup opsi saat Anda meluncurkan instans. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.
   - Untuk instans DB yang sudah ada, terapkan grup opsi dengan memodifikasi instans dan menambahkan grup opsi baru. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Menggunakan Oracle OLAP

Setelah Anda mengaktifkan opsi Oracle OLAP, Anda dapat mulai menggunakannya. Untuk daftar fitur yang didukung untuk Oracle OLAP, lihat dokumentasi Oracle.

Menghapus opsi Oracle OLAP

Setelah Anda membatalkan semua objek yang menggunakan jenis data yang disediakan oleh opsi `OLAP`, Anda dapat menghapus opsi tersebut dari instans DB. Jika Oracle Java Virtual Machine (JVM) tidak diinstal pada instans DB, akan terjadi pemadaman singkat saat opsi `OLAP` dihapus. Pemadaman tidak terjadi jika Oracle Java Virtual Machine (JVM) sudah diinstal pada instans DB. Setelah opsi `OLAP` dihapus, instans DB tidak perlu dimulai ulang.

Untuk membatalkan opsi **`OLAP`**

1. Cadangkan data Anda.

   > **Warning**
   >
   > Jika instans menggunakan jenis data yang diaktifkan sebagai bagian dari opsi, dan jika Anda menghapus opsi `OLAP`, data Anda bisa hilang. Untuk informasi selengkapnya, lihat Mencadangkan, memulihkan, dan mengekspor data.

2. Periksa apakah ada jenis data referensi objek atau fitur dari opsi `OLAP`.
3. Batalkan objek apa pun yang mereferensikan jenis data atau fitur opsi `OLAP`.
4. Lakukan salah satu langkah berikut:

- Hapus opsi `OLAP` dari grup opsi asalnya. Perubahan ini memengaruhi semua instans DB yang menggunakan grup opsi tersebut. Untuk informasi selengkapnya, lihat Menghapus opsi dari grup opsi.
- Ubah instans DB dan tentukan grup opsi lain yang tidak menyertakan opsi `OLAP`. Perubahan ini memengaruhi instans DB tunggal. Anda dapat menentukan grup opsi default (kosong) atau grup opsi kustom yang berbeda. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Lapisan Soket Aman Oracle

Anda mengaktifkan enkripsi SSL untuk instans DB RDS for Oracle dengan menambahkan opsi SSL Oracle ke grup opsi yang terkait dengan instans DB. Amazon RDS menggunakan port kedua, sebagaimana diperlukan oleh Oracle, untuk koneksi SSL. Pendekatan ini memungkinkan komunikasi teks yang jelas dan terenkripsi SSL terjadi pada waktu yang sama antara instans DB dan SQL*Plus. Misalnya, Anda dapat menggunakan port dengan komunikasi teks jelas untuk berkomunikasi dengan sumber daya lain di dalam VPC sambil menggunakan port komunikasi terenkripsi SSL untuk berkomunikasi dengan sumber daya di luar VPC.

> **Note**
>
> Anda dapat menggunakan SSL atau Enkripsi Jaringan Asli (NNE) pada RDS yang sama untuk instans DB Oracle, tetapi tidak keduanya sekaligus. Jika Anda menggunakan enkripsi SSL, pastikan untuk menonaktifkan enkripsi koneksi lainnya. Untuk informasi selengkapnya, lihat Enkripsi jaringan asli Oracle.

SSL/TLS dan NNE tidak lagi menjadi bagian dari Keamanan Lanjutan Oracle. Dalam RDS untuk Oracle, Anda dapat menggunakan enkripsi SSL dengan semua edisi berlisensi dari versi basis data berikut:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)
- Oracle Database 12c Rilis 2 (12.2) — rilis ini tidak lagi didukung
- Oracle Database 12c Rilis 1 (12.1) — rilis ini tidak lagi didukung

Versi TLS untuk opsi SSL Oracle

Amazon RDS for Oracle mendukung Keamanan Lapisan Pengangkutan (TLS) versi 1.0 dan 1.2. Saat Anda menambahkan opsi SSL Oracle baru, tetapkan `SQLNET.SSL_VERSION` secara eksplisit ke nilai yang valid. Nilai-nilai berikut diizinkan untuk pengaturan opsi ini:

- `"1.0"`— Klien dapat terhubung ke instans DB menggunakan TLS versi 1.0 saja. Untuk opsi Oracle SSL yang ada, `SQLNET.SSL_VERSION` ditetapkan ke `"1.0"` secara otomatis. Anda dapat mengubah pengaturan bila diperlukan.
- `"1.2"` – Klien dapat terhubung ke instans DB menggunakan TLS 1.2 saja.

- `"1.2 or 1.0"`- Klien dapat terhubung ke instans DB menggunakan TLS 1.2 atau 1.0.

Paket sandi untuk opsi SSL Oracle

Amazon RDS for Oracle mendukung beberapa paket sandi SSL. Secara default, opsi SSL Oracle dikonfigurasi untuk menggunakan paket sandi `SSL_RSA_WITH_AES_256_CBC_SHA`. Untuk menentukan paket sandi yang berbeda untuk digunakan melalui koneksi SSL, gunakan pengaturan opsi `SQLNET.CIPHER_SUITE`.

Tabel berikut merangkum dukungan SSL untuk RDS for Oracle. Rilis Oracle Database yang ditentukan mendukung semua edisi.

Paket sandi (SQLNET.CIPHER_SUITE)	Dukungan versi TLS (SQLNET.SSL_VERSION)	Rilis Oracle Database yang didukung	Dukungan FIPS	Sesuai dengan FedRAMP
SSL_RSA_WITH_AES_256_CBC_SHA (default)	1.0 dan 1.2	12c, 19c, 21c	Ya	Tidak
SSL_RSA_WITH_AES_256_CBC_SHA256	1.2	12c, 19c, 21c	Ya	Tidak
SSL_RSA_WITH_AES_256_GCM_SHA384	1.2	12c, 19c, 21c	Ya	Tidak
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	1.2	19c, 21c	Ya	Ya
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	1.2	19c, 21c	Ya	Ya
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	1.2	19c, 21c	Ya	Ya
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	1.2	19c, 21c	Ya	Ya

Paket sandi (SQLNET.CIPHER_SUITE)	Dukungan versi TLS (SQLNET.SSL_VERSION)	Rilis Oracle Database yang didukung	Dukungan FIPS	Sesuai dengan FedRAMP
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	1.2	19c, 21c	Ya	Ya
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	1.2	19c, 21c	Ya	Ya

Dukungan FIPS

RDS for Oracle memungkinkan Anda untuk menggunakan standar Federal Information Processing Standard (FIPS) untuk 140-2. FIPS 140-2 adalah standar pemerintah Amerika Serikat yang menetapkan persyaratan keamanan modul kriptografi. Anda mengaktifkan standar FIPS dengan menetapkan `FIPS.SSLFIPS_140` ke `TRUE` untuk opsi SSL Oracle. Ketika FIPS 140-2 dikonfigurasi untuk SSL, pustaka kriptografi mengenkripsi data antara klien dan instans DB RDS for Oracle.

Klien harus menggunakan paket sandi yang mematuhi FIPS. Saat membuat koneksi, klien dan instans DB RDS for Oracle menegosiasikan paket sandi mana yang akan digunakan saat mengirimkan pesan bolak-balik. Tabel di Paket sandi untuk opsi SSL Oracle menunjukkan paket sandi SSL yang mematuhi FIPS untuk setiap versi TLS. Untuk informasi selengkapnya, lihat Oracle database FIPS 140-2 settings di dokumentasi Oracle Database.

Menambahkan opsi SSL

Untuk menggunakan SSL, instans DB RDS for Oracle Anda harus dikaitkan dengan grup opsi yang menyertakan opsi `SSL`.

Konsol

Untuk menambahkan opsi SSL ke grup opsi

1. Buat grup opsi baru atau identifikasi grup opsi yang ada yang dapat ditambahi opsi `SSL`.

   Untuk informasi tentang cara membuat grup opsi, lihat Membuat grup opsi.
2. Tambahkan opsi `SSL` ke grup opsi.

Jika Anda hanya ingin menggunakan paket sandi yang diverifikasi FIPS untuk koneksi SSL, tetapkan opsi `FIPS.SSLFIPS_140` ke `TRUE`. Untuk informasi tentang standar FIPS, lihat Dukungan FIPS.

Untuk informasi tentang cara menambahkan opsi ke grup opsi, lihat Menambahkan opsi ke grup opsi.

3. Buat instans DB RDS for Oracle yang baru dan kaitkan grup opsi dengannya, atau modifikasi instans DB RDS for Oracle untuk mengaitkan grup opsi dengannya.

   Untuk informasi tentang cara membuat instans DB, lihat Membuat instans DB Amazon RDS.

   Untuk informasi tentang cara memodifikasi instans DB, lihat Memodifikasi instans DB Amazon RDS.

AWS CLI

Untuk menambahkan opsi SSL ke grup opsi

1. Buat grup opsi baru atau identifikasi grup opsi yang ada yang dapat ditambahi opsi `SSL`.

   Untuk informasi tentang cara membuat grup opsi, lihat Membuat grup opsi.

2. Tambahkan opsi `SSL` ke grup opsi.

   Tentukan pengaturan opsi berikut:

   - `Port`- Nomor port SSL
   - `VpcSecurityGroupMemberships`- Grup keamanan VPC yang opsinya diaktifkan
   - `SQLNET.SSL_VERSION`- Versi TLS yang dapat digunakan klien untuk terhubung ke instans DB

   Misalnya, perintah AWS CLI berikut menambahkan opsi `SSL` ke grup opsi bernama `ora-option-group`.

   Example

   Untuk Linux, macOS, atau Unix:

```
aws rds add-option-to-option-group --option-group-name ora-option-group \
```

```
  --options
 'OptionName=SSL,Port=2484,VpcSecurityGroupMemberships="sg-68184619",OptionSettings=[{Name=
```

Untuk Windows:

```
aws rds add-option-to-option-group --option-group-name ora-option-group ^
  --options
 'OptionName=SSL,Port=2484,VpcSecurityGroupMemberships="sg-68184619",OptionSettings=[{Name=
```

3. Buat instans DB RDS for Oracle yang baru dan kaitkan grup opsi dengannya, atau modifikasi instans DB RDS for Oracle untuk mengaitkan grup opsi dengannya.

   Untuk informasi tentang cara membuat instans DB, lihat Membuat instans DB Amazon RDS.

   Untuk informasi tentang cara memodifikasi instans DB, lihat Memodifikasi instans DB Amazon RDS.

Mengonfigurasi SQL*Plus untuk menggunakan SSL dengan instans DB RDS for Oracle

Sebelum Anda dapat terhubung ke instans DB RDS for Oracle yang menggunakan opsi SSL Oracle, Anda harus mengonfigurasi SQL*Plus sebelum menghubungkan.

> **Note**
>
> Untuk mengizinkan akses ke instans DB dari klien yang sesuai, pastikan bahwa grup keamanan Anda dikonfigurasi dengan benar. Untuk informasi lebih lanjut, lihat Mengontrol akses dengan grup keamanan. Selain itu, instruksi ini ditujukan untuk SQL*Plus dan klien lain yang secara langsung menggunakan Oracle home. Untuk koneksi JDBC, lihat Menyiapkan koneksi SSL melalui JDBC.

Untuk mengonfigurasi SQL*Plus untuk menggunakan SSL agar terhubung ke instans DB RDS for Oracle

1. Menetapkan variabel lingkungan `ORACLE_HOME` ke lokasi direktori Oracle home Anda.

Path ke direktori Oracle home Anda bergantung pada instalasi Anda. Contoh berikut menetapkan variabel lingkungan `ORACLE_HOME`.

```
prompt>export ORACLE_HOME=/home/user/app/user/product/12.1.0/dbhome_1
```

Untuk informasi tentang pengaturan variabel lingkungan Oracle, lihat SQL*Plus environment variables di dokumentasi Oracle, dan juga lihat panduan instalasi Oracle untuk sistem operasi Anda.

2. Menambahkan `$ORACLE_HOME/lib` ke variabel lingkungan `LD_LIBRARY_PATH`.

   Berikut ini adalah contoh yang menetapkan variabel lingkungan LD_LIBRARY_PATH.

   ```
   prompt>export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib
   ```

3. Buat direktori untuk dompet Oracle di `$ORACLE_HOME/ssl_wallet`.

   Berikut ini adalah contoh yang membuat direktori dompet Oracle.

   ```
   prompt>mkdir $ORACLE_HOME/ssl_wallet
   ```

4. Unduh file .pem paketan sertifikat yang berfungsi untuk semua Wilayah AWS dan letakkan file di direktori ssl_wallet. Untuk informasi, lihat .

5. Di direktori `$ORACLE_HOME/network/admin`, ubah atau buat file `tnsnames.ora` dan sertakan entri berikut.

   ```
   net_service_name =
     (DESCRIPTION =
       (ADDRESS_LIST =
         (ADDRESS =
           (PROTOCOL = TCPS)
           (HOST = endpoint)
           (PORT = ssl_port_number)
         )
       )
       (CONNECT_DATA =
         (SID = database_name)
       )
       (SECURITY =
         (SSL_SERVER_CERT_DN =
    "C=US,ST=Washington,L=Seattle,O=Amazon.com,OU=RDS,CN=endpoint")
   ```

```
    )
  )
```

6. Di direktori yang sama, ubah atau buat file sqlnet.ora dan sertakan parameter berikut.

> **Note**
>
> Untuk berkomunikasi dengan entitas melalui koneksi aman TLS, Oracle memerlukan dompet dengan sertifikat yang diperlukan untuk otentikasi. Anda dapat menggunakan utilitas ORAPKI Oracle untuk membuat dan memelihara dompet Oracle, seperti yang ditunjukkan pada langkah 7. Untuk informasi lebih lanjut, lihat Setting up Oracle wallet using ORAPKI di dokumentasi Oracle.

```
WALLET_LOCATION = (SOURCE = (METHOD = FILE) (METHOD_DATA = (DIRECTORY =
 $ORACLE_HOME/ssl_wallet)))
SSL_CLIENT_AUTHENTICATION = FALSE
SSL_VERSION = 1.0
SSL_CIPHER_SUITES = (SSL_RSA_WITH_AES_256_CBC_SHA)
SSL_SERVER_DN_MATCH = ON
```

> **Note**
>
> Anda dapat menetapkan `SSL_VERSION` ke nilai yang lebih tinggi jika instans DB Anda mendukungnya.

7. Jalankan perintah berikut untuk membuat dompet Oracle.

```
prompt>orapki wallet create -wallet $ORACLE_HOME/ssl_wallet -auto_login_only
```

8. Ekstrak setiap sertifikat dalam file paketan .pem ke dalam file .pem terpisah menggunakan utilitas OS.

9. Tambahkan setiap sertifikat ke dompet Anda menggunakan perintah `orapki` terpisah, mengganti *`certificate-pem-file`* dengan nama file absolut dari file .pem.

```
prompt>orapki wallet add -wallet $ORACLE_HOME/ssl_wallet -trusted_cert -cert
      certificate-pem-file -auto_login_only
```

Untuk informasi selengkapnya, lihat Merotasi sertifikat SSL/TLS.

Menghubungkan ke instans DB RDS for Oracle menggunakan SSL

Setelah Anda mengonfigurasi SQL*Plus untuk menggunakan SSL seperti yang dijelaskan sebelumnya, Anda dapat terhubung ke instans DB RDS for Oracle dengan opsi SSL. Jika perlu, Anda dapat terlebih dahulu mengekspor nilai `TNS_ADMIN` yang mengacu ke direktori yang berisi file tnsnames.ora dan sqlnet.ora. Tindakan ini memastikan bahwa SQL*Plus dapat menemukan kedua file ini secara konsisten. Contoh berikut mengekspor nilai `TNS_ADMIN`.

```
export TNS_ADMIN = ${ORACLE_HOME}/network/admin
```

Hubungkan ke instans DB. Misalnya, Anda dapat terhubung menggunakan SQL*Plus dan *<net_service_name>* dalam file tnsnames.ora.

```
sqlplus mydbuser@net_service_name
```

Anda juga dapat terhubung ke instans DB menggunakan SQL*Plus tanpa menggunakan file tnsnames.ora dengan menggunakan perintah berikut.

```
sqlplus 'mydbuser@(DESCRIPTION = (ADDRESS = (PROTOCOL = TCPS)(HOST = endpoint) (PORT
 = ssl_port_number))(CONNECT_DATA = (SID = database_name)))'
```

Anda juga dapat terhubung ke instans DB RDS for Oracle tanpa menggunakan SSL. Misalnya, perintah berikut menghubungkan ke instans DB melalui port teks yang jelas tanpa enkripsi SSL.

```
sqlplus 'mydbuser@(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = endpoint) (PORT
 = port_number))(CONNECT_DATA = (SID = database_name)))'
```

Jika Anda ingin menutup akses port Protokol Kontrol Transmisi (TCP), buat grup keamanan tanpa alamat IP masuk dan tambahkan ke instans. Penambahan ini menutup koneksi melalui port TCP, sambil tetap mengizinkan koneksi melalui port SSL yang ditentukan dari alamat IP dalam rentang yang diizinkan oleh grup keamanan opsi SSL.

Menyiapkan koneksi SSL melalui JDBC

Untuk menggunakan koneksi SSL melalui JDBC, Anda harus membuat keystore, mempercayai sertifikat CA root Amazon RDS, dan menggunakan cuplikan kode yang ditentukan berikut ini.

Untuk membuat keystore dalam format JKS, Anda dapat menggunakan perintah berikut. Untuk informasi selengkapnya tentang cara membuat keystore, lihat Creating a keystore di dokumentasi Oracle. Untuk informasi referensi, lihat keytool di Platform Java, Referensi Alat Edisi Standar.

```
keytool -genkey -alias client -validity 365 -keyalg RSA -keystore clientkeystore
```

Ikuti langkah-langkah berikut untuk mempercayai sertifikat CA root Amazon RDS.

Untuk mempercayai sertifikat CA root Amazon RDS

1. Unduh file .pem paketan sertifikat yang berfungsi untuk semua Wilayah AWS dan letakkan file di direktori ssl_wallet.

   Untuk informasi tentang cara mengunduh sertifikat, lihat .
2. Ekstrak setiap sertifikat dalam file.pem ke dalam file terpisah menggunakan utilitas OS.
3. Ubah setiap sertifikat ke format.der menggunakan `openssl` perintah terpisah, ganti *`certificate-pem-file`*dengan nama file sertifikat.pem (tanpa ekstensi.pem).

   ```
   openssl x509 -outform der -in certificate-pem-file.pem -out certificate-pem-file.der
   ```
4. Impor setiap sertifikat ke keystore menggunakan perintah berikut.

   ```
   keytool -import -alias rds-root -keystore clientkeystore.jks -file certificate-pem-file.der
   ```

   Untuk informasi selengkapnya, lihat Merotasi sertifikat SSL/TLS.
5. Konfirmasi bahwa keystore berhasil dibuat.

   ```
   keytool -list -v -keystore clientkeystore.jks
   ```

   Masukkan kata sandi keystore saat diminta.

Contoh kode berikut menunjukkan cara menyiapkan koneksi SSL menggunakan JDBC.

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Properties;

public class OracleSslConnectionTest {
    private static final String DB_SERVER_NAME = "dns-name-provided-by-amazon-rds";
    private static final Integer SSL_PORT = "ssl-option-port-configured-in-option-
group";
    private static final String DB_SID = "oracle-sid";
    private static final String DB_USER = "user-name";
    private static final String DB_PASSWORD = "password";
    // This key store has only the prod root ca.
    private static final String KEY_STORE_FILE_PATH = "file-path-to-keystore";
    private static final String KEY_STORE_PASS = "keystore-password";

    public static void main(String[] args) throws SQLException {
        final Properties properties = new Properties();
        final String connectionString = String.format(
                "jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCPS)(HOST=%s)(PORT=
%d))(CONNECT_DATA=(SID=%s)))",
                DB_SERVER_NAME, SSL_PORT, DB_SID);
        properties.put("user", DB_USER);
        properties.put("password", DB_PASSWORD);
        properties.put("oracle.jdbc.J2EE13Compliant", "true");
        properties.put("javax.net.ssl.trustStore", KEY_STORE_FILE_PATH);
        properties.put("javax.net.ssl.trustStoreType", "JKS");
        properties.put("javax.net.ssl.trustStorePassword", KEY_STORE_PASS);
        final Connection connection = DriverManager.getConnection(connectionString,
 properties);
        // If no exception, that means handshake has passed, and an SSL connection can
 be opened
    }
}
```

> **Note**
>
> Tentukan kata sandi yang berbeda dengan contoh yang ditampilkan di sini sebagai praktik terbaik keamanan.

Menerapkan kecocokan DN dengan koneksi SSL

Anda dapat menggunakan parameter Oracle `SSL_SERVER_DN_MATCH` untuk menerapkan bahwa nama yang dibedakan (DN) untuk server basis data cocok dengan nama layanannya. Jika Anda menerapkan verifikasi kecocokan, SSL memastikan bahwa sertifikat tersebut berasal dari server. Jika Anda tidak menerapkan verifikasi kecocokan, SSL melakukan pemeriksaan tetapi mengizinkan koneksi, terlepas dari apakah ada kecocokan. Jika Anda tidak menerapkan kecocokan, Anda memungkinkan server untuk berpotensi memalsukan identitasnya.

Untuk menerapkan pencocokan DN, tambahkan properti kecocokan DN dan gunakan string koneksi yang ditentukan di bawah ini.

Tambahkan properti ke koneksi klien untuk menerapkan pencocokan DN.

```
properties.put("oracle.net.ssl_server_dn_match", "TRUE");
```

Gunakan string koneksi berikut untuk menerapkan pencocokan DN saat menggunakan SSL.

```
final String connectionString = String.format(
    "jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCPS)(HOST=%s)(PORT=%d))" +
    "(CONNECT_DATA=(SID=%s))" +
    "(SECURITY = (SSL_SERVER_CERT_DN =
\"C=US,ST=Washington,L=Seattle,O=Amazon.com,OU=RDS,CN=%s\")))",
    DB_SERVER_NAME, SSL_PORT, DB_SID, DB_SERVER_NAME);
```

Pemecahan masalah koneksi SSL

Anda mungkin mengkueri basis data Anda dan menerima kesalahan `ORA-28860`.

```
ORA-28860: Fatal SSL error
28860. 00000 - "Fatal SSL error"
*Cause: An error occurred during the SSL connection to the peer. It is likely that this
 side sent data which the peer rejected.
*Action: Enable tracing to determine the exact cause of this error.
```

Kesalahan ini terjadi ketika klien mencoba untuk terhubung menggunakan versi TLS yang tidak didukung server. Untuk menghindari kesalahan ini, edit sqlnet.ora dan tetapkan `SSL_VERSION` ke versi TLS yang benar. Untuk informasi selengkapnya, lihat Oracle Support Document 2748438.1 di Dukungan Oracle.

Oracle Spatial

Amazon RDS mendukung Oracle Spatial melalui penggunaan opsi `SPATIAL`. Oracle Spatial menyediakan skema dan fungsi SQL yang memfasilitasi penyimpanan, pengambilan, pembaruan, dan kueri kumpulan data spasial dalam basis data Oracle. Untuk informasi lebih lanjut, lihat Spatial Concepts dalam dokumentasi Oracle.

Important

Jika Anda menggunakan Oracle Spatial, Amazon RDS secara otomatis memperbarui instans DB Anda ke Oracle PSU terbaru ketika salah satu dari yang berikut ini ada:

- Kerentanan keamanan dengan skor Common Vulnerability Scoring System (CVSS) 9+
- Kerentanan keamanan lain yang diumumkan

Amazon RDS hanya mendukung Oracle Spatial di Oracle Enterprise Edition (EE) dan Oracle Standard Edition 2 (SE2). Tabel berikut menunjukkan versi mesin DB yang mendukung EE dan SE2.

Versi Oracle DB	EE	SE2
21.0.0.0, semua versi	Ya	Ya
19.0.0.0, semua versi	Ya	Ya
12.2.0.1, semua versi	Ya	Ya
12.1.0.2.v13 atau versi yang lebih baru	Ya	Tidak

Note

Di Oracle Database 19c, bundel patch spasial terpisah dari database Patch Set Updates (PSU) dan Release Updates (RU). RDS untuk Oracle tidak mendukung penerapan bundel batch Spasial.

Prasyarat untuk Oracle Spatial

Berikut adalah prasyarat untuk menggunakan Oracle Spatial:

- Pastikan bahwa instans DB Anda berasal dari kelas instans yang memadai. Oracle Spatial tidak didukung untuk kelas instans DB db.t3.micro atau db.t3.small. Untuk informasi lebih lanjut, lihat Kelas instans RDS for Oracle.
- Pastikan instans DB Anda mengaktifkan Peningkatan Versi Minor Otomatis. Opsi ini memungkinkan instans DB Anda menerima peningkatan versi mesin DB minor secara otomatis ketika tersedia dan diperlukan untuk opsi apa pun yang menginstal Oracle Java Virtual Machine (JVM). Amazon RDS menggunakan opsi ini untuk memperbarui instans DB Anda ke Oracle Patch Set Update (PSU) atau Release Update (RU) terbaru. Untuk informasi lebih lanjut, lihat Memodifikasi instans DB Amazon RDS.

Praktik terbaik untuk Oracle Spatial

Berikut adalah praktik terbaik dalam penggunaan Oracle Spatial:

- Untuk keamanan maksimal, gunakan opsi `SPATIAL` dengan Secure Sockets Layer (SSL). Untuk informasi selengkapnya, lihat Lapisan Soket Aman Oracle.
- Konfigurasikan instans DB Anda untuk membatasi akses ke instans DB Anda. Untuk informasi lebih lanjut, lihat Skenario untuk mengakses instans DB di VPC dan Bekerja dengan klaster DB dalam VPC.

Menambahkan opsi Oracle Spatial

Berikut adalah proses umum untuk menambahkan opsi `SPATIAL` ke instans DB:

1. Buat grup opsi baru, atau salin atau ubah grup opsi yang ada.
2. Tambahkan opsi tersebut ke grup opsi.
3. Kaitkan grup opsi tersebut dengan instans DB.

Jika Oracle Java Virtual Machine (JVM) tidak diinstal pada instans DB, akan terjadi pemadaman singkat saat opsi `SPATIAL` ditambahkan. Pemadaman tidak terjadi jika Oracle Java Virtual Machine (JVM) sudah diinstal pada instans DB. Setelah opsi ditambahkan, instans DB tidak perlu dimulai ulang. Setelah grup opsi aktif, Oracle Spatial akan langsung tersedia.

> **Note**
>
> Selama pemadaman ini, fungsi verifikasi kata sandi akan dinonaktifkan sementara. Anda juga dapat melihat peristiwa terkait fungsi verifikasi kata sandi selama terjadi pemadaman. Fungsi verifikasi kata sandi akan diaktifkan kembali sebelum instans Oracle DB tersedia.

Untuk menambahkan opsi **`SPATIAL`** ke instans DB

1. Tentukan grup opsi yang ingin Anda gunakan. Anda dapat membuat grup opsi baru atau menggunakan grup opsi yang ada. Jika Anda ingin menggunakan grup opsi yang ada, lanjutkan ke langkah berikutnya. Jika tidak, buat grup opsi DB kustom dengan pengaturan berikut:

   a. Untuk Mesin, pilih edisi Oracle untuk instans DB Anda.

   b. Untuk Versi mesin utama, pilih versi instans DB Anda.

   Untuk informasi lebih lanjut, lihat Membuat grup opsi.
2. Tambahkan opsi SPATIAL ke grup opsi. Untuk informasi selengkapnya tentang cara menambahkan opsi, lihat Menambahkan opsi ke grup opsi.
3. Terapkan grup opsi ke instans DB baru atau yang sudah ada:

   - Untuk instans DB baru, Anda menerapkan grup opsi saat Anda meluncurkan instans. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.
   - Untuk instans DB yang ada, Anda menerapkan grup opsi dengan memodifikasi instans dan melampirkan grup opsi baru. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Menghapus opsi Oracle Spatial

Setelah Anda membatalkan semua objek yang menggunakan jenis data yang disediakan oleh opsi `SPATIAL`, Anda dapat membatalkan opsi tersebut dari instans DB. Jika Oracle Java Virtual Machine (JVM) tidak diinstal pada instans DB, akan terjadi pemadaman singkat saat opsi `SPATIAL` dihapus. Pemadaman tidak terjadi jika Oracle Java Virtual Machine (JVM) sudah diinstal pada instans DB. Setelah opsi `SPATIAL` dihapus, instans DB tidak perlu dimulai ulang.

Untuk membatalkan opsi **SPATIAL**

1. Cadangkan data Anda.

   > **Warning**
   >
   > Jika instans menggunakan jenis data yang diaktifkan sebagai bagian dari opsi, dan jika Anda menghapus opsi SPATIAL, data Anda bisa hilang. Untuk informasi selengkapnya, lihat Mencadangkan, memulihkan, dan mengekspor data.

2. Periksa apakah ada jenis data referensi objek atau fitur dari opsi SPATIAL.

   Jika opsi SPATIAL ada, instans bisa macet saat menerapkan grup opsi baru yang tidak memiliki opsi SPATIAL. Anda dapat mengidentifikasi objek menggunakan kueri berikut ini:

   ```
   SELECT OWNER, SEGMENT_NAME, TABLESPACE_NAME, BYTES/1024/1024 mbytes
   FROM   DBA_SEGMENTS
   WHERE  SEGMENT_TYPE LIKE '%TABLE%'
   AND    (OWNER, SEGMENT_NAME) IN
          (SELECT DISTINCT OWNER, TABLE_NAME
           FROM   DBA_TAB_COLUMNS
           WHERE  DATA_TYPE='SDO_GEOMETRY'
           AND    OWNER <> 'MDSYS')
   ORDER BY 1,2,3,4;

   SELECT OWNER, TABLE_NAME, COLUMN_NAME
   FROM   DBA_TAB_COLUMNS
   WHERE  DATA_TYPE = 'SDO_GEOMETRY'
   AND    OWNER <> 'MDSYS'
   ORDER BY 1,2,3;
   ```

3. Batalkan objek apa pun yang mereferensikan jenis data atau fitur opsi SPATIAL.
4. Lakukan salah satu dari berikut ini:

   - Hapus opsi SPATIAL dari grup opsi asalnya. Perubahan ini akan memengaruhi semua instans DB yang menggunakan grup opsi. Untuk informasi selengkapnya, lihat Menghapus opsi dari grup opsi.
   - Ubah instans DB dan tentukan grup opsi lain yang tidak menyertakan opsi SPATIAL. Perubahan ini memengaruhi instans DB tunggal. Anda dapat menentukan grup opsi default

(kosong) atau grup opsi kustom lain. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Oracle SQLT

Amazon RDS mendukung Oracle SQLTXPLAIN (SQLT) melalui penggunaan opsi SQLT.

Pernyataan `EXPLAIN PLAN` Oracle dapat menentukan rencana eksekusi pernyataan SQL. Hal ini dapat memverifikasi apakah pengoptimal Oracle memilih rencana eksekusi tertentu, seperti gabungan loop bersarang. Ini juga membantu Anda memahami keputusan pengoptimal, seperti mengapa pengoptimal memilih gabungan loop bersarang daripada gabungan hash. Jadi, `EXPLAIN PLAN` membantu Anda memahami performa pernyataan tersebut.

SQLT adalah utilitas Oracle yang menghasilkan laporan. Laporan tersebut mencakup statistik objek, metadata objek, parameter inisialisasi terkait pengoptimal, dan informasi lain yang dapat digunakan oleh administrator basis data untuk menyetel pernyataan SQL agar performanya optimal. SQLT membuat laporan HTML dengan hyperlink ke semua bagian dalam laporan.

Tidak seperti laporan Repositori Beban Kerja Otomatis atau Statspack, SQLT bekerja pada pernyataan SQL individual. SQLT adalah kumpulan file SQL, PL / SQL, dan SQL*Plus yang mengumpulkan, menyimpan, dan menampilkan data performa.

Berikut ini adalah versi Oracle yang didukung untuk setiap versi SQLT.

Versi SQLT	Oracle Database 21c	Oracle Database 19c	Oracle Database 12c Rilis 2 (12.2)	Oracle Database 12c Rilis 1 (12.1)
2018-07-25.v1	Didukung	Didukung	Didukung	Didukung
2018-03-31.v1	Tidak didukung	Tidak didukung	Didukung	Didukung
2016-04-29.v1	Tidak didukung	Tidak didukung	Didukung	Didukung

Untuk mengunduh SQLT dan mengakses instruksi untuk menggunakannya:

- Masuk ke akun My Oracle Support, dan buka dokumen berikut:
- Untuk mengunduh SQLT: Document 215187.1
- Untuk instruksi penggunaan SQLT: Document 1614107.1
- Untuk pertanyaan umum tentang SQLT: Document 1454160.1
- Untuk informasi tentang membaca output SQLT: Document 1456176.1
- Untuk menafsirkan Laporan Utama: Document 1922234.1

Anda dapat menggunakan SQLT dengan edisi apa pun dari versi Oracle Database berikut ini:

- Oracle Database 21c (21.0.0.0)
- Oracle Database 19c (19.0.0.0)
- Oracle Database 12c Rilis 2 (12.2.0.1)
- Oracle Database 12c Rilis 1 (12.1.0.2)

Amazon RDS tidak mendukung metode SQLT berikut:

- `XPLORE`
- `XHUME`

Prasyarat untuk SQLT

Berikut ini adalah prasyarat untuk menggunakan SQLT:

- Anda harus menghapus pengguna dan peran yang diperlukan oleh SQLT, jika ada.

  Opsi SQLT membuat pengguna dan peran berikut pada instans DB:

  - Pengguna `SQLTXPLAIN`
  - Pengguna `SQLTXADMIN`
  - Peran `SQLT_USER_ROLE`

  Jika instans DB Anda memiliki salah satu pengguna atau peran ini, masuk ke instans DB menggunakan klien SQL, dan batalkan menggunakan pernyataan berikut:

```
DROP USER SQLTXPLAIN CASCADE;
DROP USER SQLTXADMIN CASCADE;
DROP ROLE SQLT_USER_ROLE CASCADE;
```

- Anda harus menghapus tablespace yang dibutuhkan oleh SQLT, jika ada.

  Opsi SQLT membuat tablespace berikut pada instans DB:

  - `RDS_SQLT_TS`
  - `RDS_TEMP_SQLT_TS`

Jika instans DB Anda memiliki tablespace ini, masuk ke instans DB menggunakan klien SQL, lalu batalkan.

Pengaturan opsi SQLT

SQLT dapat bekerja dengan fitur berlisensi yang disediakan oleh Oracle Tuning Pack dan Oracle Diagnostics Pack. Oracle Tuning Pack menyertakan SQL Tuning Advisor, dan Oracle Diagnostics Pack menyertakan Automatic Workload Repository. Pengaturan SQLT mengaktifkan atau menonaktifkan akses ke fitur-fitur ini dari SQLT.

Amazon RDS mendukung pengaturan berikut untuk opsi SQLT.

Pengaturan opsi	Nilai valid	Nilai default	Deskripsi
`LICENSE_PACK`	`T`, `D`, `N`	`N`	Oracle Management Packs yang ingin Anda akses dengan SQLT. Gunakan salah satu nilai berikut: • `T` menunjukkan bahwa Anda memiliki lisensi untuk Oracle Tuning Pack dan Oracle Diagnostics Pack, dan Anda ingin mengakses SQL Tuning Advisor dan Automatic Workload Repository dari SQLT. • `D` menunjukkan bahwa Anda memiliki lisensi untuk Oracle Diagnostics Pack, dan Anda ingin mengakses Automatic Workload Repository dari SQLT. • `N` menunjukkan bahwa Anda tidak memiliki lisensi untuk Oracle Tuning Pack dan Oracle Diagnostics Pack, atau Anda memiliki lisensi untuk salah satu atau keduanya, tetapi Anda tidak ingin SQLT mengakses nya.

Pengaturan opsi	Nilai valid	Nilai default	Deskripsi
			Note Amazon RDS tidak memberikan lisensi untuk Oracle Management Packs ini. Jika Anda menunjukkan bahwa Anda ingin menggunakan paket yang tidak termasuk dalam instans DB Anda, Anda dapat menggunakan SQLT dengan instans DB. Namun, SQLT tidak dapat mengakses paket, dan laporan SQLT tidak menyertakan data untuk paket tersebut. Misalnya, jika Anda menentukan T, tetapi instans DB tidak menyertakan Oracle Tuning Pack, SQLT berfungsi pada instans DB, tetapi laporan yang dihasilkannya tidak berisi data yang terkait dengan Oracle Tuning Pack.

`VERSION`	`2016-04-29.v1`
	`2018-03-31.v1`
	`2018-07-25.v1`
		`2016-04-29.v1`	Versi SQLT yang ingin Anda instal.
			Note
			Untuk Oracle Database 19c dan 21c, satu-satunya versi yang didukung adalah `2018-07-25.v1` . Ini adalah versi default untuk rilis-rilis ini.

Menambahkan opsi SQLT

Berikut adalah proses umum untuk menambahkan opsi SQLT ke instans DB:

1. Buat grup opsi baru, atau salin atau ubah grup opsi yang ada.
2. Tambahkan opsi SQLT ke grup opsi.
3. Kaitkan grup opsi tersebut dengan instans DB.

Setelah Anda menambahkan opsi SQLT, SQLT langsung aktif setelah grup opsi aktif.

Untuk menambahkan opsi SQLT ke instans DB

1. Tentukan grup opsi yang ingin Anda gunakan. Anda dapat membuat grup opsi baru atau menggunakan grup opsi yang ada. Jika Anda ingin menggunakan grup opsi yang ada, lanjutkan ke langkah berikutnya. Jika tidak, buat grup opsi DB kustom dengan pengaturan berikut:

   a. Untuk Mesin, pilih edisi Oracle yang ingin Anda gunakan. Opsi SQLT didukung di semua edisi.

   b. Untuk Versi mesin utama, pilih versi instans DB Anda.

   Untuk informasi selengkapnya, lihat Membuat grup opsi.
2. Tambahkan opsi SQLT ke grup opsi. Untuk informasi selengkapnya tentang cara menambahkan opsi, lihat Menambahkan opsi ke grup opsi.
3. Terapkan grup opsi ke instans DB baru atau yang sudah ada:

   - Untuk instans DB baru, Anda menerapkan grup opsi saat Anda meluncurkan instans. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.

   - Untuk instans DB yang ada, Anda menerapkan grup opsi dengan memodifikasi instans dan melampirkan grup opsi baru. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.
4. (Opsional) Verifikasi penginstalan SQLT di setiap instans DB dengan opsi SQLT.

   a. Gunakan klien SQL untuk menyambungkan ke instans DB sebagai pengguna master.

   Untuk informasi tentang cara menghubungkan ke instans DB Oracle menggunakan klien SQL, lihat Menghubungkan ke instans RDS for Oracle DB.

b. Jalankan kueri berikut:

```
SELECT sqltxplain.sqlt$a.get_param('tool_version') sqlt_version FROM DUAL;
```

Kueri tersebut mengembalikan versi opsi SQLT saat ini di Amazon RDS. `12.1.160429` adalah contoh versi SQLT yang tersedia di Amazon RDS.

5. Ubah kata sandi pengguna yang dibuat dengan opsi SQLT.

a. Gunakan klien SQL untuk menyambungkan ke instans DB sebagai pengguna master.

b. Jalankan pernyataan SQL berikut untuk mengubah kata sandi untuk pengguna `SQLTXADMIN`:

```
ALTER USER SQLTXADMIN IDENTIFIED BY new_password ACCOUNT UNLOCK;
```

> **Note**
>
> Tetapkan kata sandi selain perintah yang ditampilkan di sini sebagai praktik terbaik keamanan.

c. Jalankan pernyataan SQL berikut untuk mengubah kata sandi untuk pengguna `SQLTXPLAIN`:

```
ALTER USER SQLTXPLAIN IDENTIFIED BY new_password ACCOUNT UNLOCK;
```

> **Note**
>
> Tetapkan kata sandi selain perintah yang ditampilkan di sini sebagai praktik terbaik keamanan.

> **Note**
>
> Untuk meningkatkan SQLT, versi SQLT yang lebih lama perlu dihapus untuk menginstal versi yang baru. Jadi, semua metadata SQLT bisa hilang saat Anda meningkatkan SQLT. Peningkatan versi utama dari basis data juga menghapus instalan dan menginstal ulang SQLT. Contoh peningkatan versi utama adalah peningkatan dari Oracle Database 12c Rilis 2 (12.2) ke Oracle Database 19c.

Menggunakan SQLT

SQLT dapat digunakan dengan utilitas Oracle SQL*Plus.

Untuk menggunakan SQLT

1. Unduh file .zip SQLT dari Document 215187.1 di situs My Oracle Support.

   > **Note**
   >
   > Anda tidak dapat mengunduh SQLT 12.1.160429 dari situs My Oracle Support. Oracle telah menghentikan versi lama ini.

2. Buka zip file .zip SQLT.
3. Dari jendela perintah, ubah ke direktori `sqlt/run` pada sistem file Anda.
4. Dari jendela perintah, buka SQL*Plus, dan sambungkan ke instans DB sebagai pengguna master.

   Lihat informasi yang lebih lengkap tentang cara menyambung ke instans DB menggunakan SQL*Plus di Menghubungkan ke instans RDS for Oracle DB.
5. Dapatkan ID SQL dari pernyataan SQL:

   ```
   SELECT SQL_ID FROM V$SQL WHERE SQL_TEXT='sql_statement';
   ```

   Output-nya akan serupa seperti yang berikut ini:

```
SQL_ID
-------------
chvsmttqjzjkn
```

6. Analisis pernyataan SQL dengan SQLT:

```
START sqltxtract.sql sql_id sqltxplain_user_password
```

Misalnya, untuk SQL ID `chvsmttqjzjkn`, masukkan:

```
START sqltxtract.sql chvsmttqjzjkn sqltxplain_user_password
```

SQLT menghasilkan laporan HTML dan sumber daya terkait sebagai file .zip di direktori tempat perintah SQLT dijalankan.

7. (Opsional) Agar pengguna aplikasi dapat mendiagnosis pernyataan SQL dengan SQLT, berikan `SQLT_USER_ROLE` kepada setiap pengguna aplikasi dengan pernyataan berikut:

```
GRANT SQLT_USER_ROLE TO application_user_name;
```

> **Note**
>
> Oracle tidak merekomendasikan untuk menjalankan SQLT dengan pengguna `SYS` atau dengan pengguna yang memiliki peran `DBA`. Praktik terbaiknya adalah menjalankan diagnostik SQLT menggunakan akun pengguna aplikasi, dengan memberikan `SQLT_USER_ROLE` kepada pengguna aplikasi.

Meningkatkan opsi SQLT

Dengan Amazon RDS for Oracle, Anda dapat meningkatkan opsi SQLT dari versi yang sudah ada ke versi yang lebih tinggi. Untuk meningkatkan opsi SQLT, selesaikan langkah 1–3 di Menggunakan SQLT untuk SQLT versi baru. Selain itu, jika Anda memberikan hak istimewa untuk versi SQLT sebelumnya pada langkah 7 di bagian tersebut, berikan hak istimewa lagi untuk versi SQLT yang baru.

Meningkatkan opsi SQLT akan membuat metadata versi SQLT yang lama hilang. Skema versi SQLT yang lama dan objek terkait dihapus, dan versi SQLT yang baru diinstal. Untuk informasi selengkapnya tentang perubahan dalam SQLT versi terbaru, lihat Document 1614201.1 di situs My Oracle Support.

> **Note**
>
> Penurunan versi tidak didukung.

Mengubah pengaturan SQLT

Setelah mengaktifkan SQLT, Anda dapat mengubah pengaturan `LICENSE_PACK` dan `VERSION` untuk opsi.

Untuk informasi selengkapnya tentang cara mengubah pengaturan opsi, lihat Memodifikasi pengaturan opsi. Untuk informasi selengkapnya tentang setiap pengaturan, lihat Pengaturan opsi SQLT.

Menghapus opsi SQLT

Anda dapat menghapus SQLT dari instans DB.

Untuk menghapus opsi SQLT dari instans DB, lakukan salah satu hal berikut:

- Untuk menghapus SQLT dari beberapa instans DB, hapus opsi SQLT dari grup opsi asal instans DB. Perubahan ini memengaruhi semua instans DB yang menggunakan grup opsi tersebut. Untuk informasi selengkapnya, lihat Menghapus opsi dari grup opsi.
- Untuk menghapus opsi SQLT dari satu instans DB, modifikasi instans DB dan tentukan grup opsi lain yang tidak menyertakan opsi SQLT. Anda dapat menentukan grup opsi default (kosong) atau grup opsi kustom lain. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Oracle Statspack

Opsi Oracle Statspack menginstal dan mengaktifkan fitur statistik kinerja Oracle Statspack. Oracle Statspack adalah kumpulan skrip SQL, PL/SQL, dan SQL*Plus yang mengumpulkan, menyimpan, dan menampilkan data kinerja. Untuk informasi tentang cara menggunakan Oracle Statspack, lihat Oracle Statspack di dokumentasi Oracle.

> **Note**
>
> Oracle Statspack tidak lagi didukung oleh Oracle dan telah digantikan oleh Automatic Workload Repository (AWR) yang lebih canggih. AWR hanya tersedia untuk pelanggan Oracle Enterprise Edition yang telah membeli Paket Diagnostik. Anda dapat menggunakan Oracle Statspack dengan mesin DB Oracle apa pun di Amazon RDS. Anda tidak dapat menjalankan Oracle Statspack pada replika baca Amazon RDS.

Menyiapkan Oracle Statspack

Untuk menjalankan skrip Statspack, Anda harus menambahkan opsi Statspack.

Untuk menyiapkan Oracle Statspack

1. Di klien SQL, masuk ke DB Oracle dengan akun administratif.
2. Lakukan salah satu tindakan berikut, bergantung pada apakah Statspack telah diinstal:
   - Jika Statspack telah diinstal dan akun `PERFSTAT` terkait dengan Statspack, lanjutkan langsung ke Langkah 4.
   - Jika Statspack tidak diinstal dan akun `PERFSTAT` ada, hapus akun sebagai berikut:

     ```
     DROP USER PERFSTAT CASCADE;
     ```

     Jika tidak, mencoba menambahkan opsi Statspack akan menghasilkan kesalahan dan `RDS-Event-0058`.
3. Tambahkan opsi Statspack ke grup opsi. Lihat Menambahkan opsi ke grup opsi.

   Amazon RDS secara otomatis menginstal skrip Statspack pada instans DB dan kemudian menyiapkan akun `PERFSTAT`.

4. Setel ulang kata sandi menggunakan pernyataan SQL berikut, dengan mengganti pwd dengan kata sandi baru Anda:

```
ALTER USER PERFSTAT IDENTIFIED BY pwd ACCOUNT UNLOCK;
```

   Anda dapat masuk menggunakan akun pengguna PERFSTAT dan jalankan skrip Statspack.

5. Lakukan salah satu tindakan berikut, bergantung pada versi mesin DB Anda:

   - Jika Anda menggunakan Basis Data Oracle 12c Rilis 2 (12.2) atau yang lebih lama, lewati langkah ini.
   - Jika Anda menggunakan Basis Data Oracle 19c atau yang lebih baru, beri CREATE JOB hak istimewa untuk akun PERFSTAT menggunakan pernyataan berikut:

```
GRANT CREATE JOB TO PERFSTAT;
```

6. Pastikan peristiwa menunggu tidak ada aktivitas di tabel PERFSTAT.STATS$IDLE_EVENT terisi.

   Karena Oracle Bug 28523746, peristiwa menunggu tidak ada aktivitas di PERFSTAT.STATS $IDLE_EVENT mungkin tidak diisi. Untuk memastikan semua peristiwa tidak ada aktivitas tersedia, jalankan pernyataan berikut:

```
INSERT INTO PERFSTAT.STATS$IDLE_EVENT (EVENT)
SELECT NAME FROM V$EVENT_NAME WHERE WAIT_CLASS='Idle'
MINUS
SELECT EVENT FROM PERFSTAT.STATS$IDLE_EVENT;
COMMIT;
```

Menghasilkan laporan Statspack

Laporan Statspack membandingkan dua snapshot.

Untuk menghasilkan laporan Statspack

1. Di klien SQL, masuk ke DB Oracle dengan akun PERFSTAT.
2. Buat snapshot menggunakan salah satu teknik berikut:

   - Buat snapshot Statspack secara manual.

- Buat pekerjaan yang mengambil snapshot Statspack setelah interval waktu tertentu. Misalnya, tugas berikut membuat snapshot Statspack setiap jam:

```
VARIABLE jn NUMBER;
exec dbms_job.submit(:jn, 'statspack.snap;',SYSDATE,'TRUNC(SYSDATE
+1/24,''HH24'')');
COMMIT;
```

3. Lihat snapshot menggunakan kueri berikut:

```
SELECT SNAP_ID, SNAP_TIME FROM STATS$SNAPSHOT ORDER BY 1;
```

4. Jalankan prosedur `rdsadmin.rds_run_spreport` Amazon RDS, mengganti begin_snap dan end_snap dengan ID snapshot:

```
exec rdsadmin.rds_run_spreport(begin_snap,end_snap);
```

Misalnya, perintah berikut membuat laporan berdasarkan interval antara snapshot Statspack 1 dan 2:

```
exec rdsadmin.rds_run_spreport(1,2);
```

Nama file dari laporan Statspack menyertakan nomor dari dua snapshot tersebut. Misalnya, file laporan yang dibuat menggunakan snapshot Statspack 1 dan 2 akan diberi nama `ORCL_spreport_1_2.lst`.

5. Pantau kesalahan pada output.

Oracle Statspack melakukan pemeriksaan sebelum menjalankan laporan. Oleh karena itu, Anda juga dapat melihat pesan kesalahan pada output perintah. Misalnya, Anda mungkin mencoba membuat laporan berdasarkan rentang yang tidak valid, di mana nilai snapshot Statspack awal lebih besar dari nilai akhir. Dalam kasus ini, output menampilkan pesan kesalahan, tetapi mesin DB tidak menghasilkan file kesalahan.

```
exec rdsadmin.rds_run_spreport(2,1);
*
ERROR at line 1:
ORA-20000: Invalid snapshot IDs. Find valid ones in perfstat.stats$snapshot.
```

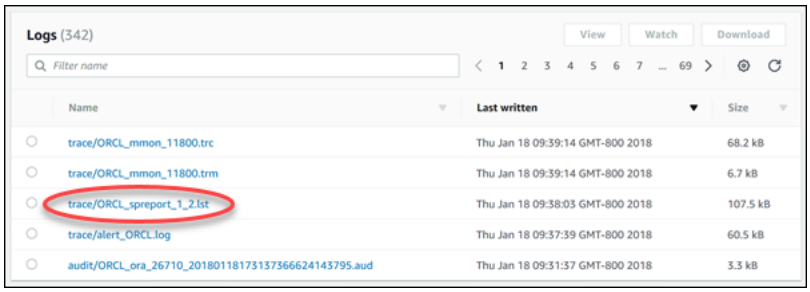
Jika Anda menggunakan angka yang tidak valid untuk snapshot Statspack, output akan menunjukkan kesalahan. Misalnya, jika Anda mencoba membuat laporan untuk snapshot 1 dan 50, tetapi snapshot 50 tidak ada, output akan menampilkan kesalahan.

```
exec rdsadmin.rds_run_spreport(1,50);
*
ERROR at line 1:
ORA-20000: Could not find both snapshot IDs
```

6. (Opsional)

   Untuk mengambil laporan, panggil prosedur file jejak, seperti yang dijelaskan di Bekerja dengan file jejak Oracle.

   Alternatifnya, unduh laporan Statspack dari konsol RDS. Buka bagian Log pada detail instans DB dan pilih Unduh:



   Jika kesalahan terjadi saat membuat laporan, mesin DB akan menggunakan konvensi penamaan yang sama seperti untuk laporan tetapi dengan ekstensi .err. Misalnya, jika kesalahan terjadi saat membuat laporan menggunakan snapshot Statspack 1 dan 7, laporan akan diberi nama `0RCL_spreport_1_7.err`. Anda dapat mengunduh laporan kesalahan menggunakan teknik yang sama seperti untuk laporan Snapshot standar.

Menghapus snapshot Statspack

Untuk menghapus berbagai snapshot Oracle Statspack, gunakan perintah berikut:

```
exec statspack.purge(begin snap, end snap);
```

Zona waktu Oracle

Untuk mengubah zona waktu sistem yang digunakan oleh instans DB Oracle Anda, gunakan opsi zona waktu. Misalnya, Anda dapat mengubah zona waktu instans DB agar kompatibel dengan lingkungan on-premise atau aplikasi lama. Opsi zona waktu mengubah zona waktu di tingkat host. Mengubah zona waktu memengaruhi semua kolom dan nilai tanggal, termasuk `SYSDATE` dan `SYSTIMESTAMP`.

Opsi zona waktu berbeda dari perintah `rdsadmin_util.alter_db_time_zone`. Perintah `alter_db_time_zone` mengubah zona waktu hanya untuk tipe data tertentu. Opsi zona waktu mengubah zona waktu untuk semua kolom dan nilai tanggal. Untuk informasi selengkapnya tentang `alter_db_time_zone`, lihat Mengatur zona waktu basis data. Untuk informasi lebih lanjut tentang pertimbangan peningkatan, lihat Pertimbangan zona waktu.

Pertimbangan untuk menetapkan zona waktu

Opsi zona waktu adalah opsi permanen dan tetap. Oleh karena itu, Anda tidak dapat melakukan hal berikut:

- Hapus opsi dari grup opsi setelah Anda menambahkan opsi.
- Hapus grup opsi dari instans DB setelah Anda menambahkan grup.
- Ubah pengaturan zona waktu dari opsi ke zona waktu yang berbeda.

Sebelum Anda menambahkan opsi zona waktu ke basis data produksi Anda, sebaiknya lakukan hal berikut:

- Ambil snapshot instans DB Anda. Jika Anda secara tidak sengaja mengatur zona waktu secara tidak benar, Anda harus memulihkan instans DB Anda ke pengaturan zona waktu sebelumnya. Untuk informasi selengkapnya, lihat Membuat snapshot DB untuk instans DB Single-AZ.
- Tambahkan opsi zona waktu ke instans DB pengujian. Menambahkan opsi zona waktu dapat menyebabkan masalah dengan tabel yang menggunakan tanggal sistem untuk menambahkan tanggal atau waktu. Sebaiknya analisis data dan aplikasi Anda pada instans pengujian untuk menilai dampak dari perubahan zona waktu pada instans produksi Anda.

Pengaturan opsi zona waktu

Amazon RDS mendukung pengaturan berikut untuk opsi zona waktu.

Pengaturan opsi	Nilai valid	Deskripsi
TIME_ZONE	Salah satu zona waktu yang tersedia. Untuk daftar lengkapnya, lihat Zona waktu yang tersedia.	Zona waktu baru untuk instans DB Anda.

Menambahkan opsi zona waktu

Proses umum untuk menambahkan opsi zona waktu ke instans DB adalah sebagai berikut:

1. Buat grup opsi baru, atau salin atau ubah grup opsi yang ada.
2. Tambahkan opsi tersebut ke grup opsi.
3. Kaitkan grup opsi dengan instans DB.

Saat Anda menambahkan opsi zona waktu, pemadaman singkat akan terjadi saat instans DB Anda dimulai ulang secara otomatis.

Konsol

Untuk menambahkan opsi zona waktu ke instans DB

1. Tentukan grup opsi yang ingin Anda gunakan. Anda dapat membuat grup opsi baru atau menggunakan grup opsi yang ada. Jika Anda ingin menggunakan grup opsi yang ada, lanjutkan ke langkah berikutnya. Jika tidak, buat grup opsi DB kustom dengan pengaturan berikut:

   a. Untuk Mesin pilih edisi Oracle untuk instans DB Anda.

   b. Untuk Versi mesin utama, pilih versi instans DB Anda.

   Untuk informasi lebih lanjut, lihat Membuat grup opsi.
2. Tambahkan opsi Zona Waktu ke grup opsi dan konfigurasikan pengaturan opsi.

> **Important**
>
> Jika Anda menambahkan opsi zona waktu ke grup opsi yang sudah ada yang sudah terpasang ke satu instans DB atau lebih, pemadaman singkat akan terjadi saat semua instans DB secara otomatis dimulai ulang.

Untuk informasi selengkapnya tentang cara menambahkan opsi, lihat Menambahkan opsi ke grup opsi. Untuk informasi selengkapnya tentang setiap pengaturan, lihat Pengaturan opsi zona waktu.

3. Terapkan grup opsi ke instans DB baru atau yang sudah ada:

   - Untuk instans DB baru, Anda menerapkan grup opsi saat Anda meluncurkan instans. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.
   - Untuk instans DB yang ada, Anda menerapkan grup opsi dengan memodifikasi instans dan melampirkan grup opsi baru. Ketika Anda menambahkan opsi zona waktu ke instans DB yang ada, pemadaman singkat akan terjadi saat instans DB Anda dimulai ulang secara otomatis. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

AWS CLI

Contoh berikut menggunakan perintah AWS CLI add-option-to-option-group untuk menambahkan `Timezone` opsi dan pengaturan `TIME_ZONE` opsi ke grup opsi yang disebut`myoptiongroup`. Zona waktu ditetapkan ke `Africa/Cairo`.

Untuk Linux, macOS, atau Unix:

```
aws rds add-option-to-option-group \
    --option-group-name "myoptiongroup" \
    --options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=Africa/
Cairo}]" \
    --apply-immediately
```

Untuk Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name "myoptiongroup" ^
```

```
    --options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=Africa/
Cairo}]" ^
    --apply-immediately
```

Mengubah pengaturan zona waktu

Opsi zona waktu adalah opsi permanen dan tetap. Anda tidak dapat menghapus opsi dari grup opsi setelah Anda menambahkannya. Anda tidak dapat menghapus grup opsi dari instans DB setelah Anda menambahkannya. Anda tidak dapat mengubah pengaturan zona waktu dari opsi ke zona waktu yang berbeda. Jika Anda menetapkan zona waktu secara tidak benar, pulihkan snapshot instans DB Anda dari sebelum opsi zona waktu ditambahkan.

Menghapus opsi zona waktu

Opsi zona waktu adalah opsi permanen dan tetap. Anda tidak dapat menghapus opsi dari grup opsi setelah Anda menambahkannya. Anda tidak dapat menghapus grup opsi dari instans DB setelah Anda menambahkannya. Untuk menghapus opsi zona waktu, pulihkan snapshot instans DB Anda dari sebelum opsi zona waktu ditambahkan.

Zona waktu yang tersedia

Anda dapat menggunakan nilai berikut untuk opsi zona waktu.

Zona	Zona waktu
Afrika	Afrika/Kairo, Afrika/Casablanca, Afrika/Harare, Afrika/Lagos, Afrika/Luanda, Afrika/Monrovia, Afrika/Nairobi, Afrika/Tripoli, Afrika/Windhoek
Amerika	Amerika/Araguaina, Amerika/Argentina/Buenos_Aires, Amerika/Asuncion, Amerika/Bogota, Amerika/Caracas, Amerika/Chicago, Amerika/Chihuahua, Amerika/Cuiaba, Amerika/Denver, Amerika/Detroit, Amerika/Fortaleza, Amerika/Godthab, Amerika/Guatemala, Amerika/Halifax, Amerika/Lima, Amerika/Los_Angeles, Amerika/Manaus, Amerika/Matamoros, Amerika/Mexico_City, Amerika/Monterrey, Amerika/Montevideo, Amerika/New_York, Amerika/Phoenix, Amerika/Santiago, Amerika/Sao_Paulo, Amerika/Tijuana, Amerika/Toronto
Asia	Asia/Amman, Asia/Ashgabat, Asia/Baghdad, Asia/Baku, Asia/Bangkok, Asia/Beirut, Asia/Calcutta, Asia/Damaskus, Asia/Dhaka, Asia/Hong_Kong, Asia/Irku

Zona	Zona waktu
	tsk, Asia/Jakarta, Asia/Yerusalem, Asia/Kabul, Asia/Karachi, Asia/Kathmandu, Asia/Kolkata, Asia/Krasnoyarsk, Asia/Magadan, Asia/Manila, Asia/Muscat, Asia/Novosibirsk, Asia/Rangoon, Asia/Riyadh, Asia/Seoul, Asia/Shanghai, Asia/Singapura, Asia/Taipei, Asia/Teheran, Asia/Tokyo, Asia/Ulaanbaatar, Asia/Vladivostok, Asia/Yakutsk, Asia/Yerevan
Atlantik	Atlantik/Azores, Atlantik/Cape_Verde
Australia	Australia/Adelaide, Australia/Brisbane, Australia/Darwin, Australia/Eucla, Australia/Hobart, Australia/Lord_Howe, Australia/Perth, Australia/Sydney
Brazil	Brasil/, Brasil/Timur DeNoronha
Canada	Kanada/Newfoundland, Kanada/Saskatchewan
Dll	Dll/GMT-3
Eropa	Eropa/Amsterdam, Eropa/Athena, Eropa/Berlin, Eropa/Dublin, Eropa/Helsinki, Eropa/Kaliningrad, Eropa/London, Eropa/Madrid, Eropa/Moskow, Eropa/Paris, Eropa/Praha, Eropa/Roma, Eropa/Sarajevo
Pasifik	Pasifik/Apia, Pasifik/Auckland, Pasifik/Chatham, Pasifik/Fiji, Pasifik/Guam, Pasifik/Honolulu, Pasifik/Kiritimati, Pasifik/Marquesas, Pasifik/Samoa, Pasifik/Tongatapu, Pasifik/Wake
AS	AS/Alaska, AS/Tengah, AS/East-Indiana, AS/Timur, AS/Pasifik
UTC	UTC

Pemutakhiran otomatis file zona waktu Oracle

Dengan `TIMEZONE_FILE_AUTOUPGRADE` opsi ini, Anda dapat memutakhirkan file zona waktu saat ini ke versi terbaru pada RDS Anda untuk instans Oracle DB.

Topik

Gambaran umum file zona waktu Oracle

File zona waktu Oracle Database menyimpan informasi berikut:

- Offset dari Coordinated Universal Time (UTC)
- Waktu transisi untuk Daylight Saving Time (DST)
- Singkatan untuk waktu standar dan DST

Oracle Database menyediakan beberapa versi file zona waktu. Saat Anda membuat basis data Oracle di lingkungan on-premise, Anda memilih versi file zona waktu. Untuk informasi selengkapnya, lihat Memilih File Zona Waktu di Panduan Dukungan Globalisasi Basis Data Oracle.

Jika aturan berubah untuk DST, Oracle akan menerbitkan file zona waktu baru. Oracle merilis file zona waktu baru ini secara independen dari jadwal untuk Pembaruan Rilis triwulanan (RU) dan Revisi Pembaruan Rilis (RUR). File zona waktu berada di host basis data di direktori `$ORACLE_HOME/oracore/zoneinfo/`. Nama file zona waktu menggunakan format DSTv*versi*, seperti pada DSTv35.

Pengaruh file zona waktu terhadap transfer data

Di Oracle Database, tipe data `TIMESTAMP WITH TIME ZONE` menyimpan stempel waktu dan data zona waktu. Data dengan tipe data `TIMESTAMP WITH TIME ZONE` menggunakan aturan dalam

versi file zona waktu terkait. Dengan demikian, `TIMESTAMP WITH TIME ZONE` data yang ada terpengaruh saat Anda memperbarui file zona waktu.

Masalah dapat terjadi ketika Anda mentransfer data antar basis data yang menggunakan versi file zona waktu yang berbeda. Misalnya, jika Anda mengimpor data dari database sumber dengan versi file zona waktu yang lebih tinggi daripada basis data target, database akan mengeluarkan `ORA-39405` kesalahan. Sebelumnya, Anda harus mengatasi kesalahan ini dengan menggunakan salah satu teknik berikut:

- Buat instans DB RDS for Oracle dengan file zona waktu yang diinginkan, ekspor data dari basis data sumber Anda, lalu impor ke basis data baru.
- Gunakan DMS AWS atau replikasi logis untuk memigrasikan data Anda.

Pembaruan otomatis menggunakan opsi TIMEZONE_FILE_AUTOUPGRADE

Ketika grup opsi yang dilampirkan ke RDS Anda untuk instans Oracle DB menyertakan `TIMEZONE_FILE_AUTOUPGRADE` opsi, RDS memperbarui file zona waktu Anda secara otomatis. Dengan memastikan bahwa database Oracle Anda menggunakan versi file zona waktu yang sama, Anda menghindari teknik manual yang memakan waktu ketika Anda memindahkan data di antara lingkungan yang berbeda. Opsi `TIMEZONE_FILE_AUTOUPGRADE` didukung untuk basis data kontainer (CDB) dan non-CDB.

Saat Anda menambahkan opsi `TIMEZONE_FILE_AUTOUPGRADE` ke grup opsi, Anda dapat memilih apakah akan menambahkan opsi sesegera mungkin atau selama periode pemeliharaan. *Setelah instans DB Anda menerapkan opsi baru, RDS memeriksa apakah itu dapat menginstal file versi DStv yang lebih baru.* DSTv*versi* target bergantung pada hal berikut:

- Versi mesin minor yang saat ini dijalankan instans DB Anda
- Versi mesin minor yang ingin dijadikan sebagai pemutakhiran instans DB Anda

Misalnya, versi file zona waktu Anda saat ini mungkin DStv33. Ketika RDS menerapkan pembaruan ke grup opsi Anda, mungkin menentukan bahwa DStv34 saat ini tersedia di sistem file instans DB Anda. RDS kemudian akan memperbarui file zona waktu Anda ke DStv34 secara otomatis.

Untuk menemukan versi DST yang tersedia dalam pembaruan rilis RDS yang didukung, lihat patch di Catatan rilis untuk Amazon Relational Database Service (Amazon RDS) for Oracle. Misalnya,

versi 19.0.0.0.ru-2022-10.rur-2022-10.r1 mencantumkan patch 34533061: RDBMS - PEMBARUAN DSTV39 - TZDATA2022C.

Strategi untuk memperbarui file zona waktu Anda

Memutakhirkan mesin DB Anda dan menambahkan `TIMEZONE_FILE_AUTOUPGRADE` opsi ke grup opsi adalah operasi terpisah. Menambahkan `TIMEZONE_FILE_AUTOUPGRADE` opsi memulai pembaruan file zona waktu Anda jika yang lebih baru tersedia. Anda menjalankan perintah berikut (hanya opsi yang relevan yang ditampilkan) baik segera atau di jendela pemeliharaan berikutnya:

- Tingkatkan mesin DB Anda hanya menggunakan perintah RDS CLI berikut:

```
modify-db-instance --engine-version name ...
```

- Tambahkan `TIMEZONE_FILE_AUTOUPGRADE` opsi hanya menggunakan perintah CLI berikut:

```
add-option-to-option-group --option-group-name name --options
 OptionName=TIMEZONE_FILE_AUTOUPGRADE ...
```

- Tingkatkan mesin DB Anda dan tambahkan grup opsi baru ke instans Anda menggunakan perintah CLI berikut:

```
modify-db-instance --engine-version name --option-group-name name ...
```

Strategi pembaruan Anda tergantung pada apakah Anda ingin memutakhirkan file database dan zona waktu bersama-sama atau hanya melakukan salah satu dari operasi ini. Perlu diingat bahwa jika Anda memperbarui grup opsi dan kemudian memutakhirkan mesin DB Anda dalam operasi API terpisah, pembaruan file zona waktu mungkin sedang berlangsung saat Anda memutakhirkan mesin DB Anda.

Contoh di bagian ini mengasumsikan hal berikut:

- Anda belum menambahkan `TIMEZONE_FILE_AUTOUPGRADE` ke grup opsi yang saat ini terkait dengan instans DB Anda.
- Instans DB Anda menggunakan basis data versi 19.0.0.0.ru-2019-07.rur-2019-07.r1 dan file zona waktu DSTv33.
- Sistem file instans DB Anda menyertakan file DSTv34.
- Pembaruan rilis 19.0.0.0.ru-2022-10.rur-2022-10.r1 menyertakan DSTv35.

Untuk memperbarui file zona waktu, Anda dapat menggunakan strategi berikut.

Topik

Memperbarui file zona waktu tanpa memutakhirkan mesin

Dalam skenario ini, basis data Anda menggunakan DSTv33, tetapi DSTv34 tersedia pada sistem file instans DB Anda. Anda ingin memperbarui file zona waktu yang digunakan oleh instans DB Anda dari DSTv33 ke DSTv34, tetapi Anda tidak ingin meningkatkan mesin Anda ke versi minor baru, yang mencakup DSTv35.

Dalam sebuah `add-option-to-option-group` perintah, tambahkan `TIMEZONE_FILE_AUTOUPGRADE` ke grup opsi yang digunakan oleh instans DB Anda. Tentukan apakah akan menambahkan opsi segera atau menundanya ke periode pemeliharaan. Setelah menerapkan `TIMEZONE_FILE_AUTOUPGRADE` opsi, RDS melakukan hal berikut:

1. Memeriksa versi DST baru.
2. Menentukan bahwa DStv34 tersedia pada sistem file.
3. Memperbarui file zona waktu segera.

Memutakhirkan file zona waktu dan versi mesin DB

Dalam skenario ini, basis data Anda menggunakan DSTv33, tetapi DSTv34 tersedia pada sistem file instans DB Anda. Anda ingin memutakhirkan mesin DB Anda ke versi minor 19.0.0.0.ru-2022-10.rur-2022-10.r1, yang mencakup DSTv35, dan memperbarui file zona waktu Anda ke DSTv35 selama pemutakhiran mesin. Dengan demikian, tujuan Anda adalah melewati DSTv34 dan memperbarui file zona waktu Anda langsung ke DSTv35.

Untuk memutakhirkan file engine dan zona waktu bersama-sama, jalankan `modify-db-instance` dengan `--engine-version` opsi `--option-group-name` dan. Anda dapat menjalankan perintah segera atau menundanya ke jendela pemeliharaan. `In --option-group-name`, tentukan grup opsi yang menyertakan `TIMEZONE_FILE_AUTOUPGRADE` opsi. Sebagai contoh:

```
aws rds modify-db-instance
```

```
    --db-instance-identifier my-instance \
    --engine-version new-version \
    ----option-group-name og-with-timezone-file-autoupgrade \
    --apply-immediately
```

RDS mulai meningkatkan mesin Anda ke 19.0.0.0.ru-2022-10.rur-2022-10.rur-2022-10.r1. Setelah menerapkan `TIMEZONE_FILE_AUTOUPGRADE` opsi, RDS memeriksa versi DST baru, melihat bahwa DStv35 tersedia di 19.0.0.0.ru-2022-10.rur-2022-10.r1, dan segera memulai pembaruan ke DStv35.

Untuk segera memutakhirkan mesin Anda dan kemudian memutakhirkan file zona waktu Anda, lakukan operasi secara berurutan:

1. Tingkatkan mesin DB Anda hanya menggunakan perintah CLI berikut:

   ```
   aws rds modify-db-instance \
       --db-instance-identifier my-instance \
       --engine-version new-version \
       --apply-immediately
   ```

2. Tambahkan `TIMEZONE_FILE_AUTOUPGRADE` opsi ke grup opsi yang dilampirkan ke instance Anda menggunakan perintah CLI berikut:

   ```
   aws rds add-option-to-option-group \
       --option-group-name og-in-use-by-your-instance \
       --options OptionName=TIMEZONE_FILE_AUTOUPGRADE \
       --apply-immediately
   ```

Memutakhirkan versi mesin DB Anda tanpa memperbarui file zona waktu

Dalam skenario ini, basis data Anda menggunakan DSTv33, tetapi DSTv34 tersedia pada sistem file instans DB Anda. Anda ingin memutakhirkan mesin DB Anda ke versi 19.0.0.0.ru-2022-10.rur-2022-10.r1, yang menyertakan DSTv35, tetapi mempertahankan file zona waktu DSTv33. Anda mungkin memilih strategi ini karena alasan berikut:

- Data Anda tidak menggunakan tipe data `TIMESTAMP WITH TIME ZONE`.
- Data Anda menggunakan tipe data `TIMESTAMP WITH TIME ZONE`, tetapi data Anda tidak terpengaruh oleh perubahan zona waktu.
- Anda ingin menunda memperbarui file zona waktu karena Anda tidak dapat menoleransi periode nonaktif tambahan.

Strategi Anda tergantung pada mana dari kemungkinan berikut yang benar:

- Instans DB Anda tidak terkait dengan grup opsi yang mencakup grup opsi yang mencakup `TIMEZONE_FILE_AUTOUPGRADE`. Dalam `modify-db-instance` perintah Anda, jangan tentukan grup opsi baru sehingga RDS tidak memperbarui file zona waktu Anda.
- Instans DB Anda saat ini dikaitkan dengan grup opsi yang menyertakan`TIMEZONE_FILE_AUTOUPGRADE`. Dalam satu `modify-db-instance` perintah, kaitkan instans DB Anda dengan grup opsi yang tidak menyertakan `TIMEZONE_FILE_AUTOUPGRADE` dan tingkatkan mesin DB Anda ke 19.0.0.0.ru-2022-10.rur-2022-10.r1.

Periode nonaktif selama pembaruan file zona waktu

Saat RDS memperbarui file zona waktu Anda, data yang ada yang menggunakan `TIMESTAMP WITH TIME ZONE` mungkin berubah. Dalam hal ini, pertimbangan utama Anda adalah periode nonaktif.

> **Warning**
>
> Jika Anda menambahkan opsi `TIMEZONE_FILE_AUTOUPGRADE`, peningkatan mesin Anda mungkin memperpanjang periode nonaktif. Memperbarui data zona waktu untuk basis data besar mungkin memakan waktu berjam-jam atau bahkan berhari-hari.

Lama pembaruan file zona waktu tergantung pada sejumlah faktor seperti berikut:

- Jumlah data `TIMESTAMP WITH TIME ZONE` dalam basis data Anda
- Konfigurasi instans DB
- Kelas instans DB
- Konfigurasi penyimpanan
- Konfigurasi basis data
- Pengaturan parameter basis data

Periode nonaktif tambahan dapat terjadi saat Anda melakukan hal berikut:

- Tambahkan opsi ke grup opsi saat instans DB menggunakan file zona waktu yang sudah tidak berlaku

- Mutakhirkan mesin basis data Oracle ketika versi mesin baru berisi versi baru dari file zona waktu

> **Note**
>
> Selama pembaruan file zona waktu, RDS for Oracle memanggil `PURGE DBA_RECYCLEBIN`.

Persiapan untuk memperbarui file zona waktu

Pemutakhiran file zona waktu memiliki dua fase terpisah: siapkan dan mutakhirkan. Meskipun tidak diperlukan, sebaiknya lakukan langkah persiapan. Pada langkah ini, Anda mengetahui data mana yang akan terpengaruh dengan menjalankan prosedur `DBMS_DST.FIND_AFFECTED_TABLES` PL/SQL. Untuk informasi selengkapnya tentang periode persiapan, lihat Upgrading the Time Zone File and Timestamp with Time Zone Data di dokumentasi Oracle Database.

Sebagai persiapan untuk memperbarui file zona waktu

1. Terhubung ke basis data Oracle Anda menggunakan klien SQL.
2. Tentukan versi file zona waktu saat ini yang digunakan.

   ```
   SELECT * FROM V$TIMEZONE_FILE;
   ```

3. Tentukan versi file zona waktu terbaru yang tersedia di instans DB Anda. Langkah ini hanya berlaku jika Anda menggunakan Oracle Database 12c Rilis 2 (12.2) atau yang lebih baru.

   ```
   SELECT DBMS_DST.GET_LATEST_TIMEZONE_VERSION FROM DUAL;
   ```

4. Tentukan ukuran total tabel yang memiliki kolom tipe `TIMESTAMP WITH LOCAL TIME ZONE` atau `TIMESTAMP WITH TIME ZONE`.

   ```
   SELECT SUM(BYTES)/1024/1024/1024 "Total_size_w_TSTZ_columns_GB"
   FROM   DBA_SEGMENTS
   WHERE  SEGMENT_TYPE LIKE 'TABLE%'
   AND    (OWNER, SEGMENT_NAME) IN
            (SELECT OWNER, TABLE_NAME
             FROM   DBA_TAB_COLUMNS
             WHERE  DATA_TYPE LIKE 'TIMESTAMP%TIME ZONE');
   ```

5. Tentukan nama dan ukuran segmen yang memiliki kolom tipe `TIMESTAMP WITH LOCAL TIME ZONE` atau `TIMESTAMP WITH TIME ZONE`.

```
SELECT OWNER, SEGMENT_NAME, SUM(BYTES)/1024/1024/1024
 "SEGMENT_SIZE_W_TSTZ_COLUMNS_GB"
FROM   DBA_SEGMENTS
WHERE  SEGMENT_TYPE LIKE 'TABLE%'
AND    (OWNER, SEGMENT_NAME) IN
         (SELECT OWNER, TABLE_NAME
          FROM   DBA_TAB_COLUMNS
          WHERE  DATA_TYPE LIKE 'TIMESTAMP%TIME ZONE')
GROUP BY OWNER, SEGMENT_NAME;
```

6. Jalankan langkah persiapan.

   - Prosedur `DBMS_DST.CREATE_AFFECTED_TABLE` membuat tabel untuk menyimpan data yang terpengaruh. Anda meneruskan nama tabel ini ke prosedur `DBMS_DST.FIND_AFFECTED_TABLES`. Untuk informasi selengkapnya, lihat CREATE_AFFECTED_TABLE Procedure di dokumentasi Oracle Database.
   - Prosedur `CREATE_ERROR_TABLE` membuat tabel untuk mencatat kesalahan. Untuk informasi selengkapnya, lihat CREATE_ERROR_TABLE Procedure di dokumentasi Oracle Database.

   Contoh berikut membuat data dan tabel kesalahan yang terpengaruh, dan menemukan semua tabel yang terpengaruh.

```
EXEC DBMS_DST.CREATE_ERROR_TABLE('my_error_table')
EXEC DBMS_DST.CREATE_AFFECTED_TABLE('my_affected_table')

EXEC DBMS_DST.BEGIN_PREPARE(new_version);
EXEC DBMS_DST.FIND_AFFECTED_TABLES('my_affected_table', TRUE, 'my_error_table');
EXEC DBMS_DST.END_PREPARE;

SELECT * FROM my_affected_table;
SELECT * FROM my_error_table;
```

7. Kueri tabel yang terpengaruh dan memiliki kesalahan.

```
SELECT * FROM my_affected_table;
SELECT * FROM my_error_table;
```

Menambahkan opsi pemutakhiran otomatis file zona waktu

Saat Anda menambahkan opsi ke grup opsi, grup opsi berada di salah satu status berikut:

- Grup opsi yang ada saat ini dilampirkan ke setidaknya satu instans DB. Saat Anda menambahkan opsi, semua instans DB yang menggunakan grup opsi ini dimulai ulang secara otomatis. Hal ini akan menyebabkan pemadaman singkat.
- Grup opsi yang ada tidak dilampirkan ke instans DB apa pun. Anda berencana untuk menambahkan opsi dan kemudian mengaitkan grup opsi yang ada dengan instans DB yang ada atau dengan instans DB baru.
- Anda membuat grup opsi baru dan menambahkan opsi. Anda berencana mengaitkan grup opsi baru dengan instans DB yang ada atau instans DB baru.

Konsol

Untuk menambahkan opsi pemutakhiran otomatis file zona waktu ke instans DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup opsi.
3. Tentukan grup opsi yang ingin Anda gunakan. Anda dapat membuat grup opsi baru atau menggunakan grup opsi yang ada. Jika Anda ingin menggunakan grup opsi yang ada, lanjutkan ke langkah berikutnya. Jika tidak, buat grup opsi DB kustom dengan pengaturan berikut:

   a. Untuk Mesin pilih edisi Oracle Database untuk instans DB Anda.

   b. Untuk Versi mesin utama, pilih versi instans DB Anda.

   Untuk informasi selengkapnya, lihat Membuat grup opsi.
4. Pilih grup opsi yang ingin Anda ubah, lalu pilih Tambahkan opsi.
5. Di periode Tambahkan opsi, lakukan hal berikut:

   a. Pilih TIMEZONE_FILE_AUTOUPGRADE.

   b. Untuk mengaktifkan opsi pada semua instans DB terkait segera setelah Anda menambahkannya, untuk Terapkan Langsung, pilih Ya. Jika Anda memilih Tidak (default), opsi diaktifkan untuk setiap instans DB terkait selama periode pemeliharaan berikutnya.
6. Jika pengaturan sesuai keinginan Anda, pilih Tambahkan opsi.

AWS CLI

Contoh berikut menggunakan perintah AWS CLI add-option-to-option-group untuk menambahkan `TIMEZONE_FILE_AUTOUPGRADE` opsi ke grup opsi yang disebut`myoptiongroup`.

Untuk Linux, macOS, atau Unix:

```
aws rds add-option-to-option-group \
    --option-group-name "myoptiongroup" \
    --options "OptionName=TIMEZONE_FILE_AUTOUPGRADE" \
    --apply-immediately
```

Untuk Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name "myoptiongroup" ^
    --options "OptionName=TIMEZONE_FILE_AUTOUPGRADE" ^
    --apply-immediately
```

Memeriksa data Anda setelah pembaruan file zona waktu

Sebaiknya periksa data Anda setelah memperbarui file zona waktu. Selama langkah persiapan, RDS for Oracle secara otomatis membuat tabel berikut:

- `rdsadmin.rds_dst_affected_tables`— Buat daftar tabel yang berisi data yang terpengaruh oleh pembaruan
- `rdsadmin.rds_dst_error_table`— Buat daftar kesalahan yang dihasilkan selama pembaruan

Tabel ini tidak tergantung pada tabel apa pun yang Anda buat di periode persiapan. Untuk melihat hasil pembaruan, kueri tabel sebagai berikut.

```
SELECT * FROM rdsadmin.rds_dst_affected_tables;
SELECT * FROM rdsadmin.rds_dst_error_table;
```

Untuk informasi selengkapnya tentang skema data yang terpengaruh dan tabel kesalahan, lihat FIND_AFFECTED_TABLES Procedure di dokumentasi Oracle.

Enkripsi Data Transparan Oracle

Amazon RDS mendukung Oracle Transparent Data Encryption (TDE), yakni sebuah fitur dari opsi Oracle Advanced Security yang tersedia di Oracle Enterprise Edition. Fitur ini secara otomatis mengenkripsi data sebelum ditulis ke penyimpanan dan secara otomatis mendekripsi data saat data dibaca dari penyimpanan.

TDE berguna dalam skenario di mana Anda perlu mengenkripsi data sensitif jika file data dan cadangan diperoleh oleh pihak ketiga. TDE juga berguna ketika Anda harus mematuhi peraturan terkait keamanan.

`TDE`Pilihannya persisten dan permanen. Jika Anda mengaitkan RDS untuk instans Oracle DB dengan grup opsi yang memiliki `TDE` opsi diaktifkan, Anda tidak dapat menonaktifkannya. Anda dapat mengubah grup opsi, tetapi grup opsi baru harus menyertakan `TDE` opsi. Untuk informasi selengkapnya tentang opsi persisten dan permanen, lihatOpsi persisten dan permanen.

> **Note**
>
> Anda tidak dapat membagikan snapshot DB yang menggunakan opsi TDE. Untuk informasi lebih lanjut tentang cara berbagi snapshot DB, lihat Berbagi snapshot DB.

Penjelasan rinci tentang TDE di Oracle Database berada di luar cakupan panduan ini. Untuk selengkapnya, lihat sumber Oracle Database berikut ini:

- Mengamankan data yang disimpan menggunakan Enkripsi Data Transparan dalam dokumentasi Oracle Database
- Keamanan tingkat lanjut Oracle dalam dokumentasi Oracle Database
- Keamanan canggih Oracle Praktik terbaik Enkripsi Data Transparan, yang merupakan whitepaper Oracle

Untuk informasi lebih lanjut tentang menggunakan TDE dengan RDS untuk Oracle, lihat blog berikut:

- Opsi Enkripsi Database Oracle di Amazon RDS
- Migrasi Amazon RDS berkemampuan TDE lintas akun untuk instans DB Amazon RDS for Oracle dengan mengurangi waktu henti menggunakan AWS DMS

Mode enkripsi TDE

Oracle Transparent Data Encryption mendukung dua mode enkripsi: enkripsi ruang tabel TDE dan enkripsi kolom TDE. Enkripsi ruang tabel TDE digunakan untuk mengenkripsi seluruh tabel aplikasi. Enkripsi kolom TDE digunakan untuk mengenkripsi elemen data individu yang berisi data sensitif. Anda juga dapat menerapkan solusi enkripsi hibrida yang menggunakan enkripsi ruang tabel dan enkripsi kolom TDE.

> **Note**
>
> Amazon RDS mengelola Oracle Wallet dan kunci master TDE untuk instans DB. Anda tidak perlu menyetel kunci enkripsi menggunakan perintah `ALTER SYSTEM set encryption key`.

Setelah Anda mengaktifkan `TDE` opsi, Anda dapat memeriksa status Oracle Wallet dengan menggunakan perintah berikut:

```
SELECT * FROM v$encryption_wallet;
```

Untuk membuat ruang tabel terenkripsi, gunakan perintah berikut:

```
CREATE TABLESPACE encrypt_ts ENCRYPTION DEFAULT STORAGE (ENCRYPT);
```

Untuk menentukan algoritma enkripsi, gunakan perintah berikut:

```
CREATE TABLESPACE encrypt_ts ENCRYPTION USING 'AES256' DEFAULT STORAGE (ENCRYPT);
```

Pernyataan sebelumnya untuk mengenkripsi tablespace sama seperti yang akan Anda gunakan pada database Oracle lokal.

Menentukan apakah instans DB Anda menggunakan TDE

Anda mungkin ingin menentukan apakah instans DB Anda dikaitkan dengan grup opsi yang mengaktifkan `TDE` opsi. Untuk melihat grup opsi yang terkait dengan instans DB, gunakan konsol RDS, describe-db-instance AWS CLI perintah, atau operasi API DescribedBInstances.

Menambahkan opsi TDE

Proses untuk menggunakan Enkripsi Data Transparan (TDE) Oracle dengan Amazon RDS adalah sebagai berikut:

1. Jika instans DB tidak terkait dengan grup opsi yang mengaktifkan `TDE` opsi, Anda harus membuat grup opsi dan menambahkan `TDE` opsi atau memodifikasi grup opsi terkait untuk menambahkan `TDE` opsi. Untuk informasi tentang membuat atau menyesuaikan grup opsi, lihat Menggunakan grup opsi. Untuk informasi tentang cara menambahkan opsi ke grup opsi, lihat Menambahkan opsi ke grup opsi.
2. Kaitkan instans DB dengan grup opsi dengan opsi TDE. Untuk informasi tentang cara mengaitkan instans DB dengan grup opsi, lihat Memodifikasi instans DB Amazon RDS.

Menyalin data Anda ke instans DB yang tidak menyertakan opsi TDE

Anda tidak dapat menghapus opsi TDE dari instans DB atau mengaitkannya dengan grup opsi yang tidak menyertakan opsi TDE. Untuk memigrasikan data Anda ke instance yang tidak menyertakan opsi TDE, lakukan hal berikut:

1. Dekripsi data pada instans DB Anda.
2. Salin data ke instans DB baru yang tidak terkait dengan grup opsi yang telah `TDE` diaktifkan.
3. Hapus instans DB asli Anda.

Anda dapat memberi nama instans baru dengan nama instans DB sebelumnya.

Menggunakan TDE dengan Oracle Data Pump

Anda dapat menggunakan Oracle Data Pump untuk mengimpor atau mengekspor file dump terenkripsi. Amazon RDS mendukung mode enkripsi kata sandi (`ENCRYPTION_MODE=PASSWORD`) untuk Oracle Data Pump. Amazon RDS tidak mendukung mode enkripsi transparan (`ENCRYPTION_MODE=TRANSPARENT`) untuk Oracle Data Pump. Lihat informasi yang lebih lengkap di Mengimpor menggunakan Oracle Data Pump.

Oracle UTL_MAIL

Amazon RDS mendukung Oracle UTL_MAIL melalui penggunaan opsi UTL_MAIL dan server SMTP. Anda dapat mengirim email langsung dari basis data Anda menggunakan paket UTL_MAIL. Amazon RDS mendukung UTL_MAIL untuk versi Oracle berikut ini:

- Oracle Database 21c (21.0.0.0), semua versi
- Oracle Database 19c (19.0.0.0), semua versi
- Oracle Database 12c Rilis 2 (12.2), semua versi
- Oracle Database 12c Rilis 1 (12.1), versi 12.1.0.2.v5 dan yang lebih baru

Berikut adalah beberapa batasan dalam menggunakan UTL_MAIL:

- UTL_MAIL tidak mendukung Keamanan Lapisan Pengangkutan (TLS) dan oleh karena itu email tidak dienkripsi.

  Untuk terhubung secara aman ke sumber daya SSL/TLS jarak jauh dengan membuat dan mengunggah dompet Oracle kustom, ikuti petunjuk di Mengonfigurasi akses UTL_HTTP menggunakan sertifikat dan dompet Oracle.

  Sertifikat khusus yang diperlukan untuk dompet Anda berbeda menurut layanan. Untuk layanan AWS, ini biasanya dapat ditemukan di repositori layanan kepercayaan Amazon.
- UTL_MAIL tidak mendukung autentikasi dengan server SMTP.
- Anda hanya dapat mengirim satu lampiran dalam satu email.
- Anda tidak dapat mengirim lampiran yang lebih besar dari 32 K.
- Anda hanya dapat menggunakan pengodean karakter ASCII dan Extended Binary Coded Decimal Interchange Code (EBCDIC).
- Port SMTP (25) dibatasi berdasarkan kebijakan pemilik antarmuka jaringan elastis.

Saat Anda mengaktifkan UTL_MAIL, hanya pengguna master untuk instans DB Anda yang diberikan hak istimewa eksekusi. Jika perlu, pengguna master dapat memberikan hak eksekusi kepada pengguna lain sehingga mereka dapat menggunakan UTL_MAIL.

> **Important**
>
> Kami menyarankan Anda mengaktifkan fitur audit bawaan Oracle untuk melacak penggunaan prosedur UTL_MAIL.

Prasyarat untuk Oracle UTL_MAIL

Berikut adalah prasyarat untuk menggunakan Oracle UTL_MAIL:

- Satu server SMTP atau lebih, dan alamat IP yang sesuai atau nama Domain Name Server (DNS) publik atau privat. Untuk informasi lebih lanjut tentang nama DNS privat yang diselesaikan melalui server DNS kustom, lihat Menyiapkan server DNS kustom.
- Untuk versi Oracle sebelum 12c, instans DB Anda juga harus menggunakan opsi XML DB. Untuk informasi selengkapnya, lihat DB XML Oracle.

Menambahkan opsi Oracle UTL_MAIL

Proses umum untuk menambahkan opsi UTL_MAIL ke instans DB adalah sebagai berikut:

1. Buat grup opsi baru, atau salin atau ubah grup opsi yang ada.
2. Tambahkan opsi tersebut ke grup opsi.
3. Kaitkan grup opsi tersebut dengan instans DB.

Setelah Anda menambahkan opsi UTL_MAIL, setelah grup opsi aktif, UTL_MAIL langsung aktif.

Untuk menambahkan opsi UTL_MAIL ke instans DB

1. Tentukan grup opsi yang ingin Anda gunakan. Anda dapat membuat grup opsi baru atau menggunakan grup opsi yang ada. Jika Anda ingin menggunakan grup opsi yang ada, lanjutkan ke langkah berikutnya. Jika tidak, buat grup opsi DB kustom dengan pengaturan berikut:

   a. Untuk Mesin, pilih edisi Oracle yang ingin Anda gunakan.

   b. Untuk Versi mesin utama, pilih versi instans DB Anda.

   Untuk informasi selengkapnya, lihat Membuat grup opsi.

2. Tambahkan opsi UTL_MAIL ke grup opsi. Untuk informasi selengkapnya tentang cara menambahkan opsi, lihat Menambahkan opsi ke grup opsi.
3. Terapkan grup opsi ke instans DB baru atau yang sudah ada:
   - Untuk instans DB baru, Anda menerapkan grup opsi saat Anda meluncurkan instans. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.
   - Untuk instans DB yang ada, Anda menerapkan grup opsi dengan memodifikasi instans dan melampirkan grup opsi baru. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Menggunakan Oracle UTL_MAIL

Setelah Anda mengaktifkan opsi UTL_MAIL, Anda harus mengonfigurasi server SMTP sebelum Anda dapat mulai menggunakannya.

Anda mengonfigurasi server SMTP dengan mengatur parameter SMTP_OUT_SERVER ke alamat IP atau nama DNS publik yang valid. Untuk parameter SMTP_OUT_SERVER, Anda dapat menentukan daftar alamat beberapa server yang dipisahkan koma. Jika server pertama tidak tersedia, UTL_MAIL mencoba server berikutnya, dan seterusnya.

Anda dapat mengatur SMTP_OUT_SERVER default untuk instans DB menggunakan grup parameter DB. Anda dapat mengatur parameter SMTP_OUT_SERVER untuk sebuah sesi dengan menjalankan kode berikut di basis data Anda pada instans DB Anda.

```
ALTER SESSION SET smtp_out_server = mailserver.domain.com:25;
```

Setelah opsi UTL_MAIL diaktifkan, dan SMTP_OUT_SERVER Anda dikonfigurasi, Anda dapat mengirim email menggunakan prosedur `SEND`. Untuk informasi lebih lanjut, lihat UTL_MAIL dalam dokumentasi Oracle.

Menghapus opsi Oracle UTL_MAIL

Anda dapat menghapus Oracle UTL_MAIL dari instans DB.

Untuk menghapus UTL_MAIL dari instans DB, lakukan salah satu hal berikut:

- Untuk menghapus UTL_MAIL dari beberapa instans DB, hapus opsi UTL_MAIL dari grup opsi yang mencakupnya. Perubahan ini memengaruhi semua instans DB yang menggunakan grup opsi tersebut. Untuk informasi selengkapnya, lihat Menghapus opsi dari grup opsi.
- Untuk menghapus UTL_MAIL dari satu instans DB, modifikasi instans DB dan tentukan grup opsi lain yang tidak menyertakan opsi UTL_MAIL. Anda dapat menentukan grup opsi default (kosong) atau grup opsi kustom yang berbeda. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Pemecahan Masalah

Berikut ini adalah masalah yang mungkin Anda temui saat menggunakan UTL_MAIL dengan Amazon RDS.

- Throttling Port SMTP (25) dibatasi berdasarkan kebijakan pemilik antarmuka jaringan elastis. Jika Anda berhasil mengirim email menggunakan UTL_MAIL dan melihat kesalahan `ORA-29278: SMTP transient error: 421 Service not available`, Anda mungkin sedang dibatasi. Jika Anda mengalami throttling dengan pengiriman email, sebaiknya Anda menerapkan algoritme mundur. Untuk informasi selengkapnya tentang algoritme mundur, lihat Percobaan ulang kesalahan dan mundur eksponensial di AWS dan Cara menangani kesalahan "throttling - Laju pengiriman maksimum terlampaui".

  Anda dapat meminta agar throttle ini dihapus. Untuk informasi lebih lanjut, lihat Bagaimana cara menghapus throttle pada port 25 dari instans EC2 saya?.

DB XML Oracle

DB XML Oracle menambahkan dukungan XML asli ke instans DB Anda. Dengan DB XML, Anda dapat menyimpan dan mengambil XML terstruktur atau tidak terstruktur, selain data relasional. DB XML sudah diinstal sebelumnya di Oracle versi 12c dan yang lebih baru.

Meng-upgrade mesin DB Oracle

Ketika Amazon RDS mendukung versi baru Oracle Database, Anda dapat meng-upgrade instans DB Anda ke versi baru. Untuk informasi tentang versi Oracle yang tersedia di Amazon RDS, lihat Catatan Rilis Amazon RDS for Oracle.

> **Important**
>
> RDS for Oracle Database 11g, 12c, dan 18c tidak lagi didukung. Jika Anda mengelola snapshot Oracle Database 11g, 12c, atau 18c, Anda dapat meng-upgrade-nya ke rilis yang lebih baru. Untuk informasi selengkapnya, lihat Meng-upgrade snapshot DB Oracle.

Topik

Gambaran umum upgrade mesin RDS for Oracle

Sebelum meng-upgrade instans DB RDS for Oracle Anda, pahami konsep-konsep berikut.

Topik

Upgrade versi mayor dan minor

Versi mayor adalah rilis mayor Oracle Database yang terjadi setiap 1-2 tahun. Contoh rilis mayor adalah Oracle Database 19c dan Oracle Database 21c.

Versi minor, yang juga disebut Pembaruan Rilis (RU), biasanya dirilis oleh Oracle setiap kuartal. Versi minor berisi penyempurnaan fitur kecil dan perbaikan bug. Contoh versi minor adalah 21.0.0.0.ru-2023-10.rur-2023-10.r1 dan 19.0.0.0.ru-2023-10.rur-2023-10.r1. Untuk informasi selengkapnya, lihat Catatan rilis untuk Amazon Relational Database Service (Amazon RDS) for Oracle.

RDS for Oracle mendukung upgrade berikut untuk instans DB.

Jenis upgrade	Kompatibilitas aplikasi	Metode upgrade	Contoh jalur upgrade
Versi mayor	Upgrade versi mayor dapat menerapkan perubahan yang tidak kompatibel dengan aplikasi yang ada.	Hanya manual	Dari Basis Data Oracle 19c ke Basis Data Oracle 21c
Versi minor	Upgrade versi minor hanya mencakup perubahan yang memiliki kompatibilitas mundur dengan aplikasi yang ada.	Otomatis atau manual	Dari 21.0.0.0.ru-2023-07.rur-2022-07.r1 ke 21.0.0.0.ru-2023-10.rur-2022-10.r1

> **Important**
>
> Saat Anda meng-upgrade mesin DB Anda, akan terjadi pemadaman. Waktu pemadaman bergantung pada versi mesin dan ukuran instans Anda.
> Pastikan bahwa Anda menguji secara menyeluruh setiap upgrade untuk memverifikasi bahwa aplikasi Anda berfungsi dengan benar sebelum menerapkan upgrade ke basis data produksi Anda. Untuk informasi selengkapnya, lihat Menguji upgrade DB Oracle.

Tanggal dukungan yang diharapkan untuk rilis mayor RDS for Oracle

RDS for Oracle versi mayor tetap tersedia setidaknya sampai akhir tanggal dukungan untuk versi rilis Oracle Database yang sesuai. Anda dapat menggunakan tanggal berikut untuk merencanakan siklus pengujian dan upgrade Anda. Tanggal-tanggal ini merepresentasikan tanggal paling awal saat upgrade ke versi yang lebih baru mungkin diperlukan. Jika Amazon memperpanjang dukungan untuk sebuah versi RDS for Oracle lebih lama dari yang dinyatakan semula, kami berencana untuk memperbarui tabel ini untuk mencerminkan tanggal yang dimundurkan tersebut.

Versi rilis mayor Oracle Database	Tanggal yang diharapkan untuk meng-upgrade ke versi yang lebih baru
Oracle Database 19c	30 April 2026 dengan BYOL Premier Support (biaya dibebaskan untuk Extended Support) 30 April 2027 dengan BYOL Extended Support (biaya tambahan) atau Perjanjian Lisensi Tidak Terbatas 30 April 2027 dengan Termasuk Lisensi (LI)
Oracle Database 21c	30 April 2024 (tidak tersedia untuk Extended Support)

Sebelum kami meminta Anda meng-upgrade ke versi mayor yang lebih baru, kami akan mengingatkan Anda setidaknya 12 bulan sebelumnya. Kami akan menjelaskan proses upgrade, termasuk milestone penting, dampaknya pada instans DB Anda, dan tindakan yang disarankan. Anda harus menguji aplikasi Anda secara menyeluruh dengan versi RDS for Oracle baru sebelum Anda meng-upgrade ke versi mayor.

Setelah periode pemberitahuan awal ini, upgrade otomatis ke versi mayor berikutnya dapat diterapkan ke instans DB RDS for Oracle yang masih menjalankan versi yang lebih lama. Jika demikian, upgrade dimulai selama periode pemeliharaan terjadwal.

Untuk informasi selengkapnya, lihat Release Schedule of Current Database Releases di My Oracle Support.

Manajemen versi mesin Oracle

Dengan manajemen versi mesin DB, Anda dapat mengontrol kapan dan bagaimana mesin basis data Anda di-patch dan di-upgrade. Dengan fitur ini, Anda mendapatkan fleksibilitas untuk

mempertahankan kompatibilitas dengan versi patch mesin basis data. Anda juga dapat menguji versi patch baru RDS for Oracle untuk memastikannya berjalan efektif dengan aplikasi Anda sebelum menerapkannya dalam produksi. Selain itu, Anda meng-upgrade versi sesuai ketentuan dan jadwal Anda sendiri.

> **Note**
>
> Amazon RDS secara berkala mengumpulkan patch Oracle Database resmi menggunakan versi DB spesifik Amazon RDS. Untuk melihat daftar patch Oracle yang terdapat dalam versi mesin spesifik Amazon RDS Oracle, kunjungi Catatan Rilis Amazon RDS for Oracle.

Snapshot otomatis selama upgrade mesin

Selama upgrade instans DB Oracle, snapshot menawarkan perlindungan terhadap masalah upgrade. Jika periode retensi cadangan untuk instans DB Anda lebih besar dari 0, Amazon RDS mengambil snapshot DB berikut selama upgrade:

1. Snapshot instans DB sebelum perubahan upgrade dibuat. Jika upgrade gagal, Anda dapat memulihkan snapshot ini untuk membuat instans DB yang menjalankan versi lama.
2. Snapshot instans DB setelah upgrade selesai.

> **Note**
>
> Untuk mengubah periode retensi cadangan Anda, lihat Memodifikasi instans DB Amazon RDS.

Setelah upgrade, Anda tidak dapat kembali ke versi mesin yang sebelumnya. Namun, Anda dapat membuat instans DB Oracle baru dengan memulihkan snapshot yang diambil sebelum upgrade.

Upgrade Oracle dalam deployment Multi-AZ

Jika instans DB Anda berada dalam deployment Multi-AZ, Amazon RDS meng-upgrade replika primer dan siaga. Jika pembaruan sistem operasi tidak diperlukan, upgrade primer dan siaga terjadi secara bersamaan. Instans tidak tersedia hingga upgrade selesai.

Jika pembaruan sistem operasi diperlukan dalam penyebaran multi-AZ, Amazon RDS menerapkan pembaruan saat Anda meminta peningkatan basis data. Amazon RDS melakukan langkah-langkah berikut:

1. Memperbarui sistem operasi pada instans DB siaga saat ini.
2. Gagal atas instans DB primer ke instans DB siaga.
3. Upgrade versi database pada instans DB primer baru, yang sebelumnya merupakan instance siaga. Database utama tidak tersedia selama upgrade.
4. Memperbarui sistem operasi pada instans DB siaga baru, yang sebelumnya merupakan instans DB utama.
5. Upgrade versi database pada instans DB siaga baru.
6. Gagal atas instans DB primer baru kembali ke instans DB primer asli, dan instans DB siaga baru kembali ke instans DB siaga asli. Dengan demikian, Amazon RDS mengembalikan konfigurasi replikasi ke keadaan semula.

Upgrade Oracle untuk replika baca

Versi mesin DB Oracle dari instans DB sumber dan semua replika baca harus sama. Amazon RDS melakukan upgrade dalam tahap-tahap berikut:

1. Meng-upgrade instans DB sumber. Replika baca tersedia selama tahap ini.
2. Meng-upgrade replika baca secara paralel, terlepas dari periode pemeliharaan replika. DB sumber tersedia selama tahap ini.

Untuk upgrade versi mayor replika baca lintas Wilayah, Amazon RDS melakukan tindakan tambahan:

- Menghasilkan grup opsi untuk versi target secara otomatis
- Menyalin semua opsi dan pengaturan opsi dari grup opsi asli ke grup opsi baru
- Mengaitkan replika baca lintas Wilayah yang di-upgrade dengan grup opsi baru

Upgrade Oracle untuk instans DB mikro

Kami tidak menyarankan upgrade basis data yang berjalan pada instans DB mikro. Karena instans ini memiliki CPU yang terbatas, upgrade dapat memerlukan waktu berjam-jam.

Anda dapat meng-upgrade instans DB mikro dengan jumlah penyimpanan kecil (10–20 GiB) dengan menyalin data Anda menggunakan Data Pump. Sebelum memigrasikan instans DB produksi Anda, sebaiknya Anda mengujinya dengan menyalin data menggunakan Data Pump.

Upgrade versi mayor Oracle

Untuk melakukan upgrade versi mayor, ubah instans DB secara manual. Upgrade versi mayor tidak terjadi secara otomatis.

> **Important**
>
> Pastikan bahwa Anda menguji secara menyeluruh setiap upgrade untuk memverifikasi bahwa aplikasi Anda berfungsi dengan benar sebelum menerapkan upgrade ke basis data produksi Anda. Untuk informasi selengkapnya, lihat Menguji upgrade DB Oracle.

Topik

Versi yang didukung untuk upgrade mayor

Amazon RDS mendukung upgrade versi mayor berikut.

Versi saat ini	Upgrade yang didukung
19.0.0.0 menggunakan arsitektur CDB	21.0.0.0

Versi mayor Oracle Database harus di-upgrade ke Pembaruan Rilis (RU) yang dirilis pada bulan yang sama atau berikutnya. Downgrade versi mayor tidak didukung untuk versi Oracle Database apa pun.

Kelas instans yang didukung untuk upgrade mayor

Instans DB Oracle Anda saat ini mungkin berjalan pada kelas instans DB yang tidak didukung untuk versi yang akan menjadi target upgrade Anda. Dalam kasus tersebut, sebelum Anda melakukan

upgrade, migrasikan instans DB ke kelas instans DB yang didukung. Untuk informasi selengkapnya tentang kelas instans DB yang didukung untuk setiap versi dan edisi Amazon RDS for Oracle, lihat Kelas instans DB .

Mengumpulkan statistik sebelum upgrade mayor

Sebelum Anda melakukan upgrade versi mayor, Oracle menyarankan agar Anda mengumpulkan statistik pengoptimisasi pada instans DB yang Anda upgrade. Tindakan ini dapat mengurangi waktu henti instans DB selama upgrade.

Untuk mengumpulkan statistik pengoptimisasi, hubungkan ke instans DB sebagai pengguna master, dan jalankan prosedur `DBMS_STATS.GATHER_DICTIONARY_STATS`, seperti pada contoh berikut.

```
EXEC DBMS_STATS.GATHER_DICTIONARY_STATS;
```

Untuk informasi selengkapnya, lihat Gathering optimizer statistics to decrease Oracle database downtime dalam dokumentasi Oracle.

Mengizinkan upgrade mayor

Upgrade versi mesin mayor mungkin tidak kompatibel dengan aplikasi Anda. Upgrade tidak dapat dikembalikan. Jika Anda menentukan versi mayor untuk parameter EngineVersion yang berbeda dari versi mayor saat ini, Anda harus mengizinkan upgrade versi mayor.

Jika Anda meng-upgrade versi mayor menggunakan perintah CLI modify-db-instance, tentukan `--allow-major-version-upgrade`. Pengaturan ini tidak persisten, jadi Anda harus menentukan `--allow-major-version-upgrade` setiap kali Anda melakukan upgrade mayor. Parameter ini tidak berdampak pada upgrade versi mesin minor. Untuk informasi selengkapnya, lihat Meng-upgrade versi mesin instans DB.

Jika meng-upgrade versi mayor menggunakan konsol, Anda tidak perlu memilih opsi untuk mengizinkan upgrade. Sebagai gantinya, konsol akan menampilkan peringatan bahwa upgrade mayor tidak dapat dikembalikan.

Upgrade versi minor Oracle

Upgrade versi minor menerapkan Pembaruan Set Patch (PSU) Oracle Database atau Pembaruan Rilis (RU) ke versi mesin mayor. Misalnya, jika instans DB Anda menjalankan Oracle Database versi

mayor 21c dan versi minor 21.0.0.0.ru-2022-07.rur-2022-07.r1, Anda dapat meng-upgrade ke versi minor 21.0.0.0.ru-2022-10.rur-2022-10.r1. Biasanya, versi minor baru tersedia setiap kuartal.

> **Note**
>
> RDS for Oracle tidak mendukung downgrade versi minor.

Anda dapat meng-upgrade mesin DB Anda ke versi minor secara manual atau otomatis. Untuk mempelajari cara meng-upgrade secara manual, lihat Meng-upgrade versi mesin secara manual. Untuk mempelajari cara mengonfigurasi upgrade otomatis, lihat Meng-upgrade versi mesin minor secara otomatis. Baik Anda meng-upgrade secara manual maupun otomatis, upgrade versi minor memerlukan waktu henti. Ingatlah hal ini saat merencanakan upgrade Anda.

> **Important**
>
> Pastikan bahwa Anda menguji secara menyeluruh setiap upgrade untuk memverifikasi bahwa aplikasi Anda berfungsi dengan benar sebelum menerapkan upgrade ke basis data produksi Anda. Untuk informasi selengkapnya, lihat Menguji upgrade DB Oracle.

Topik

Mengaktifkan upgrade versi minor otomatis untuk Oracle

Dalam upgrade versi minor otomatis, RDS menerapkan versi minor terbaru yang tersedia ke basis data Oracle Anda tanpa intervensi manual. Instans DB Amazon RDS for Oracle menjadwalkan upgrade selama periode pemeliharaan berikutnya dalam kondisi berikut:

- Instans DB Anda telah mengaktifkan opsi Peningkatan versi minor otomatis.
- Instans DB Anda belum menjalankan versi mesin DB minor terbaru.
- Instans DB Anda tidak memiliki upgrade tertunda.

Untuk mempelajari cara mengaktifkan upgrade otomatis, lihat Meng-upgrade versi mesin minor secara otomatis.

Sebelum upgrade versi minor otomatis untuk Oracle dijadwalkan

RDS menerbitkan pemberitahuan awal sebelum mulai menjadwalkan upgrade otomatis. Anda dapat menemukan notifikasi di tab Pemeliharaan & pencadangan pada halaman detail basis data. Pesannya memiliki format berikut:

```
An automatic minor version upgrade to engine version will become available
 on availability-date and will be applied during a subsequent maintenance window.
```

availability-date dalam pesan di atas adalah tanggal ketika RDS mulai menjadwalkan upgrade untuk instans DB di Wilayah AWS Anda. Ini bukanlah tanggal saat upgrade instans DB Anda dijadwalkan terjadi.

Anda juga bisa mendapatkan tanggal ketersediaan upgrade dengan menggunakan perintah `describe-pending-maintenance-actions` di AWS CLI, seperti yang ditunjukkan pada contoh berikut:

```
aws rds describe-pending-maintenance-actions

{
    "PendingMaintenanceActions": [
        {
            "ResourceIdentifier": "arn:aws:rds:us-east-1:123456789012:db:orclinst1",
            "PendingMaintenanceActionDetails": [
                {
                    "Action": "db-upgrade",
                    "Description": "Automatic minor version upgrade to
 21.0.0.0.ru-2022-10.rur-2022-10.r1",
                    "CurrentApplyDate": "2022-12-02T08:10:00Z",
                    "OptInStatus": "next-maintenance"
                }
            ]
        }, ...
```

Tabel berikut menjelaskan opsi untuk setiap jenis pesan tindakan pemeliharaan tertunda.

Pesan tindakan pemeliharaan tertunda	Kapan pesan muncul	Dapat diterapkan pada periode pemeliharaan berikutnya?	Dapat segera diterapkan?	Dapat dibatalkan?
Upgrade versi minor otomatis ke *versi mesin* akan tersedia pada *tanggal ketersediaan* dan harus diterapkan selama periode pemeliharaan berikutnya.	4-6 minggu sebelum upgrade otomatis dijadwalkan.	Ya	Ya	Ya
Upgrade versi minor otomatis ke *versi mesin*	Pada atau setelah *tanggal ketersediaan*. RDS secara otomatis menerapkan upgrade ini pada periode pemeliharaan instans DB berikutnya.	Ya	Ya	Tidak

Untuk informasi selengkapnya tentang describe-pending-maintenance-actions, lihat Referensi Perintah AWS CLI.

Waktu saat RDS menjadwalkan upgrade versi minor otomatis untuk Oracle

Ketika tanggal ketersediaan untuk upgrade otomatis tiba, RDS mulai menjadwalkan upgrade. Untuk sebagian besar Wilayah AWS, RDS menjadwalkan upgrade Anda ke RU kuartalan terbaru sekitar empat hingga enam minggu setelah tanggal ketersediaan. Tanggal yang dijadwalkan bervariasi tergantung pada Wilayah AWS dan faktor lainnya. Untuk informasi selengkapnya tentang RU dan RUR, lihat Catatan Rilis Amazon RDS for Oracle.

Saat RDS menjadwalkan upgrade, notifikasi berikut muncul di tab Pemeliharaan & pencadangan pada halaman detail basis data:

```
Automatic minor version upgrade to engine-version
```

Pesan di atas menunjukkan bahwa RDS telah menjadwalkan mesin DB Anda untuk di-upgrade pada periode pemeliharaan berikutnya.

Mengelola upgrade versi minor otomatis untuk Oracle

Ketika versi minor baru tersedia, Anda dapat meng-upgrade instans DB Anda ke versi ini secara manual. Contoh berikut meng-upgrade instans DB bernama `orclinst1` dengan segera:

```
aws rds apply-pending-maintenance-action \
    --resource-identifier arn:aws:rds:us-east-1:123456789012:db:orclinst1 \
    --apply-action db-upgrade \
    --opt-in-type immediate
```

Untuk membatalkan upgrade versi minor otomatis yang belum dijadwalkan, atur `opt-in-type` ke `undo-opt-in`, seperti pada contoh berikut:

```
aws rds apply-pending-maintenance-action \
    --resource-identifier arn:aws:rds:us-east-1:123456789012:db:orclinst1 \
    --apply-action db-upgrade \
    --opt-in-type undo-opt-in
```

Jika RDS telah menjadwalkan upgrade untuk instans DB Anda, Anda tidak dapat menggunakan `apply-pending-maintenance-action` untuk membatalkannya. Namun, Anda dapat memodifikasi instans DB Anda dan menonaktifkan fitur upgrade minor otomatis, yang kemudian akan membatalkan jadwal upgrade.

Untuk mempelajari cara menonaktifkan upgrade versi minor otomatis, lihat Meng-upgrade versi mesin minor secara otomatis. Untuk informasi selengkapnya tentang apply-pending-maintenance-action, lihat Referensi Perintah AWS CLI.

Pertimbangan untuk upgrade DB Oracle

Sebelum meng-upgrade instans Oracle Anda, tinjau informasi berikut.

Topik

Pertimbangan Oracle Multitenant

Tabel berikut menjelaskan arsitektur yang didukung dalam berbagai rilis.

Rilis Oracle Database	Status dukungan RDS	Arsitektur
Oracle Database 21c	Didukung	Hanya CDB
Oracle Database 19c	Didukung	CDB atau non-CDB
Oracle Database 12c Rilis 2 (12.2)	Tidak didukung	Hanya non-CDB
Oracle Database 12c Rilis 1 (12.1)	Tidak didukung	Hanya non-CDB

Tabel berikut menjelaskan jalur upgrade yang didukung dan tidak didukung.

Jalur upgrade	Didukung?
Non-CDB ke non-CDB	Ya
CDB ke CDB	Ya
Non-CDB ke CDB	Tidak
CDB ke non-CDB	Tidak

Untuk informasi selengkapnya tentang Multi-penghuni Oracle di RDS for Oracle, lihat Konfigurasi satu penghuni pada arsitektur CDB.

Pertimbangan grup opsi

Jika instans DB Anda menggunakan grup opsi kustom, terkadang Amazon RDS tidak dapat secara otomatis menetapkan grup opsi yang baru. Misalnya, situasi ini terjadi ketika Anda meng-upgrade ke versi mayor baru. Dalam kasus tersebut, tentukan grup opsi baru saat meng-upgrade. Kami sarankan Anda membuat grup opsi baru, dan menambahkan opsi yang sama seperti dalam grup opsi kustom yang ada.

Untuk informasi selengkapnya, lihat Membuat grup opsi atau Menyalin grup opsi.

Jika instans DB Anda menggunakan grup opsi kustom yang berisi opsi APEX, Anda terkadang dapat mengurangi waktu upgrade. Untuk melakukannya, upgrade versi APEX Anda secara bersamaan dengan instans DB Anda. Untuk informasi selengkapnya, lihat Memutakhirkan versi APEX.

Pertimbangan grup parameter

Jika instans DB Anda menggunakan grup opsi kustom, terkadang Amazon RDS tidak dapat secara otomatis menetapkan grup parameter yang baru untuk instans DB Anda. Misalnya, situasi ini terjadi ketika Anda meng-upgrade ke versi mayor baru. Dalam kasus tersebut, pastikan untuk menentukan grup parameter baru saat Anda meng-upgrade. Kami menyarankan agar Anda membuat grup parameter baru, dan mengonfigurasi parameter seperti dalam grup parameter kustom yang ada.

Untuk informasi selengkapnya, lihat Membuat grup parameter DB atau Menyalin grup parameter DB.

Pertimbangan zona waktu

Anda dapat menggunakan opsi zona waktu untuk mengubah zona waktu sistem yang digunakan oleh instans DB Oracle Anda. Misalnya, Anda dapat mengubah zona waktu instans DB agar kompatibel dengan lingkungan on-premise, atau aplikasi warisan. Opsi zona waktu mengubah zona waktu di tingkat host. Amazon RDS for Oracle memperbarui zona waktu sistem secara otomatis sepanjang tahun. Untuk informasi selengkapnya tentang zona waktu sistem, lihat Zona waktu Oracle.

Saat Anda membuat instans DB Oracle, basis data secara otomatis menetapkan zona waktu basis data. Zona waktu basis data juga dikenal sebagai zona waktu Daylight Saving Time (DST). Zona waktu basis data berbeda dari zona waktu sistem.

Setiap rilis, set patch, atau patch individu Oracle Database mungkin mencakup versi DST baru. Patch ini mencerminkan perubahan dalam aturan transisi untuk berbagai wilayah zona waktu. Misalnya, pemerintah dapat mengubah kapan DST mulai berlaku. Perubahan aturan DST dapat memengaruhi data yang ada dari jenis data `TIMESTAMP WITH TIME ZONE`.

Jika Anda meng-upgrade instans DB RDS for Oracle, Amazon RDS tidak meng-upgrade file zona waktu basis data secara otomatis. Untuk meng-upgrade file zona waktu secara otomatis, Anda dapat menyertakan opsi `TIMEZONE_FILE_AUTOUPGRADE` dalam grup opsi yang terkait dengan instans DB Anda selama atau setelah upgrade versi mesin. Untuk informasi selengkapnya, lihat Pemutakhiran otomatis file zona waktu Oracle.

Alternatifnya, untuk meng-upgrade file zona waktu basis data secara manual, buat instans Oracle DB baru yang memiliki patch DST yang diinginkan. Namun, kami menyarankan Anda meng-upgrade file zona waktu basis data menggunakan opsi `TIMEZONE_FILE_AUTOUPGRADE`.

Setelah meng-upgrade file zona waktu, migrasikan data dari instans Anda saat ini ke instans baru. Anda dapat memigrasikan data menggunakan beberapa teknik, termasuk teknik berikut:

- AWS Database Migration Service
- Oracle GoldenGate
- Oracle Data Pump
- Ekspor/Impor Asli (tidak didukung untuk penggunaan umum)

> **Note**
>
> Saat Anda memigrasikan data menggunakan Oracle Data Pump, utilitas tersebut akan menampilkan kesalahan ORA-39405 ketika versi zona waktu target lebih rendah daripada versi zona waktu sumber.

Untuk informasi selengkapnya, lihat TIMESTAMP WITH TIMEZONE restrictions dalam dokumentasi Oracle.

Menguji upgrade DB Oracle

Sebelum meng-upgrade instans DB Anda ke versi mayor, uji basis data dan semua aplikasi yang mengakses basis data tersebut terkait kompatibilitas dengan versi baru. Kami menyarankan agar Anda menggunakan prosedur berikut.

Untuk menguji upgrade versi mayor

1. Tinjau dokumentasi upgrade Oracle untuk versi baru mesin basis data guna melihat apakah ada masalah kompatibilitas yang mungkin memengaruhi basis data atau aplikasi Anda. Untuk informasi selengkapnya, lihat Database Upgrade Guide dalam dokumentasi Oracle.
2. Jika instans DB Anda menggunakan grup opsi kustom, buat grup opsi baru yang kompatibel dengan versi baru yang menjadi target upgrade Anda. Untuk informasi selengkapnya, lihat Pertimbangan grup opsi.
3. Jika instans DB Anda menggunakan grup parameter kustom, buat grup parameter baru yang kompatibel dengan versi baru yang menjadi target upgrade Anda. Untuk informasi selengkapnya, lihat Pertimbangan grup parameter.
4. Buat snapshot DB dari instans DB yang akan di-upgrade. Untuk informasi selengkapnya, lihat Membuat snapshot DB untuk instans DB Single-AZ.
5. Pulihkan snapshot DB untuk membuat instans DB uji baru. Untuk informasi selengkapnya, lihat Memulihkan dari snapshot DB.
6. Modifikasi instans DB uji baru ini untuk di-upgrade ke versi baru menggunakan salah satu metode berikut:

   - Konsol
   - AWS CLI
   - API RDS
7. Lakukan pengujian:

   - Jalankan pengujian jaminan kualitas terhadap instans DB yang di-upgrade sebanyak yang diperlukan untuk memastikan bahwa basis data dan aplikasi Anda berfungsi baik dengan versi baru.
   - Terapkan setiap pengujian baru yang diperlukan untuk mengevaluasi dampak dari masalah kompatibilitas yang Anda identifikasi dalam langkah 1.
   - Uji semua prosedur tersimpan, fungsi, dan pemicu.
   - Arahkan versi pengujian aplikasi Anda ke instans DB yang di-upgrade. Verifikasi bahwa aplikasi berfungsi baik dengan versi baru.
   - Evaluasi penyimpanan yang digunakan oleh instans yang di-upgrade untuk menentukan apakah upgrade memerlukan penyimpanan tambahan. Anda mungkin perlu memilih kelas instans yang lebih besar untuk mendukung versi baru dalam produksi. Untuk informasi selengkapnya, lihat Kelas instans DB .

8. Jika semua pengujian berhasil, upgrade instans DB produksi Anda. Sebaiknya Anda mengonfirmasi bahwa instans DB berfungsi dengan benar sebelum mengizinkan operasi tulis ke instans DB.

Meng-upgrade instans DB Oracle

Untuk mempelajari cara meng-upgrade instans Oracle, lihat Meng-upgrade versi mesin instans DB.

Meng-upgrade snapshot DB Oracle

Jika Anda sudah memiliki snapshot DB manual, Anda dapat meng-upgrade-nya ke versi lebih baru dari mesin basis data Oracle.

Ketika Oracle berhenti menyediakan patch untuk suatu versi, dan Amazon RDS meniadakan versi tersebut, Anda dapat meng-upgrade snapshot Anda yang sesuai dengan versi yang ditiadakan. Untuk informasi selengkapnya, lihat Manajemen versi mesin Oracle.

Amazon RDS mendukung upgrade snapshot di semua Wilayah AWS.

Konsol

Untuk meng-upgrade snapshot DB Oracle

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Snapshot, lalu pilih snapshot DB yang ingin Anda upgrade.
3. Untuk Tindakan, pilih Tingkatkan snapshot. Halaman Tingkatkan snapshot muncul.
4. Pilih Versi mesin baru untuk meng-upgrade snapshot.
5. (Opsional) Untuk Grup opsi, pilih grup opsi untuk snapshot DB yang di-upgrade. Pertimbangan grup opsi yang berlaku saat mengupgrade snapshot DB akan sama seperti saat mengupgrade instans DB. Untuk informasi selengkapnya, lihat Pertimbangan grup opsi.
6. Pilih Simpan perubahan untuk menyimpan perubahan Anda.

   Selama proses upgrade, semua tindakan snapshot dinonaktifkan untuk snapshot DB ini. Selain itu, status snapshot DB berubah dari tersedia menjadi meningkatkan, lalu berubah menjadi aktif setelah selesai. Jika snapshot DB tidak dapat di-upgrade karena masalah kerusakan snapshot, status berubah menjadi tidak tersedia. Anda tidak dapat memulihkan snapshot dari status ini.

> **Note**
>
> Jika upgrade snapshot DB gagal, snapshot di-rollback ke kondisi awal sesuai dengan versi asli.

AWS CLI

Untuk memutakhirkan snapshot Oracle DB dengan menggunakanAWS CLI, panggil modify-db-snapshotperintah dengan parameter berikut:

- `--db-snapshot-identifier` – Nama snapshot DB.
- `--engine-version` – Versi untuk meng-upgrade snapshot.

Anda mungkin juga perlu memasukkan parameter berikut. Pertimbangan grup opsi yang berlaku saat mengupgrade snapshot DB akan sama seperti saat mengupgrade instans DB. Untuk informasi selengkapnya, lihat Pertimbangan grup opsi.

- `--option-group-name` – Grup opsi untuk snapshot DB yang di-upgrade.

Example

Contoh berikut meng-upgrade snapshot DB.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-snapshot \
    --db-snapshot-identifier mydbsnapshot \
    --engine-version 19.0.0.0.ru-2020-10.rur-2020-10.r1 \
    --option-group-name default:oracle-se2-19
```

Untuk Windows:

```
aws rds modify-db-snapshot ^
    --db-snapshot-identifier mydbsnapshot ^
    --engine-version 19.0.0.0.ru-2020-10.rur-2020-10.r1 ^
    --option-group-name default:oracle-se2-19
```

API RDS

Untuk meng-upgrade snapshot DB Oracle dengan menggunakan API Amazon RDS, panggil operasi ModifyDBSnapshot dengan parameter berikut:

- `DBSnapshotIdentifier` – Nama snapshot DB.
- `EngineVersion` – Versi untuk meng-upgrade snapshot.

Anda mungkin juga perlu memasukkan parameter `OptionGroupName`. Pertimbangan grup opsi yang berlaku saat mengupgrade snapshot DB akan sama seperti saat mengupgrade instans DB. Untuk informasi selengkapnya, lihat Pertimbangan grup opsi.

Menggunakan perangkat lunak pihak ketiga dengan instans DB RDS for Oracle

Untuk menggunakan alat dan perangkat lunak pihak ketiga dengan instans DB RDS for Oracle, tinjau informasi di bagian berikut.

Topik

Menyiapkan Amazon RDS untuk menjadi host alat dan perangkat lunak pihak ketiga untuk Oracle

Anda dapat menggunakan Amazon RDS sebagai host instans DB Oracle yang mendukung perangkat lunak dan komponen seperti berikut:

- Siebel Customer Relationship Management (CRM)
- Oracle Fusion Middleware Metadata — diinstal oleh Repository Creation Utility (RCU)

Prosedur berikut membantu Anda membuat instans DB Oracle di Amazon RDS yang dapat Anda gunakan sebagai host perangkat lunak dan komponen tambahan untuk Oracle.

Topik

Membuat VPC untuk digunakan dengan basis data Oracle

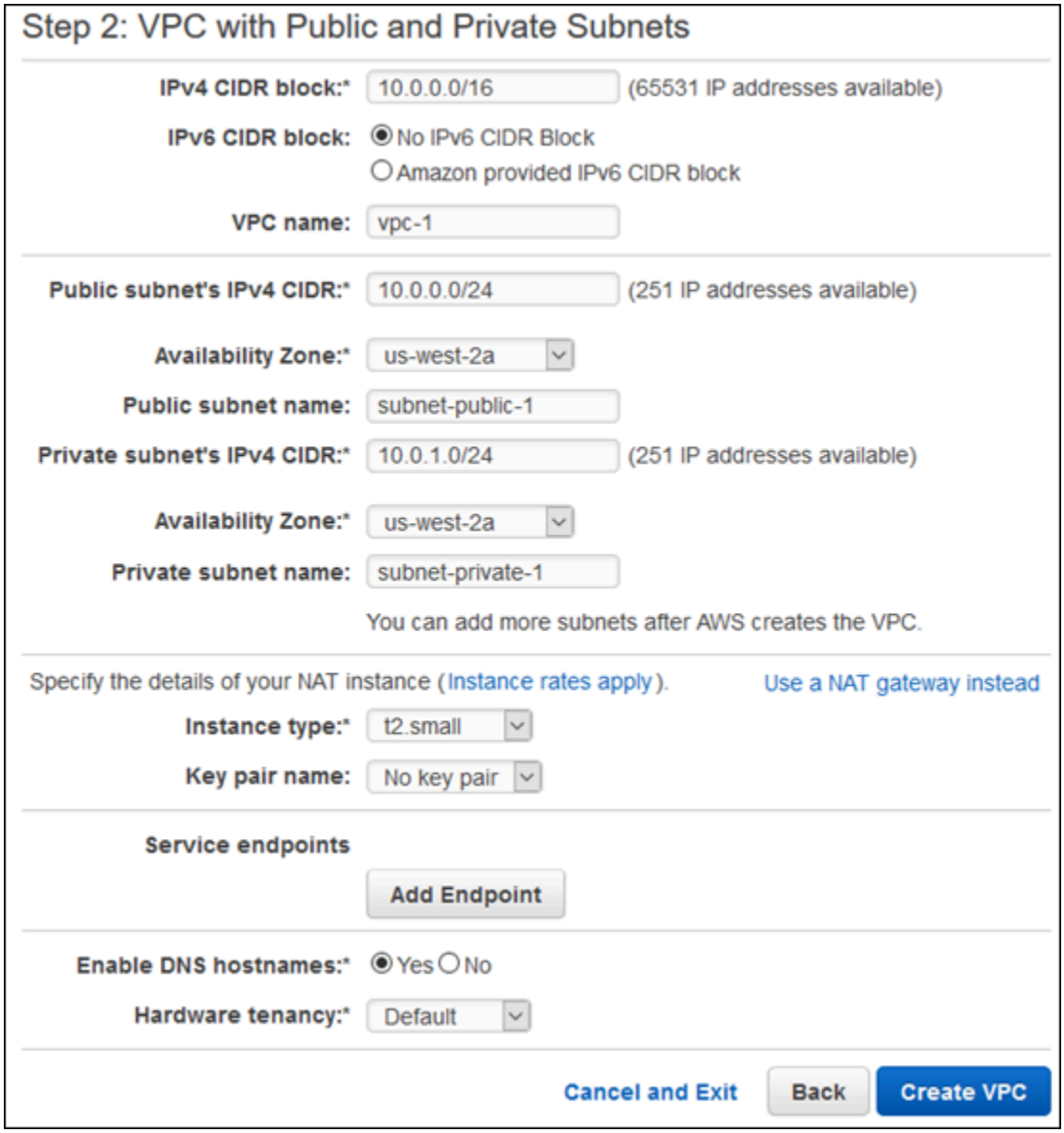
Dalam prosedur berikut, Anda membuat cloud privat virtual (VPC) berdasarkan layanan Amazon VPC, subnet privat, dan grup keamanan. Instans DB Amazon RDS Anda hanya boleh tersedia untuk komponen tingkat menengah Anda, bukan untuk internet publik. Karenanya, host untuk instans DB Amazon RDS Anda berada di subnet privat, yang keamanannya lebih baik.

Cara membuat VPC berdasarkan Amazon VPC

1. Masuk ke AWS Management Console dan buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di sudut kanan atas pada AWS Management Console, pilih Wilayah AWS untuk VPC Anda. Contoh ini menggunakan Wilayah AS Barat (Oregon).
3. Di sudut kiri atas, pilih Dasbor VPC, lalu pilih Mulai Wizard VPC.
4. Pada halaman Langkah 1: Pilih Konfigurasi VPC, pilih VPC dengan Subnet Publik dan Privat, lalu pilih Pilih.
5. Pada halaman Langkah 2: VPC dengan Subnet Publik dan Privat, yang ditampilkan berikut ini, atur nilai-nilai berikut.

Opsi	Nilai
Blok CIDR IPv4	`10.0.0.0/16` Untuk informasi lebih lanjut tentang memilih blok CIDR untuk VPC Anda, lihat Penentuan ukuran VPC.
Blok CIDR IPv6	Tidak ada Blok CIDR No IPv6
Nama VPC	Nama untuk VPC Anda, misalnya **`vpc-1`**.
CIDR IPv4 subnet publik	`10.0.0.0/24` Untuk informasi selengkapnya tentang penentuan ukuran subnet, lihat Penentuan ukuran subnet.
Zona Ketersediaan	Zona Ketersediaan untuk Wilayah AWS Anda.
Nama subnet publik	Nama untuk subnet publik Anda, misalnya **`subnet-public-1`** .
CIDR IPv4 subnet privat	`10.0.1.0/24` Untuk informasi selengkapnya tentang penentuan ukuran subnet, lihat Penentuan ukuran subnet.
Zona Ketersediaan	Zona Ketersediaan untuk Wilayah AWS Anda.
Nama subnet privat	Nama untuk subnet privat Anda, misalnya **`subnet-private-1`** .
Jenis instans	Jenis instans untuk instans NAT Anda, misalnya `t2.small`. **Note** Jika Anda tidak melihat Jenis instans di konsol, pilih Gunakan instans NAT.
Nama key pair	`No key pair`
Titik akhir layanan	`None`

Opsi	Nilai
Aktifkan nama host DNS	`Yes`
Penghunian perangkat keras	`Default`



6. Pilih Buat VPC.

Instans DB Amazon RDS di VPC memerlukan setidaknya dua subnet privat atau setidaknya dua subnet publik, untuk mendukung deployment Multi-AZ. Untuk informasi selengkapnya tentang bekerja dengan beberapa Zona Ketersediaan, lihat Wilayah, Zona Ketersediaan, dan Zona Lokal. Karena basis data Anda privat, tambahkan subnet privat kedua ke VPC Anda.

Cara membuat subnet tambahan

1. Masuk ke AWS Management Console dan buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di sudut kanan atas AWS Management Console, konfirmasikan bahwa Anda berada di Wilayah AWS yang benar untuk VPC Anda.
3. Di sudut kiri atas, pilih Dasbor VPC, pilih Subnet, lalu pilih Buat Subnet.
4. Pada halaman Buat Subnet, atur nilai berikut.

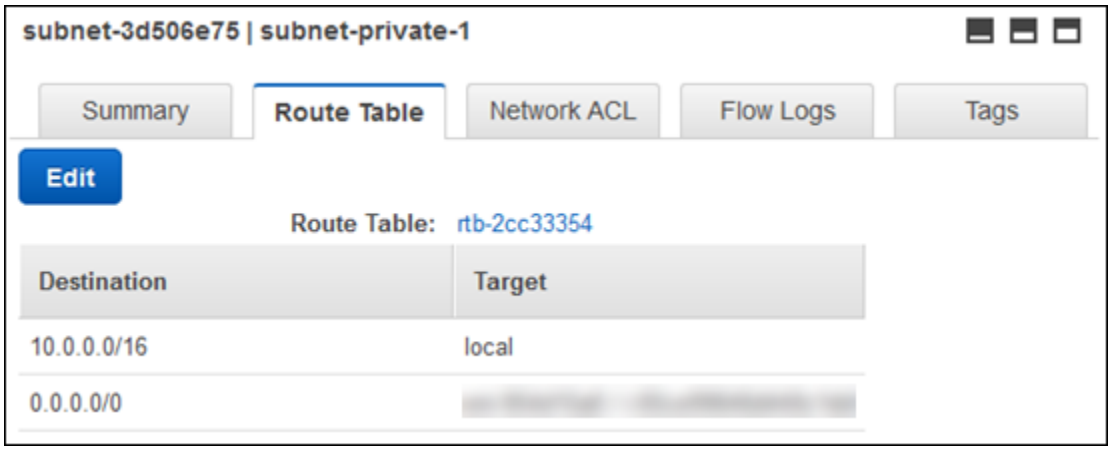
Opsi	Nilai
Tag nama	Nama untuk subnet privat kedua Anda, misalnya **subnet-private-2** .
VPC	VPC Anda, misalnya **vpc-1**.
Zona Ketersediaan	Zona Ketersediaan untuk Wilayah AWS Anda. **Note** Pilih Zona Ketersediaan yang berbeda dari yang Anda pilih untuk subnet privat pertama.
Blok CIDR	`10.0.2.0/24`

5. Pilih Ya, Buat.

Kedua subnet privat harus menggunakan tabel rute yang sama. Dalam prosedur berikut, periksa dan pastikan tabel rute cocok. Jika tidak cocok, edit salah satunya.

Untuk memastikan subnet menggunakan tabel rute yang sama.

1. Masuk ke AWS Management Console dan buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di sudut kanan atas AWS Management Console, konfirmasikan bahwa Anda berada di Wilayah AWS yang benar untuk VPC Anda.
3. Di sudut kiri atas, pilih Dasbor VPC, pilih Subnet, lalu pilih subnet privat pertama Anda, misalnya **`subnet-private-1`**.
4. Di bagian bawah konsol, pilih tab Tabel Rute, seperti ditampilkan berikut ini.



5. Buat catatan tentang tabel rute, misalnya `rtb-0d9fc668`.
6. Di daftar subnet, pilih subnet privat kedua, misalnya **`subnet-private-2`**.
7. Di bagian bawah konsol, pilih tab Tabel Rute.
8. Jika tabel rute untuk subnet kedua tidak sama dengan tabel rute untuk subnet pertama, edit agar cocok:
   a. Pilih Edit.
   b. Pada bagian Ubah ke, pilih tabel rute yang cocok dengan subnet pertama Anda.
   c. Pilih Simpan.

Grup keamanan bertindak sebagai firewall virtual untuk instans DB Anda guna mengendalikan lalu lintas masuk dan keluar. Dalam prosedur berikut ini, Anda membuat grup keamanan untuk instans DB Anda. Untuk informasi selengkapnya tentang grup keamanan, lihat Grup keamanan untuk VPC Anda.

Untuk membuat grup keamanan VPC untuk instans DB Amazon RDS privat

1. Masuk ke AWS Management Console dan buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di sudut kanan atas AWS Management Console, konfirmasikan bahwa Anda berada di Wilayah AWS yang benar untuk VPC Anda.
3. DI sudut kiri atas, pilih Dasbor VPC, pilih Grup Keamanan, lalu pilih Buat Grup Keamanan.
4. Di halaman Buat Grup Keamanan, atur nilai ini.

Opsi	Nilai
Tag nama	Nama untuk grup keamanan Anda, misalnya **sgdb-1**.
Nama grup	Nama untuk grup keamanan Anda, misalnya **sgdb-1**.
Deskripsi	Deskripsi untuk grup keamanan Anda.
VPC	VPC Anda, misalnya **vpc-1**.

5. Pilih Ya, Buat.

Dalam prosedur berikut ini, Anda menambahkan aturan ke grup keamanan Anda untuk mengontrol lalu lintas masuk ke instans DB Anda. Untuk informasi selengkapnya tentang aturan masuk, lihat Aturan grup keamanan.

Untuk menambahkan aturan masuk ke grup keamanan

1. Masuk ke AWS Management Console dan buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di sudut kanan atas AWS Management Console, konfirmasikan bahwa Anda berada di Wilayah AWS yang benar untuk VPC Anda.
3. Di sudut kiri atas, pilih Dasbor VPC, pilih Grup Keamanan, lalu pilih grup keamanan Anda, misalnya **sgdb-1**.
4. Di bagian bawah konsol, pilih tab Aturan Masuk, lalu pilih Edit.
5. Atur nilai-nilai ini, seperti yang ditunjukkan berikut ini.

Opsi	Nilai
Tipe	`Oracle (1521)`
Protokol	`TCP (6)`
Rentang Port	1521
Sumber	Pengenal grup keamanan Anda. Saat Anda memilih kotak tersebut, Anda akan melihat nama grup keamanan Anda, misalnya **`sgdb-1`**.

6. Pilih Simpan.

Membuat instans DB Oracle

Anda dapat menggunakan Amazon RDS untuk menjadi host instans DB Oracle. Saat membuat instans DB baru, tentukan VPC dan grup keamanan yang Anda buat sebelumnya menggunakan petunjuk di Membuat VPC untuk digunakan dengan basis data Oracle. Selain itu, pilih Tidak untuk Dapat diakses publik.

Untuk informasi tentang pembuatan instans DB, lihat Membuat instans DB Amazon RDS.

Antarmuka Amazon RDS tambahan

Dalam tugas sebelumnya, Anda menggunakan AWS Management Console untuk melakukan tugas. Amazon Web Services juga menyediakan AWS Command Line Interface (AWS CLI), dan antarmuka pemrograman aplikasi (API). Anda dapat menggunakan AWS CLI atau API untuk mengotomatiskan banyak tugas untuk mengelola Amazon RDS, termasuk tugas untuk mengelola instans DB Oracle dengan Amazon RDS.

Untuk informasi selengkapnya, lihat Referensi AWS Command Line Interface untuk Amazon RDS dan Referensi Amazon RDS API.

Menggunakan Oracle GoldenGate dengan Amazon RDS for Oracle

Oracle GoldenGate mengumpulkan, mereplikasi, dan mengelola data transaksional antar database. Ini adalah change data capture (CDC) dan paket perangkat lunak replikasi berbasis log yang digunakan dengan basis data untuk sistem online transaction processing (OLTP). Oracle GoldenGate membuat file jejak yang berisi data perubahan terbaru dari database sumber. Perangkat lunak ini mendorong file-file ini ke server, dan melibatkan proses mengubah file jejak menjadi SQL standar ke basis data target.

Oracle GoldenGate dengan RDS untuk Oracle mendukung fitur-fitur berikut:

- Replikasi basis data Active-Active
- Pemulihan bencana
- Perlindungan data
- Replikasi di-wilayah dan lintas-wilayah
- Migrasi dan pembaruan zero-downtime
- Replikasi data antara instans DB RDS for Oracle dan basis data non-Oracle

> **Note**
>
> Untuk daftar basis data yang didukung, lihat Oracle Fusion Middleware Supported System Configurations dalam dokumentasi Oracle.

Anda dapat menggunakan Oracle GoldenGate dengan RDS untuk Oracle untuk meningkatkan ke versi utama Oracle Database. Misalnya, Anda dapat menggunakan Oracle GoldenGate untuk meningkatkan dari database lokal Oracle Database 11g ke Oracle Database 19c pada instans Amazon RDS DB.

Topik

- Pemecahan Masalah Oracle GoldenGate

Versi yang didukung dan opsi lisensi untuk Oracle GoldenGate

Anda dapat menggunakan Standard Edition 2 (SE2) atau Enterprise Edition (EE) dari RDS untuk Oracle dengan Oracle GoldenGate versi 12c dan lebih tinggi. Anda dapat menggunakan GoldenGate fitur Oracle berikut:

- Oracle GoldenGate Remote Capture (ekstrak) didukung.
- Pengambilan (ekstrak) didukung pada RDS untuk instans Oracle DB yang menggunakan arsitektur basis data non-CDB tradisional. Oracle GoldenGate Remote PDB capture didukung pada database kontainer Oracle Database 21c (CDB).
- Oracle GoldenGate Remote Delivery (replika) didukung pada RDS untuk instans Oracle DB yang menggunakan arsitektur non-CDB atau CDB. Pengiriman Jarak Jauh mendukung Replika Terpadu, Replika Paralel, Replika Terkoordinasi, dan Replika klasik.
- RDS untuk Oracle mendukung arsitektur Classic dan Microservices dari Oracle. GoldenGate
- Oracle GoldenGate DDL dan replikasi nilai Urutan didukung saat menggunakan mode tangkapan Terpadu.

Anda bertanggung jawab untuk mengelola GoldenGate lisensi Oracle (BYOL) untuk digunakan dengan Amazon RDS secara keseluruhan. Wilayah AWS Untuk informasi selengkapnya, lihat Opsi lisensi RDS for Oracle.

Persyaratan dan batasan untuk Oracle GoldenGate

Saat Anda bekerja dengan Oracle GoldenGate dan RDS untuk Oracle, pertimbangkan persyaratan dan batasan berikut:

- Anda bertanggung jawab untuk mengatur dan mengelola Oracle GoldenGate untuk digunakan dengan RDS untuk Oracle.
- Anda bertanggung jawab untuk menyiapkan GoldenGate versi Oracle yang disertifikasi dengan sumber dan basis data target. Untuk informasi selengkapnya, lihat Oracle Fusion Middleware Supported System Configurations dalam dokumentasi Oracle.
- Anda dapat menggunakan Oracle GoldenGate di banyak AWS lingkungan yang berbeda untuk banyak kasus penggunaan yang berbeda. Jika Anda memiliki masalah terkait dukungan yang berkaitan dengan Oracle, GoldenGate hubungi Oracle Support Services.

- Anda dapat menggunakan Oracle GoldenGate pada RDS untuk instans Oracle DB yang menggunakan Oracle Transparent Data Encryption (TDE). Untuk menjaga integritas data yang direplikasi, konfigurasikan enkripsi pada GoldenGate hub Oracle menggunakan volume terenkripsi Amazon EBS atau enkripsi file jejak. Juga konfigurasikan enkripsi untuk data yang dikirim antara GoldenGate hub Oracle dan instance database sumber dan target. Instans DB RDS for Oracle mendukung enkripsi dengan Lapisan Soket Aman Oracle atau Enkripsi jaringan asli Oracle.

Arsitektur Oracle GoldenGate

GoldenGate Arsitektur Oracle untuk digunakan dengan Amazon RDS terdiri dari modul terpisah berikut:

Basis data sumber

basis data sumber Anda dapat berupa basis data Oracle on-premise, basis data Oracle di instans Amazon EC2, atau basis data Oracle di instans DB Amazon RDS.

Hub Oracle GoldenGate

GoldenGate Hub Oracle memindahkan informasi transaksi dari database sumber ke database target. Hub Anda dapat berupa salah satu dari berikut ini:

- Instans Amazon EC2 dengan Oracle Database dan Oracle diinstal GoldenGate
- Penginstalan Oracle on-premise

Anda dapat memiliki lebih dari satu hub Amazon EC2. Kami menyarankan Anda menggunakan dua hub jika Anda menggunakan Oracle GoldenGate untuk replikasi lintas wilayah.

Basis data target

Basis data target Anda dapat berada di instans DB Amazon RDS, instans Amazon EC2, atau lokasi on-premise.

Bagian berikut menjelaskan skenario umum untuk Oracle GoldenGate di Amazon RDS.

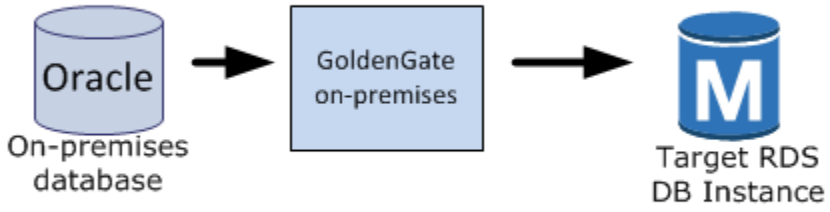
Topik

- Database sumber lokal dan hub Oracle GoldenGate
- Basis data sumber on-premise dan hub Amazon EC2
- Basis data sumber Amazon RDS dan hub Amazon EC2
- Basis data sumber Amazon EC2 dan hub Amazon EC2

- Hub Amazon EC2 di berbagai Wilayah AWS

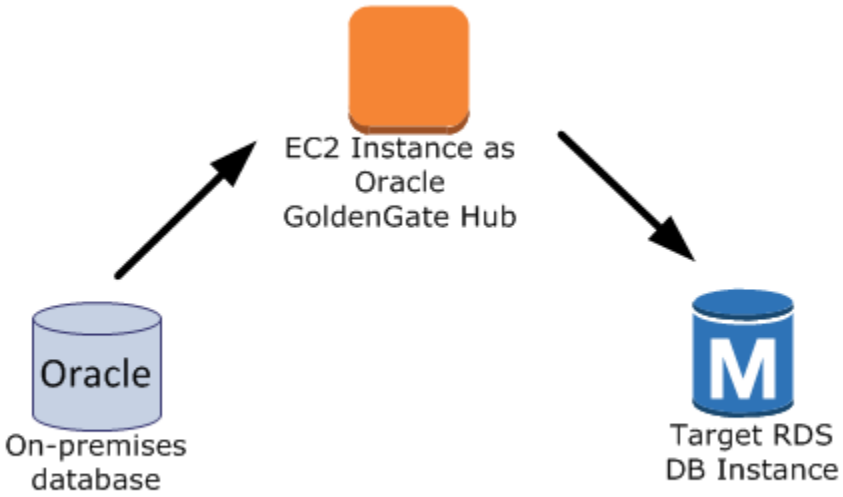
Database sumber lokal dan hub Oracle GoldenGate

Dalam skenario ini, database sumber Oracle lokal dan GoldenGate hub Oracle lokal menyediakan data ke instans Amazon RDS DB target.



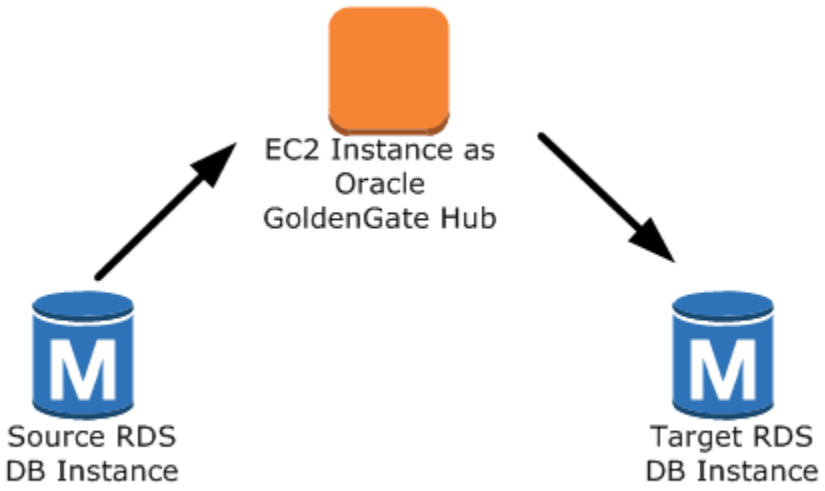
Basis data sumber on-premise dan hub Amazon EC2

Dalam skenario ini, basis data Oracle on-premise bertindak sebagai basis data sumber. Ini terhubung ke hub instans Amazon EC2. Hub ini menyediakan data ke instans DB RDS for Oracle target.



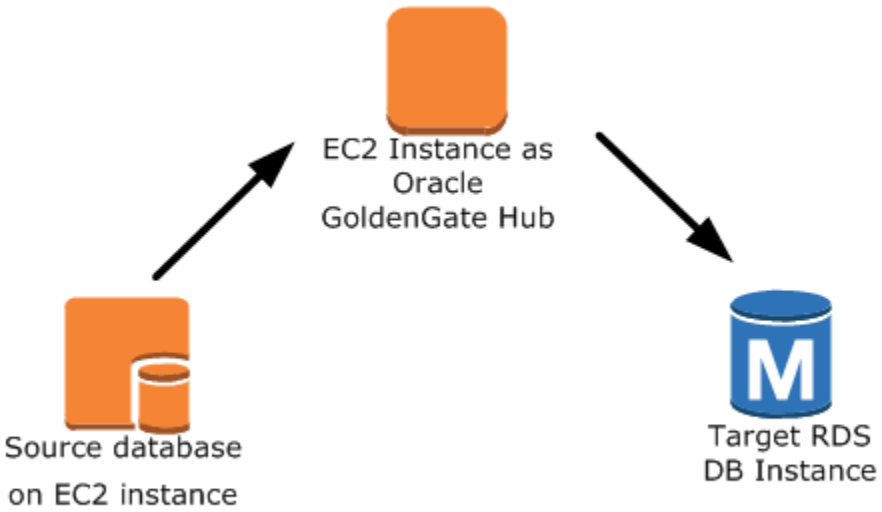
Basis data sumber Amazon RDS dan hub Amazon EC2

Dalam skenario ini, RDS untuk instans Oracle DB bertindak sebagai basis data sumber. Ini terhubung ke hub instans Amazon EC2. Hub ini menyediakan data ke instans DB RDS for Oracle target.



Basis data sumber Amazon EC2 dan hub Amazon EC2

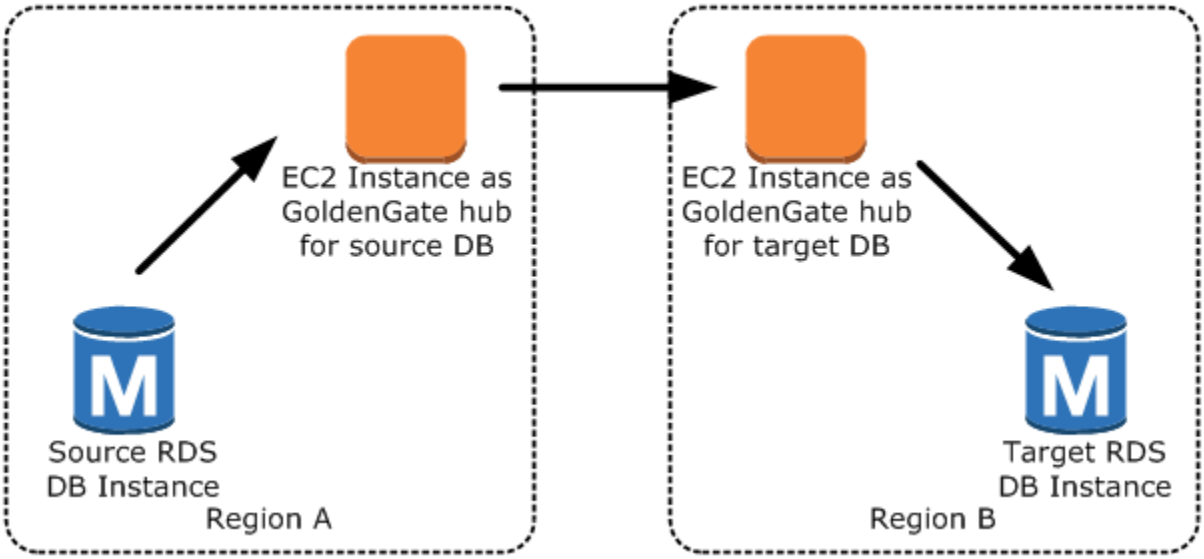
Dalam skenario ini, basis data Oracle di instans Amazon EC2 bertindak sebagai basis data sumber. Ini terhubung ke hub instans Amazon EC2. Hub ini menyediakan data ke instans DB RDS for Oracle target.



Hub Amazon EC2 di berbagai Wilayah AWS

Dalam skenario ini, basis data Oracle di instans DB Amazon RDS terhubung ke hub instans Amazon EC2 di Wilayah AWS yang sama. Hub terhubung ke hub instans Amazon EC2 di Wilayah AWS yang

berbeda. Hub kedua ini memberikan data ke instans DB RDS for Oracle target di Wilayah AWS yang sama dengan hub instans Amazon EC2 kedua.



> **Note**
>
> Masalah apa pun yang memengaruhi menjalankan Oracle GoldenGate di lingkungan lokal juga memengaruhi menjalankan GoldenGate Oracle. AWS Kami sangat menyarankan Anda memantau GoldenGate hub Oracle untuk memastikannya `EXTRACT` dan `REPLICAT` dilanjutkan jika terjadi failover. Karena GoldenGate hub Oracle dijalankan pada instans Amazon EC2, Amazon RDS tidak mengelola hub GoldenGate Oracle dan tidak dapat memastikan bahwa itu berjalan.

Menyiapkan Oracle GoldenGate

Untuk mengatur Oracle GoldenGate menggunakan Amazon RDS, konfigurasikan hub pada instans Amazon EC2, lalu konfigurasikan basis data sumber dan target. Bagian berikut memberikan contoh cara mengatur Oracle untuk digunakan dengan Amazon RDS GoldenGate for Oracle.

Topik

- Menyiapkan GoldenGate hub Oracle di Amazon EC2
- Menyiapkan database sumber untuk digunakan dengan Oracle GoldenGate di Amazon RDS
- Menyiapkan database target untuk digunakan dengan Oracle GoldenGate di Amazon RDS

Menyiapkan GoldenGate hub Oracle di Amazon EC2

Untuk membuat GoldenGate hub Oracle di instans Amazon EC2, pertama-tama Anda membuat instans Amazon EC2 dengan instalasi klien penuh Oracle RDBMS. Instans Amazon EC2 juga harus memiliki perangkat lunak Oracle GoldenGate yang diinstal. Versi GoldenGate perangkat lunak Oracle bergantung pada versi basis data sumber dan target. Untuk informasi selengkapnya tentang menginstal Oracle GoldenGate, lihat dokumentasi Oracle GoldenGate.

Instans Amazon EC2 yang berfungsi sebagai GoldenGate hub Oracle menyimpan dan memproses informasi transaksi dari database sumber ke dalam file jejak. Untuk mendukung proses ini, pastikan Anda memenuhi persyaratan berikut:

- Anda telah mengalokasikan penyimpanan yang cukup untuk file jejak.
- Instans Amazon EC2 memiliki kekuatan pemrosesan yang cukup untuk mengelola jumlah data.
- Instans EC2 juga memiliki cukup memori untuk menyimpan informasi transaksi sebelum ditulis ke file jejak.

Untuk menyiapkan hub arsitektur GoldenGate klasik Oracle pada instans Amazon EC2

1. Buat subdirektori di direktori GoldenGate Oracle.

   Di shell baris perintah Amazon EC2, mulai`ggsci`, penerjemah perintah Oracle GoldenGate . Perintah `CREATE SUBDIRS` membuat subdirektori di bawah direktori `/gg` untuk file parameter, laporan, dan titik pemeriksaan.

   ```
   prompt$ cd /gg
   prompt$ ./ggsci

   GGSCI> CREATE SUBDIRS
   ```

2. Konfigurasikan file `mgr.prm`.

   Contoh berikut menambahkan baris ke file `$GGHOME/dirprm/mgr.prm`.

   ```
   PORT 8199
   PurgeOldExtracts ./dirdat/*, UseCheckpoints, MINKEEPDAYS 5
   ```

3. Mulai pengelola.

   Contoh berikut memulai perintah `ggsci` dan menjalankan perintah `start mgr`.

```
GGSCI> start mgr
```

GoldenGate Hub Oracle sekarang siap digunakan.

Menyiapkan database sumber untuk digunakan dengan Oracle GoldenGate di Amazon RDS

Ketika database sumber Anda menjalankan Oracle Database 12c atau yang lebih baru, selesaikan tugas-tugas berikut untuk menyiapkan database sumber untuk digunakan dengan Oracle. GoldenGate

Langkah-langkah penyiapan

- Langkah 1: Mengaktifkan pencatatan log tambahan pada basis data sumber
- Langkah 2: Menetapkan parameter inisialisasi ENABLE_GOLDENGATE_REPLICATION ke benar
- Langkah 3: Menetapkan periode retensi log pada basis data sumber
- Langkah 4: Buat akun GoldenGate pengguna Oracle di database sumber
- Langkah 5: Memberi hak akses akun pengguna pada basis data sumber
- Langkah 6: Menambahkan alias TNS untuk basis data sumber

Langkah 1: Mengaktifkan pencatatan log tambahan pada basis data sumber

Untuk mengaktifkan pencatatan log tambahan tingkat basis data minimum, jalankan prosedur PL/SQL berikut:

```
EXEC rdsadmin.rdsadmin_util.alter_supplemental_logging(p_action => 'ADD')
```

Langkah 2: Menetapkan parameter inisialisasi ENABLE_GOLDENGATE_REPLICATION ke benar

Ketika Anda mengatur parameter inisialisasi `ENABLE_GOLDENGATE_REPLICATION` ke `true`, pengaturan ini memungkinkan layanan basis data untuk mendukung replikasi logis. Jika basis data sumber Anda berada di instans DB Amazon RDS, pastikan Anda memiliki grup parameter yang ditetapkan ke instans DB dengan parameter inisialisasi `ENABLE_GOLDENGATE_REPLICATION` yang diatur ke `true`. Untuk informasi parameter inisialisasi `ENABLE_GOLDENGATE_REPLICATION` selengkapnya, lihat dokumentasi Oracle Database.

Langkah 3: Menetapkan periode retensi log pada basis data sumber

Pastikan Anda mengonfigurasi basis data sumber untuk mempertahankan log pengulangan yang diarsipkan. Pertimbangkan panduan-panduan berikut ini:

- Tentukan durasi retensi log dalam jam. Nilai minimumnya adalah satu jam.
- Tetapkan durasi untuk melebihi potensi waktu henti instans DB sumber, potensi periode komunikasi, dan periode potensi masalah jaringan apa pun untuk instans sumber. Durasi seperti itu memungkinkan Oracle GoldenGate memulihkan log dari instans sumber sesuai kebutuhan.
- Pastikan Anda memiliki penyimpanan yang cukup di instans Anda untuk file.

Misalnya, setel periode retensi untuk log pengulangan yang diarsipkan menjadi 24 jam.

```
EXEC rdsadmin.rdsadmin_util.set_configuration('archivelog retention hours',24)
```

Jika Anda belum mengaktifkan retensi log, atau jika nilai retensi terlalu kecil, Anda akan menerima pesan kesalahan yang mirip dengan berikut ini.

```
2022-03-06 06:17:27  ERROR   OGG-00446  error 2 (No such file or directory)
opening redo log /rdsdbdata/db/GGTEST3_A/onlinelog/o1_mf_2_9k4bp1n6_.log for sequence
 1306
Not able to establish initial position for begin time 2022-03-06 06:16:55.
```

Karena instans DB Anda menyimpan log pengulangan yang diarsipkan, pastikan Anda memiliki ruang yang cukup untuk file tersebut. Untuk melihat seberapa besar ruang yang telah Anda gunakan dalam jam *num_hours* terakhir, jalankan kueri berikut, lalu ganti *num_hours* dengan jumlah jam.

```
SELECT SUM(BLOCKS * BLOCK_SIZE) BYTES FROM V$ARCHIVED_LOG
   WHERE NEXT_TIME>=SYSDATE-num_hours/24 AND DEST_ID=1;
```

Langkah 4: Buat akun GoldenGate pengguna Oracle di database sumber

Oracle GoldenGate berjalan sebagai pengguna database dan memerlukan hak database yang sesuai untuk mengakses redo dan arsip redo log untuk database sumber. Untuk menyediakan kebutuhan ini, buat akun pengguna di basis data sumber. Untuk informasi selengkapnya tentang izin untuk akun GoldenGate pengguna Oracle, lihat dokumentasi Oracle.

Pernyataan berikut membuat akun pengguna dengan nama `oggadm1`.

```
CREATE TABLESPACE administrator;
CREATE USER oggadm1  IDENTIFIED BY "password"
   DEFAULT TABLESPACE ADMINISTRATOR TEMPORARY TABLESPACE TEMP;
ALTER USER oggadm1 QUOTA UNLIMITED ON administrator;
```

> **Note**
>
> Tetapkan kata sandi selain prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

Langkah 5: Memberi hak akses akun pengguna pada basis data sumber

Dalam tugas ini, Anda memberikan hak istimewa akun yang diperlukan untuk pengguna basis data di basis data sumber Anda.

Memberikan hak akses akun pada basis data sumber

1. Berikan hak istimewa yang diperlukan ke akun GoldenGate pengguna Oracle menggunakan perintah SQL `grant` dan prosedurnya. `rdsadmin.rdsadmin_util grant_sys_object` Pernyataan berikut memberikan hak akses untuk pengguna dengan nama `oggadm1`.

   ```
   GRANT CREATE SESSION, ALTER SESSION TO oggadm1;
   GRANT RESOURCE TO oggadm1;
   GRANT SELECT ANY DICTIONARY TO oggadm1;
   GRANT FLASHBACK ANY TABLE TO oggadm1;
   GRANT SELECT ANY TABLE TO oggadm1;
   GRANT SELECT_CATALOG_ROLE TO rds_master_user_name WITH ADMIN OPTION;
   EXEC rdsadmin.rdsadmin_util.grant_sys_object ('DBA_CLUSTERS', 'OGGADM1');
   GRANT EXECUTE ON DBMS_FLASHBACK TO oggadm1;
   GRANT SELECT ON SYS.V_$DATABASE TO oggadm1;
   GRANT ALTER ANY TABLE TO oggadm1;
   ```

2. Berikan hak istimewa yang dibutuhkan oleh akun pengguna untuk menjadi administrator Oracle GoldenGate. Paket yang Anda gunakan untuk melakukan pemberian hak akses, `dbms_goldengate_auth` atau `rdsadmin_dbms_goldengate_auth`, bergantung pada versi mesin Oracle DB.

   - Untuk versi Oracle DB yang lebih baru dari atau sama dengan Oracle Database 12c Rilis 2 (12.2), yang memerlukan patch level 12.2.0.1.ru-2019-04.rur-2019-04.r1 atau yang lebih baru, jalankan program PL/SQL berikut.

```
EXEC rdsadmin.rdsadmin_dbms_goldengate_auth.grant_admin_privilege (
    grantee                 => 'OGGADM1',
    privilege_type          => 'capture',
    grant_select_privileges => true,
    do_grants               => TRUE);
```

- Untuk versi basis data Oracle yang lebih lama dari Oracle Database 12c Rilis 2 (12.2), jalankan program PL/SQL berikut.

```
EXEC dbms_goldengate_auth.grant_admin_privilege (
    grantee                 => 'OGGADM1',
    privilege_type          => 'capture',
    grant_select_privileges => true,
    do_grants               => TRUE);
```

Untuk mencabut hak akses, gunakan prosedur `revoke_admin_privilege` dalam paket yang sama.

Langkah 6: Menambahkan alias TNS untuk basis data sumber

Tambahkan entri berikut ke `$ORACLE_HOME/network/admin/tnsnames.ora` di beranda Oracle untuk digunakan oleh proses `EXTRACT`. Untuk informasi selengkapnya tentang file `tnsnames.ora`, lihat Oracle documentation.

```
OGGSOURCE=
   (DESCRIPTION=
        (ENABLE=BROKEN)
        (ADDRESS_LIST=
            (ADDRESS=(PROTOCOL=TCP)(HOST=goldengate-source.abcdef12345.us-
west-2.rds.amazonaws.com)(PORT=8200)))
        (CONNECT_DATA=(SERVICE_NAME=ORCL))
    )
```

Menyiapkan database target untuk digunakan dengan Oracle GoldenGate di Amazon RDS

Dalam tugas ini, Anda menyiapkan instans DB target untuk digunakan dengan Oracle GoldenGate.

Langkah-langkah penyiapan

- Langkah 1: Menetapkan parameter inisialisasi ENABLE_GOLDENGATE_REPLICATION ke benar

Langkah 1: Menetapkan parameter inisialisasi ENABLE_GOLDENGATE_REPLICATION ke benar

Ketika Anda mengatur parameter inisialisasi `ENABLE_GOLDENGATE_REPLICATION` untuk `true`, ini memungkinkan layanan basis data untuk mendukung replikasi logis. Jika basis data sumber Anda berada di instans DB Amazon RDS, pastikan Anda memiliki grup parameter yang ditetapkan ke instans DB dengan parameter inisialisasi `ENABLE_GOLDENGATE_REPLICATION` yang diatur ke `true`. Untuk informasi parameter inisialisasi `ENABLE_GOLDENGATE_REPLICATION` selengkapnya, lihat dokumentasi Oracle Database.

Langkah 2: Buat akun GoldenGate pengguna Oracle pada database target

Oracle GoldenGate berjalan sebagai pengguna database dan membutuhkan hak istimewa database yang sesuai. Untuk memastikannya, buat akun pengguna di basis data target.

Pernyataan berikut akan membuat pengguna dengan nama `oggadm1`.

```
CREATE TABLESPSACE administrator;
CREATE USER oggadm1  IDENTIFIED BY "password"
   DEFAULT TABLESPACE administrator
   TEMPORARY TABLESPACE temp;
ALTER USER oggadm1 QUOTA UNLIMITED ON administrator;
```

> **Note**
>
> Tetapkan kata sandi selain prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

Langkah 3: Memberi hak akses akun pada basis data target

Dalam tugas ini, Anda memberikan hak istimewa akun yang diperlukan untuk pengguna basis data di basis data target Anda.

Untuk memberikan hak akses akun pada basis data target

1. Berikan hak istimewa yang diperlukan ke akun GoldenGate pengguna Oracle pada basis data target. Pada contoh berikut, Anda memberikan hak akses untuk `oggadm1`.

```
GRANT CREATE SESSION        TO oggadm1;
GRANT ALTER SESSION         TO oggadm1;
GRANT CREATE CLUSTER        TO oggadm1;
GRANT CREATE INDEXTYPE      TO oggadm1;
GRANT CREATE OPERATOR       TO oggadm1;
GRANT CREATE PROCEDURE      TO oggadm1;
GRANT CREATE SEQUENCE       TO oggadm1;
GRANT CREATE TABLE          TO oggadm1;
GRANT CREATE TRIGGER        TO oggadm1;
GRANT CREATE TYPE           TO oggadm1;
GRANT SELECT ANY DICTIONARY TO oggadm1;
GRANT CREATE ANY TABLE      TO oggadm1;
GRANT ALTER ANY TABLE       TO oggadm1;
GRANT LOCK ANY TABLE        TO oggadm1;
GRANT SELECT ANY TABLE      TO oggadm1;
GRANT INSERT ANY TABLE      TO oggadm1;
GRANT UPDATE ANY TABLE      TO oggadm1;
GRANT DELETE ANY TABLE      TO oggadm1;
```

2. Berikan hak istimewa yang dibutuhkan oleh akun pengguna untuk menjadi administrator Oracle GoldenGate. Paket yang Anda gunakan untuk melakukan pemberian hak akses, `dbms_goldengate_auth` atau `rdsadmin_dbms_goldengate_auth`, bergantung pada versi mesin Oracle DB.

   - Untuk versi basis data Oracle yang lebih baru dari atau sama dengan Oracle Database 12c Rilis 2 (12.2), yang memerlukan patch level 12.2.0.1.ru-2019-04.rur-2019-04.r1 atau yang lebih baru, jalankan program PL/SQL berikut.

     ```
     EXEC rdsadmin.rdsadmin_dbms_goldengate_auth.grant_admin_privilege (
         grantee                 => 'OGGADM1',
         privilege_type          => 'apply',
         grant_select_privileges => true,
         do_grants               => TRUE);
     ```

   - Untuk versi basis data Oracle yang lebih rendah dari Oracle Database 12c Rilis 2 (12.2), jalankan program PL/SQL berikut.

     ```
     EXEC dbms_goldengate_auth.grant_admin_privilege (
         grantee                 => 'OGGADM1',
         privilege_type          => 'apply',
         grant_select_privileges => true,
     ```

```
    do_grants               => TRUE);
```

Untuk mencabut hak akses, gunakan prosedur `revoke_admin_privilege` dalam paket yang sama.

Langkah 4: Menambahkan alias TNS untuk basis data target

Tambahkan entri berikut ke `$ORACLE_HOME/network/admin/tnsnames.ora` di beranda Oracle untuk digunakan oleh proses `REPLICAT`. Untuk basis data Oracle Multitenant, pastikan alias TNS menunjuk ke nama layanan PDB. Untuk informasi selengkapnya tentang file `tnsnames.ora`, lihat Oracle documentation.

```
OGGTARGET=
    (DESCRIPTION=
        (ENABLE=BROKEN)
        (ADDRESS_LIST=
            (ADDRESS=(PROTOCOL=TCP)(HOST=goldengate-target.abcdef12345.us-
west-2.rds.amazonaws.com)(PORT=8200)))
        (CONNECT_DATA=(SERVICE_NAME=ORCL))
    )
```

Bekerja dengan utilitas EXTRACT dan REPLICAT dari Oracle GoldenGate

GoldenGate Utilitas Oracle `EXTRACT` dan `REPLICAT` bekerja sama untuk menjaga database sumber dan target tetap sinkron melalui replikasi transaksi inkremental menggunakan file jejak. Semua perubahan yang terjadi pada database sumber secara otomatis dideteksi oleh`EXTRACT`, kemudian diformat dan ditransfer ke file jejak di hub instans Oracle GoldenGate lokal atau Amazon EC2. Setelah pemuatan awal selesai, data dibaca dari file ini dan direplikasi ke basis data target oleh utilitas `REPLICAT`.

Menjalankan utilitas Oracle GoldenGate EXTRACT

Utilitas `EXTRACT` mengambil, mengubah, dan mengeluarkan data dari basis data sumber ke file jejak. Prosesnya adalah sebagai berikut:

1. `EXTRACT` memasukkan detail transaksi ke memori atau ke penyimpanan disk sementara.
2. Basis data sumber melakukan transaksi.
3. `EXTRACT` menulis detail transaksi ke file jejak.

4. File trail merutekan detail ini ke Oracle GoldenGate lokal atau hub instans Amazon EC2 dan kemudian ke database target.

Langkah-langkah berikut memulai utilitas `EXTRACT`, menangkap data dari `EXAMPLE.TABLE` pada basis data sumber `OGGSOURCE`, dan membuat file jejak.

Untuk menjalankan utilitas EXTRACT

1. Konfigurasikan file `EXTRACT` parameter di GoldenGate hub Oracle (instans Amazon EC2 lokal atau Amazon). Daftar berikut menunjukkan contoh file parameter `EXTRACT` bernama `$GGHOME/dirprm/eabc.prm`.

```
EXTRACT EABC

USERID oggadm1@OGGSOURCE, PASSWORD "my-password"
EXTTRAIL /path/to/goldengate/dirdat/ab

IGNOREREPLICATES
GETAPPLOPS
TRANLOGOPTIONS EXCLUDEUSER OGGADM1

TABLE EXAMPLE.TABLE;
```

2. Di GoldenGate hub Oracle, masuk ke database sumber dan luncurkan antarmuka baris GoldenGate perintah Oracle. `ggsci` Contoh berikut menunjukkan format untuk pencatatan log masuk.

```
dblogin oggadm1@OGGSOURCE
```

3. Tambahkan data transaksi untuk mengaktifkan pencatatan log tambahan untuk tabel basis data.

```
add trandata EXAMPLE.TABLE
```

4. Dengan menggunakan baris perintah `ggsci`, aktifkan utilitas `EXTRACT` menggunakan perintah berikut.

```
add extract EABC tranlog, INTEGRATED tranlog, begin now
add exttrail /path/to/goldengate/dirdat/ab
   extract EABC,
   MEGABYTES 100
```

5. Daftarkan utilitas EXTRACT dengan basis data sehingga log arsip tidak terhapus. Tugas ini memungkinkan Anda memulihkan transaksi lama yang belum terikat jika perlu. Untuk mendaftarkan utilitas EXTRACT dengan basis data, gunakan perintah berikut.

```
register EXTRACT EABC, DATABASE
```

6. Mulai utilitas EXTRACT dengan perintah berikut.

```
start EABC
```

Menjalankan utilitas Oracle GoldenGate REPLIKAT

Utilitas REPLICAT "mendorong" informasi transaksi dalam file jejak ke basis data target.

Langkah-langkah berikut mengaktifkan dan memulai utilitas REPLICAT sehingga dapat mereplikasi data yang diambil ke tabel EXAMPLE.TABLE dalam basis data target OGGTARGET.

Menjalankan utilitas REPLICATE

1. Konfigurasikan file REPLICAT parameter pada GoldenGate hub Oracle (instans lokal atau EC2). Daftar berikut menunjukkan contoh file parameter REPLICAT bernama $GGHOME/dirprm/rabc.prm.

```
REPLICAT RABC

USERID oggadm1@OGGTARGET, password "my-password"

ASSUMETARGETDEFS
MAP EXAMPLE.TABLE, TARGET EXAMPLE.TABLE;
```

> **Note**
>
> Tetapkan kata sandi selain penggugah/prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

2. Masuk ke database target dan luncurkan antarmuka baris GoldenGate perintah Oracle (ggsci). Contoh berikut menunjukkan format untuk pencatatan log masuk.

```
dblogin userid oggadm1@OGGTARGET
```

3. Dengan menggunakan baris perintah `ggsci`, tambahkan tabel titik pemeriksaan. Pengguna yang ditunjukkan harus menjadi akun GoldenGate pengguna Oracle, bukan pemilik skema tabel target. Contoh berikut membuat tabel titik pemeriksaan bernama `gg_checkpoint`.

```
add checkpointtable oggadm1.oggchkpt
```

4. Untuk mengaktifkan utilitas `REPLICAT`, gunakan perintah berikut.

```
add replicat RABC EXTTRAIL /path/to/goldengate/dirdat/ab CHECKPOINTTABLE
 oggadm1.oggchkpt
```

5. Mulai utilitas `REPLICAT` dengan menggunakan perintah berikut.

```
start RABC
```

Memantau Oracle GoldenGate

Ketika Anda menggunakan Oracle GoldenGate untuk replikasi, pastikan bahwa GoldenGate proses Oracle aktif dan berjalan dan sumber dan database target disinkronkan. Anda dapat menggunakan alat pemantauan berikut:

- Amazon CloudWatch adalah layanan pemantauan yang digunakan dalam pola ini untuk memantau log GoldenGate kesalahan.
- Amazon SNS adalah layanan notifikasi pesan yang digunakan dalam pola ini untuk mengirim notifikasi email.

Untuk petunjuk terperinci, lihat Memantau GoldenGate log Oracle menggunakan Amazon CloudWatch.

Pemecahan Masalah Oracle GoldenGate

Bagian ini menjelaskan masalah paling umum saat menggunakan Oracle GoldenGate dengan Amazon RDS for Oracle.

Topik

- Kesalahan saat membuka log pengulangan online
- Oracle GoldenGate tampaknya dikonfigurasi dengan benar tetapi replikasi tidak berfungsi
- REPLICAT terintegrasi lambat karena kueri pada SYS."_DBA_APPLY_CDR_INFO"

Kesalahan saat membuka log pengulangan online

Pastikan Anda mengonfigurasi basis data Anda untuk mempertahankan log pengulangan yang diarsipkan. Pertimbangkan panduan-panduan berikut ini:

- Tentukan durasi retensi log dalam jam. Nilai minimumnya adalah satu jam.
- Tetapkan durasi untuk melebihi potensi waktu henti instans DB sumber, potensi periode komunikasi, dan periode potensi masalah jaringan apa pun untuk instans DB sumber. Durasi seperti itu memungkinkan Oracle GoldenGate memulihkan log dari instance DB sumber sesuai kebutuhan.
- Pastikan Anda memiliki penyimpanan yang cukup di instans Anda untuk file.

Jika Anda belum mengaktifkan retensi log, atau jika nilai retensi terlalu kecil, Anda akan menerima pesan kesalahan yang mirip dengan berikut ini.

```
2022-03-06 06:17:27  ERROR   OGG-00446  error 2 (No such file or directory)
opening redo log /rdsdbdata/db/GGTEST3_A/onlinelog/o1_mf_2_9k4bp1n6_.log for sequence
 1306
Not able to establish initial position for begin time 2022-03-06 06:16:55.
```

Oracle GoldenGate tampaknya dikonfigurasi dengan benar tetapi replikasi tidak berfungsi

Untuk tabel yang sudah ada sebelumnya, Anda harus menentukan SCN tempat Oracle bekerja GoldenGate .

Untuk memperbaiki masalah ini

1. Masuk ke database sumber dan luncurkan antarmuka baris GoldenGate perintah Oracle (`ggsci`). Contoh berikut menunjukkan format untuk pencatatan log masuk.

   ```
   dblogin userid oggadm1@OGGSOURCE
   ```

2. Dengan menggunakan baris perintah `ggsci`, siapkan SCN awal untuk proses `EXTRACT`. Contoh berikut mengatur SCN ke 223274 untuk `EXTRACT`.

   ```
   ALTER EXTRACT EABC SCN 223274
   start EABC
   ```

3. Masuk ke basis data target. Contoh berikut menunjukkan format untuk pencatatan log masuk.

```
dblogin userid oggadm1@OGGTARGET
```

4. Dengan menggunakan baris perintah `ggsci`, siapkan SCN awal untuk proses `REPLICAT`. Contoh berikut mengatur SCN ke 223274 untuk `REPLICAT`.

```
start RABC atcsn 223274
```

REPLICAT terintegrasi lambat karena kueri pada SYS."_DBA_APPLY_CDR_INFO"

Oracle GoldenGate Conflict Detection and Resolution (CDR) menyediakan rutinitas resolusi konflik dasar. Misalnya, CDR dapat menyelesaikan konflik unik untuk pernyataan `INSERT`.

Saat CDR menyelesaikan bentrokan, CDR dapat memasukkan rekaman ke dalam tabel pengecualian `_DBA_APPLY_CDR_INFO` untuk sementara. `REPLICAT` yang terintegrasi menghapus catatan ini nanti. Dalam skenario yang jarang terjadi, `REPLICAT` yang terintegrasi dapat memproses banyak bentrokan, tetapi `REPLICAT` baru yang terintegrasi tidak dapat menggantikannya. Alih-alih dihapus, baris yang ada di `_DBA_APPLY_CDR_INFO` menjadi terabaikan. Semua proses `REPLICAT` baru yang terintegrasi melambat karena mengkueri baris yang terabaikan di `_DBA_APPLY_CDR_INFO`.

Untuk menghapus semua baris dari `_DBA_APPLY_CDR_INFO`, gunakan prosedur Amazon RDS `rdsadmin.rdsadmin_util.truncate_apply$_cdr_info`. Prosedur ini dirilis sebagai bagian dari rilis Oktober 2020 dan pembaruan patch. Prosedur ini tersedia dalam versi basis data berikut:

- Versi 21.0.0.0.ru-2022-01.rur-2022-01.r1 dan yang lebih baru
- Versi 19.0.0.0.ru-2020-10.rur-2020-10.r1 dan yang lebih baru

Contoh berikut memotong tabel `_DBA_APPLY_CDR_INFO`.

```
SET SERVEROUTPUT ON SIZE 2000
EXEC rdsadmin.rdsadmin_util.truncate_apply$_cdr_info;
```

Menggunakan Oracle Repository Creation Utility pada RDS for Oracle

Anda dapat menggunakan Amazon RDS untuk melakukan host instans DB RDS for Oracle yang memiliki skema untuk mendukung komponen Oracle Fusion Middleware Anda. Sebelum Anda dapat menggunakan komponen Fusion Middleware, buat dan isi skema untuk komponen tersebut dalam basis data Anda. Anda membuat dan mengisi skema tersebut menggunakan Oracle Repository Creation Utility (RCU).

Opsi lisensi dan versi yang didukung untuk RCU

Amazon RDS hanya mendukung Oracle Repository Creation Utility (RCU) versi 12c. Anda dapat menggunakan RCU tersebut dalam konfigurasi berikut:

- RCU 12c dengan Oracle Database 21c
- RCU 12c dengan Oracle Database 19c
- RCU 12c dengan Oracle Database 12c Rilis 2 (12.2)
- RCU 12c dengan Oracle Database 12c Rilis 1 (12.1) menggunakan 12.1.0.2.v4 atau yang lebih tinggi

Sebelum Anda dapat menggunakan RCU, lakukan hal berikut:

- Dapatkan lisensi untuk Oracle Fusion Middleware.
- Ikuti pedoman lisensi Oracle untuk basis data Oracle yang menjadi host repositori. Untuk informasi selengkapnya, lihat Oracle Fusion Middleware Licensing Information User Manual dalam dokumentasi Oracle.

Fusion MiddleWare mendukung repositori pada Oracle Database Enterprise Edition dan Standard Edition 2. Oracle merekomendasikan Enterprise Edition untuk penginstalan produksi yang memerlukan pembuatan partisi dan penginstalan yang perlu membangun ulang indeks online.

Sebelum Anda membuat instans RDS for Oracle, konfirmasikan versi basis data Oracle yang Anda butuhkan untuk mendukung komponen yang ingin Anda deploy. Gunakan Certification Matrix untuk menemukan persyaratan bagi komponen dan versi Fusion Middleware yang ingin Anda deploy. Untuk informasi selengkapnya, lihat Oracle Fusion Middleware Supported System Configurations dalam dokumentasi Oracle.

Amazon RDS mendukung peningkatan versi basis data Oracle sesuai kebutuhan. Untuk informasi selengkapnya, lihat Meng-upgrade versi mesin instans DB.

Persyaratan dan batasan RCU

Untuk menggunakan RCU, Anda memerlukan Amazon VPC. Instans DB Amazon RDS Anda hanya boleh tersedia untuk komponen Fusion Middleware Anda, bukan untuk Internet publik. Karenanya, host instans DB Amazon RDS Anda di subnet privat, yang memberikan keamanan yang lebih baik. Untuk informasi tentang cara membuat Amazon VPC untuk digunakan dengan instans RDS for Oracle, lihat Membuat VPC untuk digunakan dengan basis data Oracle.

Anda juga memerlukan instans DB RDS for Oracle. Ketahui informasi tentang cara membuat instans DB RDS for Oracle untuk digunakan dengan metadata Fusion Middleware di Membuat instans DB Oracle.

Anda dapat menyimpan skema untuk setiap komponen Fusion Middleware di instans DB Amazon RDS Anda. Skema berikut telah diverifikasi bahwa penginstalannya berjalan dengan benar:

- Analytics (ACTIVITIES)
- Audit Services (IAU)
- Audit Services Append (IAU_APPEND)
- Audit Services Viewer (IAU_VIEWER)
- Discussions (DISCUSSIONS)
- Metadata Services (MDS)
- Oracle Business Intelligence (BIPLATFORM)
- Oracle Platform Security Services (OPSS)
- Portal and Services (WEBCENTER)
- Portlet Producers (PORTLET)
- Service Table (STB)
- SOA Infrastructure (SOAINFRA)
- User Messaging Service (UCSUMS)
- WebLogic Layanan (WLS)

Pedoman penggunaan RCU

Berikut adalah beberapa rekomendasi untuk bekerja dengan instans DB Anda dalam skenario ini:

- Sebaiknya gunakanlah Multi-AZ untuk beban kerja produksi. Informasi selengkapnya tentang bekerja dengan beberapa Zona Ketersediaan bisa dilihat di Wilayah, Zona Ketersediaan, dan Zona Lokal.
- Untuk keamanan tambahan, Oracle merekomendasikan agar Anda menggunakan Enkripsi Data Transparan (TDE) untuk mengenkripsi data diam. Jika Anda memiliki lisensi Enterprise Edition yang menyertakan Opsi Keamanan Lanjutan, Anda dapat mengaktifkan enkripsi data diam menggunakan opsi TDE. Untuk informasi selengkapnya, lihat Enkripsi Data Transparan Oracle.

  Amazon RDS juga menyediakan opsi enkripsi data diam untuk semua edisi basis data. Untuk informasi selengkapnya, lihat Mengenkripsi sumber daya Amazon RDS.
- Konfigurasikan Grup Keamanan VPC Anda agar komunikasi antara server aplikasi Anda dan instans DB Amazon RDS Anda bisa dilakukan. Server aplikasi yang menjadi host komponen Fusion Middleware dapat berada di Amazon EC2 atau on-premise.

Menjalankan RCU

Untuk membuat dan mengisi skema guna mendukung komponen Fusion Middleware Anda, gunakan Oracle Repository Creation Utility (RCU). Anda dapat menjalankan RCU dengan berbagai cara.

Topik

- Menjalankan RCU menggunakan baris perintah dalam satu langkah
- Menjalankan RCU menggunakan baris perintah dalam beberapa langkah
- Menjalankan RCU dalam mode interaktif

Menjalankan RCU menggunakan baris perintah dalam satu langkah

Jika Anda tidak perlu mengedit skema apa pun sebelum mengisinya, Anda dapat menjalankan RCU dalam satu langkah. Jika tidak, lihat bagian berikut untuk menjalankan RCU dalam beberapa langkah.

Anda dapat menjalankan RCU dalam mode senyap menggunakan parameter baris perintah `-silent`. Saat Anda menjalankan RCU dalam mode senyap, Anda tidak perlu mengetik kata sandi pada baris perintah dengan membuat file teks yang berisi kata sandi. Buat file teks dengan kata sandi untuk `dbUser` di baris pertama, dan kata sandi untuk setiap komponen di baris berikutnya. Anda menentukan nama file kata sandi sebagai parameter terakhir untuk perintah RCU.

Example

Contoh berikut membuat dan mengisi skema untuk komponen Infrastruktur SOA (dan dependensinya) dalam satu langkah.

Untuk Linux, macOS, atau Unix:

```
export ORACLE_HOME=/u01/app/oracle/product/12.2.1.0/fmw
export JAVA_HOME=/usr/java/jdk1.8.0_65
${ORACLE_HOME}/oracle_common/bin/rcu \
-silent \
-createRepository \
-connectString ${dbhost}:${dbport}:${dbname} \
-dbUser ${dbuser} \
-dbRole Normal \
-honorOMF \
-schemaPrefix ${SCHEMA_PREFIX} \
-component MDS \
-component STB \
-component OPSS \
-component IAU \
-component IAU_APPEND \
-component IAU_VIEWER \
-component UCSUMS \
-component WLS \
-component SOAINFRA \
-f < /tmp/passwordfile.txt
```

Untuk informasi selengkapnya, lihat Running Repository Creation Utility from the command line dalam dokumentasi Oracle.

Menjalankan RCU menggunakan baris perintah dalam beberapa langkah

Untuk mengedit skrip skema secara manual, jalankan RCU dalam beberapa langkah:

1. Jalankan RCU dalam mode Mempersiapkan Skrip untuk Pemuatan Sistem menggunakan parameter baris perintah `-generateScript` untuk membuat skrip bagi skema Anda.
2. Edit secara manual dan jalankan skrip `script_systemLoad.sql` yang dihasilkan.
3. Jalankan RCU lagi dalam mode Lakukan Pemuatan Produk menggunakan parameter baris perintah `-dataLoad` untuk mengisi skema.
4. Jalankan `script_postDataLoad.sql` skrip pembersihan yang dihasilkan.

Untuk menjalankan RCU dalam mode senyap, tentukan parameter baris perintah `-silent`. Saat Anda menjalankan RCU dalam mode senyap, Anda tidak perlu mengetik kata sandi pada baris perintah dengan membuat file teks yang berisi kata sandi. Buat file teks dengan kata sandi untuk `dbUser` di baris pertama, dan kata sandi untuk setiap komponen di baris berikutnya. Tentukan nama file kata sandi sebagai parameter terakhir untuk perintah RCU.

Example

Contoh berikut membuat skrip skema untuk komponen SOA Infrastructure dan dependensinya.

Untuk Linux, macOS, atau Unix:

```
export ORACLE_HOME=/u01/app/oracle/product/12.2.1.0/fmw
export JAVA_HOME=/usr/java/jdk1.8.0_65
${ORACLE_HOME}/oracle_common/bin/rcu \
-silent \
-generateScript \
-connectString ${dbhost}:${dbport}:${dbname} \
-dbUser ${dbuser} \
-dbRole Normal \
-honorOMF \
[-encryptTablespace true] \
-schemaPrefix ${SCHEMA_PREFIX} \
-component MDS \
-component STB \
-component OPSS \
-component IAU \
-component IAU_APPEND \
-component IAU_VIEWER \
-component UCSUMS \
-component WLS \
-component SOAINFRA \
-scriptLocation /tmp/rcuscripts \
-f < /tmp/passwordfile.txt
```

Sekarang Anda dapat mengedit skrip yang dihasilkan, terhubung ke instans DB Oracle Anda, dan menjalankan skrip. Skrip yang dihasilkan bernama `script_systemLoad.sql`. Untuk informasi tentang terhubung ke instans DB Oracle Anda, lihat Langkah 3: Hubungkan klien SQL Anda ke instans DB Oracle.

Contoh berikut mengisi skema untuk komponen SOA Infrastructure (dan dependensinya).

Untuk Linux, macOS, atau Unix:

```
export JAVA_HOME=/usr/java/jdk1.8.0_65
${ORACLE_HOME}/oracle_common/bin/rcu \
-silent \
-dataLoad \
-connectString ${dbhost}:${dbport}:${dbname} \
-dbUser ${dbuser} \
-dbRole Normal \
-honorOMF \
-schemaPrefix ${SCHEMA_PREFIX} \
-component MDS \
-component STB \
-component OPSS \
-component IAU \
-component IAU_APPEND \
-component IAU_VIEWER \
-component UCSUMS \
-component WLS \
-component SOAINFRA \
-f < /tmp/passwordfile.txt
```

Untuk menyelesaikannya, terhubung ke instans DB Oracle Anda, lalu jalankan skrip pembersihan. Skrip ini bernama `script_postDataLoad.sql`.

Untuk informasi selengkapnya, lihat Running Repository Creation Utility from the command line dalam dokumentasi Oracle.

Menjalankan RCU dalam mode interaktif

Untuk menggunakan antarmuka pengguna grafis RCU, jalankan RCU dalam mode interaktif. Sertakan parameter `-interactive` dan hilangkan parameter `-silent`. Untuk informasi selengkapnya, lihat Understanding Repository Creation Utility screens dalam dokumentasi Oracle.

Example

Contoh berikut memulai RCU dalam mode interaktif dan mengisi informasi koneksi secara otomatis.

Untuk Linux, macOS, atau Unix:

```
export ORACLE_HOME=/u01/app/oracle/product/12.2.1.0/fmw
export JAVA_HOME=/usr/java/jdk1.8.0_65
${ORACLE_HOME}/oracle_common/bin/rcu \
-interactive \
```

```
-createRepository \
-connectString ${dbhost}:${dbport}:${dbname} \
-dbUser ${dbuser} \
-dbRole Normal
```

Pemecahan masalah RCU

Berhati-hatilah dengan masalah-masalah berikut.

Topik

- Oracle Managed Files (OMF)
- Hak istimewa objek
- Enterprise Scheduler Service

Oracle Managed Files (OMF)

Amazon RDS menggunakan file data OMF untuk menyederhanakan manajemen penyimpanan. Anda dapat menyesuaikan atribut tablespace, seperti manajemen ukuran dan luas. Namun, jika Anda menentukan nama file data saat Anda menjalankan RCU, kode tablespace gagal dengan `ORA-20900`. Anda dapat menggunakan RCU dengan OMF dalam cara-cara berikut:

- Di RCU 12.2.1.0 dan yang lebih baru, gunakan parameter baris perintah `-honorOMF`.
- Di RCU 12.1.0.3 dan yang lebih baru, gunakan beberapa langkah dan edit skrip yang dibuat. Untuk informasi selengkapnya, lihat Menjalankan RCU menggunakan baris perintah dalam beberapa langkah.

Hak istimewa objek

Karena Amazon RDS adalah layanan terkelola, Anda tidak memiliki akses `SYSDBA` penuh ke instans DB RDS for Oracle Anda. Namun, RCU 12c mendukung pengguna dengan hak istimewa yang lebih rendah. Dalam kebanyakan kasus, hak istimewa pengguna master cukup untuk membuat repositori.

Akun master dapat langsung memberikan hak istimewa yang telah diberikan `WITH GRANT OPTION`. Dalam beberapa kasus, saat Anda mencoba memberikan hak akses objek `SYS`, RCU mungkin gagal dengan `ORA-01031`. Anda dapat mencoba lagi dan menjalankan prosedur `rdsadmin_util.grant_sys_object` tersimpan, seperti yang ditunjukkan pada contoh berikut:

```
BEGIN
```

```
  rdsadmin.rdsadmin_util.grant_sys_object('GV_$SESSION','MY_DBA','SELECT');
END;
/
```

Jika Anda mencoba memberikan hak istimewa SYS pada objek SCHEMA_VERSION_REGISTRY, operasi mungkin gagal dengan ORA-20199: Error in rdsadmin_util.grant_sys_object. Anda dapat mengualifikasi tabel SCHEMA_VERSION_REGISTRY$ dan tampilan SCHEMA_VERSION_REGISTRY dengan nama pemilik skema, yaitu SYSTEM, dan mencoba lagi operasi tersebut. Atau, Anda dapat membuat sinonim. Masuk sebagai pengguna master dan jalankan pernyataan berikut:

```
CREATE OR REPLACE VIEW SYSTEM.SCHEMA_VERSION_REGISTRY
  AS SELECT * FROM SYSTEM.SCHEMA_VERSION_REGISTRY$;
CREATE OR REPLACE PUBLIC SYNONYM SCHEMA_VERSION_REGISTRY FOR
 SYSTEM.SCHEMA_VERSION_REGISTRY;
CREATE OR REPLACE PUBLIC SYNONYM SCHEMA_VERSION_REGISTRY$ FOR SCHEMA_VERSION_REGISTRY;
```

Enterprise Scheduler Service

Saat Anda menggunakan RCU untuk melepaskan repositori Enterprise Scheduler Service, RCU mungkin gagal dengan Error: Component drop check failed.

Mengonfigurasi Oracle Connection Manager di instans Amazon EC2

Oracle Connection Manager (CMAN) adalah server proksi yang meneruskan permintaan koneksi ke server basis data atau server proksi lainnya. Anda dapat menggunakan CMAN untuk mengonfigurasi hal berikut:

Kontrol akses

Anda dapat membuat aturan yang menyaring permintaan klien yang ditentukan pengguna dan menerima permintaan lainnya.

Multiplexing sesi

Anda dapat menyalurkan beberapa sesi klien melalui koneksi jaringan ke tujuan server bersama.

Biasanya, CMAN berada pada host yang terpisah dari server basis data dan host klien. Untuk informasi selengkapnya, lihat Configuring Oracle Connection Manager di dokumentasi Oracle Database.

Topik

- Versi dan opsi lisensi yang didukung untuk CMAN
- Persyaratan dan batasan CMAN
- Mengonfigurasi CMAN

Versi dan opsi lisensi yang didukung untuk CMAN

CMAN mendukung Enterprise Edition pada semua versi Oracle Database yang didukung oleh Amazon RDS. Untuk informasi selengkapnya, lihat Rilis RDS for Oracle.

Anda dapat menginstal Oracle Connection Manager pada host yang terpisah dari host yang diinstal Oracle Database. Anda tidak memerlukan lisensi terpisah untuk host yang menjalankan CMAN.

Persyaratan dan batasan CMAN

Untuk memberikan pengalaman yang sepenuhnya terkelola, Amazon RDS membatasi akses ke sistem operasi. Anda tidak dapat mengubah parameter basis data yang memerlukan akses sistem operasi. Oleh karena itu, Amazon RDS tidak mendukung fitur CMAN yang mengharuskan Anda masuk ke sistem operasi.

Mengonfigurasi CMAN

Saat mengonfigurasi CMAN, sebagian besar pekerjaan dilakukan di luar basis data RDS for Oracle Anda.

Topik

Langkah 1: Konfigurasikan CMAN di instans Amazon EC2 di VPC yang sama seperti instans RDS for Oracle

Untuk mempelajari cara menyiapkan CMAN, ikuti petunjuk lengkapnya di posting blog Mengonfigurasi dan menggunakan Oracle Connection Manager di Amazon EC2 untuk Amazon RDS for Oracle.

Langkah 2: Konfigurasikan parameter basis data untuk CMAN

Untuk fitur CMAN seperti Traffic Director Mode dan multiplexing sesi, atur parameter `REMOTE_LISTENER` ke alamat instans CMAN dalam grup parameter DB. Pertimbangkan skenario berikut:

- Instans CMAN berada di sebuah host dengan alamat IP `10.0.159.100` dan menggunakan port `1521`.
- `orcla`, `orclb`, and `orclc` basis data berada di instans DB RDS for Oracle terpisah.

Tabel berikut ini menunjukkan cara penentuan nilai `REMOTE_LISTENER`. Nilai `LOCAL_LISTENER` ditentukan secara otomatis oleh Amazon RDS.

Nama instans DB	IP instans DB	Nilai pendengar lokal (ditetapkan secara otomatis)	Nilai pendengar jarak jauh (ditetapkan oleh pengguna)
`orcla`	`10.0.159.200`	`( address= (protocol=tcp) (host=10.0.159.200)`	`10.0.159.100:1521`

Nama instans DB	IP instans DB	Nilai pendengar lokal (ditetapkan secara otomatis)	Nilai pendengar jarak jauh (ditetapkan oleh pengguna)
		(port=1521))	
orclb	10.0.159.300	(address= (protocol=tcp) (host=10.0.159.300) (port=1521))	10.0.159.100:1521
orclc	10.0.159.400	(address= (protocol=tcp) (host=10.0.159.400) (port=1521))	10.0.159.100:1521

Langkah 3: Kaitkan instans DB Anda dengan grup parameter

Buat atau modifikasi instans DB Anda untuk menggunakan grup parameter yang Anda konfigurasi di Langkah 2: Konfigurasikan parameter basis data untuk CMAN. Untuk informasi selengkapnya, lihat Mengaitkan grup parameter DB dengan instans DB.

Menginstal Siebel Database di Oracle pada Amazon RDS

Anda dapat menggunakan Amazon RDS sebagai host untuk Siebel Database di instans DB Oracle. Siebel Database adalah bagian dari arsitektur aplikasi Manajemen Relasi Pelanggan (CRM) Siebel. Sebagai ilustrasi, lihat Generic architecture of Siebel business application.

Gunakan topik berikut untuk membantu menyiapkan Siebel Database di instans DB Oracle pada Amazon RDS. Anda juga dapat mengetahui cara menggunakan Amazon Web Services untuk mendukung komponen lain yang diperlukan oleh arsitektur aplikasi CRM Siebel.

> **Note**
>
> Untuk menginstal Siebel Database di Oracle pada Amazon RDS, Anda perlu menggunakan akun pengguna master. Anda tidak membutuhkan hak akses SYSDBA; hak akses pengguna master sudah cukup. Untuk informasi selengkapnya, lihat Hak akses akun pengguna master.

Lisensi dan versi

Untuk menginstal Siebel Database di Amazon RDS, Anda harus menggunakan lisensi Oracle Database Anda sendiri, dan lisensi Siebel Anda sendiri. Anda harus memiliki lisensi Oracle Database yang sesuai (dengan Dukungan dan Lisensi Pembaruan Perangkat Lunak) untuk kelas instans DB dan edisi Oracle Database. Untuk informasi selengkapnya, lihat Opsi lisensi RDS for Oracle.

Oracle Database Enterprise Edition adalah satu-satunya edisi yang disertifikasi oleh Siebel untuk skenario ini. Amazon RDS mendukung Siebel CRM versi 15.0 atau 16.0. Gunakan Oracle Database 12c Rilis 1 (12.1.0.2.0). Untuk prosedur berikut, kami menggunakan CRM Siebel versi 15.0 dan Oracle Database Rilis 1 (12.1.0.2) atau Oracle Database Rilis 2 (12.2.0.1). Untuk informasi selengkapnya, lihat Oracle Database 12c dengan Amazon RDS.

Amazon RDS mendukung peningkatan versi basis data. Untuk informasi selengkapnya, lihat Meng-upgrade versi mesin instans DB.

Sebelum Anda memulai

Sebelum memulai, Anda memerlukan Amazon VPC. Karena instans DB Amazon RDS Anda harus tersedia hanya untuk komponen Siebel Enterprise Server, dan bukan untuk Internet publik, instans DB Amazon RDS Anda di-hosting di subnet privat, memberikan keamanan yang lebih baik. Untuk

informasi tentang cara membuat Amazon VPC untuk digunakan dengan CRM Siebel, lihat Membuat VPC untuk digunakan dengan basis data Oracle.

Sebelum memulai, Anda juga memerlukan instans DB Oracle. Untuk informasi tentang cara membuat instans DB Oracle untuk digunakan dengan CRM Siebel, lihat Membuat instans DB Oracle.

Menginstal dan mengonfigurasi Siebel Database

Setelah Anda membuat instans DB Oracle, Anda dapat menginstal Siebel Database. Anda menginstal basis data tersebut dengan membuat akun pemilik dan administrator tabel, menginstal prosedur dan fungsi yang tersimpan, lalu menjalankan Wizard Konfigurasi Siebel Database. Untuk informasi selengkapnya, lihat Installing the Siebel database on the RDBMS.

Untuk menjalankan Wizard Konfigurasi Siebel Database, Anda perlu menggunakan akun pengguna master. Anda tidak membutuhkan hak akses SYSDBA; hak akses pengguna master sudah cukup. Untuk informasi selengkapnya, lihat Hak akses akun pengguna master.

Menggunakan fitur Amazon RDS lainnya dengan basis data Siebel

Setelah Anda membuat instans DB Oracle, Anda dapat menggunakan fitur Amazon RDS tambahan untuk membantu Anda menyesuaikan Siebel Database.

Mengumpulkan statistik dengan opsi Oracle Statspack

Anda dapat menambahkan fitur ke instans DB Anda melalui penggunaan opsi dalam grup opsi DB. Saat Anda membuat instans DB Oracle, Anda menggunakan grup opsi DB default. Jika Anda ingin menambahkan fitur ke basis data, Anda dapat membuat grup opsi baru untuk instans DB.

Jika Anda ingin mengumpulkan statistik performa pada Siebel Database, Anda dapat menambahkan fitur Oracle Statspack. Untuk informasi selengkapnya, lihat Oracle Statspack.

Beberapa perubahan opsi langsung diterapkan, dan beberapa perubahan opsi diterapkan pada masa pemeliharaan berikutnya untuk instans DB. Untuk informasi selengkapnya, lihat Menggunakan grup opsi. Setelah Anda membuat grup opsi yang disesuaikan, ubah instans DB Anda untuk memasangnya. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Penyelarasan performa dengan parameter

Anda mengelola konfigurasi mesin DB Anda melalui penggunaan parameter dalam grup parameter DB. Saat Anda membuat instans DB Oracle, Anda menggunakan grup parameter DB default. Jika

Anda ingin menyesuaikan konfigurasi basis data, Anda dapat membuat grup parameter baru untuk instans DB.

Saat Anda mengubah parameter, tergantung jenis parameternya, perubahan tersebut langsung diterapkan atau setelah Anda melakukan reboot instans DB secara manual. Untuk informasi selengkapnya, lihat Bekerja dengan grup parameter. Setelah Anda membuat grup parameter yang disesuaikan, ubah instans DB Anda untuk memasangnya. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

Untuk mengoptimalkan instans DB Oracle untuk CRM Siebel, Anda dapat menyesuaikan parameter tertentu. Tabel berikut menunjukkan beberapa pengaturan parameter yang direkomendasikan. Untuk informasi selengkapnya tentang penyesuaian performa CRM Siebel, lihat Siebel CRM Performance Tuning Guide.

Nama parameter	Nilai default	Panduan untuk performa Siebel CRM yang optimal
`_always_semi_join`	`CHOOSE`	`OFF`
`_b_tree_bitmap_plans`	`TRUE`	`FALSE`
`_like_with_bind_as_equality`	`FALSE`	`TRUE`
`_no_or_expansion`	`FALSE`	`FALSE`
`_optimizer_join_sel_sanity_check`	`TRUE`	`TRUE`
`_optimizer_max_permutations`	2000	100

Nama parameter	Nilai default	Panduan untuk performa Siebel CRM yang optimal
`_optimizer_sortmerge_join_enabled`	`TRUE`	`FALSE`
`_partition_view_enabled`	`TRUE`	`FALSE`
`open_cursors`	`300`	Setidaknya **`2000`**.

Membuat snapshot

Setelah Anda membuat Siebel Database, Anda dapat menyalin basis data tersebut menggunakan fitur snapshot Amazon RDS. Untuk informasi selengkapnya, lihat Membuat snapshot DB untuk instans DB Single-AZ dan Memulihkan dari snapshot DB.

Dukungan untuk komponen CRM Siebel lainnya

Selain Siebel Database, Anda juga dapat menggunakan Amazon Web Services untuk mendukung komponen lain dari arsitektur aplikasi CRM Siebel Anda. Anda dapat menemukan informasi lebih lanjut tentang dukungan yang disediakan oleh Amazon AWS untuk komponen CRM Siebel tambahan di tabel berikut.

Komponen CRM Siebel	Amazon AWS Support
Siebel Enterprise (dengan satu atau beberapa Siebel Server)	Anda dapat menjadikan instans Amazon Elastic Compute Cloud (Amazon EC2) sebagai host untuk Siebel Server. Anda dapat menggunakan Amazon EC2 untuk meluncurkan server virtual sesuai jumlah yang Anda butuhkan. Menggunakan Amazon EC2, Anda dapat menaikkan atau menurunkan skala dengan mudah untuk menangani perubahan dalam persyaratan. Untuk informasi selengkapnya, lihat Apa Itu Amazon EC2?

Komponen CRM Siebel	Amazon AWS Support
	Anda dapat menempatkan server Anda di VPC yang sama dengan instans DB Anda dan menggunakan grup keamanan VPC untuk mengakses basis data tersebut. Untuk informasi selengkapnya, lihat Bekerja dengan klaster DB dalam VPC.
Server Web (dengan Siebel Web Server Extensions)	Anda dapat menginstal beberapa Server Web pada beberapa instans EC2. Anda kemudian dapat menggunakan Elastic Load Balancing untuk mendistribusikan lalu lintas masuk di antara instans. Untuk informasi lebih lanjut, lihat Apa itu Elastic Load Balancing?
Siebel Gateway Name Server	Anda dapat menjadikan instans EC2 sebagai host untuk Siebel Gateway Name Server. Selanjutnya Anda dapat menempatkan server Anda di VPC yang sama dengan instans DB dan menggunakan grup keamanan VPC untuk mengakses basis data tersebut. Untuk informasi selengkapnya, lihat Bekerja dengan klaster DB dalam VPC.

Catatan rilis mesin Basis Data Oracle

Pembaruan pada instans DB Amazon RDS for Oracle Anda membuatnya tetap mutakhir. Jika Anda menerapkan pembaruan, Anda dapat yakin bahwa instans DB Anda sedang menjalankan versi perangkat lunak basis data yang telah diuji baik oleh Oracle maupun Amazon. Kami tidak mendukung penerapan patch satu kali untuk setiap instans DB RDS for Oracle.

Anda dapat menentukan setiap versi Basis Data Oracle yang saat ini didukung ketika Anda membuat instans DB baru. Anda dapat menentukan versi utama, seperti Basis Data Oracle 19c, dan versi minor apa pun yang didukung untuk versi utama yang telah ditentukan. Jika tidak ada versi yang ditentukan, Amazon RDS menetapkan ke versi yang didukung secara default, biasanya versi terbaru. Jika versi utama ditentukan tetapi versi minornya tidak, Amazon RDS menetapkan bawaan ke rilis terbaru versi utama yang telah Anda tentukan. Untuk melihat daftar versi yang didukung dan versi default untuk instans basis data yang baru dibuat, gunakan perintah `describe-db-engine-versions` AWS CLI.

Untuk detail tentang versi Basis Data Oracle yang didukung Amazon RDS, lihat Catatan Rilis Amazon RDS for Oracle.

Amazon RDS for PostgreSQL

Amazon RDS mendukung instans DB yang menjalankan beberapa versi PostgreSQL. Untuk daftar versi yang tersedia, lihat Versi basis data PostgreSQL yang tersedia.

> **Note**
>
> Pengakhiran PostgreSQL 9.6 dijadwalkan pada 26 April 2022. Untuk informasi selengkapnya, lihat Penghentian PostgreSQL versi 9.6.

Anda dapat membuat instance DB dan snapshot DB, point-in-time mengembalikan, dan membuat cadangan. Instans DB yang menjalankan PostgreSQL mendukung deployment Multi-AZ, replika baca, IOPS yang Tersedia, dan dapat dibuat di dalam cloud privat virtual (VPC). Anda juga dapat menggunakan Lapisan Soket Aman (SSL) untuk terhubung ke instans DB yang menjalankan PostgreSQL.

Sebelum membuat instans DB, selesaikan langkah-langkah di Menyiapkan Amazon RDS.

Anda dapat menggunakan aplikasi klien SQL standar apa pun untuk menjalankan perintah terhadap instans dari komputer klien. Aplikasi tersebut mencakup pgAdmin, alat pengembangan dan administrasi Sumber Terbuka yang populer untuk PostgreSQL, atau psql, utilitas baris perintah yang merupakan bagian dari instalasi PostgreSQL. Untuk memberikan pengalaman layanan terkelola, Amazon RDS tidak memberikan akses host ke instans DB. Hal tersebut juga membatasi akses ke prosedur dan tabel sistem tertentu yang membutuhkan hak istimewa tingkat lanjut. Amazon RDS mendukung akses ke basis data di instans DB menggunakan aplikasi klien SQL standar. Amazon RDS tidak mengizinkan akses host langsung ke instans DB dengan menggunakan Telnet atau Secure Shell (SSH).

Amazon RDS for PostgreSQL memenuhi banyak standar industri. Misalnya, Anda dapat menggunakan basis data Amazon RDS for PostgreSQL untuk membangun aplikasi yang sesuai dengan HIPAA dan untuk menyimpan informasi terkait perawatan kesehatan. Ini termasuk penyimpanan informasi kesehatan yang dilindungi (PHI) berdasarkan Perjanjian Rekanan Bisnis (BAA) lengkap dengan AWS. Amazon RDS for PostgreSQL juga memenuhi persyaratan keamanan Federal Risk and Authorization Management Program (FedRAMP). Amazon RDS for PostgreSQL telah menerima Otoritas Sementara FedRAMP Joint Authorization Board (JAB) untuk Beroperasi (P-ATO) di FedRAMP HIGH Baseline di Wilayah. AWS GovCloud (US) Untuk informasi selengkapnya tentang standar kepatuhan yang didukung, lihat kepatuhan cloud AWS.

Untuk mengimpor data ke dalam instans DB, ikuti informasi di bagian Mengimpor data ke PostgreSQL di Amazon RDS.

Topik

Tugas manajemen umum untuk Amazon RDS for PostgreSQL

Berikut adalah tugas manajemen umum yang Anda lakukan dengan instans DB Amazon RDS for PostgreSQL, dengan tautan ke dokumentasi yang relevan untuk setiap tugas.

Area tugas	Dokumentasi terkait
Menyiapkan Amazon RDS untuk penggunaan pertama kali Sebelum Anda dapat membuat instans DB, pastikan untuk menyelesaikan beberapa prasyarat. Misalnya, instans DB dibuat secara default dengan firewall yang mencegah akses ke instans DB tersebut. Oleh karena itu, Anda harus membuat grup keamanan dengan alamat IP dan konfigurasi jaringan yang benar untuk mengakses instans DB.	Menyiapkan Amazon RDS
Memahami instans DB Amazon RDS Jika membuat instans DB untuk tujuan produksi, Anda harus memahami cara kerja kelas instans, jenis penyimpanan, dan pekerjaan IOPS yang Tersedia di Amazon RDS.	Kelas instans DB Jenis penyimpanan Amazon RDS Penyimpanan SSD IOPS yang Tersedia
Menemukan versi PostgreSQL yang tersedia Amazon RDS mendukung beberapa versi PostgreSQL.	Versi basis data PostgreSQL yang tersedia
Menyiapkan ketersediaan tinggi dan dukungan failover Instans DB produksi harus menggunakan deployment Multi-AZ. Deployment Multi-AZ memberikan peningkatan ketersediaan, durabilitas data, dan toleransi kesalahan untuk instans DB.	Mengonfigurasi dan mengelola deployment Multi-AZ
Memahami jaringan Amazon Virtual Private Cloud (VPC) Jika AWS akun Anda memiliki VPC default, maka instans DB Anda secara otomatis dibuat di dalam VPC default. Dalam beberapa kasus, akun Anda mungkin tidak memiliki VPC default, dan Anda mungkin menginginkan instans DB di VPC. Dalam kasus ini, buat grup VPC dan subnet sebelum Anda membuat instans DB.	Bekerja dengan klaster DB dalam VPC

Area tugas	Dokumentasi terkait
Mengimpor data ke Amazon RDS PostgreSQL Anda dapat menggunakan berbagai alat untuk mengimpor data ke dalam instans DB PostgreSQL di Amazon RDS.	Mengimpor data ke PostgreSQL di Amazon RDS
Menyiapkan replika baca hanya baca (primer dan siaga) RDS untuk PostgreSQL mendukung replika baca di Wilayah yang AWS sama dan di Wilayah yang berbeda dari instance utama. AWS	Menggunakan replika baca instans DB Menggunakan replika baca untuk Amazon RDS for PostgreSQL Membuat replika baca di tempat yang berbeda Wilayah AWS
Memahami grup keamanan Secara default, instans DB dibuat dengan firewall yang mencegah akses ke instans tersebut. Untuk menyediakan akses melalui firewall tersebut, edit aturan masuk untuk grup keamanan VPC yang terkait dengan VPC yang meng-hosting instans DB.	Mengontrol akses dengan grup keamanan
Menyiapkan grup parameter dan fitur Untuk mengubah parameter default instans DB Anda, buat grup parameter DB kustom dan ubah pengaturannya. Jika melakukan nya sebelum membuat instans DB, Anda dapat memilih grup parameter DB kustom ketika membuat instans.	Bekerja dengan grup parameter
Menghubungkan ke sampel instans DB PostgreSQL Setelah membuat grup keamanan dan mengaitkannya ke instans DB, Anda dapat menghubungkan ke instans DB menggunakan aplikasi klien SQL standar seperti `psql` atau `pgAdmin`.	Menghubungkan ke instans DB yang menjalankan mesin basis data PostgreSQL Menggunakan SSL dengan instans DB PostgreSQL

Area tugas	Dokumentasi terkait
Mencadangkan dan memulihkan instans DB Anda dapat mengonfigurasi instans DB Anda untuk melakukan pencadangan otomatis, atau melakukan snapshot manual, kemudian memulihkan instans dari cadangan atau snapshot.	Mencadangkan, memulihkan, dan mengekspor data
Memantau performa instans DB Anda dapat memantau instans PostgreSQL DB dengan menggunakan metrik CloudWatch Amazon RDS, peristiwa, dan pemantauan yang disempurnakan.	Melihat metrik di konsol Amazon RDS Melihat peristiwa Amazon RDS
Meningkatkan versi basis data PostgreSQL Anda dapat melakukan peningkatan versi utama dan minor untuk instans DB PostgreSQL Anda.	Meningkatkan mesin DB PostgreSQL untuk Amazon RDS Memilih peningkatan versi mayor untuk PostgreSQL
Menggunakan file log Anda dapat mengakses file log untuk instans DB PostgreSQL Anda.	File log basis data RDS for PostgreSQL
Memahami praktik terbaik untuk instans DB PostgreSQL Temukan beberapa praktik terbaik untuk menggunakan PostgreSQL di Amazon RDS.	Praktik terbaik untuk menggunakan PostgreSQL

Berikut ini adalah daftar bagian lain dalam panduan ini yang dapat membantu Anda memahami dan menggunakan fitur penting RDS for PostgreSQL:

- Memahami peran dan izin PostgreSQL
- Mengontrol akses pengguna ke basis data PostgreSQL
- Bekerja dengan parameter pada instans DB RDS for PostgreSQL
- Memahami mekanisme pencatatan log yang didukung oleh RDS for PostgreSQL

- Bekerja dengan fitur autovacuum PostgreSQL di Amazon RDS for PostgreSQL
- Menggunakan server DNS khusus untuk akses jaringan keluar

Menggunakan lingkungan Pratinjau Basis Data

Komunitas PostgreSQL terus merilis versi dan ekstensi PostgreSQL baru, termasuk versi beta. Hal ini memberi pengguna PostgreSQL kesempatan untuk mencoba versi PostgreSQL baru lebih awal. Untuk mempelajari selengkapnya tentang proses rilis beta komunitas PostgreSQL, lihat Beta Information dalam dokumentasi PostgreSQL. Demikian pula, Amazon RDS membuat versi beta PostgreSQL tertentu tersedia sebagai rilis Pratinjau. Ini memungkinkan Anda membuat instans DB menggunakan versi Pratinjau dan menguji fitur-fiturnya di Lingkungan Pratinjau Basis Data.

Instans DB RDS untuk PostgreSQL di Lingkungan Pratinjau Basis Data secara fungsional mirip dengan instans RDS for PostgreSQL lainnya. Namun, Anda tidak dapat menggunakan versi Pratinjau untuk produksi.

Perhatikan batasan penting berikut:

- Semua instans DB dihapus pada 60 hari setelah pembuatannya, bersama dengan semua cadangan dan snapshot.
- Anda hanya dapat membuat instans DB dalam Cloud Privat Virtual (VPC) berdasarkan layanan Amazon VPC.
- Anda hanya dapat menggunakan SSD Tujuan Umum dan penyimpanan SSD IOPS yang Tersedia.
- Anda tidak bisa mendapatkan bantuan dari AWS Support dengan instans DB. Sebagai gantinya, Anda dapat memposting pertanyaan Anda ke komunitas Tanya Jawab yang AWS dikelola, Re:post.AWS
- Anda tidak dapat menyalin snapshot instans DB ke lingkungan produksi.

Opsi berikut didukung oleh Pratinjau.

- Anda dapat membuat instans DB hanya menggunakan jenis instans M6i, R6i, M6g, M5, T3, R6g, dan R5. Untuk informasi selengkapnya tentang kelas instans RDS, lihat Kelas instans DB .
- Anda dapat menggunakan deployment AZ tunggal dan multi-AZ.
- Anda dapat menggunakan pembuangan PostgreSQL standar dan memuat fungsi untuk mengekspor basis data dari atau mengimpor basis data ke Lingkungan Pratinjau Basis Data.

Fitur yang tidak didukung di lingkungan Pratinjau Basis Data

Fitur berikut ini tidak tersedia di lingkungan Pratinjau Basis Data:

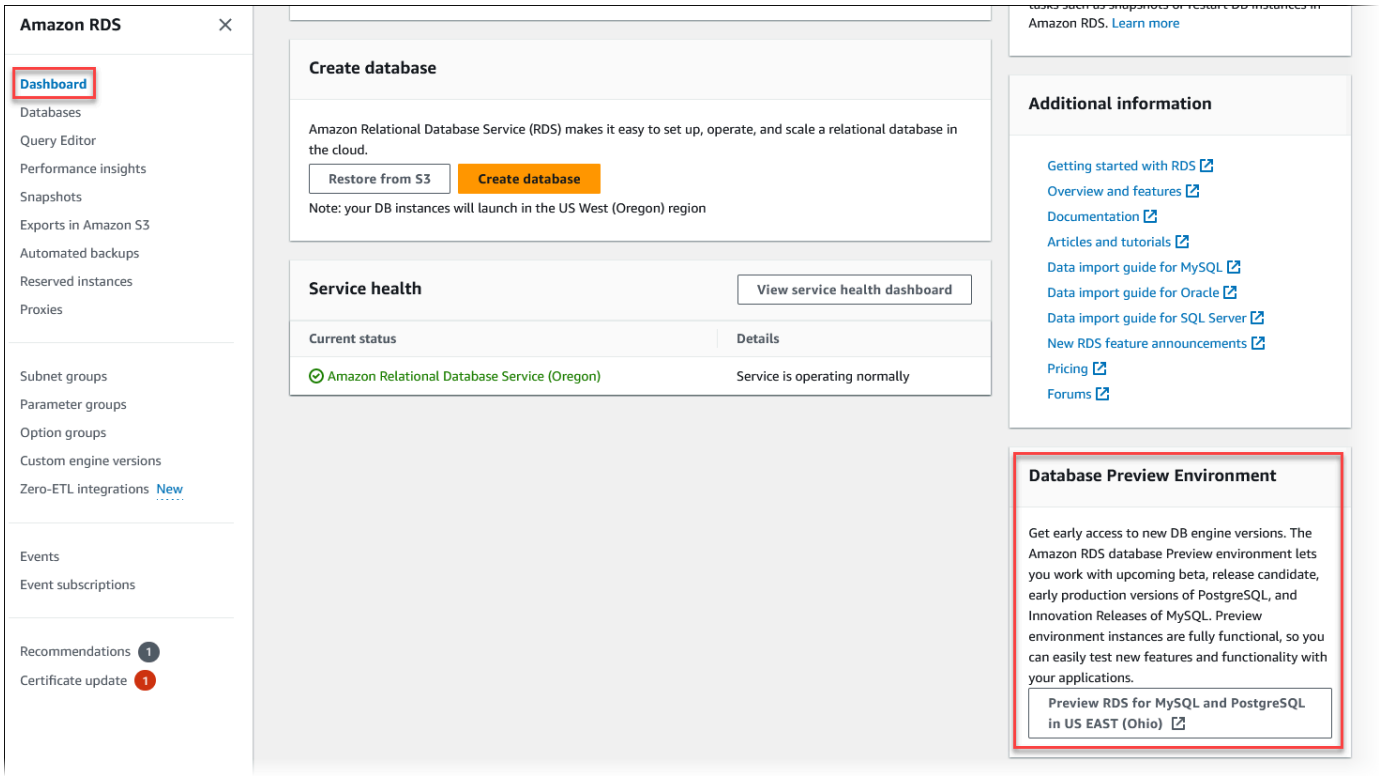
- Salinan snapshot lintas Wilayah
- Replika baca lintas Wilayah

Membuat instans DB baru di Lingkungan Pratinjau Basis Data

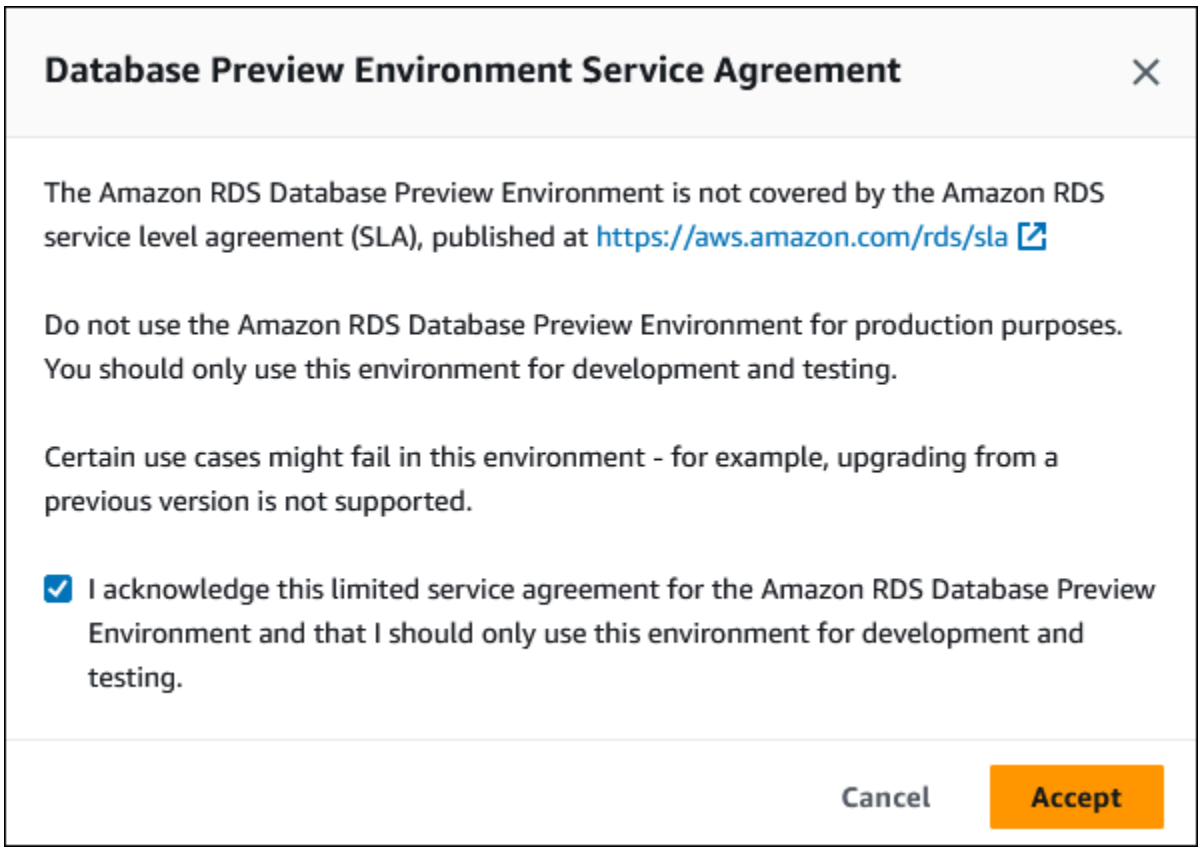
Gunakan prosedur berikut untuk membuat instans DB di lingkungan pratinjau.

Untuk membuat instans DB di lingkungan Pratinjau Basis Data

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Pilih Dasbor dari panel navigasi.
3. Di halaman Dasbor, temukan bagian Lingkungan Pratinjau Basis Data, seperti yang ditunjukkan pada gambar berikut.



Anda dapat langsung menuju Lingkungan pratinjau basis data. Sebelum dapat melanjutkan, Anda harus mengakui dan menerima batasan.



4. Untuk membuat instans DB RDS for PostgreSQL, ikuti proses yang sama seperti untuk membuat instans DB Amazon RDS apa pun. Untuk informasi lebih lanjut, lihat prosedur Konsol di Membuat instans DB.

Untuk membuat instance di Lingkungan Pratinjau Database menggunakan RDS API atau AWS CLI, gunakan endpoint berikut.

```
rds-preview.us-east-2.amazonaws.com
```

PostgreSQL versi 16 di lingkungan Pratinjau Basis Data

⚠ Ini adalah dokumentasi pratinjau untuk Amazon RDS PostgreSQL versi 16. Dokumentasi ini dapat mengalami perubahan.

> **Note**
>
> RDS for PostgreSQL versi 16 RC1, 16 Beta 3, 16 Beta 2, dan 16 Beta 1 tidak akan didukung setelah RDS for PostgreSQL versi 16.0 dirilis di lingkungan Pratinjau Basis Data.

PostgreSQL versi 16.0 sekarang tersedia di lingkungan Pratinjau Basis Data Amazon RDS. PostgreSQL versi 16 berisi beberapa perbaikan yang dijelaskan dalam dokumentasi PostgreSQL berikut:

- PostgreSQL 16 Dirilis
- PostgreSQL 16 RC1 Dirilis
- PostgreSQL 16 Beta 1 Dirilis!
- PostgreSQL 16 Beta 2 Dirilis!
- PostgreSQL 16 Beta 3 Dirilis!

Untuk informasi tentang Lingkungan Pratinjau Basis Data, lihat the section called “ Lingkungan Pratinjau Basis Data”. Untuk mengakses Lingkungan Pratinjau dari konsol, pilih https://console.aws.amazon.com/rds-preview/.

Versi basis data PostgreSQL yang tersedia

Amazon RDS mendukung instans DB yang menjalankan beberapa edisi PostgreSQL. Anda dapat menentukan versi PostgreSQL mana pun yang saat ini tersedia ketika membuat instans DB baru. Anda dapat menentukan versi utama (seperti PostgreSQL 14) dan versi minor yang tersedia untuk versi utama tertentu. Jika tidak ada versi yang ditentukan, Amazon RDS menetapkan ke versi yang tersedia secara default, biasanya versi terbaru. Jika versi utama ditentukan tetapi versi minor tidak, Amazon RDS menetapkan default ke rilis versi utama terbaru yang telah Anda tentukan.

Untuk melihat daftar versi yang tersedia, serta default untuk instance DB yang baru dibuat, gunakan perintah. `describe-db-engine-versions` AWS CLI Misalnya, untuk menampilkan versi mesin PostgreSQL default, gunakan perintah berikut:

```
aws rds describe-db-engine-versions --default-only --engine postgres
```

Untuk detail tentang versi PostgreSQL yang didukung di Amazon RDS, lihat Catatan Rilis Amazon RDS for PostgreSQL.

Jika Anda belum siap untuk meningkatkan secara manual ke versi mesin utama baru sebelum tanggal dukungan standar berakhir RDS, Amazon RDS akan secara otomatis mendaftarkan database Anda di Amazon RDS Extended Support setelah RDS berakhir pada tanggal dukungan standar. Kemudian, Anda dapat terus menjalankan RDS untuk PostgreSQL versi 11 dan lebih tinggi. Untuk informasi selengkapnya, lihat Menggunakan Dukungan Diperpanjang Amazon RDS dan Harga Amazon RDS.

Penghentian PostgreSQL versi 10

Pada 17 April 2023, Amazon RDS berencana untuk menghentikan PostgreSQL 10 menggunakan jadwal berikut. Sebaiknya Anda mengambil tindakan dan meningkatkan basis data PostgreSQL Anda yang berjalan pada versi utama 10 ke versi yang lebih baru, seperti PostgreSQL versi 14. Untuk meningkatkan instans DB RDS for PostgreSQL versi utama 10 dari versi PostgreSQL yang lebih lama dari 10.19, sebaiknya Anda meningkatkan ke versi 10.19 terlebih dahulu lalu meningkatkan ke versi 14. Untuk informasi selengkapnya, lihat Meningkatkan mesin DB PostgreSQL untuk Amazon RDS.

Tindakan atau rekomendasi	Tanggal
Komunitas PostgreSQL berencana untuk menghentikan penggunaan PostgreSQL 10 dan tidak akan memberikan patch keamanan apa pun setelah tanggal ini.	10 November 2022
Mulai tingkatkan instans DB RDS for PostgreSQL 10 ke versi utama yang lebih baru, seperti PostgreSQL 14. Meskipun Anda dapat terus memulihkan snapshot PostgreSQL 10 dan membuat replika baca dengan versi 10, perhatikan tanggal penting lainnya dalam jadwal penghentian ini serta dampaknya.	Hingga 14 Februari 2023
Setelah tanggal ini, Anda tidak dapat membuat instans Amazon RDS baru dengan PostgreSQL mayor versi 10 baik dari atau versi. AWS Management Console AWS CLI	14 Februari 2023
Setelah tanggal ini, Amazon RDS meningkatkan instans PostgreSQL 10 ke versi 14 secara otomatis. Jika Anda memulihkan snapshot basis data PostgreSQL 10, Amazon RDS otomatis meningkatkan basis data yang dipulihkan ke PostgreSQL 14.	17 April 2023

Untuk informasi selengkapnya tentang penghentian RDS untuk PostgreSQL versi 10, lihat [Pengumuman]: RDS untuk penghentian PostgreSQL 10 di re:Post. AWS

Penghentian PostgreSQL versi 9.6

Pada 31 Maret 2022, Amazon RDS berencana menghentikan PostgreSQL 9.6 menggunakan jadwal berikut. Ini memperpanjang tanggal yang diumumkan sebelumnya pada 18 Januari 2022 hingga 26 April 2022. Anda harus meningkatkan semua instans DB PostgreSQL 9.6 ke PostgreSQL 12 atau yang lebih tinggi sesegera mungkin. Sebaiknya Anda meningkatkan ke versi minor 9.6.20 atau yang lebih tinggi terlebih dahulu, lalu meningkatkan langsung ke PostgreSQL 12 daripada meningkatkan ke versi utama menengah. Untuk informasi selengkapnya, lihat Meningkatkan mesin DB PostgreSQL untuk Amazon RDS.

Tindakan atau rekomendasi	Tanggal
Komunitas PostgreSQL menghentikan dukungan untuk PostgreSQL 9.6, dan tidak akan lagi menyediakan perbaikan bug atau patch keamanan untuk versi ini.	11 November 2021
Mulai tingkatkan instans DB RDS for PostgreSQL 9.6 ke PostgreSQL 12 atau yang lebih tinggi sesegera mungkin. Meskipun Anda dapat terus memulihkan snapshot PostgreSQL 9.6 dan membuat replika baca dengan versi 9.6, perhatikan tanggal penting lainnya dalam jadwal penghentian ini serta dampaknya.	Hingga 31 Maret 2022
Setelah tanggal ini, Anda tidak dapat membuat instans Amazon RDS baru dengan PostgreSQL mayor versi 9.6 baik dari atau. AWS Management Console AWS CLI	31 Maret 2022
Setelah tanggal ini, Amazon RDS meningkat kan instans PostgreSQL 9.6 ke versi 12 secara otomatis. Jika Anda memulihkan snapshot basis data PostgreSQL 9.6, Amazon RDS otomatis meningkat kan basis data yang dipulihkan ke PostgreSQL 12.	26 April 2022

Versi usang untuk Amazon RDS for PostgreSQL

RDS for PostgreSQL 9.5 telah dihentikan sejak Maret 2021. Untuk informasi selengkapnya tentang penghentian RDS untuk PostgreSQL 9.5, lihat Memutakhirkan dari versi 9.5. Amazon RDS for PostgreSQL

Untuk mempelajari selengkapnya tentang kebijakan penghentian untuk RDS for PostgreSQL, lihat FAQ Amazon RDS. Untuk informasi selengkapnya tentang versi PostgreSQL, lihat Versioning Policy di dokumentasi PostgreSQL.

Versi ekstensi PostgreSQL yang didukung

RDS for PostgreSQL mendukung banyak ekstensi PostgreSQL. Komunitas PostgreSQL terkadang mengacu pada berbagai modul ini. Ekstensi memperluas fungsionalitas yang disediakan oleh mesin PostgreSQL. Anda dapat menemukan daftar ekstensi yang didukung oleh Amazon RDS di grup parameter DB default untuk versi PostgreSQL tersebut. Anda juga dapat melihat daftar ekstensi saat ini menggunakan `psql` dengan menampilkan parameter `rds.extensions` seperti pada contoh berikut.

```
SHOW rds.extensions;
```

> **Note**
>
> Parameter yang ditambahkan dalam rilis versi minor mungkin ditampilkan secara tidak akurat saat menggunakan parameter `rds.extensions` di `psql`.

Pada RDS for PostgreSQL 13, ekstensi tertentu dapat diinstal oleh pengguna basis data selain `rds_superuser`. Ini dikenal sebagai ekstensi tepercaya. Untuk mempelajari selengkapnya, lihat Ekstensi terpercaya PostgreSQL.

Versi RDS for PostgreSQL tertentu mendukung parameter `rds.allowed_extensions`. Parameter ini memungkinkan `rds_superuser` membatasi ekstensi yang dapat diinstal di instans DB RDS for PostgreSQL. Untuk informasi selengkapnya, lihat Membatasi penginstalan ekstensi PostgreSQL.

Untuk daftar ekstensi dan versi PostgreSQL yang didukung oleh setiap versi RDS for PostgreSQL yang tersedia, see Ekstensi PostgreSQL yang didukung di Amazon RDS di Catatan Rilis Amazon RDS for PostgreSQL.

Membatasi penginstalan ekstensi PostgreSQL

Anda dapat membatasi ekstensi yang dapat diinstal pada instans DB PostgreSQL. Secara default, parameter ini tidak ditetapkan, jadi ekstensi apa pun yang didukung dapat ditambahkan jika pengguna memiliki izin untuk melakukannya. Untuk melakukannya, tetapkan parameter `rds.allowed_extensions` ke string nama ekstensi yang dipisahkan koma. Dengan menambahkan daftar ekstensi ke parameter ini, Anda secara eksplisit mengidentifikasi ekstensi yang dapat digunakan oleh instans DB RDS for PostgreSQL Anda. Hanya ekstensi ini yang kemudian dapat diinstal di instans DB PostgreSQL.

String default untuk parameter `rds.allowed_extensions` adalah '*', yang berarti ekstensi apa pun yang tersedia untuk versi mesin dapat diinstal. Mengubah parameter `rds.allowed_extensions` tidak memerlukan mulai ulang basis data karena parameter tersebut bersifat dinamis.

Mesin instans DB PostgreSQL harus merupakan salah satu versi berikut agar Anda dapat menggunakan parameter `rds.allowed_extensions`:

- Semua versi PostgreSQL 16
- PostgreSQL 15 dan semua versi yang lebih tinggi
- PostgreSQL 14 dan semua versi yang lebih tinggi
- PostgreSQL 13.3 dan versi minor yang lebih tinggi
- PostgreSQL 12.7 dan versi minor yang lebih tinggi

Untuk melihat instalasi ekstensi yang diizinkan, gunakan perintah psql berikut.

```
postgres=> SHOW rds.allowed_extensions;
 rds.allowed_extensions
------------------------
 *
```

Jika ekstensi telah diinstal tetapi sebelumnya tidak dimasukkan dalam daftar di parameter `rds.allowed_extensions`, ekstensi tersebut masih dapat digunakan secara normal, dan perintah seperti `ALTER EXTENSION` dan `DROP EXTENSION` akan terus berfungsi. Namun, setelah ekstensi dibatasi, perintah `CREATE EXTENSION` untuk ekstensi yang dibatasi akan gagal.

Instalasi dependensi ekstensi dengan `CREATE EXTENSION CASCADE` juga dibatasi. Ekstensi dan dependensinya harus ditentukan dalam `rds.allowed_extensions`. Jika instalasi dependensi ekstensi gagal, seluruh pernyataan `CREATE EXTENSION CASCADE` akan gagal.

Jika ekstensi tidak disertakan dengan parameter `rds.allowed_extensions`, Anda akan melihat kesalahan seperti berikut jika mencoba menginstalnya.

```
ERROR: permission denied to create extension "extension-name"
HINT: This extension is not specified in "rds.allowed_extensions".
```

Ekstensi terpercaya PostgreSQL

Untuk menginstal sebagian besar ekstensi PostgreSQL membutuhkan hak istimewa `rds_superuser`. PostgreSQL 13 memperkenalkan ekstensi tepercaya, yang mengurangi kebutuhan untuk memberikan hak istimewa `rds_superuser` kepada pengguna biasa. Dengan fitur ini, pengguna dapat menginstal banyak ekstensi jika mereka memiliki hak istimewa `CREATE` pada basis data saat ini alih-alih memerlukan peran `rds_superuser`. Untuk informasi selengkapnya, lihat perintah SQL CREATE EXTENSION di dokumentasi PostgreSQL.

Berikut ini daftar ekstensi yang dapat diinstal oleh pengguna yang memiliki hak istimewa `CREATE` pada basis data saat ini dan tidak memerlukan peran `rds_superuser`:

- bool_plperl
- btree_gin
- btree_gist
- citext
- cube
- dict_int
- fuzzystrmatch
- hstore
- intarray
- isn
- jsonb_plperl
- ltree
- pg_trgm
- pgcrypto
- plperl
- plpgsql
- pltcl
- tablefunc
- tsm_system_rows
- tsm_system_time
- unaccent

- uuid-ossp

Untuk daftar ekstensi dan versi PostgreSQL yang didukung oleh setiap versi RDS for PostgreSQL yang tersedia, see Ekstensi PostgreSQL yang didukung di Amazon RDS di Catatan Rilis Amazon RDS for PostgreSQL.

Menggunakan fitur PostgreSQL yang didukung oleh Amazon RDS for PostgreSQL

Amazon RDS for PostgreSQL mendukung banyak fitur PostgreSQL yang paling umum. Misalnya, PostgreSQL memiliki fitur autovacuum yang melakukan pemeliharaan rutin pada basis data. Fitur autovacuum aktif secara default. Meskipun Anda dapat menonaktifkan fitur ini, kami sangat menyarankan Anda untuk tetap mengaktifkannya. Memahami fitur ini dan apa yang dapat Anda lakukan untuk memastikannya berfungsi sebagaimana mestinya adalah tugas dasar setiap DBA. Untuk informasi selengkapnya tentang autovacuum, lihat Bekerja dengan fitur autovacuum PostgreSQL di Amazon RDS for PostgreSQL. Untuk mempelajari lebih lanjut tentang tugas DBA umum lainnya, Tugas DBA umum untuk Amazon RDS for PostgreSQL.

RDS for PostgreSQL juga mendukung ekstensi yang menambahkan fungsionalitas penting pada instans DB. Misalnya, Anda dapat menggunakan ekstensi PostGIS untuk bekerja dengan data spasial, atau menggunakan ekstensi pg_cron untuk menjadwalkan pemeliharaan dari dalam instans. Untuk informasi selengkapnya tentang ekstensi PostgreSQL, lihat Menggunakan ekstensi PostgreSQL dengan Amazon RDS for PostgreSQL.

Pembungkus data asing adalah jenis ekstensi spesifik yang dirancang agar instans DB RDS for PostgreSQL Anda bekerja dengan basis data komersial atau jenis data lainnya. Untuk informasi selengkapnya tentang pembungkus data asing yang didukung untuk RDS for PostgreSQL, lihat Bekerja dengan pembungkus data asing yang didukung untuk Amazon RDS for PostgreSQL.

Berikut ini, Anda dapat menemukan informasi tentang beberapa fitur lain yang didukung oleh RDS for PostgreSQL.

Topik

- Jenis data kustom dan enumerasi dengan RDS for PostgreSQL
- Pemicu peristiwa untuk RDS for PostgreSQL
- Halaman besar untuk RDS for PostgreSQL
- Melakukan replikasi logis untuk Amazon RDS for PostgreSQL
- Disk RAM untuk stats_temp_directory
- Tablespace untuk RDS for PostgreSQL
- Kolasi RDS for PostgreSQL untuk EBCDIC dan migrasi mainframe lainnya

Jenis data kustom dan enumerasi dengan RDS for PostgreSQL

PostgreSQL mendukung pembuatan jenis data kustom dan dapat digunakan dengan enumerasi. Untuk informasi selengkapnya tentang membuat dan bekerja dengan enumerasi serta jenis data lainnya, lihat Enumerated types dalam dokumentasi PostgreSQL.

Berikut ini adalah contoh pembuatan suatu jenis sebagai enumerasi lalu memasukkan nilai ke dalam tabel.

```
CREATE TYPE rainbow AS ENUM ('red', 'orange', 'yellow', 'green', 'blue', 'purple');
CREATE TYPE
CREATE TABLE t1 (colors rainbow);
CREATE TABLE
INSERT INTO t1 VALUES ('red'), ( 'orange');
INSERT 0 2
SELECT * from t1;
colors
--------
red
orange
(2 rows)
postgres=> ALTER TYPE rainbow RENAME VALUE 'red' TO 'crimson';
ALTER TYPE
postgres=> SELECT * from t1;
colors
---------
crimson
orange
(2 rows)
```

Pemicu peristiwa untuk RDS for PostgreSQL

Semua versi PostgreSQL saat ini mendukung pemicu peristiwa, dan begitu juga semua versi RDS yang tersedia untuk PostgreSQL. Anda dapat menggunakan akun pengguna utama (default, `postgres`) untuk membuat, memodifikasi, mengganti nama, dan menghapus pemicu peristiwa. Pemicu peristiwa berada di tingkat instans DB, sehingga dapat diterapkan ke semua basis data pada sebuah instans.

Misalnya, kode berikut membuat pemicu peristiwa yang mencetak pengguna saat ini di akhir setiap perintah bahasa definisi data (DDL).

```
CREATE OR REPLACE FUNCTION raise_notice_func()
    RETURNS event_trigger
    LANGUAGE plpgsql AS
$$
BEGIN
    RAISE NOTICE 'In trigger function: %', current_user;
END;
$$;

CREATE EVENT TRIGGER event_trigger_1
    ON ddl_command_end
EXECUTE PROCEDURE raise_notice_func();
```

Untuk informasi lebih lanjut tentang pemicu peristiwa PostgreSQL, lihat Event triggers dalam dokumentasi PostgreSQL.

Ada beberapa batasan dalam penggunaan pemicu peristiwa PostgreSQL di Amazon RDS. Hal ini mencakup:

- Anda tidak dapat membuat pemicu peristiwa pada replika baca. Namun, Anda dapat membuat pemicu peristiwa di sumber replika baca. Pemicu peristiwa kemudian disalin ke replika baca. Pemicu peristiwa pada replika baca tidak diaktifkan pada replika baca saat perubahan didorong dari sumber. Namun, jika replika baca dipromosikan, pemicu peristiwa yang ada aktif saat operasi basis data terjadi.
- Untuk melakukan peningkatan versi utama ke instans DB PostgreSQL yang menggunakan pemicu peristiwa, hapus pemicu peristiwa sebelum meningkatkan instans tersebut.

Halaman besar untuk RDS for PostgreSQL

Halaman besar adalah fitur manajemen memori yang mengurangi overhead saat instans DB menangani potongan besar memori yang berdekatan, seperti yang digunakan oleh buffer bersama. Fitur PostgreSQL ini didukung oleh semua versi RDS for PostgreSQL yang tersedia saat ini. Anda mengalokasikan halaman besar untuk aplikasi Anda menggunakan panggilan ke memori bersama `mmap` atau `SYSV`. RDS for PostgreSQL mendukung ukuran halaman 4-KB dan 2-MB.

Anda dapat mengaktifkan atau menonaktifkan halaman besar dengan mengubah nilai parameter `huge_pages`. Fitur ini diaktifkan secara default untuk semua kelas instans DB selain kelas instans DB mikro, kecil, dan menengah.

RDS for PostgreSQL menggunakan halaman besar berdasarkan memori bersama yang tersedia. Jika instans DB tidak dapat menggunakan halaman besar karena batasan memori bersama, Amazon RDS mencegah dimulainya instans DB. Dalam kasus ini, Amazon RDS mengatur status instans DB ke parameter yang tidak kompatibel. Jika ini terjadi, Anda dapat mengatur parameter `huge_pages` ke `off` agar Amazon RDS dapat memulai instans DB.

Parameter `shared_buffers` adalah kunci untuk mengatur kumpulan memori bersama yang diperlukan untuk menggunakan halaman besar. Nilai default untuk parameter `shared_buffers` menggunakan basis data parameter makro. Makro ini mengatur persentase dari total 8 KB halaman yang tersedia untuk memori instans DB. Saat Anda menggunakan halaman besar, halaman tersebut berada dengan halaman besar. Amazon RDS menempatkan instans DB ke dalam status parameter yang tidak kompatibel jika parameter memori bersama diatur untuk memerlukan lebih dari 90 persen memori instans DB.

Untuk mempelajari lebih lanjut tentang manajemen memori PostgreSQL, lihat Resource Consumption dalam dokumentasi PostgreSQL.

Melakukan replikasi logis untuk Amazon RDS for PostgreSQL

Dimulai dengan versi 10.4, RDS for PostgreSQL mendukung publikasi dan langganan sintaks yang diperkenalkan di PostgreSQL 10. Untuk mempelajari selengkapnya, lihat Logical replication dalam dokumentasi PostgreSQL.

> **Note**
>
> Selain fitur replikasi logis PostgreSQL asli yang diperkenalkan di PostgreSQL 10, RDS for PostgreSQL juga mendukung ekstensi `pglogical`. Untuk informasi selengkapnya, lihat Menggunakan pglogical untuk menyinkronkan data di seluruh instans.

Berikut ini, Anda dapat menemukan informasi tentang pengaturan replikasi logis untuk instans DB RDS for PostgreSQL.

Topik

- Memahami replikasi logis dan decoding logis
- Menggunakan slot replikasi logis

Memahami replikasi logis dan decoding logis

RDS for PostgreSQL mendukung streaming write-ahead log (WAL) menggunakan slot replikasi logis PostgreSQL. Penggunaan decoding logis juga didukung. Anda dapat menyiapkan slot replikasi logis pada instans Anda dan mengalirkan perubahan basis data melalui slot ini ke klien seperti `pg_recvlogical`. Anda membuat slot replikasi logis di tingkat basis data, dan slot tersebut mendukung koneksi replikasi ke satu basis data.

Klien paling umum untuk replikasi logis PostgreSQL adalah AWS Database Migration Service atau host yang dikelola secara kustom pada instans Amazon EC2. Slot replikasi logis tidak memiliki informasi tentang penerima aliran. Selain itu, target tidak perlu merupakan basis data replika. Jika Anda menyiapkan slot replikasi logis dan tidak membaca dari slot, data dapat ditulis dan mengisi penyimpanan instans DB Anda dengan cepat.

Anda mengaktifkan replikasi logis PostgreSQL dan decoding logis untuk Amazon RDS dengan parameter, jenis koneksi replikasi, dan peran keamanan. Klien untuk decoding logis dapat berupa klien yang dapat membuat koneksi replikasi ke basis data pada instans DB PostgreSQL.

Cara mengaktifkan decoding logis untuk instans DB RDS for PostgreSQL

1. Pastikan akun pengguna yang Anda gunakan memiliki peran berikut:
   - Peran `rds_superuser` agar Anda dapat mengaktifkan replikasi logis
   - Peran `rds_replication` untuk memberikan izin guna mengelola slot logis dan mengalirkan data menggunakan slot logis
2. Atur parameter statis `rds.logical_replication` ke 1. Sebagai bagian dari penerapan parameter ini, atur juga parameter`wal_level`, `max_wal_senders`, `max_replication_slots`, dan `max_connections`. Perubahan parameter ini dapat meningkatkan pembuatan WAL, jadi atur parameter `rds.logical_replication` hanya saat Anda menggunakan slot logis.
3. Reboot instans DB agar parameter `rds.logical_replication` statis berlaku.
4. Buat slot replikasi logis sebagaimana dijelaskan di bagian selanjutnya. Proses ini mengharuskan Anda menentukan plugin decoding. Saat ini, RDS for PostgreSQL mendukung plugin output test_decoding dan wal2json yang dikirimkan dengan PostgreSQL.

Untuk informasi selengkapnya tentang decoding logis PostgreSQL, lihat dokumentasi PostgreSQL.

Menggunakan slot replikasi logis

Anda dapat menggunakan perintah SQL untuk bekerja dengan slot logis. Misalnya, perintah berikut membuat slot logis bernama `test_slot` menggunakan plugin output `test_decoding` PostgreSQL default.

```
SELECT * FROM pg_create_logical_replication_slot('test_slot', 'test_decoding');
slot_name    | xlog_position
-----------------+---------------
regression_slot | 0/16B1970
(1 row)
```

Untuk membuat daftar slot logis, gunakan perintah berikut.

```
SELECT * FROM pg_replication_slots;
```

Untuk membatalkan daftar slot logis, gunakan perintah berikut.

```
SELECT pg_drop_replication_slot('test_slot');
pg_drop_replication_slot
-----------------------
(1 row)
```

Untuk contoh selengkapnya tentang bekerja dengan slot replikasi logis, lihat Logical decoding examples dalam dokumentasi PostgreSQL.

Setelah Anda membuat slot replikasi logis, Anda dapat memulai pengaliran. Contoh berikut menunjukkan bagaimana decoding logis dikontrol melalui protokol replikasi streaming. Contoh ini menggunakan program pg_recvlogical, yang termasuk dalam distribusi PostgreSQL. Untuk melakukan hal ini, autentikasi klien perlu disiapkan untuk memungkinkan koneksi replikasi.

```
pg_recvlogical -d postgres --slot test_slot -U postgres
    --host -instance-name.111122223333.aws-region.rds.amazonaws.com
    -f -  --start
```

Untuk melihat isi tampilan `pg_replication_origin_status`, kueri fungsi `pg_show_replication_origin_status`.

```
SELECT * FROM pg_show_replication_origin_status();
local_id | external_id | remote_lsn | local_lsn
```

```
----------+-------------+------------+-----------
(0 rows)
```

Disk RAM untuk stats_temp_directory

Anda dapat menggunakan parameter `rds.pg_stat_ramdisk_size` RDS for PostgreSQL untuk menentukan memori sistem yang dialokasikan ke disk RAM untuk menyimpan `stats_temp_directory` PostgreSQL. Parameter disk RAM tersedia untuk semua versi PostgreSQL di Amazon RDS.

Pada beban kerja tertentu, mengatur parameter ini dapat meningkatkan performa dan menurunkan kebutuhan IO. Untuk informasi lebih lanjut tentang `stats_temp_directory`, lihat dokumentasi PostgreSQL..

Untuk mengatur disk RAM untuk `stats_temp_directory`, atur parameter `rds.pg_stat_ramdisk_size` ke nilai literal integer dalam grup parameter yang digunakan oleh instans DB Anda. Parameter ini menunjukkan MB, jadi Anda harus menggunakan nilai integer. Ekspresi, rumus, dan fungsi tidak valid untuk parameter `rds.pg_stat_ramdisk_size`. Pastikan untuk melakukan rebott instans DB agar perubahan diterapkan. Untuk informasi tentang mengatur parameter, lihat Bekerja dengan grup parameter.

Misalnya, perintah AWS CLI berikut mengatur parameter disk RAM ke 256 MB.

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name pg-95-ramdisk-testing \
    --parameters "ParameterName=rds.pg_stat_ramdisk_size, ParameterValue=256,
 ApplyMethod=pending-reboot"
```

Setelah Anda melakukan reboot, jalankan perintah berikut untuk melihat status `stats_temp_directory`.

```
postgres=> SHOW stats_temp_directory;
```

Perintah tersebut akan menghasilkan hal berikut.

```
stats_temp_directory
--------------------------
/rdsdbramdisk/pg_stat_tmp
(1 row)
```

Tablespace untuk RDS for PostgreSQL

RDS for PostgreSQL mendukung tablespace untuk kompatibilitas. Karena semua penyimpanan berada pada satu volume logis, Anda tidak dapat menggunakan tablespace untuk pemisahan atau isolasi I/O. Tolok ukur dan pengalaman kami menunjukkan bahwa satu volume logis adalah penyiapan terbaik untuk sebagian besar kasus penggunaan.

Untuk membuat dan menggunakan tablespace dengan instans DB RDS for PostgreSQL Anda memerlukan peran `rds_superuser`. Akun pengguna utama instans DB RDS for PostgreSQL Anda (nama default, `postgres`) adalah anggota peran ini. Untuk informasi selengkapnya, lihat Memahami peran dan izin PostgreSQL.

Jika Anda menentukan nama file saat membuat tablespace, awalan jalurnya adalah `/rdsdbdata/db/base/tablespace`. Contoh berikut menempatkan file tablespace di `/rdsdbdata/db/base/tablespace/data`. Contoh ini mengasumsikan bahwa pengguna (peran) `dbadmin` ada dan telah diberikan peran `rds_superuser` yang diperlukan untuk bekerja dengan tablespace.

```
postgres=> CREATE TABLESPACE act_data
  OWNER dbadmin
  LOCATION '/data';
CREATE TABLESPACE
```

Untuk mempelajari selengkapnya tentang tablespace PostgreSQL, lihat Tablespaces dalam dokumentasi PostgreSQL.

Kolasi RDS for PostgreSQL untuk EBCDIC dan migrasi mainframe lainnya

RDS for PostgreSQL versi 10 dan yang lebih tinggi termasuk ICU versi 60.2, yang didasarkan pada Unicode 10.0 dan mencakup kolasi dari Unicode Common Locale Data Repository, CLDR 32. Pustaka internasionalisasi perangkat lunak ini memastikan bahwa pengodean karakter disajikan secara konsisten, terlepas dari sistem operasi atau platform. Untuk informasi selengkapnya tentang Unicode CLDR-32, lihat CLDR 32 Release Note di situs web Unicode CLDR. Anda dapat mempelajari lebih lanjut tentang komponen internasionalisasi untuk Unicode (ICU) di situs web ICU Technical Committee (ICU-TC). Untuk informasi tentang ICU-60, lihat Download ICU 60.

Mulai dari versi 14.3, RDS for PostgreSQL juga mencakup kolasi yang membantu integrasi data dan konversi dari sistem berbasis EBCDIC. Kode pertukaran desimal kode biner yang diperluas atau pengodean EBCDIC biasanya digunakan oleh sistem operasi mainframe. Kolasi yang disediakan Amazon RDS ini didefinisikan secara sempit untuk hanya mengurutkan karakter Unicode

yang langsung dipetakan ke halaman kode EBCDIC. Karakter diurutkan dalam urutan titik kode EBCDIC untuk memungkinkan validasi data setelah konversi. Kolasi ini tidak menyertakan formulir denormalisasi, juga tidak menyertakan karakter Unicode yang tidak langsung memetakan ke karakter di halaman kode EBCDIC sumber.

Pemetaan karakter antara halaman kode EBCDIC dan titik kode Unicode didasarkan pada tabel yang diterbitkan oleh IBM. Set lengkap tersedia dari IBM sebagai file terkompresi yang dapat diunduh. RDS for PostgreSQL menggunakan pemetaan ini dengan alat yang disediakan oleh ICU untuk membuat kolasi yang tercantum dalam tabel di bagian ini. Nama kolasi mencakup bahasa dan negara seperti yang dipersyaratkan oleh ICU. Namun, halaman kode EBCDIC tidak menentukan bahasa, dan beberapa halaman kode EBCDIC mencakup beberapa negara. Itu artinya porsi bahasa dan negara dari nama kolasi dalam tabel bersifat arbitrer, dan tidak perlu cocok dengan lokal saat ini. Dengan kata lain, nomor halaman kode adalah bagian terpenting dari nama kolasi dalam tabel ini. Anda dapat menggunakan kolasi apa pun yang tertera dalam tabel berikut di basis data RDS for PostgreSQL.

- Unicode to EBCDIC collations table – Beberapa alat migrasi data mainframe secara internal menggunakan LATIN1 atau LATIN9 untuk mengodekan dan memproses data Alat tersebut menggunakan skema pulang-pergi untuk menjaga integritas data dan mendukung konversi terbalik. Kolasi dalam tabel ini dapat digunakan oleh alat yang memproses data menggunakan pengodean LATIN1, yang tidak memerlukan penanganan khusus.
- Unicode to LATIN9 collations table— Anda dapat menggunakan kolasi ini di RDS apa pun untuk basis data PostgreSQL.

Dalam tabel berikut, ada kolasi yang tersedia di RDS for PostgreSQL yang memetakan halaman kode EBCDIC ke titik kode Unicode. Kami menyarankan Anda menggunakan kolasi dalam tabel ini untuk pengembangan aplikasi yang memerlukan pengurutan berdasarkan urutan halaman kode IBM.

Nama kolasi PostgreSQL	Deskripsi pemetaan halaman kode dan pengurutan urutan
da-DK-cp277-x-icu	Karakter Unicode yang langsung memetakan ke Kode EBCDIC IBM Halaman 277 (sesuai tabel konversi) diurutkan dalam urutan titik kode IBM CP 277

Nama kolasi PostgreSQL	Deskripsi pemetaan halaman kode dan pengurutan urutan
de-DE-cp273-x-icu	Karakter Unicode yang langsung memetakan ke Kode EBCDIC IBM Halaman 273 (sesuai tabel konversi) diurutkan dalam urutan titik kode IBM CP 273
en-GB-cp285-x-icu	Karakter Unicode yang langsung memetakan ke Kode EBCDIC IBM Halaman 285 (sesuai tabel konversi) diurutkan dalam urutan titik kode IBM CP 285
en-US-cp037-x-icu	Karakter Unicode yang langsung memetakan ke Kode EBCDIC IBM Halaman 037 (sesuai tabel konversi) diurutkan dalam urutan titik kode IBM CP 37
es-ES-cp284-x-icu	Karakter Unicode yang langsung memetakan ke Kode EBCDIC IBM Halaman 284 (sesuai tabel konversi) diurutkan dalam urutan titik kode IBM CP 284
fi-FI-cp278-x-icu	Karakter Unicode yang langsung memetakan ke Kode EBCDIC IBM Halaman 278 (sesuai tabel konversi) diurutkan dalam urutan titik kode IBM CP 278
fr-FR-cp297-x-icu	Karakter Unicode yang langsung memetakan ke Kode EBCDIC IBM Halaman 297 (sesuai tabel konversi) diurutkan dalam urutan titik kode IBM CP 297
it-IT-cp280-x-icu	Karakter Unicode yang langsung memetakan ke Kode EBCDIC IBM Halaman 280 (sesuai tabel konversi) diurutkan dalam urutan titik kode IBM CP 280

Nama kolasi PostgreSQL	Deskripsi pemetaan halaman kode dan pengurutan urutan
nl-BE-cp500-x-icu	Karakter Unicode yang langsung memetakan ke Kode EBCDIC IBM Halaman 500 (sesuai tabel konversi) diurutkan dalam urutan titik kode IBM CP 500

Amazon RDS menyediakan satu set kolasi tambahan yang mengurutkan titik kode Unicode yang memetakan ke karakter LATIN9 menggunakan tabel yang diterbitkan oleh IBM, dalam urutan titik kode asli sesuai dengan halaman kode EBCDIC dari data sumber.

Nama kolasi PostgreSQL	Deskripsi pemetaan halaman kode dan pengurutan urutan
da-DK-cp1142m-x-icu	Karakter Unicode yang memetakan ke karakter LATIN9 yang awalnya dikonversi dari Kode EBCDIC IBM Halaman 1142 (sesuai tabel konversi) diurutkan dalam urutan titik kode IBM CP 1142
de-DE-cp1141m-x-icu	Karakter Unicode yang memetakan ke karakter LATIN9 yang awalnya dikonversi dari Kode EBCDIC IBM Halaman 1141 (sesuai tabel konversi) diurutkan dalam urutan titik kode IBM CP 1141
en-GB-cp1146m-x-icu	Karakter Unicode yang memetakan ke karakter LATIN9 yang awalnya dikonversi dari Kode EBCDIC IBM Halaman 1146 (sesuai tabel konversi) diurutkan dalam urutan titik kode IBM CP 1146
en-US-cp1140m-x-icu	Karakter Unicode yang memetakan ke karakter LATIN9 yang awalnya dikonversi dari Kode EBCDIC IBM Halaman 1140 (sesuai tabel

Nama kolasi PostgreSQL	Deskripsi pemetaan halaman kode dan pengurutan urutan
	konversi) diurutkan dalam urutan titik kode IBM CP 1140
es-ES-cp1145m-x-icu	Karakter Unicode yang memetakan ke karakter LATIN9 yang awalnya dikonversi dari Kode EBCDIC IBM Halaman 1145 (sesuai tabel konversi) diurutkan dalam urutan titik kode IBM CP 1145
fi-FI-cp1143m-x-icu	Karakter Unicode yang memetakan ke karakter LATIN9 yang awalnya dikonversi dari Kode EBCDIC IBM Halaman 1143 (sesuai tabel konversi) diurutkan dalam urutan titik kode IBM CP 1143
fr-FR-cp1147m-x-icu	Karakter Unicode yang memetakan ke karakter LATIN9 yang awalnya dikonversi dari Kode EBCDIC IBM Halaman 1147 (sesuai tabel konversi) diurutkan dalam urutan titik kode IBM CP 1147
it-IT-cp1144m-x-icu	Karakter Unicode yang memetakan ke karakter LATIN9 yang awalnya dikonversi dari Kode EBCDIC IBM Halaman 1144 (sesuai tabel konversi) diurutkan dalam urutan titik kode IBM CP 1144
nl-BE-cp1148m-x-icu	Karakter Unicode yang memetakan ke karakter LATIN9 yang awalnya dikonversi dari Kode EBCDIC IBM Halaman 1148 (sesuai tabel konversi) diurutkan dalam urutan titik kode IBM CP 1148

Berikut ini, Anda dapat menemukan contoh penggunaan RDS untuk kolasi PostgreSQL.

```
db1=> SELECT pg_import_system_collations('pg_catalog');
 pg_import_system_collations
----------------------------
                         36
db1=> SELECT '¤' < 'a' col1;
 col1
------
 t
db1=> SELECT '¤' < 'a' COLLATE "da-DK-cp277-x-icu" col1;
 col1
------
 f
```

Kami menyarankan Anda menggunakan kolasi di Unicode to EBCDIC collations table dan di Unicode to LATIN9 collations table untuk pengembangan aplikasi yang memerlukan pengurutan berdasarkan urutan halaman kode IBM. Kolasi berikut (akhiran dengan huruf “b”) juga terlihat di `pg_collation`, tetapi dimaksudkan untuk digunakan oleh integrasi data mainframe dan alat migrasi di AWS yang memetakan halaman kode dengan pergeseran titik kode tertentu dan memerlukan penanganan khusus dalam kolasi. Dengan kata lain, penggunaan kolasi berikut tidak direkomendasikan.

- da-DK-cp277b-x-icu
- da-DK-1142b-x-icu
- de-DE-cp273b-x-icu
- de-DE-cp1141b-x-icu
- en-GB-cp1146b-x-icu
- en-GB-cp285b-x-icu
- en-US-cp037b-x-icu
- en-US-cp1140b-x-icu
- es-ES-cp1145b-x-icu
- es-ES-cp284b-x-icu
- fi-FI-cp1143b-x-icu
- fr-FR-cp1147b-x-icu
- fr-FR-cp297b-x-icu
- it-IT-cp1144b-x-icu
- it-IT-cp280b-x-icu

- nl-BE-cp1148b-x-icu
- nl-BE-cp500b-x-icu

Untuk mempelajari lebih lanjut tentang memigrasi aplikasi dari lingkungan mainframe ke AWS, lihat Apa itu AWS Mainframe Modernization?.

Untuk mempelajari selengkapnya tentang mengelola kolasi di PostgreSQL, lihat Collation Support dalam dokumentasi PostgreSQL.

Menghubungkan ke instans DB yang menjalankan mesin basis data PostgreSQL

Setelah Amazon RDS menyediakan instans DB, Anda dapat menggunakan aplikasi klien SQL standar untuk terhubung ke instans. Sebelum Anda dapat terhubung ke instans DB, instans tersebut harus tersedia dan dapat diakses. Apakah Anda dapat terhubung ke instans dari luar VPC atau tidak tergantung pada cara Anda membuat instans Amazon RDS DB:

- Jika Anda membuat instans DB sebagai publik, perangkat dan instans Amazon EC2 di luar VPC dapat terhubung ke basis data Anda.
- Jika Anda membuat instans DB sebagai publik, hanya perangkat dan instans Amazon EC2 di dalam Amazon VPC yang dapat terhubung ke basis data Anda.

Untuk terhubung ke instans DB Anda dari instans EC2, Anda dapat menginstal klien PostgreSQL pada instans EC2. Untuk menginstal psql di Amazon Linux 2023, jalankan perintah berikut:

```
sudo dnf install postgresql15
```

Untuk menginstal psql di Amazon Linux 2, jalankan perintah berikut:

```
sudo amazon-linux-extras install postgresql14
```

Untuk menginstal psql di Ubuntu, jalankan perintah berikut:

```
sudo apt-get install -y postgresql14
```

Untuk memeriksa apakah instans DB Anda bersifat publik atau pribadi, gunakan AWS Management Console untuk melihat tab Konektivitas & keamanan untuk instans Anda. Di bagian Keamanan, Anda dapat menemukan nilai "Dapat diakses publik", dengan Tidak untuk pribadi, Ya untuk publik.

Untuk mempelajari selengkapnya tentang berbagai konfigurasi Amazon RDS dan Amazon VPC serta pengaruhnya terhadap aksesibilitas, lihat Skenario untuk mengakses instans DB di VPC.

Jika instans DB tersedia dan dapat diakses, Anda dapat terhubung dengan memberikan informasi berikut ke aplikasi klien SQL:

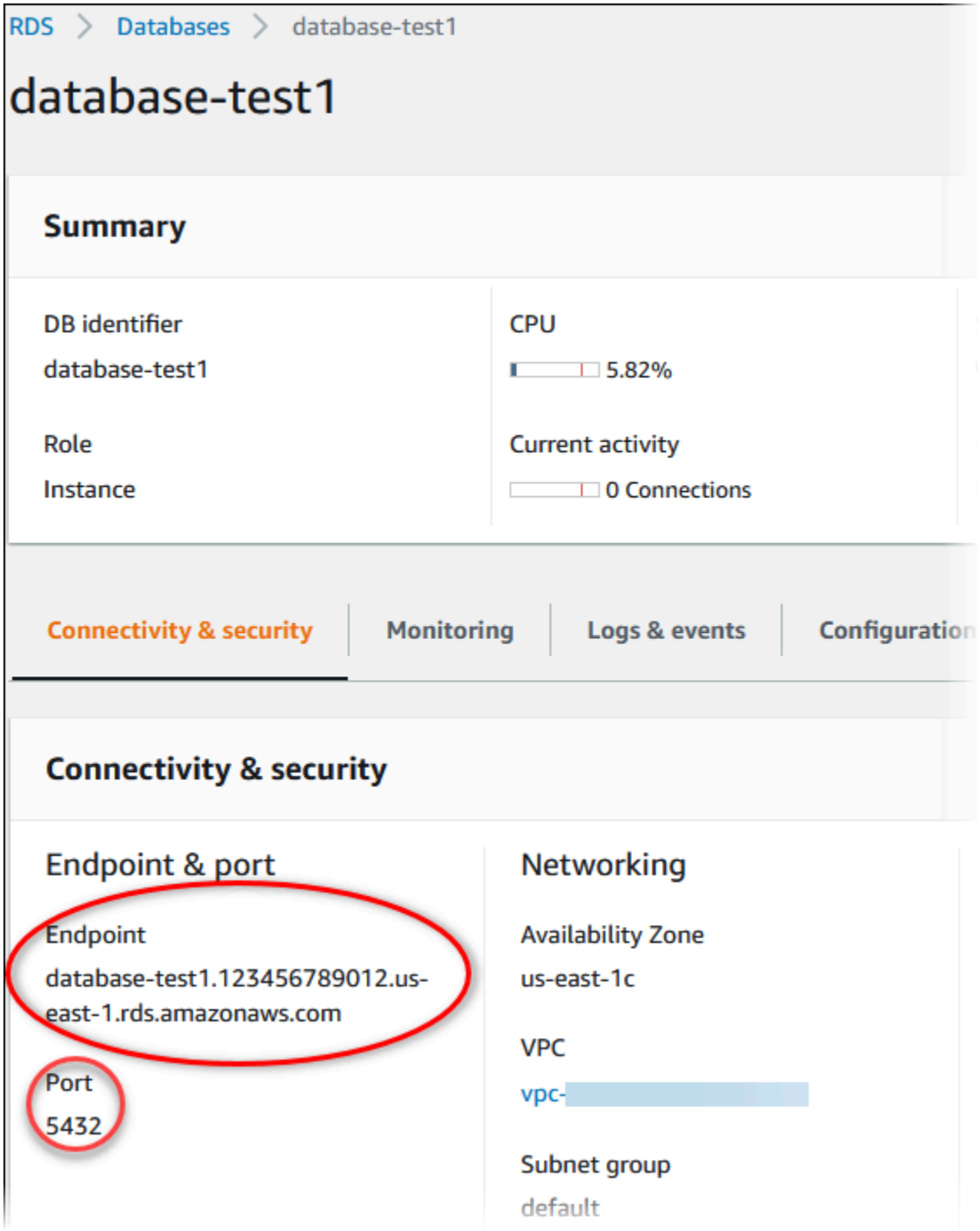
- Titik akhir instans DB, yang berfungsi sebagai nama host (nama DNS) untuk instans.

- Port tempat instans DB penulis mendengarkan. Port default untuk PostgreSQL adalah 5432.
- Nama pengguna dan kata sandi untuk instans DB. Default 'nama pengguna utama' untuk PostgreSQL adalah `postgres`.
- Nama dan kata sandi basis data (nama DB).

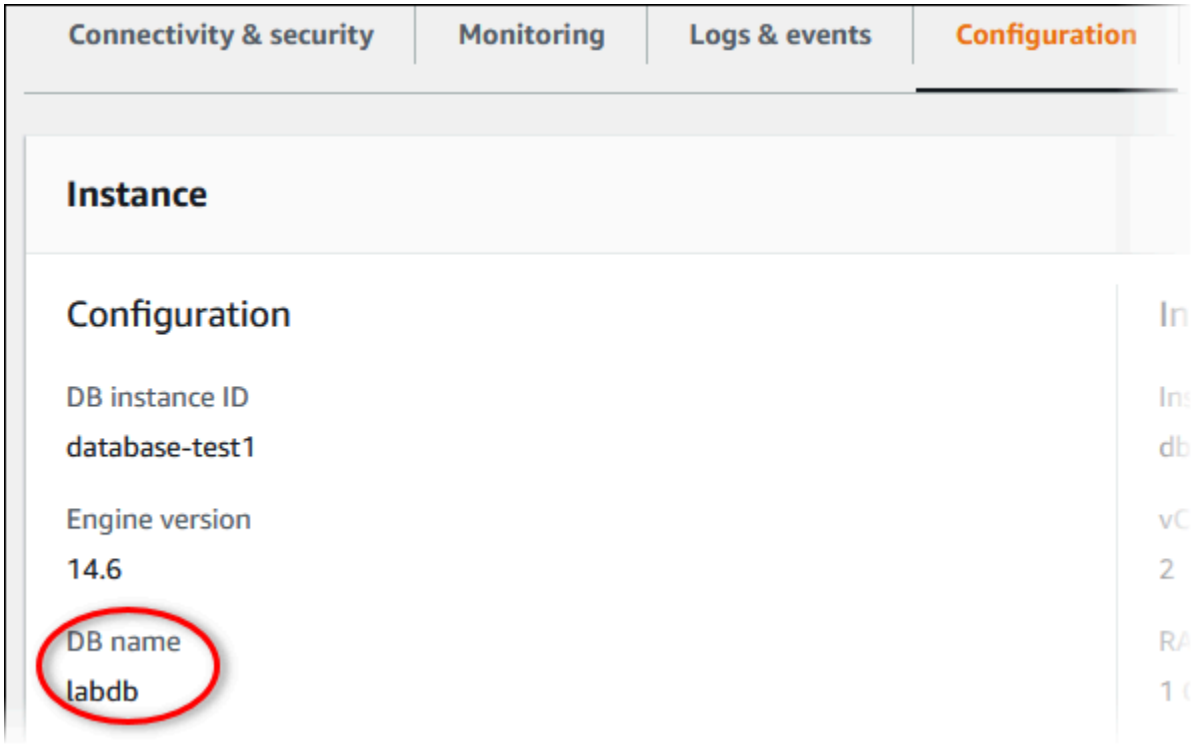
Anda dapat memperoleh detail ini menggunakan AWS Management Console, perintah describe-db-instances AWS CLI, atau operasi Amazon RDS API DescribeDBInstances.

Untuk menemukan titik akhir, nomor port, dan nama DB menggunakan AWS Management Console

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Buka konsol RDS, lalu pilih Basis data untuk menampilkan daftar instans DB Anda.
3. Pilih nama instans PostgreSQL DB untuk menampilkan detailnya.
4. Di tab Konektivitas & keamanan, salin titik akhir. Selain itu, catat nomor port. Anda memerlukan titik akhir dan nomor port untuk terhubung ke instans DB.



5. Pada tab Konfigurasi, perhatikan nama DB. Jika Anda membuat basis data saat membuat instans RDS for PostgreSQL, Anda melihat nama yang tercantum di bawah nama DB. Jika Anda tidak membuat basis data, nama DB akan menampilkan tanda hubung (-).



Berikut ini adalah dua cara untuk terhubung dengan instans PostgreSQL DB. Contoh pertama menggunakan pgAdmin, alat administrasi dan pengembangan sumber terbuka yang populer untuk PostgreSQL. Contoh kedua menggunakan psql, utilitas baris perintah yang merupakan bagian dari penginstalan PostgreSQL.

Topik

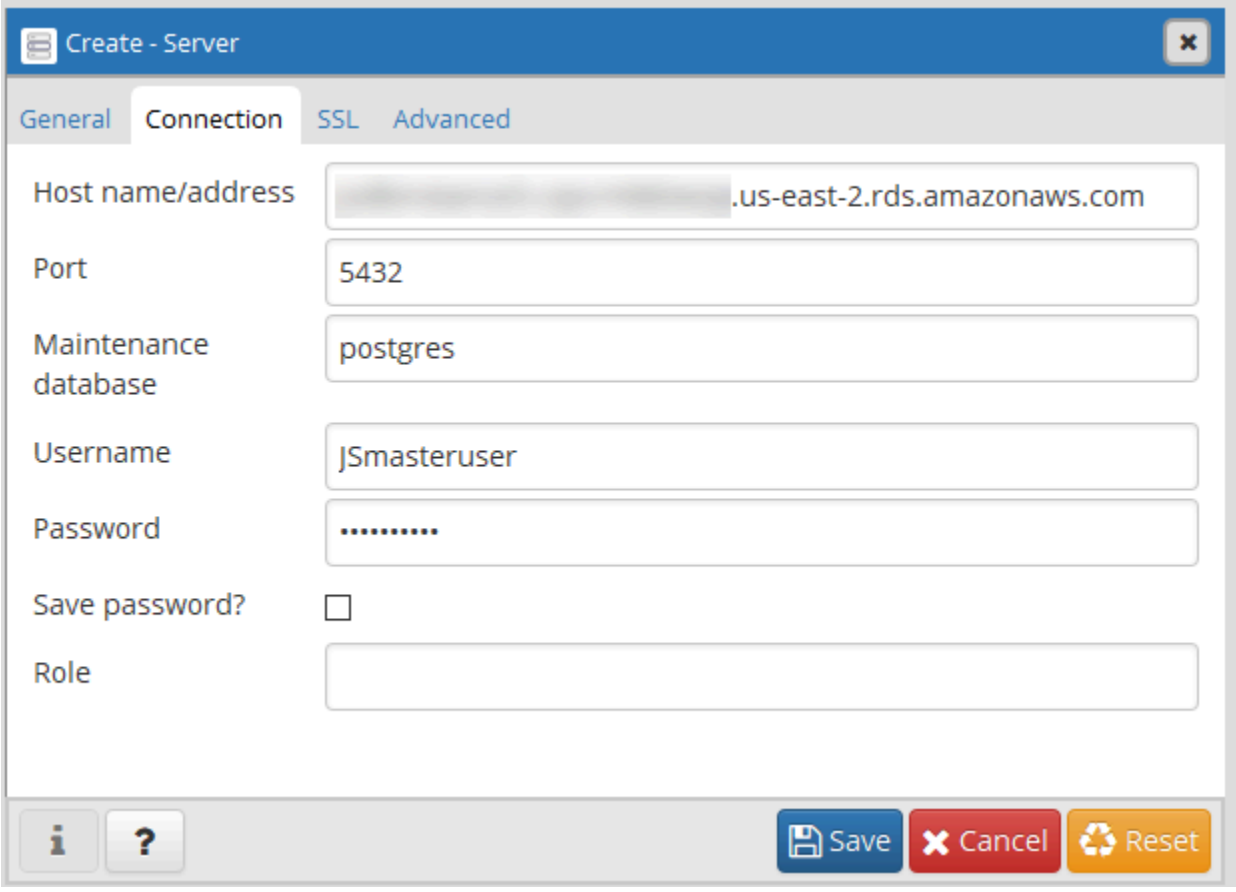
- Menggunakan pgAdmin untuk terhubung ke instans RDS for PostgreSQL DB
- Menggunakan psql untuk terhubung ke instans RDS for PostgreSQL DB Anda
- Menghubungkan dengan AWS JDBC Driver for PostgreSQL
- Memecahkan masalah koneksi ke instans RDS for PostgreSQL Anda

Menggunakan pgAdmin untuk terhubung ke instans RDS for PostgreSQL DB

Anda dapat menggunakan alat sumber terbuka pgAdmin untuk terhubung ke instans RDS for PostgreSQL DB. Anda dapat mengunduh dan menginstal pgAdmin dari http://www.pgadmin.org/ tanpa memiliki instans lokal PostgreSQL di komputer klien Anda.

Untuk terhubung ke instans RDS for PostgreSQL DB menggunakan pgAdmin

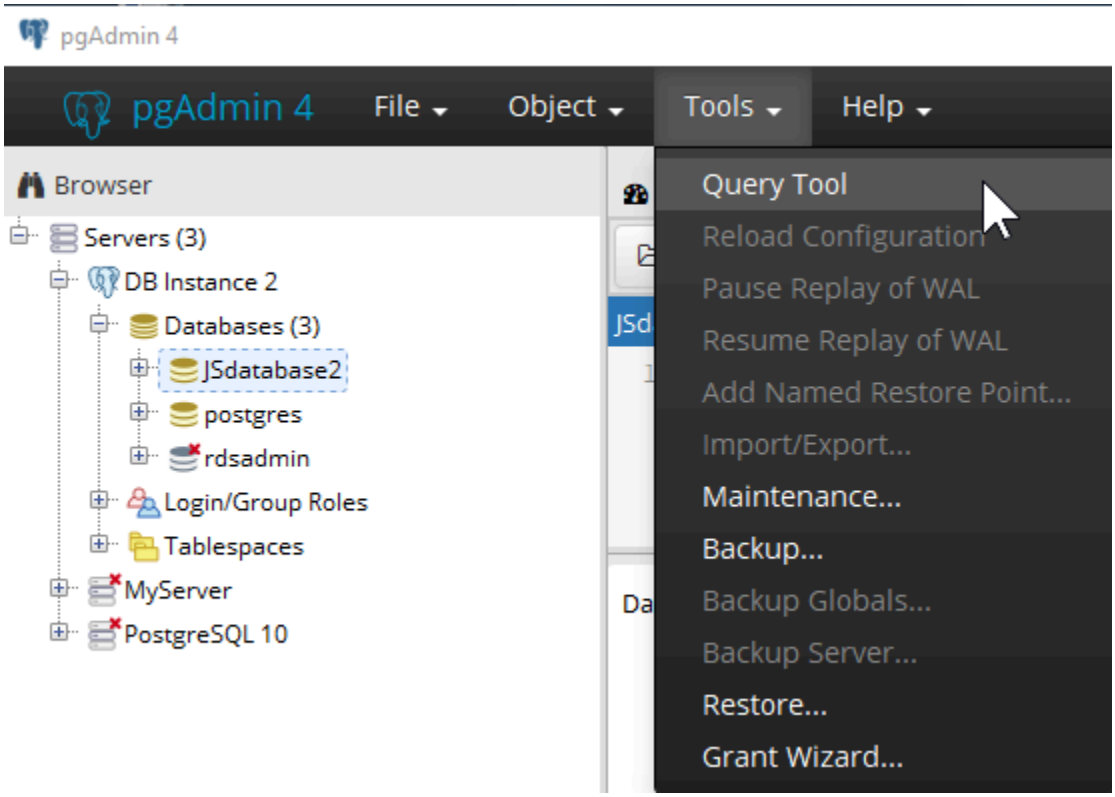
1. Luncurkan aplikasi pgAdmin di komputer klien Anda.
2. Pada tab Dasbor, pilih Tambahkan Server Baru.
3. Di kotak dialog Buat - Server, ketik nama pada tab Umum untuk mengidentifikasi server di pgAdmin.
4. Di tab Koneksi, ketik informasi berikut dari instans DB Anda:
   - Untuk Host, ketik titik akhir, misalnya `mypostgresql.c6c8dntfzzhgv0.us-east-2.rds.amazonaws.com`.
   - Untuk Port, ketik port yang ditetapkan.
   - Untuk Nama pengguna, ketik nama pengguna yang Anda masukkan saat membuat instans DB (jika Anda mengubah 'nama pengguna utama' dari default, `postgres`).
   - Untuk Kata sandi, ketik kata sandi yang Anda masukkan saat membuat instans DB.



5. Pilih Simpan.

Jika Anda mengalami masalah saat menghubungkan, lihat Memecahkan masalah koneksi ke instans RDS for PostgreSQL Anda.

6. Untuk mengakses basis data di browser pgAdmin, perluas Server, instans DB, dan Basis data. Pilih nama basis data instans DB.



7. Untuk membuka panel tempat Anda dapat memasukkan perintah SQL, pilih Alat, Alat Queri.

Menggunakan psql untuk terhubung ke instans RDS for PostgreSQL DB Anda

Anda dapat menggunakan instans lokal dari utilitas baris perintah psql untuk terhubung ke instans RDS for PostgreSQL DB. Anda memerlukan menginstal klien PostgreSQL atau psql pada komputer klien Anda.

Anda dapat mengunduh klien PostgreSQL dari situs web PostgreSQL. Ikuti petunjuk khusus untuk versi sistem operasi Anda guna menginstal psql.

Untuk terhubung ke instans RDS for PostgreSQL DB menggunakan psql, Anda perlu memberikan informasi host (DNS), kredensial akses, dan nama basis data.

Gunakan salah satu format berikut untuk terhubung ke instans RDS for PostgreSQL DB. Saat terhubung, Anda diminta memasukkan kata sandi. Untuk pekerjaan atau skrip batch, gunakan opsi `--no-password`. Opsi ini ditetapkan untuk seluruh sesi.

> **Note**
>
> Upaya koneksi dengan `--no-password` akan gagal saat server memerlukan autentikasi kata sandi dan kata sandi tidak tersedia dari sumber lain. Lihat mengetahui informasi selengkapnya, lihat dokumentasi psql.

Jika Anda pertama kali terhubung ke instans DB ini, atau jika Anda belum membuat basis data untuk instans RDS for PostgreSQL ini, Anda dapat terhubung ke basis data postgres menggunakan 'nama pengguna utama' dan kata sandi.

Untuk Unix, gunakan format berikut.

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

Untuk Unix, gunakan format berikut.

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

Misalnya, perintah berikut menghubungkan ke basis data bernama `mypgdb` pada instans PostgreSQL DB bernama `mypostgresql` menggunakan kredensial fiktif.

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

Menghubungkan dengan AWS JDBC Driver for PostgreSQL

AWS JDBC Driver for PostgreSQL adalah pembungkus klien yang dirancang untuk RDS for PostgreSQL. AWS JDBC Driver for PostgreSQL memperluas fungsionalitas driver PGJDBC komunitas dengan mengaktifkan fitur AWS seperti autentikasi. Untuk mengetahui informasi selengkapnya tentang AWS JDBC Driver for PostgreSQL dan petunjuk lengkap untuk menggunakannya, lihat Repositori GitHub AWS JDBC Driver for PostgreSQL.

AWS JDBC Driver for PostgreSQL mendukung autentikasi basis data AWS Identity and Access Management(IAM) dan AWS Secrets Manager. Untuk mengetahui informasi selengkapnya tentang penggunaan mekanisme autentikasi ini dengan driver, lihat Plugin Autentikasi AWS IAM dan Plugin AWS Secrets Manager di repositori GitHub AWS JDBC Driver for PostgreSQL.

Untuk mengetahui informasi selengkapnya tentang autentikasi basis data IAM, lihat Autentikasi basis data IAM untuk MariaDB, MySQL, dan PostgreSQL. Untuk mengetahui informasi selengkapnya tentang Secrets Manager, lihat Panduan Pengguna AWS Secrets Manager.

Memecahkan masalah koneksi ke instans RDS for PostgreSQL Anda

Topik

- Kesalahan — FATAL: basis data name tidak ada
- Kesalahan — Tidak dapat terhubung ke server: Waktu koneksi habis
- Kesalahan dengan aturan akses grup keamanan

Kesalahan — FATAL: basis data *name* tidak ada

Jika Anda menerima pesan kesalahan seperti `FATAL: database name does not exist` saat menghubungkan, coba gunakan nama basis data default postgres untuk opsi `--dbname`.

Kesalahan — Tidak dapat terhubung ke server: Waktu koneksi habis

Jika Anda tidak dapat terhubung ke instans DB, kesalahan yang paling umum adalah `Could not connect to server: Connection timed out.`. Jika Anda menerima kesalahan ini, periksa hal berikut:

- Periksa bahwa nama host yang digunakan adalah titik akhir instans DB dan bahwa nomor port yang digunakan sudah benar.

- Pastikan bahwa aksesibilitas publik instans DB tersebut diatur ke Ya untuk mengizinkan koneksi eksternal. Untuk mengubah pengaturan Akses publik, lihat Memodifikasi instans DB Amazon RDS.
- Pastikan bahwa pengguna yang terhubung ke basis data memiliki akses CONNECT. Anda dapat menggunakan kueri berikut untuk memberikan akses koneksi ke basis data tersebut.

```
GRANT CONNECT ON DATABASE database name TO username;
```

- Periksa bahwa grup keamanan yang ditetapkan ke instans DB memiliki aturan untuk memungkinkan akses melalui firewall yang mungkin dilalui koneksi Anda. Misalnya, jika instans DB dibuat menggunakan port default 5432, dan perusahaan Anda mungkin memiliki aturan firewall yang memblokir koneksi ke port tersebut dari perangkat perusahaan eksternal.

  Untuk memperbaikinya, ubah instans DB untuk menggunakan port yang berbeda. Selain itu, pastikan bahwa grup keamanan yang diterapkan ke instans DB memungkinkan koneksi ke port baru. Untuk mengubah pengaturan Port basis data, lihat Memodifikasi instans DB Amazon RDS.
- Lihat juga Kesalahan dengan aturan akses grup keamanan.

Kesalahan dengan aturan akses grup keamanan

Sejauh ini, masalah koneksi yang paling umum adalah dengan aturan akses grup keamanan yang ditetapkan untuk instans DB. Jika Anda menggunakan grup keamanan default saat membuat instans DB, grup keamanan kemungkinan tidak memiliki aturan akses yang memungkinkan Anda mengakses instans tersebut.

Agar koneksi dapat berfungsi, grup keamanan yang Anda tetapkan ke instans DB pada pembuatannya harus mengizinkan akses ke instans DB. Misalnya, jika instans DB dibuat di VPC, instans tersebut harus memiliki grup keamanan VPC yang mengotorisasi koneksi. Periksa apakah instans DB dibuat menggunakan grup keamanan yang tidak mengotorisasi koneksi dari perangkat atau instans Amazon EC2 tempat aplikasi tersebut berjalan.

Anda dapat menambahkan atau mengedit aturan masuk di grup keamanan. Untuk Sumber, memilih IP Saya akan mengizinkan akses ke instans DB dari alamat IP yang terdeteksi di browser Anda. Untuk mengetahui informasi selengkapnya, lihat Memberikan akses ke instans DB di VPC Anda dengan membuat grup keamanan.

Atau, jika instans DB dibuat di luar VPC, instans tersebut harus memiliki grup keamanan basis data yang mengotorisasi koneksinya.

Untuk mengetahui informasi selengkapnya tentang grup keamanan Amazon RDS, lihat Mengontrol akses dengan grup keamanan.

Mengamankan koneksi ke RDS for PostgreSQL dengan SSL/TLS

RDS for PostgreSQL mendukung enkripsi Secure Socket Layer (SSL) untuk instans DB PostgreSQL. Dengan menggunakan SSL, Anda dapat mengenkripsi koneksi PostgreSQL antara aplikasi Anda dan instans DB PostgreSQL Anda. Anda juga dapat memaksa semua koneksi ke instans DB PostgreSQL Anda untuk menggunakan SSL. RDS for PostgreSQL juga mendukung Keamanan Lapisan Pengangkutan (TLS), protokol penerus SSL.

Untuk mempelajari lebih lanjut tentang Amazon RDS dan perlindungan data, termasuk mengenkripsi koneksi menggunakan SSL/TLS, lihat Perlindungan data di Amazon RDS.

Topik

- Menggunakan SSL dengan instans DB PostgreSQL
- Memperbarui aplikasi untuk terhubung ke instans DB PostgreSQL menggunakan sertifikat SSL/TLS baru

Menggunakan SSL dengan instans DB PostgreSQL

Amazon RDS mendukung enkripsi Secure Socket Layer (SSL) untuk instans DB PostgreSQL. Dengan menggunakan SSL, Anda dapat mengenkripsi koneksi PostgreSQL antara aplikasi Anda dan instans DB PostgreSQL Anda. Secara default, RDS for PostgreSQL menggunakan dan mengharapkan semua klien untuk terhubung menggunakan SSL/TLS, tetapi Anda juga dapat memerlukannya. RDS untuk PostgreSQL mendukung Transport Layer Security (TLS) versi 1.1, 1.2, dan 1.3.

Untuk informasi umum tentang dukungan SSL dan basis data PostgreSQL, lihat SSL support dalam dokumentasi PostgreSQL. Untuk informasi tentang menggunakan koneksi SSL melalui JDBC, lihat Configuring the client dalam dokumentasi PostgreSQL.

Dukungan SSL tersedia di semua AWS Wilayah untuk PostgreSQL. Amazon RDS membuat sertifikat SSL untuk instans DB PostgreSQL Anda ketika instans dibuat. Jika Anda mengaktifkan verifikasi sertifikat SSL, maka sertifikat SSL mencakup titik akhir instans DB sebagai Nama Umum (CN) untuk sertifikat SSL untuk melindungi dari serangan spoofing.

Topik

- Menghubungkan ke instans DB PostgreSQL melalui SSL
- Mewajibkan koneksi SSL ke instans DB PostgreSQL

- Menentukan status koneksi SSL
- Rangkaian penyandian SSL di RDS for PostgreSQL

Menghubungkan ke instans DB PostgreSQL melalui SSL

Untuk menghubungkan ke instans DB PostgreSQL melalui SSL

1. Unduh sertifikatnya.

   Untuk informasi tentang mengunduh sertifikat, lihat .
2. Hubungkan instans DB PostgreSQL Anda melalui SSL.

   Saat Anda menghubungkan menggunakan SSL, klien Anda dapat memilih untuk memverifikasi rantai sertifikat. Jika parameter koneksi Anda menentukan `sslmode=verify-ca` atau `sslmode=verify-full`, kemudian klien Anda mengharuskan sertifikat RDS CA berada di penyimpanan terpercaya atau dirujuk di URL koneksi. Persyaratan ini untuk memverifikasi rantai sertifikat yang menandatangani sertifikat basis data Anda.

   Ketika klien, seperti psql atau JDBC, dikonfigurasikan dengan dukungan SSL, klien pertama kali mencoba untuk terhubung ke basis data dengan SSL secara default. Jika klien tidak dapat terhubung dengan SSL, maka akan kembali ke koneksi tanpa SSL. Mode `sslmode` default yang digunakan berbeda antara klien berbasis libpq (seperti psql) dan JDBC. Klien berbasis libpq secara default diatur ke `prefer`, dan klien JDBC secara default diatur ke `verify-full`.

   Gunakan parameter `sslrootcert` untuk mereferensikan sertifikat, misalnya `sslrootcert=rds-ssl-ca-cert.pem`.

Berikut ini adalah contoh penggunaan `psql` untuk terhubung ke instans DB PostgreSQL menggunakan SSL dengan verifikasi sertifikat.

```
$ psql "host=db-name.555555555555.ap-southeast-1.rds.amazonaws.com
    -p 5432 dbname=testDB user=testuser sslrootcert=rds-ca-rsa2048-g1.pem
 sslmode=verify-full"
```

Mewajibkan koneksi SSL ke instans DB PostgreSQL

Anda dapat mewajibkan koneksi tersebut ke instans DB RDS for PostgreSQL Anda dengan SSL menggunakan parameter `rds.force_ssl`. Parameter `rds.force_ssl` default diatur ke 1 (aktif)

untuk RDS for PostgreSQL versi 15. Semua RDS for PostgreSQL versi utama 14 lainnya dan yang lebih lama memiliki nilai default untuk parameter `rds.force_ssl` yang diatur ke 0 (nonaktif). Anda dapat mengatur parameter `rds.force_ssl` ke 1 (aktif) guna mewajibkan SSL untuk koneksi dengan instans basis data Anda.

Untuk mengubah nilai parameter ini, Anda perlu membuat grup parameter DB kustom. Anda kemudian mengubah nilai untuk `rds.force_ssl` dalam grup parameter DB kustom Anda menjadi 1 untuk mengaktifkan fitur ini. Jika Anda menyiapkan grup parameter DB kustom sebelum membuat RDS untuk instans DB PostgreSQL, Anda dapat memilihnya (bukan grup parameter default) selama proses pembuatan. Jika Anda melakukan ini setelah instans DB RDS for PostgreSQL Anda sudah berjalan, Anda perlu melakukan reboot instans sehingga instans Anda menggunakan grup parameter kustom. Untuk informasi selengkapnya, lihat Bekerja dengan grup parameter.

Ketika fitur `rds.force_ssl` aktif pada instans DB Anda, upaya koneksi yang tidak menggunakan SSL ditolak dengan pesan berikut:

```
$ psql -h db-name.555555555555.ap-southeast-1.rds.amazonaws.com -p 5432 dbname=testDB
 user=testuser
psql: error: FATAL: no pg_hba.conf entry for host "w.x.y.z", user "testuser", database
 "testDB", SSL off
```

Menentukan status koneksi SSL

Status terenkripsi dari koneksi Anda ditunjukkan pada banner logon saat Anda terhubung ke instans DB:

```
Password for user master:
psql (10.3)
SSL connection (cipher: DHE-RSA-AES256-SHA, bits: 256)
Type "help" for help.
postgres=>
```

Anda juga dapat memuat ekstensi `sslinfo`, lalu memanggil fungsi `ssl_is_used()` untuk menentukan apakah SSL digunakan. Fungsi akan kembali `t` jika koneksi menggunakan SSL, jika tidak akan menghasilkan `f`.

```
postgres=> CREATE EXTENSION sslinfo;
CREATE EXTENSION
postgres=> SELECT ssl_is_used();
```

```
ssl_is_used
---------
t
(1 row)
```

Untuk informasi lebih rinci, Anda dapat menggunakan kueri berikut untuk mendapatkan informasi dari `pg_settings`:

```
SELECT name as "Parameter name", setting as value, short_desc FROM pg_settings WHERE
 name LIKE '%ssl%';
             Parameter name             |                  value                  |
                  short_desc
----------------------------------------+-----------------------------------------
+-------------------------------------------------------
 ssl                                    | on                                      |
 Enables SSL connections.
 ssl_ca_file                            | /rdsdbdata/rds-metadata/ca-cert.pem     |
 Location of the SSL certificate authority file.
 ssl_cert_file                          | /rdsdbdata/rds-metadata/server-cert.pem |
 Location of the SSL server certificate file.
 ssl_ciphers                            | HIGH:!aNULL:!3DES                       |
 Sets the list of allowed SSL ciphers.
 ssl_crl_file                           |                                         |
 Location of the SSL certificate revocation list file.
 ssl_dh_params_file                     |                                         |
 Location of the SSL DH parameters file.
 ssl_ecdh_curve                         | prime256v1                              |
 Sets the curve to use for ECDH.
 ssl_key_file                           | /rdsdbdata/rds-metadata/server-key.pem  |
 Location of the SSL server private key file.
 ssl_library                            | OpenSSL                                 |
 Name of the SSL library.
 ssl_max_protocol_version               |                                         |
 Sets the maximum SSL/TLS protocol version to use.
 ssl_min_protocol_version               | TLSv1.2                                 |
 Sets the minimum SSL/TLS protocol version to use.
 ssl_passphrase_command                 |                                         |
 Command to obtain passphrases for SSL.
 ssl_passphrase_command_supports_reload | off                                     |
 Also use ssl_passphrase_command during server reload.
 ssl_prefer_server_ciphers              | on                                      |
 Give priority to server ciphersuite order.
(14 rows)
```

Anda juga dapat mengumpulkan semua informasi tentang penggunaan SSL instans DB RDS for PostgreSQL Anda berdasarkan proses, klien, dan aplikasi menggunakan kueri berikut:

```
SELECT datname as "Database name", usename as "User name", ssl, client_addr,
 application_name, backend_type
   FROM pg_stat_ssl
   JOIN pg_stat_activity
   ON pg_stat_ssl.pid = pg_stat_activity.pid
   ORDER BY ssl;
 Database name | User name | ssl |  client_addr   |    application_name    |
 backend_type
---------------+-----------+-----+----------------+------------------------
+------------------------------
               |           | f   |                |                        | autovacuum
 launcher
               | rdsadmin  | f   |                |                        | logical
 replication launcher
               |           | f   |                |                        | background
 writer
               |           | f   |                |                        |
 checkpointer
               |           | f   |                |                        | walwriter
 rdsadmin      | rdsadmin  | t   | 127.0.0.1      |                        | client
 backend
 rdsadmin      | rdsadmin  | t   | 127.0.0.1      | PostgreSQL JDBC Driver | client
 backend
 postgres      | postgres  | t   | 204.246.162.36 | psql                   | client
 backend
(8 rows)
```

Untuk mengidentifikasi penyandian yang digunakan untuk koneksi SSL Anda, Anda dapat melakukan kueri sebagai berikut:

```
postgres=> SELECT ssl_cipher();
ssl_cipher
--------------------
DHE-RSA-AES256-SHA
(1 row)
```

Untuk informasi tentang opsi `sslmode`, lihat Database connection control functions dalam dokumentasi PostgreSQL.

Rangkaian penyandian SSL di RDS for PostgreSQL

Parameter konfigurasi PostgreSQL ssl_ciphers menentukan kategori rangkaian penyandian yang diizinkan untuk koneksi SSL. Tabel berikut mencantumkan rangkaian penyandian default yang digunakan dalam RDS for PostgreSQL.

Versi mesin PostgreSQL	Rangkaian penyandian
16	`HIGH:!aNULL:!3DES`
15	`HIGH:!aNULL:!3DES`
14	`HIGH:!aNULL:!3DES`
13	`HIGH:!aNULL:!3DES`
12	`HIGH:!aNULL:!3DES`
11.4 dan versi minor yang lebih tinggi	`HIGH:MEDIUM:+3DES:!aNULL:!RC4`
11.1, 11.2	`HIGH:MEDIUM:+3DES:!aNULL`
10.9 dan versi minor yang lebih tinggi	`HIGH:MEDIUM:+3DES:!aNULL:!RC4`
10.7 dan versi minor yang lebih rendah	`HIGH:MEDIUM:+3DES:!aNULL`

Memperbarui aplikasi untuk terhubung ke instans DB PostgreSQL menggunakan sertifikat SSL/TLS baru

Sertifikat yang digunakan untuk Secure Socket Layer atau Transport Layer Security (SSL/TLS) biasanya memiliki masa pakai yang sudah ditetapkan. Ketika penyedia layanan memperbarui sertifikat Certificate Authority (CA) mereka, klien harus memperbarui aplikasi mereka untuk menggunakan sertifikat baru. Setelah itu, Anda dapat menemukan informasi tentang cara menentukan apakah aplikasi klien Anda menggunakan SSL/TLS untuk terhubung ke instans DB Amazon RDS for PostgreSQL. Anda juga menemukan informasi tentang cara memeriksa apakah aplikasi tersebut memverifikasi sertifikat server saat aplikasi terhubung.

> **Note**
>
> Aplikasi klien yang dikonfigurasi untuk memverifikasi sertifikat server sebelum koneksi SSL/TLS harus memiliki sertifikat CA yang valid di penyimpanan kepercayaan klien. Perbarui penyimpanan kepercayaan klien bila diperlukan untuk sertifikat baru.

Setelah Anda memperbarui sertifikat CA di penyimpanan kepercayaan aplikasi klien, Anda dapat merotasi sertifikat di instans DB Anda. Kami sangat menyarankan Anda menguji prosedur ini di lingkungan non-produksi sebelum menerapkannya di lingkungan produksi Anda.

Untuk informasi selengkapnya tentang rotasi sertifikat, lihat Merotasi sertifikat SSL/TLS. Untuk informasi selengkapnya tentang mengunduh sertifikat, lihat . Untuk informasi tentang menggunakan SSL/TLS dengan instans DB PostgreSQL, lihat Menggunakan SSL dengan instans DB PostgreSQL.

Topik

- Menentukan apakah aplikasi terhubung ke instans DB PostgreSQL menggunakan SSL
- Menentukan apakah klien memerlukan verifikasi sertifikat agar dapat terhubung
- Memperbarui penyimpanan kepercayaan aplikasi Anda
- Menggunakan koneksi SSL/TLS untuk berbagai jenis aplikasi

Menentukan apakah aplikasi terhubung ke instans DB PostgreSQL menggunakan SSL

Periksa konfigurasi instans DB untuk nilai parameter `rds.force_ssl`. Secara default, parameter `rds.force_ssl` diatur ke `0` (tidak aktif) untuk instance DB menggunakan versi PostgreSQL sebelum versi 15. Secara default, `rds.force_ssl` diatur ke `1` (aktif) untuk instans DB menggunakan PostgreSQL versi 15 dan versi utama yang lebih baru. Jika parameter `rds.force_ssl` diatur ke `1` (aktif), klien harus menggunakan SSL/TLS untuk koneksi. Untuk informasi lebih lanjut tentang grup parameter, lihat Bekerja dengan grup parameter.

Jika Anda menggunakan RDS PostgreSQL versi 9.5 atau versi utama yang lebih baru dan `rds.force_ssl` tidak diatur ke `1` (aktif), lakukan kueri tampilan `pg_stat_ssl` untuk memeriksa koneksi menggunakan SSL. Misalnya, kueri berikut hanya mengembalikan koneksi SSL dan informasi tentang klien menggunakan SSL.

```
SELECT datname, usename, ssl, client_addr
```

```
  FROM pg_stat_ssl INNER JOIN pg_stat_activity ON pg_stat_ssl.pid =
 pg_stat_activity.pid
  WHERE ssl is true and usename<>'rdsadmin';
```

Hanya baris yang menggunakan koneksi SSL/TLS yang ditampilkan dengan informasi mengenai koneksi. Berikut ini adalah contoh outputnya.

```
 datname  | usename | ssl | client_addr
----------+---------+-----+-------------
 benchdb  | pgadmin | t   | 53.95.6.13
 postgres | pgadmin | t   | 53.95.6.13
(2 rows)
```

Kueri ini hanya menampilkan koneksi saat ini pada waktu kueri. Tidak adanya hasil tidak menunjukkan bahwa tidak ada aplikasi yang menggunakan koneksi SSL. Koneksi SSL lainnya mungkin terhubung pada waktu yang lain.

Menentukan apakah klien memerlukan verifikasi sertifikat agar dapat terhubung

Ketika klien, seperti psql atau JDBC, dikonfigurasikan dengan dukungan SSL, klien pertama kali mencoba untuk terhubung ke basis data dengan SSL secara default. Jika klien tidak dapat terhubung dengan SSL, maka akan kembali ke koneksi tanpa SSL. Mode `sslmode` default yang digunakan berbeda antara klien berbasis libpq (seperti psql) dan JDBC. Klien berbasis libpq secara default diatur ke `prefer`, di mana klien JDBC secara default diatur ke `verify-full`. Sertifikat di server diverifikasi hanya ketika `sslrootcert` disediakan dengan `sslmode` diatur ke `verify-ca` atau `verify-full`. Kesalahan akan muncul jika sertifikat tidak valid.

Gunakan `PGSSLROOTCERT` untuk memverifikasi sertifikat dengan variabel lingkungan `PGSSLMODE`, dengan `PGSSLMODE` diatur ke `verify-ca` atau `verify-full`.

```
PGSSLMODE=verify-full PGSSLROOTCERT=/fullpath/ssl-cert.pem psql -h
 pgdbidentifier.cxxxxxxxx.us-east-2.rds.amazonaws.com -U masteruser -d postgres
```

Gunakan argumen `sslrootcert` untuk memverifikasi sertifikat dengan `sslmode` dalam menghubungkan format string, dengan `sslmode` diatur ke `verify-ca` atau `verify-full` untuk memverifikasi sertifikat.

```
psql "host=pgdbidentifier.cxxxxxxxx.us-east-2.rds.amazonaws.com sslmode=verify-full
 sslrootcert=/full/path/ssl-cert.pem user=masteruser dbname=postgres"
```

Misalnya, dalam kasus sebelumnya, jika Anda menggunakan sertifikat root yang tidak valid, maka Anda akan melihat kesalahan yang serupa dengan yang berikut pada klien Anda.

```
psql: SSL error: certificate verify failed
```

Memperbarui penyimpanan kepercayaan aplikasi Anda

Untuk informasi tentang pembaruan penyimpanan kepercayaan untuk aplikasi PostgreSQL, lihat Amankan koneksi TCP/IP dengan SSL dalam dokumentasi PostgreSQL.

Untuk informasi tentang mengunduh sertifikat root, lihat .

Untuk contoh skrip yang mengimpor sertifikat, lihat Contoh skrip untuk mengimpor sertifikat ke trust store Anda.

> **Note**
>
> Saat memperbarui penyimpanan kepercayaan, Anda dapat mempertahankan sertifikat lama selain menambahkan sertifikat baru.

Menggunakan koneksi SSL/TLS untuk berbagai jenis aplikasi

Berikut ini informasi tentang menggunakan koneksi SSL/TLS untuk berbagai jenis aplikasi:

- psql

  Klien diinvokasi dari baris perintah dengan menentukan opsi sebagai string koneksi atau sebagai variabel lingkungan. Untuk koneksi SSL/TLS, opsi yang relevan adalah `sslmode` (variabel lingkungan `PGSSLMODE`), `sslrootcert` (variabel lingkungan `PGSSLROOTCERT`).

  Untuk daftar lengkap opsi, lihat Kata kunci parameter dalam dokumentasi PostgreSQL. Untuk daftar lengkap variabel lingkungan, lihat Variabel lingkungan dalam dokumentasi PostgreSQL.
- pgAdmin

  Klien berbasis browser ini adalah antarmuka yang lebih mudah untuk terhubung ke basis data PostgreSQL.

  Untuk informasi tentang konfigurasi koneksi, lihat dokumentasi pgAdmin.
- JDBC

JDBC memungkinkan koneksi basis data dengan aplikasi Java.

Untuk informasi umum tentang koneksi ke basis data PostgreSQL dengan JDBC, lihat Menghubungkan ke basis data dalam dokumentasi driver JDBC PostgreSQL. Untuk informasi tentang koneksi dengan SSL/TLS, lihat Mengonfigurasi klien dalam dokumentasi driver JDBC PostgreSQL.

- Python

  Perpustakaan Python yang populer untuk terhubung ke basis data PostgreSQL adalah `psycopg2`.

  Untuk informasi tentang menggunakan `psycopg2`, lihat Dokumentasi psycopg2. Untuk tutorial pendek tentang cara menghubungkan ke basis data PostgreSQL, lihat Tutorial Psycopg2. Anda dapat menemukan informasi tentang opsi penerimaan perintah koneksi di Konten modul psycopg2.

> **Important**
>
> Setelah Anda menentukan bahwa koneksi database Anda menggunakan SSL/TLS dan telah memperbarui toko kepercayaan aplikasi Anda, Anda dapat memperbarui database Anda untuk menggunakan sertifikat 2048-g1. rds-ca-rsa Untuk petunjuk, lihat langkah 3 dalam Memperbarui sertifikat CA Anda dengan memodifikasi instans atau cluster DB.

Menggunakan autentikasi Kerberos dengan Amazon RDS for PostgreSQL

Anda dapat menggunakan Kerberos untuk mengautentikasi pengguna saat terhubung ke instans DB Anda yang menjalankan PostgreSQL. Untuk melakukannya, konfigurasi instans DB Anda agar menggunakan AWS Directory Service for Microsoft Active Directory untuk autentikasi Kerberos. AWS Directory Service for Microsoft Active Directory juga disebut AWS Managed Microsoft AD. Ini adalah fitur yang tersedia dengan AWS Directory Service. Untuk mempelajari selengkapnya, lihat Apa itu AWS Directory Service? dalam Panduan Administrasi AWS Directory Service.

Untuk memulai, buat direktori AWS Managed Microsoft AD untuk menyimpan kredensial pengguna. Kemudian, berikan domain Active Directory dan informasi lainnya ke instans DB PostgreSQL Anda. Saat pengguna mengautentikasi dengan instans DB PostgreSQL, permintaan autentikasi diteruskan ke direktori AWS Managed Microsoft AD.

Dengan menyimpan semua kredensial Anda di direktori yang sama, Anda dapat menghemat waktu dan tenaga. Anda memiliki sebuah lokasi terpusat untuk menyimpan dan mengelola kredensial bagi beberapa instans DB. Penggunaan direktori juga dapat meningkatkan profil keamanan Anda secara keseluruhan.

Selain itu, Anda dapat mengakses kredensial dari Microsoft Active Directory on-premise Anda sendiri. Untuk melakukannya, buat hubungan domain tepercaya sehingga direktori AWS Managed Microsoft AD mempercayai Microsoft Active Directory on-premise Anda. Dengan cara ini, pengguna Anda dapat mengakses instans PostgreSQL Anda dengan pengalaman masuk tunggal (SSO) Windows yang sama seperti ketika mereka mengakses beban kerja di jaringan on-premise Anda.

Basis data dapat menggunakan autentikasi kata sandi dengan autentikasi Kerberos atau AWS Identity and Access Management (IAM). Untuk informasi selengkapnya tentang autentikasi IAM, lihat Autentikasi basis data IAM untuk MariaDB, MySQL, dan PostgreSQL.

Topik

Ketersediaan Wilayah dan versi

Ketersediaan dan dukungan fitur bervariasi di seluruh versi khusus dari setiap mesin basis data, dan di seluruh Wilayah AWS. Lihat informasi selengkapnya tentang Ketersediaan wilayah dan versi RDS for PostgreSQL dengan autentikasi Kerberos di Autentikasi Kerberos.

Ikhtisar autentikasi Kerberos untuk instans DB PostgreSQL

Untuk menyiapkan autentikasi Kerberos untuk instans DB PostgreSQL, lakukan langkah-langkah berikut, yang dijelaskan secara lebih mendetail nanti:

1. Gunakan AWS Managed Microsoft AD untuk membuat direktori AWS Managed Microsoft AD. Anda dapat menggunakan AWS Management Console, AWS CLI, atau AWS Directory Service API untuk membuat direktori. Pastikan untuk membuka port keluar yang relevan pada grup keamanan direktori sehingga direktori dapat berkomunikasi dengan instans.
2. Buat peran yang memberikan akses ke Amazon RDS untuk melakukan panggilan ke direktori AWS Managed Microsoft AD Anda. Untuk melakukannya, buat peran (IAM) AWS Identity and Access Management yang menggunakan kebijakan IAM terkelola `AmazonRDSDirectoryServiceAccess`.

   Agar peran IAM mengizinkan akses, titik akhir AWS Security Token Service (AWS STS) harus diaktifkan di Wilayah AWS yang tepat untuk akun AWS Anda. Titik akhir AWS STS aktif secara default di semua Wilayah AWS, dan Anda dapat menggunakannya tanpa tindakan lebih lanjut. Lihat informasi selengkapnya di Mengaktifkan dan menonaktifkan AWS STS di Wilayah AWS dalam Panduan Pengguna IAM.
3. Buat dan konfigurasikan pengguna dalam direktori AWS Managed Microsoft AD menggunakan alat Microsoft Active Directory. Untuk mengetahui informasi selengkapnya tentang cara membuat pengguna di Active Directory, lihat Mengelola pengguna dan grup di Microsoft AD Terkelola AWS dalam Panduan Administrasi AWS Directory Service.
4. Jika Anda ingin menemukan direktori dan instans DB dalam akun AWS atau cloud privat virtual (VPC) yang berbeda, konfigurasikan peering VPC. Untuk informasi selengkapnya, lihat Apa yang dimaksud peering VPC? di Panduan Peering Amazon VPC.
5. Buat atau modifikasi instans DB PostgreSQL dari konsol, CLI, atau RDS API menggunakan salah satu metode berikut:
   - Membuat instans DB Amazon RDS
   - Memodifikasi instans DB Amazon RDS

- Memulihkan dari snapshot DB
- Memulihkan instans DB dengan waktu yang ditentukan

Anda dapat menemukan instans di Cloud Privat Virtual (VPC) Amazon yang sama dengan direktori atau di VPC atau akun AWS yang berbeda. Saat membuat atau memodifikasi instans DB PostgreSQL, lakukan hal berikut:

- Sediakan pengidentifikasi domain (pengidentifikasi `d-*`) yang dihasilkan saat Anda membuat direktori.
- Beri nama peran IAM yang Anda buat.
- Pastikan bahwa grup keamanan instans DB dapat menerima lalu lintas masuk dari grup keamanan direktori.

6. Gunakan kredensial pengguna utama RDS untuk terhubung ke instans DB PostgreSQL. Buat pengguna dalam PostgreSQL untuk diidentifikasi secara eksternal. Pengguna yang diidentifikasi secara eksternal dapat masuk ke instans DB PostgreSQL menggunakan autentikasi Kerberos.

Menyiapkan autentikasi Kerberos untuk instans DB PostgreSQL

Untuk menyiapkan autentikasi Kerberos, lakukan langkah berikut.

Topik

- Langkah 1: Buat direktori menggunakan AWS Managed Microsoft AD
- Langkah 2: (Opsional) Buat hubungan kepercayaan antara Active Directory lokal dan AWS Directory Service
- Langkah 3: Buat peran IAM untuk RDS untuk mengakses AWS Directory Service
- Langkah 4: Buat dan konfigurasikan pengguna
- Langkah 5: Aktifkan lalu lintas antar-VPC antara direktori dan instans DB
- Langkah 6: Buat atau modifikasi instans DB PostgreSQL
- Langkah 7: Buat pengguna PostgreSQL untuk pengguna utama Kerberos Anda
- Langkah 8: Konfigurasikan klien PostgreSQL

Langkah 1: Buat direktori menggunakan AWS Managed Microsoft AD

AWS Directory Service membuat Direktori Aktif yang dikelola sepenuhnya di AWS Cloud. Saat Anda membuat AWS Managed Microsoft AD direktori, AWS Directory Service buat dua pengontrol

domain dan server DNS untuk Anda. Server-server direktori dibuat di subnet yang berbeda di VPC. Redundansi ini membantu memastikan bahwa direktori Anda tetap dapat diakses meskipun terjadi kegagalan.

Saat Anda membuat AWS Managed Microsoft AD AWS direktori, Directory Service melakukan tugas-tugas berikut atas nama Anda:

- Menyiapkan Active Directory di dalam VPC Anda.
- Membuat akun administrator direktori dengan nama pengguna `Admin` dan kata sandi yang ditentukan. Anda menggunakan akun ini untuk mengelola direktori.

  > **Important**
  >
  > Pastikan untuk menyimpan kata sandi ini. AWS Directory Service tidak menyimpan kata sandi ini, dan tidak dapat diambil atau diatur ulang.

- Membuat grup keamanan untuk pengendali direktori. Grup keamanan harus mengizinkan komunikasi dengan instans DB PostgreSQL.

Saat Anda meluncurkan AWS Directory Service for Microsoft Active Directory, AWS buat Unit Organisasi (OU) yang berisi semua objek direktori Anda. OU ini memiliki nama NetBIOS yang Anda masukkan saat membuat direktori, dan terletak di root domain. Root domain dimiliki dan dikelola oleh AWS.

`Admin`Akun yang dibuat dengan AWS Managed Microsoft AD direktori Anda memiliki izin untuk kegiatan administratif yang paling umum untuk OU Anda:

- Membuat, memperbarui, atau menghapus pengguna
- Menambahkan sumber daya ke domain Anda seperti server file atau cetak, lalu menetapkan izin untuk sumber daya tersebut kepada pengguna di OU Anda
- Membuat OU dan kontainer tambahan
- Melimpahkan kewenangan
- Memulihkan objek-objek yang dihapus dari Keranjang Sampah Active Directory
- Jalankan modul Active Directory dan Domain Name Service (DNS) untuk Windows PowerShell pada Layanan Web Direktori Aktif

Akun `Admin` juga memiliki hak untuk melakukan aktivitas di seluruh domain berikut:

- Mengelola konfigurasi DNS (menambahkan, menghapus, atau memperbarui catatan, zona, dan penerus)
- Melihat log peristiwa DNS
- Melihat log peristiwa keamanan

Untuk membuat direktori dengan AWS Managed Microsoft AD

1. Di panel navigasi konsol AWS Directory Service, pilih Direktori, lalu pilih Siapkan direktori.
2. Pilih AWS Managed Microsoft AD. AWS Managed Microsoft AD adalah satu-satunya opsi yang saat ini didukung untuk digunakan dengan Amazon RDS.
3. Pilih Berikutnya.
4. Di halaman Masukkan informasi direktori, berikan informasi berikut:

   Edisi

   Pilih edisi sesuai kebutuhan Anda.

   Nama DNS direktori

   Nama berkualifikasi penuh untuk direktori, seperti **`corp.example.com`**.

   Nama NetBIOS direktori

   Nama pendek opsional untuk direktori, seperti `CORP`.

   Deskripsi direktori

   Deskripsi opsional untuk direktori.

   Kata sandi admin

   Kata sandi administrator direktori. Proses pembuatan direktori menciptakan akun administrator dengan nama pengguna `Admin` dan kata sandi ini.

   Kata sandi administrator direktori tidak boleh menyertakan kata "admin". Kata sandi peka terhadap huruf besar/kecil dan harus terdiri dari 8–64 karakter. Kata sandi juga harus berisi setidaknya satu karakter dari tiga di antara empat kategori berikut:

   - Huruf kecil (a-z)
   - Huruf besar (A-Z)
   - Angka (0-9)

- Karakter non-alfanumerik (~!@#$%^&*_-+=`|\(){}[]:;"'<>,.?/)

Konfirmasi kata sandi

Ketik ulang kata sandi administrator.

> **Important**
>
> Pastikan Anda menyimpan kata sandi ini. AWS Directory Service tidak menyimpan kata sandi ini, dan tidak dapat diambil atau diatur ulang.

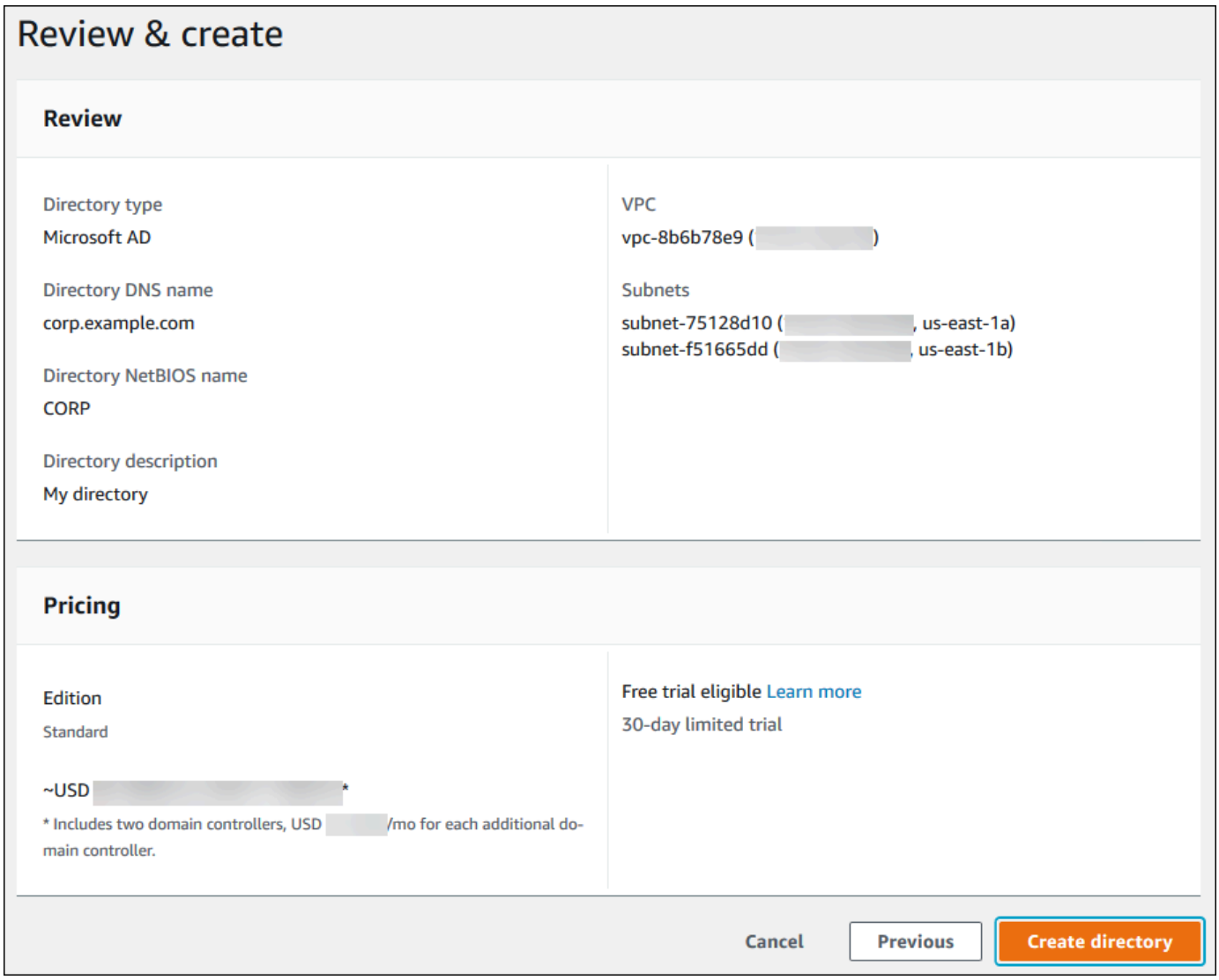
5. Pilih Berikutnya.
6. Di halaman Pilih VPC dan subnet, berikan informasi berikut:

   VPC

   Pilih VPC untuk direktori. Anda dapat membuat instans DB PostgreSQL dalam VPC yang sama ini atau dalam VPC yang berbeda.

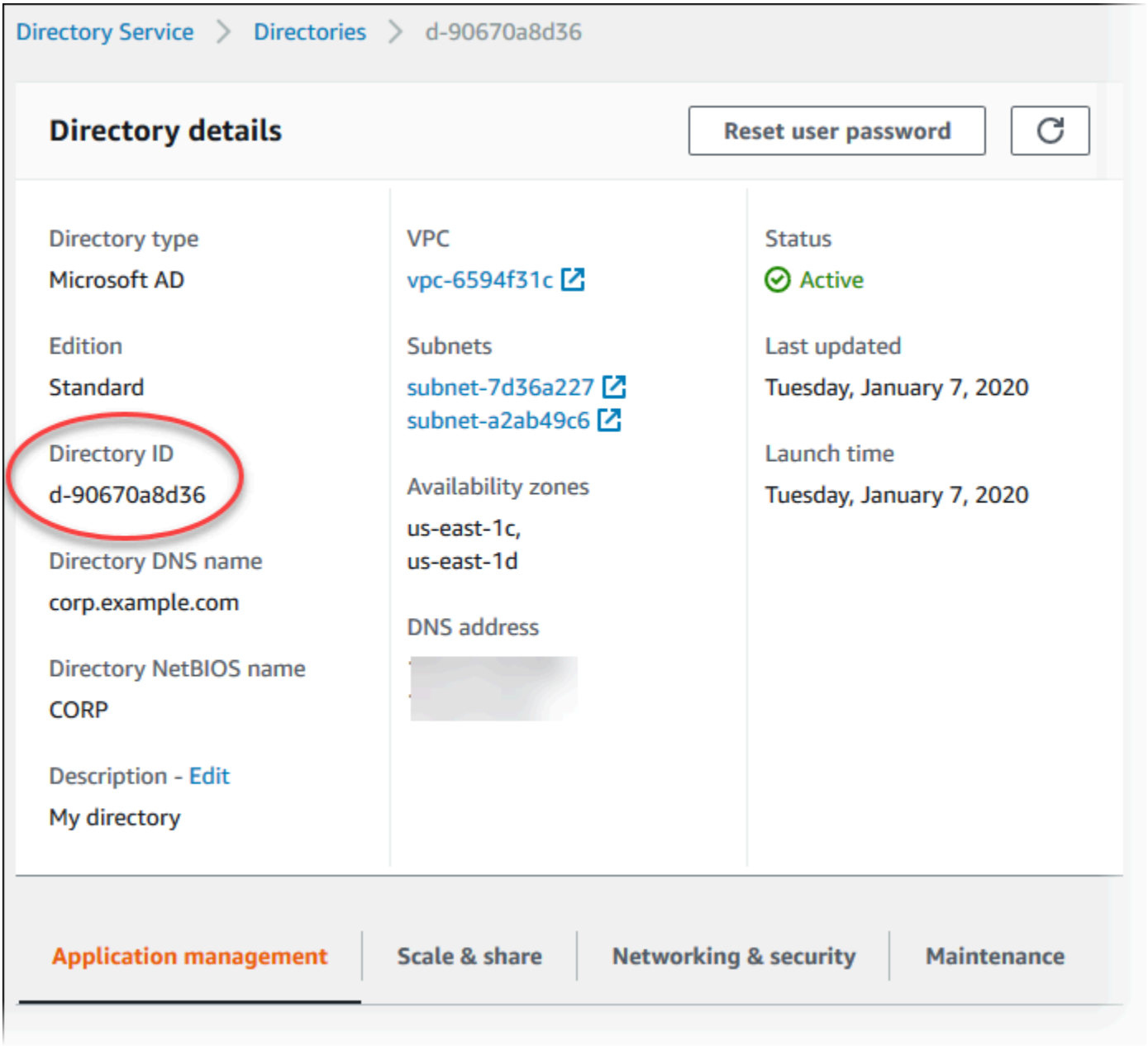
   Subnet

   Pilih subnet untuk server direktori. Kedua subnet harus berada di Zona Ketersediaan yang berbeda.
7. Pilih Berikutnya.
8. Tinjau informasi direktori. Jika ada yang perlu diubah, pilih Sebelumnya dan lakukan perubahan. Jika informasi sudah benar, pilih Buat direktori.



Pembuatan direktori memerlukan waktu beberapa menit. Setelah berhasil dibuat, nilai Status berubah menjadi Aktif.

Untuk melihat informasi tentang direktori Anda, pilih ID direktori di daftar direktori. Buat catatan tentang nilai ID Direktori. Anda memerlukan nilai ini saat membuat atau mengubah instans DB PostgreSQL.



Langkah 2: (Opsional) Buat hubungan kepercayaan antara Active Directory lokal dan AWS Directory Service

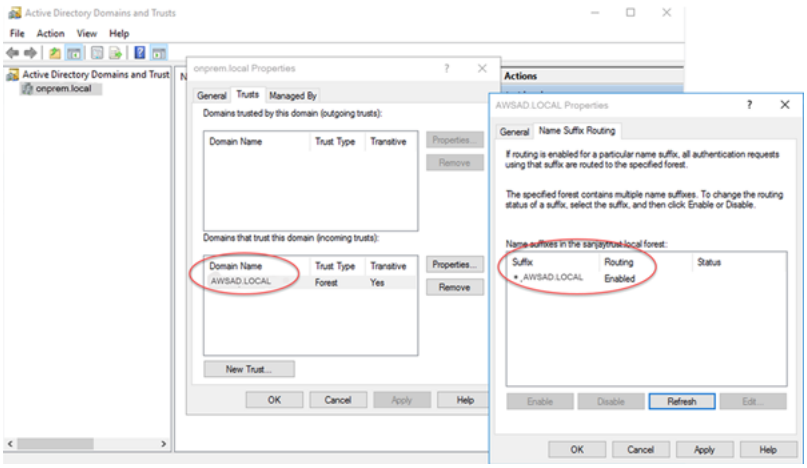
Jika Anda tidak berencana untuk menggunakan Microsoft Active Directory on-premise Anda sendiri, langsung ke Langkah 3: Buat peran IAM untuk RDS untuk mengakses AWS Directory Service.

Untuk mendapatkan autentikasi Kerberos menggunakan Active Directory lokal, Anda perlu membuat relasi domain kepercayaan menggunakan trust hutan antara Microsoft Active Directory lokal dan direktori (dibuat di AWS Managed Microsoft AD). Langkah 1: Buat direktori menggunakan AWS Managed Microsoft AD Kepercayaan bisa satu arah, di mana AWS Managed Microsoft AD direktori

mempercayai Microsoft Active Directory lokal. Kepercayaan juga dapat bersifat dua arah, di mana kedua Active Directory saling mempercayai. Untuk informasi selengkapnya tentang menyiapkan trust menggunakan AWS Directory Service, lihat Kapan membuat hubungan kepercayaan di Panduan AWS Directory Service Administrasi.

> **Note**
>
> Jika Anda menggunakan Microsoft Active Directory lokal, klien Windows akan terhubung menggunakan nama domain di titik akhir, bukan AWS Directory Service rds.amazonaws.com. Untuk mempelajari selengkapnya, lihat Menghubungkan ke PostgreSQL dengan autentikasi Kerberos.

Pastikan bahwa nama domain Microsoft Active Directory on-premise Anda mencakup perutean akhiran DNS yang sesuai dengan hubungan kepercayaan yang baru dibuat. Tangkapan layar berikut menunjukkan sebuah contoh.



Langkah 3: Buat peran IAM untuk RDS untuk mengakses AWS Directory Service

Agar RDS memanggil AWS Directory Service Anda, AWS akun Anda memerlukan peran IAM yang menggunakan kebijakan IAM terkelola. `AmazonRDSDirectoryServiceAccess` Peran ini membuat Amazon RDS dapat melakukan panggilan ke AWS Directory Service.

Saat Anda membuat instans DB menggunakan AWS Management Console dan akun pengguna konsol Anda memiliki `iam:CreateRole` izin, konsol akan membuat peran IAM yang diperlukan secara otomatis. Dalam hal ini, nama perannya adalah `rds-directoryservice-kerberos-access-role`. Jika tidak, Anda harus membuat peran IAM secara manual. Saat Anda

membuat peran IAM ini, pilih`Directory Service`, dan lampirkan kebijakan AWS terkelola `AmazonRDSDirectoryServiceAccess` padanya.

Untuk informasi selengkapnya tentang membuat peran IAM untuk layanan, lihat Membuat peran untuk mendelegasikan izin ke AWS layanan di Panduan Pengguna IAM.

> **Note**
>
> Peran IAM yang digunakan untuk Autentikasi Windows untuk RDS for Microsoft SQL Server tidak dapat digunakan untuk Amazon RDS for PostgreSQL.

Sebagai alternatif untuk menggunakan kebijakan terkelola `AmazonRDSDirectoryServiceAccess`, Anda dapat membuat kebijakan dengan izin yang diperlukan. Dalam hal ini, peran IAM harus memiliki kebijakan kepercayaan IAM berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "directoryservice.rds.amazonaws.com",
          "rds.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Peran ini juga harus memiliki kebijakan peran IAM berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ds:DescribeDirectories",
        "ds:AuthorizeApplication",
```

```
        "ds:UnauthorizeApplication",
        "ds:GetAuthorizedApplicationDetails"
      ],
    "Effect": "Allow",
    "Resource": "*"
    }
  ]
}
```

Langkah 4: Buat dan konfigurasikan pengguna

Anda dapat membuat pengguna dengan alat Active Directory Users and Computers. Alat ini merupakan salah satu alat Active Directory Domain Services dan Active Directory Lightweight Directory Services. Untuk informasi selengkapnya, lihat Add Users and Computers to the Active Directory domain dalam dokumentasi Microsoft. Dalam hal ini, pengguna adalah individu atau entitas lain, seperti komputer mereka yang merupakan bagian dari domain dan yang identitasnya dipertahankan dalam direktori.

Untuk membuat pengguna di AWS Directory Service direktori, Anda harus terhubung ke instans Amazon EC2 berbasis Windows yang merupakan anggota direktori. AWS Directory Service Pada saat yang sama, Anda harus masuk sebagai pengguna yang memiliki hak untuk membuat pengguna. Untuk informasi selengkapnya, lihat Membuat pengguna dalam Panduan Administrasi AWS Directory Service .

Langkah 5: Aktifkan lalu lintas antar-VPC antara direktori dan instans DB

Jika Anda ingin menemukan direktori dan instans DB dalam VPC yang sama, lewati langkah ini dan lanjutkan ke Langkah 6: Buat atau modifikasi instans DB PostgreSQL.

Jika Anda ingin menemukan direktori dan instans DB di VPC yang berbeda, konfigurasikan lalu lintas antar-VPC menggunakan peering VPC atau AWS Transit Gateway.

Prosedur berikut mengaktifkan lalu lintas antar-VPC menggunakan peering VPC. Ikuti petunjuk di Apa yang dimaksud dengan peering VPC? dalam Panduan Peering Amazon Virtual Private Cloud.

Untuk mengaktifkan lalu lintas VPC menggunakan peering VPC

1. Siapkan aturan perutean VPC yang sesuai untuk memastikan lalu lintas jaringan dapat berjalan dua arah.
2. Pastikan bahwa grup keamanan instans DB dapat menerima lalu lintas masuk dari grup keamanan direktori.

3. Pastikan tidak ada aturan daftar kontrol akses (ACL) jaringan yang memblokir lalu lintas.

Jika AWS akun lain memiliki direktori, Anda harus berbagi direktori.

Untuk berbagi direktori antar AWS akun

1. Mulai berbagi direktori dengan AWS akun tempat instans DB akan dibuat dengan mengikuti petunjuk di Tutorial: Berbagi direktori AD Microsoft AWS Terkelola Anda untuk Domain EC2 yang mulus-Bergabung dalam Panduan Administrasi.AWS Directory Service
2. Masuk ke AWS Directory Service konsol menggunakan akun untuk instans DB, dan pastikan domain memiliki `SHARED` status sebelum melanjutkan.
3. Saat masuk ke AWS Directory Service konsol menggunakan akun untuk instans DB, perhatikan nilai ID Direktori. Anda menggunakan ID direktori ini untuk menggabungkan instans DB ke domain.

Langkah 6: Buat atau modifikasi instans DB PostgreSQL

Buat atau modifikasi instans DB PostgreSQL untuk digunakan dengan direktori Anda. Anda dapat menggunakan konsol, CLI, atau RDS API untuk mengaitkan instans DB dengan direktori. Anda dapat melakukannya dengan salah satu cara berikut:

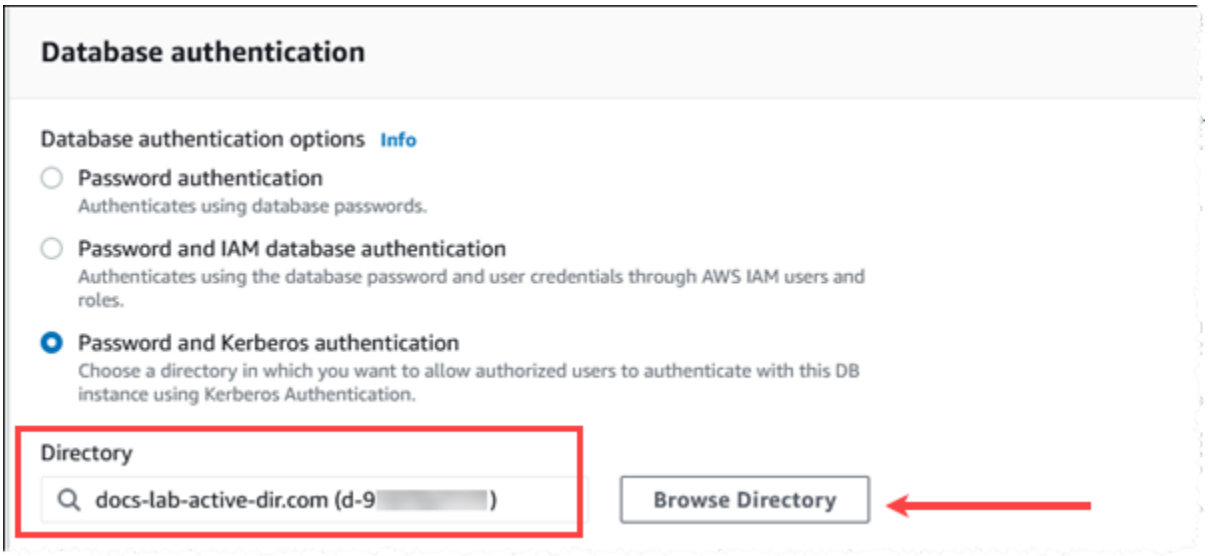
- Buat instance PostgreSQL DB baru menggunakan konsol, perintah create-db-instanceCLI, atau operasi CreateDBInstance RDS API. Untuk petunjuk, lihat Membuat instans DB Amazon RDS.
- Ubah instance PostgreSQL DB yang ada menggunakan konsol, perintah modify-db-instanceCLI, atau operasi ModifyDBInstance RDS API. Untuk petunjuk, lihat Memodifikasi instans DB Amazon RDS.
- Pulihkan instance PostgreSQL DB dari snapshot DB menggunakan konsol, perintah CLI restore-db-instance-from-db-snapshot, atau operasi RestoreDB dbSnapshot RDS API. InstanceFrom Untuk petunjuk, lihat Memulihkan dari snapshot DB.
- Kembalikan instance PostgreSQL DB ke point-in-time menggunakan konsol, perintah restore-db-instance-to- point-in-time CLI, atau operasi RestoreDB RDS API. InstanceToPointInTime Untuk petunjuk, lihat Memulihkan instans DB dengan waktu yang ditentukan.

Autentikasi Kerberos hanya didukung untuk instans DB PostgreSQL dalam sebuah VPC. Instans DB boleh berada dalam VPC yang sama dengan direktori, atau dalam VPC yang berbeda. Instans

DB harus menggunakan grup keamanan yang memungkinkan ingress and egress di dalam VPC direktori, sehingga instans DB dapat berkomunikasi dengan direktori.

Konsol

Saat Anda menggunakan konsol untuk membuat, memodifikasi, atau memulihkan instans DB, pilih Kata sandi dan autentikasi Kerberos di bagian Autentikasi basis data. Kemudian pilih Cari Direktori. Pilih direktori atau pilih Buat direktori baru untuk menggunakan Directory Service.



AWS CLI

Saat Anda menggunakan AWS CLI, parameter berikut diperlukan untuk instance DB agar dapat menggunakan direktori yang Anda buat:

- Untuk parameter `--domain`, gunakan pengidentifikasi domain (pengidentifikasi "d-*") yang dihasilkan saat Anda membuat direktori.
- Untuk parameter `--domain-iam-role-name`, gunakan peran yang Anda buat yang menggunakan kebijakan IAM terkelola `AmazonRDSDirectoryServiceAccess`.

Misalnya, perintah CLI berikut memodifikasi instans DB untuk menggunakan direktori.

```
aws rds modify-db-instance --db-instance-identifier mydbinstance --domain d-Directory-ID --domain-iam-role-name role-name
```

> **Important**
>
> Jika Anda memodifikasi instans DB untuk mengaktifkan autentikasi Kerberos, boot ulang instans DB setelah membuat perubahan.

Langkah 7: Buat pengguna PostgreSQL untuk pengguna utama Kerberos Anda

Pada titik ini, instans DB RDS for PostgreSQL Anda digabungkan ke domain AWS Managed Microsoft AD . Pengguna yang Anda buat di direktori Langkah 4: Buat dan konfigurasikan pengguna perlu diatur sebagai pengguna basis data PostgreSQL dan diberi hak istimewa untuk masuk ke basis data. Anda dapat melakukannya dengan masuk sebagai pengguna basis data dengan hak istimewa `rds_superuser`. Misalnya, jika Anda menerima default saat membuat instans DB RDS for PostgreSQL, gunakan `postgres`, seperti yang ditunjukkan dalam langkah berikut.

Untuk membuat pengguna basis data PostgreSQL untuk pengguna utama Kerberos

1. Gunakan `psql` untuk menghubungkan ke titik akhir instans DB RDS for PostgreSQL menggunakan `psql`. Contoh berikut menggunakan akun `postgres` default untuk peran `rds_superuser`.

   ```
   psql --host=cluster-instance-1.111122223333.aws-region.rds.amazonaws.com --
   port=5432 --username=postgres --password
   ```

2. Buat nama pengguna basis data untuk setiap pengguna utama Kerberos (nama pengguna Active Directory) yang ingin Anda beri akses ke basis data. Gunakan nama pengguna (identitas) kanonik seperti yang didefinisikan dalam instans Active Directory, yaitu `alias` dalam huruf kecil (nama pengguna di Active Directory) dan nama domain Active Directory dalam huruf besar untuk nama pengguna tersebut. Nama pengguna Active Directory adalah pengguna yang diautentikasi secara eksternal, jadi gunakan tanda kutip pada nama ini seperti yang ditunjukkan berikut.

   ```
   postgres=> CREATE USER "username@CORP.EXAMPLE.COM" WITH LOGIN;
   CREATE ROLE
   ```

3. Beri peran `rds_ad` kepada pengguna basis data.

   ```
   postgres=> GRANT rds_ad TO "username@CORP.EXAMPLE.COM";
   GRANT ROLE
   ```

Setelah Anda selesai membuat semua pengguna PostgreSQL untuk identitas pengguna Active Directory Anda, pengguna dapat mengakses instans DB RDS for PostgreSQL menggunakan kredensial Kerberos mereka.

Diasumsikan bahwa pengguna basis data yang melakukan autentikasi menggunakan Kerberos melakukannya dari mesin klien yang merupakan anggota domain Active Directory.

Pengguna basis data yang telah diberi peran `rds_ad` tidak dapat memiliki peran `rds_iam`. Aturan ini juga berlaku untuk keanggotaan bertingkat. Untuk informasi selengkapnya, lihat Autentikasi basis data IAM untuk MariaDB, MySQL, dan PostgreSQL.

Langkah 8: Konfigurasikan klien PostgreSQL

Untuk mengonfigurasi klien PostgreSQL, lakukan langkah berikut:

- Buat file krb5.conf (atau yang setara) untuk menunjuk ke domain.
- Verifikasi bahwa lalu lintas dapat mengalir antara host klien dan AWS Directory Service. Gunakan utilitas jaringan seperti Netcat untuk hal berikut:
  - Memeriksa lalu lintas melalui DNS untuk port 53.
  - Periksa lalu lintas atas TCP/UDP untuk port 53 dan untuk Kerberos, yang mencakup port 88 dan 464 untuk AWS Directory Service.
- Memastikan lalu lintas dapat mengalir di antara host klien dan instans DB melalui port basis data. Misalnya, gunakan psql untuk menghubungkan dan mengakses basis data.

Berikut ini adalah contoh konten krb5.conf untuk. AWS Managed Microsoft AD

```
[libdefaults]
 default_realm = EXAMPLE.COM
[realms]
 EXAMPLE.COM = {
  kdc = example.com
  admin_server = example.com
 }
[domain_realm]
 .example.com = EXAMPLE.COM
 example.com = EXAMPLE.COM
```

Berikut adalah contoh konten krb5.conf untuk Microsoft Active Directory on-premise.

```
[libdefaults]
 default_realm = EXAMPLE.COM
[realms]
 EXAMPLE.COM = {
  kdc = example.com
  admin_server = example.com
 }
 ONPREM.COM = {
  kdc = onprem.com
  admin_server = onprem.com
 }
[domain_realm]
 .example.com = EXAMPLE.COM
 example.com = EXAMPLE.COM
 .onprem.com = ONPREM.COM
 onprem.com = ONPREM.COM
 .rds.amazonaws.com = EXAMPLE.COM
 .amazonaws.com.cn = EXAMPLE.COM
 .amazon.com = EXAMPLE.COM
```

Mengelola instans DB di Domain

Anda dapat menggunakan konsol, CLI, atau API RDS untuk mengelola instans DB dan hubungannya dengan Microsoft Active Directory. Misalnya, Anda dapat mengaitkan Active Directory untuk mengaktifkan autentikasi Kerberos. Anda juga dapat menghapus pengaitan Active Directory untuk menonaktifkan autentikasi Kerberos. Anda juga dapat memindahkan instans DB untuk diautentikasi secara eksternal oleh satu Microsoft Active Directory ke yang lain.

Misalnya, dengan CLI, Anda dapat melakukan hal berikut:

- Untuk mencoba kembali mengaktifkan autentikasi Kerberos untuk keanggotaan yang gagal, gunakan perintah CLI modify-db-instance. Tentukan ID direktori keanggotaan saat ini untuk opsi `--domain`.
- Untuk menonaktifkan autentikasi Kerberos pada instans DB, gunakan perintah CLI modify-db-instance . Tentukan `none` untuk opsi `--domain`.
- Untuk memindahkan instans DB dari satu domain ke domain lain, gunakan perintah CLI modify-db-instance . Tentukan pengidentifikasi domain dari domain baru untuk opsi `--domain`.

Memahami keanggotaan Domain

Setelah Anda membuat atau memodifikasi instans DB, klaster DB menjadi anggota domain. Anda dapat melihat status keanggotaan domain di konsol atau dengan menjalankan perintah CLI describe-db-instances. Status instans DB dapat berupa salah satu dari daftar berikut:

- `kerberos-enabled` – Instans DB mengaktifkan autentikasi Kerberos.
- `enabling-kerberos` – AWS sedang mengaktifkan autentikasi Kerberos pada instans DB ini.
- `pending-enable-kerberos` – Aktivasi autentikasi Kerberos pada instans DB ini tertunda.
- `pending-maintenance-enable-kerberos` – AWS akan mencoba mengaktifkan autentikasi Kerberos pada instans DB pada jendela pemeliharaan terjadwal berikutnya.
- `pending-disable-kerberos` – Penonaktifan autentikasi Kerberos pada instans DB ini tertunda.
- `pending-maintenance-disable-kerberos` – AWS akan mencoba menonaktifkan autentikasi Kerberos pada instans DB pada jendela pemeliharaan terjadwal berikutnya.
- `enable-kerberos-failed` – Masalah konfigurasi membuat AWS tidak dapat mengaktifkan autentikasi Kerberos pada instans DB. Perbaiki masalah konfigurasi tersebut sebelum menerbitkan ulang perintah untuk memodifikasi instans DB.
- `disabling-kerberos` – AWS sedang menonaktifkan autentikasi Kerberos pada instans DB ini.

Permintaan untuk mengaktifkan autentikasi Kerberos dapat gagal karena masalah koneksi jaringan atau kesalahan peran IAM. Dalam beberapa kasus, upaya untuk mengaktifkan autentikasi Kerberos mungkin gagal saat Anda membuat atau memodifikasi instans DB. Jika demikian, pastikan Anda menggunakan peran IAM yang benar, kemudian ubah instans DB untuk bergabung ke domain.

> **Note**
>
> Hanya autentikasi Kerberos dengan RDS for PostgreSQL yang mengirimkan lalu lintas ke server DNS domain. Semua permintaan DNS lainnya diperlakukan sebagai akses jaringan keluar pada instans DB Anda yang menjalankan PostgreSQL. Untuk informasi selengkapnya tentang akses jaringan keluar dengan RDS for PostgreSQL, lihat Menggunakan server DNS khusus untuk akses jaringan keluar.

Menghubungkan ke PostgreSQL dengan autentikasi Kerberos

Anda dapat terhubung ke PostgreSQL dengan autentikasi Kerberos, antarmuka pgAdmin, atau antarmuka baris perintah seperti psql. Untuk informasi selengkapnya tentang cara menghubungkan, lihat Menghubungkan ke instans DB yang menjalankan mesin basis data PostgreSQL . Untuk informasi tentang mendapatkan titik akhir, nomor port, dan detail lain yang diperlukan untuk koneksi, lihat Langkah 3: Hubungkan ke instans DB PostgreSQL.

pgAdmin

Untuk menggunakan pgAdmin untuk terhubung ke PostgreSQL dengan autentikasi Kerberos, lakukan langkah berikut:

1. Luncurkan aplikasi pgAdmin di komputer klien Anda.
2. Pada tab Dasbor, pilih Tambahkan Server Baru.
3. Di kotak dialog Buat - Server, masukkan nama pada tab Umum untuk mengidentifikasi server di pgAdmin.
4. Pada tab Koneksi, masukkan informasi berikut dari basis data RDS for PostgreSQL Anda.
   - Untuk Host, masukkan titik akhir untuk . Instans DB RDS for PostgreSQL. Titik akhir terlihat seperti berikut ini:

     ```
     RDS-DB-instance.111122223333.aws-region.rds.amazonaws.com
     ```

     Untuk terhubung ke Microsoft Active Directory on-premise dari klien Windows, gunakan nama domain AWS Managed Active Directory, bukan `rds.amazonaws.com`, di titik akhir host. Misalnya, anggaplah nama domain untuk AWS Managed Active Directory adalah `corp.example.com`. Kemudian untuk Host, titik akhir akan ditentukan sebagai berikut:

     ```
     RDS-DB-instance.111122223333.aws-region.corp.example.com
     ```

   - Untuk Port, masukkan port yang ditetapkan.
   - Untuk Basis data pemeliharaan, masukkan nama basis data awal yang akan dihubungkan ke klien.
   - Untuk Nama pengguna, masukkan nama pengguna yang Anda masukkan untuk autentikasi Kerberos di Langkah 7: Buat pengguna PostgreSQL untuk pengguna utama Kerberos Anda .
5. Pilih Simpan.

Psql

Untuk menggunakan psql untuk terhubung ke PostgreSQL dengan autentikasi Kerberos, lakukan langkah berikut:

1. Pada jendela perintah, jalankan perintah berikut.

   ```
   kinit username
   ```

   Ganti *username* dengan nama pengguna. Saat diminta, masukkan kata sandi yang disimpan dalam Microsoft Active Directory untuk pengguna.

2. Jika instans DB PostgreSQL menggunakan VPC yang dapat diakses publik, masukkan alamat IP untuk titik akhir instans DB di file `/etc/hosts` Anda pada klien EC2. Misalnya, perintah berikut mendapatkan alamat IP lalu memasukkannya ke dalam file `/etc/hosts`.

   ```
   % dig +short PostgreSQL-endpoint.AWS-Region.rds.amazonaws.com
   ;; Truncated, retrying in TCP mode.
   ec2-34-210-197-118.AWS-Region.compute.amazonaws.com.
   34.210.197.118

   % echo " 34.210.197.118  PostgreSQL-endpoint.AWS-Region.rds.amazonaws.com" >> /etc/
   hosts
   ```

   Jika Anda menggunakan Microsoft Active Directory on-premise dari klien Windows, Anda perlu terhubung menggunakan titik akhir khusus. Alih-alih menggunakan domain Amazon `rds.amazonaws.com` di titik akhir host, gunakan nama domain AWS Managed Active Directory.

   Misalnya, anggaplah nama domain untuk AWS Managed Active Directory Anda adalah `corp.example.com`. Kemudian gunakan format *PostgreSQL-endpoint.AWS-Region.corp.example.com* untuk titik akhir dan masukkan ke dalam file `/etc/hosts`.

   ```
   % echo " 34.210.197.118  PostgreSQL-endpoint.AWS-Region.corp.example.com" >> /etc/
   hosts
   ```

3. Gunakan perintah psql berikut untuk masuk ke instans DB PostgreSQL yang terintegrasi dengan Active Directory.

   ```
   psql -U username@CORP.EXAMPLE.COM -p 5432 -h PostgreSQL-endpoint.AWS-
   Region.rds.amazonaws.com postgres
   ```

Untuk masuk ke klaster DB PostgreSQL dari klien Windows menggunakan Active Directory on-premise, gunakan perintah psql berikut dengan nama domain dari langkah sebelumnya (`corp.example.com`):

```
psql -U username@CORP.EXAMPLE.COM -p 5432 -h PostgreSQL-endpoint.AWS-
Region.corp.example.com postgres
```

Menggunakan server DNS khusus untuk akses jaringan keluar

Amazon RDS for PostgreSQL mendukung akses jaringan keluar di instans DB Anda dan mengizinkan resolusi Layanan Nama Domain (DNS) dari server DNS khusus yang dimiliki pelanggan. Anda hanya dapat menyelesaikan nama domain yang benar-benar memenuhi syarat dari instans DB RDS for PostgreSQL melalui server DNS khusus.

Topik

Mengaktifkan resolusi DNS khusus

Untuk mengaktifkan resolusi DNS di VPC pelanggan Anda, pertama-tama kaitkan grup parameter DB khusus ke instans RDS for PostgreSQL. Kemudian nyalakan parameter `rds.custom_dns_resolution` dengan mengaturnya ke 1, lalu mulai ulang instans DB agar perubahan terjadi.

Menonaktifkan resolusi DNS khusus

Untuk menonaktifkan resolusi DNS di VPC pelanggan Anda, matikan parameter `rds.custom_dns_resolution` dari grup parameter DB khusus dengan mengaturnya ke 0. Kemudian mulai ulang instans DB agar perubahan terjadi.

Menyiapkan server DNS khusus

Setelah Anda menyiapkan nama server DNS, tunggu selama 30 menit hingga proses menyebarkan perubahan ke instans DB Anda selesai. Setelah perubahan diterapkan ke instans DB Anda, semua lalu lintas jaringan keluar memerlukan kueri pencarian DNS melalui port 53.

> **Note**
>
> Jika Anda tidak menyiapkan server DNS khusus dan `rds.custom_dns_resolution` diatur ke 1, host akan diselesaikan menggunakan zona pribadi Amazon Route 53. Untuk informasi selengkapnya, lihat Menggunakan zona yang di-hosting pribadi.

Untuk menyiapkan server DNS khusus untuk instans DB RDS for PostgreSQL

1. Dari opsi Protokol Konfigurasi Host Dinamis (DHCP) yang dipasang ke VPC, atur opsi `domain-name-servers` ke alamat IP dari server nama DNS Anda. Untuk informasi selengkapnya, lihat Set pilihan DHCP.

   > **Note**
   >
   > Opsi `domain-name-servers` menerima hingga empat nilai, tetapi instans DB Amazon RDS Anda hanya menggunakan nilai pertama.

2. Pastikan server DNS Anda dapat menyelesaikan semua kueri pencarian, termasuk nama DNS publik, nama DNS privat Amazon EC2 dan nama DNS khusus pelanggan. Jika lalu lintas jaringan keluar memuat setiap pencarian DNS yang tidak dapat ditangani server DNS Anda, server DNS Anda harus dikonfigurasikan oleh penyedia DNS hulu yang sesuai.
3. Konfigurasikan server DNS Anda untuk menghasilkan respons Protokol Datagram Pengguna (UDP) sebesar 512 bita atau kurang dari jumlah tersebut.
4. Konfigurasikan server DNS Anda untuk menghasilkan respons Protokol Kontrol Transmisi (TCP) sebesar 1024 bita atau kurang dari jumlah tersebut.
5. Konfigurasikan server DNS Anda agar lalu lintas dapat masuk dari instans DB Amazon RDS melalui port 53. Jika server DNS Anda berada di Amazon VPC, VPC harus memiliki grup keamanan yang berisi aturan masuk yang memungkinkan lalu lintas UDP dan TCP di port 53. Jika server DNS Anda tidak berada di Amazon VPC, VPC harus memiliki pengaturan firewall yang sesuai untuk memungkinkan traffic masuk UDP dan TCP di port 53.

   Untuk informasi lebih lanjut, lihat Grup keamanan untuk VPC Anda serta Menambahkan dan menghapus aturan.
6. Konfigurasikan VPC dari instans DB Amazon RDS agar lalu lintas keluar dapat melalui port 53. VPC Anda harus memiliki grup keamanan yang berisi aturan keluar yang memungkinkan lalu lintas UDP dan TCP di port 53.

   Untuk informasi selengkapnya, lihat Grup keamanan untuk VPC Anda serta Menambahkan dan menghapus aturan di Panduan Pengguna Amazon VPC.
7. Pastikan jalur perutean antara instans DB Amazon RDS dan server DNS dikonfigurasi dengan benar untuk memungkinkan lalu lintas DNS.

Selain itu, jika instans DB Amazon RDS dan server DNS tidak berada dalam VPC yang sama, pastikan koneksi peering disiapkan di antara keduanya. Untuk informasi selengkapnya, lihat Apa yang dimaksud peering VPC? di Panduan Peering Amazon VPC.

Meningkatkan mesin DB PostgreSQL untuk Amazon RDS

Ada dua jenis peningkatan yang dapat Anda kelola untuk basis data PostgreSQL Anda:

- Pembaruan sistem operasi – Terkadang, Amazon RDS mungkin perlu memperbarui sistem operasi yang mendasari basis data Anda untuk menerapkan perbaikan keamanan atau perubahan OS. Anda dapat memutuskan kapan Amazon RDS menerapkan pembaruan OS dengan menggunakan konsol RDS, AWS Command Line Interface (AWS CLI), atau RDS API. Untuk informasi selengkapnya tentang pembaruan OS, lihat Menerapkan pembaruan untuk instans DB.
- Peningkatan mesin basis data – Ketika Amazon RDS mendukung versi baru mesin basis data, Anda dapat meningkatkan basis data Anda ke versi baru.

Sebuah basis data dalam konteks ini adalah instans DB atau klaster DB Multi-AZ RDS for PostgreSQL.

Ada dua jenis peningkatan mesin untuk basis data PostgreSQL: peningkatan versi mayor dan peningkatan versi minor.

Peningkatan versi mayor

Peningkatan versi mayor dapat berisi perubahan basis data yang tidak memiliki kompatibilitas mundur dengan aplikasi yang ada. Oleh karena itu, Anda harus melakukan peningkatan versi mayor untuk basis data Anda secara manual. Anda dapat memulai peningkatan versi mayor dengan memodifikasi instans DB atau klaster DB Multi-AZ. Sebelum Anda melakukan peningkatan versi mayor, kami sarankan agar Anda mengikuti langkah-langkah yang dijelaskan dalam Memilih peningkatan versi mayor untuk PostgreSQL.

Jika Anda meningkatkan instans DB yang memiliki replika baca dalam Wilayah, Amazon RDS meningkatkan replika tersebut beserta instans DB primer.

Amazon RDS tidak meningkatkan replika baca klaster DB Multi-AZ. Jika Anda melakukan peningkatan versi mayor klaster DB Multi-AZ, maka status replikasi replika bacanya berubah menjadi diakhiri. Anda harus menghapus dan membuat ulang replika baca secara manual setelah peningkatan selesai.

> **Tip**
>
> Anda dapat meminimalkan waktu henti yang diperlukan untuk peningkatan versi mayor dengan menggunakan deployment blue/green. Untuk informasi selengkapnya, lihat Menggunakan Deployment Blue/Green untuk pembaruan basis data.

Peningkatan versi minor

Sebaliknya, tingkatkan versi minor hanya menyertakan perubahan yang kompatibel dengan aplikasi yang ada. Anda dapat memulai peningkatan versi minor secara manual dengan memodifikasi basis data Anda. Atau, Anda dapat mengaktifkan opsi Peningkatan versi minor otomatis saat membuat atau memodifikasi basis data. Tindakan ini akan membuat Amazon RDS secara otomatis meningkatkan basis data Anda setelah menguji dan menyetujui versi baru. Jika basis data PostgreSQL Anda menggunakan replika baca, Anda harus meningkatkan semua replika baca sebelum meningkatkan instans atau klaster sumber.

Jika basis data Anda adalah deployment instans DB Multi-AZ, Amazon RDS akan meningkatkan semua instans primer dan siaga secara bersamaan. Oleh karena itu, basis data Anda dapat tidak tersedia hingga peningkatan selesai. Jika basis data Anda adalah deployment klaster DB Multi-AZ, Amazon RDS meningkatkan instans DB pembaca satu per satu. Kemudian, salah satu instans DB pembaca beralih menjadi instans DB penulis baru. Amazon RDS kemudian meningkatkan instans penulis lama (yang sekarang menjadi instans pembaca).

> **Note**
>
> Waktu henti untuk peningkatan versi minor deployment instans DB Multi-AZ dapat berlangsung selama beberapa menit. Klaster DB Multi-AZ biasanya mengurangi waktu henti peningkatan versi minor menjadi sekitar 35 detik. Saat digunakan dengan Proksi RDS, Anda dapat mengurangi waktu henti menjadi satu detik atau kurang. Untuk informasi selengkapnya, lihat Menggunakan Proksi RDS. Sebagai alternatif, Anda dapat menggunakan proxy database open source seperti ProxySQL, PgBouncer, atau Driver AWS JDBC untuk MySQL.

Untuk informasi selengkapnya, lihat Peningkatan versi minor otomatis untuk PostgreSQL. Untuk informasi tentang melakukan peningkatan versi minor secara manual, lihat Meng-upgrade versi mesin secara manual.

Untuk informasi selengkapnya tentang versi mesin basis data dan kebijakan untuk menghentikan penggunaan versi mesin basis data, lihat Versi Mesin Basis Data dalam FAQ Amazon RDS.

Topik

Gambaran umum peningkatan PostgreSQL

Untuk meningkatkan basis data Anda secara aman, Amazon RDS menggunakan utilitas `pg_upgrade` yang dijelaskan dalam dokumentasi PostgreSQL.

Bila Anda menggunakan AWS Management Console untuk meng-upgrade database, ini menunjukkan target upgrade valid untuk database. Anda juga dapat menggunakan AWS CLI perintah berikut untuk mengidentifikasi target pemutakhiran yang valid untuk database:

Untuk Linux, macOS, atau Unix:

```
aws rds describe-db-engine-versions \
  --engine postgres \
  --engine-version version-number \
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Untuk Windows:

```
aws rds describe-db-engine-versions ^
  --engine postgres ^
  --engine-version version-number ^
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Misalnya, untuk mengidentifikasi target pemutakhiran yang valid untuk database PostgreSQL versi 12.13, jalankan perintah berikut: AWS CLI

Untuk Linux, macOS, atau Unix:

```
aws rds describe-db-engine-versions \
  --engine postgres \
  --engine-version 12.13 \
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Untuk Windows:

```
aws rds describe-db-engine-versions ^
  --engine postgres ^
  --engine-version 12.13 ^
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Amazon RDS mengambil dua snapshot DB selama proses peningkatan jika periode retensi cadangan Anda lebih besar dari 0. Snapshot DB pertama adalah snapshot dari basis data sebelum perubahan peningkatan dibuat. Jika peningkatan gagal untuk basis data Anda, Anda dapat memulihkan snapshot ini untuk membuat basis data yang menjalankan versi lama. Snapshot DB kedua diambil setelah peningkatan selesai.

> **Note**
>
> Amazon RDS mengambil snapshot DB selama proses peningkatan hanya jika Anda telah mengatur periode retensi cadangan untuk basis data Anda ke angka yang lebih besar dari 0. Untuk mengubah periode retensi cadangan untuk DB, lihat the section called “Memodifikasi instans DB”. Anda tidak dapat mengonfigurasi periode retensi cadangan kustom untuk klaster DB Multi-AZ.

Saat Anda melakukan peningkatan versi mayor pada instans DB, setiap replika baca dalam Wilayah juga otomatis ditingkatkan. Setelah alur kerja peningkatan dimulai, replika baca menunggu `pg_upgrade` untuk berhasil diselesaikan pada instans DB primer. Kemudian, tingkatkan instans DB primer menunggu peningkatan replika baca selesai. Anda akan mengalami pemadaman hingga

peningkatan selesai. Saat Anda melakukan peningkatan versi mayor klaster DB Multi-AZ, status replikasi replika bacanya berubah menjadi diakhiri.

Setelah peningkatan selesai, Anda tidak dapat kembali ke versi mesin DB yang lebih lama. Jika Anda ingin kembali ke versi yang lebih lama, pulihkan snapshot DB yang diambil sebelum peningkatan untuk membuat basis data baru.

Nomor versi PostgreSQL

Urutan penomoran versi untuk mesin basis data PostgreSQL adalah sebagai berikut:

- Untuk PostgreSQL versi 10 dan yang lebih baru, nomor versi mesin memiliki format mayor.minor. Nomor versi mayor adalah bagian bilangan bulat dari nomor versi. Nomor versi minor adalah bagian pecahan dari nomor versi.

  Peningkatan versi mayor akan meningkatkan bagian bilangan bulat dari nomor versi, seperti peningkatan dari 10.minor ke 11.minor.
- Untuk versi PostgreSQL yang lebih lama dari 10, nomor versi mesin memiliki format mayor.mayor.minor. Nomor versi mesin mayor adalah bagian bilangan bulat dan pecahan pertama dari nomor versi. Misalnya, 9.6 adalah versi mayor. Nomor versi minor adalah bagian ketiga dari nomor versi. Misalnya, untuk versi 9.6.12, angka 12 adalah nomor versi minor.

  Peningkatan versi mayor akan meningkatkan bagian mayor dari nomor versi. Misalnya, tingkatkan dari 9.6.12 ke 11.14 adalah peningkatan versi mayor, dengan 9.6 dan 11 adalah nomor versi mayor.

Untuk informasi tentang penomoran versi RDS Extended Support, lihat. Penamaan versi Amazon RDS Extended Support

Nomor versi RDS

Nomor versi RDS menggunakan skema penamaan *major*.*minor*.*patch*. Versi patch RDS mencakup perbaikan bug penting yang ditambahkan ke versi minor setelah dirilis. Untuk informasi tentang penomoran versi RDS Extended Support, lihat. Penamaan versi Amazon RDS Extended Support

Untuk mengidentifikasi nomor versi Amazon RDS untuk basis data Anda, Anda harus terlebih dahulu membuat ekstensi `rds_tools` dengan menggunakan perintah berikut:

```
CREATE EXTENSION rds_tools;
```

Mulai dari rilis PostgreSQL versi 15.2-R2, Anda dapat mengetahui nomor versi RDS untuk basis data RDS for PostgreSQL Anda dengan kueri SQL berikut:

```
postgres=> SELECT rds_tools.rds_version();
```

Misalnya, mengueri basis data RDS for PostgreSQL 15.2 akan menampilkan hal berikut:

```
rds_version
----------------
 15.2.R2
(1 row)
```

Memilih peningkatan versi mayor untuk PostgreSQL

Peningkatan versi mayor dapat berisi perubahan yang tidak memiliki kompatibilitas mundur dengan versi basis data yang lebih lama. Fungsi baru dapat menyebabkan aplikasi yang ada berhenti berfungsi dengan benar. Untuk alasan ini, Amazon RDS tidak menerapkan peningkatan versi mayor secara otomatis. Untuk melakukan peningkatan versi mayor, ubah basis data Anda secara manual. Pastikan bahwa Anda menguji secara menyeluruh setiap peningkatan untuk memverifikasi bahwa aplikasi Anda berfungsi dengan benar sebelum menerapkan peningkatan ke basis data produksi Anda. Saat Anda melakukan peningkatan versi mayor PostgreSQL, kami sarankan agar Anda mengikuti langkah yang dijelaskan dalam Cara melakukan peningkatan versi mayor.

Saat Anda meningkatkan instans DB AZ Tunggal atau deployment instans DB Multi-AZ PostgreSQL ke versi mayor berikutnya, replika baca apa pun yang terkait dengan basis data ini juga akan ditingkatkan ke versi mayor berikutnya. Dalam beberapa kasus, Anda dapat melompat ke versi mayor yang lebih tinggi saat meningkatkan. Jika peningkatan Anda melewati sebuah versi mayor, replika baca juga akan ditingkatkan ke versi mayor target tersebut. Peningkatan ke versi 11 yang melewati versi mayor lainnya akan memiliki batasan tertentu. Anda dapat menemukan detailnya dalam langkah-langkah yang dijelaskan dalam Cara melakukan peningkatan versi mayor.

Sebagian besar ekstensi PostgreSQL tidak ditingkatkan selama peningkatan mesin PostgreSQL. Hal ini harus ditingkatkan secara terpisah. Untuk informasi selengkapnya, lihat Meningkatkan ekstensi PostgreSQL.

Anda dapat mengetahui versi utama mana yang tersedia untuk database RDS untuk PostgreSQL Anda dengan menjalankan kueri berikut: AWS CLI

```
aws rds describe-db-engine-versions --engine postgres  --engine-version your-version
 --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Tabel berikut merangkum hasil kueri ini untuk semua versi yang tersedia. Tanda bintang (*) pada nomor versi berarti versi tersebut sudah tidak disarankan lagi. Jika versi Anda saat ini sudah tidak disarankan lagi, kami sarankan Anda meningkatkan ke target peningkatan versi minor terbaru atau ke salah satu target peningkatan lain yang tersedia untuk versi tersebut. Untuk informasi selengkapnya tentang penghentian RDS for PostgreSQL versi 9.6, lihat Penghentian PostgreSQL versi 9.6. Untuk informasi selengkapnya tentang penghentian RDS for PostgreSQL versi 10, lihat Penghentian PostgreSQL versi 10.

Vers sumb saat ini (* usan	Ta pe an ve ma ter	Target peningkatan lain yang tersedia						
16.1		16						
15.6	16							
15.5	16	16	15					
15.4	16	16	15	15				
15.3	16	16	15	15	15			
15.2	16	16	15	15	15	15		
14.1	16	15						
14.1(	16	15	15	14				
14.9	15	15	15	14	14			
14.8	15	15	15	15	14	14	14	

Vers sumb saat ini (* usan	Ta pe an ve ma ter	Target peningkatan lain yang tersedia																			
14.7	15	15	15	15	15	14	14	14	14												
14.6	15	15	15	15	15	14	14	14	14	14											
14.5	15	15	15	15	15	14	14	14	14	14	14										
14.4	15	15	15	15	15	14	14	14	14	14	14	14									
14.3	15	15	15	15	15	14	14	14	14	14	14	14	14								
14.2	15	15	15	15	15	14	14	14	14	14	14	14	14	14							
14.1	15	15	15	15	15	14	14	14	14	14	14	14	14	14	14						
13.1	16	15	14																		
13.1	16	15	14	14	13																
13.1	15	14	14	14	13	13															
13.1	15	14	14	14	14	13	13	13													
13.1	15	14	14	14	14	14	13	13	13	13											
13.9	14	14	14	14	14	14	13	13	13	13	13										
13.8	14	14	14	14	14	14	14	13	13	13	13	13	13								
13.7	14	14	14	14	14	14	14	14	14	13	13	13	13	13	13						
13.6	14	14	14	14	14	14	14	14	14	14	13	13	13	13	13	13	13				
13.5	14	14	14	14	14	14	14	14	14	14	14	13	13	13	13	13	13	13	13		

Versi sumber saat ini (* usang)	Tanggal pembaruan versi mayor terakhir	Target peningkatan lain yang tersedia																							
13.4*	14	14	14	14	14	14	14	14	14	14	14	13	13	13	13	13	13	13	13	13					
13.3*	14	14	14	14	14	14	14	14	14	14	14	13	13	13	13	13	13	13	13	13	13				
13.2* 13.1*	14	14	14	14	14	14	14	14	14	14	14	13	13	13	13	13	13	13	13	13	13				
12.18*	16	15	14	13																					
12.17*	16	15	14	13	13	12																			
12.16*	15	14	13	13	13	12	12																		
12.15*	15	14	13	13	13	13	12	12	12																
12.14*	15	14	13	13	13	13	13	12	12	12	12														
12.13*	14	13	13	13	13	13	13	12	12	12	12	12													
12.12*	14	13	13	13	13	13	13	13	12	12	12	12	12	12											
12.11*	14	14	13	13	13	13	13	13	13	13	12	12	12	12	12	12	12								
12.10*	14	13	13	13	13	13	13	13	13	13	12	12	12	12	12	12	12	12							
12.9*	14	13	13	13	13	13	13	13	13	13	13	12	12	12	12	12	12	12	12	12					
12.8*	13	13	13	13	13	13	13	13	13	13	13	12	12	12	12	12	12	12	12	12	12				
12.7*	13	13	13	13	13	13	13	13	13	13	13	13	12	12	12	12	12	12	12	12	12	12	12.8		

Vers sumb saat ini (* usan	Ta pe an ve ma ter	Target peningkatan lain yang tersedia																					
12.6 12.5 12.4 12.3 12.2	13	13	13	13	13	13	12	12	12	12	12	12	12	12	12	12	12	12					
11.2	16	15	14	13	12																		
11.2	15	14	13	12	12	11																	
11.2	15	14	13	12	12	12	11	11															
11.1	15	14	13	12	12	12	12	11	11	11													
11.1	14	13	12	12	12	12	12	11	11	11	11												
11.1	14	13	12	12	12	12	12	12	11	11	11	11	11										
11.1	14	14	13	12	12	12	12	12	12	12	11	11	11	11	11	11							
11.1	14	13	12	12	12	12	12	12	12	12	11	11	11	11	11	11	11						
11.1	14	13	12	12	12	12	12	12	12	12	12	11	11	11	11	11	11	11	11				
11.1	13	12	12	12	12	12	12	12	12	12	12	11	11	11	11	11	11	11	11	11			
11.1	13	12	12	12	12	12	12	12	12	12	12	12	11	11	11	11	11	11	11	11	11	11	
10.2	14	13	12	11	11	11	11	11															
10.2	14	13	12	11	11	11	11	11	11	10													
10.2	14	14	13	12	11	11	11	11	11	11	11	10	10										

Vers sumb saat ini (* usan	Ta pe an ve ma ter	Target peningkatan lain yang tersedia																			
10.2(	14	13	12	11	11	11	11	11	11	11	11	10	10	10							
10.1!	14	13	12	11	11	11	11	11	11	11	11	11	10	10	10	10					
10.1!	13	12	11	11	11	11	11	11	11	11	11	11	10	10	10	10	10				
10.1'	13	12	11	11	11	11	11	11	11	11	11	11	11	10	10	10	10	10	10		
9.6.2	14	13	12	11	10	10															
9.6.2	13	12	11	10	10	10	9.6														
9.6.2	13	12	11	10	10	10	10	9.6	9.6												
9.6.1	9.6	14	13	12	11	10	10	9.6	9.6												
9.6.1																					
9.6.1																					
9.6.1																					
9.6.1																					
9.6.1																					
9.6.1																					
9.6.1																					
6.10'																					
9.6.9																					
9.6.8																					
9.6.6																					
9.6.5																					
9.6.3																					
9.6.2																					
9.6.1																					

Cara melakukan peningkatan versi mayor

Kami merekomendasikan proses berikut saat melakukan peningkatan versi mayor pada basis data Amazon RDS for PostgreSQL:

1. Siapkan grup parameter yang kompatibel dengan versi – Jika Anda menggunakan grup parameter kustom, Anda memiliki dua opsi. Anda dapat menentukan grup parameter default untuk versi mesin DB baru. Atau Anda dapat membuat grup parameter kustom Anda sendiri untuk versi mesin DB baru. Lihat informasi yang lebih lengkap di the section called “Bekerja dengan grup parameter” dan the section called “Bekerja dengan grup parameter klaster DB”.
2. Periksa kelas basis data yang tidak didukung – Periksa apakah kelas instans basis data Anda kompatibel dengan versi PostgreSQL yang menjadi target peningkatan Anda. Untuk informasi selengkapnya, lihat Mesin DB yang didukung untuk kelas instans DB.
3. Periksa penggunaan yang tidak didukung:
   - Transaksi yang disiapkan – Komit atau rollback semua transaksi terbuka yang disiapkan sebelum mencoba melakukan peningkatan.

     Anda dapat menggunakan kueri berikut untuk memverifikasi bahwa tidak ada transaksi terbuka yang disiapkan pada basis data Anda.

     ```
     SELECT count(*) FROM pg_catalog.pg_prepared_xacts;
     ```

   - Jenis data reg* – Hapus semua penggunaan jenis data reg* sebelum mencoba peningkatan. Kecuali untuk `regtype` dan `regclass`, Anda tidak dapat meningkatkan jenis data reg*. Utilitas `pg_upgrade` tidak dapat mempersistensi jenis data ini, yang digunakan oleh Amazon RDS untuk melakukan peningkatan.

     Untuk memverifikasi bahwa jenis data reg* yang tidak didukung tidak digunakan, gunakan kueri berikut untuk setiap basis data.

     ```
     SELECT count(*) FROM pg_catalog.pg_class c, pg_catalog.pg_namespace n,
      pg_catalog.pg_attribute a
       WHERE c.oid = a.attrelid
           AND NOT a.attisdropped
           AND a.atttypid IN ('pg_catalog.regproc'::pg_catalog.regtype,
                              'pg_catalog.regprocedure'::pg_catalog.regtype,
                              'pg_catalog.regoper'::pg_catalog.regtype,
                              'pg_catalog.regoperator'::pg_catalog.regtype,
     ```

```
                        'pg_catalog.regconfig'::pg_catalog.regtype,
                        'pg_catalog.regdictionary'::pg_catalog.regtype)
      AND c.relnamespace = n.oid
      AND n.nspname NOT IN ('pg_catalog', 'information_schema');
```

4. Tangani slot replikasi logis – Peningkatan tidak dapat terjadi jika basis data memiliki slot replikasi logis. Slot replikasi logis biasanya digunakan untuk migrasi AWS DMS dan untuk mereplikasi tabel dari basis data ke danau data, alat BI, dan target lainnya. Sebelum meningkatkan, pastikan Anda mengetahui tujuan dari setiap slot replikasi logis yang digunakan, dan konfirmasikan bahwa menghapusnya tidak akan jadi masalah. Jika slot replikasi logis masih digunakan, Anda tidak boleh menghapusnya, dan Anda tidak dapat melakukan peningkatan.

   Jika slot replikasi logis tidak diperlukan, Anda dapat menghapusnya menggunakan SQL berikut:

   ```
   SELECT * FROM pg_replication_slots;
   SELECT pg_drop_replication_slot(slot_name);
   ```

   Pengaturan replikasi logis yang menggunakan ekstensi `pglogical` juga harus dihapus beberapa slotnya agar peningkatan versi mayor berhasil dilakukan. Untuk informasi tentang cara mengidentifikasi dan menghapus slot yang dibuat menggunakan ekstensi `pglogical`, lihat Mengelola slot replikasi logis untuk RDS for PostgreSQL.

5. Tangani replika baca – Peningkatan instans DB AZ Tunggal atau deployment instans DB Multi-AZ juga akan meningkatkan replika baca dalam Wilayah bersama instans DB primer. Amazon RDS tidak meningkatkan replika baca klaster DB Multi-AZ.

   Anda tidak dapat meningkatkan replika baca secara terpisah. Jika Anda bisa, hal ini dapat menyebabkan situasi ketika basis data primer dan replika memiliki perbedaan versi mayor PostgreSQL. Namun, tingkatkan replika baca dapat menambah waktu henti pada instans DB primer. Untuk mencegah peningkatan replika, promosikan replika menjadi instans mandiri atau hapus replika tersebut sebelum memulai proses peningkatan.

   Proses peningkatan membuat ulang grup parameter replika baca berdasarkan grup parameter saat ini dari replika baca. Anda dapat menerapkan grup parameter kustom ke replika baca hanya setelah peningkatan selesai dengan memodifikasi replika baca. Untuk informasi selengkapnya tentang replika baca, lihat Menggunakan replika baca untuk Amazon RDS for PostgreSQL.

6. Lakukan pencadangan – Kami sarankan Anda melakukan pencadangan sebelum melakukan peningkatan versi mayor sehingga Anda memiliki titik pemulihan yang diketahui untuk basis

data Anda. Jika periode retensi cadangan Anda lebih besar dari 0, proses peningkatan akan membuat snapshot DB dari basis data Anda sebelum dan setelah peningkatan. Untuk mengubah periode retensi cadangan Anda, lihat Memodifikasi instans DB Amazon RDS dan the section called "Mengubah klaster basis data Multi-AZ".

Untuk melakukan pencadangan secara manual, lihat the section called "Membuat snapshot DB untuk instans DB Single-AZ" dan the section called "Membuat snapshot klaster DB Multi-AZ".

7. Tingkatkan ekstensi tertentu sebelum peningkatan versi mayor – Jika Anda berencana untuk melewati sebuah versi mayor dengan peningkatan, Anda perlu memperbarui ekstensi tertentu sebelum melakukan peningkatan versi mayor. Misalnya, sebuah versi mayor dilewati dalam peningkatan dari versi 9.5.x atau 9,6.x ke versi 11.x. Ekstensi yang perlu diperbarui mencakup PostGIS dan ekstensi terkait untuk memproses data spasial.
   - `address_standardizer`
   - `address_standardizer_data_us`
   - `postgis_raster`
   - `postgis_tiger_geocoder`
   - `postgis_topology`

   Jalankan perintah berikut untuk setiap ekstensi yang Anda gunakan:

   ```
   ALTER EXTENSION PostgreSQL-extension UPDATE TO 'new-version';
   ```

   Untuk informasi selengkapnya, lihat Meningkatkan ekstensi PostgreSQL. Untuk mempelajari selengkapnya tentang peningkatan PostGIS, lihat Langkah 6: Meningkatkan ekstensi PostGIS.
8. Hapus ekstensi tertentu sebelum peningkatan versi mayor – Peningkatan yang melewati versi mayor ke versi 11.x tidak mendukung pembaruan ekstensi `pgRouting`. Sebuah versi mayor dilewati dalam peningkatan dari versi 9.4.x, 9,5.x, atau 9,6.x ke versi 11.x. Aman untuk menghapus ekstensi `pgRouting` lalu menginstalnya kembali ke versi yang kompatibel setelah peningkatan. Untuk versi ekstensi yang dapat Anda perbarui, lihat Versi ekstensi PostgreSQL yang didukung.

   Ekstensi `tsearch2` dan `chkpass` tidak lagi didukung untuk PostgreSQL versi 11 atau yang lebih baru. Jika Anda meningkatkan ke versi 11.x, hapus ekstensi `tsearch2` dan `chkpass` sebelum peningkatan.
9. Hapus jenis data yang tidak diketahui – Hapus jenis data `unknown` tergantung pada versi target.

PostgreSQL versi 10 berhenti mendukung jenis data `unknown`. Jika basis data versi 9.6 menggunakan jenis data `unknown`, tingkatkan ke versi 10 akan menunjukkan pesan kesalahan seperti berikut ini:

```
Database instance is in a state that cannot be upgraded: PreUpgrade checks failed:
The instance could not be upgraded because the 'unknown' data type is used in user
 tables.
Please remove all usages of the 'unknown' data type and try again."
```

Untuk menemukan jenis data `unknown` di basis data Anda sehingga Anda dapat menghapus kolom yang melanggar atau mengubahnya ke jenis data yang didukung, gunakan SQL berikut:

```
SELECT DISTINCT data_type FROM information_schema.columns WHERE data_type ILIKE
 'unknown';
```

10. Lakukan percobaan peningkatan – Kami sangat menyarankan Anda untuk menguji peningkatan versi mayor pada duplikat basis data produksi Anda sebelum mencoba peningkatan pada basis data produksi Anda. Anda dapat memantau rencana eksekusi pada basis data uji duplikat untuk setiap kemungkinan regresi rencana eksekusi dan untuk mengevaluasi performanya. Untuk membuat contoh pengujian duplikat, Anda dapat memulihkan database Anda dari snapshot terbaru atau melakukan point-in-time pemulihan database Anda ke waktu restorable terbaru.

    Untuk informasi selengkapnya, lihat the section called “Memulihkan dari snapshot” atau the section called “Memulihkan instans DB dengan waktu yang ditentukan”. Untuk klaster DB Multi-AZ, lihat the section called “Memulihkan dari snapshot ke klaster DB Multi-AZ” atau the section called “Memulihkan klaster DB Multi-AZ ke waktu tertentu”.

    Untuk detail tentang melakukan peningkatan, lihat the section called “Meng-upgrade versi mesin secara manual”.

    Dalam meningkatkan basis data versi 9.6 ke versi 10, ketahuilah bahwa PostgreSQL 10 mengaktifkan kueri paralel secara default. Anda dapat menguji dampak paralelisme sebelum peningkatan dengan mengubah parameter `max_parallel_workers_per_gather` pada basis data uji Anda menjadi 2.

> **Note**
>
> Nilai default untuk parameter `max_parallel_workers_per_gather` dalam grup parameter DB `default.postgresql10` adalah 2.

Untuk informasi selengkapnya, lihat Parallel Query dalam dokumentasi PostgreSQL. Untuk menonaktifkan paralelisme pada versi 10, atur parameter `max_parallel_workers_per_gather` ke 0.

Selama peningkatan versi mayor, basis data `public` dan `template1` serta skema `public` di setiap basis data akan diubah namanya untuk sementara. Objek-objek ini muncul dalam log menggunakan nama aslinya dan string acak. String ditambahkan sehingga pengaturan kustom seperti `locale` dan `owner` dipertahankan selama peningkatan versi mayor. Setelah peningkatan selesai, objek diubah namanya kembali ke nama aslinya.

> **Note**
>
> Selama proses pemutakhiran versi utama, Anda tidak dapat melakukan point-in-time pemulihan instans DB atau cluster DB multi-AZ Anda. Setelah melakukan peningkatan, Amazon RDS akan membuat cadangan otomatis dari basis data. Anda dapat melakukan point-in-time pemulihan ke waktu sebelum pemutakhiran dimulai dan setelah pencadangan otomatis database Anda selesai.

11. Jika peningkatan gagal karena kesalahan prosedur pra-pemeriksaan, atasi masalah tersebut – Selama proses peningkatan versi mayor, Amazon RDS for PostgreSQL pertama-tama menjalankan prosedur pra-pemeriksaan untuk mengidentifikasi masalah yang mungkin menyebabkan peningkatan gagal. Prosedur pra-pemeriksaan akan memeriksa semua kemungkinan kondisi yang tidak kompatibel di seluruh basis data dalam instans.

    Jika pra-pemeriksaan menemui masalah, proses ini akan membuat peristiwa log yang menunjukkan bahwa pra-pemeriksaan peningkatan gagal. Detail proses pra-pemeriksaan ada dalam log peningkatan yang bernama `pg_upgrade_precheck.log` untuk semua basis data yang berasal dari sebuah basis data tertentu. Amazon RDS menambahkan stempel waktu ke nama file. Untuk informasi selengkapnya tentang melihat log, lihat Memantau file log Amazon RDS.

Jika peningkatan replika baca gagal pada saat pra-pemeriksaan, replikasi pada replika baca yang gagal rusak dan replika baca diberi status diakhiri. Hapus replika baca ini dan buat ulang replika baca baru berdasarkan instans DB primer yang ditingkatkan.

Selesaikan semua masalah yang teridentifikasi dalam log pra-pemeriksaan lalu coba lagi peningkatan versi mayor. Berikut ini adalah contoh log pra-pemeriksaan.

```
-------------------------------------------------------------------------
Upgrade could not be run on Wed Apr 4 18:30:52 2018
-------------------------------------------------------------------------
The instance could not be upgraded from 9.6.11 to 10.6 for the following reasons.
Please take appropriate action on databases that have usage incompatible with the
 requested major engine version upgrade and try the upgrade again.

* There are uncommitted prepared transactions. Please commit or rollback all prepared
 transactions.* One or more role names start with 'pg_'. Rename all role names that
 start with 'pg_'.

* The following issues in the database 'my"million$"db' need to be corrected before
 upgrading:** The ["line","reg*"] data types are used in user tables. Remove all
 usage of these data types.
** The database name contains characters that are not supported by RDS for
 PostgreSQL. Rename the database.
** The database has extensions installed that are not supported on the target
 database version. Drop the following extensions from your database: ["tsearch2"].

* The following issues in the database 'mydb' need to be corrected before
 upgrading:** The database has views or materialized views that depend on
 'pg_stat_activity'. Drop the views.
```

12. Jika peningkatan replika baca gagal saat meningkatkan basis data, atasi masalah tersebut – Replika baca gagal diubah ke status `incompatible-restore` dan replikasi diakhiri pada basis data. Hapus replika baca ini dan buat ulang replika baca baru berdasarkan instans DB primer yang ditingkatkan.

> **Note**
>
> Amazon RDS tidak meningkatkan replika baca untuk klaster DB Multi-AZ. Jika Anda melakukan peningkatan versi mayor klaster DB Multi-AZ, maka status replikasi replika bacanya berubah menjadi diakhiri.

Peningkatan replika baca dapat gagal karena alasan berikut:

- Replika baca tidak dapat mengimbangi instans primer bahkan setelah waktu tunggu.
- Replika baca berada dalam status siklus hidup akhir atau tidak kompatibel seperti penyimpanan penuh, pemulihan yang tidak kompatibel, dan seterusnya.
- Saat peningkatan instans DB primer dimulai, terdapat peningkatan versi minor terpisah yang berjalan pada replika baca.
- Replika baca menggunakan parameter yang tidak kompatibel.
- Replika baca tidak dapat berkomunikasi dengan instans DB primer untuk menyinkronkan folder data.

13. Tingkatkan basis data produksi Anda – Ketika percobaan peningkatan versi mayor yang dijalankan berhasil, Anda dapat meningkatkan basis data produksi Anda dengan percaya diri. Untuk informasi selengkapnya, lihat Meng-upgrade versi mesin secara manual.

14. Jalankan operasi `ANALYZE` untuk menyegarkan tabel `pg_statistic`. Anda harus melakukannya untuk setiap basis data pada semua basis data PostgreSQL Anda. Statistik pengoptimisasi tidak ditransfer selama peningkatan versi mayor, jadi Anda perlu membuat ulang semua statistik untuk menghindari masalah performa. Jalankan perintah tanpa parameter apa pun untuk menghasilkan statistik untuk semua tabel reguler dalam basis data saat ini, sebagai berikut:

```
ANALYZE VERBOSE;
```

Bendera `VERBOSE` bersifat opsional, tetapi kemajuannya akan ditampilkan jika digunakan. Untuk informasi selengkapnya, lihat ANALYZE di dokumentasi PostgreSQL.

> **Note**
>
> Jalankan ANALYZE pada sistem Anda setelah peningkatan untuk menghindari masalah performa.

Setelah peningkatan versi mayor selesai, kami menyarankan hal berikut:

- Peningkatan mesin PostgreSQL tidak meningkatkan ekstensi PostgreSQL apa pun. Untuk meningkatkan ekstensi, lihat Meningkatkan ekstensi PostgreSQL.

- Atau, gunakan Amazon RDS untuk melihat dua log yang dihasilkan oleh utilitas `pg_upgrade`. Log tersebut adalah `pg_upgrade_internal.log` dan `pg_upgrade_server.log`. Amazon RDS menambahkan stempel waktu ke nama file untuk log ini. Anda dapat melihat log ini sebagaimana Anda dapat melihat log lainnya. Untuk informasi selengkapnya, lihat Memantau file log Amazon RDS.

  Anda juga dapat mengunggah log pemutakhiran ke Amazon CloudWatch Logs. Untuk informasi selengkapnya, lihat Menerbitkan log PostgreSQL ke Amazon Logs CloudWatch .
- Untuk memverifikasi bahwa segalanya berfungsi seperti yang diharapkan, uji aplikasi Anda pada basis data yang ditingkatkan dengan beban kerja serupa. Setelah peningkatan diverifikasi, Anda dapat menghapus instans uji ini.

Peningkatan versi minor otomatis untuk PostgreSQL

Jika Anda mengaktifkan Peningkatan versi minor otomatis saat membuat atau memodifikasi instans DB atau klaster DB Multi-AZ, Anda dapat melakukan peningkatan basis data secara otomatis.

Untuk setiap versi mayor RDS for PostgreSQL, satu versi minor ditetapkan oleh RDS sebagai versi peningkatan otomatis. Setelah versi minor diuji dan disetujui oleh Amazon RDS, tingkatkan versi minor terjadi secara otomatis selama periode pemeliharaan Anda. RDS tidak secara otomatis menetapkan versi minor yang lebih baru sebagai versi peningkatan otomatis. Sebelum RDS menetapkan versi peningkatan otomatis yang lebih baru, beberapa kriteria dipertimbangkan, seperti yang berikut ini:

- Masalah keamanan yang diketahui
- Bug dalam versi komunitas PostgreSQL
- Stabilitas armada secara keseluruhan sejak versi minor dirilis

Anda dapat menggunakan AWS CLI perintah berikut untuk menentukan versi target pemutakhiran minor otomatis saat ini untuk versi minor PostgreSQL tertentu secara spesifik. Wilayah AWS

Untuk Linux, macOS, atau Unix:

```
aws rds describe-db-engine-versions \
--engine postgres \
--engine-version minor-version \
--region region \
```

```
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
--output text
```

Untuk Windows:

```
aws rds describe-db-engine-versions ^
--engine postgres ^
--engine-version minor-version ^
--region region ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output text
```

Misalnya, AWS CLI perintah berikut menentukan target upgrade minor otomatis untuk PostgreSQL minor versi 12.13 di US East (Ohio) (us-east-2). Wilayah AWS

Untuk Linux, macOS, atau Unix:

```
aws rds describe-db-engine-versions \
--engine postgres \
--engine-version 12.13 \
--region us-east-2 \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
--output table
```

Untuk Windows:

```
aws rds describe-db-engine-versions ^
--engine postgres ^
--engine-version 12.13 ^
--region us-east-2 ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output table
```

Output Anda akan seperti yang berikut ini.

```
---------------------------------
```

```
|    DescribeDBEngineVersions    |
+--------------+-----------------+
|  AutoUpgrade |  EngineVersion  |
+--------------+-----------------+
|  True        |  12.14          |
|  False       |  12.15          |
|  False       |  13.9           |
|  False       |  13.10          |
|  False       |  13.11          |
|  False       |  14.6           |
+--------------+-----------------+
```

Dalam contoh ini, nilai `AutoUpgrade` adalah `True` untuk PostgreSQL versi 12.14. Jadi, target peningkatan minor otomatis adalah PostgreSQL versi 12.14, yang disorot dalam output.

Basis data PostgreSQL secara otomatis ditingkatkan selama periode pemeliharaan Anda jika kriteria berikut terpenuhi:

- Basis data memiliki opsi Peningkatan versi minor otomatis diaktifkan.
- Basis data menjalankan versi mesin DB minor yang lebih rendah dari versi minor peningkatan otomatis saat ini.

Untuk informasi selengkapnya, lihat Meng-upgrade versi mesin minor secara otomatis.

> **Note**
>
> Peningkatan mesin PostgreSQL tidak meningkatkan ekstensi PostgreSQL apa pun. Untuk meningkatkan ekstensi, lihat Meningkatkan ekstensi PostgreSQL.

Meningkatkan ekstensi PostgreSQL

Peningkatan mesin PostgreSQL tidak meningkatkan sebagian besar ekstensi PostgreSQL. Untuk memperbarui ekstensi setelah peningkatan versi, gunakan perintah `ALTER EXTENSION UPDATE`.

> **Note**
>
> Untuk informasi selengkapnya tentang ekstensi PostGIS, lihat Mengelola data spasial dengan ekstensi PostGIS (Langkah 6: Meningkatkan ekstensi PostGIS).

Untuk memperbarui ekstensi `pg_repack`, hapus ekstensi ini lalu buat versi baru di basis data yang ditingkatkan. Untuk informasi selengkapnya, lihat pg_repack installation dalam dokumentasi `pg_repack`.

Untuk meningkatkan ekstensi, gunakan perintah berikut.

```
ALTER EXTENSION extension_name UPDATE TO 'new_version';
```

Untuk daftar versi ekstensi PostgreSQL yang didukung, lihat Versi ekstensi PostgreSQL yang didukung.

Untuk menampilkan daftar ekstensi yang saat ini diinstal, gunakan katalog pg_extension PostgreSQL dalam perintah berikut.

```
SELECT * FROM pg_extension;
```

Untuk melihat daftar versi ekstensi tertentu yang tersedia untuk penginstalan Anda, gunakan tampilan pg_available_extension_versions PostgreSQL dalam perintah berikut.

```
SELECT * FROM pg_available_extension_versions;
```

Meng-upgrade versi mesin snapshot DB PostgreSQL

Dengan Amazon RDS, Anda dapat membuat volume penyimpanan snapshot DB dari instans DB PostgreSQL. Saat Anda membuat snapshot DB, snapshot didasarkan pada versi mesin yang digunakan oleh instans Amazon RDS Anda. Selain meng-upgrade versi mesin DB dari instans DB Anda, Anda juga dapat mengupgrade versi mesin untuk snapshot DB Anda.

Setelah memulihkan snapshot DB yang di-upgrade ke versi mesin baru, pastikan untuk menguji bahwa upgrade berhasil. Untuk informasi selengkapnya tentang upgrade versi mayor, lihat Meningkatkan mesin DB PostgreSQL untuk Amazon RDS. Untuk mempelajari cara memulihkan snapshot DB RDS, lihat Memulihkan dari snapshot DB.

Anda dapat meng-upgrade snapshot DB manual yang dienkripsi atau tidak dienkripsi.

Untuk daftar versi mesin yang tersedia untuk meng-upgrade snapshot DB, lihat Meng-upgrade mesin DB PostgreSQL untuk Amazon RDS.

> **Note**
>
> Anda tidak dapat meng-upgrade snapshot DB otomatis yang dibuat selama proses pencadangan otomatis.

Konsol

Untuk meng-upgrade snapshot DB

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Snapshot.
3. Pilih snapshot yang ingin Anda upgrade.
4. Untuk Tindakan, pilih Tingkatkan snapshot. Halaman Tingkatkan snapshot muncul.
5. Pilih Versi mesin baru yang akan menjadi target upgrade.
6. Pilih Simpan perubahan untuk meng-upgrade snapshot.

   Selama proses upgrade, semua tindakan snapshot dinonaktifkan untuk snapshot DB ini. Selain itu, status snapshot DB berubah dari tersedia menjadi meningkatkan, lalu berubah menjadi aktif

setelah selesai. Jika snapshot DB tidak dapat di-upgrade karena masalah kerusakan snapshot, status berubah menjadi tidak tersedia. Anda tidak dapat memulihkan snapshot dari status ini.

> **Note**
>
> Jika upgrade snapshot DB gagal, snapshot di-rollback ke kondisi awal sesuai dengan versi asli.

AWS CLI

Untuk memutakhirkan snapshot DB ke versi mesin database baru, gunakan AWS CLI modify-db-snapshotperintah.

Parameter

- `--db-snapshot-identifier` – Pengidentifikasi untuk snapshot DB. Pengidentifikasi harus berupa Amazon Resource Name (ARN) yang unik. Untuk informasi selengkapnya, lihat Bekerja dengan Amazon Resource Name (ARN) di Amazon RDS.
- `--engine-version` – Versi mesin untuk meng-upgrade snapshot DB.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-snapshot \
    --db-snapshot-identifier my_db_snapshot \
    --engine-version new_version
```

Untuk Windows:

```
aws rds modify-db-snapshot ^
    --db-snapshot-identifier my_db_snapshot ^
    --engine-version new_version
```

API RDS

Untuk meng-upgrade snapshot DB ke versi mesin basis data baru, panggil operasi API Amazon RDS ModifyDBSnapshot.

- `DBSnapshotIdentifier` – Pengidentifikasi untuk snapshot DB. Pengidentifikasi harus berupa Amazon Resource Name (ARN) yang unik. Untuk informasi selengkapnya, lihat Bekerja dengan Amazon Resource Name (ARN) di Amazon RDS.
- `EngineVersion` – Versi mesin untuk meng-upgrade snapshot DB.

Menggunakan replika baca untuk Amazon RDS for PostgreSQL

Anda dapat menskalakan pembacaan untuk instans DB Amazon RDS for PostgreSQL dengan menambahkan replika baca ke instans. Seperti mesin basis data Amazon RDS lainnya, RDS for PostgreSQL menggunakan mekanisme replikasi native PostgreSQL untuk terus memperbarui replika baca dengan perubahan pada DB sumber. Untuk informasi umum tentang replika baca dan Amazon RDS, lihat Menggunakan replika baca instans DB.

Di bagian berikut ini, Anda dapat menemukan informasi spesifik untuk menggunakan replika baca dengan RDS for PostgreSQL.

Pendekodean logis pada replika baca

RDS for PostgreSQL mendukung replikasi logis dari replika siaga dengan PostgreSQL 16.1. Hal ini memungkinkan Anda membuat pendekodean logis dari replika siaga hanya baca yang mengurangi beban pada instans DB primer. Anda dapat mencapai ketersediaan yang lebih tinggi untuk aplikasi Anda yang perlu menyinkronkan data di beberapa sistem. Fitur ini meningkatkan performa gudang data dan analitik data Anda.

Selain itu, slot replikasi pada replika siaga tertentu akan mempersistensi promosi replika siaga tersebut menjadi replika primer. Artinya, jika terjadi failover instans DB primer atau promosi replika siaga menjadi replika primer baru, slot replikasi akan dipersistensi dan pelanggan replika siaga sebelumnya tidak akan terpengaruh.

Untuk membuat pendekodean logis pada replika baca

1. Aktifkan replikasi logis – Untuk membuat pendekodean logis pada replika siaga, Anda harus mengaktifkan replikasi logis pada instans DB sumber Anda dan replika fisiknya. Untuk informasi selengkapnya, lihat Konfigurasi replika baca dengan PostgreSQL.

   - Untuk mengaktifkan replikasi logis untuk instans DB RDS for PostgreSQL yang baru dibuat – Buat grup parameter kustom DB baru dan atur parameter statis `rds.logical_replication` ke 1. Kemudian, kaitkan grup parameter DB ini dengan instans DB sumber dan replika baca fisiknya. Untuk informasi selengkapnya, lihat Mengaitkan grup parameter DB dengan instans DB.
   - Untuk mengaktifkan replikasi logis untuk instans DB RDS for PostgreSQL yang ada – Ubah grup parameter kustom DB dari instans DB sumber dan replika baca fisiknya untuk mengatur

parameter statis `rds.logical_replication` ke 1. Untuk informasi selengkapnya, lihat Memodifikasi parameter dalam grup parameter DB.

> **Note**
>
> Anda harus mem-boot ulang instans DB untuk menerapkan perubahan parameter ini.

Anda dapat menggunakan kueri berikut untuk memverifikasi nilai untuk `wal_level` dan `rds.logical_replication` pada instans DB sumber dan replika baca fisiknya.

```
Postgres=>SELECT name,setting FROM pg_settings WHERE name IN
 ('wal_level','rds.logical_replication');

 name                    | setting
-------------------------+---------
 rds.logical_replication | on
 wal_level               | logical
(2 rows)
```

2. Buat tabel di basis data sumber – Hubungkan ke basis data di instans DB sumber Anda. Untuk informasi selengkapnya, lihat Menghubungkan ke instans DB yang menjalankan mesin basis data PostgreSQL.

   Gunakan kueri berikut untuk membuat tabel di basis data sumber Anda dan untuk menyisipkan nilai:

   ```
   Postgres=>CREATE TABLE LR_test (a int PRIMARY KEY);
   CREATE TABLE
   ```

   ```
   Postgres=>INSERT INTO LR_test VALUES (generate_series(1,10000));
   INSERT 0 10000
   ```

3. Buat penerbitan untuk tabel sumber – Gunakan kueri berikut untuk membuat penerbitan untuk tabel pada instans DB sumber.

   ```
   Postgres=>CREATE PUBLICATION testpub FOR TABLE LR_test;
   CREATE PUBLICATION
   ```

Gunakan kueri SELECT untuk memverifikasi detail penerbitan yang dibuat pada instans DB sumber dan instans replika baca fisik.

```
Postgres=>SELECT * from pg_publication;

oid    | pubname | pubowner | puballtables | pubinsert | pubupdate | pubdelete |
 pubtruncate | pubviaroot
-------+---------+----------+--------------+-----------+-----------+-----------
+-------------+------------
 16429 | testpub |    16413 | f            | t         | t         | t         | t
          | f
(1 row)
```

4. Buat langganan dari instans replika logis – Buat instans DB RDS for PostgreSQL lain sebagai instans replika logis. Pastikan VPC diatur dengan benar untuk memastikan bahwa instans replika logis ini dapat mengakses instans replika baca fisik. Untuk informasi selengkapnya, lihat Amazon VPC dan Amazon RDS. Jika instans DB sumber Anda dalam kondisi idle, masalah konektivitas mungkin terjadi dan replika primer tidak mengirim data ke replika siaga.

```
Postgres=>CREATE SUBSCRIPTION testsub CONNECTION 'host=Physical replica host name
 port=port
                dbname=source_db_name user=user password=password
 PUBLICATION testpub;
NOTICE:  created replication slot "testsub" on publisher
CREATE SUBSCRIPTION
```

```
Postgres=>CREATE TABLE LR_test (a int PRIMARY KEY);
CREATE TABLE
```

Gunakan kueri SELECT untuk memverifikasi detail langganan pada instans replika logis.

```
Postgres=>SELECT oid,subname,subenabled,subslotname,subpublications FROM
 pg_subscription;

oid    | subname | subenabled | subslotname | subpublications
-------+---------+------------+-------------+-----------------
 16429 | testsub | t          | testsub     | {testpub}
(1 row)
postgres=> select count(*) from LR_test;
 count
```

```
-------
 10000
(1 row)
```

5. Periksa status slot replikasi logis – Anda hanya dapat melihat slot replikasi fisik pada instans DB sumber Anda.

```
Postgres=>select slot_name, slot_type, confirmed_flush_lsn from
 pg_replication_slots;

slot_name                                    | slot_type | confirmed_flush_lsn
---------------------------------------------+-----------+---------------------
 rds_us_west_2_db_dhqfsmo5wbbjqrn3m6b6ivdhu4 | physical  |
(1 row)
```

Namun, pada instans replika baca Anda, Anda dapat melihat slot replikasi logis dan nilai `confirmed_flush_lsn` berubah saat aplikasi secara aktif mengonsumsi perubahan logis.

```
Postgres=>select slot_name, slot_type, confirmed_flush_lsn from
 pg_replication_slots;

slot_name | slot_type | confirmed_flush_lsn
-----------+-----------+---------------------
 testsub   | logical   | 0/500002F0
(1 row)
```

```
Postgres=>select slot_name, slot_type, confirmed_flush_lsn from
 pg_replication_slots;

slot_name | slot_type | confirmed_flush_lsn
-----------+-----------+---------------------
 testsub   | logical   | 0/5413F5C0
(1 row)
```

Batasan replika baca dengan PostgreSQL

Berikut ini adalah batasan untuk replika baca PostgreSQL:

> **Note**
>
> Replika baca untuk instans DB Multi-AZ dan AZ Tunggal RDS for PostgreSQL yang menjalankan PostgreSQL versi 12 dan yang lebih lama akan di-boot ulang secara otomatis untuk menerapkan rotasi kata sandi selama periode pemeliharaan 60 hingga 90 hari.

- Replika baca PostgreSQL bersifat hanya baca. Meskipun replika baca bukan instans DB yang dapat ditulis, Anda dapat mempromosikannya menjadi instans DB RDS for PostgreSQL mandiri. Namun, prosesnya tidak dapat dikembalikan.
- Anda tidak dapat membuat replika baca dari replika baca lain jika instans DB RDS for PostgreSQL Anda menjalankan versi PostgreSQL yang lebih lama dari 14.1. RDS for PostgreSQL mendukung replika baca kaskade pada RDS for PostgreSQL versi 14.1 dan rilis yang lebih tinggi saja. Untuk informasi selengkapnya, lihat Menggunakan replika baca kaskade dengan RDS for PostgreSQL.
- Jika Anda mempromosikan replika baca PostgreSQL, replika baca ini akan menjadi instans DB yang dapat ditulis. Replika baca ini akan berhenti menerima file write-ahead log (WAL) dari instans DB sumber, dan bukan lagi merupakan instans hanya baca. Anda dapat membuat replika baca baru dari instans DB yang dipromosikan seperti halnya instans DB RDS for PostgreSQL apa pun. Untuk informasi selengkapnya, lihat Mempromosikan replika baca menjadi instans DB mandiri.
- Jika Anda mempromosikan replika baca PostgreSQL dari dalam rantai replikasi (serangkaian replika baca kaskade), setiap replika baca hilir yang ada akan terus menerima file WAL dari instans yang dipromosikan secara otomatis. Untuk informasi selengkapnya, lihat Menggunakan replika baca kaskade dengan RDS for PostgreSQL.
- Jika tidak ada transaksi pengguna yang berjalan pada instans DB sumber, replika baca PostgreSQL terkait akan melaporkan lag replikasi hingga lima menit. Lag replika dihitung sebagai `currentTime - lastCommitedTransactionTimestamp`, yang berarti bahwa ketika tidak ada transaksi yang sedang diproses, nilai lag replika akan meningkat untuk jangka waktu tertentu sampai segmen write-ahead log (WAL) beralih. Secara default RDS for PostgreSQL mengganti segmen WAL setiap 5 menit, yang menghasilkan catatan transaksi dan penurunan lag yang dilaporkan.
- Anda tidak dapat mengaktifkan pencadangan otomatis untuk replika baca PostgreSQL untuk versi RDS for PostgreSQL yang lebih lama dari 14.1. Pencadangan otomatis untuk replika baca didukung untuk RDS for PostgreSQL 14.1 dan versi yang lebih tinggi saja. Untuk RDS for PostgreSQL 13 dan versi yang lebih lama, buat snapshot dari replika baca jika Anda menginginkan cadangannya.

- oint-in-time Pemulihan P (PITR) tidak didukung untuk replika baca. Anda dapat menggunakan PITR dengan instans primer (penulis) saja, bukan replika baca. Untuk mempelajari selengkapnya, lihat Memulihkan instans DB dengan waktu yang ditentukan.

Konfigurasi replika baca dengan PostgreSQL

RDS for PostgreSQL menggunakan replikasi streaming native PostgreSQL untuk membuat salinan hanya baca instans DB sumber. Instans DB replika baca ini adalah replika fisik yang dibuat secara asinkron dari instans DB sumber. Instans ini dibuat oleh koneksi khusus yang mentransmisikan data write ahead log (WAL) dari instans DB sumber ke replika baca. Untuk informasi selengkapnya, lihat Streaming Replication dalam dokumentasi PostgreSQL.

PostgreSQL secara asinkron mengalirkan perubahan basis data ke koneksi aman ini saat perubahan tersebut dibuat pada instans DB sumber. Anda dapat mengenkripsi komunikasi dari aplikasi klien Anda ke instans DB sumber atau replika baca apa pun dengan mengatur parameter `ssl` ke `1`. Untuk informasi selengkapnya, lihat Menggunakan SSL dengan instans DB PostgreSQL.

PostgreSQL menggunakan peran replikasi untuk melakukan replikasi streaming. Peran ini memiliki hak akses, tetapi Anda tidak dapat menggunakannya untuk mengubah data apa pun. PostgreSQL menggunakan proses tunggal untuk menangani replikasi.

Anda dapat membuat replika baca PostgreSQL tanpa memengaruhi operasi atau pengguna instans DB sumber. Amazon RDS menetapkan parameter dan izin yang diperlukan untuk Anda, pada instans DB sumber dan replika baca, tanpa memengaruhi layanan. Snapshot diambil dari instans DB sumber, dan snapshot ini digunakan untuk membuat replika baca. Jika Anda menghapus replika baca pada suatu waktu di masa mendatang, tidak ada pemadaman yang akan terjadi.

Anda dapat membuat hingga 15 replika baca dari satu instans DB sumber di Wilayah yang sama. Di RDS for PostgreSQL 14.1, Anda juga dapat membuat hingga tiga tingkat replika baca dalam rantai (kaskade) dari satu instans DB sumber. Untuk informasi selengkapnya, lihat Menggunakan replika baca kaskade dengan RDS for PostgreSQL. Dalam semua kasus, instans DB sumber perlu memiliki pencadangan otomatis yang dikonfigurasi. Anda melakukannya dengan mengatur periode retensi cadangan pada instans DB Anda ke nilai apa pun selain 0. Untuk informasi selengkapnya, lihat Membuat replika baca.

Anda dapat membuat replika baca untuk RDS Anda untuk instans PostgreSQL DB sama dengan instans DB sumber Anda. Wilayah AWS Hal ini dikenal sebagai replikasi dalam Wilayah. Anda juga dapat membuat replika baca berbeda Wilayah AWS dari instance DB sumber. Hal ini dikenal sebagai

replikasi lintas Wilayah. Untuk informasi selengkapnya tentang menyiapkan replika baca lintas Wilayah, lihat Membuat replika baca di tempat yang berbeda Wilayah AWS. Berbagai mekanisme yang mendukung proses replikasi untuk dalam Wilayah dan lintas Wilayah sedikit berbeda tergantung pada versi RDS for PostgreSQL seperti yang dijelaskan dalam Cara kerja replikasi streaming untuk berbagai versi RDS for PostgreSQL.

Agar replikasi beroperasi secara efektif, setiap replika baca harus memiliki jumlah sumber daya komputasi dan penyimpanan yang sama seperti instans DB sumber. Jika Anda menskalakan instans DB sumber, pastikan untuk juga menskalakan replika baca.

Amazon RDS mengganti parameter yang tidak kompatibel pada replika baca jika parameter tersebut mencegah replika baca dimulai. Misalnya, anggaplah nilai parameter `max_connections` pada instans DB sumber lebih tinggi daripada replika baca. Dalam hal ini, Amazon RDS memperbarui nilai parameter pada replika baca agar sama dengan nilai pada instans DB sumber.

Replika baca RDS for PostgreSQL memiliki akses ke basis data eksternal yang tersedia melalui foreign data wrapper (FDW) pada instans DB sumber. Misalnya, anggaplah instans DB RDS for PostgreSQL Anda menggunakan wrapper `mysql_fdw` untuk mengakses data dari RDS for MySQL. Jika demikian, replika baca Anda juga dapat mengakses data tersebut. FDW lain yang didukung termasuk `oracle_fdw`, `postgres_fdw`, dan `tds_fdw`. Untuk informasi selengkapnya, lihat Bekerja dengan pembungkus data asing yang didukung untuk Amazon RDS for PostgreSQL.

Menggunakan replika baca RDS for PostgreSQL dengan konfigurasi Multi-AZ

Anda dapat membuat replika baca dari instans DB AZ Tunggal atau Multi-AZ. Anda dapat menggunakan deployment Multi-AZ untuk meningkatkan durabilitas dan ketersediaan data kritis, dengan replika siaga. Replika siaga adalah replika baca khusus yang dapat mengambil alih beban kerja jika DB sumber melakukan failover. Anda tidak dapat menggunakan replika siaga Anda untuk melayani lalu lintas baca. Namun, Anda dapat membuat replika baca dari instans DB Multi-AZ yang memiliki lalu lintas tinggi untuk mengalihkan kueri hanya baca. Untuk mempelajari selengkapnya tentang deployment Multi-AZ, lihat Deployment instans DB Multi-AZ.

Jika instans DB sumber dari deployment Multi-AZ melakukan failover ke replika siaga, replika baca terkait akan beralih menggunakan replika siaga (sekarang menjadi replika primer) sebagai sumber replikasinya. Replika baca mungkin perlu diaktifkan ulang, tergantung pada versi RDS for PostgreSQL sebagai berikut:

- PostgreSQL 13 dan versi yang lebih tinggi – Pengaktifan ulang tidak diperlukan. Replika baca secara otomatis disinkronkan dengan replika primer baru. Namun, dalam beberapa kasus,

aplikasi klien Anda mungkin meng-cache detail Layanan Nama Domain (DNS) untuk replika baca Anda. Jika demikian, atur nilai time-to-live (TTL) menjadi kurang dari 30 detik. Tindakan ini akan mencegah replika baca mempertahankan alamat IP yang sudah tidak berlaku (dan dengan demikian, mencegah replika baca ini disinkronkan dengan replika primer baru). Untuk mempelajari selengkapnya tentang hal ini dan praktik terbaik lainnya, lihat Pedoman operasional dasar Amazon RDS.

- PostgreSQL 12 dan semua versi yang lebih lama – Replika baca diaktifkan ulang secara otomatis setelah failover ke replika siaga karena replika siaga (sekarang menjadi replika primer) memiliki alamat IP dan nama instans yang berbeda. Pengaktifan ulang ini akan menyinkronkan replika baca dengan replika primer baru.

Untuk mempelajari selengkapnya tentang failover, lihat Proses failover untuk Amazon RDS. Untuk mempelajari selengkapnya tentang cara kerja replika baca dalam deployment Multi-AZ, lihat Menggunakan replika baca instans DB.

Untuk memberikan dukungan failover untuk replika baca, Anda dapat membuat replika baca sebagai instans DB Multi-AZ sehingga Amazon RDS akan membuat replika siaga Anda di Zona Ketersediaan (AZ) lain. Pembuatan replika baca Anda sebagai instans DB Multi-AZ tidak tergantung pada apakah basis data sumber merupakan instans DB Multi-AZ.

Menggunakan replika baca kaskade dengan RDS for PostgreSQL

Mulai dari versi 14.1, RDS for PostgreSQL mendukung replika baca kaskade. Dengan replika baca kaskade, Anda dapat menskalakan pembacaan tanpa menambahkan overhead ke instans DB RDS for PostgreSQL sumber Anda. Pembaruan pada log WAL tidak dikirim oleh instans DB sumber ke setiap replika baca. Sebagai gantinya, setiap replika baca dalam rangkaian kaskade akan mengirimkan pembaruan log WAL ke replika baca berikutnya dalam rangkaian tersebut. Hal ini akan mengurangi beban pada instans DB sumber.

Dengan replika baca kaskade, instans DB RDS for PostgreSQL Anda akan mengirimkan data WAL ke replika baca pertama dalam rantai. Replika baca tersebut kemudian mengirimkan data WAL ke replika kedua dalam rantai, dan seterusnya. Hasil akhirnya adalah bahwa semua replika baca dalam rantai memiliki perubahan dari instans DB RDS for PostgreSQL, tetapi overhead-nya tidak hanya berada pada instans DB sumber.

Anda dapat membuat rangkaian yang terdiri dari maksimal tiga replika baca dalam rantai dari instans DB RDS for PostgreSQL sumber. Misalnya, anggaplah bahwa Anda memiliki instans DB RDS for PostgreSQL 14.1, yaitu `rpg-db-main`. Anda dapat melakukan hal berikut:

- Dimulai dengan `rpg-db-main`, buat replika baca pertama dalam rantai, `read-replica-1`.
- Selanjutnya, dari `read-replica-1`, buat replika baca berikutnya dalam rantai, `read-replica-2`.
- Akhirnya, dari `read-replica-2`, buat replika baca ketiga dalam rantai, `read-replica-3`.

Anda tidak dapat membuat replika baca lain di luar replika baca kaskade ketiga ini dalam rangkaian untuk `rpg-db-main`. Rangkaian lengkap instans dari instans DB sumber RDS for PostgreSQL hingga akhir rangkaian replika baca kaskade dapat terdiri dari maksimal empat instans DB.

Agar replika baca kaskade berfungsi, aktifkan pencadangan otomatis pada RDS for PostgreSQL Anda. Buat replika baca terlebih dahulu lalu aktifkan pencadangan otomatis pada instans DB RDS for PostgreSQL. Proses ini sama dengan mesin DB Amazon RDS lainnya. Untuk informasi selengkapnya, lihat Membuat replika baca.

Seperti halnya replika baca lainnya, Anda dapat mempromosikan replika baca yang merupakan bagian dari kaskade. Jika replika baca dipromosikan dari dalam rantai replika baca, replika baca ini akan dihapus dari rantai tersebut. Misalnya, anggaplah Anda ingin memindahkan sebagian beban kerja dari instans DB `rpg-db-main` Anda ke instans baru yang akan digunakan oleh departemen akuntansi saja. Berdasarkan rantai tiga replika baca dari contoh, Anda memutuskan untuk mempromosikan `read-replica-2`. Rantai terpengaruh sebagai berikut:

- Mempromosikan `read-replica-2` menghapusnya dari rantai replikasi.
  - Replika ini sekarang menjadi instans DB baca/tulis penuh.
  - Replika ini terus mereplikasi menjadi `read-replica-3`, seperti yang dilakukan sebelum promosi.
- `rpg-db-main` Anda terus mereplikasi ke `read-replica-1`.

Untuk informasi selengkapnya tentang mempromosikan replika baca, lihat Mempromosikan replika baca menjadi instans DB mandiri.

> **Note**
>
> Untuk replika baca kaskade, RDS for PostgreSQL mendukung 15 replika baca untuk setiap instans DB sumber pada replikasi tingkat pertama, dan 5 replika baca untuk setiap instans DB sumber pada tingkat replikasi kedua dan ketiga.

Cara kerja replikasi streaming untuk berbagai versi RDS for PostgreSQL

Seperti dibahas dalam Konfigurasi replika baca dengan PostgreSQL, RDS for PostgreSQL menggunakan protokol replikasi streaming native PostgreSQL untuk mengirim data WAL dari instans DB sumber. Layanan ini mengirimkan data WAL sumber ke replika baca untuk replika baca dalam Wilayah dan lintas Wilayah. Dengan versi 9.4, PostgreSQL memperkenalkan slot replikasi fisik sebagai mekanisme pendukung untuk proses replikasi.

Slot replikasi fisik mencegah instans DB sumber menghapus data WAL sebelum dikonsumsi oleh semua replika baca. Setiap replika baca memiliki slot fisiknya sendiri pada instans DB sumber. Slot melacak WAL terlama (berdasarkan nomor urutan logis, LSN) yang mungkin diperlukan oleh replika. Setelah semua slot dan koneksi DB telah berkembang melampaui WAL (LSN) tertentu, LSN tersebut akan menjadi kandidat yang akan dihapus di checkpoint berikutnya.

Amazon RDS menggunakan Amazon S3 untuk mengarsipkan data WAL. Untuk replika baca dalam Wilayah, Anda dapat menggunakan data yang diarsipkan ini untuk memulihkan replika baca jika diperlukan. Contoh situasi saat Anda mungkin perlu melakukannya adalah jika koneksi antara DB sumber dan replika baca terputus karena alasan apa pun.

Dalam tabel berikut, Anda dapat menemukan ringkasan perbedaan antara versi PostgreSQL dan mekanisme pendukung untuk dalam Wilayah dan lintas Wilayah yang digunakan oleh RDS for PostgreSQL.

Dalam Wilayah	Lintas Wilayah
PostgreSQL 14.1 and higher versions	
• Slot replikasi • Arsip Amazon S3	• Slot replikasi
PostgreSQL 13 and lower versions	
• Arsip Amazon S3	• Slot replikasi

Untuk informasi selengkapnya, lihat Memantau dan menyetel proses replikasi.

Memahami parameter yang mengontrol replikasi PostgreSQL

Parameter berikut memengaruhi proses replikasi dan menentukan seberapa baik replika baca dalam mengikuti instans DB sumber:

max_wal_senders

Parameter `max_wal_senders` menentukan jumlah maksimum koneksi yang dapat didukung oleh instans DB sumber pada saat yang sama melalui protokol replikasi streaming. Default untuk RDS for PostgreSQL 13 dan rilis yang lebih tinggi adalah 20. Parameter ini harus diatur sedikit lebih tinggi dari jumlah replika baca sebenarnya. Jika parameter ini diatur terlalu rendah untuk jumlah replika baca, replikasi akan berhenti.

Untuk informasi selengkapnya, lihat max_wal_senders dalam dokumentasi PostgreSQL.

wal_keep_segments

Parameter `wal_keep_segments` menentukan jumlah file write-ahead log (WAL) yang dipertahankan oleh instans DB sumber di direktori `pg_wal`. Pengaturan default adalah 32.

Jika `wal_keep_segments` tidak diatur ke nilai yang cukup besar untuk deployment Anda, replika baca dapat tertinggal jauh sehingga replikasi streaming berhenti. Jika itu terjadi, Amazon RDS akan menghasilkan kesalahan replikasi dan memulai pemulihan pada replika baca. Layanan ini melakukannya dengan memutar ulang data WAL yang diarsipkan milik instans DB sumber dari Amazon S3. Proses pemulihan ini berlanjut sampai replika baca tersebut dapat melanjutkan replikasi streaming. Anda dapat melihat proses ini dalam praktiknya seperti yang dicatat oleh log PostgreSQL dalam Contoh: Bagaimana replika baca pulih dari interupsi replikasi.

> **Note**
>
> Di PostgreSQL versi 13, parameter `wal_keep_segments` diberi nama `wal_keep_size`. Hal ini memiliki fungsi yang sama seperti `wal_keep_segments`, tetapi nilai default-nya menggunakan megabyte (MB) (2048 MB), bukan jumlah file. Untuk informasi selengkapnya, lihat wal_keep_segments dan wal_keep_size dalam dokumentasi PostgreSQL.

max_slot_wal_keep_size

Parameter `max_slot_wal_keep_size` mengontrol kuantitas data WAL yang dipertahankan oleh DB RDS for PostgreSQL dalam direktori `pg_wal` untuk melayani slot. Parameter ini

digunakan untuk konfigurasi yang menggunakan slot replikasi. Nilai default untuk parameter ini adalah `-1`, artinya tidak ada batasan berapa banyak data WAL yang dipertahankan pada instans DB sumber. Untuk informasi tentang cara memantau slot replikasi Anda, lihat Memantau slot replikasi untuk instans DB RDS for PostgreSQL Anda.

Untuk informasi selengkapnya tentang parameter ini, lihat max_slot_wal_keep_size dalam dokumentasi PostgreSQL.

Setiap kali stream yang menyediakan data WAL ke replika baca terputus, PostgreSQL akan beralih ke mode pemulihan. Ini mengembalikan replika baca dengan menggunakan data WAL yang diarsipkan dari Amazon S3 atau dengan menggunakan data WAL yang terkait dengan slot replikasi. Saat proses ini selesai, PostgreSQL membuat ulang replikasi streaming.

Contoh: Bagaimana replika baca pulih dari interupsi replikasi

Dalam contoh berikut, Anda akan menemukan detail log yang menunjukkan proses pemulihan untuk replika baca. Contohnya adalah dari RDS untuk instance PostgreSQL DB yang menjalankan PostgreSQL versi 12.9 sama dengan DB sumber, jadi slot replikasi tidak digunakan. Wilayah AWS Proses pemulihannya sama untuk instans DB RDS for PostgreSQL lain yang menjalankan PostgreSQL yang lebih lama dari versi 14.1 dengan replika baca dalam Wilayah.

Ketika replika baca kehilangan kontak dengan instans DB sumber, Amazon RDS mencatat masalahnya di log sebagai pesan `FATAL: could not receive data from WAL stream`, bersama dengan `ERROR: requested WAL segment ... has already been removed`. Seperti yang ditunjukkan pada baris tebal, Amazon RDS memulihkan replika dengan memutar ulang file WAL yang diarsipkan.

```
2014-11-07 19:01:10 UTC::@:[23180]:DEBUG:  switched WAL source from archive to stream
 after failure
2014-11-07 19:01:10 UTC::@:[11575]:LOG: started streaming WAL from primary at 1A/
D3000000 on timeline 1
2014-11-07 19:01:10 UTC::@:[11575]:FATAL: could not receive data from WAL stream:
ERROR:  requested WAL segment 000000010000001A000000D3 has already been removed
2014-11-07 19:01:10 UTC::@:[23180]:DEBUG: could not restore file "00000002.history"
 from archive: return code 0
2014-11-07 19:01:15 UTC::@:[23180]:DEBUG: switched WAL source from stream to archive
 after failure recovering 000000010000001A000000D3
2014-11-07 19:01:16 UTC::@:[23180]:LOG:  restored log file "000000010000001A000000D3"
 from archive
```

Saat Amazon RDS memutar ulang data WAL yang diarsipkan pada replika untuk mengejar ketinggalan, streaming ke replika baca dimulai lagi. Saat streaming dilanjutkan, Amazon RDS menulis entri ke file log seperti yang berikut ini.

```
2014-11-07 19:41:36 UTC::@:[24714]:LOG:started streaming WAL from primary at 1B/
B6000000 on timeline 1
```

Mengatur parameter yang mengontrol memori bersama

Parameter yang Anda tetapkan menentukan ukuran memori bersama untuk melacak ID transaksi, kunci, dan transaksi yang disiapkan. Struktur memori bersama instans siaga harus sama atau lebih besar dari instans primer. Hal ini memastikan bahwa instans siaga tidak kehabisan memori bersama selama pemulihan. Jika nilai parameter pada replika kurang dari nilai parameter pada instans primer, Amazon RDS akan secara otomatis menyesuaikan parameter replika dan mengaktifkan ulang mesin.

Parameter yang terpengaruh adalah:

- max_connections
- max_worker_processes
- max_wal_senders
- max_prepared_transactions
- max_locks_per_transaction

Untuk menghindari boot ulang replika RDS karena memori yang tidak mencukupi, kami sarankan untuk menerapkan perubahan parameter sebagai boot ulang bergulir pada setiap replika. Anda harus menerapkan aturan berikut saat Anda mengatur parameter:

- Meningkatkan nilai parameter:
  - Anda harus selalu meningkatkan nilai parameter dari semua replika baca terlebih dahulu, dan melakukan boot ulang bergulir pada semua replika. Kemudian, terapkan perubahan parameter pada instans primer dan boot ulang.
- Menurunkan nilai parameter:
  - Anda harus terlebih dahulu mengurangi nilai parameter instans primer dan melakukan boot ulang. Kemudian, terapkan perubahan parameter ke semua replika baca terkait dan lakukan boot ulang bergulir.

Memantau dan menyetel proses replikasi

Kami sangat menyarankan agar Anda secara rutin memantau instans DB dan replika baca RDS for PostgreSQL Anda. Anda perlu memastikan bahwa replika baca Anda mengikuti perubahan pada instans DB sumber. Amazon RDS secara transparan memulihkan replika baca Anda saat terjadi gangguan pada proses replikasi. Namun, yang terbaik adalah menghindari kebutuhan pemulihan sama sekali. Pemulihan menggunakan slot replikasi akan lebih cepat daripada menggunakan arsip Amazon S3, tetapi proses pemulihan apa pun dapat memengaruhi performa baca.

Untuk menentukan seberapa baik replika baca Anda dalam mengikuti instans DB sumber, Anda dapat melakukan hal berikut:

- Periksa jumlah **`ReplicaLag`** antara instans DB sumber dan replika. Lag replika adalah jumlah waktu, dalam hitungan detik, untuk ketertinggalan replika baca dari instans DB sumbernya. Metrik ini melaporkan hasil dari kueri berikut.

  ```
  SELECT extract(epoch from now() - pg_last_xact_replay_timestamp()) AS "ReplicaLag";
  ```

  Lag replika adalah indikasi seberapa baik replika baca dalam mengikuti instans DB sumber. Lag replika adalah jumlah latensi antara instans DB sumber dan instans baca tertentu. Nilai tinggi untuk lag replika dapat menunjukkan ketidakcocokan antara kelas instans DB atau jenis penyimpanan (atau keduanya) yang digunakan oleh instans DB sumber dan replika bacanya. Kelas instans DB dan jenis penyimpanan untuk instans DB sumber dan semua replika baca harus sama.

  Lag replika juga dapat diakibatkan oleh masalah koneksi intermiten. Anda dapat memantau kelambatan replikasi di Amazon CloudWatch dengan melihat metrik Amazon RDS. `ReplicaLag` Untuk mempelajari selengkapnya tentang `ReplicaLag` dan metrik lainnya untuk Amazon RDS, lihat CloudWatch Metrik Amazon untuk Amazon RDS.

- Periksa log PostgreSQL untuk menemukan informasi yang dapat Anda gunakan untuk menyesuaikan pengaturan Anda. Di setiap checkpoint, log PostgreSQL mengambil jumlah file log transaksi yang didaur ulang, seperti yang ditunjukkan pada contoh berikut.

  ```
  2014-11-07 19:59:35 UTC::@:[26820]:LOG:  checkpoint complete: wrote 376 buffers
   (0.2%);
  0 transaction log file(s) added, 0 removed, 1 recycled; write=35.681 s, sync=0.013 s,
   total=35.703 s;
  sync files=10, longest=0.013 s, average=0.001 s
  ```

Anda dapat menggunakan informasi ini untuk mengetahui berapa banyak file transaksi yang didaur ulang dalam periode waktu tertentu. Anda kemudian dapat mengubah pengaturan `wal_keep_segments` jika perlu. Misalnya, anggaplah log PostgreSQL di `checkpoint complete` menampilkan `35 recycled` selama interval 5 menit. Dalam hal ini, `wal_keep_segments` dengan nilai default 32 tidak cukup untuk mengimbangi aktivitas streaming, jadi Anda harus meningkatkan nilai parameter ini.

- Gunakan Amazon CloudWatch untuk memantau metrik yang dapat memprediksi masalah replikasi. Daripada menganalisis log PostgreSQL secara langsung, Anda dapat menggunakan CloudWatch Amazon untuk memeriksa metrik yang telah dikumpulkan. Misalnya, Anda dapat memeriksa nilai metrik `TransactionLogsGeneration` untuk melihat berapa banyak data WAL yang dihasilkan oleh instans DB sumber. Dalam beberapa kasus, beban kerja pada instans DB Anda mungkin menghasilkan sejumlah besar data WAL. Jika demikian, Anda mungkin perlu mengubah kelas instans DB untuk instans DB sumber dan replika baca Anda. Penggunaan kelas instans dengan performa jaringan tinggi (10 Gbps) dapat mengurangi lag replika.

Memantau slot replikasi untuk instans DB RDS for PostgreSQL Anda

Semua versi RDS for PostgreSQL menggunakan slot replikasi untuk replika baca lintas Wilayah. RDS for PostgreSQL 14.1 dan versi yang lebih tinggi menggunakan slot replikasi untuk replika baca dalam Wilayah. Replika baca dalam Wilayah juga menggunakan Amazon S3 untuk mengarsipkan data WAL. Dengan kata lain, jika instans DB dan replika baca Anda menjalankan PostgreSQL 14.1 atau lebih tinggi, slot replikasi dan arsip Amazon S3 keduanya tersedia untuk memulihkan replika baca. Memulihkan replika baca menggunakan slot replikasi lebih cepat daripada memulihkan dari arsip Amazon S3. Jadi, kami menyarankan Anda memantau slot replikasi dan metrik terkait.

Anda dapat melihat slot replikasi pada instans DB RDS for PostgreSQL Anda dengan mengueri tampilan `pg_replication_slots`, sebagai berikut.

```
postgres=> SELECT * FROM pg_replication_slots;
slot_name                  | plugin | slot_type | datoid | database | temporary |
 active | active_pid | xmin | catalog_xmin | restart_lsn | confirmed_flush_lsn |
 wal_status | safe_wal_size | two_phase
---------------------------+--------+-----------+--------+----------+-----------
+--------+------------+------+--------------+-------------+---------------------
+------------+--------------+-----------
rds_us_west_1_db_555555555 |        | physical  |        |          | f         | t
  |      13194 |      |              | 23/D8000060 |                     | reserved   |
               | f
```

```
(1 row)
```

`wal_status` dengan nilai `reserved` berarti bahwa jumlah data WAL yang dipegang oleh slot berada dalam batas-batas parameter `max_wal_size`. Dengan kata lain, slot replikasi berukuran sesuai. Kemungkinan nilai lainnya adalah sebagai berikut:

- `extended` – Slot melebihi pengaturan `max_wal_size`, tetapi data WAL dipertahankan.
- `unreserved` – Slot tidak lagi memiliki semua data WAL yang diperlukan. Beberapa di antaranya akan dihapus di checkpoint berikutnya.
- `lost` – Beberapa data WAL yang diperlukan telah dihapus. Slot tidak lagi dapat digunakan.

Keadaan `unreserved` dan `lost` keadaan hanya `wal_status` terlihat ketika `max_slot_wal_keep_size` tidak negatif.

Tampilan `pg_replication_slots` menunjukkan status slot replikasi Anda saat ini. Untuk menilai kinerja slot replikasi Anda, Anda dapat menggunakan Amazon CloudWatch dan memantau metrik berikut:

- **`OldestReplicationSlotLag`** – Menampilkan daftar slot yang memiliki lag paling banyak, yaitu yang paling tertinggal dari replika primer. Lag ini dapat dikaitkan dengan replika baca, tetapi juga koneksi.
- **`TransactionLogsDiskUsage`** – Menunjukkan berapa banyak penyimpanan yang digunakan untuk data WAL. Ketika replika baca mengalami lag yang signifikan, nilai metrik ini dapat meningkat secara substansial.

Untuk mempelajari selengkapnya tentang menggunakan Amazon CloudWatch dan metriknya untuk RDS untuk PostgreSQL, lihat. Memantau metrik Amazon RDS dengan Amazon CloudWatch Untuk informasi selengkapnya tentang memantau replikasi streaming pada instans DB RDS for PostgreSQL Anda, lihat Praktik terbaik untuk replikasi Amazon RDS PostgreSQL di Blog Basis Data AWS .

Pemecahan masalah untuk RDS untuk PostgreSQL baca replika

Berikut ini, Anda dapat menemukan ide pemecahan masalah untuk beberapa RDS umum untuk masalah replika baca PostgreSQL.

Mengakhiri kueri yang menyebabkan kelambatan replika baca

Transaksi baik dalam keadaan aktif atau idle dalam keadaan transaksi yang berjalan untuk waktu yang lama dalam database dapat mengganggu proses replikasi WAL, sehingga meningkatkan kelambatan replikasi. Oleh karena itu, pastikan untuk memantau runtime transaksi ini dengan tampilan `pg_stat_activity` PostgreSQL.

Jalankan kueri pada instance utama yang mirip dengan berikut ini untuk menemukan ID proses (PID) kueri yang berjalan untuk waktu yang lama:

```
SELECT datname, pid,usename, client_addr, backend_start,
xact_start, current_timestamp - xact_start AS xact_runtime, state,
backend_xmin FROM pg_stat_activity WHERE state='active';
```

```
SELECT now() - state_change as idle_in_transaction_duration, now() - xact_start as
 xact_duration,*
FROM  pg_stat_activity
WHERE state  = 'idle in transaction'
AND   xact_start is not null
ORDER BY 1 DESC;
```

Setelah mengidentifikasi PID kueri, Anda dapat memilih untuk mengakhiri kueri.

Jalankan kueri pada instance utama yang mirip dengan berikut ini untuk menghentikan kueri yang berjalan dalam waktu lama:

```
SELECT pg_terminate_backend(PID);
```

Meningkatkan performa kueri untuk RDS for PostgreSQL dengan Amazon RDS Optimized Reads

Anda dapat mencapai pemrosesan kueri yang lebih cepat untuk RDS for PostgreSQL dengan Amazon RDS Optimized Reads. Instans DB RDS for PostgreSQL atau klaster DB Multi-AZ yang menggunakan RDS Optimized Reads dapat mencapai pemrosesan kueri hingga 50% lebih cepat dibandingkan dengan yang tidak menggunakannya.

Topik

Ikhtisar RDS Optimized Reads di PostgreSQL

Optimized Reads tersedia secara default di RDS for PostgreSQL versi 15.2 dan lebih baru, 14.7 dan lebih baru, serta 13.10 dan lebih baru.

Saat Anda menggunakan instans DB RDS for PostgreSQL atau klaster DB Multi-AZ dengan RDS Optimized Reads diaktifkan, performa kueri yang 50% lebih cepat tercapai menggunakan penyimpanan tingkat blok solid state drive (SSD) berbasis Non-Volatile Memory Express (NVMe) lokal. Anda dapat mencapai pemrosesan kueri yang lebih cepat dengan menempatkan tabel sementara yang dihasilkan oleh PostgreSQL di penyimpanan lokal, yang akan mengurangi lalu lintas ke Elastic Block Storage (EBS) melalui jaringan.

Di PostgreSQL, objek sementara ditetapkan ke namespace sementara yang menurun secara otomatis di akhir sesi. Saat menurun, namespace sementara menghapus objek apa pun yang bergantung pada sesi, termasuk objek yang memenuhi syarat skema, seperti tabel, fungsi, operator, atau bahkan ekstensi.

Di RDS for PostgreSQL, parameter `temp_tablespaces` dikonfigurasi untuk area kerja sementara ini di mana objek sementara disimpan.

Kueri berikut mengembalikan nama tablespace dan lokasinya.

```
postgres=> show temp_tablespaces;
temp_tablespaces
---------------------
rds_temp_tablespace
(1 row)
```

`rds_temp_tablespace` adalah tablespace yang dikonfigurasi oleh RDS yang menunjuk ke penyimpanan lokal NVMe. Anda selalu dapat beralih kembali ke penyimpanan Amazon EBS dengan memodifikasi parameter ini di `Parameter group` menggunakan AWS Management Console untuk menunjuk ke tablespace apa pun selain `rds_temp_tablespace`. Untuk informasi lebih lanjut, lihat Memodifikasi parameter dalam grup parameter DB. Anda juga dapat menggunakan perintah SET untuk memodifikasi nilai parameter `temp_tablespaces` menjadi `pg_default` di tingkat sesi menggunakan perintah SET. Memodifikasi parameter akan mengalihkan area kerja sementara ke Amazon EBS. Peralihan kembali ke Amazon EBS akan membantu ketika penyimpanan lokal untuk klaster atau instans RDS Anda tidak cukup untuk melakukan operasi SQL tertentu.

```
postgres=> SET temp_tablespaces TO 'pg_default';
SET
```

```
postgres=> show temp_tablespaces;

 temp_tablespaces
------------------
 pg_default
```

Kasus penggunaan untuk RDS Optimized Reads

Berikut ini adalah beberapa kasus penggunaan yang dapat memperoleh manfaat dari Optimized Reads:

- Kueri analitis yang mencakup Ekspresi Tabel Umum (CTE), tabel turunan, dan operasi pengelompokan.
- Replika baca yang menangani kueri yang tidak dioptimalkan untuk aplikasi.
- Kueri pelaporan sesuai permintaan atau dinamis dengan operasi kompleks seperti GROUP BY dan ORDER BY yang tidak selalu dapat menggunakan indeks yang sesuai.

- Beban kerja lain yang menggunakan tabel sementara internal.
- `CREATE INDEX`atau `REINDEX` operasi untuk menyortir.

Praktik terbaik untuk RDS Optimized Reads

Gunakan praktik terbaik berikut untuk RDS Optimized Reads:

- Tambahkan logika coba lagi untuk kueri hanya baca jika gagal karena penyimpanan instans sudah penuh selama pelaksanaan.
- Pantau ruang penyimpanan yang tersedia di penyimpanan instans dengan CloudWatch metrik`FreeLocalStorage`. Jika penyimpanan instans mencapai batasnya karena beban kerja pada instans DB atau klaster DB Multi-AZ, modifikasi untuk menggunakan kelas instans DB yang lebih besar.

Menggunakan RDS Optimized Reads

Saat Anda menyediakan instans DB RDS for PostgreSQL dengan salah satu kelas instans DB berbasis NVMe dalam deployment instans DB satu AZ, deployment instans DB Multi-AZ, atau deployment klaster DB Multi-AZ, instans DB secara otomatis menggunakan RDS Optimized Reads.

Untuk informasi lebih lanjut tentang deployment Multi-AZ, lihat Mengonfigurasi dan mengelola deployment Multi-AZ.

Untuk mengaktifkan RDS Optimized Reads, lakukan salah satu tindakan berikut ini:

- Buat instans DB RDS for PostgreSQL atau klaster DB Multi-AZ menggunakan salah satu kelas instans berbasis NVMe. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS.
- Modifikasi instans DB RDS for PostgreSQL atau klaster DB Multi-AZ yang sudah ada untuk menggunakan salah satu kelas instans DB berbasis NVMe. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.

RDS Optimized Reads tersedia di semua Wilayah AWS di mana satu atau lebih kelas instans DB dengan penyimpanan SSD NVMe lokal didukung. Untuk informasi selengkapnya, lihat Kelas instans DB .

Untuk beralih kembali ke instans RDS baca yang tidak dioptimalkan, modifikasi kelas instans DB dari klaster atau instans RDS Anda menjadi kelas instans serupa yang hanya mendukung

penyimpanan EBS untuk beban kerja basis data Anda. Misalnya, jika kelas instans DB saat ini adalah db.r6gd.4xlarge, pilih db.r6g.4xlarge untuk beralih kembali. Untuk informasi lebih lanjut, lihat Memodifikasi instans DB Amazon RDS.

Memantau instans DB yang menggunakan RDS Optimized Reads

Anda dapat memantau instans DB yang menggunakan Bacaan yang Dioptimalkan RDS menggunakan metrik berikut: CloudWatch

- `FreeLocalStorage`
- `ReadIOPSLocalStorage`
- `ReadLatencyLocalStorage`
- `ReadThroughputLocalStorage`
- `WriteIOPSLocalStorage`
- `WriteLatencyLocalStorage`
- `WriteThroughputLocalStorage`

Metrik ini menyediakan data tentang penyimpanan instans, IOPS, dan throughput yang tersedia. Untuk informasi selengkapnya tentang metrik ini, lihat Metrik CloudWatch tingkat instans Amazon untuk Amazon RDS.

Untuk memantau penggunaan penyimpanan lokal Anda saat ini, masuk ke basis data Anda menggunakan kueri berikut:

```
SELECT
    spcname AS "Name",
    pg_catalog.pg_size_pretty(pg_catalog.pg_tablespace_size(oid)) AS "size"
FROM
    pg_catalog.pg_tablespace
WHERE
    spcname IN ('rds_temp_tablespace');
```

Untuk informasi selengkapnya tentang file sementara dan penggunaannya, lihat Mengelola file sementara dengan PostgreSQL.

Batasan untuk RDS Optimized Reads di PostgreSQL

Batasan berikut berlaku untuk RDS Optimized Reads di PostgreSQL:

- Transaksi dapat gagal ketika penyimpanan instans penuh.

Mengimpor data ke PostgreSQL di Amazon RDS

Misalkan Anda memiliki deployment PostgreSQL yang ingin Anda pindahkan ke Amazon RDS. Kompleksitas tugas Anda bergantung pada ukuran basis data Anda dan jenis objek basis data yang Anda transfer. Misalnya, pertimbangkan basis data yang berisi set data dalam urutan gigabyte, bersama dengan prosedur yang tersimpan dan pemicu. Basis data seperti itu akan menjadi lebih rumit dibandingkan basis data sederhana dengan hanya beberapa megabyte data uji dan tanpa pemicu atau prosedur tersimpan.

Kami menyarankan Anda untuk menggunakan alat migrasi basis data PostgreSQL native dalam kondisi berikut:

- Anda memiliki migrasi yang homogen, yaitu Anda bermigrasi dari basis data dengan mesin basis data yang sama dengan basis data target.
- Anda memigrasikan seluruh basis data.
- Alat native memungkinkan Anda untuk memigrasi sistem Anda dengan waktu henti minimal.

Dalam kebanyakan kasus lain, melakukan migrasi basis data menggunakan AWS Database Migration Service (AWS DMS) adalah pendekatan terbaik. AWS DMS dapat memigrasikan basis data tanpa waktu henti dan, untuk sebagian besar mesin basis data, melanjutkan replikasi yang sedang berlangsung sampai Anda siap untuk berpindah ke basis data target. Anda dapat bermigrasi ke mesin basis data yang sama ataupun mesin basis data yang berbeda menggunakan AWS DMS. Jika Anda bermigrasi ke mesin basis data yang berbeda dari basis data sumber, Anda dapat menggunakan AWS Schema Conversion Tool (AWS SCT). Anda menggunakan AWS SCT untuk memigrasi objek skema yang tidak dimigrasi oleh AWS DMS. Untuk informasi selengkapnya tentang AWS DMS, lihat Apa itu AWS Database Migration Service?.

Ubah grup parameter DB Anda untuk menyertakan pengaturan berikut hanya untuk impor Anda. Selain itu, uji juga pengaturan parameter guna menemukan pengaturan yang paling efisien untuk instans DB Anda. Anda juga perlu mengembalikan parameter ini ke ke nilai produksi setelah pengimporan selesai.

Ubah pengaturan instans DB Anda sebagai berikut:

- Nonaktifkan pencadangan instans DB (ubah backup_retention menjadi 0).
- Nonaktifkan Multi-AZ.

Ubah grup parameter DB Anda untuk menyertakan pengaturan berikut. Anda sebaiknya hanya menggunakan pengaturan ini saat mengimpor data. Selain itu, uji juga pengaturan parameter guna menemukan pengaturan yang paling efisien untuk instans DB Anda. Anda juga perlu mengembalikan parameter ini ke ke nilai produksi setelah pengimporan selesai.

Parameter	Nilai yang disarankan saat mengimpor	Deskripsi
`maintenance_work_mem`	524288, 1048576, 2097152 atau 4194304 (dalam KB). Pengaturan ini setara dengan 512 MB, 1 GB, 2 GB, dan 4 GB.	Nilai untuk pengaturan ini bergantung pada ukuran host Anda. Parameter ini digunakan selama pernyataan CREATE INDEX dan setiap perintah paralel dapat menggunakan memori sebanyak ini. Hitung nilai terbaik agar Anda tidak mengatur nilai ini terlalu tinggi dan kehabisan memori.
`max_wal_size`	256 (untuk versi 9.6), 4096 (untuk versi 10 dan yang lebih tinggi)	Ukuran maksimum untuk membiarkan WAL tumbuh selama pemeriksaan otomatis. Meningkatkan parameter ini dapat meningkatkan jumlah waktu yang dibutuhkan untuk pemulihan setelah crash. Parameter ini menggantikan `checkpoint_segments` fntuk PostgreSQL 9.6 dan yang lebih baru. Untuk PostgreSQL versi 9.6, nilai ini dalam unit 16 MB. Untuk versi yang lebih baru, nilainya dalam unit 1 MB. Misalnya, dalam versi 9.6, 128 berarti 128 potongan yang masing-masing berukuran 16 MB. Misalnya, dalam versi 12.4, 2048 berarti 2048 potongan yang masing-masing berukuran 1 MB.
`checkpoint_timeout`	1800	Nilai untuk pengaturan ini memungkinkan rotasi WAL yang lebih jarang.
`synchronous_commit`	Mati	Nonaktifkan pengaturan ini untuk mempercepat penulisan. Menonaktifkan parameter ini dapat meningkatkan risiko kehilangan data jika

Parameter	Nilai yang disarankan saat mengimpor	Deskripsi
		terjadi crash pada server (jangan menonaktfkan FSYNC).
`wal_buffers`	8192	Ini adalah nilai dalam unit 8 KB. Ini juga membantu kecepatan pembuatan WAL Anda
`autovacuum`	0	Nonaktifkan parameter vakum otomatis PostgreSQL saat Anda memuat data agar sumber daya tidak digunakan

Gunakan perintah `pg_dump -Fc` (terkompresi) atau `pg_restore -j` (paralel) dengan pengaturan ini.

> **Note**
>
> Perintah PostgreSQL `pg_dumpall` memerlukan izin super_user yang tidak diberikan saat Anda membuat instans DB, sehingga tidak dapat digunakan untuk mengimpor data.

Topik

- Mengimpor basis data PostgreSQL dari instans Amazon EC2
- Menggunakan perintah \copy untuk mengimpor data ke tabel di instans DB PostgreSQL
- Mengimpor data dari Amazon S3 ke instans DB RDS for PostgreSQL
- Mentranspor basis data PostgreSQL antara instans DB

Mengimpor basis data PostgreSQL dari instans Amazon EC2

Jika Anda memiliki data di server PostgreSQL pada instans Amazon EC2 dan ingin memindahkannya ke instans DB PostgreSQL, Anda dapat menggunakan proses berikut. Daftar berikut menunjukkan langkah-langkah yang harus diambil. Setiap langkah dibahas secara lebih mendetail di bagian berikut.

1. Buat berkas menggunakan pg_dump yang berisi data yang akan dimuat

2. Buat instans DB target
3. Gunakan psql untuk membuat basis data pada instans DB dan muat data
4. Buat snapshot DB dari instans DB

Langkah 1: Buat file menggunakan pg_dump yang berisi data yang akan dimuat

Utilitas pg_dump menggunakan perintah COPY untuk membuat skema dan dump data dari basis data PostgreSQL. Skrip dump yang dibuat oleh pg_dump memuat data ke dalam basis data dengan nama yang sama dan membuat ulang tabel, indeks, dan kunci asing. Anda dapat menggunakan perintah pg_restore dan parameter -d untuk memulihkan data ke basis data dengan nama yang berbeda.

Sebelum Anda membuat dump data, Anda harus melakukan kueri tabel yang akan di-dump untuk mendapatkan jumlah baris sehingga Anda dapat mengonfirmasi jumlah pada instans DB target.

Perintah berikut membuat file dump bernama mydb2dump.sql untuk basis data bernama mydb2.

```
prompt>pg_dump dbname=mydb2 -f mydb2dump.sql
```

Langkah 2: Buat instans DB target

Buat instans DB PostgreSQL target menggunakan konsol Amazon RDS, AWS CLI, atau API. Buat instans dengan pengaturan retensi cadangan yang diatur ke 0 dan nonaktifkan Multi-AZ. Hal ini akan mempercepat impor data. Anda harus membuat basis data pada instans tersebut sebelum data dapat di-dump. Basis data tersebut bisa memiliki nama yang sama dengan basis data yang berisi data yang di-dump. Anda juga dapat membuat basis data dengan nama berbeda. Dalam kasus ini, gunakan perintah pg_restore dan parameter -d untuk memulihkan data ke basis data yang baru diberi nama.

Misalnya, perintah berikut dapat digunakan untuk mengeluarkan, memulihkan, dan mengubah nama basis data.

```
pg_dump -Fc -v -h [endpoint of instance] -U [master username] [database]
 > [database].dump
createdb [new database name]
pg_restore -v -h [endpoint of instance] -U [master username] -d [new database
 name] [database].dump
```

Langkah 3: Gunakan psql untuk membuat basis data pada instans DB dan muat data

Anda dapat menggunakan koneksi yang sama yang Anda gunakan untuk menjalankan perintah pg_dump untuk terhubung ke instans DB target dan membuat ulang basis data. Menggunakan psql, Anda bisa menggunakan nama pengguna master dan kata sandi master untuk membuat basis data pada instans DB

Contoh berikut menggunakan psql dan file dump bernama mydb2dump.sql untuk membuat basis data bernama mydb2 di instans DB PostgreSQL bernama mypginstance:

Untuk Linux, macOS, atau Unix:

```
psql \
   -f mydb2dump.sql \
   --host mypginstance.555555555555.aws-region.rds.amazonaws.com \
   --port 8199 \
   --username myawsuser \
   --password password \
   --dbname mydb2
```

Untuk Windows:

```
psql ^
   -f mydb2dump.sql ^
   --host mypginstance.555555555555.aws-region.rds.amazonaws.com ^
   --port 8199 ^
   --username myawsuser ^
   --password password ^
   --dbname mydb2
```

> **Note**
>
> Tentukan kata sandi selain perintah yang ditampilkan di sini sebagai praktik terbaik keamanan.

Langkah 4: Buat snapshot DB dari instans DB

Setelah Anda memverifikasi bahwa data telah dimuat ke dalam instans DB Anda, kami sarankan Anda membuat snapshot DB dari instans DB PostgreSQL target. Snapshot DB adalah cadangan lengkap instans DB Anda yang dapat digunakan untuk memulihkan instans DB Anda ke status yang

diketahui. Dengan snapshot DB yang diambil segera setelah pemuatan, Anda tidak perlu memuat data lagi jika terjadi hal yang tidak diinginkan. Anda juga dapat menggunakan snapshot tersebut untuk memulai instans DB baru. Untuk informasi selengkapnya tentang cara membuat snapshot DB secara manual, lihat Membuat snapshot DB untuk instans DB Single-AZ.

Menggunakan perintah \copy untuk mengimpor data ke tabel di instans DB PostgreSQL

Perintah `\copy` PostgreSQL adalah perintah meta yang tersedia dari alat klien interaktif `psql`. Anda dapat menggunakan `\copy` untuk mengimpor data ke tabel di instans DB RDS for PostgreSQL. Untuk menggunakan perintah `\copy`, Anda harus membuat struktur tabel pada instans DB target terlebih dahulu agar `\copy` memiliki tujuan untuk salinan data.

Anda dapat menggunakan `\copy` untuk memuat data dari file nilai yang dipisahkan koma (CSV), seperti file yang telah diekspor dan disimpan ke workstation klien Anda.

Untuk mengimpor data CSV ke instans DB RDS for PostgreSQL target, pertama-tama sambungkan ke instans DB target menggunakan `psql`.

```
psql --host=db-instance.111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password --dbname=target-db
```

Anda kemudian menjalankan perintah `\copy` dengan parameter berikut untuk mengidentifikasi target untuk data dan formatnya.

- `target_table` – Nama tabel yang akan menerima data yang disalin dari file CSV.
- `column_list` – Spesifikasi kolom untuk tabel.
- `'filename'` – Jalur lengkap ke file CSV di workstation lokal Anda.

```
 \copy target_table from '/path/to/local/filename.csv' WITH DELIMITER ',' CSV;
```

Jika file CSV Anda memiliki informasi judul kolom, Anda dapat menggunakan versi perintah dan parameter ini.

```
\copy target_table (column-1, column-2, column-3, ...)
    from '/path/to/local/filename.csv' WITH DELIMITER ',' CSV HEADER;
```

Jika perintah `\copy` gagal, PostgreSQL mengeluarkan pesan kesalahan.

Membuat instans DB baru di lingkungan Database Preview menggunakan perintah `psql` dengan perintah meta `\copy` seperti yang ditunjukkan pada contoh berikut. Contoh ini menggunakan source-table sebagai nama tabel sumber, source-table.csv sebagai file .csv, dan target-db sebagai basis data target:

Untuk Linux, macOS, atau Unix:

```
$psql target-db \
    -U <admin user> \
    -p <port> \
    -h <DB instance name> \
    -c "\copy source-table from 'source-table.csv' with DELIMITER ','"
```

Untuk Windows:

```
$psql target-db ^
    -U <admin user> ^
    -p <port> ^
    -h <DB instance name> ^
    -c "\copy source-table from 'source-table.csv' with DELIMITER ','"
```

Untuk detail lengkap tentang perintah `\copy`, lihat halaman psql dalam dokumentasi PostgreSQL, di bagian Meta-Commands.

Mengimpor data dari Amazon S3 ke instans DB RDS for PostgreSQL

Anda dapat mengimpor data yang telah disimpan menggunakan Amazon Simple Storage Service ke dalam tabel pada instans DB RDS for PostgreSQL. Untuk melakukannya, instal ekstensi RDS for PostgreSQL `aws_s3` terlebih dahulu. Ekstensi ini menyediakan fungsi yang Anda gunakan untuk mengimpor data dari bucket Amazon S3. Bucket adalah kontainer Amazon S3 untuk objek dan file. Data dapat berada dalam file nilai yang dipisahkan koma (CSV), file teks, atau file terkompresi (gzip). Di bagian berikut ini, Anda dapat mempelajari cara menginstal ekstensi tersebut dan cara mengimpor data dari Amazon S3 ke dalam tabel.

Basis data Anda harus menjalankan PostgreSQL versi 10.7 atau yang lebih tinggi untuk mengimpor dari Amazon S3 ke RDS for PostgreSQL.

Jika Anda tidak memiliki data yang tersimpan di Amazon S3, Anda harus terlebih dahulu membuat bucket dan menyimpan data. Untuk informasi selengkapnya, lihat topik berikut dalam Panduan Pengguna Amazon Simple Storage Service.

- Buat bucket
- Tambahkan objek ke bucket

Impor lintas akun dari Amazon S3 didukung. Untuk informasi selengkapnya, lihat Memberikan izin lintas akun dalam Panduan Pengguna Amazon Simple Storage Service.

Anda dapat menggunakan kunci yang dikelola pelanggan untuk enkripsi saat mengimpor data dari S3. Untuk informasi selengkapnya, lihat Kunci KMS yang disimpan di AWS KMS dalam Panduan Pengguna Amazon Simple Storage Service.

> **Note**
>
> Pengimporan data dari Amazon S3 tidak didukung untuk Aurora Serverless v1. Hal ini didukung untuk Aurora Serverless v2.

Topik

- Menginstal ekstensi aws_s3
- Gambaran umum pengimporan data dari data Amazon S3
- Mengatur akses ke bucket Amazon S3
- Mengimpor data dari Amazon S3 ke instans DB RDS for PostgreSQL
- Referensi fungsi

Menginstal ekstensi aws_s3

Sebelum Anda dapat menggunakan Amazon S3 dengan instans DB RDS for PostgreSQL, Anda perlu menginstal ekstensi `aws_s3`. Ekstensi ini menyediakan fungsi untuk mengimpor data dari Amazon S3. Ekstensi ini juga dilengkapi dengan fungsi untuk mengekspor data dari instans DB RDS for PostgreSQL ke bucket Amazon S3. Untuk informasi selengkapnya, lihat Mengekspor data dari instans DB RDS for PostgreSQL ke Amazon S3. Ekstensi `aws_s3` bergantung pada beberapa fungsi pembantu dalam ekstensi `aws_commons`, yang diinstal secara otomatis jika diperlukan.

Untuk menginstal ekstensi **`aws_s3`**

1. Gunakan psql (atau pgAdmin) untuk terhubung ke instans DB RDS for PostgreSQL sebagai pengguna yang memiliki hak istimewa `rds_superuser`. Jika Anda tetap menggunakan nama default selama proses penyiapan, Anda terhubung sebagai `postgres`.

```
psql --host=111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password
```

2. Untuk menginstal ekstensi, jalankan perintah berikut.

```
postgres=> CREATE EXTENSION aws_s3 CASCADE;
NOTICE: installing required extension "aws_commons"
CREATE EXTENSION
```

3. Untuk memastikan bahwa ekstensi telah diinstal, Anda dapat menggunakan metacommand `\dx` psql.

```
postgres=> \dx
       List of installed extensions
    Name     | Version |   Schema   |                 Description
-------------+---------+------------+---------------------------------------------
 aws_commons | 1.2     | public     | Common data types across AWS services
 aws_s3      | 1.1     | public     | AWS S3 extension for importing data from S3
 plpgsql     | 1.0     | pg_catalog | PL/pgSQL procedural language
(3 rows)
```

Fungsi untuk mengimpor data dari Amazon S3 dan mengekspor data ke Amazon S3 kini dapat digunakan.

Gambaran umum pengimporan data dari data Amazon S3

Untuk mengimpor data S3 ke Amazon RDS

Pertama, kumpulkan detail yang perlu Anda berikan ke fungsi tersebut. Hal ini termasuk nama tabel pada instans DB RDS for PostgreSQL Anda, dan nama bucket, jalur file, jenis file, serta Wilayah AWS tempat data Amazon S3 disimpan. Untuk informasi selengkapnya, buka Melihat objek dalam Panduan Pengguna Amazon Simple Storage Service.

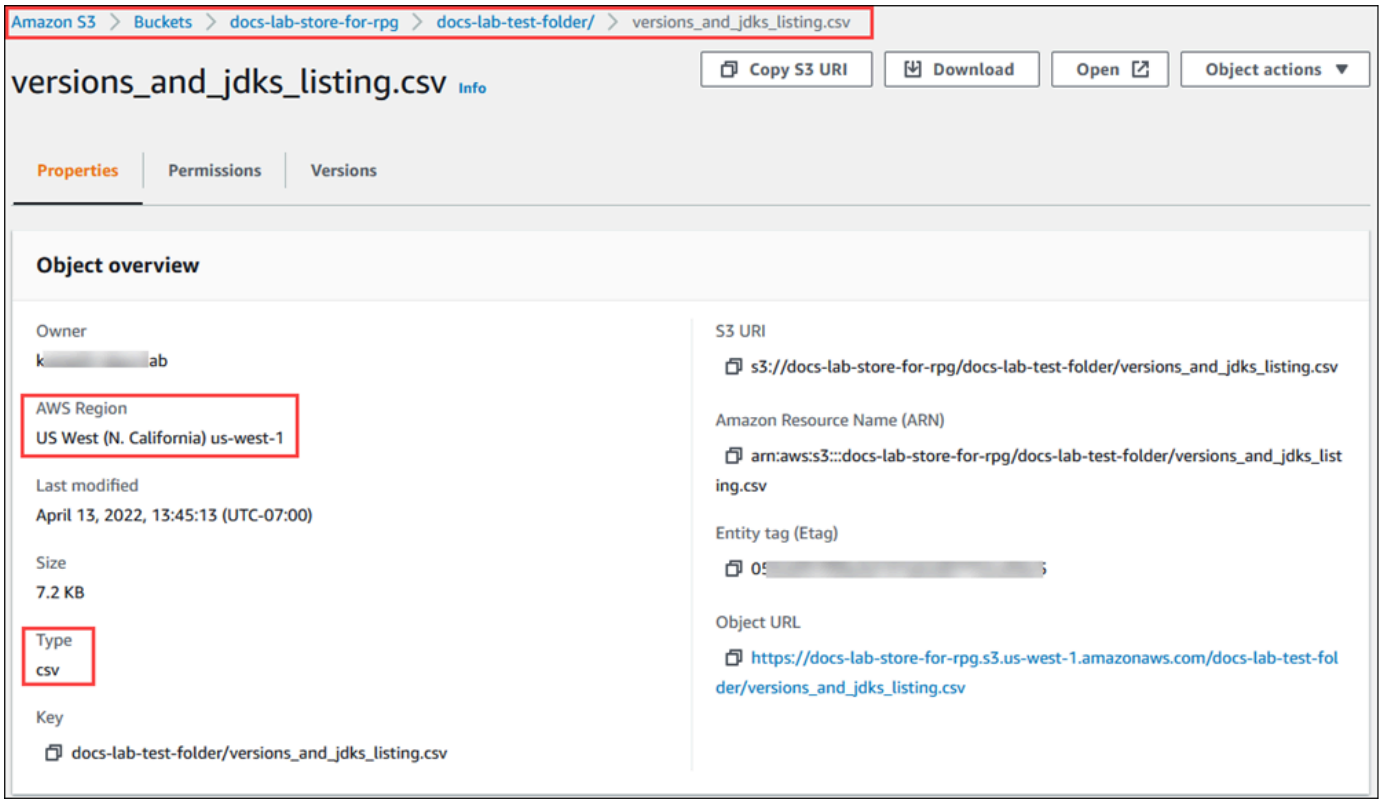
> **Note**
>
> Impor data multibagian dari Amazon S3 saat ini tidak didukung.

1. Dapatkan nama tabel di mana fungsi `aws_s3.table_import_from_s3` adalah untuk mengimpor data. Sebagai contoh, perintah berikut membuat tabel `t1` yang dapat digunakan di langkah selanjutnya.

```
postgres=> CREATE TABLE t1
    (col1 varchar(80),
    col2 varchar(80),
    col3 varchar(80));
```

2. Dapatkan detail tentang bucket Amazon S3 dan data yang akan diimpor. Untuk melakukannya, buka konsol Amazon S3 di https://console.aws.amazon.com/s3/, dan pilih Bucket. Temukan bucket yang berisi data Anda dalam daftar. Pilih bucket, buka halaman Ikhtisar objek, lalu pilih Properti.

   Catat nama bucket, jalur, Wilayah AWS, dan jenis file. Anda memerlukan Amazon Resource Name (ARN) nanti untuk mengatur akses ke Amazon S3 melalui peran IAM. Untuk informasi selengkapnya, lihat Mengatur akses ke bucket Amazon S3. Gambar berikut menunjukkan sebuah contoh.



3. Anda dapat memverifikasi jalur ke data di bucket Amazon S3 dengan menggunakan perintah AWS CLI `aws s3 cp`. Jika informasinya benar, perintah ini akan mengunduh salinan file Amazon S3.

```
aws s3 cp s3://sample_s3_bucket/sample_file_path ./
```

4. Siapkan izin di instans DB RDS for PostgreSQL untuk mengizinkan akses ke file di bucket Amazon S3. Untuk melakukannya, gunakan peran AWS Identity and Access Management (IAM) atau kredensial keamanan. Untuk informasi selengkapnya, lihat Mengatur akses ke bucket Amazon S3.
5. Berikan jalur dan detail objek Amazon S3 lainnya yang dikumpulkan (lihat langkah 2) ke fungsi `create_s3_uri` untuk membuat objek URI Amazon S3. Untuk mempelajari selengkapnya tentang fungsi ini, lihat aws_commons.create_s3_uri. Berikut ini adalah contoh penyusunan objek ini selama sesi psql.

   ```
   postgres=> SELECT aws_commons.create_s3_uri(
      'docs-lab-store-for-rpg',
      'versions_and_jdks_listing.csv',
      'us-west-1'
   ) AS s3_uri \gset
   ```

   Pada langkah berikutnya, Anda meneruskan objek ini (`aws_commons._s3_uri_1`) ke fungsi `aws_s3.table_import_from_s3` untuk mengimpor data ke tabel.
6. Invokasi fungsi `aws_s3.table_import_from_s3` untuk mengimpor data dari Amazon S3 ke dalam tabel Anda. Untuk informasi referensi, lihat aws_s3.table_import_from_s3. Sebagai contoh, lihat Mengimpor data dari Amazon S3 ke instans DB RDS for PostgreSQL.

Mengatur akses ke bucket Amazon S3

Untuk mengimpor data dari file Amazon S3, berikan izin pada instans DB RDS for PostgreSQL untuk mengakses bucket Amazon S3 yang berisi file tersebut. Anda dapat menyediakan akses ke bucket Amazon S3 dalam satu dari dua cara, seperti yang dijelaskan dalam topik berikut.

Topik

- Menggunakan peran IAM untuk mengakses bucket Amazon S3
- Menggunakan kredensial keamanan untuk mengakses bucket Amazon S3
- Memecahkan masalah akses ke Amazon S3

Menggunakan peran IAM untuk mengakses bucket Amazon S3

Sebelum Anda memuat data dari file Amazon S3, berikan izin pada instans DB RDS for PostgreSQL untuk mengakses bucket Amazon S3 yang berisi file tersebut. Dengan cara ini, Anda tidak perlu mengelola informasi kredensial tambahan atau menyediakannya di panggilan fungsi aws_s3.table_import_from_s3.

Untuk melakukannya, buat kebijakan IAM yang memberikan akses ke bucket Amazon S3. Buat peran IAM dan lampirkan kebijakan ke peran tersebut. Kemudian, tetapkan peran IAM ke klaster DB Anda.

> **Note**
>
> Anda tidak dapat mengaitkan peran IAM dengan klaster DB Aurora Serverless v1, sehingga langkah berikut tidak berlaku.

Untuk memberi instans DB RDS for PostgreSQL akses ke Amazon S3 melalui peran IAM

1. Buat kebijakan IAM.

   Kebijakan ini memberikan izin pada bucket dan objek yang memungkinkan instans DB RDS for PostgreSQL Anda mengakses Amazon S3.

   Dalam kebijakan, sertakan tindakan yang diperlukan berikut ini untuk memungkinkan transfer file dari bucket Amazon S3 ke Amazon RDS:

   - `s3:GetObject`
   - `s3:ListBucket`

   Sertakan sumber daya berikut dalam kebijakan untuk mengidentifikasi bucket dan objek Amazon S3 di bucket. Hal ini menunjukkan format Amazon Resource Name (ARN) untuk mengakses Amazon S3.

   - arn:aws:s3:::*your-s3-bucket*
   - arn:aws:s3:::*your-s3-bucket*/*

   Untuk informasi selengkapnya tentang cara membuat kebijakan IAM untuk RDS for PostgreSQL, lihat Membuat dan menggunakan kebijakan IAM untuk akses basis data IAM. Lihat juga Tutorial:

Membuat dan melampirkan kebijakan yang dikelola pelanggan pertama Anda dalam Panduan Pengguna IAM.

Perintah AWS CLI berikut membuat kebijakan IAM bernama `rds-s3-import-policy` dengan opsi ini. Perintah tersebut akan memberikan akses ke bucket yang bernama `your-s3-bucket`.

> **Note**
>
> Catat Amazon Resource Name (ARN) dari kebijakan yang ditampilkan oleh perintah ini. Anda memerlukan ARN di langkah berikutnya saat Anda melampirkan kebijakan ke peran IAM.

Example

Untuk Linux, macOS, atau Unix:

```
aws iam create-policy \
   --policy-name rds-s3-import-policy \
   --policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Sid": "s3import",
         "Action": [
           "s3:GetObject",
           "s3:ListBucket"
         ],
         "Effect": "Allow",
         "Resource": [
           "arn:aws:s3:::your-s3-bucket",
           "arn:aws:s3:::your-s3-bucket/*"
         ]
       }
     ]
   }'
```

Untuk Windows:

```
aws iam create-policy ^
   --policy-name rds-s3-import-policy ^
```

```
  --policy-document '{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Sid": "s3import",
        "Action": [
          "s3:GetObject",
          "s3:ListBucket"
        ],
        "Effect": "Allow",
        "Resource": [
          "arn:aws:s3:::your-s3-bucket",
          "arn:aws:s3:::your-s3-bucket/*"
        ]
      }
    ]
  }'
```

2. Buat peran IAM.

   Anda melakukannya agar Amazon RDS dapat mengambil peran IAM ini untuk mengakses bucket Amazon S3 Anda. Untuk informasi selengkapnya, lihat Membuat peran untuk mendelegasikan izin ke pengguna IAM dalam Panduan Pengguna IAM.

   Sebaiknya gunakan kunci konteks kondisi global `aws:SourceArn` dan `aws:SourceAccount` dalam kebijakan berbasis sumber daya untuk membatasi izin layanan ke sumber daya tertentu. Ini adalah perlindungan paling efektif dari masalah confused deputy.

   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.

   - Gunakan `aws:SourceArn` jika Anda menginginkan akses lintas layanan untuk satu sumber daya.
   - Gunakan `aws:SourceAccount` jika Anda ingin mengizinkan pengaitan sumber daya apa pun di akun tersebut dengan penggunaan lintas layanan.

   Dalam kebijakan, pastikan untuk menggunakan kunci konteks kondisi global `aws:SourceArn` dengan ARN penuh sumber daya. Contoh berikut menunjukkan cara melakukannya menggunakan perintah AWS CLI untuk membuat peran dengan nama `rds-s3-import-role`.

Example

Untuk Linux, macOS, atau Unix:

```
aws iam create-role \
   --role-name rds-s3-import-role \
   --assume-role-policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
            "Service": "rds.amazonaws.com"
          },
         "Action": "sts:AssumeRole",
         "Condition": {
             "StringEquals": {
                "aws:SourceAccount": "111122223333",
                "aws:SourceArn": "arn:aws:rds:us-east-1:111122223333:db:dbname"
                }
             }
       }
     ]
   }'
```

Untuk Windows:

```
aws iam create-role ^
   --role-name rds-s3-import-role ^
   --assume-role-policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
            "Service": "rds.amazonaws.com"
          },
         "Action": "sts:AssumeRole",
         "Condition": {
             "StringEquals": {
                "aws:SourceAccount": "111122223333",
                "aws:SourceArn": "arn:aws:rds:us-east-1:111122223333:db:dbname"
```

```
                }
            }
        }
    ]
}'
```

3. Lampirkan kebijakan IAM yang Anda buat ke peran IAM yang Anda buat.

   Perintah AWS CLI berikut melampirkan kebijakan yang dibuat di langkah sebelumnya ke peran bernama `rds-s3-import-role`. Ganti *`your-policy-arn`* dengan kebijakan ARN yang Anda catat di langkah sebelumnya.

   Example

   Untuk Linux, macOS, atau Unix:

   ```
   aws iam attach-role-policy \
      --policy-arn your-policy-arn \
      --role-name rds-s3-import-role
   ```

   Untuk Windows:

   ```
   aws iam attach-role-policy ^
      --policy-arn your-policy-arn ^
      --role-name rds-s3-import-role
   ```

4. Tambahkan peran IAM ke klaster DB.

   Lakukan menggunakan AWS Management Console atau AWS CLI, seperti yang dijelaskan berikut.

Konsol

Untuk menambahkan peran IAM untuk instans DB PostgreSQL menggunakan konsol

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Pilih nama instans DB PostgreSQL untuk menampilkan detailnya.
3. Di tab Konektivitas & keamanan, di bagian Kelola peran IAM, pilih peran yang akan ditambahkan pada bagian Tambahkan peran IAM ke instans ini.

4. Di bagian Fitur, pilih s3Import.
5. Pilih Tambahkan peran.

AWS CLI

Untuk menambahkan peran IAM untuk instans DB PostgreSQL menggunakan CLI

- Gunakan perintah berikut untuk menambahkan peran ke instans DB PostgreSQL bernama `my-db-instance`. Ganti *`your-role-arn`* dengan ARN peran yang Anda catat pada langkah sebelumnya. Gunakan `s3Import` untuk nilai opsi `--feature-name`.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds add-role-to-db-instance \
   --db-instance-identifier my-db-instance \
   --feature-name s3Import \
   --role-arn your-role-arn   \
   --region your-region
```

  Untuk Windows:

```
aws rds add-role-to-db-instance ^
   --db-instance-identifier my-db-instance ^
   --feature-name s3Import ^
   --role-arn your-role-arn ^
   --region your-region
```

API RDS

Menggunakan kredensial keamanan untuk mengakses bucket Amazon S3

Jika Anda mau, Anda dapat menggunakan kredensial keamanan untuk memberikan akses ke bucket Amazon S3 alih-alih memberikan akses ke peran IAM. Anda melakukannya dengan menentukan parameter `credentials` dalam panggilan fungsi aws_s3.table_import_from_s3.

Parameter `credentials` adalah struktur dari jenis `aws_commons._aws_credentials_1`, yang berisi kredensial AWS. Gunakan fungsi aws_commons.create_aws_credentials untuk mengatur

kunci akses dan kunci rahasia dalam struktur `aws_commons._aws_credentials_1`, seperti yang ditunjukkan berikut ini.

```
postgres=> SELECT aws_commons.create_aws_credentials(
   'sample_access_key', 'sample_secret_key', '')
AS creds \gset
```

Setelah membuat struktur `aws_commons._aws_credentials_1` , gunakan fungsi aws_s3.table_import_from_s3 dengan parameter `credentials` untuk mengimpor data, seperti yang ditunjukkan berikut.

```
postgres=> SELECT aws_s3.table_import_from_s3(
   't', '', '(format csv)',
   :'s3_uri',
   :'creds'
);
```

Atau Anda dapat menyertakan panggilan fungsi aws_commons.create_aws_credentials sebaris dalam panggilan fungsi `aws_s3.table_import_from_s3`.

```
postgres=> SELECT aws_s3.table_import_from_s3(
   't', '', '(format csv)',
   :'s3_uri',
   aws_commons.create_aws_credentials('sample_access_key', 'sample_secret_key', '')
);
```

Memecahkan masalah akses ke Amazon S3

Jika Anda mengalami masalah koneksi saat mencoba mengimpor data dari Amazon S3, lihat rekomendasi berikut:

- Memecahkan masalah identitas dan akses Amazon RDS
- Memecahkan Masalah Amazon S3 dalam Panduan Pengguna Amazon Simple Storage Service
- Pemecahan Masalah Amazon S3 dan IAM dalam Panduan Pengguna IAM

Mengimpor data dari Amazon S3 ke instans DB RDS for PostgreSQL

Anda mengimpor data dari bucket Amazon S3 dengan menggunakan fungsi `table_import_from_s3` aws_s3. Untuk informasi referensi, lihat aws_s3.table_import_from_s3.

> **Note**
>
> Contoh berikut menggunakan metode peran IAM untuk mengizinkan akses ke bucket Amazon S3. Dengan demikian, panggilan fungsi `aws_s3.table_import_from_s3` tidak menyertakan parameter kredensial.

Berikut ini adalah contoh umumnya.

```
postgres=> SELECT aws_s3.table_import_from_s3(
   't1',
   '',
   '(format csv)',
   :'s3_uri'
);
```

Parameternya adalah sebagai berikut:

- `t1` – Nama untuk tabel dalam instans DB PostgreSQL tempat tujuan penyalinan data.
- `''` – Daftar opsional kolom dalam tabel basis data. Anda dapat menggunakan parameter ini untuk menunjukkan kolom data S3 yang akan dimasukkan dan kolom tabel tempat memasukkan kolom data S3 tersebut. Jika tidak ada kolom yang ditentukan, semua kolom disalin ke tabel. Untuk contoh cara menggunakan daftar kolom, lihat Mengimpor file Amazon S3 yang menggunakan pemisah kustom.
- `(format csv)` – Argumen PostgreSQL COPY. Proses penyalinan menggunakan argumen dan format perintah PostgreSQL COPY untuk mengimpor data. Pilihan formatnya mencakup nilai yang dipisahkan dengan koma (CSV), seperti yang ditunjukkan dalam contoh ini, serta teks dan biner. Default-nya adalah teks.
- `s3_uri` – Struktur yang berisi informasi yang mengidentifikasi file Amazon S3. Untuk contoh cara menggunakan fungsi aws_commons.create_s3_uri untuk membuat struktur `s3_uri`, lihat Gambaran umum pengimporan data dari data Amazon S3.

Untuk informasi selengkapnya tentang fungsi ini, lihat aws_s3.table_import_from_s3.

Fungsi `aws_s3.table_import_from_s3` menampilkan teks. Untuk menentukan jenis file lain yang akan diimpor dari bucket Amazon S3, lihat salah satu contoh berikut.

> **Note**
>
> Mengimpor file 0 byte akan menyebabkan kesalahan.

Topik

- Mengimpor file Amazon S3 yang menggunakan pemisah kustom
- Mengimpor file (gzip) terkompresi Amazon S3
- Mengimpor file Amazon S3 yang dienkode

Mengimpor file Amazon S3 yang menggunakan pemisah kustom

Contoh berikut menunjukkan cara mengimpor file yang menggunakan pemisah kustom. Hal ini juga menunjukkan cara mengontrol lokasi penempatan data dalam tabel basis data menggunakan parameter `column_list` dari fungsi aws_s3.table_import_from_s3.

Untuk contoh ini, asumsikan bahwa informasi berikut ini diatur ke dalam kolom yang dipisahkan tanda pipa di file Amazon S3.

```
1|foo1|bar1|elephant1
2|foo2|bar2|elephant2
3|foo3|bar3|elephant3
4|foo4|bar4|elephant4
...
```

Untuk mengimpor file yang menggunakan pemisah kustom

1. Buat tabel di basis data untuk data yang diimpor.

   ```
   postgres=> CREATE TABLE test (a text, b text, c text, d text, e text);
   ```

2. Gunakan form berikut dari fungsi aws_s3.table_import_from_s3 untuk mengimpor data dari file Amazon S3.

   Anda dapat memasukkan panggilan fungsi aws_commons.create_s3_uri sebaris dalam panggilan fungsi `aws_s3.table_import_from_s3` untuk menentukan file.

   ```
   postgres=> SELECT aws_s3.table_import_from_s3(
      'test',
   ```

```
   'a,b,d,e',
   'DELIMITER ''|''',
   aws_commons.create_s3_uri('sampleBucket', 'pipeDelimitedSampleFile', 'us-
east-2')
);
```

Data tersebut sekarang berada dalam tabel di kolom berikut.

```
postgres=> SELECT * FROM test;
a | b | c | d | e
---+------+---+---+------+-----------
1 | foo1 | | bar1 | elephant1
2 | foo2 | | bar2 | elephant2
3 | foo3 | | bar3 | elephant3
4 | foo4 | | bar4 | elephant4
```

Mengimpor file (gzip) terkompresi Amazon S3

Contoh berikut menunjukkan cara mengimpor file dari Amazon S3 yang dikompresi menggunakan gzip. File yang Anda impor harus memiliki metadata Amazon S3 berikut:

- Kunci: `Content-Encoding`
- Nilai: `gzip`

Jika Anda mengunggah file menggunakan AWS Management Console, metadata biasanya diterapkan oleh sistem. Untuk informasi tentang cara mengunggah file ke Amazon S3 menggunakan AWS Management Console, AWS CLI, atau API, lihat Mengunggah objek dalam Panduan Pengguna Amazon Simple Storage Service.

Untuk informasi selengkapnya tentang metadata Amazon S3 dan detail tentang metadata yang disediakan sistem, lihat Mengedit metadata objek di konsol Amazon S3 dalam Panduan Pengguna Amazon Simple Storage Service.

Impor file gzip ke dalam instans DB RDS for PostgreSQL seperti yang ditunjukkan berikut ini.

```
postgres=> CREATE TABLE test_gzip(id int, a text, b text, c text, d text);
postgres=> SELECT aws_s3.table_import_from_s3(
 'test_gzip', '', '(format csv)',
 'myS3Bucket', 'test-data.gz', 'us-east-2'
```

```
);
```

Mengimpor file Amazon S3 yang dienkode

Contoh berikut menunjukkan cara mengimpor file dari Amazon S3 yang memiliki pengenkodean Windows-1252.

```
postgres=> SELECT aws_s3.table_import_from_s3(
 'test_table', '', 'encoding ''WIN1252''',
 aws_commons.create_s3_uri('sampleBucket', 'SampleFile', 'us-east-2')
);
```

Referensi fungsi

Fungsi

- aws_s3.table_import_from_s3
- aws_commons.create_s3_uri
- aws_commons.create_aws_credentials

aws_s3.table_import_from_s3

Mengimpor data Amazon S3 ke tabel Amazon RDS. Ekstensi `aws_s3` memberikan fungsi `aws_s3.table_import_from_s3`. Nilai yang ditampilkan berupa teks.

Sintaksis

Parameter yang diperlukan adalah `table_name`, `column_list`, dan `options`. Parameter ini mengidentifikasi tabel basis data dan menentukan cara data disalin ke dalam tabel.

Anda juga dapat menggunakan parameter berikut ini:

- Parameter `s3_info` menentukan file Amazon S3 yang akan diimpor. Saat Anda menggunakan parameter ini, akses ke Amazon S3 disediakan oleh peran IAM untuk instans DB PostgreSQL.

```
aws_s3.table_import_from_s3 (
   table_name text,
   column_list text,
   options text,
   s3_info aws_commons._s3_uri_1
)
```

- Parameter `credentials` menentukan kredensial untuk mengakses Amazon S3. Saat Anda menggunakan parameter ini, Anda tidak menggunakan peran IAM.

```
aws_s3.table_import_from_s3 (
   table_name text,
   column_list text,
   options text,
   s3_info aws_commons._s3_uri_1,
   credentials aws_commons._aws_credentials_1
)
```

Parameter

table_name

String teks wajib yang berisi nama tabel basis data PostgreSQL sebagai tujuan impor data.

column_list

String teks wajib yang berisi daftar opsional kolom tabel basis data PostgreSQL untuk menyalin data. Jika string kosong, semua kolom tabel digunakan. Sebagai contoh, lihat Mengimpor file Amazon S3 yang menggunakan pemisah kustom.

options

String teks wajib yang berisi argumen untuk perintah `COPY` PostgreSQL. Argumen ini menentukan cara data akan disalin ke dalam tabel PostgreSQL. Untuk detail selengkapnya, lihat dokumentasi COPY PostgreSQL.

s3_info

Jenis komposit `aws_commons._s3_uri_1` yang berisi informasi tentang objek S3 berikut:

- `bucket` – Nama bucket Amazon S3 yang berisi file.
- `file_path` – Nama file Amazon S3 termasuk jalur file.
- `region` – Wilayah AWS tempat file berada. Untuk daftar nama Wilayah AWS dan nilai terkait, lihat Wilayah, Zona Ketersediaan, dan Zona Lokal.

credentials

Jenis komposit `aws_commons._aws_credentials_1` yang berisi kredensial berikut yang akan digunakan untuk operasi impor:

- Kunci akses
- Kunci rahasia
- Token sesi

Untuk informasi tentang cara membuat struktur komposit `aws_commons._aws_credentials_1`, lihat aws_commons.create_aws_credentials.

Sintaksis alternatif

Untuk memudahkan pengujian, Anda dapat menggunakan serangkaian parameter yang diperluas, bukan parameter `s3_info` dan `credentials`. Berikut ini adalah variasi sintaksis tambahan untuk fungsi `aws_s3.table_import_from_s3`:

- Alih-alih menggunakan parameter `s3_info` untuk mengidentifikasi file Amazon S3, gunakan kombinasi parameter `bucket`, `file_path`, dan `region`. Dengan form fungsi ini, akses ke Amazon S3 disediakan oleh peran IAM pada instans DB PostgreSQL.

```
aws_s3.table_import_from_s3 (
   table_name text,
   column_list text,
   options text,
   bucket text,
   file_path text,
   region text
)
```

- Alih-alih menggunakan parameter `credentials` untuk menentukan akses Amazon S3, gunakan kombinasi parameter `access_key`, `session_key`, dan `session_token`.

```
aws_s3.table_import_from_s3 (
   table_name text,
   column_list text,
   options text,
   bucket text,
   file_path text,
   region text,
   access_key text,
   secret_key text,
   session_token text
)
```

Parameter alternatif

bucket

String teks yang berisi nama bucket Amazon S3 yang berisi file.

file_path

String teks yang berisi nama file Amazon S3 beserta jalur file.

region

String teks yang mengidentifikasi lokasi Wilayah AWS file. Untuk daftar nama Wilayah AWS dan nilai terkait, lihat Wilayah, Zona Ketersediaan, dan Zona Lokal.

access_key

String teks yang berisi kunci akses yang akan digunakan dalam operasi impor. Default-nya adalah NULL.

secret_key

String teks yang berisi kunci rahasia yang akan digunakan dalam operasi impor. Default-nya adalah NULL.

session_token

(Opsional) String teks yang berisi kunci sesi yang akan digunakan untuk operasi impor. Default-nya adalah NULL.

aws_commons.create_s3_uri

Membuat struktur `aws_commons._s3_uri_1` untuk menyimpan informasi file Amazon S3. Gunakan hasil dari fungsi `aws_commons.create_s3_uri` di parameter `s3_info` dari fungsi aws_s3.table_import_from_s3.

Sintaksis

```
aws_commons.create_s3_uri(
   bucket text,
   file_path text,
   region text
)
```

Parameter

bucket

String teks wajib berisi nama bucket Amazon S3 untuk file.

file_path

String teks wajib yang berisi nama file Amazon S3 beserta jalurnya.

region

String teks wajib yang berisi Wilayah AWS tempat file berada. Untuk daftar nama Wilayah AWS dan nilai terkait, lihat Wilayah, Zona Ketersediaan, dan Zona Lokal.

aws_commons.create_aws_credentials

Mengatur kunci akses dan kunci rahasia dalam struktur `aws_commons._aws_credentials_1`. Gunakan hasil dari fungsi `aws_commons.create_aws_credentials` di parameter `credentials` dari fungsi aws_s3.table_import_from_s3.

Sintaksis

```
aws_commons.create_aws_credentials(
   access_key text,
   secret_key text,
   session_token text
)
```

Parameter

access_key

String teks wajib yang berisi kunci akses yang digunakan untuk mengimpor file Amazon S3. Default-nya adalah NULL.

secret_key

String teks wajib yang berisi kunci rahasia yang digunakan untuk mengimpor file Amazon S3. Default-nya adalah NULL.

session_token

String teks opsional yang berisi token sesi yang akan digunakan untuk mengimpor file Amazon S3. Default-nya adalah NULL. Jika Anda memberikan `session_token` opsional, Anda dapat menggunakan kredensial sementara.

Mentranspor basis data PostgreSQL antara instans DB

Dengan menggunakan basis data PostgreSQL yang dapat ditranspor untuk Amazon RDS, Anda dapat memindahkan basis data PostgreSQL antara dua instans DB. Ini adalah cara yang sangat cepat untuk memigrasikan basis data besar antara instans DB yang berbeda. Untuk menggunakan pendekatan ini, instans DB Anda harus menjalankan PostgreSQL versi utama.

Kemampuan ini mengharuskan Anda menginstal ekstensi `pg_transport` di instans DB sumber dan tujuan. Ekstensi `pg_transport` menyediakan mekanisme transportasi fisik yang memindahkan file basis data dengan pemrosesan minimal. Mekanisme ini memindahkan data jauh lebih cepat daripada proses dump dan load tradisional, dengan waktu henti yang lebih sedikit.

> **Note**
>
> Basis data PostgreSQL yang dapat ditranspor tersedia dalam RDS for PostgreSQL 11.5 dan yang lebih tinggi, dan RDS for PostgreSQL versi 10.10 dan yang lebih tinggi.

Untuk mentranspor instans DB PostgreSQL dari satu instans DB RDS for PostgreSQL ke instans DB lainnya, pertama-tama siapkan instans sumber dan tujuan sebagaimana dijelaskan dalam Menyiapkan instans DB untuk transportasi. Anda kemudian dapat mentranspor basis data menggunakan fungsi yang dijelaskan dalam Mentranspor basis data PostgreSQL.

Topik

- Batasan dalam penggunaan basis data PostgreSQL yang dapat ditranspor
- Bersiap untuk mentranspor basis data PostgreSQL
- Mentranspor basis data PostgreSQL ke tujuan dari sumber
- Apa yang terjadi selama transportasi basis data
- Referensi fungsi basis data yang dapat ditranspor
- Referensi parameter basis data yang dapat ditranspor

Batasan dalam penggunaan basis data PostgreSQL yang dapat ditranspor

Basis data yang dapat ditranspor memiliki batasan berikut:

- Replika baca – Anda tidak dapat menggunakan basis data yang dapat ditranspor pada replika baca atau instans induk replika baca.
- Jenis kolom yang tidak didukung – Anda tidak dapat menggunakan jenis data `reg` dalam tabel basis data apa pun yang akan Anda transportasikan dengan metode ini. Jenis ini bergantung pada ID objek (OID) katalog sistem, yang sering berubah selama transportasi.
- Tablespace – Semua objek basis data sumber harus dalam tablespace `pg_default` default.
- Kompatibilitas – Instans DB sumber dan tujuan harus menjalankan PostgreSQL dalam versi utama yang sama.
- Ekstensi — Instans DB sumber hanya dapat memiliki penginstalan `pg_transport`.
- Peran dan ACL – Informasi hak istimewa akses dan kepemilikan basis data sumber tidak dibawa ke basis data tujuan. Semua objek basis data dibuat dan dimiliki oleh pengguna tujuan transportasi lokal.
- Transportasi bersamaan — Instans DB tunggal dapat mendukung hingga 32 transportasi bersamaan, termasuk impor dan ekspor, jika proses pekerja telah dikonfigurasi dengan benar.
- Khusus instans DB RDS for PostgreSQL - Basis data PostgreSQL yang dapat ditranspor didukung hanya pada instans DB RDS for PostgreSQL. Anda tidak dapat menggunakannya dengan basis data on-premise atau basis data yang berjalan di Amazon EC2.

Bersiap untuk mentranspor basis data PostgreSQL

Sebelum memulai, pastikan bahwa instans DB RDS for PostgreSQL Anda memenuhi persyaratan berikut:

- Instans DB RDS for PostgreSQL untuk sumber dan tujuan harus menjalankan versi PostgreSQL yang sama.
- DB tujuan tidak dapat memiliki basis data dengan nama yang sama dengan DB sumber yang ingin Anda transportasikan.
- Akun yang Anda gunakan untuk menjalankan transportasi membutuhkan hak istimewa `rds_superuser` pada DB sumber dan DB tujuan.

- Grup keamanan untuk instans DB sumber harus mengizinkan akses masuk dari instans DB tujuan. Izin ini mungkin sudah ada jika instans DB sumber dan tujuan Anda berada di VPC. Untuk informasi selengkapnya tentang grup keamanan, lihat Mengontrol akses dengan grup keamanan.

Mentranspor basis data dari instans DB sumber ke instans DB tujuan memerlukan beberapa perubahan pada grup parameter DB yang terkait dengan setiap instans. Artinya Anda harus membuat grup parameter DB kustom untuk instans DB sumber dan membuat grup parameter DB kustom untuk instans DB tujuan.

> **Note**
>
> Jika instans DB Anda sudah dikonfigurasi menggunakan grup parameter DB kustom, Anda dapat memulai dengan langkah 2 dalam prosedur berikut.

Cara mengonfigurasi parameter grup DB kustom untuk mentranspor basis data

Untuk langkah-langkah berikut, gunakan akun yang memiliki hak istimewa `rds_superuser`.

1. Jika instans DB sumber dan tujuan menggunakan grup parameter DB default, Anda perlu membuat parameter DB kustom menggunakan versi yang sesuai untuk instans Anda. Ini dilakukan agar Anda dapat mengubah nilai untuk beberapa parameter. Untuk informasi selengkapnya, lihat Bekerja dengan grup parameter.
2. Dalam grup parameter DB kustom, ubah nilai untuk parameter berikut:
   - `shared_preload_libraries`— Tambahkan `pg_transport` ke daftar pustaka.
   - `pg_transport.num_workers` – Nilai default-nya adalah 3. Tingkatkan atau kurangi nilai ini sesuai kebutuhan untuk basis data Anda. Untuk basis data 200 GB, kami sarankan tidak lebih dari 8. Perlu diingat bahwa jika Anda meningkatkan nilai default untuk parameter ini, Anda juga harus meningkatkan nilai `max_worker_processes`.
   - `pg_transport.work_mem` – Nilai default-nya adalah 128 MB atau 256 MB, tergantung versi PostgreSQL. Pengaturan default biasanya dapat dibiarkan saja.
   - `max_worker_processes` – Nilai parameter ini perlu diatur menggunakan perhitungan berikut:

     ```
     3 * pg_transport.num_workers) + 9
     ```

Nilai ini diperlukan di tujuan untuk menangani berbagai proses pekerja latar belakang yang terlibat dalam transportasi. Untuk mempelajari lebih lanjut tentang `max_worker_processes`, , lihat Resource Consumption dalam dokumentasi PostgreSQL.

Untuk informasi selengkapnya tentang parameter `pg_transport`, lihat Referensi parameter basis data yang dapat ditranspor .

3. Reboot instans DB RDS for PostgreSQL sumber dan instans tujuan agar pengaturan untuk parameter tersebut berlaku.
4. Terhubung ke instans DB RDS for PostgreSQL sumber Anda.

```
psql --host=source-instance.111122223333.aws-region.rds.amazonaws.com --port=5432
 --username=postgres --password
```

5. Hapus ekstensi asing dari skema publik instans DB. Hanya ekstensi `pg_transport` yang diizinkan selama operasi transportasi aktual.
6. Instal ekstensi `pg_transport` sebagai berikut:

```
postgres=> CREATE EXTENSION pg_transport;
CREATE EXTENSION
```

7. Terhubung ke instans DB RDS for PostgreSQL tujuan Anda. Hapus ekstensi asing apa pun, lalu instal ekstensi `pg_transport`.

```
postgres=> CREATE EXTENSION pg_transport;
CREATE EXTENSION
```

Mentranspor basis data PostgreSQL ke tujuan dari sumber

Setelah Anda menyelesaikan proses yang dijelaskan dalam Bersiap untuk mentranspor basis data PostgreSQL, Anda dapat memulai transportasi. Untuk melakukannya, jalankan fungsi `transport.import_from_server` pada instans DB tujuan. Dalam sintaks berikut ini Anda dapat menemukan parameter fungsi.

```
SELECT transport.import_from_server(
   'source-db-instance-endpoint',
    source-db-instance-port,
```

```
   'source-db-instance-user',
   'source-user-password',
   'source-database-name',
   'destination-user-password',
   false);
```

Nilai `false` yang ditunjukkan dalam contoh memberi tahu fungsi bahwa ini bukan dry run. Untuk menguji penyiapan transportasi, Anda dapat menentukan `true` untuk opsi `dry_run` saat memanggil fungsi, seperti yang ditunjukkan berikut ini:

```
postgres=> SELECT transport.import_from_server(
    'docs-lab-source-db.666666666666aws-region.rds.amazonaws.com', 5432,
    'postgres', '********', 'labdb', '******', true);
INFO:  Starting dry-run of import of database "labdb".
INFO:  Created connections to remote database        (took 0.03 seconds).
INFO:  Checked remote cluster compatibility          (took 0.05 seconds).
INFO:  Dry-run complete                          (took 0.08 seconds total).
 import_from_server
--------------------

(1 row)
```

Baris INFO adalah output karena parameter `pg_transport.timing` diatur ke nilai default-nya, `true`. Atur `dry_run` ke `false` ketika Anda menjalankan perintah dan basis data sumber diimpor ke tujuan, seperti yang ditunjukkan berikut:

```
INFO:  Starting import of database "labdb".
INFO:  Created connections to remote database        (took 0.02 seconds).
INFO:  Marked remote database as read only           (took 0.13 seconds).
INFO:  Checked remote cluster compatibility          (took 0.03 seconds).
INFO:  Signaled creation of PITR blackout window     (took 2.01 seconds).
INFO:  Applied remote database schema pre-data       (took 0.50 seconds).
INFO:  Created connections to local cluster          (took 0.01 seconds).
INFO:  Locked down destination database              (took 0.00 seconds).
INFO:  Completed transfer of database files          (took 0.24 seconds).
INFO:  Completed clean up                            (took 1.02 seconds).
INFO:  Physical transport complete               (took 3.97 seconds total).
import_from_server
--------------------
(1 row)
```

Fungsi ini mengharuskan Anda memberikan kata sandi pengguna basis data. Oleh karena itu, kami menyarankan Anda untuk mengubah kata sandi dari peran pengguna yang Anda gunakan setelah transportasi selesai. Atau, Anda dapat menggunakan variabel terikat SQL untuk membuat peran pengguna sementara. Gunakan peran sementara ini untuk transportasi, lalu buang peran tersebut setelahnya.

Jika transportasi Anda tidak berhasil, Anda mungkin melihat pesan kesalahan yang mirip dengan yang berikut ini:

```
pg_transport.num_workers=8 25% of files transported failed to download file data
```

Pesan kesalahan "gagal mengunduh data file" menunjukkan bahwa jumlah proses pekerja tidak diatur dengan benar untuk ukuran basis data. Anda mungkin perlu meningkatkan atau mengurangi nilai yang diatur untuk `pg_transport.num_workers`. Setiap kegagalan melaporkan persentase penyelesaian, sehingga Anda dapat melihat dampak perubahan Anda. Misalnya, mengubah pengaturan dari 8 menjadi 4 dalam satu kasus menghasilkan hal berikut:

```
pg_transport.num_workers=4 75% of files transported failed to download file data
```

Perlu diingat bahwa parameter `max_worker_processes` juga diperhitungkan selama proses transportasi. Dengan kata lain, Anda mungkin perlu memodifikasi `pg_transport.num_workers` dan `max_worker_processes` agar transportasi basis data berhasil dijalankan. Contoh yang ditampilkan akhirnya berfungsi ketika `pg_transport.num_workers` diatur ke 2:

```
pg_transport.num_workers=2 100% of files transported
```

Lihat informasi selengkapnya tentang fungsi `transport.import_from_server` dan parameternya di Referensi fungsi basis data yang dapat ditranspor.

Apa yang terjadi selama transportasi basis data

Fitur basis data PostgreSQL yang dapat ditranspor menggunakan model tarik untuk mengimpor basis data dari instans DB sumber ke tujuan. Fungsi `transport.import_from_server` membuat basis data bergerak pada instans DB tujuan. Basis data bergerak tidak dapat diakses pada instans DB tujuan selama durasi transportasi.

Ketika transportasi dimulai, semua sesi saat ini pada basis data sumber berakhir. Setiap basis data selain basis data sumber pada instans DB sumber tidak terpengaruh oleh transportasi.

Basis data sumber dibuat menjadi mode hanya-baca khusus. Saat berada dalam mode ini, Anda dapat terhubung ke basis data sumber dan menjalankan kueri hanya baca. Namun, kueri berkemampuan tulis dan beberapa jenis perintah lainnya diblokir. Hanya basis data sumber spesifik yang ditranspor yang terpengaruh oleh pembatasan ini.

Selama transportasi, Anda tidak dapat memulihkan instans DB tujuan ke suatu titik waktu. Ini karena transportasi tersebut tidak bersifat transaksional dan tidak menggunakan log write-ahead PostgreSQL untuk mencatat perubahan. Jika instans DB tujuan mengaktifkan pencadangan otomatis, pencadangan otomatis diambil setelah transportasi selesai. Pemulihan titik waktu tersedia selama beberapa waktu setelah pencadangan selesai.

Jika transportasi gagal, ekstensi `pg_transport` berupaya untuk membatalkan semua perubahan ke instans DB sumber dan tujuan. Ini termasuk menghapus basis data yang ditranspor sebagian di tujuan. Bergantung pada jenis kegagalan, basis data sumber dapat terus menolak kueri berkemampuan tulis. Jika ini terjadi, gunakan perintah berikut untuk memungkinkan kueri berkemampuan tulis.

```
ALTER DATABASE db-name SET default_transaction_read_only = false;
```

Referensi fungsi basis data yang dapat ditranspor

Fungsi `transport.import_from_server` mentranspor basis data PostgreSQL dengan mengimpornya dari sumber instans DB ke instans DB tujuan. Hal ini dilakukan menggunakan mekanisme transportasi koneksi basis data fisik.

Sebelum memulai transportasi, fungsi ini memverifikasi bahwa instans DB sumber dan tujuan merupakan versi yang sama dan kompatibel untuk migrasi. Hal ini juga menegaskan bahwa instans DB tujuan memiliki cukup ruang untuk sumbernya.

Sintaks

```
transport.import_from_server(
   host text,
   port int,
   username text,
   password text,
   database text,
   local_password text,
   dry_run bool
)
```

Nilai yang Ditampilkan

Tidak ada.

Parameter

Anda dapat menemukan deskripsi parameter fungsi `transport.import_from_server` dalam tabel berikut.

Parameter	Deskripsi
`host`	Titik akhir instans DB sumber.
`port`	Integer yang mewakili port instans DB sumber. Instans DB PostgreSQL kerap menggunakan port 5432.
`username`	Pengguna instans DB sumber. Pengguna ini harus menjadi anggota peran `rds_superuser` .
`password`	Kata sandi pengguna instans DB sumber.
`database`	Nama basis data dalam instans DB sumber untuk ditranspor.
`local_password`	Kata sandi lokal pengguna saat ini untuk instans DB tujuan. Pengguna ini harus menjadi anggota peran `rds_superuser` .
`dry_run`	Nilai Boolean opsional yang menetapkan apakah dry run akan dijalankan. Default-nya adalah `false`, yang artinya transportasi berlanjut. Untuk mengonfirmasi kompatibilitas antara instans DB sumber dan tujuan tanpa benar-benar melakukan transportasi, atur `dry_run` ke `true`.

Contoh

Sebagai contoh, lihat Mentranspor basis data PostgreSQL ke tujuan dari sumber.

Referensi parameter basis data yang dapat ditranspor

Beberapa parameter mengontrol perilaku ekstensi `pg_transport`. Selanjutnya, Anda dapat menemukan deskripsi parameter ini.

pg_transport.num_workers

Jumlah pekerja yang akan digunakan dalam proses transportasi. Default-nya adalah 3. Nilai yang valid adalah 1–32. Bahkan transportasi basis data terbesar biasanya membutuhkan kurang dari 8 pekerja. Nilai pengaturan ini pada instans DB tujuan digunakan oleh tujuan dan sumber selama transportasi.

pg_transport.timing

Menentukan apakah akan melaporkan informasi waktu selama transportasi. Default-nya adalah `true`, artinya informasi waktu dilaporkan. Kami menyarankan agar Anda membiarkan parameter ini diatur ke `true` agar Anda dapat memantau progresnya. Untuk contoh hasilnya, lihat Mentranspor basis data PostgreSQL ke tujuan dari sumber.

pg_transport.work_mem

Jumlah maksimum memori untuk dialokasikan kepada setiap pekerja. Default-nya adalah 131072 kilobyte (KB) atau 262144 KB (256 MB), tergantung versi PostgreSQL. Nilai minimumnya adalah 64 megabyte (65536 KB). Nilai yang valid adalah kilobyte (KB) sebagai unit basis-2 biner, yaitu 1 KB = 1024 byte.

Transportasi mungkin menggunakan lebih sedikit memori dari yang ditentukan dalam parameter ini. Bahkan transportasi basis data besar biasanya membutuhkan kurang dari 256 MB (262144 KB) memori per pekerja.

Mengekspor data dari instans DB RDS for PostgreSQL ke Amazon S3

Anda dapat mengueri data dari instans DB RDS for PostgreSQL dan mengekspornya langsung ke file yang disimpan dalam bucket Amazon S3. Untuk melakukannya, instal ekstensi RDS for PostgreSQL `aws_s3` terlebih dahulu. Ekstensi ini memberi Anda fungsi yang Anda gunakan untuk mengekspor hasil kueri ke Amazon S3. Berikut ini, Anda dapat mengetahui cara menginstal ekstensi dan cara mengekspor data ke Amazon S3.

Anda dapat mengekspor dari instans DB Aurora Serverless v2 atau yang tersedia. Langkah-langkah ini tidak didukung untuk Aurora Serverless v1.

> **Note**
>
> Ekspor lintas akun ke Amazon S3 tidak didukung.

Semua versi RDS for PostgreSQL yang tersedia saat ini mendukung ekspor data ke Amazon Simple Storage Service. Untuk informasi versi terperinci, lihat Pembaruan Amazon RDS for PostgreSQL dalam Catatan Rilis Amazon RDS for PostgreSQL.

Jika Anda belum menyiapkan bucket untuk ekspor, lihat topik berikut, Panduan Pengguna Amazon Simple Storage Service.

- Menyiapkan Amazon S3
- Membuat bucket

Secara default, data yang diekspor dari RDS untuk PostgreSQL ke Amazon S3 menggunakan enkripsi sisi server dengan file. Kunci yang dikelola AWS Jika Anda menggunakan enkripsi bucket, bucket Amazon S3 harus dienkripsi dengan kunci AWS Key Management Service (AWS KMS) (SSE-KMS). Saat ini, bucket yang dienkripsi dengan kunci terkelola Amazon S3 (SSE-S3) tidak didukung.

> **Note**
>
> Anda dapat menyimpan data snapshot DB ke Amazon S3 menggunakan AWS Management Console AWS CLI,, atau Amazon RDS API. Untuk informasi selengkapnya, lihat Mengekspor data snapshot DB ke Amazon S3.

Topik

Menginstal ekstensi aws_s3

Sebelum Anda dapat menggunakan Amazon Simple Storage Service dengan instans DB RDS for PostgreSQL, Anda perlu menginstal ekstensi `aws_s3`. Ekstensi ini memberikan fungsi untuk mengekspor data dari instans DB RDS for PostgreSQL ke bucket Amazon S3. Ini juga menyediakan fungsi untuk mengimpor data dari Amazon S3. Untuk informasi selengkapnya, lihat Mengimpor data dari Amazon S3 ke instans DB RDS for PostgreSQL. Ekstensi `aws_s3` bergantung pada beberapa fungsi pembantu dalam ekstensi `aws_commons`, yang diinstal secara otomatis bila diperlukan.

Untuk menginstal ekstensi **`aws_s3`**

1. Gunakan psql (atau pgAdmin) untuk terhubung ke instans DB RDS for PostgreSQL sebagai pengguna yang memiliki hak istimewa `rds_superuser`. Jika Anda menyimpan nama default selama proses penyiapan, Anda terhubung sebagai `postgres`.

   ```
   psql --host=111122223333.aws-region.rds.amazonaws.com --port=5432 --
   username=postgres --password
   ```

2. Untuk menginstal ekstensi, jalankan perintah berikut.

   ```
   postgres=> CREATE EXTENSION aws_s3 CASCADE;
   NOTICE: installing required extension "aws_commons"
   CREATE EXTENSION
   ```

3. Untuk memverifikasi bahwa ekstensi sudah diinstal, Anda dapat menggunakan metacommand psql `\dx`.

   ```
   postgres=> \dx
   ```

```
       List of installed extensions
    Name     | Version |   Schema   |                Description
-------------+---------+------------+-------------------------------------------
 aws_commons | 1.2     | public     | Common data types across AWS services
 aws_s3      | 1.1     | public     | AWS S3 extension for importing data from S3
 plpgsql     | 1.0     | pg_catalog | PL/pgSQL procedural language
(3 rows)
```

Fungsi untuk mengimpor data dari Amazon S3 dan mengekspor data ke Amazon S3 kini dapat digunakan.

Verifikasi bahwa versi RDS for PostgreSQL Anda mendukung ekspor ke Amazon S3

Anda dapat memverifikasi bahwa versi RDS for PostgreSQL mendukung ekspor ke Amazon S3 dengan menggunakan perintah `describe-db-engine-versions`. Contoh berikut memverifikasi dukungan untuk versi 10.14.

```
aws rds describe-db-engine-versions --region us-east-1
--engine postgres --engine-version 10.14 | grep s3Export
```

Jika output-nya menyertakan string `"s3Export"`, berarti mesinnya mendukung ekspor Amazon S3. Jika tidak, mesin tidak mendukungnya.

Ikhtisar ekspor data ke Amazon S3

Untuk mengekspor data yang disimpan dalam basis data RDS for PostgreSQL ke bucket Amazon S3, gunakan prosedur berikut.

Untuk mengekspor data RDS for PostgreSQL ke S3

1. Identifikasi jalur file Amazon S3 yang akan digunakan untuk mengekspor data. Untuk detail tentang proses ini, lihat Menentukan jalur file Amazon S3 tujuan ekspor.
2. Berikan izin untuk mengakses bucket Amazon S3.

   Untuk mengekspor data ke file Amazon S3, beri instans DB RDS for PostgreSQL izin untuk mengakses bucket Amazon S3 yang akan digunakan untuk penyimpanan data yang diekspor. Berikut adalah langkah-langkahnya:

   1. Buat kebijakan IAM yang memberikan akses ke bucket Amazon S3 tempat tujuan ekspor.
   2. Buat peran IAM.

3. Lampirkan kebijakan yang Anda buat ke peran yang Anda buat.
4. Tambahkan peran IAM ini ke instans DB.

   Untuk detail tentang proses ini, lihat Menyiapkan akses ke bucket Amazon S3.

3. Identifikasi kueri basis data untuk mendapatkan data. Ekspor data kueri dengan memanggil fungsi `aws_s3.query_export_to_s3`.

   Setelah menyelesaikan tugas persiapan sebelumnya, gunakan fungsi aws_s3.query_export_to_s3 untuk mengekspor hasil kueri ke Amazon S3. Untuk detail tentang proses ini, lihat Mengekspor data kueri menggunakan fungsi aws_s3.query_export_to_s3.

Menentukan jalur file Amazon S3 tujuan ekspor

Tentukan informasi berikut untuk mengidentifikasi lokasi di Amazon S3 tempat Anda ingin mengekspor data:

- Nama bucket – Bucket adalah kontainer untuk objek atau file Amazon S3.

  Untuk informasi selengkapnya tentang menyimpan data dengan Amazon S3, lihat Membuat bucket dan Melihat objek dalam Panduan Pengguna Amazon Simple Storage Service.
- Jalur file – Jalur file mengidentifikasi tempat penyimpanan data yang diekspor dalam bucket Amazon S3. Jalur file terdiri atas:
  - Awalan jalur opsional yang mengidentifikasi jalur folder virtual.
  - Awalan file yang mengidentifikasi satu atau beberapa file yang akan disimpan. Ekspor yang lebih besar disimpan dalam beberapa file, masing-masing berukuran maksimum sekitar 6 GB. Nama file tambahan memiliki awalan file yang sama, tetapi dengan penambahan `_part`*XX*. *XX* mewakili 2, lalu 3, dan seterusnya.

  Misalnya, jalur file dengan folder `exports` dan awalan file `query-1-export` adalah `/exports/query-1-export`.
- AWS Wilayah (opsional) - AWS Wilayah tempat bucket Amazon S3 berada.

> **Note**
>
> Saat ini, AWS Wilayah harus sama dengan wilayah instans yang mengekspor.

Untuk daftar nama AWS Wilayah dan nilai terkait, lihatWilayah, Zona Ketersediaan, dan Zona Lokal.

Untuk menyimpan informasi file Amazon S3 tentang lokasi penyimpanan file yang diekspor, Anda dapat menggunakan fungsi aws_commons.create_s3_uri untuk membuat struktur komposit `aws_commons._s3_uri_1` sebagai berikut.

```
psql=> SELECT aws_commons.create_s3_uri(
   'sample-bucket',
   'sample-filepath',
   'us-west-2'
) AS s3_uri_1 \gset
```

Kemudian, berikan nilai `s3_uri_1` ini sebagai parameter untuk memanggil fungsi aws_s3.query_export_to_s3. Sebagai contoh, lihat Mengekspor data kueri menggunakan fungsi aws_s3.query_export_to_s3.

Menyiapkan akses ke bucket Amazon S3

Untuk mengekspor data ke Amazon S3, berikan izin kepada instans DB PostgreSQL Anda untuk mengakses bucket Amazon S3 yang akan dimasuki file.

Untuk melakukannya, gunakan prosedur berikut.

Untuk memberi instans DB PostgreSQL akses ke Amazon S3 melalui peran IAM

1. Buat kebijakan IAM.

   Kebijakan ini memberikan izin kepada bucket dan objek yang memungkinkan instans DB PostgreSQL Anda mengakses Amazon S3.

   Sebagai bagian dari pembuatan kebijakan ini, lakukan langkah-langkah berikut:

   a. Sertakan tindakan yang diperlukan berikut dalam kebijakan untuk mengizinkan transfer file dari instans DB PostgreSQL ke bucket Amazon S3:

      - `s3:PutObject`
      - `s3:AbortMultipartUpload`

b. Sertakan Amazon Resource Name (ARN) yang mengidentifikasi bucket dan objek Amazon S3 dalam bucket. Format ARN untuk mengakses Amazon S3 adalah: `arn:aws:s3:::`*`your-s3-bucket`*`/*`

Untuk informasi selengkapnya tentang cara membuat kebijakan IAM untuk Amazon RDS for PostgreSQL, lihat Membuat dan menggunakan kebijakan IAM untuk akses basis data IAM. Lihat juga Tutorial: Membuat dan melampirkan kebijakan yang dikelola pelanggan pertama Anda di Panduan Pengguna IAM.

AWS CLI Perintah berikut membuat kebijakan IAM bernama `rds-s3-export-policy` dengan opsi ini. Ini memberikan akses ke bucket bernama `your-s3-bucket`.

> **Warning**
>
> Sebaiknya Anda menyiapkan basis data Anda dengan VPC privat yang memiliki kebijakan titik akhir yang dikonfigurasi untuk mengakses bucket tertentu. Untuk informasi selengkapnya, lihat Menggunakan kebijakan titik akhir untuk Amazon S3 di Panduan Pengguna Amazon VPC.
> Kami sangat menyarankan Anda agar tidak membuat kebijakan dengan akses semua sumber daya. Akses ini dapat menjadi ancaman bagi data keamanan. Jika Anda membuat kebijakan yang memberi akses `S3:PutObject` ke semua sumber daya menggunakan `"Resource":"*"`, pengguna yang memiliki hak istimewa ekspor dapat mengekspor data ke semua bucket di akun Anda. Selain itu, pengguna dapat mengekspor data ke bucket yang dapat ditulis secara publik Wilayah AWS Anda.

Setelah Anda membuat kebijakan, catat Amazon Resource Name (ARN) dari kebijakan tersebut. Anda memerlukan ARN ini untuk langkah berikutnya ketika Anda melampirkan kebijakan ke peran IAM.

```
aws iam create-policy  --policy-name rds-s3-export-policy  --policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Sid": "s3export",
         "Action": [
           "s3:PutObject",
           "s3:AbortMultipartUpload"
```

```
        ],
        "Effect": "Allow",
        "Resource": [
          "arn:aws:s3:::your-s3-bucket/*"
        ]
      }
    ]
  }'
```

2. Buat peran IAM.

   Lakukan langkah ini agar Amazon RDS dapat mengambil peran IAM ini atas nama Anda untuk mengakses bucket Amazon S3. Untuk informasi selengkapnya, lihat Membuat peran untuk mendelegasikan izin ke pengguna IAM dalam Panduan Pengguna IAM.

   Sebaiknya gunakan kunci konteks kondisi global `aws:SourceArn` dan `aws:SourceAccount` dalam kebijakan berbasis sumber daya untuk membatasi izin layanan ke sumber daya tertentu. Ini adalah cara paling efektif untuk melindungi dari masalah deputi yang membingungkan.

   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.

   - Gunakan `aws:SourceArn` jika Anda menginginkan akses lintas layanan untuk satu sumber daya.
   - Gunakan `aws:SourceAccount` jika Anda ingin mengizinkan sumber daya apa pun di akun tersebut dikaitkan dengan penggunaan lintas layanan.

   Dalam kebijakan, pastikan untuk menggunakan kunci konteks kondisi global `aws:SourceArn` dengan ARN penuh sumber daya. Contoh berikut menunjukkan bagaimana melakukannya dengan menggunakan AWS CLI perintah untuk membuat peran bernama`rds-s3-export-role`.

   Example

   Untuk Linux, macOS, atau Unix:

```
aws iam create-role  \
    --role-name rds-s3-export-role  \
    --assume-role-policy-document '{
```

```
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
           "Service": "rds.amazonaws.com"
         },
        "Action": "sts:AssumeRole",
        "Condition": {
            "StringEquals": {
               "aws:SourceAccount": "111122223333",
               "aws:SourceArn": "arn:aws:rds:us-east-1:111122223333:db:dbname"
               }
            }
      }
    ]
  }'
```

Untuk Windows:

```
aws iam create-role  ^
    --role-name rds-s3-export-role  ^
    --assume-role-policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
            "Service": "rds.amazonaws.com"
          },
         "Action": "sts:AssumeRole",
         "Condition": {
             "StringEquals": {
                "aws:SourceAccount": "111122223333",
                "aws:SourceArn": "arn:aws:rds:us-east-1:111122223333:db:dbname"
                }
             }
       }
     ]
   }'
```

3. Lampirkan kebijakan IAM yang Anda buat ke peran IAM yang Anda buat.

AWS CLI Perintah berikut melampirkan kebijakan yang dibuat sebelumnya ke peran bernama `rds-s3-export-role`. Ganti *`your-policy-arn`* dengan ARN kebijakan yang Anda catat di langkah sebelumnya.

```
aws iam attach-role-policy  --policy-arn your-policy-arn  --role-name rds-s3-
export-role
```

4. Tambahkan peran IAM ke instans DB. Anda melakukannya dengan menggunakan AWS Management Console atau AWS CLI, seperti yang dijelaskan berikut.

Konsol

Untuk menambahkan peran IAM untuk instans DB PostgreSQL menggunakan konsol

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Pilih nama instans DB PostgreSQL untuk menampilkan detailnya.
3. Di tab Konektivitas & keamanan, di bagian Kelola peran IAM, pilih peran yang akan ditambahkan pada bagian Tambahkan peran IAM ke instans ini.
4. Di bagian Fitur, pilih s3Export.
5. Pilih Tambahkan peran.

AWS CLI

Untuk menambahkan peran IAM untuk instans DB PostgreSQL menggunakan CLI

- Gunakan perintah berikut untuk menambahkan peran ke instans DB PostgreSQL bernama `my-db-instance`. Ganti *`your-role-arn`* dengan ARN peran yang Anda catat pada langkah sebelumnya. Gunakan `s3Export` untuk nilai opsi `--feature-name`.

  Example

  Untuk Linux, macOS, atau Unix:

```
aws rds add-role-to-db-instance \
   --db-instance-identifier my-db-instance \
   --feature-name s3Export \
   --role-arn your-role-arn   \
```

```
    --region your-region
```

Untuk Windows:

```
aws rds add-role-to-db-instance ^
    --db-instance-identifier my-db-instance ^
    --feature-name s3Export ^
    --role-arn your-role-arn ^
    --region your-region
```

Mengekspor data kueri menggunakan fungsi aws_s3.query_export_to_s3

Ekspor data PostgreSQL Anda ke Amazon S3 dengan memanggil fungsi aws_s3.query_export_to_s3.

Topik

- Prasyarat
- Memanggil aws_s3.query_export_to_s3
- Mengekspor ke file CSV yang menggunakan pembatas kustom
- Mengekspor ke file biner dengan pengodean

Prasyarat

Sebelum menggunakan fungsi `aws_s3.query_export_to_s3`, pastikan untuk melengkapi prasyarat berikut:

- Instal ekstensi PostgreSQL yang diperlukan seperti yang dijelaskan di Ikhtisar ekspor data ke Amazon S3.
- Tentukan tempat untuk mengekspor data Anda ke Amazon S3 seperti yang dijelaskan di Menentukan jalur file Amazon S3 tujuan ekspor.
- Pastikan bahwa instans DB memiliki akses ekspor ke Amazon S3 seperti yang dijelaskan di Menyiapkan akses ke bucket Amazon S3.

Contoh berikut menggunakan tabel basis data yang disebut `sample_table`. Contoh ini mengekspor data ke dalam bucket bernama `sample-bucket`. Contoh tabel dan data dibuat dengan pernyataan SQL berikut di psql.

```
psql=> CREATE TABLE sample_table (bid bigint PRIMARY KEY, name varchar(80));
psql=> INSERT INTO sample_table (bid,name) VALUES (1, 'Monday'), (2,'Tuesday'), (3,
 'Wednesday');
```

Memanggil aws_s3.query_export_to_s3

Berikut ini adalah cara-cara dasar untuk memanggil fungsi aws_s3.query_export_to_s3.

Contoh ini menggunakan variabel `s3_uri_1` untuk mengidentifikasi struktur berisi informasi yang mengidentifikasi file Amazon S3. Gunakan fungsi aws_commons.create_s3_uri untuk membuat struktur.

```
psql=> SELECT aws_commons.create_s3_uri(
   'sample-bucket',
   'sample-filepath',
   'us-west-2'
) AS s3_uri_1 \gset
```

Meskipun parameter bervariasi untuk dua panggilan fungsi `aws_s3.query_export_to_s3` berikut, hasilnya sama untuk contoh ini. Semua baris dari tabel `sample_table` diekspor ke dalam bucket yang disebut `sample-bucket`.

```
psql=> SELECT * FROM aws_s3.query_export_to_s3('SELECT * FROM
 sample_table', :'s3_uri_1');

psql=> SELECT * FROM aws_s3.query_export_to_s3('SELECT * FROM
 sample_table', :'s3_uri_1', options :='format text');
```

Parameternya dijelaskan sebagai berikut:

- `'SELECT * FROM sample_table'` – Parameter pertama adalah string teks wajib yang berisi kueri SQL. Mesin PostgreSQL menjalankan kueri ini. Hasil kueri disalin ke bucket S3 yang diidentifikasi dalam parameter lain.
- `:'s3_uri_1'` – Parameter ini adalah struktur yang mengidentifikasi file Amazon S3. Contoh ini menggunakan variabel untuk mengidentifikasi struktur yang dibuat sebelumnya. Anda dapat membuat struktur dengan menyertakan baris panggilan fungsi `aws_commons.create_s3_uri` sebaris dalam panggilan fungsi `aws_s3.query_export_to_s3` sebagai berikut.

  ```
  SELECT * from aws_s3.query_export_to_s3('select * from sample_table',
  ```

```
   aws_commons.create_s3_uri('sample-bucket', 'sample-filepath', 'us-west-2')
);
```

- `options :='format text'` – Parameter `options` adalah string teks opsional yang berisi argumen `COPY` PostgreSQL. Proses penyalinan menggunakan argumen dan format perintah PostgreSQL COPY.

Jika file yang ditentukan tidak ada dalam bucket Amazon S3, file tersebut akan dibuat. Jika file sudah ada, file tersebut akan ditimpa. Sintaks untuk mengakses data yang diekspor di Amazon S3 adalah sebagai berikut.

```
s3-region://bucket-name[/path-prefix]/file-prefix
```

Ekspor yang lebih besar disimpan dalam beberapa file, masing-masing berukuran maksimum sekitar 6 GB. Nama file tambahan memiliki awalan file yang sama, tetapi dengan penambahan `_part`*XX*. *XX* mewakili 2, lalu 3, dan seterusnya. Sebagai contoh, misalkan Anda menentukan jalur tempat Anda menyimpan file data sebagai berikut.

```
s3-us-west-2://my-bucket/my-prefix
```

Jika ekspor harus membuat tiga file data, bucket Amazon S3 berisi file data berikut.

```
s3-us-west-2://my-bucket/my-prefix
s3-us-west-2://my-bucket/my-prefix_part2
s3-us-west-2://my-bucket/my-prefix_part3
```

Untuk referensi selengkapnya tentang fungsi ini dan cara lain untuk memanggilnya, lihat aws_s3.query_export_to_s3. Untuk informasi selengkapnya tentang cara mengakses file di Amazon S3, buka Melihat objek dalam Panduan Pengguna Amazon Simple Storage Service.

Mengekspor ke file CSV yang menggunakan pembatas kustom

Contoh berikut menunjukkan cara memanggil fungsi aws_s3.query_export_to_s3 untuk mengekspor data ke file yang menggunakan pembatas kustom. Contoh ini menggunakan argumen perintah PostgreSQL COPY untuk menetapkan format nilai yang dipisahkan koma (CSV) dan pembatas titik dua (:).

```
SELECT * from aws_s3.query_export_to_s3('select * from basic_test', :'s3_uri_1',
 options :='format csv, delimiter $$:$$');
```

Mengekspor ke file biner dengan pengodean

Contoh berikut menunjukkan cara memanggil fungsi aws_s3.query_export_to_s3 untuk mengekspor data ke file biner yang memiliki pengodean Windows-1253.

```
SELECT * from aws_s3.query_export_to_s3('select * from basic_test', :'s3_uri_1',
 options :='format binary, encoding WIN1253');
```

Memecahkan masalah akses ke Amazon S3

Jika Anda mengalami masalah koneksi saat mencoba mengekspor data ke Amazon S3, pertama pastikan aturan akses keluar untuk grup keamanan VPC yang terkait dengan instans DB Anda mengizinkan konektivitas jaringan. Khususnya, grup keamanan harus memiliki aturan yang mengizinkan instans DB mengirim lalu lintas TCP ke port 443 dan ke alamat IPv4 mana pun (0.0.0.0/0). Untuk informasi selengkapnya, lihat Memberikan akses ke instans DB di VPC Anda dengan membuat grup keamanan.

Lihat juga rekomendasi berikut:

- Memecahkan masalah identitas dan akses Amazon RDS
- Memecahkan Masalah Amazon S3 di Panduan Pengguna Amazon Simple Storage Service
- Memecahkan Masalah Amazon S3 dan IAM di Panduan Pengguna IAM

Referensi fungsi

Fungsi

- aws_s3.query_export_to_s3
- aws_commons.create_s3_uri

aws_s3.query_export_to_s3

Mengekspor hasil kueri PostgreSQL ke bucket Amazon S3. Ekstensi `aws_s3` memberikan fungsi `aws_s3.query_export_to_s3`.

Dua parameter yang dibutuhkan adalah `query` dan `s3_info`. Parameter ini menentukan kueri yang akan diekspor dan mengidentifikasi bucket Amazon S3 tempat tujuan ekspor. Parameter opsional

yang disebut `options` disediakan untuk menentukan berbagai parameter ekspor. Sebagai contoh penggunaan fungsi `aws_s3.query_export_to_s3`, lihat Mengekspor data kueri menggunakan fungsi aws_s3.query_export_to_s3.

Sintaksis

```
aws_s3.query_export_to_s3(
    query text,
    s3_info aws_commons._s3_uri_1,
    options text,
    kms_key text
)
```

Parameter input

query

String teks yang diperlukan yang berisi kueri SQL yang dijalankan mesin PostgreSQL. Hasil kueri ini disalin ke bucket S3 yang diidentifikasi dalam parameter `s3_info`.

s3_info

Jenis komposit `aws_commons._s3_uri_1` yang berisi informasi tentang objek S3 berikut:

- `bucket` – Nama bucket Amazon S3 yang akan diisi file.
- `file_path` – Nama dan jalur file Amazon S3.
- `region`— AWS Wilayah tempat ember berada. Untuk daftar nama AWS Wilayah dan nilai terkait, lihatWilayah, Zona Ketersediaan, dan Zona Lokal.

  Saat ini, nilai ini harus AWS Wilayah yang sama dengan instans yang mengekspor. Defaultnya adalah AWS Wilayah instance yang mengekspor.

Untuk membuat struktur komposit `aws_commons._s3_uri_1`, lihat fungsi aws_commons.create_s3_uri.

options

String teks opsional yang berisi argumen untuk perintah `COPY` PostgreSQL. Argumen ini menentukan cara menyalin data saat diekspor. Untuk detail selengkapnya, lihat Dokumentasi PostgreSQL COPY.

Parameter input alternatif

Untuk memudahkan pengujian, Anda dapat menggunakan serangkaian parameter yang diperluas, bukan parameter `s3_info`. Berikut ini adalah variasi sintaks tambahan untuk fungsi `aws_s3.query_export_to_s3`.

Alih-alih menggunakan parameter `s3_info` untuk mengidentifikasi file Amazon S3, gunakan kombinasi parameter `bucket`, `file_path`, dan `region`.

```
aws_s3.query_export_to_s3(
    query text,
    bucket text,
    file_path text,
    region text,
    options text,
)
```

query

String teks yang diperlukan yang berisi kueri SQL yang dijalankan mesin PostgreSQL. Hasil kueri ini disalin ke bucket S3 yang diidentifikasi dalam parameter `s3_info`.

bucket

String teks yang diperlukan yang berisi nama bucket Amazon S3 yang berisi file.

file_path

String teks yang diperlukan yang berisi nama file Amazon S3 beserta jalurnya.

region

String teks opsional yang berisi AWS Wilayah tempat bucket berada. Untuk daftar nama AWS Wilayah dan nilai terkait, lihatWilayah, Zona Ketersediaan, dan Zona Lokal.

Saat ini, nilai ini harus AWS Wilayah yang sama dengan instans yang mengekspor. Defaultnya adalah AWS Wilayah instance yang mengekspor.

options

String teks opsional yang berisi argumen untuk perintah `COPY` PostgreSQL. Argumen ini menentukan cara menyalin data saat diekspor. Untuk detail selengkapnya, lihat Dokumentasi PostgreSQL COPY.

Parameter output

```
aws_s3.query_export_to_s3(
    OUT rows_uploaded bigint,
    OUT files_uploaded bigint,
    OUT bytes_uploaded bigint
)
```

rows_uploaded

Jumlah baris tabel yang berhasil diunggah ke Amazon S3 untuk kueri tertentu.

files_uploaded

Jumlah file yang diunggah ke Amazon S3. File dibuat dalam ukuran kira-kira 6 GB. Setiap file tambahan yang dibuat memiliki `_part`*XX* yang ditambahkan pada namanya. *XX* mewakili 2, kemudian 3, dan seterusnya sesuai kebutuhan.

bytes_uploaded

Jumlah total byte yang diunggah ke Amazon S3.

Contoh-contoh

```
psql=> SELECT * from aws_s3.query_export_to_s3('select * from sample_table', 'sample-
bucket', 'sample-filepath');
psql=> SELECT * from aws_s3.query_export_to_s3('select * from sample_table', 'sample-
bucket', 'sample-filepath','us-west-2');
psql=> SELECT * from aws_s3.query_export_to_s3('select * from sample_table', 'sample-
bucket', 'sample-filepath','us-west-2','format text');
```

aws_commons.create_s3_uri

Membuat struktur `aws_commons._s3_uri_1` untuk menyimpan informasi file Amazon S3. Gunakan hasil dari fungsi `aws_commons.create_s3_uri` dalam parameter `s3_info` dari fungsi aws_s3.query_export_to_s3. Untuk contoh penggunaan fungsi `aws_commons.create_s3_uri`, lihat Menentukan jalur file Amazon S3 tujuan ekspor.

Sintaksis

```
aws_commons.create_s3_uri(
```

```
    bucket text,
    file_path text,
    region text
)
```

Parameter input

bucket

String teks yang diperlukan yang berisi nama bucket Amazon S3 untuk file tersebut.

file_path

String teks yang diperlukan yang berisi nama file Amazon S3 beserta jalurnya.

region

String teks yang diperlukan yang berisi AWS Wilayah tempat file tersebut berada. Untuk daftar nama AWS Wilayah dan nilai terkait, lihatWilayah, Zona Ketersediaan, dan Zona Lokal.

Memanggil AWS Lambda fungsi dari

AWS Lambda adalah layanan komputasi berbasis peristiwa yang memungkinkan Anda menjalankan kode tanpa menyediakan atau mengelola server. Ini tersedia untuk digunakan dengan banyak AWS layanan, termasuk . Misalnya, Anda dapat menggunakan fungsi Lambda untuk memproses pemberitahuan peristiwa dari basis data, atau memuat data dari file setiap kali file baru diunggah ke Amazon S3. Untuk mempelajari lebih lanjut tentang Lambda, lihat Apa itu? AWS Lambda di Panduan AWS Lambda Pengembang.

> **Note**
>
> Memanggil AWS Lambda fungsi didukung dalam RDS ini untuk versi PostgreSQL:
>
> - Semua PostgreSQL versi 16
> - Semua PostgreSQL versi 15
> - PostgreSQL 14.1 dan versi minor yang lebih tinggi
> - PostgreSQL 13.2 dan versi minor yang lebih tinggi
> - PostgreSQL 12.6 dan versi minor yang lebih tinggi

Berikut ini, Anda dapat menemukan ringkasan langkah-langkah yang diperlukan.

Untuk informasi selengkapnya tentang fungsi Lambda, lihat Mulai menggunakan Lambda dan Dasar-dasar AWS Lambda di Panduan Developer AWS Lambda .

Topik

Langkah 1: Konfigurasikan untuk koneksi keluar ke AWS Lambda

Fungsi Lambda selalu berjalan di dalam VPC Amazon yang dimiliki oleh layanan. AWS Lambda Lambda menerapkan akses jaringan dan aturan keamanan untuk VPC ini dan mempertahankan serta memantau VPC secara otomatis. Instans DB RDS for PostgreSQL Anda mengirimkan lalu lintas jaringan ke VPC layanan Lambda. Cara Anda mengonfigurasi ini bergantung pada apakah instans DB Anda bersifat publik atau pribadi.

- Public RDS untuk instans PostgreSQL DB — Instans DB instans DB adalah publik jika terletak di subnet publik di VPC Anda, dan jika properti "" instans adalah. PubliclyAccessible `true` Untuk menemukan nilai properti ini, Anda dapat menggunakan describe-db-instances AWS CLI perintah. Atau, Anda dapat menggunakan AWS Management Console untuk membuka tab Konektivitas & keamanan dan memeriksa apakah opsi Dapat diakses publik adalah Ya. Untuk memverifikasi bahwa instans ini ada di subnet publik VPC, Anda dapat menggunakan AWS Management Console atau AWS CLI.

  Untuk mengatur akses ke Lambda, Anda menggunakan AWS Management Console atau AWS CLI untuk membuat aturan keluar pada grup keamanan VPC Anda. Aturan outbound menentukan bahwa TCP dapat menggunakan port 443 untuk mengirim paket ke alamat IPv4 (0.0.0.0/0) mana pun.
- Private RDS untuk instance PostgreSQL DB — Dalam hal ini, properti "" instance ada atau berada di subnet pribadi. PubliclyAccessible `false` Agar instans dapat berfungsi dengan Lambda, Anda dapat menggunakan gateway Network Address Translation (NAT). Untuk informasi selengkapnya, silakan lihat Gateway NAT. Atau, Anda dapat mengonfigurasi VPC dengan titik akhir VPC untuk Lambda. Untuk informasi selengkapnya, lihat Titik akhir VPC di Panduan Pengguna Amazon VPC. Titik akhir ini merespons panggilan yang dilakukan oleh instans DB RDS for PostgreSQL ke fungsi Lambda Anda. Titik akhir VPC menggunakan resolusi DNS pribadinya sendiri. RDS for PostgreSQL tidak dapat menggunakan titik akhir VPC Lambda hingga Anda mengubah nilai `rds.custom_dns_resolution` dari default-nya 0 (tidak aktif) menjadi 1. Untuk melakukannya:
  - Buat grup parameter DB kustom.
  - Ubah nilai parameter `rds.custom_dns_resolution` dari default `0` ke `1`.
  - Ubah instans DB Anda untuk menggunakan grup parameter DB kustom.
  - Boot ulang instans agar parameter yang diubah dapat diterapkan.

VPC Anda sekarang dapat berinteraksi dengan AWS Lambda VPC di tingkat jaringan. Selanjutnya, konfigurasikan izin menggunakan IAM.

Langkah 2: Konfigurasikan IAM untuk danAWS Lambda

Menginvokasi fungsi Lambda dari instans DB RDS for PostgreSQL memerlukan hak istimewa tertentu. Untuk mengonfigurasi hak istimewa yang diperlukan, sebaiknya Anda membuat kebijakan IAM yang memungkinkan invokasi fungsi Lambda, menetapkan kebijakan ke peran, dan kemudian menerapkan peran tersebut ke instans DB Anda. Pendekatan ini memberikan hak istimewa instans DB untuk menginvokasi fungsi Lambda yang ditentukan atas nama Anda. Langkah-langkah berikut menunjukkan cara melakukannya dengan menggunakan AWS CLI.

Untuk mengonfigurasi izin IAM untuk menggunakan instans Amazon RDS dengan Lambda

1. Gunakan AWS CLI perintah create-policy untuk membuat kebijakan IAM yang memungkinkan Aurora untuk menjalankan fungsi Lambda yang ditentukan. (ID pernyataan (Sid) adalah deskripsi opsional untuk pernyataan kebijakan Anda dan tidak berpengaruh pada penggunaan.) Kebijakan ini memberi instans DB izin minimum yang diperlukan untuk menginvokasi fungsi Lambda yang ditentukan.

   ```
   aws iam create-policy  --policy-name rds-lambda-policy --policy-document '{
       "Version": "2012-10-17",
       "Statement": [
           {
           "Sid": "AllowAccessToExampleFunction",
           "Effect": "Allow",
           "Action": "lambda:InvokeFunction",
           "Resource": "arn:aws:lambda:aws-region:444455556666:function:my-function"
           }
       ]
   }'
   ```

   Sebagai alternatif, Anda dapat menggunakan kebijakan `AWSLambdaRole` yang ditentukan sebelumnya yang memungkinkan Anda menginvokasi fungsi Lambda apa pun. Untuk informasi selengkapnya, lihat Kebijakan IAM berbasis identitas untuk Lambda

2. Gunakan AWS CLI perintah create-role untuk membuat peran IAM yang dapat diasumsikan oleh kebijakan saat runtime.

   ```
   aws iam create-role  --role-name rds-lambda-role --assume-role-policy-document '{
       "Version": "2012-10-17",
       "Statement": [
           {
           "Effect": "Allow",
   ```

```
        "Principal": {
            "Service": "rds.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
        }
    ]
}'
```

3. Terapkan kebijakan ke peran dengan menggunakan attach-role-policy AWS CLI perintah.

```
aws iam attach-role-policy \
    --policy-arn arn:aws:iam::444455556666:policy/rds-lambda-policy \
    --role-name rds-lambda-role --region aws-region
```

4. AWS CLI Langkah terakhir ini memungkinkan pengguna basis data instans DB menginvokasi fungsi Lambda.

```
aws rds add-role-to-db-instance \
       --db-instance-identifier my-instance-name \
       --feature-name Lambda \
       --role-arn  arn:aws:iam::444455556666:role/rds-lambda-role   \
       --region aws-region
```

Setelah menyelesaikan konfigurasi VPC dan IAM, Anda sekarang dapat menginstal ekstensi `aws_lambda`. (Perhatikan bahwa Anda dapat menginstal ekstensi kapan saja, tetapi sebelum menyiapkan dukungan VPC dan hak istimewa IAM yang benar, ekstensi `aws_lambda` tidak menambahkan apa pun ke kapabilitas instans DB RDS for PostgreSQL.)

Langkah 3: Instal ekstensi **`aws_lambda`** untuk instans DB RDS for PostgreSQL

Ekstensi ini memberi instans DB RDS for PostgreSQL kemampuan untuk memanggil fungsi Lambda dari PostgreSQL.

Untuk menginstal ekstensi **`aws_lambda`** di instans DB RDS for PostgreSQL Anda

Gunakan baris perintah `psql` PostgreSQL atau alat pgAdmin untuk terhubung ke instans DB RDS for PostgreSQL.

1. Hubungkan ke instans DB RDS for PostgreSQL sebagai pengguna dengan hak istimewa `rds_superuser`. Pengguna `postgres` default ditampilkan dalam contoh.

```
psql -h instance.444455556666.aws-region.rds.amazonaws.com -U postgres -p 5432
```

2. Instal ekstensi `aws_lambda`. Ekstensi `aws_commons` juga diperlukan. Ini memberikan fungsi pembantu `aws_lambda` dan berbagai ekstensi Aurora lainnya untuk PostgreSQL. Jika belum ada di instans DB RDS for PostgreSQL, ekstensi akan diinstal dengan `aws_lambda` seperti yang ditunjukkan sebagai berikut.

```
CREATE EXTENSION IF NOT EXISTS aws_lambda CASCADE;
NOTICE:  installing required extension "aws_commons"
CREATE EXTENSION
```

Ekstensi `aws_lambda` diinstal di instans DB . Anda sekarang dapat membuat struktur kemudahan untuk menginvokasi fungsi Lambda.

Langkah 4: Gunakan fungsi pembantu Lambda dengan instans DB RDS for PostgreSQL (Opsional)

Anda dapat menggunakan fungsi pembantu di ekstensi `aws_commons` untuk menyiapkan entitas yang dapat diinvokasi dengan lebih mudah dari PostgreSQL. Untuk melakukannya, Anda harus memiliki informasi berikut tentang fungsi Lambda:

- Nama fungsi – Nama, Amazon Resource Name (ARN), versi, atau alias fungsi Lambda. Kebijakan IAM yang dibuat Langkah 2: Konfigurasikan IAM untuk instans dan Lambda memerlukan ARN, jadi kami sarankan Anda menggunakan ARN fungsi Anda.
- AWS Wilayah - (Opsional) AWS Wilayah tempat fungsi Lambda berada jika tidak berada di Wilayah yang sama dengan Kluster PostgreSQL DB Aurora Anda RDS untuk instans .

Untuk menyimpan informasi nama fungsi Lambda, Anda menggunakan fungsi aws_commons.create_lambda_function_arn. Fungsi pembantu ini menciptakan struktur komposit `aws_commons._lambda_function_arn_1` dengan detail yang dibutuhkan oleh fungsi invokasi. Berikut ini, Anda dapat menemukan tiga pendekatan alternatif untuk menyiapkan struktur komposit ini.

```
SELECT aws_commons.create_lambda_function_arn(
```

```
   'my-function',
   'aws-region'
) AS aws_lambda_arn_1 \gset
```

```
SELECT aws_commons.create_lambda_function_arn(
   '111122223333:function:my-function',
   'aws-region'
) AS lambda_partial_arn_1 \gset
```

```
SELECT aws_commons.create_lambda_function_arn(
   'arn:aws:lambda:aws-region:111122223333:function:my-function'
) AS lambda_arn_1 \gset
```

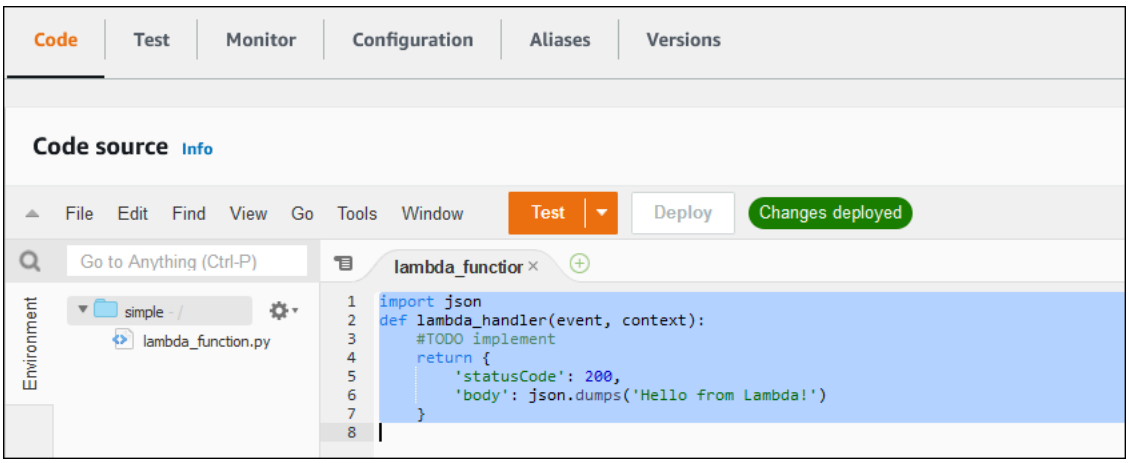
Salah satu dari nilai ini dapat digunakan dalam panggilan ke fungsi aws_lambda.invoke. Sebagai contoh, lihat Langkah 5: Invokasi fungsi Lambda dari instans DB RDS for PostgreSQL Anda.

Langkah 5: Invokasi fungsi Lambda dari instans DB RDS for PostgreSQL Anda

Fungsi `aws_lambda.invoke` berperilaku sinkron atau asinkron, bergantung pada `invocation_type`. Dua alternatif untuk parameter ini adalah `RequestResponse` (default) dan `Event`, sebagai berikut.

- **`RequestResponse`** – Jenis invokasi ini sinkron. Ini adalah perilaku default saat panggilan dilakukan tanpa menentukan jenis invokasi. Payload respons mencakup hasil dari fungsi `aws_lambda.invoke`. Gunakan jenis invokasi ini jika alur kerja Anda perlu menerima hasil dari fungsi Lambda sebelum melanjutkan.
- **`Event`** – Jenis invokasi ini asinkron. Respons tidak mencakup payload yang berisi hasil. Gunakan jenis invokasi ini jika alur kerja Anda tidak memerlukan hasil dari fungsi Lambda untuk melanjutkan pemrosesan.

Sebagai pengujian sederhana terhadap pengaturan Anda, Anda dapat terhubung ke instans DB menggunakan `psql` dan menginvokasi contoh fungsi dari baris perintah. Misalkan Anda memiliki salah satu fungsi dasar yang disiapkan pada layanan Lambda, seperti fungsi Python sederhana yang diperlihatkan pada tangkapan layar berikut.



Untuk menginvokasi contoh fungsi

1. Hubungkan ke instans DB menggunakan `psql` atau pgAdmin.

   ```
   psql -h instance.444455556666.aws-region.rds.amazonaws.com -U postgres -p 5432
   ```

2. Invokasi fungsi menggunakan ARN-nya.

   ```
   SELECT * from
    aws_lambda.invoke(aws_commons.create_lambda_function_arn('arn:aws:lambda:aws-
   region:444455556666:function:simple', 'us-west-1'), '{"body": "Hello from
    Postgres!"}'::json );
   ```

   Respons-nya terlihat sebagai berikut.

   ```
   status_code |                         payload                         |
    executed_version | log_result
   -------------+---------------------------------------------------------
   +------------------+------------
            200 | {"statusCode": 200, "body": "\"Hello from Lambda!\""} | $LATEST
        |
   (1 row)
   ```

Jika upaya invokasi tidak berhasil, lihat Pesan kesalahan fungsi Lambda .

Langkah 6: Berikan pengguna lain izin untuk menginvokasi fungsi Lambda

Dalam langkah ini, hanya Anda sebagai `rds_superuser` yang dapat menginvokasi fungsi Lambda. Untuk mengizinkan pengguna lain menginvokasi fungsi apa pun yang Anda buat, Anda harus memberi mereka izin.

Untuk memberi pengguna lain izin untuk menginvokasi fungsi Lambda

1. Hubungkan ke instans DB menggunakan `psql` atau pgAdmin.

```
psql -h instance.444455556666.aws-region.rds.amazonaws.com -U postgres -p 5432
```

2. Jalankan perintah SQL berikut:

```
postgres=>  GRANT USAGE ON SCHEMA aws_lambda TO db_username;
GRANT EXECUTE ON ALL FUNCTIONS IN SCHEMA aws_lambda TO db_username;
```

Contoh: Menginvokasi fungsi Lambda dari instans DB RDS for PostgreSQL

Berikut ini, Anda dapat menemukan beberapa contoh pemanggilan fungsi aws_lambda.invoke. Sebagian besar contoh menggunakan struktur komposit `aws_lambda_arn_1` yang Anda buat di Langkah 4: Gunakan fungsi pembantu Lambda dengan instans DB RDS for PostgreSQL (Opsional) untuk menyederhanakan penerusan detail fungsi. Untuk contoh panggilan asinkron, lihat Contoh: Invokasi fungsi Lambda asinkron (Event). Semua contoh lain yang tercantum menggunakan panggilan sinkron.

Untuk mempelajari lebih lanjut tentang jenis invokasi Lambda, lihat Menginvokasi fungsi Lambda di Panduan Developer AWS Lambda . Untuk informasi selengkapnya tentang `aws_lambda_arn_1`, lihat aws_commons.create_lambda_function_arn.

Daftar contoh

- Contoh: Synchronous (RequestResponse) pemanggilan fungsi Lambda
- Contoh: Invokasi fungsi Lambda asinkron (Event)
- Contoh: Menangkap log eksekusi Lambda dalam respons fungsi
- Contoh: Menyertakan konteks klien dalam fungsi Lambda
- Contoh: Menginvokasi fungsi Lambda versi spesifik

Contoh: Synchronous (RequestResponse) pemanggilan fungsi Lambda

Berikut ini adalah dua contoh dari invokasi fungsi Lambda sinkron. Hasil dari panggilan fungsi `aws_lambda.invoke` ini sama.

```
SELECT * FROM aws_lambda.invoke('aws_lambda_arn_1', '{"body": "Hello from
 Postgres!"}'::json);
```

```
SELECT * FROM aws_lambda.invoke('aws_lambda_arn_1', '{"body": "Hello from
 Postgres!"}'::json, 'RequestResponse');
```

Parameternya dijelaskan sebagai berikut:

- `:'aws_lambda_arn_1'` – Parameter ini mengidentifikasi struktur komposit yang dibuat di[Langkah 4: Gunakan fungsi pembantu Lambda dengan instans DB RDS for PostgreSQL (Opsional)](), dengan fungsi pembantu `aws_commons.create_lambda_function_arn`. Anda juga dapat membuat struktur ini sebaris dalam panggilan `aws_lambda.invoke` Anda sebagai berikut.

  ```
  SELECT * FROM aws_lambda.invoke(aws_commons.create_lambda_function_arn('my-function',
   'aws-region'),
  '{"body": "Hello from Postgres!"}'::json
  );
  ```

- `'{"body": "Hello from PostgreSQL!"}'::json` – Payload JSON untuk diteruskan ke fungsi Lambda.
- `'RequestResponse'` – Jenis invokasi Lambda.

Contoh: Invokasi fungsi Lambda asinkron (Event)

Berikut ini adalah contoh invokasi fungsi Lambda asinkron. Jenis invokasi `Event` menjadwalkan invokasi fungsi Lambda dengan payload input yang ditentukan dan segera kembali. Gunakan jenis invokasi `Event` di alur kerja tertentu yang tidak bergantung pada hasil fungsi Lambda.

```
SELECT * FROM aws_lambda.invoke('aws_lambda_arn_1', '{"body": "Hello from
 Postgres!"}'::json, 'Event');
```

Contoh: Menangkap log eksekusi Lambda dalam respons fungsi

Anda dapat menyertakan 4 KB terakhir log eksekusi di respons fungsi dengan menggunakan parameter `log_type` dalam panggilan fungsi `aws_lambda.invoke` Anda. Secara default, parameter ini diatur ke `None`, tetapi Anda dapat menentukan `Tail` untuk menangkap hasil log eksekusi Lambda dalam respons, seperti yang ditunjukkan berikut.

```
SELECT *, select convert_from(decode(log_result, 'base64'), 'utf-8') as log FROM
 aws_lambda.invoke(:'aws_lambda_arn_1', '{"body": "Hello from Postgres!"}'::json,
 'RequestResponse', 'Tail');
```

Atur parameter `log_type` fungsi aws_lambda.invoke ke `Tail` untuk menyertakan log eksekusi dalam respons. Nilai default untuk parameter `log_type` adalah `None`.

`log_result` yang ditampilkan string yang dienkode `base64`. Anda dapat mendekode kontennya menggunakan kombinasi fungsi PostgreSQL `decode` dan `convert_from`.

Untuk informasi selengkapnya tentang `log_type`, lihat aws_lambda.invoke.

Contoh: Menyertakan konteks klien dalam fungsi Lambda

Fungsi `aws_lambda.invoke` memiliki parameter `context` yang dapat Anda gunakan untuk meneruskan informasi yang terpisah dari payload, seperti yang diperlihatkan di bawah.

```
SELECT *, convert_from(decode(log_result, 'base64'), 'utf-8') as log FROM
 aws_lambda.invoke(:'aws_lambda_arn_1', '{"body": "Hello from Postgres!"}'::json,
 'RequestResponse', 'Tail');
```

Untuk menyertakan konteks klien, gunakan objek JSON untuk parameter `context` fungsi aws_lambda.invoke.

Untuk informasi selengkapnya tentang parameter `context`, lihat referensi aws_lambda.invoke.

Contoh: Menginvokasi fungsi Lambda versi spesifik

Anda dapat menentukan versi tertentu dari fungsi Lambda dengan menyertakan parameter `qualifier` dengan panggilan `aws_lambda.invoke`. Berikut ini, Anda dapat menemukan contoh yang melakukan ini menggunakan '*custom_version*' sebagai alias untuk versi.

```
SELECT * FROM aws_lambda.invoke('aws_lambda_arn_1', '{"body": "Hello from
 Postgres!"}'::json, 'RequestResponse', 'None', NULL, 'custom_version');
```

Anda juga dapat menyediakan pengualifikasi fungsi Lambda dengan detail nama fungsi sebagai berikut.

```
SELECT * FROM aws_lambda.invoke(aws_commons.create_lambda_function_arn('my-
function:custom_version', 'us-west-2'),
'{"body": "Hello from Postgres!"}'::json);
```

Untuk informasi selengkapnya tentang `qualifier` dan parameter lainnya, lihat referensi aws_lambda.invoke.

Pesan kesalahan fungsi Lambda

Dalam daftar berikut, Anda dapat menemukan informasi tentang pesan kesalahan, dengan kemungkinan penyebab dan solusi.

- Masalah konfigurasi VPC

  Masalah konfigurasi VPC dapat memunculkan pesan kesalahan berikut saat mencoba menghubungkan:

  ```
  ERROR:  invoke API failed
  DETAIL: AWS Lambda client returned 'Unable to connect to endpoint'.
  CONTEXT:  SQL function "invoke" statement 1
  ```

  Penyebab umum kesalahan ini adalah grup keamanan VPC tidak dikonfigurasi dengan benar. Pastikan Anda memiliki aturan keluar untuk TCP yang terbuka pada di 443 grup keamanan VPC Anda, sehingga VPC Anda dapat terhubung ke VPC Lambda.

  Jika instans DB Anda bersifat pribadi, periksa penyiapan DNS pribadi untuk VPC Anda. Pastikan bahwa Anda mengatur `rds.custom_dns_resolution` parameter ke 1 dan setup AWS PrivateLink seperti yang diuraikan dalamLangkah 1: Konfigurasikan untuk koneksi keluar ke AWS Lambda. Untuk informasi selengkapnya, lihat Titik akhir VPC Antarmuka ().AWS PrivateLink

- Kurangnya izin yang diperlukan untuk menginvokasi fungsi Lambda

  Jika Anda melihat salah satu pesan galat berikut, berarti pengguna (peran) yang menginvokasi fungsi ini tidak memiliki izin yang tepat.

  ```
  ERROR:  permission denied for schema aws_lambda
  ```

```
ERROR:  permission denied for function invoke
```

Pengguna (peran) harus diberi izin khusus untuk menginvokasi fungsi Lambda. Untuk informasi selengkapnya, lihat Langkah 6: Berikan pengguna lain izin untuk menginvokasi fungsi Lambda.

- Penanganan kesalahan yang tidak tepat dalam fungsi Lambda

  Jika fungsi Lambda menampilkan pengecualian selama pemrosesan permintaan, berarti `aws_lambda.invoke` gagal dengan kesalahan PostgreSQL seperti berikut.

```
SELECT * FROM aws_lambda.invoke('aws_lambda_arn_1', '{"body": "Hello from
 Postgres!"}'::json);
ERROR:  lambda invocation failed
DETAIL:  "arn:aws:lambda:us-west-2:555555555555:function:my-function" returned error
 "Unhandled", details: "<Error details string>".
```

  Pastikan untuk menangani kesalahan dalam fungsi Lambda Anda atau di aplikasi PostgreSQL Anda.

AWS Lambdafungsi dan referensi parameter

Berikut ini adalah referensi untuk fungsi dan parameter yang akan digunakan untuk memanggil Lambda dengan PostgreSQL RDS untuk PostgreSQL.

Fungsi dan parameter

- aws_lambda.invoke
- aws_commons.create_lambda_function_arn
- parameter aws_lambda

aws_lambda.invoke

Menjalankan fungsi Lambda untuk instans DB RDS for PostgreSQL.

Untuk detail lebih lanjut tentang memanggil fungsi Lambda, lihat juga Invokasi di Panduan Developer AWS Lambda.

Sintaksis

JSON

```
aws_lambda.invoke(
IN function_name TEXT,
IN payload JSON,
IN region TEXT DEFAULT NULL,
IN invocation_type TEXT DEFAULT 'RequestResponse',
IN log_type TEXT DEFAULT 'None',
IN context JSON DEFAULT NULL,
IN qualifier VARCHAR(128) DEFAULT NULL,
OUT status_code INT,
OUT payload JSON,
OUT executed_version TEXT,
OUT log_result TEXT)
```

```
aws_lambda.invoke(
IN function_name aws_commons._lambda_function_arn_1,
IN payload JSON,
IN invocation_type TEXT DEFAULT 'RequestResponse',
IN log_type TEXT DEFAULT 'None',
IN context JSON DEFAULT NULL,
IN qualifier VARCHAR(128) DEFAULT NULL,
OUT status_code INT,
OUT payload JSON,
OUT executed_version TEXT,
OUT log_result TEXT)
```

JSONB

```
aws_lambda.invoke(
IN function_name TEXT,
IN payload JSONB,
IN region TEXT DEFAULT NULL,
IN invocation_type TEXT DEFAULT 'RequestResponse',
IN log_type TEXT DEFAULT 'None',
IN context JSONB DEFAULT NULL,
IN qualifier VARCHAR(128) DEFAULT NULL,
OUT status_code INT,
OUT payload JSONB,
OUT executed_version TEXT,
OUT log_result TEXT)
```

```
aws_lambda.invoke(
IN function_name aws_commons._lambda_function_arn_1,
IN payload JSONB,
IN invocation_type TEXT DEFAULT 'RequestResponse',
IN log_type TEXT DEFAULT 'None',
IN context JSONB DEFAULT NULL,
IN qualifier VARCHAR(128) DEFAULT NULL,
OUT status_code INT,
OUT payload JSONB,
OUT executed_version TEXT,
OUT log_result TEXT
)
```

Parameter input

function_name

Nama yang mengidentifikasi fungsi Lambda. Nilai tersebut dapat berupa nama fungsi, sebuah ARN, atau ARN parsial. Untuk daftar format yang memungkinkan, lihat Format nama fungsi Lambda dalam Panduan Developer AWS Lambda.

payload

Input untuk fungsi Lambda. Formatnya dapat berupa JSON atau JSONB. Untuk informasi selengkapnya, lihat Jenis JSON dalam dokumentasi PostgreSQL.

region

(Opsional) Wilayah Lambda untuk fungsi tersebut. Secara default, RDS menyelesaikan Wilayah AWS dari ARN penuh di `function_name` atau menggunakan Wilayah instans DB RDS for PostgreSQL. Jika nilai Wilayah ini bertentangan dengan nilai yang disediakan dalam ARN `function_name`, pesan kesalahan akan muncul.

invocation_type

Jenis invokasi fungsi Lambda. Nilai ini peka huruf besar/kecil. Kemungkinan nilainya termasuk yang berikut ini:

- `RequestResponse` – Default. Jenis invokasi untuk fungsi Lambda bersifat sinkron dan menampilkan payload respons dalam hasilnya. Gunakan jenis invokasi `RequestResponse` ketika alur kerja Anda bergantung pada penerimaan hasil fungsi Lambda dengan segera.

- `Event` – Jenis invokasi untuk fungsi Lambda ini bersifat asinkron dan segera kembali tanpa menampilkan payload. Gunakan jenis invokasi `Event` ketika Anda tidak membutuhkan hasil dari fungsi Lambda sebelum alur kerja Anda berlanjut.
- `DryRun` – Jenis invokasi ini menguji akses tanpa menjalankan fungsi Lambda.

log_type

Jenis log Lambda untuk ditampilkan dalam parameter output `log_result`. Nilai ini peka huruf besar/kecil. Kemungkinan nilainya termasuk yang berikut ini:

- Ekor – Parameter output `log_result` yang ditampilkan akan mencakup 4 KB terakhir log eksekusi.
- Tidak Ada – Tidak ada informasi log Lambda yang ditampilkan.

context

Konteks klien dalam format JSON atau JSONB. Kolom yang akan digunakan termasuk `custom` dan `env`.

qualifier

Pengualifikasi yang mengidentifikasi versi fungsi Lambda yang akan diinvokasi. Jika nilai ini bertentangan dengan nilai yang disediakan dalam ARN `function_name`, pesan kesalahan akan muncul.

Parameter output

status_code

Kode respons status HTTP. Untuk informasi selengkapnya, lihat Elemen respons invokasi Lambda di Panduan Developer AWS Lambda.

payload

Informasi yang ditampilkan dari fungsi Lambda yang berjalan. Formatnya berupa JSON atau JSONB.

executed_version

Versi fungsi Lambda yang berjalan.

log_result

Informasi log eksekusi yang ditampilkan jika nilai `log_type` adalah `Tail` ketika fungsi Lambda diinvokasi. Hasilnya berisi 4 KB terakhir log eksekusi yang dikodekan dalam Base64.

aws_commons.create_lambda_function_arn

Membuat struktur `aws_commons._lambda_function_arn_1` untuk menyimpan informasi nama fungsi Lambda. Gunakan hasil fungsi `aws_commons.create_lambda_function_arn` dalam parameter `function_name` dari fungsi aws_lambda.invoke.

Sintaksis

```
aws_commons.create_lambda_function_arn(
    function_name TEXT,
    region TEXT DEFAULT NULL
    )
    RETURNS aws_commons._lambda_function_arn_1
```

Parameter input

function_name

String teks yang diperlukan berisi nama fungsi Lambda. Nilai tersebut dapat berupa nama fungsi, ARN penuh, atau ARN parsial.

region

String teks opsional yang berisi Wilayah AWS tempat fungsi Lambda berada. Untuk daftar nama Wilayah dan nilai terkait, lihat Wilayah, Zona Ketersediaan, dan Zona Lokal.

parameter aws_lambda

Dalam tabel ini, Anda dapat menemukan parameter yang terkait dengan `aws_lambda` fungsi tersebut.

Parameter	Deskripsi
`aws_lambda.connect_timeout_ms`	Ini adalah parameter dinamis dan menetapkan waktu tunggu maksimum saat menghubungkan ke AWS Lambda. Nilai defaultnya adalah`1000`. Nilai yang diizinkan untuk parameter ini adalah 1 - 900000.
`aws_lambda.request_timeout_ms`	Ini adalah parameter dinamis dan menetapkan waktu tunggu maksimum sambil menunggu respons dari AWS Lambda. Nilai

Parameter	Deskripsi
	defaultnya adalah`3000`. Nilai yang diizinkan untuk parameter ini adalah 1 - 900000.
`aws_lambda.endpoint_override`	Menentukan endpoint yang dapat digunakan untuk terhubung ke LambdaAWS. String kosong memilih titik akhir AWS Lambda default untuk wilayah tersebut. Anda harus me-restart database agar perubahan parameter statis ini berlaku.

Tugas DBA umum untuk Amazon RDS for PostgreSQL

Administrator basis data (DBA) melakukan berbagai tugas saat mengelola instans DB Amazon RDS for PostgreSQL. Jika Anda seorang DBA yang sudah terbiasa dengan PostgreSQL, Anda perlu menyadari beberapa perbedaan penting antara menjalankan PostgreSQL di perangkat keras Anda dan RDS for PostgreSQL. Misalnya, karena ini adalah layanan terkelola, Amazon RDS tidak mengizinkan akses shell ke instans DB Anda. Ini berarti Anda tidak memiliki akses langsung ke file `pg_hba.conf` dan file konfigurasi lainnya. Untuk RDS for PostgreSQL, perubahan yang biasanya dilakukan pada file konfigurasi PostgreSQL dari instans on-premise dibuat ke grup parameter DB kustom yang terkait dengan instans DB RDS for PostgreSQL. Untuk informasi selengkapnya, lihat Bekerja dengan grup parameter.

Anda juga tidak dapat mengakses file log dengan cara yang sama seperti yang Anda lakukan dengan instans PostgreSQL on-premise. Untuk mempelajari pencatatan selengkapnya, lihat File log basis data RDS for PostgreSQL.

Sebagai contoh lain, Anda tidak memiliki akses ke akun PostgreSQL `superuser`. Di RDS for PostgreSQL, peran `rds_superuser` merupakan peran yang paling istimewa, dan diberikan ke `postgres` pada saat penyiapan. Meskipun Anda telah terbiasa menggunakan PostgreSQL on-premise atau baru menggunakan RDS for PostgreSQL, sebaiknya Anda memahami peran `rds_superuser`, serta cara bekerja dengan peran, pengguna, grup, dan izin. Untuk informasi selengkapnya, lihat Memahami peran dan izin PostgreSQL.

Berikut ini adalah beberapa tugas DBA umum untuk RDS for PostgreSQL.

Topik

- Kolasi yang didukung di RDS for PostgreSQL
- Memahami peran dan izin PostgreSQL
- Bekerja dengan fitur autovacuum PostgreSQL di Amazon RDS for PostgreSQL
- Bekerja dengan mekanisme pencatatan log yang didukung oleh RDS for PostgreSQL
- Mengelola file sementara dengan PostgreSQL
- Menggunakan pgBadger untuk analisis log dengan PostgreSQL
- Menggunakan PGSnapper untuk memantau PostgreSQL
- Bekerja dengan parameter pada instans DB RDS for PostgreSQL

Kolasi yang didukung di RDS for PostgreSQL

Kolasi adalah seperangkat aturan yang menentukan cara pengurutan dan perbandingan string karakter yang disimpan di basis data. Kolasi memiliki peran mendasar dalam sistem komputer dan dimasukkan sebagai bagian dari sistem operasi. Kolasi berubah dari waktu ke waktu ketika karakter baru ditambahkan ke bahasa atau ketika aturan urutan berubah.

Pustaka kolasi menentukan aturan dan algoritma khusus untuk kolasi. Pustaka kolasi paling populer yang digunakan dalam PostgreSQL adalah GNU C (glibc) dan komponen Internasionalisasi untuk Unicode (ICU). Secara default, RDS for PostgreSQL menggunakan kolasi glibc yang mencakup urutan karakter unicode untuk urutan karakter multi-byte.

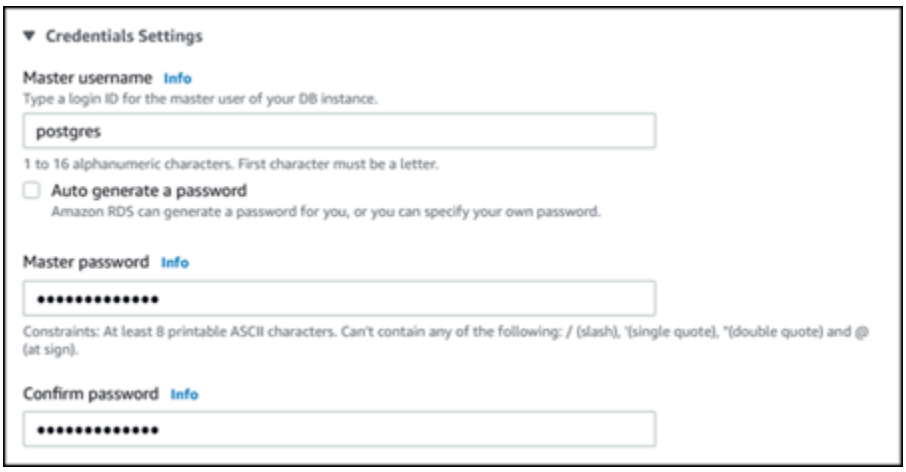
Saat Anda membuat instans DB di RDS for PostgreSQL baru, ini akan memeriksa sistem operasi untuk kolasi yang tersedia. Parameter PostgreSQL dari perintah `CREATE DATABASE`, `LC_COLLATE`, dan `LC_CTYPE` digunakan untuk menentukan kolasi, yang merupakan kolasi default dalam basis data tersebut. Atau, Anda juga dapat menggunakan parameter `LOCALE` di `CREATE DATABASE` untuk menetapkan parameter ini. Parameter ini menentukan kolasi default untuk string karakter dalam basis data dan aturan untuk mengklasifikasikan karakter sebagai huruf, angka, atau simbol. Anda juga dapat memilih kolasi untuk digunakan pada kolom, indeks, atau kueri.

RDS for PostgreSQL bergantung pada pustaka glibc di sistem operasi untuk dukungan kolasi. Instans RDS for PostgreSQL diperbarui secara berkala dengan versi terbaru sistem operasi. Pembaruan ini terkadang menyertakan versi pustaka glibc yang lebih baru. Jarang sekali versi glibc yang lebih baru mengubah tata urutan atau kolasi beberapa karakter, yang dapat menyebabkan data diurutkan secara berbeda atau menghasilkan entri indeks yang tidak valid. Jika terjadi masalah terkait tata urutan kolasi selama pembaruan, Anda mungkin perlu membuat ulang indeks.

Untuk memperkecil potensi dampak pembaruan glibc, RDS for PostgreSQL kini menyertakan pustaka kolasi default independen. Pustaka kolasi ini tersedia di RDS for PostgreSQL 14.6, 13.9, 12.13, 11.18, 10.23, dan rilis versi minor yang lebih baru. Pustaka ini kompatibel dengan glibc 2.26-59.amzn2, dan menyediakan stabilitas tata urutan untuk mencegah kesalahan hasil kueri.

Memahami peran dan izin PostgreSQL

Bila Anda membuat menggunakan, akun administrator dibuat pada saat yang sama. AWS Management Console Secara default, namanya adalah `postgres`, seperti yang ditunjukkan dalam tangkapan layar berikut:



Anda dapat memilih nama lain daripada harus menerima default (`postgres`). Jika ya, nama yang Anda pilih harus dimulai dengan huruf dan berada antara 1 dan 16 karakter alfanumerik. Untuk memudahkan, kami merujuk ke akun pengguna utama ini berdasarkan nilai default-nya (`postgres`) di seluruh panduan ini.

Jika Anda menggunakan `create-db-instance` AWS CLI bukan AWS Management Console, Anda membuat nama dengan meneruskannya dengan `master-username` parameter dalam perintah. Lihat informasi yang lebih lengkap di Membuat instans DB Amazon RDS.

Baik Anda menggunakan AWS Management Console, API AWS CLI, atau Amazon RDS, dan apakah Anda menggunakan `postgres` nama default atau memilih nama yang berbeda, akun pengguna database pertama ini adalah anggota `rds_superuser` grup dan memiliki `rds_superuser` hak istimewa.

Topik

- Memahami peran rds_superuser
- Mengontrol akses pengguna ke basis data PostgreSQL
- Mendelegasikan dan mengendalikan pengelolaan kata sandi pengguna
- Menggunakan SCRAM untuk enkripsi kata sandi PostgreSQL

Memahami peran rds_superuser

Di PostgreSQL, peran dapat menentukan pengguna, grup, atau sekumpulan izin khusus yang diberikan kepada grup atau pengguna untuk berbagai objek dalam basis data. Perintah PostgreSQL ke `CREATE USER` dan `CREATE GROUP` telah digantikan oleh perintah yang lebih umum, `CREATE`

ROLE dengan properti khusus untuk membedakan pengguna basis data. Pengguna basis data dapat dianggap sebagai peran dengan hak istimewa LOGIN.

> **Note**
>
> Perintah CREATE USER dan CREATE GROUP masih dapat digunakan. Untuk informasi selengkapnya, lihat Database Role dalam dokumentasi PostgreSQL.

Pengguna postgres adalah pengguna basis data yang paling istimewa di instans DB RDS for PostgreSQL. Pengguna tersebut memiliki karakteristik yang ditentukan oleh pernyataan CREATE ROLE berikut.

```
CREATE ROLE postgres WITH LOGIN NOSUPERUSER INHERIT CREATEDB CREATEROLE NOREPLICATION
 VALID UNTIL 'infinity'
```

Properti NOSUPERUSER, NOREPLICATION, INHERIT, dan VALID UNTIL 'infinity' merupakan opsi default untuk CREATE ROLE, kecuali ditentukan lain.

Secara default, postgres memiliki hak istimewa yang diberikan untuk peran rds_superuser, serta izin untuk membuat peran dan basis data. Peran rds_superuser memungkinkan pengguna postgres untuk melakukan berbagai hal berikut:

- Menambahkan ekstensi yang tersedia untuk digunakan dengan Amazon RDS. Untuk informasi selengkapnya, lihat Menggunakan fitur PostgreSQL yang didukung oleh Amazon RDS for PostgreSQL
- Membuat peran untuk pengguna dan memberikan hak istimewa kepada pengguna. Untuk informasi selengkapnya, lihat CREATE ROLE dan GRANT dalam dokumentasi PostgreSQL.
- Membuat basis data. Untuk informasi selengkapnya, lihat CREATE DATABASE dalam dokumentasi PostgreSQL.
- Memberikan hak istimewa rds_superuser untuk peran pengguna yang tidak memiliki hak istimewa ini, dan mencabut hak istimewa sesuai kebutuhan. Sebaiknya Anda memberikan peran ini hanya kepada pengguna yang melakukan tugas pengguna super. Dengan kata lain, Anda dapat memberikan peran ini kepada administrator basis data (DBA) atau administrator sistem.
- Memberikan (dan mencabut) peran rds_replication kepada pengguna basis data yang tidak memiliki peran rds_superuser.

- Memberikan (dan mencabut) peran `rds_password` kepada pengguna basis data yang tidak memiliki peran `rds_superuser`.
- Mendapatkan informasi status semua koneksi basis data menggunakan tampilan `pg_stat_activity`. Jika diperlukan, `rds_superuser` dapat menghentikan koneksi apa pun menggunakan `pg_terminate_backend` atau `pg_cancel_backend`.

Dalam pernyataan `CREATE ROLE postgres...`, Anda dapat melihat bahwa peran pengguna `postgres` secara khusus melarang izin `superuser` PostgreSQL. RDS for PostgreSQL adalah layanan terkelola sehingga Anda tidak dapat mengakses OS host, dan tidak dapat terhubung menggunakan akun `superuser` PostgreSQL. Banyak tugas yang memerlukan akses `superuser` pada PostgreSQL mandiri dikelola secara otomatis oleh Amazon RDS.

Untuk informasi pemberian hak istimewa selengkapnya, lihat GRANT dalam dokumentasi PostgreSQL.

Peran `rds_superuser` adalah salah satu dari beberapa peran yang telah ditentukan dalam Instans DB RDS for PostgreSQL.

> **Note**
>
> Dalam PostgreSQL 13 dan rilis sebelumnya, peran yang telah ditentukan dikenal sebagai peran default.

Dalam daftar berikut, Anda akan menemukan beberapa peran standar lainnya yang dibuat secara otomatis untuk baru. Instans DB RDS for PostgreSQL. Peran yang telah ditentukan dan hak istimewa mereka tidak dapat diubah. Anda tidak dapat menghapus, mengganti nama, atau memodifikasi hak istimewa untuk peran yang telah ditentukan ini. Mencoba untuk melakukannya akan menghasilkan kesalahan.

- rds_password – Peran yang dapat mengubah kata sandi dan mengatur batasan kata sandi untuk pengguna basis data. `rds_superuser`Peran diberikan dengan peran ini secara default, dan dapat memberikan peran tersebut kepada pengguna database. Untuk informasi selengkapnya, lihat Mengontrol akses pengguna ke basis data PostgreSQL.
  - Untuk RDS untuk PostgreSQL versi yang lebih tua dari 14`rds_password`, peran dapat mengubah kata sandi dan mengatur batasan kata sandi untuk pengguna database dan pengguna dengan peran. `rds_superuser` Dari RDS untuk PostgreSQL versi 14 dan yang

lebih barurds_password, peran dapat mengubah kata sandi dan mengatur batasan kata sandi hanya untuk pengguna database. Hanya pengguna dengan rds_superuser peran yang dapat melakukan tindakan ini pada pengguna lain yang memiliki rds_superuser peran.

- rdsadmin – Peran yang dibuat untuk menangani banyak tugas pengelolaan yang akan dilakukan oleh administrator dengan hak istimewa superuser di basis data PostgreSQL mandiri. Peran ini digunakan secara internal oleh RDS for PostgreSQL untuk banyak tugas pengelolaan.
- rdstopmgr – Peran yang digunakan secara internal oleh Amazon RDS untuk mendukung deployment multi-AZ.

Untuk melihat semua peran yang telah ditentukan sebelumnya, Anda dapat terhubung ke instans DB RDS for PostgreSQL dan menggunakan metacommand psql \du. Output akan terlihat sebagai berikut:

```
List of roles
  Role name   |      Attributes                   |      Member of
--------------+-----------------------------------+-----------------------------------
postgres      | Create role, Create DB           +| {rds_superuser}
              | Password valid until infinity     |
rds_superuser | Cannot login                      | {pg_monitor,pg_signal_backend,
              |                                  +|   rds_replication,rds_password}
...
```

Dalam output, Anda dapat melihat bahwa rds_superuser bukan merupakan peran pengguna basis data (tidak dapat masuk), tetapi memiliki hak istimewa dari banyak peran lainnya. Anda juga dapat melihat bahwa pengguna basis data postgres adalah anggota peran rds_superuser. Seperti yang disebutkan sebelumnya, postgres adalah nilai default di halaman Buat basis data konsol Amazon RDS. Jika Anda memilih nama lain, nama tersebut akan ditampilkan dalam daftar peran sebagai gantinya.

Mengontrol akses pengguna ke basis data PostgreSQL

Basis data baru di PostgreSQL selalu dibuat dengan serangkaian hak istimewa default dalam skema public basis data yang memungkinkan semua pengguna basis data dan peran untuk membuat objek. Dengan hak istimewa ini, pengguna basis data dapat terhubung ke basis data, misalnya, dan membuat tabel sementara saat terhubung.

Agar dapat lebih baik dalam mengontrol akses pengguna ke instans basis data yang Anda buat di instans DB RDS for PostgreSQL, sebaiknya Anda mencabut hak istimewa public default. Setelah

melakukannya, Anda kemudian dapat memberikan hak khusus untuk pengguna basis data dengan lebih terperinci, seperti yang diperlihatkan dalam prosedur berikut ini.

Untuk mengatur peran dan hak istimewa instans basis data baru

Misalkan Anda sedang menyiapkan basis data di instans DB RDS for PostgreSQL untuk digunakan oleh beberapa peneliti yang semuanya membutuhkan akses baca-tulis ke basis data.

1. Gunakan `psql` (or pgAdmin) untuk terhubun ke instans DB RDS for PostgreSQL:

   ```
   psql --host=your-db-instance.666666666666.aws-region.rds.amazonaws.com --port=5432
    --username=postgres --password
   ```

   Saat diminta, masukkan kata sandi Anda. Klien `psql` menghubungkan dan menampilkan basis data koneksi administratif default, `postgres=>`, sebagai prompt.

2. Untuk mencegah pengguna basis data membuat objek dalam skema `public`, lakukan hal berikut:

   ```
   postgres=> REVOKE CREATE ON SCHEMA public FROM PUBLIC;
   REVOKE
   ```

3. Selanjutnya, Anda akan membuat instans basis data baru:

   ```
   postgres=> CREATE DATABASE lab_db;
   CREATE DATABASE
   ```

4. Cabut semua hak istimewa dari skema `PUBLIC` pada basis data baru ini.

   ```
   postgres=> REVOKE ALL ON DATABASE lab_db FROM public;
   REVOKE
   ```

5. Buat peran untuk pengguna basis data.

   ```
   postgres=> CREATE ROLE lab_tech;
   CREATE ROLE
   ```

6. Beri kemampuan untuk terhubung ke basis data kepada pengguna basis data yang memiliki peran ini.

   ```
   postgres=> GRANT CONNECT ON DATABASE lab_db TO lab_tech;
   ```

```
GRANT
```

7. Beri semua hak istimewa pada basis data ini kepada semua pengguna dengan peran `lab_tech`.

```
postgres=> GRANT ALL PRIVILEGES ON DATABASE lab_db TO lab_tech;
GRANT
```

8. Buat pengguna basis data, sebagai berikut:

```
postgres=> CREATE ROLE lab_user1 LOGIN PASSWORD 'change_me';
CREATE ROLE
postgres=> CREATE ROLE lab_user2 LOGIN PASSWORD 'change_me';
CREATE ROLE
```

9. Beri hak istimewa yang terkait dengan peran lab_tech kepada dua pengguna ini:

```
postgres=> GRANT lab_tech TO lab_user1;
GRANT ROLE
postgres=> GRANT lab_tech TO lab_user2;
GRANT ROLE
```

Di titik ini, `lab_user1` dan `lab_user2` dapat terhubung ke basis data `lab_db`. Contoh ini tidak mengikuti praktik terbaik untuk penggunaan perusahaan yang mungkin termasuk membuat beberapa instans basis data, skema yang berbeda, dan pemberian izin terbatas. Untuk informasi lengkap dan skenario tambahan selengkapnya, lihat Mengelola Pengguna dan Peran PostgreSQL.

Untuk informasi hak istimewa di basis data PostgreSQL selengkapnya, lihat perintah GRANT dalam dokumentasi PostgreSQL.

Mendelegasikan dan mengendalikan pengelolaan kata sandi pengguna

Sebagai DBA, Anda mungkin ingin mendelegasikan pengelolaan kata sandi pengguna. Mungkin Anda juga ingin mencegah pengguna basis data mengubah kata sandi mereka atau mengonfigurasi ulang batasan kata sandi, seperti masa pakai kata sandi. Untuk memastikan bahwa hanya pengguna basis data terpilih yang dapat mengubah pengaturan kata sandi, Anda dapat mengaktifkan fitur pengelolaan kata sandi terbatas. Saat Anda mengaktifkan fitur ini, hanya pengguna basis data yang telah diberi peran `rds_password` yang dapat mengelola kata sandi.

> **Note**
>
> Untuk menggunakan manajemen kata sandi terbatas, instans DB RDS for PostgreSQL harus menjalankan PostgreSQL 10.6 atau yang lebih baru.

Secara default, fitur ini dalam keadaan `off`, seperti yang ditunjukkan berikut:

```
postgres=> SHOW rds.restrict_password_commands;
  rds.restrict_password_commands
--------------------------------
 off
(1 row)
```

Untuk mengaktifkan fitur ini, Anda harus menggunakan grup parameter khusus dan mengubah pengaturan `rds.restrict_password_commands` ke 1. Pastikan untuk melakukan booting ulang instans DB RDS for PostgreSQL agar pengaturan dapat berlaku.

Dalam keadaan fitur ini aktif, hak istimewa `rds_password` diperlukan untuk perintah SQL berikut:

```
CREATE ROLE myrole WITH PASSWORD 'mypassword';
CREATE ROLE myrole WITH PASSWORD 'mypassword' VALID UNTIL '2023-01-01';
ALTER ROLE myrole WITH PASSWORD 'mypassword' VALID UNTIL '2023-01-01';
ALTER ROLE myrole WITH PASSWORD 'mypassword';
ALTER ROLE myrole VALID UNTIL '2023-01-01';
ALTER ROLE myrole RENAME TO myrole2;
```

Mengganti nama peran (`ALTER ROLE myrole RENAME TO newname`) juga dibatasi jika kata sandi menggunakan algoritma hashing MD5.

Jika fitur ini dalam keadaan aktif, mencoba salah satu perintah SQL ini tanpa izin peran `rds_password` akan menghasilkan kesalahan berikut:

```
ERROR: must be a member of rds_password to alter passwords
```

Sebaiknya Anda memberikan `rds_password` hanya untuk beberapa peran yang Anda gunakan semata untuk pengelolaan kata sandi. Jika memberikan hak istimewa `rds_password` kepada pengguna basis data yang tidak memiliki hak istimewa `rds_superuser`, Anda juga harus memberi mereka atribut `CREATEROLE`.

Pastikan Anda memverifikasi persyaratan kata sandi seperti kedaluwarsa dan kompleksitas yang diperlukan di sisi klien. Jika Anda menggunakan utilitas sisi klien Anda sendiri untuk perubahan terkait kata sandi, utilitas harus menjadi anggota `rds_password` dan memiliki hak istimewa `CREATE ROLE`.

Menggunakan SCRAM untuk enkripsi kata sandi PostgreSQL

Salted Challenge Response Authentication Mechanism (SCRAM) adalah alternatif untuk algoritma intisari pesan (MD5) default PostgreSQL untuk mengenkripsi kata sandi. Mekanisme autentikasi SCRAM dianggap lebih aman daripada MD5. Untuk mempelajari dua pendekatan berbeda guna mengamankan kata sandi ini selengkapnya, lihat Password Authentication dalam dokumentasi PostgreSQL.

Sebaiknya menggunakan SCRAM daripada MD5 sebagai skema enkripsi kata sandi untuk Instans DB RDS for PostgreSQL. Ini adalah mekanisme respons tantangan kriptografi yang menggunakan algoritma scram-sha-256 untuk autentikasi dan enkripsi kata sandi.

Anda mungkin perlu memperbarui pustaka untuk aplikasi klien Anda agar dapat mendukung SCRAM. Misalnya, versi JDBC sebelum 42.2.0 tidak mendukung SCRAM. Untuk informasi selengkapnya, lihat PostgreSQL JDBC Driver dalam dokumentasi Driver JDBC PostgreSQL. Untuk daftar driver PostgreSQL lainnya dan dukungan SCRAM, lihat List of drivers dalam dokumentasi PostgreSQL.

> **Note**
>
> RDS for PostgreSQL versi 13.1 dan dukungan scram-sha-256 yang lebih baru. Versi ini juga memungkinkan Anda mengonfigurasi instans DB agar meminta SCRAM, seperti yang dibahas dalam prosedur berikut.

Menyiapkan instans DB RDS for PostgreSQL agar meminta SCRAM

Anda dapat meminta instans DB RDS for PostgreSQL untuk hanya menerima kata sandi yang menggunakan algoritma scram-sha-256.

> **Important**
>
> Untuk Proksi RDS yang ada dengan basis data PostgreSQL, jika Anda mengubah autentikasi basis data untuk menggunakan `SCRAM` saja, proksi akan menjadi tidak tersedia selama maksimal 60 detik. Untuk menghindari masalah ini, lakukan salah satu hal berikut:

- Pastikan basis data memungkinkan autentikasi `SCRAM` dan `MD5`.
- Untuk hanya menggunakan autentikasi `SCRAM`, buat proksi baru, migrasi lalu lintas aplikasi Anda ke proksi baru, lalu hapus proksi yang sebelumnya terkait dengan basis data.

Sebelum melakukan perubahan pada sistem, pastikan Anda telah memahami proses lengkapnya sebagai berikut:

- Dapatkan informasi semua peran dan enkripsi kata sandi untuk semua pengguna basis data.
- Periksa kembali pengaturan parameter instans DB RDS for PostgreSQL untuk parameter yang mengontrol enkripsi kata sandi.
- Jika instans DB RDS for PostgreSQL menggunakan grup parameter default, Anda harus membuat grup parameter DB khusus lalu mengaplikasikannya ke instans DB RDS for PostgreSQL agar Anda dapat memodifikasi parameter saat saat diperlukan. Jika instans DB RDS for PostgreSQL menggunakan grup parameter khusus, Anda dapat memodifikasi parameter yang diperlukan nanti sesuai kebutuhan saat proses berlangsung.
- Ubah parameter `password_encryption` ke `scram-sha-256`.
- Beri tahu semua pengguna basis data bahwa mereka perlu memperbarui kata sandi. Lakukan hal yang sama untuk akun `postgres` Anda. Kata sandi baru dienkripsi dan disimpan menggunakan algoritma scram-sha-256.
- Verifikasi bahwa semua kata sandi dienkripsi menggunakan jenis enkripsi.
- Jika semua kata sandi menggunakan scram-sha-256, Anda dapat mengubah parameter `rds.accepted_password_auth_method` dari `md5+scram` ke `scram-sha-256`.

> **Warning**
>
> Setelah Anda mengubah `rds.accepted_password_auth_method` ke scram-sha-256, setiap pengguna (peran) dengan kata sandi terenkripsi `md5` tidak dapat terhubung.

Penyiapan meminta SCRAM untuk instans DB RDS for PostgreSQL

Sebelum membuat perubahan apa pun pada instans DB RDS for PostgreSQL, periksa semua akun pengguna basis data yang ada. Periksa juga jenis enkripsi yang digunakan untuk kata sandi. Anda

dapat melakukan tugas-tugas ini menggunakan ekstensi `rds_tools`. Ekstensi ini didukung di RDS for PostgreSQL 13.1 dan rilis yang lebih baru.

Untuk mendapatkan daftar pengguna basis data (peran) dan metode enkripsi kata sandi

1. Gunakan `psql` untuk terhubung ke instans DB RDS for PostgreSQL Anda, seperti yang ditunjukkan di bawah ini.

```
psql --host=db-name.111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password
```

2. Instal ekstensi `rds_tools`.

```
postgres=> CREATE EXTENSION rds_tools;
CREATE EXTENSION
```

3. Dapatkan daftar peran dan enkripsi.

```
postgres=> SELECT * FROM
      rds_tools.role_password_encryption_type();
```

   Anda akan melihat output yang mirip dengan berikut ini.

```
       rolname        | encryption_type
----------------------+-----------------
 pg_monitor           |
 pg_read_all_settings |
 pg_read_all_stats    |
 pg_stat_scan_tables  |
 pg_signal_backend    |
 lab_tester           | md5
 user_465             | md5
 postgres             | md5
(8 rows)
```

Membuat grup parameter DB khusus

> **Note**
>
> Jika instans DB RDS for PostgreSQL sudah menggunakan grup parameter khusus, Anda tidak perlu membuat grup parameter yang baru.

Untuk ringkasan grup parameter Amazon RDS, lihat Bekerja dengan parameter pada instans DB RDS for PostgreSQL.

Jenis enkripsi kata sandi yang digunakan untuk kata sandi diatur dalam satu parameter, `password_encryption`. Enkripsi yang diizinkan instans DB RDS for PostgreSQL diatur dalam parameter lain, `rds.accepted_password_auth_method`. Mengubah salah satu dari nilai default ini mengharuskan Anda membuat grup parameter DB khusus dan mengaplikasikannya ke instans.

Anda juga dapat menggunakan AWS Management Console atau RDS API untuk membuat DB. Untuk informasi selengkapnya, lihat

Anda kini dapat mengaitkan grup parameter khusus dengan instans DB.

Untuk membuat grup parameter DB khusus

1. Gunakan perintah CLI `create-db-parameter-group` untuk membuat grup parameter DB khusus. Contoh ini menggunakan `postgres13` sebagai sumber untuk grup parameter khusus ini.

   Untuk Linux, macOS, atau Unix:

   ```
   aws rds create-db-parameter-group --db-parameter-group-name 'docs-lab-scram-passwords' \
     --db-parameter-group-family postgres13  --description 'Custom parameter group for SCRAM'
   ```

   Untuk Windows:

   ```
   aws rds create-db-parameter-group --db-parameter-group-name "docs-lab-scram-passwords" ^
     --db-parameter-group-family postgres13  --description "Custom DB parameter group for SCRAM"
   ```

2. Gunakan perintah CLI `modify-db-instance` untuk menerapkan grup parameter khusus ini ke klaster DB RDS for PostgreSQL.

   Untuk Linux, macOS, atau Unix:

   ```
   aws rds modify-db-instance --db-instance-identifier 'your-instance-name' \
           --db-parameter-group-name "docs-lab-scram-passwords
   ```

   Untuk Windows:

   ```
   aws rds modify-db-instance --db-instance-identifier "your-instance-name" ^
           --db-parameter-group-name "docs-lab-scram-passwords
   ```

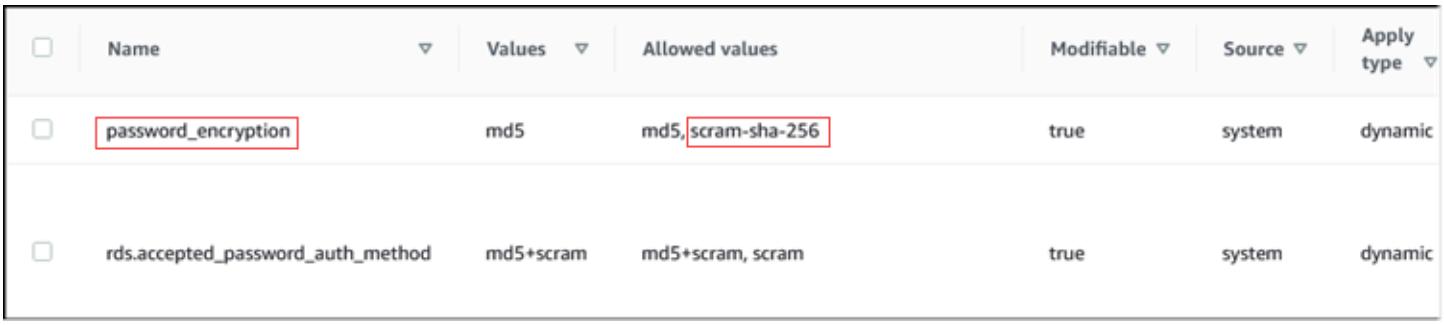
   Untuk menyinkronkan ulang instans DB RDS for PostgreSQL dengan grup parameter DB khusus, Anda perlu melakukan boot ulang instans utama dan semua instans lain klaster. Untuk meminimalkan dampak pada pengguna Anda, jadwalkan proses ini agar berlangsung selama masa pemeliharaan rutin.

Mengonfigurasi enkripsi kata sandi agar menggunakan SCRAM

Mekanisme enkripsi kata sandi yang digunakan oleh instans DB RDS for PostgreSQL diatur dalam grup parameter DB di parameter `password_encryption`. Nilai yang diizinkan tidak disetel, `md5`, atau `scram-sha-256`. Nilai default bergantung pada versi RDS for PostgreSQL, sebagai berikut:

- RDS for PostgreSQL 14 dan versi di atasnya – Default adalah `scram-sha-256`
- RDS for PostgreSQL 13 – Default adalah `md5`

Dengan grup parameter DB khusus yang dilampirkan ke instans DB RDS for PostgreSQL, Anda dapat memodifikasi nilai untuk parameter enkripsi kata sandi.



Name	Values	Allowed values	Modifiable	Source	Apply type
password_encryption	md5	md5, scram-sha-256	true	system	dynamic
rds.accepted_password_auth_method	md5+scram	md5+scram, scram	true	system	dynamic

Untuk mengubah pengaturan enkripsi kata sandi menjadi scram-sha-256

- Ubah nilai enkripsi kata sandi menjadi scram-sha-256, seperti yang ditunjukkan berikut. Perubahan dapat diterapkan segera karena parameter bersifat dinamis sehingga tidak perlu memulai ulang agar perubahan diterapkan.

  Untuk Linux, macOS, atau Unix:

  ```
  aws rds modify-db-parameter-group --db-parameter-group-name \
    'docs-lab-scram-passwords' --parameters
   'ParameterName=password_encryption,ParameterValue=scram-
  sha-256,ApplyMethod=immediate'
  ```

  Untuk Windows:

  ```
  aws rds modify-db-parameter-group --db-parameter-group-name ^
    "docs-lab-scram-passwords" --parameters
   "ParameterName=password_encryption,ParameterValue=scram-
  sha-256,ApplyMethod=immediate"
  ```

Memigrasikan kata sandi untuk peran pengguna ke SCRAM

Anda dapat memigrasikan kata sandi untuk peran pengguna ke SCRAM seperti yang dijelaskan berikut.

Untuk memigrasikan kata sandi pengguna (peran) basis data dari MD5 ke SCRAM

1. Masuk sebagai pengguna administrator (nama pengguna default, `postgres`) seperti yang ditunjukkan berikut.

   ```
   psql --host=db-name.111122223333.aws-region.rds.amazonaws.com --port=5432 --
   username=postgres --password
   ```

2. Periksa pengaturan parameter `password_encryption` pada instans DB RDS for PostgreSQL menggunakan perintah berikut.

   ```
   postgres=> SHOW password_encryption;
    password_encryption
   ---------------------
    md5
   ```

```
 (1 row)
```

3. Ubah nilai parameter ini menjadi scram-sha-256. Ini adalah parameter dinamis sehingga Anda tidak perlu melakukan boot ulang instans setelah melakukan perubahan ini. Periksa kembali nilainya untuk memastikan parameter sekarang telah diatur ke `scram-sha-256`, sebagai berikut.

```
postgres=> SHOW password_encryption;
 password_encryption
---------------------
 scram-sha-256
 (1 row)
```

4. Beri tahu semua pengguna basis data untuk mengubah kata sandi mereka. Pastikan juga untuk mengubah kata sandi Anda sendiri untuk akun `postgres` (pengguna basis data dengan hak istimewa `rds_superuser`).

```
labdb=> ALTER ROLE postgres WITH LOGIN PASSWORD 'change_me';
ALTER ROLE
```

5. Ulangi proses untuk semua basis data di Anda. Instans DB RDS for PostgreSQL.

Mengubah parameter agar memerlukan SCRAM

Ini adalah langkah terakhir dalam prosesnya. Setelah Anda membuat perubahan dalam prosedur berikut, setiap akun pengguna (peran) yang masih menggunakan enkripsi `md5` untuk kata sandi tidak dapat masuk ke Instans DB RDS for PostgreSQL.

`rds.accepted_password_auth_method` menentukan metode enkripsi yang diterima instans DB RDS for PostgreSQL untuk kata sandi pengguna selama proses login. Nilai default-nya adalah `md5+scram`, yang berarti bahwa salah satu metode diterima. Pada gambar berikut, Anda dapat menemukan pengaturan default untuk parameter ini.

Name	Values	Allowed values	Modifiable	Source	Apply type
password_encryption	scram-sha-256	md5, scram-sha-256	true	system	dynamic
rds.accepted_password_auth_method	md5+scram	md5+scram, scram	true	system	dynamic

Nilai yang diizinkan untuk parameter ini adalah `md5+scram` atau `scram`. Mengubah nilai parameter ini ke `scram` akan menjadikannya sebagai persyaratan.

Untuk mengubah nilai parameter agar memerlukan autentikasi SCRAM untuk kata sandi

1. Verifikasi bahwa semua kata sandi pengguna basis data untuk semua basis data di instans DB RDS for PostgreSQL menggunakan `scram-sha-256` untuk enkripsi kata sandi. Untuk melakukannya, kueri `rds_tools` untuk peran (pengguna) dan jenis enkripsi, sebagai berikut.

```
postgres=> SELECT * FROM rds_tools.role_password_encryption_type();
  rolname        | encryption_type
  ---------------------+-----------------
  pg_monitor           |
  pg_read_all_settings |
  pg_read_all_stats    |
  pg_stat_scan_tables  |
  pg_signal_backend    |
  lab_tester           | scram-sha-256
  user_465             | scram-sha-256
  postgres             | scram-sha-256
  ( rows)
```

2. Ulangi kueri di semua instans DB di . Instans DB RDS for PostgreSQL.

   Jika semua kata sandi menggunakan scram-sha-256, Anda dapat melanjutkan.

3. Ubah nilai autentikasi kata sandi yang diterima menjadi scram-sha-256, sebagai berikut.

   Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-parameter-group --db-parameter-group-name 'docs-lab-scram-
passwords' \
  --parameters
 'ParameterName=rds.accepted_password_auth_method,ParameterValue=scram,ApplyMethod=immediat
```

   Untuk Windows:

```
aws rds modify-db-parameter-group --db-parameter-group-name "docs-lab-scram-
passwords" ^
  --parameters
 "ParameterName=rds.accepted_password_auth_method,ParameterValue=scram,ApplyMethod=immediat
```

Bekerja dengan fitur autovacuum PostgreSQL di Amazon RDS for PostgreSQL

Kami sangat menyarankan Anda menggunakan fitur autovacuum untuk menjaga kondisi instans DB PostgreSQL Anda. Autovacuum akan mengotomatiskan awal perintah VACUUM dan ANALYZE. Kemudian memeriksa tabel yang memuat banyak tupel yang dimasukkan, diperbarui, atau dihapus. Setelah pemeriksaan ini, autovacuum akan mengambil kembali penyimpanan dengan menghapus data usang atau tupel dari basis data PostgreSQL.

Secara default, autovacuum diaktifkan untuk instans DB Amazon RDS for PostgreSQL yang Anda buat menggunakan salah satu grup parameter DB PostgreSQL default. Grup parameter ini termasuk `default.postgres10`,`default.postgres11`, dan sebagainya. Semua grup parameter DB PostgreSQL default memiliki parameter `rds.adaptive_autovacuum` yang disetel ke `1` sehingga mengaktifkan fitur tersebut. Parameter konfigurasi lain yang terkait dengan fitur autovacuum juga diatur secara default. Karena parameter default ini cukup umum, Anda dapat memanfaatkannya dengan menyetel beberapa parameter yang terkait dengan fitur autovacuum untuk beban kerja spesifik Anda.

Setelah itu, Anda dapat menemukan informasi lebih lanjut mengenai autovacuum dan cara menyetel beberapa parameternya di RDS Anda untuk instans DB PostgreSQL. Untuk informasi tingkat tinggi, lihat Praktik terbaik untuk menggunakan PostgreSQL.

Topik

- Mengalokasikan memori untuk autovacuum
- Mengurangi kemungkinan penyelesaian ID transaksi
- Menentukan apakah tabel di basis data Anda perlu divakum atau tidak
- Menentukan tabel mana yang saat ini memenuhi syarat untuk autovacuum
- Menentukan apakah autovacuum saat ini sedang berjalan atau tidak, dan berapa lama durasinya
- Melakukan pembekuan vakum manual
- Mengindeks ulang tabel saat autovacuum berjalan
- Mengelola autovacuum dengan indeks berukuran besar
- Parameter lain yang memengaruhi autovacuum
- Mengatur parameter autovacuum tingkat tabel
- Melakukan log aktivitas autovacuum dan vakum

Mengalokasikan memori untuk autovacuum

Salah satu parameter paling penting yang memengaruhi performa autovacuum adalah maintenance_work_mem. Parameter ini menentukan berapa banyak memori yang Anda alokasikan agar dapat digunakan oleh untuk memindai tabel basis data dan menyimpan semua ID baris yang akan divakum. Jika Anda mengatur nilai parameter `maintenance_work_mem` terlalu rendah, proses vakum mungkin harus memindai tabel beberapa kali untuk menyelesaikan pekerjaannya. Pemindaian berulang seperti ini dapat berdampak negatif pada performa.

Saat melakukan perhitungan untuk menentukan nilai parameter `maintenance_work_mem`, ingatlah dua hal berikut:

- Unit default untuk parameter ini adalah kilobita (KB).
- Parameter `maintenance_work_mem` bekerja secara bersama-sama dengan parameter `autovacuum_max_workers`. Jika Anda memiliki banyak tabel kecil, alokasikan lebih banyak parameter `autovacuum_max_workers` dan lebih sedikit parameter `maintenance_work_mem`. Jika Anda memiliki tabel besar (misalnya lebih besar dari 100 GB), alokasikan lebih banyak memori dan lebih sedikit proses pekerja. Anda harus memiliki cukup memori yang dialokasikan agar berhasil di tabel terbesar Anda. Setiap parameter `autovacuum_max_workers` dapat menggunakan memori yang Anda alokasikan. Dengan demikian, pastikan bahwa kombinasi proses pekerja dan memori sama dengan memori total yang ingin Anda alokasikan.

Secara umum, untuk host besar tetapkan parameter `maintenance_work_mem` ke nilai antara satu dan dua gigabita (antara 1.048.576 dan 2.097.152 KB). Untuk host yang sangat besar, atur parameter ke nilai antara dua dan empat gigabita (antara 2.097.152 dan 4.194.304 KB). Nilai yang Anda tetapkan untuk parameter ini bergantung pada beban kerja. Amazon RDS telah memperbarui unit default untuk parameter ini menjadi kilobita yang dihitung sebagai berikut.

`GREATEST({DBInstanceClassMemory/63963136*1024},65536)`.

Mengurangi kemungkinan penyelesaian ID transaksi

Dalam beberapa kasus, pengaturan kelompok parameter yang terkait dengan autovacuum mungkin tidak cukup agresif untuk mencegah penyelesaian ID transaksi. Untuk mengatasi hal ini, RDS for PostgreSQL menyediakan mekanisme yang dapat menyesuaikan nilai parameter autovacuum secara otomatis. Penyesuaian parameter autovacuum adaptif adalah fitur untuk RDS for PostgreSQL. Penjelasan terperinci tentang Penyelesaian ID Transaksi dapat ditemukan dalam dokumentasi PostgreSQL.

Penyesuaian parameter autovacuum adaptif diaktifkan secara default untuk instans RDS for PostgreSQL dengan mengaktifkan parameter dinamis `rds.adaptive_autovacuum`. Kami sangat menyarankan Anda untuk tetap mengaktifkan parameter dinamis ini. Namun, untuk menonaktifkan penyesuaian parameter autovacuum adaptif, atur parameter `rds.adaptive_autovacuum` ke 0 atau nonaktifkan.

Penyelesaian ID Transaksi masih memungkinkan meskipun Amazon RDS telah menyesuaikan parameter autovacuum. Kami mendorong Anda untuk menerapkan CloudWatch alarm Amazon untuk sampul ID transaksi. Untuk informasi selengkapnya, lihat postingan Menerapkan sistem peringatan awal untuk penyelesaian ID transaksi di RDS for PostgreSQL di AWS Database Blog.

Dengan penyetelan parameter autovacuum adaptif diaktifkan, Amazon RDS mulai menyesuaikan parameter autovacuum ketika CloudWatch metrik `MaximumUsedTransactionIDs` mencapai nilai parameter atau 500.000.000, mana yang lebih besar. `autovacuum_freeze_max_age`

Amazon RDS terus menyesuaikan parameter untuk autovacuum jika tabel terus menuju ke penyelesaian ID transaksi. Setiap penyesuaian ini secara khusus mengalokasikan lebih banyak sumber daya ke autovacuum untuk menghindari penyelesaian. Amazon RDS memperbarui parameter terkait autovacuum berikut ini:

- autovacuum_vacuum_cost_delay
- autovacuum_vacuum_cost_limit
- autovacuum_work_mem
- autovacuum_naptime

RDS akan memodifikasi parameter ini hanya jika nilai yang baru membuat autovacuum lebih agresif. Parameter dimodifikasi dalam memori di instans DB. Nilai di grup parameter tidak diubah. Untuk melihat pengaturan dalam memori saat ini, gunakan perintah PostgreSQL SHOW SQL.

Jika Amazon RDS memodifikasi salah satu parameter autovacuum ini, Amazon RDS akan menghasilkan peristiwa untuk instans DB yang terdampak. Peristiwa ini dapat dilihat di AWS Management Console dan melalui API Amazon RDS. Setelah `MaximumUsedTransactionIDs` CloudWatch metrik kembali di bawah ambang batas, Amazon RDS me-reset parameter terkait autovacuum dalam memori kembali ke nilai yang ditentukan dalam grup parameter. Kemudian, Amazon RDS akan menghasilkan peristiwa lain yang sesuai dengan perubahan ini.

Menentukan apakah tabel di basis data Anda perlu divakum atau tidak

Anda dapat menggunakan kueri berikut untuk menunjukkan jumlah transaksi yang tidak divakum di dalam basis data. Kolom `datfrozenxid` pada baris `pg_database` basis data merupakan batas bawah di ID transaksi normal yang muncul di basis data tersebut. Kolom ini adalah minimum dari nilai per tabel `relfrozenxid` di dalam basis data.

```
SELECT datname, age(datfrozenxid) FROM pg_database ORDER BY age(datfrozenxid) desc
 limit 20;
```

Sebagai contoh, hasil dari menjalankan kueri sebelumnya adalah sebagai berikut.

```
datname    | age
mydb       | 1771757888
template0  | 1721757888
template1  | 1721757888
rdsadmin   | 1694008527
postgres   | 1693881061
(5 rows)
```

Saat usia basis data mencapai 2 miliar transaksi ID, penyelesaian ID transaksi (XID) akan dilakukan dan akses ke basis data menjadi hanya baca. Anda dapat menggunakan kueri ini untuk menghasilkan metrik dan menjalankannya beberapa kali dalam sehari. Secara default, autovacuum diatur untuk menjaga usia transaksi tidak lebih dari 200.000.000 (`autovacuum_freeze_max_age`).

Contoh strategi pemantauan mungkin terlihat seperti ini:

- Tetapkan nilai `autovacuum_freeze_max_age` hingga 200 juta transaksi.
- Jika tabel mencapai 500 juta transaksi yang tidak divakum, ini akan memicu alarm tingkat rendah. Tidak ada yang dengan nilai ini, tetapi dapat mengindikasikan bahwa autovacuum tidak dapat diteruskan.
- Jika tabel berumur 1 miliar, indikasi ini harus diperlakukan sebagai peringatan untuk diambil tindakan. Umumnya, Anda ingin mempertahankan usia tabel mendekati `autovacuum_freeze_max_age` karena alasan performa. Sebaiknya Anda menyelidiki hal ini menggunakan rekomendasi berikut.
- Jika tabel mencapai 1,5 miliar transaksi yang tidak divakum, ini akan memicu alarm dengan keparahan tingkat tinggi. Bergantung pada seberapa cepat basis data Anda menggunakan transaksi ID, alarm ini dapat mengindikasikan bahwa sistem kehabisan waktu untuk menjalankan autovacuum. Dalam hal ini, sebaiknya Anda segera menyelesaikan proses ini.

Jika tabel terus-menerus melanggar ambang batas ini, ubah parameter autovacuum Anda. Secara default, menggunakan VACUUM secara manual (yang menonaktifkan penundaan berbasis biaya) bersifat lebih agresif dibandingkan menggunakan autovacuum default, tetapi juga lebih mengganggu sistem secara keseluruhan.

Sebaiknya lakukan hal berikut:

- Waspada dan aktifkan mekanisme pemantauan agar Anda dapat mengetahui usia transaksi Anda yang paling lama.

  Untuk informasi mengenai pembuatan proses yang memperingatkan Anda tentang penyelesaian ID transaksi, lihat postingan AWS Database Blog yang berjudul Menerapkan sistem peringatan awal untuk penyelesaian ID transaksi di Amazon RDS for PostgreSQL.
- Untuk tabel yang lebih sibuk, lakukan pembekuan vakum manual secara teratur selama pemeliharaan, selain mengandalkan autovacuum. Untuk informasi cara melakukan pembekuan vakum manual, lihat Melakukan pembekuan vakum manual.

Menentukan tabel mana yang saat ini memenuhi syarat untuk autovacuum

Sering kali, ada satu atau dua tabel yang harus divakum. Tabel dengan nilai `relfrozenxid` lebih besar dari jumlah transaksi dalam `autovacuum_freeze_max_age` selalu menjadi target autovacuum. Sebaliknya, jika jumlah tupel yang dibuat usang sejak VACUUM terakhir melebihi “ambang batas vakum”, tabel akan divakum.

Ambang batas autovacuum didefenisikan sebagai:

```
Vacuum-threshold = vacuum-base-threshold + vacuum-scale-factor * number-of-tuples
```

di mana `vacuum base threshold` adaa`utovacuum_vacuum_threshold`, `vacuum scale factor` adalaha`utovacuum_vacuum_scale_factor`, dan `number of tuples` adalah`pg_class.reltuples`.

Saat Anda terkoneksi ke basis data, jalankan kueri berikut untuk melihat daftar tabel yang dianggap oleh autovacuum telah memenuhi syarat untuk divakum.

```
WITH vbt AS (SELECT setting AS autovacuum_vacuum_threshold FROM
pg_settings WHERE name = 'autovacuum_vacuum_threshold'),
vsf AS (SELECT setting AS autovacuum_vacuum_scale_factor FROM
pg_settings WHERE name = 'autovacuum_vacuum_scale_factor'),
```

```
fma AS (SELECT setting AS autovacuum_freeze_max_age FROM pg_settings WHERE name =
 'autovacuum_freeze_max_age'),
sto AS (select opt_oid, split_part(setting, '=', 1) as param,
split_part(setting, '=', 2) as value from (select oid opt_oid, unnest(reloptions)
 setting from pg_class) opt)
SELECT '"'||ns.nspname||'"."'||c.relname||'"' as relation,
pg_size_pretty(pg_table_size(c.oid)) as table_size,
age(relfrozenxid) as xid_age,
coalesce(cfma.value::float, autovacuum_freeze_max_age::float)
 autovacuum_freeze_max_age,
(coalesce(cvbt.value::float, autovacuum_vacuum_threshold::float) +
coalesce(cvsf.value::float,autovacuum_vacuum_scale_factor::float) * c.reltuples)
AS autovacuum_vacuum_tuples, n_dead_tup as dead_tuples FROM
pg_class c join pg_namespace ns on ns.oid = c.relnamespace
join pg_stat_all_tables stat on stat.relid = c.oid join vbt on (1=1) join vsf on (1=1)
 join fma on (1=1)
left join sto cvbt on cvbt.param = 'autovacuum_vacuum_threshold' and c.oid =
 cvbt.opt_oid
left join sto cvsf on cvsf.param = 'autovacuum_vacuum_scale_factor' and c.oid =
 cvsf.opt_oid
left join sto cfma on cfma.param = 'autovacuum_freeze_max_age' and c.oid = cfma.opt_oid
WHERE c.relkind = 'r' and nspname <> 'pg_catalog'
AND (age(relfrozenxid) >= coalesce(cfma.value::float, autovacuum_freeze_max_age::float)
OR coalesce(cvbt.value::float, autovacuum_vacuum_threshold::float) +
coalesce(cvsf.value::float,autovacuum_vacuum_scale_factor::float) *
c.reltuples <= n_dead_tup)
ORDER BY age(relfrozenxid) DESC LIMIT 50;
```

Menentukan apakah autovacuum saat ini sedang berjalan atau tidak, dan berapa lama durasinya

Jika harus memvakum tabel secara manual, pastikan Anda bisa menentukan apakah autovacuum saat ini sedang berjalan atau tidak. Jika sudah berjalan, Anda mungkin perlu menyesuaikan parameter agar berjalan lebih efisien, atau mematikan autovacuum sementara sehingga Anda dapat menjalankan VACUUM secara manual.

Gunakan kueri berikut untuk menentukan apakah autovacuum berjalan, berapa lama sudah berjalan, dan apakah sedang menunggu sesi lainnya atau tidak.

```
SELECT datname, usename, pid, state, wait_event, current_timestamp - xact_start AS
 xact_runtime, query
FROM pg_stat_activity
```

```
WHERE upper(query) LIKE '%VACUUM%'
ORDER BY xact_start;
```

Setelah menjalankan kueri, Anda akan melihat hasil yang serupa dengan yang berikut ini.

```
 datname | usename  |  pid  | state  | wait_event |      xact_runtime       | query
 --------+----------+-------+--------+------------+-------------------------
+-------------------------------------------------------------------------------------------------
 mydb    | rdsadmin | 16473 | active |            | 33 days 16:32:11.600656 |
 autovacuum: VACUUM ANALYZE public.mytable1 (to prevent wraparound)
 mydb    | rdsadmin | 22553 | active |            | 14 days 09:15:34.073141 |
 autovacuum: VACUUM ANALYZE public.mytable2 (to prevent wraparound)
 mydb    | rdsadmin | 41909 | active |            | 3 days 02:43:54.203349  |
 autovacuum: VACUUM ANALYZE public.mytable3
 mydb    | rdsadmin |   618 | active |            | 00:00:00                |
 SELECT datname, usename, pid, state, wait_event, current_timestamp - xact_start AS
 xact_runtime, query+
         |          |       |        |            |                         | FROM
 pg_stat_activity
           +
         |          |       |        |            |                         | WHERE
 query like '%VACUUM%'
          +
         |          |       |        |            |                         | ORDER BY
 xact_start;
       +
```

Beberapa masalah dapat menyebabkan sesi autovacuum yang lama (yaitu beberapa hari). Masalah yang paling umum adalah nilai parameter [maintenance_work_mem](#) Anda diatur terlalu rendah untuk ukuran tabel atau laju pembaruan.

Sebaiknya Anda menggunakan rumus berikut untuk menetapkan nilai parameter `maintenance_work_mem`.

```
GREATEST({DBInstanceClassMemory/63963136*1024},65536)
```

Sesi autovacuum yang singkat juga dapat mengindikasikan adanya masalah:

- Sesi ini dapat mengindikasikan bahwa `autovacuum_max_workers` tidak cukup untuk beban kerja Anda. Dalam hal ini, Anda perlu menetapkan jumlah pekerja.

- Itu dapat mengindikasikan adanya kerusakan indeks (error pada autovacuum dan mulai ulang di hal yang sama, tetapi tidak ada kemajuan). Dalam hal ini, jalankan `vacuum freeze verbose table` manual untuk mengetahui penyebab pastinya.

Melakukan pembekuan vakum manual

Anda mungkin dapat melakukan vakum manual di tabel yang memiliki proses vakum yang sudah berjalan. Cara ini berguna jika Anda telah mengidentifikasi tabel dengan usia yang mendekati 2 miliar transaksi (atau di atas ambang batas yang Anda pantau).

Langkah-langkah berikut ini merupakan panduan, yang disertai dengan beberapa variasi proses. Misalnya, selama pengujian, anggaplah Anda menemukan bahwa nilai parameter `maintenance_work_mem` diatur terlalu kecil dan Anda harus segera mengambil tindakan di tabel. Namun, mungkin pada saat ini Anda tidak ingin mengembalikan instans. Dengan menggunakan kueri di bagian sebelumnya, Anda dapat menentukan tabel mana yang menjadi masalah dan melihat sesi autovacuum yang berjalan lama. Anda tahu bahwa Anda perlu mengubah pegaturan parameter `maintenance_work_mem`, tetapi Anda juga harus mengambil tindakan cepat dan memvakum tabel yang dimaksud. Prosedur berikut menunjukkan tindakan apa yang harus dilakukan dalam situasi ini.

Untuk melakukan pembekuan vakum secara manual

1. Buka dua sesi ke basis data yang berisi tabel yang ingin divakum. Untuk sesi kedua, gunakan "layar" atau peralatan lain yang dapat mempertahankan sesi jika koneksi terputus.
2. Dalam sesi pertama, dapatkan ID proses (PID) dari sesi autovacuum yang berjalan di tabel.

   Jalankan kueri berikut untuk mendapatkan PID dari sesi autovacuum.

   ```
   SELECT datname, usename, pid, current_timestamp - xact_start
   AS xact_runtime, query
   FROM pg_stat_activity WHERE upper(query) LIKE '%VACUUM%' ORDER BY
   xact_start;
   ```

3. Dalam sesi kedua, hitung jumlah memori yang Anda butuhkan untuk menjalankan operasi ini. Dalam contoh ini, kita akan menentukan bahwa kita mampu menggunakan memori hingga 2 GB untuk operasi ini sehingga kita menetapkan `maintenance_work_mem` untuk sesi saat ini hingga 2 GB.

   ```
   SET maintenance_work_mem='2 GB';
   SET
   ```

4. Dalam sesi kedua, munculkan perintah `vacuum freeze verbose` untuk tabel. Pengaturan verbose berguna karena Anda tetap dapat melihat aktivitas pembekuan tersebut meskipun tidak ada laporan kemajuannya di PostgreSQL saat ini.

```
\timing on
Timing is on.
vacuum freeze verbose pgbench_branches;
```

```
INFO:  vacuuming "public.pgbench_branches"
INFO:  index "pgbench_branches_pkey" now contains 50 row versions in 2 pages
DETAIL:  0 index row versions were removed.
0 index pages have been deleted, 0 are currently reusable.
CPU 0.00s/0.00u sec elapsed 0.00 sec.
INFO:  index "pgbench_branches_test_index" now contains 50 row versions in 2 pages
DETAIL:  0 index row versions were removed.
0 index pages have been deleted, 0 are currently reusable.
CPU 0.00s/0.00u sec elapsed 0.00 sec.
INFO:  "pgbench_branches": found 0 removable, 50 nonremovable row versions
     in 43 out of 43 pages
DETAIL:  0 dead row versions cannot be removed yet.
There were 9347 unused item pointers.
0 pages are entirely empty.
CPU 0.00s/0.00u sec elapsed 0.00 sec.
VACUUM
Time: 2.765 ms
```

5. Dalam sesi pertama, jika autovacuum memblokir sesi vakum, Anda akan melihat di `pg_stat_activity` bahwa proses menunggu untuk sesi vakum diberi tanda "T". Dalam hal ini, Anda harus mengakhiri proses autovacuum sesuai cara berikut.

```
SELECT pg_terminate_backend('the_pid');
```

   Di titik ini, sesi Anda dimulai. Perlu diperhatikan bahwa proses autovacuum akan segera dimulai ulang karena tabel ini mungkin merupakan urutan tertinggi di daftar pekerjaan.

6. Mulai perintah `vacuum freeze verbose` di sesi kedua, lalu akhiri proses autovacuum di sesi pertama.

Mengindeks ulang tabel saat autovacuum berjalan

Jika indeks rusak, autovacuum tetap akan terus memproses tabel lalu gagal. Jika mencoba vakum manual dalam situasi ini, Anda akan menerima pesan error seperti berikut ini.

```
postgres=>  vacuum freeze pgbench_branches;
ERROR: index "pgbench_branches_test_index" contains unexpected
   zero page at block 30521
HINT: Please REINDEX it.
```

Saat indeks rusak dan autovacuum mencoba untuk tetap berjalan di tabel, maka Anda bersaing dengan sesi autovacuum yang sudah berjalan. Jika Anda mengeluarkan perintah "REINDEX", kunci eksklusif akan diambil di tabel. Operasi penulisan, dan juga operasi membaca yang menggunakan indeks spesifik tersebut, akan diblokir.

Untuk mengindeks ulang tabel saat autovacuum berjalan di tabel

1. Buka dua sesi ke basis data yang berisi tabel yang ingin divakum. Untuk sesi kedua, gunakan "layar" atau peralatan lain yang dapat mempertahankan sesi jika koneksi terputus.
2. Dalam sesi pertama, dapatkan PID dari sesi autovacuum yang berjalan di tabel.

   Jalankan kueri berikut untuk mendapatkan PID dari sesi autovacuum.

   ```
   SELECT datname, usename, pid, current_timestamp - xact_start
   AS xact_runtime, query
   FROM pg_stat_activity WHERE upper(query) like '%VACUUM%' ORDER BY
   xact_start;
   ```

3. Dalam sesi kedua, munculkan perintah pengindeksan ulang.

   ```
   \timing on
   Timing is on.
   reindex index pgbench_branches_test_index;
   REINDEX
     Time: 9.966 ms
   ```

4. Dalam sesi pertama, jika autovacuum memblokir proses, Anda akan melihat di `pg_stat_activity` bahwa proses menunggu untuk sesi vakum diberi tanda "T". Dalam hal ini, Anda mengakhiri proses autovacuum.

```
SELECT pg_terminate_backend('the_pid');
```

Di titik ini, sesi Anda dimulai. Perlu diperhatikan bahwa proses autovacuum akan segera dimulai ulang karena tabel ini mungkin merupakan urutan tertinggi di daftar pekerjaan.

5. Mulai perintah Anda di sesi kedua, lalu akhiri proses autovacuum di sesi 1.

Mengelola autovacuum dengan indeks berukuran besar

Sebagai bagian dari operasinya, autovacuum akan melakukan beberapa fase vakum saat berjalan di tabel. Sebelum tabel dibersihkan, semua indeksnya akan divakum terlebih dahulu. Fase menghapus beberapa indeks berukuran besar akan menghabiskan banyak waktu dan sumber daya. Oleh karena itu, sebagai praktik terbaik, pastikan Anda mengontrol jumlah indeks di tabel dan menyingkirkan indeks yang tidak digunakan.

Untuk proses ini, pertama-tama periksa ukuran indeks keseluruhan. Kemudian, tentukan apakah ada indeks yang berpotensi tidak digunakan dan dapat dihapus seperti yang ditunjukkan dalam contoh berikut.

Untuk memeriksa ukuran tabel beserta indeksnya

```
postgres=> select pg_size_pretty(pg_relation_size('pgbench_accounts'));
pg_size_pretty
6404 MB
(1 row)
```

```
postgres=> select pg_size_pretty(pg_indexes_size('pgbench_accounts'));
pg_size_pretty
11 GB
(1 row)
```

Dalam contoh ini, ukuran indeks lebih besar dari tabel. Perbedaan ini dapat menyebabkan masalah performa karena indeks membengkak atau tidak digunakan sehingga berdampak pada operasi autovacuum serta operasi penyisipan.

Untuk memeriksa indeks yang tidak digunakan

Dengan menggunakan tampilan pg_stat_user_indexes, Anda dapat memeriksa seberapa sering indeks digunakan dengan kolom idx_scan. Dalam contoh berikut, indeks yang tidak digunakan memiliki nilai idx_scan dari 0.

```
postgres=> select * from pg_stat_user_indexes where relname = 'pgbench_accounts' order
 by idx_scan desc;

relid  | indexrelid | schemaname | relname          | indexrelname          | idx_scan
 | idx_tup_read | idx_tup_fetch
-------+------------+------------+------------------+-----------------------+----------
+--------------+--------------
16433  | 16454      | public     | pgbench_accounts | index_f               | 6
 | 6            | 0
16433  | 16450      | public     | pgbench_accounts | index_b               | 3
 | 199999       | 0
16433  | 16447      | public     | pgbench_accounts | pgbench_accounts_pkey | 0
 | 0            | 0
16433  | 16452      | public     | pgbench_accounts | index_d               | 0
 | 0            | 0
16433  | 16453      | public     | pgbench_accounts | index_e               | 0
 | 0            | 0
16433  | 16451      | public     | pgbench_accounts | index_c               | 0
 | 0            | 0
16433  | 16449      | public     | pgbench_accounts | index_a               | 0
 | 0            | 0
(7 rows)
```

```
postgres=> select schemaname, relname, indexrelname, idx_scan from pg_stat_user_indexes
 where relname = 'pgbench_accounts' order by idx_scan desc;

schemaname  | relname          | indexrelname          | idx_scan
------------+------------------+-----------------------+----------
public      | pgbench_accounts | index_f               | 6
public      | pgbench_accounts | index_b               | 3
public      | pgbench_accounts | pgbench_accounts_pkey | 0
public      | pgbench_accounts | index_d               | 0
public      | pgbench_accounts | index_e               | 0
public      | pgbench_accounts | index_c               | 0
public      | pgbench_accounts | index_a               | 0
(7 rows)
```

> **Note**
>
> Statistik ini bersifat inkremental sejak statistik diatur ulang. Misalkan Anda memiliki indeks yang hanya digunakan pada akhir kuartal bisnis atau hanya untuk laporan tertentu. Ada kemungkinan bahwa indeks ini belum digunakan sejak statistik diatur ulang. Untuk informasi selengkapnya, lihat Fungsi Statistik. Indeks yang digunakan untuk menerapkan keunikan tidak akan memiliki pemindaian yang dilakukan dan tidak boleh diidentifikasi sebagai indeks yang tidak digunakan. Untuk mengidentifikasi indeks yang tidak digunakan, Anda harus memiliki pengetahuan mendalam tentang aplikasi dan pertanyaannya.

Untuk memeriksa kapan statistik terakhir basis data disetel ulang, gunakan `pg_stat_database`

```
postgres=> select datname, stats_reset from pg_stat_database where datname =
 'postgres';

datname   | stats_reset
----------+-------------------------------
postgres  | 2022-11-17 08:58:11.427224+00
(1 row)
```

Melakukan vakum di tabel secepat mungkin

RDS untuk PostgreSQL 12 dan versi lebih tinggi

Jika Anda memiliki terlalu banyak indeks dalam tabel besar, instans DB Anda mungkin mendekati penyelesaian ID transaksi (XID), yaitu saat penghitung XID mendekati nol. Jika tidak terkendali, situasi ini dapat mengakibatkan kehilangan data. Namun, Anda dapat dengan cepat melakuan vakum di tabel tanpa harus membersihkan indeks. Dalam RDS untuk PostgreSQL 12 dan versi lebih tinggi, Anda dapat menggunakan VACUUM dengan klausa `INDEX_CLEANUP`.

```
postgres=> VACUUM (INDEX_CLEANUP FALSE, VERBOSE TRUE) pgbench_accounts;

INFO: vacuuming "public.pgbench_accounts"
INFO: table "pgbench_accounts": found 0 removable, 8 nonremovable row versions in 1 out
 of 819673 pages
DETAIL: 0 dead row versions cannot be removed yet, oldest xmin: 7517
Skipped 0 pages due to buffer pins, 0 frozen pages.
CPU: user: 0.01 s, system: 0.00 s, elapsed: 0.01 s.
```

Jika sesi autovacuum sudah berjalan, Anda harus menghentikannya agar dapat memulai VACUUM manual. Untuk informasi cara melakukan pembekuan vakum manual, lihat Melakukan pembekuan vakum manual.

> **Note**
>
> Melewatkan pembersihan indeks secara terus-menerus dapat menyebabkan indeks menggembung sehingga berdampak pada performa pemindaian secara keseluruhan. Sebagai praktik terbaik, gunakan prosedur sebelumnya hanya untuk mencegah penyelesaian ID transaksi.

RDS untuk PostgreSQL 11 dan versi lama

Namun, dalam RDS untuk PostgreSQL 11 dan versi lama, satu-satunya cara agar vakum dapat selesai lebih cepat adalah dengan mengurangi jumlah indeks di tabel. Menghapus sementara indeks dapat memengaruhi rencana kueri. Sebaiknya Anda mengapus sementara indeks yang tidak digunakan terlebih dahulu, lalu menghapus sementara indeks saat penyelesaian XID sangat dekat. Setelah proses vakum selesai, Anda dapat membuat ulang indeks ini.

Parameter lain yang memengaruhi autovacuum

Kueri berikut menunjukkan nilai dari beberapa parameter yang secara langsung memengaruhi autovacuum dan perilakunya. Parameter autovacuum dijelaskan dengan lengkap dalam dokumentasi PostgreSQL.

```
SELECT name, setting, unit, short_desc
FROM pg_settings
WHERE name IN (
'autovacuum_max_workers',
'autovacuum_analyze_scale_factor',
'autovacuum_naptime',
'autovacuum_analyze_threshold',
'autovacuum_analyze_scale_factor',
'autovacuum_vacuum_threshold',
'autovacuum_vacuum_scale_factor',
'autovacuum_vacuum_threshold',
'autovacuum_vacuum_cost_delay',
```

```
'autovacuum_vacuum_cost_limit',
'vacuum_cost_limit',
'autovacuum_freeze_max_age',
'maintenance_work_mem',
'vacuum_freeze_min_age');
```

Meskipun semua ini memengaruhi autovacuum, beberapa hal yang paling penting adalah:

- maintenance_work_mem
- autovacuum_freeze_max_age
- autovacuum_max_workers
- autovacuum_vacuum_cost_delay
- autovacuum_vacuum_cost_limit

Mengatur parameter autovacuum tingkat tabel

Anda dapat mengatur parameter penyimpanan terkait autovacuum di tingkat tabel, dan biasanya cara ini bisa lebih baik daripada harus mengubah perilaku seluruh basis data. Untuk tabel besar, Anda mungkin perlu mengatur pengaturan yang agresif dan tidak ingin membuat autovacuum berperilaku seperti itu untuk semua tabel.

Kueri berikut menunjukkan tabel mana yang saat ini memiliki opsi tingkat tabel.

```
SELECT relname, reloptions
FROM pg_class
WHERE reloptions IS NOT null;
```

Pengaturan ini mungkin berguna pada tabel yang jauh lebih besar daripada tabel lainnya, misalnya. Misalkan Anda memiliki satu tabel berukuran 300 GB dan 30 table lainnya berukuran kurang dari 1 GB. Dalam hal ini, Anda dapat mengatur beberapa parameter khusus untuk tabel besar sehingga tidak perlu mengubah perilaku dari seluruh sistem.

```
ALTER TABLE mytable set (autovacuum_vacuum_cost_delay=0);
```

Melakukan tindakan ini akan menonaktifkan penundaan autovacuum berbasis biaya untuk tabel ini dengan mengorbankan lebih banyak penggunaan sumber daya di sistem Anda. Biasanya, jeda autovacuum untuk `autovacuum_vacuum_cost_delay` setiap kali `autovacuum_cost_limit`

tercapai. Untuk mengetahui detail selengkapnya, lihat dokumentasi PostgreSQL tentang pemvakuman berbasis biaya.

Melakukan log aktivitas autovacuum dan vakum

Informasi mengenai aktivitas autovacuum dikirim ke `postgresql.log` berdasarkan level yang ditentukan dalam parameter `rds.force_autovacuum_logging_level`. Berikut ini adalah nilai yang diizinkan untuk parameter ini dan versi PostgreSQL dengan nilai berupa pengaturan default:

- `disabled` (PostgreSQL 10, PostgreSQL 9.6)
- `debug5`, `debug4`, `debug3`, `debug2`, `debug1`
- `info` (PostgreSQL 12, PostgreSQL 11)
- `notice`
- `warning` (PostgreSQL 13 dan versi lebih tinggi)
- `error`, log `fatal`, `panic`

`rds.force_autovacuum_logging_level` berfungsi dengan parameter `log_autovacuum_min_duration`. Nilai parameter `log_autovacuum_min_duration` adalah ambang batas (dalam milidetik) tempat tindakan autovacuum dicatat. Pengaturan `-1` tidak mencatat apa pun, sedangkan pengaturan 0 mencatat semua tindakan. Seperti halnya`rds.force_autovacuum_logging_level`, nilai default untuk `log_autovacuum_min_duration` bergantung pada versi, sebagai berikut:

- `10000 ms` – PostgreSQL 14, PostgreSQL 13, PostgreSQL 12, dan PostgreSQL 11
- `(empty)` – Tidak ada nilai default untuk PostgreSQL 10 dan PostgreSQL 9.6

Sebaiknya Anda mengatur `rds.force_autovacuum_logging_level` ke `WARNING`. Sebaiknya Anda juga mengatur `log_autovacuum_min_duration` ke nilai dari 1000 hingga 5000. Pengaturan aktivitas 5000 log yang membutuhkan waktu lebih dari 5.000 milidetik. Pengaturan apa pun selain -1 juga akan mencatat pesan jika tindakan autovacuum dilewati karena kunci yang bertentangan atau hubungan yang dihapus sementara secara bersamaan. Untuk informasi selengkapnya, lihat Automatic Vacuuming dalam dokumentasi PostgreSQL.

Untuk memecahkan masalah, Anda dapat mengubah parameter `rds.force_autovacuum_logging_level` ke salah satu level debug, dari `debug1` hingga `debug5` untuk informasi panjang. Sebaiknya Anda menggunakan pengaturan debug untuk jangka

waktu singkat dan hanya untuk tujuan pemecahan masalah. Untuk mempelajari selengkapnya, lihat When to log dalam dokumentasi PostgreSQL.

> **Note**
>
> PostgreSQL memungkinkan akun `rds_superuser` untuk melihat sesi autovacuum di `pg_stat_activity`. Misalnya, Anda dapat mengidentifikasi dan mengakhiri sesi autovacuum yang memblokir perintah agar tidak berjalan, atau berjalan lebih lambat daripada perintah vakum yang dikeluarkan secara manual.

Bekerja dengan mekanisme pencatatan log yang didukung oleh RDS for PostgreSQL

Ada beberapa parameter, ekstensi, dan item lain yang dapat dikonfigurasi dan Anda atur untuk mencatat aktivitas yang terjadi di instans DB PostgreSQL. Sumber daya yang dimaksud meliputi:

- Parameter `log_statement` dapat digunakan untuk mencatat aktivitas pengguna di basis data PostgreSQL. Untuk mempelajari pencatatan log RDS for PostgreSQL dan cara memantau log selengkapnya, lihat File log basis data RDS for PostgreSQL.
- Parameter `rds.force_admin_logging_level` mencatat tindakan yang dilakukan oleh pengguna internal Amazon RDS (rdsadmin) di basis data instans DB. Parameter ini akan menulis output ke kesalahan log PostgreSQL. Nilai yang diizinkan adalah `disabled`, `debug5`, `debug4`, `debug3`, `debug2`, `debug1`, `info`, `notice`, `warning`, `error`, log, `fatal`, dan `panic`. Nilai default-nya adalah `disabled`.
- Parameter `rds.force_autovacuum_logging_level` dapat diatur untuk mengambil berbagai operasi autovacuum di log kesalahan PostgreSQL. Untuk informasi selengkapnya, lihat Melakukan log aktivitas autovacuum dan vakum.
- Ekstensi PostgreSQL Audit (pgAaudit) dapat diinstal dan dikonfigurasi untuk mengambil aktivitas di tingkat sesi atau di tingkat objek. Untuk informasi selengkapnya, lihat Menggunakan pgAudit untuk membuat log aktivitas basis data.
- Dengan ekstensi `log_fdw`, Anda dapat mengakses log mesin basis data menggunakan SQL. Untuk informasi selengkapnya, lihat Menggunakan ekstensi log_fdw untuk mengakses log DB menggunakan SQL.
- Pustaka `pg_stat_statements` ditentukan sebagai default untuk parameter `shared_preload_libraries` dalam RDS for PostgreSQL versi 10 dan yang lebih baru.

Pustaka inilah yang dapat Anda gunakan untuk menganalisis kueri yang sedang berjalan. Pastikan `pg_stat_statements` diatur dalam grup parameter DB Anda. Untuk informasi cara memantau instans DB RDS for PostgreSQL menggunakan informasi yang disediakan pustaka ini selengkapnya, lihat Statistik SQL untuk RDS PostgreSQL.

- Parameter `log_hostname` mengambil log nama host dari setiap koneksi klien. Untuk RDS for PostgreSQL versi 12 dan versi yang lebih baru, parameter ini diatur ke `off` secara default. Jika diaktifkan, pastikan Anda memantau waktu koneksi sesi. Ketika diaktifkan, layanan menggunakan permintaan pencarian terbalik sistem nama domain (DNS) untuk mendapatkan nama host klien yang membuat koneksi dan menambahkannya ke log PostgreSQL. Hal ini memberikan dampak nyata selama koneksi sesi. Sebaiknya mengaktifkan parameter ini hanya untuk memecahkan masalah.

Secara umum, tujuan pencatatan log adalah agar DBA dapat memantau, menyesuaikan performa, dan memecahkan masalah. Banyak log yang diunggah secara otomatis ke Amazon CloudWatch atau Performance Insights. Di sini, log diurutkan dan dikelompokkan agar dapat memberikan metrik lengkap untuk instans DB Anda. Untuk mempelajari pemantauan dan metrik Amazon RDS selengkapnya, lihat Memantau metrik dalam instans Amazon RDS.

Mengelola file sementara dengan PostgreSQL

Di PostgreSQL, kueri yang melakukan operasi pengurutan dan hash menggunakan memori instans untuk menyimpan hasil hingga nilai yang ditentukan dalam parameter `work_mem`. Jika memori instans tidak cukup, file sementara akan dibuat untuk menyimpan hasil. Hasil ini ditulis ke disk untuk menyelesaikan eksekusi kueri. Kemudian, file-file ini secara otomatis dihapus setelah kueri selesai. Dalam RDS for PostgreSQL, file ini disimpan di Amazon EBS pada volume data. Untuk informasi selengkapnya, lihat Penyimpanan instans DB Amazon RDS. Sebaiknya Anda terus memantau metrik `FreeStorageSpace` yang diterbitkan di CloudWatch untuk memastikan bahwa instans DB Anda memiliki ruang penyimpanan kosong yang cukup. Untuk informasi selengkapnya, lihat `FreeStorageSpace` .

Sebaiknya gunakan instans Amazon RDS Optimized Reads untuk beban kerja yang melibatkan beberapa kueri bersamaan yang meningkatkan penggunaan file sementara. Instans ini menggunakan penyimpanan tingkat blok solid state drive (SSD) berbasis Non-Volatile Memory Express (NVMe) lokal untuk menempatkan file sementara. Untuk informasi selengkapnya, lihat Amazon RDS Optimized Reads.

Anda dapat menggunakan parameter dan fungsi berikut untuk mengelola file sementara dalam instans Anda.

- **temp_file_limit** – Parameter ini membatalkan kueri apa pun yang melebihi ukuran temp_files dalam KB. Batas ini mencegah kueri apa pun berjalan tanpa henti dan menghabiskan ruang disk dengan file sementara. Anda dapat memperkirakan nilai menggunakan hasil dari parameter `log_temp_files`. Sebagai praktik terbaik, periksa perilaku beban kerja dan tetapkan batas sesuai dengan estimasi. Contoh berikut menunjukkan bagaimana kueri dibatalkan saat melampaui batas.

```
postgres=> select * from pgbench_accounts, pg_class, big_table;
```

```
ERROR: temporary file size exceeds temp_file_limit (64kB)
```

- **log_temp_files** – Parameter ini mengirimkan pesan ke postgresql.log ketika file sementara dari sebuah sesi dihapus. Parameter ini menghasilkan log setelah kueri berhasil diselesaikan. Oleh karena itu, ini mungkin tidak membantu dalam memecahkan masalah kueri yang aktif dan berjalan lama.

  Contoh berikut menunjukkan bahwa ketika kueri berhasil diselesaikan, entri dicatat dalam file postgresql.log, sedangkan file sementara dibersihkan.

```
2023-02-06 23:48:35 UTC:205.251.233.182(12456):adminuser@postgres:[31236]:LOG:
 temporary file: path "base/pgsql_tmp/pgsql_tmp31236.5", size 140353536
2023-02-06 23:48:35 UTC:205.251.233.182(12456):adminuser@postgres:[31236]:STATEMENT:
  select a.aid from pgbench_accounts a, pgbench_accounts b where a.bid=b.bid order by
 a.bid limit 10;
2023-02-06 23:48:35 UTC:205.251.233.182(12456):adminuser@postgres:[31236]:LOG:
 temporary file: path "base/pgsql_tmp/pgsql_tmp31236.4", size 180428800
2023-02-06 23:48:35 UTC:205.251.233.182(12456):adminuser@postgres:[31236]:STATEMENT:
  select a.aid from pgbench_accounts a, pgbench_accounts b where a.bid=b.bid order by
 a.bid limit 10;
```

- **pg_ls_tmpdir** – Fungsi yang tersedia dari RDS untuk PostgreSQL 13 dan versi yang lebih baru ini memberikan visibilitas terhadap penggunaan file sementara saat ini. Kueri yang sudah selesai tidak muncul di hasil fungsi. Dalam contoh berikut, Anda dapat melihat hasil dari fungsi ini.

```
postgres=> select * from pg_ls_tmpdir();
```

```
      name       |    size    |      modification
-----------------+------------+------------------------
 pgsql_tmp8355.1 | 1072250880 | 2023-02-06 22:54:56+00
 pgsql_tmp8351.0 | 1072250880 | 2023-02-06 22:54:43+00
 pgsql_tmp8327.0 | 1072250880 | 2023-02-06 22:54:56+00
 pgsql_tmp8351.1 |  703168512 | 2023-02-06 22:54:56+00
 pgsql_tmp8355.0 | 1072250880 | 2023-02-06 22:54:00+00
 pgsql_tmp8328.1 |  835031040 | 2023-02-06 22:54:56+00
 pgsql_tmp8328.0 | 1072250880 | 2023-02-06 22:54:40+00
(7 rows)
```

```
postgres=> select query from pg_stat_activity where pid = 8355;

query
-------------------------------------------------------------------------------------
select a.aid from pgbench_accounts a, pgbench_accounts b where a.bid=b.bid order by
 a.bid
(1 row)
```

Nama file mencakup ID pemrosesan (PID) dari sesi yang menghasilkan file sementara. Kueri yang lebih maju, seperti pada contoh berikut, melakukan penjumlahan file sementara untuk setiap PID.

```
postgres=> select replace(left(name, strpos(name, '.')-1),'pgsql_tmp','') as pid,
 count(*), sum(size) from pg_ls_tmpdir() group by pid;
```

```
 pid  | count |   sum
------+-------+------------
 8355 |     2 | 2144501760
 8351 |     2 | 2090770432
 8327 |     1 | 1072250880
 8328 |     2 | 2144501760
```

```
(4 rows)
```

- **pg_stat_statements** – Jika Anda mengaktifkan parameter pg_stat_statement, Anda dapat melihat rata-rata penggunaan file sementara per panggilan. Anda dapat mengidentifikasi query_id dari kueri dan menggunakannya untuk memeriksa penggunaan file sementara seperti yang ditunjukkan pada contoh berikut.

```
postgres=> select queryid from pg_stat_statements where query like 'select a.aid from
 pgbench%';
```

```
       queryid
----------------------
 -7170349228837045701
(1 row)
```

```
postgres=> select queryid, substr(query,1,25), calls, temp_blks_read/calls
 temp_blks_read_per_call, temp_blks_written/calls temp_blks_written_per_call from
 pg_stat_statements where queryid = -7170349228837045701;
```

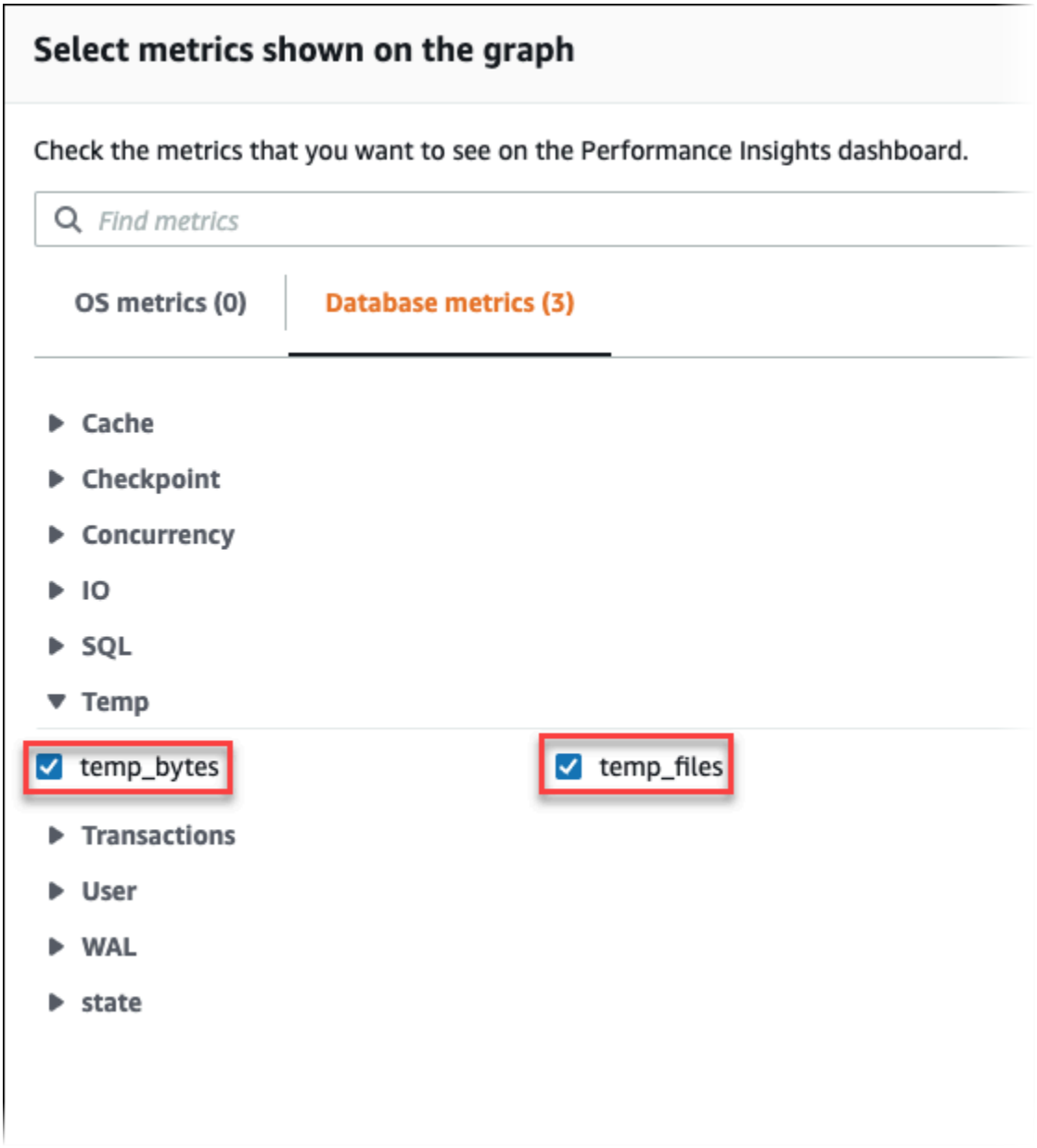
```
       queryid        |          substr           | calls | temp_blks_read_per_call |
 temp_blks_written_per_call
----------------------+---------------------------+-------+-------------------------
+----------------------------
 -7170349228837045701 | select a.aid from pgbench |    50 |                  239226 |
                     388678
(1 row)
```

- **Performance Insights** – Di dasbor Wawasan Performa, Anda dapat melihat penggunaan file sementara dengan mengaktifkan metrik temp_bytes dan temp_files. Kemudian, Anda dapat melihat rata-rata kedua metrik ini dan melihat sejauh mana kesesuaiannya dengan beban kerja kueri. Tampilan dalam Wawasan Performa tidak secara khusus menampilkan kueri yang menghasilkan file sementara. Namun, jika Anda menggabungkan Wawasan Performa dengan kueri yang ditampilkan untuk `pg_ls_tmpdir`, Anda dapat memecahkan masalah, menganalisis, dan menentukan perubahan dalam beban kerja kueri.

Untuk informasi selengkapnya tentang cara menganalisis metrik dan kueri dengan Wawasan Performa, lihat Menganalisis metrik dengan dasbor Wawasan Performa

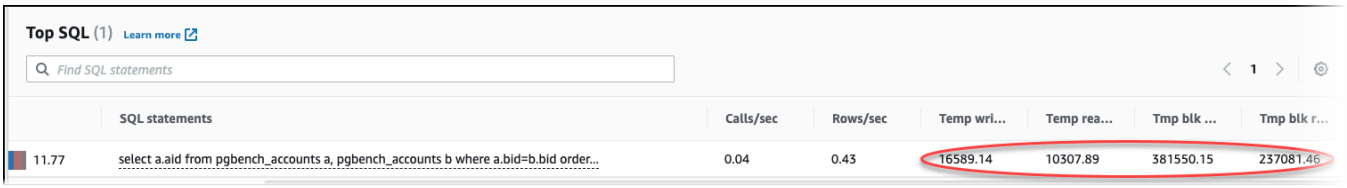
Untuk melihat penggunaan file sementara dengan Wawasan Performa

1. Di dasbor Wawasan Performa, pilih Kelola Metrik.
2. Pilih Metrik basis data, lalu pilih metrik temp_bytes dan temp_files seperti yang ditunjukkan pada gambar berikut.

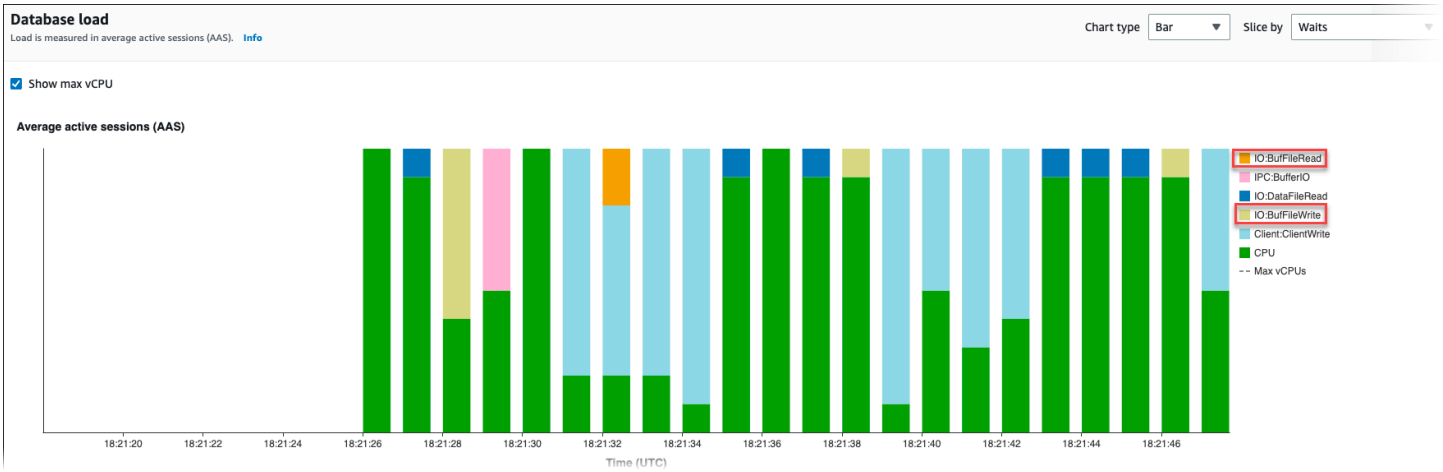


3. Di tab SQL Teratas, pilih ikon Preferensi.

4. Di jendela Preferensi, aktifkan statistik berikut agar muncul di tab SQL Teratas dan pilih Lanjutkan.
   - Temp writes/detik
   - Temp reads/detik
   - Tmp blk write/panggilan
   - Tmp blk read/panggilan
5. File sementara rusak saat digabungkan kueri yang ditampilkan untuk `pg_ls_tmpdir`, seperti yang ditunjukkan pada contoh berikut.



Peristiwa `IO:BufFileRead` dan `IO:BufFileWrite` terjadi ketika kueri teratas di beban kerja Anda sering membuat file sementara. Anda dapat menggunakan Wawasan Performa untuk mengidentifikasi kueri teratas yang menunggu pada `IO:BufFileRead` dan `IO:BufFileWrite` dengan meninjau Sesi Aktif Rata-rata (AAS) di bagian Muatan Basis Data dan SQL Teratas.



Untuk informasi selengkapnya tentang cara menganalisis kueri teratas dan muatan berdasarkan peristiwa tunggu dengan Wawasan Performa, lihat Ringkasan tab SQL Teratas. Anda harus mengidentifikasi dan menyetel kueri yang menyebabkan peningkatan penggunaan file sementara dan peristiwa tunggu terkait. Untuk informasi selengkapnya tentang peristiwa tunggu dan remediasi ini, lihat IO:BufFileRead dan IO:BufFileWrite.

> **Note**
>
> Parameter `work_mem` mengontrol ketika operasi pengurutan kehabisan memori dan hasilnya ditulis ke dalam file sementara. Sebaiknya Anda tidak mengubah pengaturan parameter ini lebih tinggi dari nilai default karena akan memungkinkan setiap sesi basis data mengonsumsi lebih banyak memori. Selain itu, satu sesi yang melakukan penggabungan dan pengurutan kompleks dapat melakukan operasi paralel di mana setiap operasi mengonsumsi memori. Sebagai praktik terbaik, ketika Anda memiliki laporan besar dengan beberapa penggabungan dan pengurutan, atur parameter ini pada tingkat sesi dengan menggunakan perintah `SET work_mem`. Kemudian, perubahan hanya diterapkan pada sesi saat ini dan tidak mengubah nilai secara global.

Menggunakan pgBadger untuk analisis log dengan PostgreSQL

Anda dapat menggunakan penganalisis log seperti pgBadger untuk menganalisis log PostgreSQL. Dokumentasi pgBadger menyatakan bahwa pola %l (baris log untuk sesi atau proses) harus merupakan bagian dari awalan. Namun, jika Anda memberikan RDS `log_line_prefix` saat ini sebagai parameter pgBadger, laporan tetap akan dihasilkan.

Misalnya, perintah berikut memformat file log Amazon RDS for PostgreSQL tertanggal 04-02-2014 dengan benar menggunakan pgBadger.

```
./pgbadger -f stderr -p '%t:%r:%u@%d:[%p]:' postgresql.log.2014-02-04-00
```

Menggunakan PGSnapper untuk memantau PostgreSQL

Anda dapat menggunakan PGSnapper untuk membantu pengumpulan berkala statistik dan metrik terkait performa Amazon RDS for PostgreSQL. Untuk informasi selengkapnya, lihat Memantau performa Amazon RDS for PostgreSQL menggunakan PGSnapper.

Bekerja dengan parameter pada instans DB RDS for PostgreSQL

Dalam beberapa kasus, Anda mungkin membuat instans DB RDS for PostgreSQL tanpa menentukan grup parameter kustom. Jika demikian, instans DB Anda dibuat menggunakan grup parameter default untuk versi PostgreSQL yang dipilih. Misalnya, Anda membuat instans DB RDS for PostgreSQL menggunakan PostgreSQL 13.3. Dalam hal ini, instans DB dibuat menggunakan nilai dalam grup parameter untuk rilis PostgreSQL 13, `default.postgres13`.

Anda juga dapat membuat grup parameter DB kustom Anda sendiri. Anda perlu melakukan ini jika ingin memodifikasi pengaturan untuk instans DB RDS for PostgreSQL dari nilai default-nya. Untuk mempelajari caranya, lihat Bekerja dengan grup parameter.

Anda dapat melacak pengaturan pada instans DB RDS for PostgreSQL melalui beberapa cara berbeda. Anda dapat menggunakan AWS Management Console, AWS CLI, atau Amazon RDS API. Anda juga dapat membuat kueri pada nilai dari tabel `pg_settings` PostgreSQL instans Anda, seperti yang ditunjukkan berikut.

```
SELECT name, setting, boot_val, reset_val, unit
 FROM pg_settings
 ORDER BY name;
```

Untuk mempelajari selengkapnya tentang nilai yang ditampilkan dari kueri ini, lihat `pg_settings` dalam dokumentasi PostgreSQL.

Berhati-hatilah saat mengubah pengaturan untuk `max_connections` dan `shared_buffers` pada instans DB RDS for PostgreSQL. Misalnya, Anda memodifikasi pengaturan untuk `max_connections` atau `shared_buffers`, dan Anda menggunakan nilai yang terlalu tinggi untuk beban kerja yang sebenarnya. Dalam hal ini, maka instans DB RDS for PostgreSQL tidak akan dimulai. Jika ini terjadi, Anda akan melihat kesalahan seperti berikut di `postgres.log`.

```
2018-09-18 21:13:15 UTC::@:[8097]:FATAL:  could not map anonymous shared memory: Cannot
 allocate memory
2018-09-18 21:13:15 UTC::@:[8097]:HINT:  This error usually means that PostgreSQL's
 request for a shared memory segment
exceeded available memory or swap space. To reduce the request size (currently
 3514134274048 bytes), reduce
PostgreSQL's shared memory usage, perhaps by reducing shared_buffers or
 max_connections.
```

Namun, Anda tidak dapat mengubah nilai pengaturan apa pun yang terdapat dalam grup parameter DB RDS for PostgreSQL. Untuk mengubah pengaturan setiap parameter, buat grup parameter DB kustom. Kemudian, ubah pengaturan di grup kustom tersebut, lalu terapkan grup parameter kustom ke instans DB RDS for PostgreSQL. Untuk mempelajari selengkapnya, lihat Bekerja dengan grup parameter.

Ada dua jenis parameter DB RDS for PostgreSQL.

- Parameter statis – Parameter statis mengharuskan instans DB RDS for PostgreSQL di-boot ulang setelah perubahan agar nilai baru dapat diterapkan.
- Parameter dinamis – Parameter dinamis tidak memerlukan boot ulang setelah mengubah pengaturannya.

> **Note**
>
> Jika instans DB RDS for PostgreSQL menggunakan grup parameter DB kustom Anda sendiri, Anda dapat mengubah nilai parameter dinamis pada instans DB yang sedang berjalan. Anda dapat melakukannya menggunakan AWS Management Console, AWS CLI, atau API Amazon RDS.

Jika memiliki hak istimewa untuk melakukan tindakan ini, Anda juga dapat mengubah nilai parameter menggunakan perintah `ALTER DATABASE`, `ALTER ROLE`, dan `SET`.

Daftar parameter instans DB RDS for PostgreSQL

Tabel berikut mencantumkan beberapa (tetapi tidak semua) parameter yang tersedia dalam instans DB RDS for PostgreSQL. Untuk melihat semua parameter yang tersedia, Anda menggunakan describe-db-parameters AWS CLI perintah. Misalnya, untuk mendapatkan daftar semua parameter yang tersedia di grup parameter default untuk RDS for PostgreSQL versi 13, jalankan perintah berikut ini.

```
aws rds describe-db-parameters --db-parameter-group-name default.postgres13
```

Anda juga dapat menggunakan Konsol. Pilih Grup parameter dari menu Amazon RDS, lalu pilih grup parameter yang tersedia di menu Wilayah AWS.

Nama parameter	Apply_Type	Deskripsi
application_name	Dinamis	Mengatur nama aplikasi yang akan dilaporkan dalam statistik dan log.
archive_command	Dinamis	Menetapkan perintah shell yang akan dipanggil untuk mengarsipkan file WAL.
array_nulls	Dinamis	Memungkinkan input elemen NULL dalam array.
authentication_timeout	Dinamis	Mengatur waktu maksimum yang diizinkan untuk menyelesaikan autentikasi klien.
autovacuum	Dinamis	Memulai subproses autovacuum.
autovacuum_analyze_scale_factor	Dinamis	Jumlah penyisipan, pembaruan, atau penghapusan tuple sebelum dianalisis sebagai pecahan dari reltuple.
autovacuum_analyze_threshold	Dinamis	Jumlah minimum penyisipan, pembaruan, atau penghapusan tuple sebelum dianalisis.
autovacuum_freeze_max_age	Statis	Usia untuk melakukan autovacuum tabel guna mencegah penyelesaian ID transaksi.

Nama parameter	Apply_Type	Deskripsi
autovacuum_naptime	Dinamis	Waktu tidur selama autovacuum berjalan.
autovacuum_max_workers	Statis	Mengatur jumlah maksimum proses pekerja autovacuum yang berjalan secara bersamaan.
autovacuum_vacuum_cost_delay	Dinamis	Penundaan biaya vakum, dalam milidetik, untuk autovacuum.
autovacuum_vacuum_cost_limit	Dinamis	Jumlah biaya vakum yang tersedia sebelum napping, untuk autovacuum.
autovacuum_vacuum_scale_factor	Dinamis	Jumlah pembaruan atau penghapusan tuple sebelum divakum sebagai pecahan dari retuple.
autovacuum_vacuum_threshold	Dinamis	Jumlah minimum pembaruan atau penghapusan tuple sebelum divakum.
backslash_quote	Dinamis	Mengatur apakah garis miring terbalik (\) diizinkan dalam string literal atau tidak.
bgwriter_delay	Dinamis	Waktu tidur latar belakang penulis di sela-sela putaran.
bgwriter_lru_maxpages	Dinamis	Jumlah maksimum halaman LRU penulis latar belakang yang akan dibersihkan per putaran.
bgwriter_lru_multiplier	Dinamis	Kelipatan dari penggunaan buffer rata-rata yang akan dikosongkan per putaran.
bytea_output	Dinamis	Mengatur format output untuk byte.
check_function_bodies	Dinamis	Memeriksa konten fungsi selama CREATE FUNCTION.
checkpoint_completion_target	Dinamis	Waktu yang dihabiskan untuk membersihkan buffer kotor selama operasi titik pemeriksaan, sebagai bagian dari interval titik pemeriksaan.

Nama parameter	Apply_Type	Deskripsi
checkpoint_segments	Dinamis	Mengatur jarak maksimum dalam segmen log antara titik pemeriksaan write-ahead log (WAL).
checkpoint_timeout	Dinamis	Mengatur waktu maksimum antara titik pemeriksaan WAL otomatis.
checkpoint_warning	Dinamis	Mengaktifkan peringatan jika segmen titik pemeriksaan diisi lebih sering daripada ini.
client_connection_check_interval	Dinamis	Menetapkan interval waktu di antara pemeriksaan pemutusan koneksi saat menjalankan kueri.
client_encoding	Dinamis	Mengatur pengenkodean set karakter klien.
client_min_messages	Dinamis	Mengatur tingkatan pesan yang dikirimkan kepada klien.
commit_delay	Dinamis	Mengatur penundaan dalam mikrodetik antara transaksi commit dan melakukan pembersihan WAL ke disk.
commit_siblings	Dinamis	Mengatur minimum transaksi terbuka serentak sebelum melakukan commit_delay.
constraint_exclusion	Dinamis	Memungkinkan perencana untuk menggunakan batasan agar dapat mengoptimalkan kueri.
cpu_index_tuple_cost	Dinamis	Menetapkan perkiraan perencana untuk biaya pemrosesan setiap entri indeks selama pemindaian indeks.
cpu_operator_cost	Dinamis	Menetapkan perkiraan perencana untuk biaya pemrosesan setiap operator atau panggilan fungsi.
cpu_tuple_cost	Dinamis	Menetapkan perkiraan perencana untuk biaya pemrosesan setiap tuple (baris).

Nama parameter	Apply_Type	Deskripsi
cursor_tuple_fraction	Dinamis	Menetapkan perkiraan perencana untuk pecahan dari baris kursor yang akan diambil.
datestyle	Dinamis	Mengatur format tampilan nilai tanggal dan waktu.
deadlock_timeout	Dinamis	Mengatur waktu menunggu kunci sebelum memeriksa deadlock.
debug_pretty_print	Dinamis	Mengindentasi tampilan hierarki penguraian dan rencana.
debug_print_parse	Dinamis	Membuat log hierarki penguraian setiap kueri.
debug_print_plan	Dinamis	Membuat log rencana eksekusi setiap kueri.
debug_print_rewritten	Dinamis	Membuat log hierarki penguraian yang ditulis ulang oleh setiap kueri.
default_statistics_target	Dinamis	Mengatur target statistik default.
default_tablespace	Dinamis	Mengatur tablespace default untuk membuat tabel dan indeks.
default_transaction_deferrable	Dinamis	Mengatur status default yang dapat ditangguhkan dari transaksi baru.
default_transaction_isolation	Dinamis	Menetapkan tingkat isolasi transaksi dari setiap transaksi baru.
default_transaction_read_only	Dinamis	Mengatur status hanya-baca default dari transaksi baru.
default_with_oids	Dinamis	Membuat tabel baru dengan ID objek (OID) berdasarkan default-nya.
effective_cache_size	Dinamis	Mengatur asumsi perencana ukuran cache disk.

Nama parameter	Apply_Type	Deskripsi
effective_io_concurrency	Dinamis	Jumlah permintaan serentak yang dapat ditangani secara efisien oleh subsistem disk.
enable_bitmapscan	Dinamis	Memungkinkan penggunaan rencana pemindaian bitmap oleh perencana.
enable_hashagg	Dinamis	Memungkinkan penggunaan rencana agregasi hash oleh perencana.
enable_hashjoin	Dinamis	Memungkinkan penggunaan rencana hash join oleh perencana.
enable_indexscan	Dinamis	Memungkinkan penggunaan rencana pemindaian indeks oleh perencana.
enable_material	Dinamis	Memungkinkan penggunaan materialisasi oleh perencana.
enable_mergejoin	Dinamis	Memungkinkan penggunaan rencana merge join oleh perencana.
enable_nestloop	Dinamis	Memungkinkan penggunaan rencana nested-loop join oleh perencana.
enable_seqscan	Dinamis	Memungkinkan penggunaan rencana pemindaian sekuensial oleh perencana.
enable_sort	Dinamis	Memungkinkan penggunaan langkah singkat eksplisit oleh perencana.
enable_tidscan	Dinamis	Memungkinkan penggunaan rencana pemindaian TID oleh perencana.
escape_string_warning	Dinamis	Memperingatkan tentang escape garis miring terbalik (\) dalam string literal biasa.

Nama parameter	Apply_Type	Deskripsi
extra_float_digits	Dinamis	Mengatur jumlah digit yang ditampilkan untuk nilai floating-point.
from_collapse_limit	Dinamis	Mengatur ukuran daftar FROM yang tidak menciutkan subkueri.
fsync	Dinamis	Memaksa sinkronisasi pembaruan ke disk.
full_page_writes	Dinamis	Menulis halaman penuh ke WAL saat pertama kali dimodifikasi setelah titik pemeriksaan.
geqo	Dinamis	Memungkinkan pengoptimalan kueri genetik.
geqo_effort	Dinamis	GEQO: upaya digunakan untuk mengatur default parameter GEQO lainnya.
geqo_generations	Dinamis	GEQO: jumlah iterasi algoritma.
geqo_pool_size	Dinamis	GEQO: jumlah individu dalam populasi.
geqo_seed	Dinamis	GEQO: seed untuk pemilihan jalur acak.
geqo_selection_bias	Dinamis	GEQO: tekanan selektif di dalam populasi.
geqo_threshold	Dinamis	Mengatur ambang batas item FROM yang menggunakan GEQO.
gin_fuzzy_search_limit	Dinamis	Mengatur hasil maksimum yang diperbolehkan untuk pencarian akurat oleh GIN.
hot_standby_feedback	Dinamis	Menentukan apakah hot standby mengirimkan pesan umpan balik ke standby utama atau standby hulu.
intervalstyle	Dinamis	Mengatur format tampilan nilai interval.

Nama parameter	Apply_Type	Deskripsi
join_collapse_limit	Dinamis	Menetapkan ukuran daftar FROM yang tidak meratakan konsep JOIN.
lc_messages	Dinamis	Mengatur bahasa untuk menampilkan pesan.
lc_monetary	Dinamis	Menetapkan lokal untuk memformat jumlah uang.
lc_numeric	Dinamis	Mengatur lokal untuk memformat angka.
lc_time	Dinamis	Mengatur lokal untuk memformat nilai tanggal dan waktu.
log_autovacuum_min_duration	Dinamis	Mengatur waktu berjalan minimum yang akan membuat log tindakan autovacuum.
log_checkpoints	Dinamis	Membuat log setiap titik pemeriksaan.
log_connections	Dinamis	Membuat log setiap koneksi yang berhasil.
log_disconnections	Dinamis	Membuat log dari akhir sebuah sesi, termasuk durasinya.
log_duration	Dinamis	Membuat log durasi setiap pernyataan SQL yang diselesaikan.
log_error_verbosity	Dinamis	Mengatur panjang pesan yang dicatat.
log_executor_stats	Dinamis	Menulis statistik performa pelaksana ke log server.
log_filename	Dinamis	Mengatur pola nama file untuk file log.
log_file_mode	Dinamis	Mengatur izin file untuk file log. Nilai default-nya adalah 0644.

Nama parameter	Apply_Type	Deskripsi
log_hostname	Dinamis	Membuat log nama host dalam log koneksi. Pada PostgreSQL 12 dan versi yang lebih baru, parameter ini 'nonaktif' secara default. Saat diaktifkan, koneksi akan menggunakan pencarian balik DNS untuk mendapatkan nama host yang terambil ke log koneksi. Jika mengaktifkan parameter ini, Anda harus memantau dampaknya pada waktu yang diperlukan agar dapat membuat koneksi.
log_line_prefix	Dinamis	Mengontrol informasi yang diawali untuk setiap baris log.
log_lock_waits	Dinamis	Membuat log waktu tunggu kunci yang panjang.
log_min_duration_statement	Dinamis	Menetapkan waktu berjalan minimum yang akan Membuat log pernyataan.
log_min_error_statement	Dinamis	Menyebabkan semua pernyataan yang menghasilkan kesalahan pada atau di atas level ini dicatat.
log_min_messages	Dinamis	Mengatur tingkat pesan yang dicatat.
log_parser_stats	Dinamis	Menulis statistik performa pengurai ke log server.
log_planner_stats	Dinamis	Menulis statistik performa perencana ke log server.
log_rotation_age	Dinamis	Rotasi file log otomatis akan terjadi setelah N menit.
log_rotation_size	Dinamis	Rotasi file log otomatis akan terjadi setelah N kilobita.
log_statement	Dinamis	Mengatur jenis pernyataan yang dicatat.
log_statement_stats	Dinamis	Menulis statistik performa kumulatif ke log server.

Nama parameter	Apply_Type	Deskripsi
log_temp_files	Dinamis	Membuat log penggunaan file sementara yang lebih besar dari angka kilobyte ini.
log_timezone	Dinamis	Mengatur zona waktu yang akan digunakan dalam pesan log.
log_truncate_on_rotation	Dinamis	Memotong file log yang ada dengan nama yang sama selama rotasi log.
logging_collector	Statis	Memulai subproses untuk mengambil output stderr dan/atau csvlog ke dalam file log.
maintenance_work_mem	Dinamis	Mengatur memori maksimum yang akan digunakan untuk operasi pemeliharaan.
max_connections	Statis	Mengatur jumlah maksimum koneksi serentak.
max_files_per_process	Statis	Mengatur jumlah maksimum file yang terbuka secara bersamaan untuk setiap proses server.
max_locks_per_transaction	Statis	Menetapkan jumlah maksimum kunci per transaksi.
max_pred_locks_per_transaction	Statis	Menetapkan jumlah maksimum kunci predikat per transaksi.
max_prepared_transactions	Statis	Menetapkan jumlah maksimum transaksi yang disiapkan secara bersamaan.
max_stack_depth	Dinamis	Mengatur kedalaman tumpukan maksimum, dalam kilobita.
max_standby_archive_delay	Dinamis	Mengatur penundaan maksimum sebelum membatalkan kueri saat hot standby sedang memproses data WAL yang diarsipkan.

Nama parameter	Apply_Type	Deskripsi
`max_standby_streaming_delay`	Dinamis	Mengatur penundaan maksimum sebelum membatalkan kueri saat hot standby sedang memproses data WAL yang dialirkan.
`max_wal_size`	Dinamis	Menetapkan ukuran WAL (MB) yang memicu titik pemeriksaan. Untuk semua versi setelah RDS for PostgreSQL 10, ukuran default-nya minimal 1 GB (1024 MB). Misalnya, pengaturan `max_wal_size` untuk RDS for PostgreSQL 14 adalah 2 GB (2048 MB). Gunakan perintah `SHOW max_wal_size;` pada instans DB RDS for PostgreSQL untuk melihat nilainya saat ini.
`min_wal_size`	Dinamis	Mengatur ukuran minimum untuk menyusutkan WAL. Untuk PostgreSQL versi 9.6 dan yang sebelumnya, pengaturan `min_wal_size` berada dalam unit 16 MB. Untuk PostgreSQL versi 10 dan yang lebih baru, pengaturan `min_wal_size` berada dalam unit 1 MB.
`quote_all_identifiers`	Dinamis	Menambahkan tanda kutip (") ke semua pengidentifikasi ketika membuat fragmen SQL.
`random_page_cost`	Dinamis	Mengatur perkiraan perencana untuk biaya dari halaman disk yang diambil secara tidak berurutan. Parameter ini tidak memiliki nilai kecuali manajemen rencana kueri (QPM) diaktifkan. Saat QPM aktif, nilai default-nya adalah 4.
`rds.adaptive_autovacuum`	Dinamis	Secara otomatis menyesuaikan parameter autovacuum setiap kali ambang batas ID transaksi terlampaui.

Nama parameter	Apply_Type	Deskripsi
rds.force_ssl	Dinamis	Membutuhkan penggunaan koneksi SSL. Nilai default diatur ke 1 (aktif) untuk RDS for PostgreSQL versi 15. Semua RDS for PostgreSQL versi utama 14 lainnya dan yang lebih lama memiliki nilai default yang disetel ke 0 (nonaktif).
rds.log_retention_period	Dinamis	Mengatur retensi log sedemikian rupa sehingga Amazon RDS dapat menghapus log PostgreSQL yang lebih lama dari n menit.
rds.restrict_password_commands	Statis	Membatasi individu yang dapat mengelola kata sandi untuk pengguna yang memiliki peran rds_password . Atur parameter ini ke 1 untuk mengaktifkan pembatasan kata sandi. Default-nya adalah 0.
search_path	Dinamis	Menetapkan urutan pencarian skema untuk nama yang tidak memenuhi syarat skema.
seq_page_cost	Dinamis	Mengatur perkiraan perencana untuk biaya dari halaman disk yang diambil secara berurutan.
session_replication_role	Dinamis	Menetapkan perilaku sesi untuk pemicu dan aturan penulisan ulang.
shared_buffers	Statis	Mengatur jumlah buffer memori bersama yang digunakan oleh server.
shared_preload_libraries	Statis	Memigrasi pustaka bersama untuk dimuat ke instans DB RDS for PostgreSQL. Nilai yang didukung meliputi auto_explain, orafce, pgaudit, pglogical, pg_bigm, pg_cron, pg_hint_plan, pg_prewarm, pg_similarity, pg_stat_statements, pg_tle, pg_transport, plprofiler, dan plrust.

Nama parameter	Apply_Type	Deskripsi
ssl	Dinamis	Mengaktifkan koneksi SSL.
sql_inheritance	Dinamis	Menyebabkan subtabel disertakan secara default dalam berbagai perintah.
ssl_renegotiation_limit	Dinamis	Menetapkan jumlah lalu lintas untuk dikirim dan diterima sebelum melakukan negosiasi ulang kunci enkripsi.
standard_conforming_strings	Dinamis	Menyebabkan string ... memperlakukan garis miring terbalik secara literal.
statement_timeout	Dinamis	Menetapkan durasi maksimum yang diizinkan untuk setiap pernyataan.
synchronize_seqscans	Dinamis	Memungkinkan pemindaian berurutan yang disinkronkan.
synchronous_commit	Dinamis	Mengatur tingkat sinkronisasi transaksi saat ini.
tcp_keepalives_count	Dinamis	Jumlah maksimum pengiriman ulang keepalive TCP.
tcp_keepalives_idle	Dinamis	Waktu antara penerbitan keepalive TCP.
tcp_keepalives_interval	Dinamis	Waktu antara pengiriman ulang keepalive TCP.
temp_buffers	Dinamis	Mengatur jumlah maksimum buffer sementara yang digunakan oleh setiap sesi.
temp_file_limit	Dinamis	Mengatur ukuran maksimum file sementara dapat berkembang dalam KB.
temp_tablespaces	Dinamis	Mengatur tablespace yang akan digunakan untuk tabel sementara dan mengurutkan file.

Nama parameter	Apply_Type	Deskripsi
`timezone`	Dinamis	Mengatur zona waktu untuk menampilkan dan menginterpretasikan stempel waktu. Internet Assigned Numbers Authority (IANA) menerbitkan zona waktu baru di https://www.iana.org/time-zones beberapa kali dalam setahun. Setiap kali RDS mengeluarkan rilis pemeliharaan minor PostgreSQL yang baru, rilis tersebut akan dikirimkan beserta data zona waktu terbaru pada saat rilis. Jika menggunakan versi RDS for PostgreSQL terbaru, Anda akan memiliki data zona waktu terbaru dari RDS. Untuk memastikan bahwa instans DB Anda memiliki data zona waktu terbaru, sebaiknya tingkatkan ke versi mesin DB yang lebih baru. Anda tidak dapat mengubah tabel zona waktu dalam instans DB PostgreSQL secara manual. RDS tidak memodifikasi atau mengatur ulang data zona waktu dari instans DB yang berjalan. Data zona waktu baru diinstal hanya ketika Anda melakukan peningkatan versi mesin basis data.
`track_activities`	Dinamis	Mengumpulkan informasi cara menjalankan perintah.
`track_activity_query_size`	Statis	Mengatur ukuran terpesan untuk pg_stat_activity.current_query, dalam byte.
`track_counts`	Dinamis	Mengumpulkan statistik aktivitas basis data.
`track_functions`	Dinamis	Mengumpulkan statistik tingkat fungsi aktivitas basis data.
`track_io_timing`	Dinamis	Mengumpulkan statistik waktu aktivitas I/O basis data.

Nama parameter	Apply_Type	Deskripsi
transaction_deferrable	Dinamis	Menunjukkan apakah akan menunda transaksi hanya-baca berseri sampai bisa dimulai tanpa ada kegagalan serialisasi.
transaction_isolation	Dinamis	Menetapkan tingkat isolasi transaksi saat ini.
transaction_read_only	Dinamis	Mengatur status hanya-baca transaksi saat ini.
transform_null_equals	Dinamis	Memperlakukan expr=NULL sebagai expr IS NULL.
update_process_title	Dinamis	Memperbarui judul proses untuk menampilkan perintah SQL yang aktif.
vacuum_cost_delay	Dinamis	Penundaan biaya vakum dalam milidetik.
vacuum_cost_limit	Dinamis	Jumlah biaya vakum yang tersedia sebelum napping.
vacuum_cost_page_dirty	Dinamis	Biaya vakum untuk halaman yang kotor karena vakum.
vacuum_cost_page_hit	Dinamis	Biaya vakum untuk halaman yang ditemukan di cache buffer.
vacuum_cost_page_miss	Dinamis	Biaya vakum untuk halaman yang tidak ditemukan dalam cache buffer.
vacuum_defer_cleanup_age	Dinamis	Jumlah transaksi yang harus ditangguhkan dengan vakum dan hot cleanup, jika ada.
vacuum_freeze_min_age	Dinamis	Usia minimum saat vakum harus membekukan baris tabel.

Nama parameter	Apply_Type	Deskripsi
vacuum_freeze_table_age	Dinamis	Usia saat vakum harus memindai seluruh tabel untuk membekukan tuple.
wal_buffers	Statis	Mengatur jumlah buffer halaman disk dalam memori bersama untuk WAL.
wal_writer_delay	Dinamis	Waktu tidur penulis WAL antara beberapa pengosongan WAL.
work_mem	Dinamis	Mengatur memori maksimum yang akan digunakan untuk ruang kerja kueri.
xmlbinary	Dinamis	Menetapkan cara pengenkodean nilai biner dalam XML.
xmloption	Dinamis	Menetapkan apakah data XML dalam operasi penguraian implisit dan serialisasi dianggap sebagai dokumen atau fragmen konten.

Amazon RDS menggunakan unit PostgreSQL default untuk semua parameter. Tabel berikut menunjukkan unit default PostgreSQL untuk setiap parameter.

Nama parameter	Unit
archive_timeout	detik
authentication_timeout	detik
autovacuum_naptime	detik
autovacuum_vacuum_cost_delay	milidetik
bgwriter_delay	milidetik
checkpoint_timeout	detik

Nama parameter	Unit
checkpoint_warning	detik
deadlock_timeout	milidetik
effective_cache_size	8 KB
lock_timeout	milidetik
log_autovacuum_min_duration	milidetik
log_min_duration_statement	milidetik
log_rotation_age	menit
log_rotation_size	KB
log_temp_files	KB
maintenance_work_mem	KB
max_stack_depth	KB
max_standby_archive_delay	milidetik
max_standby_streaming_delay	milidetik
post_auth_delay	detik
pre_auth_delay	detik
segment_size	8 KB
shared_buffers	8 KB
statement_timeout	milidetik
ssl_renegotiation_limit	KB
tcp_keepalives_idle	detik

Nama parameter	Unit
tcp_keepalives_interval	detik
temp_file_limit	KB
work_mem	KB
temp_buffers	8 KB
vacuum_cost_delay	milidetik
wal_buffers	8 KB
wal_receiver_timeout	milidetik
wal_segment_size	B
wal_sender_timeout	milidetik
wal_writer_delay	milidetik
wal_receiver_status_interval	detik

Menyetel dengan peristiwa tunggu di RDS for PostgreSQL

Peristiwa tunggu adalah alat penyetelan penting untuk RDS for PostgreSQL. Ketika Anda dapat mengetahui mengapa sesi menunggu sumber daya dan apa yang sedang dilakukan, Anda lebih mampu mengurangi kemacetan. Anda dapat menggunakan informasi di bagian ini untuk menemukan kemungkinan penyebab dan tindakan korektif. Bagian ini juga membahas konsep penyetelan dasar PostgreSQL.

Peristiwa tunggu di bagian ini spesifik untuk RDS for PostgreSQL.

Topik

Konsep penting dalam penyetelan RDS for PostgreSQL

Sebelum Anda menyetel basis data RDS for PostgreSQL, pastikan untuk mempelajari apa itu peristiwa tunggu dan mengapa peristiwa tersebut terjadi. Baca juga memori dasar dan arsitektur disk RDS for PostgreSQL. Anda dapat melihat diagram arsitektur yang berguna di wikibook PostgreSQL.

Topik

Peristiwa tunggu RDS for PostgreSQL

Peristiwa tunggu menunjukkan bahwa sesi sedang menunggu sumber daya. Misalnya, peristiwa tunggu `Client:ClientRead` terjadi ketika RDS for PostgreSQL menunggu untuk menerima data dari klien. Sesi biasanya menunggu sumber daya seperti berikut ini.

- Akses thread tunggal ke buffer, misalnya, saat sesi mencoba memodifikasi buffer
- Baris yang saat ini dikunci oleh sesi lain
- Pembacaan file data
- Penulisan file log

Misalnya, untuk memenuhi kueri, sesi mungkin melakukan pemindaian tabel lengkap. Jika data belum ada dalam memori, sesi akan menunggu I/O disk selesai. Ketika buffer dibaca ke dalam memori, sesi mungkin perlu menunggu karena sesi lain mengakses buffer yang sama. Basis data mencatat peristiwa tunggu dengan menggunakan peristiwa tunggu standar. Peristiwa tersebut dikelompokkan ke dalam kategori.

Dengan sendirinya, satu peristiwa tunggu tidak menunjukkan masalah performa. Misalnya, jika data yang diminta tidak ada dalam memori, data perlu dibaca dari disk. Jika satu sesi mengunci baris untuk pembaruan, sesi lain akan menunggu baris tersebut dibuka sehingga dapat memperbaruinya. Komit perlu menunggu penulisan ke file log selesai. Peristiwa tunggu merupakan bagian integral dari fungsi normal basis data.

Di sisi lain, sejumlah besar peristiwa tunggu biasanya menunjukkan masalah performa. Dalam kasus seperti itu, Anda dapat menggunakan data peristiwa tunggu untuk menentukan tempat sesi

menghabiskan waktu. Misalnya, jika laporan yang biasanya berjalan dalam hitungan menit sekarang berjalan selama berjam-jam, Anda dapat mengidentifikasi peristiwa tunggu yang berkontribusi paling besar terhadap total waktu tunggu. Jika Anda dapat menentukan penyebab peristiwa tunggu teratas, terkadang Anda dapat membuat perubahan yang meningkatkan performa. Misalnya, jika sesi Anda menunggu baris yang telah dikunci oleh sesi lain, Anda dapat mengakhiri sesi penguncian ini.

Memori RDS for PostgreSQL

Memori RDS for PostgreSQL dibagi menjadi bersama dan lokal.

Topik

- Memori bersama dalam RDS for PostgreSQL
- Memori lokal dalam RDS for PostgreSQL

Memori bersama dalam RDS for PostgreSQL

RDS for PostgreSQL mengalokasikan memori bersama saat instans dimulai. Memori bersama dibagi menjadi beberapa subarea. Anda dapat menemukan deskripsi untuk yang paling penting berikut ini.

Topik

- Buffer bersama
- Buffer log write ahead (WAL)

Buffer bersama

Kumpulan buffer bersama adalah area memori RDS for PostgreSQL yang menampung semua halaman yang sedang atau telah digunakan oleh koneksi aplikasi. Halaman adalah versi memori dari blok disk. Kumpulan buffer bersama menyimpan blok data yang dibaca dari disk. Kumpulan tersebut mengurangi kebutuhan untuk membaca ulang data dari disk, sehingga membuat basis data beroperasi lebih efisien.

Setiap tabel dan indeks disimpan sebagai susunan halaman dengan ukuran tetap. Setiap blok berisi beberapa tuple, yang sesuai dengan baris. Tuple dapat disimpan di halaman mana pun.

Kumpulan buffer bersama memiliki memori terbatas. Jika permintaan baru memerlukan halaman yang tidak ada dalam memori, dan memori sudah tidak ada lagi, RDS for PostgreSQL mengosongkan halaman yang jarang digunakan untuk mengakomodasi permintaan tersebut. Kebijakan pengosongan diimplementasikan oleh algoritma clock sweep.

Parameter `shared_buffers` menentukan berapa banyak memori server dikhususkan untuk menyimpan data dalam cache.

Buffer log write ahead (WAL)

Buffer log write-ahead (WAL) menyimpan data transaksi yang kemudian ditulis RDS for PostgreSQL ke penyimpanan persisten. Menggunakan mekanisme WAL, RDS for PostgreSQL dapat melakukan hal berikut:

- Memulihkan data setelah kegagalan
- Mengurangi disk I/O dengan menghindari penulisan ke disk yang sering

Ketika klien mengubah data, RDS for PostgreSQL menulis perubahan ke buffer WAL. Ketika klien mengeluarkan `COMMIT`, proses penulis WAL menulis data transaksi ke file WAL.

Parameter `wal_level` menentukan berapa banyak informasi yang ditulis ke WAL.

Memori lokal dalam RDS for PostgreSQL

Setiap proses backend mengalokasikan memori lokal untuk pemrosesan kueri.

Topik

- Area memori kerja
- Area memori kerja pemeliharaan
- Area buffer sementara

Area memori kerja

Area memori kerja menyimpan data sementara untuk kueri yang melakukan pengurutan dan hash. Misalnya, kueri dengan klausa `ORDER BY` melakukan pengurutan. Kueri menggunakan tabel hash dalam gabungan dan agregasi hash.

Parameter `work_mem` jumlah memori yang akan digunakan oleh operasi pengurutan internal dan tabel hash sebelum menulis ke file disk sementara. Nilai default-nya adalah 4 MB. Beberapa sesi dapat berjalan secara bersamaan, dan setiap sesi dapat menjalankan operasi pemeliharaan secara paralel. Karena alasan ini, total memori kerja yang digunakan dapat menjadi kelipatan dari pengaturan `work_mem`.

Area memori kerja pemeliharaan

Area memori kerja pemeliharaan menyimpan data untuk operasi pemeliharaan. Operasi ini termasuk memvakum, membuat indeks, dan menambahkan kunci asing.

Parameter `maintenance_work_mem` menentukan jumlah maksimum memori yang akan digunakan oleh operasi pemeliharaan. Nilai default-nya adalah 64 MB. Sebuah sesi basis data hanya dapat menjalankan satu operasi pemeliharaan dalam satu waktu.

Area buffer sementara

Area buffer sementara menyimpan tabel sementara untuk setiap sesi basis data.

Setiap sesi mengalokasikan buffer sementara sesuai kebutuhan hingga batas yang Anda tentukan. Saat sesi berakhir, server menghapus buffer.

Parameter `temp_buffers` mengatur jumlah maksimum buffer sementara yang digunakan oleh setiap sesi. Sebelum penggunaan pertama tabel sementara dalam sesi, Anda dapat mengubah nilai `temp_buffers`.

Proses RDS for PostgreSQL

RDS for PostgreSQL menggunakan beberapa proses.

Topik

- Proses postmaster
- Proses backend
- Proses latar belakang

Proses postmaster

Proses postmaster adalah proses pertama yang dimulai ketika Anda memulai RDS for PostgreSQL. Proses postmaster memiliki tanggung jawab utama berikut:

- Membagi dan memantau proses latar belakang
- Menerima permintaan autentikasi dari proses klien, dan mengautentikasi mereka sebelum mengizinkan basis data untuk melayani permintaan

Proses backend

Jika postmaster mengautentikasi permintaan klien, postmaster akan melakukan proses backend baru, juga disebut proses postgres. Satu proses klien terhubung ke persis satu proses backend. Proses klien dan proses backend berkomunikasi secara langsung tanpa intervensi oleh proses postmaster.

Proses latar belakang

Proses postmaster membagi beberapa proses yang melakukan tugas backend yang berbeda. Beberapa hal yang lebih penting termasuk berikut ini:

- Penulis WAL

  RDS for PostgreSQL menulis data dalam buffer WAL (log write ahead) ke file log. Prinsip log write ahead adalah bahwa basis data tidak dapat menulis perubahan pada file data sampai setelah basis data menulis catatan log yang menjelaskan perubahan tersebut ke disk. Mekanisme WAL mengurangi disk I/O, dan memungkinkan RDS for PostgreSQL untuk menggunakan log untuk memulihkan basis data setelah kegagalan.
- Penulis latar belakang

  Proses ini secara berkala menulis halaman kotor (diubah) dari buffer memori ke file data. Sebuah halaman menjadi kotor ketika proses backend memodifikasinya dalam memori.
- Daemon autovacuum

  Daemon terdiri dari hal-hal berikut:

  - Peluncur autovacuum
  - Proses pekerja autovacuum

  Saat autovacuum diaktifkan, autovacuum tersebut memeriksa tabel yang memiliki banyak tuple yang dimasukkan, diperbarui, atau dihapus. Daemon memiliki tanggung jawab sebagai berikut:

  - Memulihkan atau menggunakan kembali ruang disk yang ditempati oleh baris yang diperbarui atau dihapus
  - Memperbarui statistik yang digunakan oleh perencana
  - Melindungi dari kehilangan data lama karena wraparound ID transaksi

  Fitur autovacuum mengotomatiskan eksekusi `VACUUM` dan perintah `ANALYZE`. `VACUUM` memiliki varian berikut: standar dan penuh. Vakum standar berjalan secara paralel dengan operasi basis

data lainnya. `VACUUM FULL` membutuhkan kunci eksklusif di tabel yang sedang dikerjakannya. Dengan demikian, tidak dapat berjalan secara paralel dengan operasi yang mengakses tabel yang sama. `VACUUM` membuat sejumlah besar lalu lintas I/O, yang dapat menyebabkan kinerja buruk untuk sesi aktif lainnya.

Peristiwa tunggu RDS for PostgreSQL

Tabel berikut mencantumkan peristiwa tunggu untuk RDS for PostgreSQL yang paling sering menunjukkan masalah performa, dan meringkas penyebab paling umum dan tindakan korektifnya.

Peristiwa tunggu	Definisi
Client:ClientRead	Peristiwa ini terjadi ketika RDS for PostgreSQL menunggu untuk menerima data dari klien.
Client:ClientWrite	Peristiwa ini terjadi ketika RDS for PostgreSQL menunggu untuk menulis data ke klien.
CPU	Peristiwa ini terjadi saat thread aktif di CPU atau sedang menunggu CPU.
IO:BufFileRead dan IO:BufFileWrite	Peristiwa ini terjadi ketika RDS for PostgreSQL membuat file sementara.
IO: DataFileRead	Peristiwa ini terjadi ketika koneksi menunggu pada proses backend untuk membaca halaman yang diperlukan dari penyimpanan karena halaman tidak tersedia dalam memori bersama.
IO:WALWrite	Peristiwa ini terjadi ketika RDS for PostgreSQL sedang menunggu buffer write-ahead log (WAL) ditulis ke file WAL.
Lock:advisory	Peristiwa ini terjadi ketika aplikasi PostgreSQL menggunakan kunci untuk mengoordinasikan aktivitas di beberapa sesi.

Peristiwa tunggu	Definisi
Lock:extend	Peristiwa ini terjadi ketika proses backend menunggu untuk mengunci relasi untuk memperpanjangnya selagi proses lain memiliki kunci pada relasi itu untuk tujuan yang sama.
Lock:Relation	Peristiwa ini terjadi ketika kueri menunggu untuk memperoleh kunci pada tabel atau tampilan yang saat ini dikunci oleh transaksi lain.
Lock:transactionid	Peristiwa ini terjadi ketika transaksi sedang menunggu kunci tingkat baris.
Lock:tuple	Peristiwa ini terjadi ketika proses backend menunggu untuk mendapatkan kunci pada tuple.
LWLock:BufferMapping (LWLock:buffer_mapping)	Peristiwa ini terjadi saat sesi menunggu untuk mengaitkan blok data dengan buffer di pool buffer bersama.
LWLock:BufferIO (IPC:BufferIO)	Peristiwa ini terjadi ketika RDS for PostgreSQL menunggu proses lain untuk menyelesaikan operasi input/output (I/O)-nya ketika secara konkuren mencoba mengakses halaman.
LWLock:buffer_content (BufferContent)	Peristiwa ini terjadi ketika suatu sesi menunggu untuk membaca atau menulis halaman data di memori selagi sesi lain mengunci halaman tersebut untuk penulisan.
LWLock:lock_manager (LWLock:lockmanager)	Peristiwa ini terjadi saat mesin RDS for PostgreSQL mempertahankan area memori kunci bersama untuk mengalokasikan, memeriksa, dan mendealokasikan kunci saat kunci jalur cepat tidak memungkinkan.

Peristiwa tunggu	Definisi
Timeout:PgSleep	Peristiwa ini terjadi ketika proses server telah memanggil fungsi `pg_sleep` dan menunggu batas waktu tidur berakhir.
Timeout:VacuumDelay	Peristiwa ini menunjukkan bahwa proses vakum sedang tidur karena perkiraan batas biaya telah tercapai.

Client:ClientRead

Peristiwa `Client:ClientRead` terjadi ketika RDS for PostgreSQL menunggu untuk menerima data dari klien.

Topik

- Versi mesin yang didukung
- Konteks
- Kemungkinan penyebab peningkatan peristiwa tunggu
- Tindakan

Versi mesin yang didukung

Informasi peristiwa tunggu ini didukung untuk RDS for PostgreSQL versi 10 dan yang lebih tinggi.

Konteks

Instans DB RDS for PostgreSQL menunggu untuk menerima data dari klien. Instans DB RDS for PostgreSQL harus menerima data dari klien sebelum dapat mengirim lebih banyak data ke klien. Waktu instans menunggu sebelum menerima data dari klien adalah sebuah peristiwa `Client:ClientRead`.

Kemungkinan penyebab peningkatan peristiwa tunggu

Penyebab umum peristiwa `Client:ClientRead` muncul dalam peristiwa tunggu teratas mencakup yang berikut:

Peningkatan latensi jaringan

Mungkin terdapat peningkatan latensi jaringan antara instans DB RDS for PostgreSQL dan klien. Latensi jaringan yang lebih tinggi meningkatkan waktu yang dibutuhkan instans DB untuk menerima data dari klien.

Peningkatan beban pada klien

Mungkin terdapat tekanan CPU atau saturasi jaringan pada klien. Peningkatan beban pada klien dapat menunda transmisi data dari klien ke instans DB RDS for PostgreSQL.

Round trip jaringan yang berlebihan

Sejumlah besar round trip jaringan antara instans DB RDS for PostgreSQL dan klien dapat menunda transmisi data dari klien ke instans DB RDS for PostgreSQL.

Operasi penyalinan besar

Selama operasi penyalinan, data ditransfer dari sistem file klien ke instans DB RDS for PostgreSQL. Mengirim sejumlah besar data ke instans DB dapat menunda transmisi data dari klien ke instans DB.

Koneksi klien idle

Ketika klien terhubung ke instans DB RDS for PostgreSQL dalam status `idle in transaction`, instans DB ini mungkin menunggu klien mengirim lebih banyak data atau memberikan perintah. Koneksi dalam keadaan ini dapat menyebabkan peningkatan peristiwa `Client:ClientRead`.

PgBouncer digunakan untuk pooling koneksi

PgBouncer memiliki pengaturan konfigurasi jaringan tingkat rendah yang disebut `pkt_buf`, yang diatur ke 4.096 secara default. Jika beban kerja mengirimkan paket kueri yang lebih besar dari 4.096 byte melalui PgBouncer, sebaiknya tingkatkan pengaturan `pkt_buf` ke 8.192. Jika pengaturan baru tidak mengurangi jumlah peristiwa `Client:ClientRead`, sebaiknya tingkatkan pengaturan `pkt_buf` ke nilai yang lebih besar, seperti 16.384 atau 32.768. Jika teks kueri berukuran besar, pengaturan yang lebih besar dapat sangat membantu.

Tindakan

Kami merekomendasikan berbagai tindakan, tergantung pada penyebab peristiwa tunggu Anda.

Topik

Tempatkan klien di Zona Ketersediaan dan subnet VPC yang sama dengan instans.

Untuk mengurangi latensi jaringan dan meningkatkan throughput jaringan, tempatkan klien di Zona Ketersediaan dan subnet cloud privat virtual (VPC) yang sama dengan instans DB RDS for PostgreSQL. Pastikan bahwa klien secara geografis sedekat mungkin dengan instans DB.

Skalakan klien Anda

Dengan menggunakan Amazon CloudWatch atau metrik host lainnya, ketahui apakah klien Anda saat ini dibatasi oleh CPU atau bandwidth jaringan, atau keduanya. Jika klien dibatasi, skalakan klien sesuai yang diperlukan.

Gunakan instans generasi saat ini

Dalam kasus tertentu, Anda mungkin tidak menggunakan kelas instans DB yang mendukung frame jumbo. Jika Anda menjalankan aplikasi di Amazon EC2, pertimbangkan untuk menggunakan instans generasi saat ini untuk klien. Selain itu, konfigurasikan unit transmisi maksimum (MTU) pada sistem operasi klien. Teknik ini dapat mengurangi jumlah round trip jaringan dan meningkatkan throughput jaringan. Untuk informasi selengkapnya, lihat Frame jumbo (9001 MTU) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Untuk informasi tentang kelas instans DB, lihat Kelas instans DB . Untuk menentukan kelas instans DB yang setara dengan jenis instans Amazon EC2, tempatkan `db.` sebelum nama jenis instans Amazon EC2. Misalnya, instans Amazon EC2 `r5.8xlarge` setara dengan kelas instans DB `db.r5.8xlarge`.

Tingkatkan bandwidth jaringan

Gunakan metrik `NetworkReceiveThroughput` dan `NetworkTransmitThroughput` Amazon CloudWatch untuk memantau lalu lintas jaringan masuk dan keluar pada instans DB. Metrik ini dapat membantu Anda menentukan apakah bandwidth jaringan cukup untuk beban kerja Anda.

Jika bandwidth jaringan Anda tidak cukup, tingkatkan. Jika klien AWS atau instans DB Anda mencapai batas bandwidth jaringan, satu-satunya cara untuk meningkatkan bandwidth adalah dengan meningkatkan ukuran instans DB Anda. Untuk informasi selengkapnya, lihat Jenis kelas instans DB.

Untuk informasi selengkapnya tentang metrik CloudWatch, lihat CloudWatch Metrik Amazon untuk Amazon RDS.

Pantau nilai maksimum untuk performa jaringan

Jika Anda menggunakan klien Amazon EC2, Amazon EC2 menyediakan nilai maksimum untuk metrik performa jaringan, termasuk bandwidth jaringan masuk dan keluar agregat. Amazon EC2 juga menyediakan pelacakan koneksi untuk memastikan paket dikembalikan sebagaimana diharapkan dan akses layanan tautan-lokal untuk layanan seperti Sistem Nama Domain (DNS). Untuk memantau nilai maksimum ini, gunakan driver jaringan yang ditingkatkan saat ini dan pantau performa jaringan untuk klien Anda.

Untuk informasi selengkapnya, lihat Memantau performa jaringan untuk instans Amazon EC2 dalam Panduan Pengguna Amazon EC2 untuk Instans Linux dan Memantau performa jaringan untuk instans Amazon EC2 dalam Panduan Pengguna Amazon EC2 untuk Instans Windows.

Pantau transaksi dalam status "idle dalam transaksi"

Periksa apakah Anda memiliki peningkatan jumlah koneksi `idle in transaction`. Untuk melakukannya, pantau kolom `state` dalam tabel `pg_stat_activity`. Anda mungkin dapat mengidentifikasi sumber koneksi dengan menjalankan kueri seperti yang berikut ini.

```
select client_addr, state, count(1) from pg_stat_activity
where state like 'idle in transaction%'
group by 1,2
order by 3 desc
```

Client:ClientWrite

Peristiwa `Client:ClientWrite` terjadi ketika RDS for PostgreSQL menunggu untuk menulis data ke klien.

Topik

- Versi mesin yang didukung

Versi mesin yang didukung

Informasi peristiwa tunggu ini didukung untuk RDS for PostgreSQL versi 10 dan yang lebih tinggi.

Konteks

Proses klien harus membaca semua data yang diterima dari klaster DB RDS for PostgreSQL sebelum klaster dapat mengirim lebih banyak data. Waktu klaster menunggu sebelum mengirim lebih banyak data ke klien adalah sebuah peristiwa `Client:ClientWrite`.

Throughput jaringan yang berkurang untuk instans DB RDS for PostgreSQL dan klien dapat menyebabkan peristiwa ini. Tekanan CPU dan saturasi jaringan pada klien juga dapat menyebabkan peristiwa ini. Tekanan CPU adalah ketika CPU sepenuhnya digunakan dan ada tugas yang menunggu waktu CPU. Saturasi jaringan adalah saat jaringan antara basis data dan klien membawa lebih banyak data dari yang dapat ditanganinya.

Kemungkinan penyebab peningkatan peristiwa tunggu

Penyebab umum peristiwa `Client:ClientWrite` muncul dalam peristiwa tunggu teratas mencakup yang berikut:

Peningkatan latensi jaringan

Mungkin terdapat peningkatan latensi jaringan antara instans DB RDS for PostgreSQL dan klien. Latensi jaringan yang lebih tinggi meningkatkan waktu yang dibutuhkan klien untuk menerima data.

Peningkatan beban pada klien

Mungkin terdapat tekanan CPU atau saturasi jaringan pada klien. Peningkatan beban pada klien menunda penerimaan data dari instans DB RDS for PostgreSQL.

Data dalam volume besar yang dikirim ke klien

Instans DB RDS for PostgreSQL mungkin mengirimkan sejumlah besar data ke klien. Klien mungkin tidak dapat menerima data secepat klaster mengirimkannya. Aktivitas seperti penyalinan tabel besar dapat mengakibatkan peningkatan peristiwa `Client:ClientWrite`.

Tindakan

Kami merekomendasikan berbagai tindakan, tergantung pada penyebab peristiwa tunggu Anda.

Topik

- Tempatkan klien di Zona Ketersediaan dan subnet VPC yang sama dengan klaster
- Gunakan instans generasi saat ini
- Kurangi jumlah data yang dikirim ke klien Anda
- Skalakan klien Anda

Tempatkan klien di Zona Ketersediaan dan subnet VPC yang sama dengan klaster

Untuk mengurangi latensi jaringan dan meningkatkan throughput jaringan, tempatkan klien di Zona Ketersediaan dan subnet cloud privat virtual (VPC) yang sama dengan instans DB RDS for PostgreSQL.

Gunakan instans generasi saat ini

Dalam kasus tertentu, Anda mungkin tidak menggunakan kelas instans DB yang mendukung frame jumbo. Jika Anda menjalankan aplikasi di Amazon EC2, pertimbangkan untuk menggunakan instans generasi saat ini untuk klien. Selain itu, konfigurasikan unit transmisi maksimum (MTU) pada sistem operasi klien. Teknik ini dapat mengurangi jumlah round trip jaringan dan meningkatkan throughput jaringan. Untuk informasi selengkapnya, lihat Frame jumbo (9001 MTU) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Untuk informasi tentang kelas instans DB, lihat Kelas instans DB . Untuk menentukan kelas instans DB yang setara dengan jenis instans Amazon EC2, tempatkan `db.` sebelum nama jenis instans Amazon EC2. Misalnya, instans Amazon EC2 `r5.8xlarge` setara dengan kelas instans DB `db.r5.8xlarge`.

Kurangi jumlah data yang dikirim ke klien Anda

Jika memungkinkan, sesuaikan aplikasi Anda untuk mengurangi jumlah data yang dikirim instans DB RDS for PostgreSQL ke klien. Penyesuaian ini akan mengurangi pertentangan CPU dan jaringan pada klien.

Skalakan klien Anda

Dengan menggunakan Amazon CloudWatch atau metrik host lainnya, ketahui apakah klien Anda saat ini dibatasi oleh CPU atau bandwidth jaringan, atau keduanya. Jika klien dibatasi, skalakan klien sesuai yang diperlukan.

CPU

Peristiwa ini terjadi saat thread aktif di CPU atau sedang menunggu CPU.

Topik

- Versi mesin yang didukung
- Konteks
- Kemungkinan penyebab peningkatan peristiwa tunggu
- Tindakan

Versi mesin yang didukung

Informasi peristiwa tunggu ini relevan untuk semua versi RDS for PostgreSQL.

Konteks

Central Processing Unit (CPU) adalah komponen komputer yang menjalankan instruksi. Misalnya, instruksi CPU melakukan operasi aritmetika dan bertukar data dalam memori. Jika kueri meningkatkan jumlah instruksi yang dilakukannya melalui mesin basis data, waktu yang dihabiskan untuk menjalankan kueri akan meningkat. Penjadwalan CPU memberikan waktu CPU untuk suatu proses. Penjadwalan diatur oleh kernel sistem operasi.

Topik

- Bagaimana cara mengetahui kapan peristiwa tunggu ini terjadi?
- Metrik DBLoadCPU
- Metrik os.cpuUtilization
- Kemungkinan penyebab penjadwalan CPU

Bagaimana cara mengetahui kapan peristiwa tunggu ini terjadi?

Peristiwa tunggu CPU ini menunjukkan bahwa proses backend aktif di CPU atau sedang menunggu CPU. Anda mengetahuinya terjadi saat kueri menunjukkan informasi berikut:

- Kolom pg_stat_activity.state memiliki nilai active.
- Kolom wait_event_type dan wait_event di pg_stat_activity adalah null.

Untuk melihat proses backend yang menggunakan atau menunggu CPU, jalankan kueri berikut.

```
SELECT *
FROM   pg_stat_activity
WHERE  state = 'active'
AND    wait_event_type IS NULL
AND    wait_event IS NULL;
```

Metrik DBLoadCPU

Metrik Wawasan Performa untuk CPU adalah DBLoadCPU. Nilai untuk DBLoadCPU dapat berbeda dengan nilai untuk metrik Amazon CloudWatch CPUUtilization. Metrik CloudWatch dikumpulkan dari HyperVisor untuk instans basis data.

Metrik os.cpuUtilization

Metrik sistem operasi Wawasan Performa memberikan informasi terperinci tentang pemanfaatan CPU. Misalnya, Anda dapat menampilkan metrik berikut:

- os.cpuUtilization.nice.avg
- os.cpuUtilization.total.avg
- os.cpuUtilization.wait.avg
- os.cpuUtilization.idle.avg

Wawasan Performa melaporkan penggunaan CPU oleh mesin basis data sebagai os.cpuUtilization.nice.avg.

Kemungkinan penyebab penjadwalan CPU

Kernel sistem operasi (OS) menangani penjadwalan untuk CPU. Ketika CPU aktif, sebuah proses mungkin perlu menunggu untuk dijadwalkan. CPU aktif saat melakukan penghitungan. CPU ini juga

aktif saat memiliki thread idle yang tidak berjalan, yaitu thread idle yang menunggu I/O memori. Jenis I/O ini mendominasi beban kerja basis data yang biasa.

Proses cenderung menunggu untuk dijadwalkan pada CPU saat kondisi berikut terpenuhi:

- Metrik `CPUUtilization` CloudWatch mendekati 100%.
- Beban rata-rata lebih besar dari jumlah vCPU, yang menunjukkan beban berat. Anda dapat menemukan metrik `loadAverageMinute` di bagian metrik OS dalam Wawasan Performa.

Kemungkinan penyebab peningkatan peristiwa tunggu

Saat peristiwa tunggu CPU terjadi lebih dari biasanya, yang mungkin menunjukkan adanya masalah performa, berikut adalah penyebab umumnya:

Topik

Kemungkinan penyebab lonjakan mendadak

Penyebab lonjakan mendadak yang paling memungkinkan adalah sebagai berikut:

- Aplikasi Anda membuka terlalu banyak koneksi bersamaan ke basis data. Skenario ini dikenal sebagai "connection storm".
- Beban kerja aplikasi Anda berubah dengan salah satu cara berikut:
  - Kueri baru
  - Peningkatan ukuran set data
  - Pemeliharaan atau pembuatan indeks
  - Fungsi baru
  - Operator baru
  - Peningkatan eksekusi kueri paralel
- Rencana eksekusi kueri Anda telah berubah. Dalam beberapa kasus, perubahan dapat menyebabkan peningkatan buffer. Misalnya, kueri sekarang menggunakan pemindaian berurutan saat sebelumnya menggunakan indeks. Dalam hal ini, kueri membutuhkan lebih banyak CPU untuk mencapai tujuan yang sama.

Kemungkinan penyebab frekuensi tinggi jangka panjang

Berikut adalah penyebab paling memungkinkan dari peristiwa yang berulang dalam jangka waktu lama:

- Terlalu banyak proses backend yang berjalan secara konkuren pada CPU. Proses-proses ini dapat berupa pekerja paralel.
- Kueri beperforma suboptimal karena membutuhkan buffer dalam jumlah besar.

Corner cases

Jika tidak ada kemungkinan penyebab yang merupakan penyebab sebenarnya, situasi berikut mungkin terjadi:

- CPU menukar proses masuk dan keluar.
- CPU mungkin mengelola entri tabel halaman jika fitur huge page telah dinonaktifkan. Fitur manajemen memori ini diaktifkan secara default untuk semua kelas instans DB selain kelas instans DB mikro, kecil, dan menengah. Untuk informasi selengkapnya, lihat Halaman besar untuk RDS for PostgreSQL .

Tindakan

Jika peristiwa tunggu CPU mendominasi aktivitas basis data, hal tersebut tidak selalu menunjukkan adanya masalah performa. Tanggapi peristiwa ini hanya saat performa menurun.

Topik

- Selidiki apakah basis data menyebabkan peningkatan CPU
- Tentukan apakah jumlah koneksi meningkat
- Tanggapi perubahan beban kerja

Selidiki apakah basis data menyebabkan peningkatan CPU

Periksa metrik `os.cpuUtilization.nice.avg` dalam Wawasan Performa. Jika nilai ini jauh lebih kecil daripada penggunaan CPU, proses non-basis data adalah kontributor utama ke CPU.

Tentukan apakah jumlah koneksi meningkat

Periksa metrik `DatabaseConnections` di Amazon CloudWatch. Tindakan Anda bergantung pada apakah jumlahnya meningkat atau menurun selama periode peningkatan peristiwa tunggu CPU.

Koneksi meningkat

Jika jumlah koneksi meningkat, bandingkan jumlah proses backend yang mengonsumsi CPU terhadap jumlah vCPU. Skenario berikut mungkin terjadi:

- Jumlah proses backend yang mengonsumsi CPU lebih kecil dari jumlah vCPU.

  Dalam hal ini, jumlah koneksi tidak menjadi masalah. Namun, Anda masih dapat mencoba mengurangi pemanfaatan CPU.
- Jumlah proses backend yang mengonsumsi CPU lebih besar dari jumlah vCPU.

  Jika demikian, pertimbangkan opsi berikut:
  - Kurangi jumlah proses backend yang terhubung ke basis data Anda. Misalnya, terapkan solusi pooling koneksi seperti Proksi RDS. Untuk mempelajari selengkapnya, lihat Menggunakan Proksi Amazon RDS.
  - Upgrade ukuran instans Anda untuk mendapatkan jumlah vCPU yang lebih tinggi.
  - Jika berlaku, arahkan ulang beberapa beban kerja hanya-baca ke simpul pembaca.

Koneksi tidak meningkat

Periksa metrik `blks_hit` dalam Wawasan Performa. Cari korelasi antara peningkatan `blks_hit` dan penggunaan CPU. Skenario berikut mungkin terjadi:

- Penggunaan CPU dan `blks_hit` berkorelasi.

  Dalam hal ini, temukan pernyataan SQL teratas yang terkait dengan penggunaan CPU, lalu cari perubahan rencana. Anda dapat menggunakan salah satu teknik berikut:
  - Jelaskan rencana secara manual, lalu bandingkan dengan rencana eksekusi yang diperkirakan.
  - Cari peningkatan hit blok per detik dan hit blok lokal per detik. Di bagian SQL Teratas pada dasbor Wawasan Performa, pilih Preferensi.
- Penggunaan CPU dan `blks_hit` tidak berkorelasi.

  Jika demikian, ketahui apakah salah satu hal berikut terjadi:

- Aplikasi dengan cepat terhubung ke dan terputus dari basis data.

  Jalankan diagnosis perilaku ini dengan mengaktifkan `log_connections` dan `log_disconnections`, lalu menganalisis log PostgreSQL. Pertimbangkan untuk menggunakan penganalisis log `pgbadger`. Untuk informasi selengkapnya, lihat https://github.com/darold/pgbadger.

- OS kelebihan beban.

  Dalam hal ini, Wawasan Performa menunjukkan bahwa proses backend menggunakan CPU untuk waktu yang lebih lama dari biasanya. Cari buktinya di metrik `os.cpuUtilization` Wawasan Performa atau metrik `CPUUtilization` CloudWatch. Jika sistem operasi kelebihan beban, lihat metrik Pemantauan yang Ditingkatkan untuk mendiagnosis lebih lanjut. Secara khusus, lihat daftar proses dan persentase CPU yang dikonsumsi oleh setiap proses.

- Pernyataan SQL teratas mengonsumsi terlalu banyak CPU.

  Periksa pernyataan yang terkait dengan penggunaan CPU untuk melihat apakah pernyataan tersebut dapat menggunakan lebih sedikit CPU. Jalankan perintah `EXPLAIN`, lalu fokus pada simpul rencana yang memiliki dampak terbesar. Pertimbangkan untuk menggunakan pemvisualisasi rencana eksekusi PostgreSQL. Untuk mencoba alat ini, lihat http://explain.dalibo.com/.

Tanggapi perubahan beban kerja

Jika beban kerja Anda telah berubah, cari jenis perubahan berikut:

Kueri baru

Periksa apakah kueri baru memang diharapkan. Jika demikian, pastikan bahwa rencana eksekusinya dan jumlah eksekusi per detik memang diharapkan.

Peningkatan ukuran set data

Ketahui apakah pemartisiab, jika belum diterapkan, dapat membantu. Strategi ini dapat mengurangi jumlah halaman yang perlu diambil kueri.

Pemeliharaan atau pembuatan indeks

Periksa apakah jadwal pemeliharaan memang diharapkan. Praktik terbaiknya adalah menjadwalkan aktivitas pemeliharaan di luar aktivitas puncak.

Fungsi baru

Periksa apakah fungsi-fungsi ini berfungsi seperti yang diharapkan selama pengujian. Secara khusus, periksa apakah jumlah eksekusi per detik memang diharapkan.

Operator baru

Periksa apakah operator baru berfungsi seperti yang diharapkan selama pengujian.

Peningkatan dalam menjalankan kueri paralel

Ketahui apakah salah satu situasi berikut telah terjadi:

- Relasi atau indeks yang terkait tiba-tiba bertambah ukurannya sehingga sangat berbeda dari `min_parallel_table_scan_size` atau `min_parallel_index_scan_size`.
- Perubahan terkini telah dilakukan pada `parallel_setup_cost` atau `parallel_tuple_cost`.
- Perubahan terkini telah dilakukan pada `max_parallel_workers` atau `max_parallel_workers_per_gather`.

IO:BufFileRead dan IO:BufFileWrite

Peristiwa `IO:BufFileRead` dan `IO:BufFileWrite` terjadi ketika RDS for PostgreSQL membuat file sementara. Saat operasi membutuhkan lebih banyak memori daripada yang saat ini ditentukan oleh parameter memori kerja, operasi ini akan menulis data sementara ke penyimpanan persisten. Operasi ini terkadang disebut "spilling to disk".

Topik

Versi mesin yang didukung

Informasi peristiwa tunggu ini didukung untuk semua versi RDS for PostgreSQL.

Konteks

`IO:BufFileRead` dan `IO:BufFileWrite` berkaitan dengan area memori kerja dan area memori kerja pemeliharaan. Untuk informasi selengkapnya tentang penyetelan memori, lihat Resource Consumption dalam dokumentasi PostgreSQL.

Nilai default untuk `work_mem` adalah 4 MB. Jika satu sesi melakukan operasi secara paralel, setiap pekerja yang menangani paralelisme ini akan menggunakan memori 4 MB. Untuk alasan ini, atur `work_mem` dengan hati-hati. Jika Anda meningkatkan nilai ini terlalu banyak, basis data yang menjalankan banyak sesi mungkin akan mengonsumsi terlalu banyak memori. Jika Anda menetapkan nilai terlalu rendah, Aurora PostgreSQL akan membuat file sementara di penyimpanan lokal. I/O disk untuk file sementara ini dapat mengurangi performa.

Jika Anda mengamati urutan peristiwa berikut, basis data mungkin menghasilkan file sementara:

1. Penurunan ketersediaan secara tiba-tiba dan drastis
2. Pemulihan cepat untuk ruang kosong

Anda mungkin juga melihat pola "gergaji". Pola ini dapat menunjukkan bahwa basis data Anda membuat file kecil terus-menerus.

Kemungkinan penyebab peningkatan peristiwa tunggu

Secara umum, peristiwa tunggu ini disebabkan oleh operasi yang mengonsumsi lebih banyak memori daripada yang dialokasikan oleh parameter `work_mem` atau `maintenance_work_mem`. Untuk mengompensasi, operasi menulis ke file sementara. Penyebab umum peristiwa `IO:BufFileRead` dan `IO:BufFileWrite` mencakup hal berikut:

Kueri yang membutuhkan lebih banyak memori daripada yang ada di area memori kerja

Kueri dengan karakteristik berikut menggunakan area memori kerja:

- Hash join
- Klausa `ORDER BY`
- Klausa `GROUP BY`
- `DISTINCT`
- Fungsi jendela
- `CREATE TABLE AS SELECT`

- Penyegaran tampilan terwujud

Pernyataan yang membutuhkan lebih banyak memori daripada yang ada di area memori kerja pemeliharaan

Pernyataan berikut menggunakan area memori kerja pemeliharaan:

- `CREATE INDEX`
- `CLUSTER`

Tindakan

Kami merekomendasikan berbagai tindakan, tergantung pada penyebab peristiwa tunggu Anda.

Topik

- Identifikasi masalah
- Periksa kueri join
- Periksa kueri ORDER BY dan GROUP BY
- Hindari menggunakan operasi DISTINCT
- Mempertimbangkan untuk menggunakan fungsi jendela alih-alih fungsi GROUP BY
- Selidiki tampilan terwujud dan pernyataan CTAS
- Gunakan pg_repack saat Anda membuat kembali indeks
- Tingkatkan maintenance_work_mem saat Anda membuat klaster tabel
- Setel memori untuk mencegah IO:BufFileRead dan IO:BufFileWrite

Identifikasi masalah

Misalkan ada situasi saat Wawasan Performa tidak diaktifkan dan Anda menduga bahwa `IO:BufFileRead` dan `IO:BufFileWrite` terjadi lebih sering daripada biasanya. Untuk mengidentifikasi sumber masalahnya, Anda dapat mengatur parameter `log_temp_files` untuk mencatat log semua kueri yang menghasilkan file sementara lebih dari ambang batas KB yang Anda tentukan. Secara default, `log_temp_files` diatur ke `-1`, yang menonaktifkan fitur logging ini. Jika Anda mengatur parameter ini ke `0`, RDS for PostgreSQL mencatat log semua file sementara. Jika nilainya `1024`, Aurora PostgreSQL mencatat semua kueri yang menghasilkan file sementara yang berukuran lebih besar dari 1 MB. Untuk informasi selengkapnya tentang `log_temp_files`, lihat Error Reporting and Logging dalam dokumentasi PostgreSQL.

Periksa kueri join

Kemungkinan kueri Anda menggunakan join. Misalnya, kueri berikut menggabungkan empat tabel.

```
SELECT *
       FROM "order"
 INNER JOIN order_item
       ON (order.id = order_item.order_id)
 INNER JOIN customer
       ON (customer.id = order.customer_id)
 INNER JOIN customer_address
       ON (customer_address.customer_id = customer.id AND
           order.customer_address_id = customer_address.id)
 WHERE customer.id = 1234567890;
```

Kemungkinan penyebab lonjakan penggunaan file sementara adalah masalah dalam kueri itu sendiri. Misalnya, klausa yang rusak mungkin tidak memfilter join dengan benar. Pertimbangkan inner join kedua dalam contoh berikut.

```
SELECT *
       FROM "order"
 INNER JOIN order_item
       ON (order.id = order_item.order_id)
 INNER JOIN customer
       ON (customer.id = customer.id)
 INNER JOIN customer_address
       ON (customer_address.customer_id = customer.id AND
           order.customer_address_id = customer_address.id)
 WHERE customer.id = 1234567890;
```

Kueri sebelumnya secara keliru menggabungkan `customer.id` ke `customer.id`, sehingga memberikan hasil perkalian Cartesian antara setiap pelanggan dan setiap pesanan. Jenis join yang tak terduga ini menghasilkan file sementara yang besar. Tergantung pada ukuran tabel, kueri Cartesian bahkan dapat memenuhi penyimpanan. Aplikasi Anda dapat memiliki join Cartesian jika kondisi berikut terpenuhi:

- Anda melihat penurunan besar dan drastis dalam ketersediaan penyimpanan, yang diikuti oleh pemulihan cepat.
- Tidak ada indeks yang dibuat.
- Tidak ada pernyataan `CREATE TABLE FROM SELECT` yang dikeluarkan.

- Tidak ada tampilan terwujud yang disegarkan.

Untuk melihat apakah tabel sedang digabungkan menggunakan kunci yang tepat, periksa kueri dan petunjuk pemetaan relasional objek Anda. Perlu diperhatikan bahwa kueri tertentu dari aplikasi Anda tidak dipanggil sepanjang waktu, dan beberapa kueri dihasilkan secara dinamis.

Periksa kueri ORDER BY dan GROUP BY

Dalam beberapa kasus, klausa `ORDER BY` dapat menghasilkan file sementara yang berlebihan. Pertimbangkan panduan berikut ini:

- Hanya sertakan kolom dalam klausa `ORDER BY` saat kolom tersebut perlu diurutkan. Pedoman ini sangat penting untuk kueri yang menampilkan ribuan baris dan menentukan banyak kolom dalam klausa `ORDER BY`.
- Pertimbangkan untuk membuat indeks guna mempercepat klausa `ORDER BY` saat klausa cocok dengan kolom yang memiliki urutan naik atau turun yang sama. Indeks sebagian lebih direkomendasikan karena lebih kecil. Indeks yang lebih kecil lebih cepat untuk dibaca dan di-traverse.
- Jika Anda membuat indeks untuk kolom yang dapat menerima nilai kosong, pertimbangkan apakah Anda ingin nilai kosong disimpan di akhir atau di awal indeks.

  Jika memungkinkan, kurangi jumlah baris yang perlu diurutkan dengan memfilter set hasil. Jika Anda menggunakan pernyataan klausa atau subkueri `WITH`, perlu diperhatikan bahwa kueri dalam menghasilkan set hasil, lalu meneruskannya ke kueri luar. Semakin banyak baris yang dapat difilter kueri, semakin sedikit pengurutan yang perlu dilakukan kueri.
- Jika Anda tidak perlu mendapatkan set hasil lengkap, gunakan klausa `LIMIT`. Misalnya, jika Anda hanya menginginkan lima baris teratas, kueri yang menggunakan klausa `LIMIT` tidak akan terus memberikan hasil. Dengan cara ini, kueri membutuhkan lebih sedikit memori dan file sementara.

Kueri yang menggunakan klausa `GROUP BY` juga dapat memerlukan file sementara. Kueri `GROUP BY` meringkas nilai dengan menggunakan fungsi seperti berikut:

- `COUNT`
- `AVG`
- `MIN`
- `MAX`

- SUM
- STDDEV

Untuk menyetel kueri GROUP BY, ikuti rekomendasi untuk kueri ORDER BY.

Hindari menggunakan operasi DISTINCT

Jika memungkinkan, jangan gunakan operasi DISTINCT untuk menghapus baris duplikat. Semakin banyak baris duplikat yang tidak perlu, yang ditampilkan oleh kueri Anda, operasi DISTINCT menjadi semakin mahal. Jika memungkinkan, tambahkan filter dalam klausa WHERE meskipun Anda menggunakan filter yang sama untuk tabel yang berbeda. Memfilter kueri dan melakukan join dengan benar akan meningkatkan performa Anda serta mengurangi penggunaan sumber daya. Hal tersebut juga mencegah laporan dan hasil yang salah.

Jika Anda perlu menggunakan DISTINCT untuk beberapa baris dari tabel yang sama, pertimbangkan untuk membuat indeks komposit. Mengelompokkan beberapa kolom dalam indeks dapat meningkatkan waktu untuk mengevaluasi baris yang berbeda. Selain itu, jika Anda menggunakan RDS for PostgreSQL versi 10 atau lebih tinggi, Anda dapat mengorelasikan statistik di antara beberapa kolom dengan menggunakan perintah CREATE STATISTICS.

Mempertimbangkan untuk menggunakan fungsi jendela alih-alih fungsi GROUP BY

Dengan menggunakan GROUP BY, Anda mengubah set hasil, lalu mengambil hasil agregat. Dengan menggunakan fungsi jendela, Anda mengumpulkan data tanpa mengubah set hasil. Fungsi jendela menggunakan klausa OVER untuk melakukan penghitungan di seluruh set yang ditentukan oleh kueri, dengan mengorelasikan satu baris dengan yang lain. Anda dapat menggunakan semua fungsi GROUP BY dalam fungsi jendela, tetapi juga menggunakan fungsi seperti berikut:

- RANK
- ARRAY_AGG
- ROW_NUMBER
- LAG
- LEAD

Untuk meminimalkan jumlah file sementara yang dihasilkan oleh fungsi jendela, hapus duplikasi untuk set hasil yang sama saat Anda membutuhkan dua agregasi yang berbeda. Pertimbangkan kueri berikut.

```
SELECT sum(salary) OVER (PARTITION BY dept ORDER BY salary DESC) as sum_salary
     , avg(salary) OVER (PARTITION BY dept ORDER BY salary ASC) as avg_salary
  FROM empsalary;
```

Anda dapat menulis ulang kueri dengan klausa `WINDOW` sebagai berikut:

```
SELECT sum(salary) OVER w as sum_salary
         , avg(salary) OVER w as_avg_salary
    FROM empsalary
  WINDOW w AS (PARTITION BY dept ORDER BY salary DESC);
```

Secara default, perencana eksekusi Aurora PostgreSQL menggabungkan simpul yang serupa, sehingga tidak menggandakan operasi. Namun, dengan menggunakan deklarasi eksplisit untuk blok jendela, Anda dapat mengelola kueri dengan lebih mudah. Anda juga dapat meningkatkan performa dengan mencegah duplikasi.

Selidiki tampilan terwujud dan pernyataan CTAS

Saat tampilan terwujud disegarkan, kueri akan dijalankan. Kueri ini dapat berisi operasi seperti `GROUP BY`, `ORDER BY`, atau `DISTINCT`. Selama penyegaran, Anda mungkin mengamati sejumlah besar file sementara serta peristiwa tunggu `IO:BufFileWrite` dan `IO:BufFileRead`. Demikian pula, saat Anda membuat tabel berdasarkan pernyataan `SELECT`, pernyataan `CREATE TABLE` tersebut menjalankan kueri. Untuk mengurangi file sementara yang dibutuhkan, optimalkan kueri.

Gunakan pg_repack saat Anda membuat kembali indeks

Saat Anda membuat indeks, mesin mengurutkan set hasil. Seiring tabel bertambah besar, dan seiring nilai di kolom yang diindeks menjadi lebih beragam, file sementara membutuhkan lebih banyak ruang. Dalam kebanyakan kasus, Anda tidak dapat mencegah pembuatan file sementara untuk tabel besar tanpa memodifikasi area memori kerja pemeliharaan. Untuk informasi selengkapnya tentang `maintenance_work_mem`, lihat https://www.postgresql.org/docs/current/runtime-config-resource.html dalam dokumentasi PostgreSQL.

Solusi yang mungkin saat membuat ulang indeks besar adalah dengan menggunakan ekstensi pg_repack. Untuk informasi selengkapnya, lihat Reorganize tables in PostgreSQL databases with minimal locks dalam dokumentasi pg_repack. Untuk informasi tentang menyiapkan ekstensi di instans DB RDS for PostgreSQL Anda, lihat Mengurangi bloat dalam tabel dan indeks dengan ekstensi pg_repack.

Tingkatkan maintenance_work_mem saat Anda membuat klaster tabel

Perintah `CLUSTER` membuat klaster tabel yang ditentukan menurut table_name berdasarkan indeks yang ada yang ditentukan menurut index_name. RDS for PostgreSQL secara fisik membuat ulang tabel agar sesuai dengan urutan indeks yang diberikan.

Saat penyimpanan magnetik lazim digunakan, pembuatan klaster menjadi umum dilakukan karena throughput penyimpanan terbatas. Sekarang penyimpanan berbasis SSD sudah umum, sehingga pembuatan klaster menjadi kurang populer. Namun, jika Anda membuat klaster tabel, Anda masih dapat sedikit meningkatkan performa tergantung pada ukuran tabel, indeks, kueri, dan banyak lagi.

Jika Anda menjalankan perintah `CLUSTER` dan mengamati peristiwa tunggu `IO:BufFileWrite` dan `IO:BufFileRead`, setel `maintenance_work_mem`. Tingkatkan ukuran memori ke jumlah yang cukup besar. Nilai tinggi berarti mesin dapat menggunakan lebih banyak memori untuk operasi klaster.

Setel memori untuk mencegah IO:BufFileRead dan IO:BufFileWrite

Dalam beberapa situasi, Anda perlu menyetel memori. Tujuan Anda adalah menyeimbangkan memori di seluruh area konsumsi berikut menggunakan parameter yang sesuai, sebagai berikut.

- Nilai `work_mem`
- Memori yang tersisa setelah mengecualikan nilai `shared_buffers`
- Koneksi maksimum yang dibuka dan digunakan, yang dibatasi oleh `max_connections`

Untuk informasi selengkapnya tentang penyetelan memori, lihat Resource Consumption dalam dokumentasi PostgreSQL.

Tingkatkan ukuran area memori kerja

Dalam beberapa situasi, satu-satunya pilihan adalah menambah memori yang digunakan oleh sesi Anda. Jika kueri Anda ditulis dengan benar dan menggunakan kunci yang benar untuk join, pertimbangkan untuk meningkatkan nilai `work_mem`.

Untuk mengetahui jumlah file sementara yang dihasilkan kueri, atur `log_temp_files` ke `0`. Jika meningkatkan nilai `work_mem` ke nilai maksimum yang diidentifikasi dalam log, Anda mencegah kueri menghasilkan file sementara. Namun, `work_mem` menetapkan nilai maksimum per simpul rencana untuk setiap koneksi atau pekerja paralel. Jika basis data memiliki 5.000 koneksi, dan jika masing-masing menggunakan memori 256 MiB, mesin akan membutuhkan RAM 1,2 TiB. Oleh karena itu, instans Anda dapat kehabisan memori.

Cadangkan memori yang cukup untuk pool buffer bersama

Basis data Anda menggunakan area memori seperti pool buffer bersama, bukan hanya area memori kerja. Pertimbangkan persyaratan area memori tambahan ini sebelum Anda meningkatkan `work_mem`.

Misalnya, anggaplah kelas instans Aurora PostgreSQL Anda adalah db.r5.2xlarge. Kelas ini memiliki memori 64 GiB. Secara default, 25 persen memori dicadangkan untuk pool buffer bersama. Setelah Anda mengurangi jumlah yang dialokasikan ke area memori bersama, 16.384 MB tetap ada. Jangan mengalokasikan memori yang tersisa hanya ke area memori kerja karena sistem operasi dan mesin juga memerlukan memori.

Memori yang dapat Anda alokasikan ke `work_mem` tergantung pada kelas instans. Jika Anda menggunakan kelas instans yang lebih besar, memori yang tersedia akan lebih banyak. Namun, dalam contoh sebelumnya, Anda tidak dapat menggunakan lebih dari 16 GiB. Jika melakukannya, instans Anda menjadi tidak tersedia saat kehabisan memori. Untuk memulihkan instans dari status tidak tersedia, layanan otomatisasi PostgreSQL Aurora secara otomatis dimulai ulang.

Kelola jumlah koneksi

Misalnya, instans basis data Anda memiliki 5.000 koneksi simultan. Setiap koneksi menggunakan setidaknya 4 MiB `work_mem`. Konsumsi memori yang tinggi dari koneksi cenderung menurunkan performa. Untuk mengatasinya, Anda memiliki opsi berikut:

- Upgrade ke kelas instans yang lebih besar.
- Kurangi jumlah koneksi basis data simultan dengan menggunakan proksi atau pooler koneksi.

Untuk proksi, pertimbangkan Proksi Amazon RDS, pgBouncer, atau pooler koneksi berdasarkan aplikasi Anda. Solusi ini mengurangi beban CPU. Solusi ini juga mengurangi risiko saat semua koneksi memerlukan area memori kerja. Saat koneksi basis data lebih sedikit, Anda dapat meningkatkan nilai `work_mem`. Dengan cara ini, Anda mengurangi munculnya peristiwa tunggu `IO:BufFileRead` dan `IO:BufFileWrite`. Selain itu, kueri yang menunggu area memori kerja akan dipercepat secara signifikan.

IO: DataFileRead

Peristiwa `IO:DataFileRead` terjadi saat koneksi menunggu proses backend untuk membaca halaman yang diperlukan dari penyimpanan karena halaman tidak tersedia dalam memori bersama.

Topik

Versi mesin yang didukung

Informasi peristiwa tunggu ini didukung untuk semua versi RDS for PostgreSQL.

Konteks

Semua kueri dan operasi manipulasi data (DML) mengakses halaman di pool buffer. Pernyataan yang dapat menimbulkan pembacaan mencakup `SELECT`, `UPDATE`, dan `DELETE`. Misalnya, `UPDATE` dapat membaca halaman dari tabel atau indeks. Jika halaman yang diminta atau diperbarui tidak berada dalam pool buffer bersama, pembacaan ini dapat mengarah ke peristiwa `IO:DataFileRead`.

Karena bersifat terbatas, pool buffer bersama dapat menjadi penuh. Dalam hal ini, permintaan untuk halaman yang tidak berada dalam memori akan memaksa basis data untuk membaca blok dari disk. Jika peristiwa `IO:DataFileRead` sering terjadi, pool buffer bersama mungkin terlalu kecil untuk mengakomodasi beban kerja Anda. Masalah ini bersifat akut untuk kueri `SELECT` yang membaca sejumlah besar baris yang tidak dapat ditampung pool buffer. Untuk informasi selengkapnya tentang pool buffer, lihat Resource Consumption dalam dokumentasi PostgreSQL.

Kemungkinan penyebab peningkatan peristiwa tunggu

Penyebab umum peristiwa `IO:DataFileRead` mencakup:

Lonjakan koneksi

Anda mungkin menemukan beberapa koneksi yang menghasilkan jumlah acara IO: DataFileRead wait yang sama. Dalam hal ini, lonjakan (peningkatan tiba-tiba dan besar) dalam peristiwa `IO:DataFileRead` dapat terjadi.

Pernyataan SELECT dan DML yang melakukan pemindaian berurutan

Aplikasi Anda mungkin melakukan operasi baru. Atau, operasi yang ada mungkin berubah karena rencana eksekusi baru. Dalam kasus ini, cari tabel (terutama tabel besar) yang memiliki nilai `seq_scan` yang lebih besar. Temukan tabel ini dengan membuat kueri `pg_stat_user_tables`.

Untuk melacak kueri yang menghasilkan lebih banyak operasi baca, gunakan ekstensi `pg_stat_statements`.

CTAS dan CREATE INDEX untuk set data besar

CTAS adalah pernyataan `CREATE TABLE AS SELECT`. Jika Anda menjalankan CTAS menggunakan set data besar sebagai sumber, atau membuat indeks pada tabel besar, maka peristiwa `IO:DataFileRead` dapat terjadi. Saat Anda membuat indeks, basis data mungkin perlu membaca seluruh objek menggunakan pemindaian berurutan. CTAS menghasilkan pembacaan `IO:DataFile` saat halaman tidak ada dalam memori.

Beberapa pekerja vakum berjalan pada waktu yang sama

Pekerja vakum dapat dipicu secara manual atau otomatis. Sebaiknya adopsi strategi vakum yang agresif. Namun, saat tabel memiliki banyak baris yang diperbarui atau dihapus, peristiwa tunggu `IO:DataFileRead` bertambah. Setelah ruang diklaim kembali, waktu vakum yang dihabiskan pada `IO:DataFileRead` berkurang.

Menyerap data dalam jumlah besar

Saat aplikasi Anda menyerap data dalam jumlah besar, operasi `ANALYZE` mungkin terjadi lebih sering. Proses `ANALYZE` dapat dipicu oleh peluncur autovacuum atau diinvokasi secara manual.

Operasi `ANALYZE` membaca subset tabel. Jumlah halaman yang harus dipindai dihitung dengan mengalikan 30 dengan nilai `default_statistics_target`. Untuk informasi selengkapnya, lihat dokumentasi PostgreSQL. Parameter `default_statistics_target` menerima nilai antara 1 hingga 10.000, dengan nilai default 100.

Kekurangan sumber daya

Jika bandwidth jaringan instans atau CPU dikonsumsi, peristiwa `IO:DataFileRead` mungkin terjadi lebih sering.

Tindakan

Kami merekomendasikan berbagai tindakan, tergantung pada penyebab peristiwa tunggu Anda.

Topik

- Memeriksa filter predikat untuk kueri yang menghasilkan peristiwa tunggu
- Minimalkan efek operasi pemeliharaan
- Tangani jumlah koneksi yang tinggi

Memeriksa filter predikat untuk kueri yang menghasilkan peristiwa tunggu

Asumsikan bahwa Anda mengidentifikasi kueri spesifik yang menghasilkan peristiwa tunggu `IO:DataFileRead`. Anda mungkin mengidentifikasinya menggunakan teknik berikut:

- Wawasan Performa
- Tampilan katalog seperti yang disediakan oleh ekstensi `pg_stat_statements`
- Tampilan katalog `pg_stat_all_tables`, jika secara berkala menunjukkan peningkatan jumlah pembacaan fisik
- Tampilan `pg_statio_all_tables`, jika menunjukkan bahwa penghitung `_read` meningkat

Sebaiknya Anda menentukan filter yang akan digunakan dalam predikat (klausa `WHERE`) kueri ini. Ikuti pedoman berikut:

- Jalankan perintah `EXPLAIN`. Pada output, identifikasi jenis pemindaian yang digunakan. Pemindaian berurutan tidak selalu menunjukkan adanya masalah. Kueri yang menggunakan pemindaian berurutan tentunya akan menghasilkan lebih banyak peristiwa `IO:DataFileRead` jika dibandingkan dengan kueri yang menggunakan filter.

  Cari tahu apakah kolom yang tercantum dalam klausa `WHERE` diindeks. Jika tidak, coba buat indeks untuk kolom ini. Pendekatan ini mencegah pemindaian berurutan dan mengurangi peristiwa `IO:DataFileRead`. Jika kueri memiliki filter yang bersifat membatasi dan masih menghasilkan pemindaian berurutan, evaluasi apakah indeks yang tepat sedang digunakan.
- Cari tahu apakah kueri mengakses tabel yang sangat besar. Dalam beberapa kasus, partisi tabel dapat meningkatkan performa, dengan memungkinkan kueri hanya membaca partisi yang diperlukan.
- Periksa kardinalitas (jumlah total baris) dari operasi join Anda. Perhatikan seberapa membatasikah nilai yang Anda teruskan di filter untuk klausa `WHERE` Anda. Jika memungkinkan, setel kueri Anda untuk mengurangi jumlah baris yang diteruskan di setiap langkah rencana.

Minimalkan efek operasi pemeliharaan

Operasi pemeliharaan seperti `VACUUM` dan `ANALYZE` penting untuk dilakukan. Sebaiknya jangan dinonaktifkan karena peristiwa tunggu `IO:DataFileRead` berkaitan dengan operasi pemeliharaan ini. Pendekatan berikut dapat meminimalkan efek operasi ini:

- Jalankan operasi pemeliharaan secara manual di luar jam sibuk. Teknik ini mencegah basis data mencapai ambang batas untuk operasi otomatis.
- Untuk tabel yang sangat besar, pertimbangkan untuk mempartisi tabel. Teknik ini mengurangi overhead operasi pemeliharaan. Basis data hanya mengakses partisi yang memerlukan pemeliharaan.
- Saat Anda menyerap data dalam jumlah besar, coba nonaktifkan fitur analisis otomatis.

Fitur autovacuum secara otomatis dipicu untuk tabel saat rumus berikut benar.

```
pg_stat_user_tables.n_dead_tup > (pg_class.reltuples x autovacuum_vacuum_scale_factor)
 + autovacuum_vacuum_threshold
```

Tampilan `pg_stat_user_tables` dan katalog `pg_class` memiliki beberapa baris. Satu baris dapat sesuai dengan satu baris di tabel Anda. Rumus ini mengasumsikan bahwa `reltuples` tersebut ditujukan untuk tabel tertentu. Parameter `autovacuum_vacuum_scale_factor` (0.20 secara default) dan `autovacuum_vacuum_threshold` (50 tuple secara default) biasanya diatur secara global untuk keseluruhan instans. Namun, Anda dapat menetapkan nilai berbeda untuk tabel tertentu.

Topik

- Temukan tabel yang mengonsumsi ruang secara tidak perlu
- Temukan tabel yang mengonsumsi ruang secara tidak perlu
- Temukan tabel yang memenuhi syarat untuk di-autovacuum

Temukan tabel yang mengonsumsi ruang secara tidak perlu

Untuk menemukan tabel yang menghabiskan ruang secara tidak perlu, Anda dapat menggunakan fungsi dari ekstensi `pgstattuple` PostgreSQL. Ekstensi (modul) ini tersedia secara default di semua instans DB RDS for PostgreSQL dan dapat diinstansiasi pada instans dengan perintah berikut.

```
CREATE EXTENSION pgstattuple;
```

Untuk informasi selengkapnya tentang ekstensi ini, lihat pgstattuple dalam dokumentasi PostgreSQL.

Anda dapat memeriksa bloat tabel dan indeks di aplikasi Anda. Untuk informasi selengkapnya, lihat Mendiagnosis bloat tabel dan indeks.

Temukan tabel yang mengonsumsi ruang secara tidak perlu

Untuk menemukan indeks yang mengalami bloat dan memperkirakan jumlah ruang yang dikonsumsi secara tidak perlu pada tabel yang hak akses bacanya Anda miliki, Anda dapat menjalankan kueri berikut.

```
-- WARNING: rows with is_na = 't' are known to have bad statistics ("name" type is not
 supported).
-- This query is compatible with PostgreSQL 8.2 and later.

SELECT current_database(), nspname AS schemaname, tblname, idxname,
 bs*(relpages)::bigint AS real_size,
  bs*(relpages-est_pages)::bigint AS extra_size,
  100 * (relpages-est_pages)::float / relpages AS extra_ratio,
  fillfactor, bs*(relpages-est_pages_ff) AS bloat_size,
  100 * (relpages-est_pages_ff)::float / relpages AS bloat_ratio,
  is_na
  -- , 100-(sub.pst).avg_leaf_density, est_pages, index_tuple_hdr_bm,
  -- maxalign, pagehdr, nulldatawidth, nulldatahdrwidth, sub.reltuples, sub.relpages
  -- (DEBUG INFO)
FROM (
  SELECT coalesce(1 +
       ceil(reltuples/floor((bs-pageopqdata-pagehdr)/(4+nulldatahdrwidth)::float)), 0
       -- ItemIdData size + computed avg size of a tuple (nulldatahdrwidth)
    ) AS est_pages,
    coalesce(1 +
       ceil(reltuples/floor((bs-pageopqdata-pagehdr)*fillfactor/
(100*(4+nulldatahdrwidth)::float))), 0
    ) AS est_pages_ff,
    bs, nspname, table_oid, tblname, idxname, relpages, fillfactor, is_na
    -- , stattuple.pgstatindex(quote_ident(nspname)||'.'||quote_ident(idxname)) AS
 pst,
    -- index_tuple_hdr_bm, maxalign, pagehdr, nulldatawidth, nulldatahdrwidth,
 reltuples
    -- (DEBUG INFO)
  FROM (
    SELECT maxalign, bs, nspname, tblname, idxname, reltuples, relpages, relam,
 table_oid, fillfactor,
      ( index_tuple_hdr_bm +
          maxalign - CASE -- Add padding to the index tuple header to align on MAXALIGN
            WHEN index_tuple_hdr_bm%maxalign = 0 THEN maxalign
            ELSE index_tuple_hdr_bm%maxalign
          END
```

```
       + nulldatawidth + maxalign - CASE -- Add padding to the data to align on
MAXALIGN
           WHEN nulldatawidth = 0 THEN 0
           WHEN nulldatawidth::integer%maxalign = 0 THEN maxalign
           ELSE nulldatawidth::integer%maxalign
         END
     )::numeric AS nulldatahdrwidth, pagehdr, pageopqdata, is_na
     -- , index_tuple_hdr_bm, nulldatawidth -- (DEBUG INFO)
   FROM (
     SELECT
       i.nspname, i.tblname, i.idxname, i.reltuples, i.relpages, i.relam, a.attrelid
AS table_oid,
       current_setting('block_size')::numeric AS bs, fillfactor,
       CASE -- MAXALIGN: 4 on 32bits, 8 on 64bits (and mingw32 ?)
         WHEN version() ~ 'mingw32' OR version() ~ '64-bit|x86_64|ppc64|ia64|amd64'
THEN 8
         ELSE 4
       END AS maxalign,
       /* per page header, fixed size: 20 for 7.X, 24 for others */
       24 AS pagehdr,
       /* per page btree opaque data */
       16 AS pageopqdata,
       /* per tuple header: add IndexAttributeBitMapData if some cols are null-able */
       CASE WHEN max(coalesce(s.null_frac,0)) = 0
         THEN 2 -- IndexTupleData size
         ELSE 2 + (( 32 + 8 - 1 ) / 8)
         -- IndexTupleData size + IndexAttributeBitMapData size ( max num filed per
index + 8 - 1 /8)
       END AS index_tuple_hdr_bm,
       /* data len: we remove null values save space using it fractionnal part from
stats */
       sum( (1-coalesce(s.null_frac, 0)) * coalesce(s.avg_width, 1024)) AS
nulldatawidth,
       max( CASE WHEN a.atttypid = 'pg_catalog.name'::regtype THEN 1 ELSE 0 END ) > 0
AS is_na
     FROM pg_attribute AS a
       JOIN (
         SELECT nspname, tbl.relname AS tblname, idx.relname AS idxname,
           idx.reltuples, idx.relpages, idx.relam,
           indrelid, indexrelid, indkey::smallint[] AS attnum,
           coalesce(substring(
             array_to_string(idx.reloptions, ' ')
              from 'fillfactor=([0-9]+)')::smallint, 90) AS fillfactor
         FROM pg_index
```

```
            JOIN pg_class idx ON idx.oid=pg_index.indexrelid
            JOIN pg_class tbl ON tbl.oid=pg_index.indrelid
            JOIN pg_namespace ON pg_namespace.oid = idx.relnamespace
          WHERE pg_index.indisvalid AND tbl.relkind = 'r' AND idx.relpages > 0
        ) AS i ON a.attrelid = i.indexrelid
        JOIN pg_stats AS s ON s.schemaname = i.nspname
          AND ((s.tablename = i.tblname AND s.attname =
 pg_catalog.pg_get_indexdef(a.attrelid, a.attnum, TRUE))
          -- stats from tbl
          OR  (s.tablename = i.idxname AND s.attname = a.attname))
          -- stats from functional cols
        JOIN pg_type AS t ON a.atttypid = t.oid
      WHERE a.attnum > 0
      GROUP BY 1, 2, 3, 4, 5, 6, 7, 8, 9
    ) AS s1
  ) AS s2
    JOIN pg_am am ON s2.relam = am.oid WHERE am.amname = 'btree'
) AS sub
-- WHERE NOT is_na
ORDER BY 2,3,4;
```

Temukan tabel yang memenuhi syarat untuk di-autovacuum

Untuk menemukan tabel yang memenuhi syarat untuk di-autovacuum, jalankan kueri berikut.

```
--This query shows tables that need vacuuming and are eligible candidates.
--The following query lists all tables that are due to be processed by autovacuum.
-- During normal operation, this query should return very little.
WITH  vbt AS (SELECT setting AS autovacuum_vacuum_threshold
              FROM pg_settings WHERE name = 'autovacuum_vacuum_threshold')
    , vsf AS (SELECT setting AS autovacuum_vacuum_scale_factor
              FROM pg_settings WHERE name = 'autovacuum_vacuum_scale_factor')
    , fma AS (SELECT setting AS autovacuum_freeze_max_age
              FROM pg_settings WHERE name = 'autovacuum_freeze_max_age')
    , sto AS (SELECT opt_oid, split_part(setting, '=', 1) as param,
                split_part(setting, '=', 2) as value
              FROM (SELECT oid opt_oid, unnest(reloptions) setting FROM pg_class) opt)
SELECT
    '"'||ns.nspname||'"."'||c.relname||'"' as relation
    , pg_size_pretty(pg_table_size(c.oid)) as table_size
    , age(relfrozenxid) as xid_age
    , coalesce(cfma.value::float, autovacuum_freeze_max_age::float)
 autovacuum_freeze_max_age
    , (coalesce(cvbt.value::float, autovacuum_vacuum_threshold::float) +
```

```
        coalesce(cvsf.value::float,autovacuum_vacuum_scale_factor::float) *
 c.reltuples)
        as autovacuum_vacuum_tuples
    , n_dead_tup as dead_tuples
FROM pg_class c
JOIN pg_namespace ns ON ns.oid = c.relnamespace
JOIN pg_stat_all_tables stat ON stat.relid = c.oid
JOIN vbt on (1=1)
JOIN vsf ON (1=1)
JOIN fma on (1=1)
LEFT JOIN sto cvbt ON cvbt.param = 'autovacuum_vacuum_threshold' AND c.oid =
 cvbt.opt_oid
LEFT JOIN sto cvsf ON cvsf.param = 'autovacuum_vacuum_scale_factor' AND c.oid =
 cvsf.opt_oid
LEFT JOIN sto cfma ON cfma.param = 'autovacuum_freeze_max_age' AND c.oid = cfma.opt_oid
WHERE c.relkind = 'r'
AND nspname <> 'pg_catalog'
AND (
    age(relfrozenxid) >= coalesce(cfma.value::float, autovacuum_freeze_max_age::float)
    or
    coalesce(cvbt.value::float, autovacuum_vacuum_threshold::float) +
      coalesce(cvsf.value::float,autovacuum_vacuum_scale_factor::float) * c.reltuples
 <= n_dead_tup
    -- or 1 = 1
)
ORDER BY age(relfrozenxid) DESC;
```

Tangani jumlah koneksi yang tinggi

Saat Anda memantau Amazon CloudWatch, Anda mungkin menemukan bahwa `DatabaseConnections` metrik melonjak. Peningkatan ini menunjukkan bertambahnya jumlah koneksi ke basis data Anda. Sebaiknya lakukan pendekatan berikut:

- Batasi jumlah koneksi yang dapat dibuka aplikasi dengan setiap instans. Jika aplikasi Anda memiliki fitur pool koneksi tersemat, tetapkan jumlah koneksi yang wajar. Tentukan jumlahnya berdasarkan paralelisasi yang dapat secara efektif ditangani vCPU dalam instans Anda.

  Jika aplikasi Anda tidak menggunakan fitur pool koneksi, coba gunakan Proksi Amazon RDS atau alternatifnya. Pendekatan ini memungkinkan aplikasi Anda membuka beberapa koneksi dengan penyeimbang beban. Penyeimbang selanjutnya dapat membuka jumlah koneksi yang terbatas dengan basis data. Karena lebih sedikit koneksi yang berjalan secara paralel, instans DB Anda melakukan lebih sedikit peralihan konteks di kernel. Kueri akan berjalan lebih cepat, sehingga

mengurangi peristiwa tunggu. Untuk informasi selengkapnya, lihat Menggunakan Proksi Amazon RDS.

- Jika memungkinkan, manfaatkan replika baca RDS for PostgreSQL. Saat aplikasi Anda menjalankan operasi hanya-baca, kirim permintaan ini ke replika pembaca. Teknik ini mengurangi tekanan I/O pada simpul primer (penulis).
- Coba naikkan skala instans DB Anda. Kelas instans berkapasitas lebih tinggi memberikan memori lebih banyak, yang memberi RDS for PostgreSQL pool buffer bersama yang lebih besar untuk menampung halaman. Ukuran lebih besar juga memberikan instans DB lebih banyak vCPU untuk menangani koneksi. Lebih banyak vCPU akan sangat membantu saat operasi yang menghasilkan peristiwa tunggu `IO:DataFileRead` adalah operasi tulis.

IO:WALWrite

Topik

Versi mesin yang didukung

Informasi peristiwa tunggu ini didukung untuk semua RDS for PostgreSQL versi 10 dan yang lebih tinggi.

Konteks

Aktivitas dalam basis data yang menghasilkan data log write-ahead mengisi akan buffer WAL terlebih dahulu lalu menulis ke disk, secara asinkron. Peristiwa tunggu `IO:WALWrite` dihasilkan ketika sesi SQL menunggu data WAL selesai ditulis ke disk sehingga dapat melepaskan panggilan COMMIT transaksi.

Kemungkinan penyebab peningkatan peristiwa tunggu

Jika peristiwa tunggu ini sering terjadi, Anda harus meninjau beban kerja Anda dan jenis pembaruan yang dilakukan beban kerja Anda serta frekuensinya. Secara khusus, cari jenis aktivitas berikut.

Aktivitas DML yang berat

Perubahan data pada tabel basis data tidak terjadi secara instan. Penyisipan ke satu tabel mungkin perlu menunggu penyisipan atau pembaruan ke tabel yang sama dari klien lain. Pernyataan bahasa manipulasi data (DML) untuk mengubah nilai data (INSERT, UPDATE, DELETE, COMMIT, ROLLBACK TRANSACTION) dapat mengakibatkan pertentangan sehingga file log write-ahead yang menunggu buffer di-flushing. Situasi ini dicatat dalam metrik Wawasan Performa Amazon RDS berikut yang menunjukkan aktivitas DML yang berat.

- `tup_inserted`
- `tup_updated`
- `tup_deleted`
- `xcat_rollback`
- `xact_commit`

Untuk informasi selengkapnya tentang metrik ini, lihat Penghitung Wawasan Performa untuk Amazon RDS for PostgreSQL.

Aktivitas checkpoint yang sering

Checkpoint yang sering akan berkontribusi pada ukuran WAL yang besar. Di RDS for PostgreSQL, penulisan halaman lengkap selalu "aktif". Penulisan halaman penuh membantu melindungi dari kehilangan data. Namun, jika pembuatan checkpoint terjadi terlalu sering, sistem dapat mengalami masalah performa secara keseluruhan. Hal ini terutama berlaku pada sistem dengan aktivitas DML yang berat. Dalam beberapa kasus, Anda mungkin menemukan pesan kesalahan di `postgresql.log` Anda yang menyatakan bahwa “checkpoint terjadi terlalu sering".

Saat menyetel checkpoint, sebaiknya Anda menyeimbangkan performa dengan hati-hati berdasarkan perkiraan waktu pemulihan jika terjadi penonaktifan yang tidak normal.

Tindakan

Kami merekomendasikan tindakan berikut untuk mengurangi jumlah peristiwa tunggu ini.

Topik

- Kurangi jumlah commit
- Pantau checkpoint Anda

- Naikkan skala IO
- Volume log khusus (DLV)

Kurangi jumlah commit

Untuk mengurangi jumlah commit, gabungkan pernyataan ke dalam blok transaksi. Gunakan Wawasan Performa Amazon RDS untuk memeriksa jenis kueri yang dijalankan. Anda juga dapat memindahkan operasi pemeliharaan besar ke waktu di luar jam sibuk. Misalnya, buat indeks atau gunakan operasi `pg_repack` selama jam non-produksi.

Pantau checkpoint Anda

Ada dua parameter yang dapat Anda pantau untuk melihat seberapa sering instans DB RDS for PostgreSQL Anda menulis ke file WAL untuk checkpoint.

- `log_checkpoints` – Parameter ini diatur ke "aktif" secara default. Hal ini menyebabkan pesan dikirim ke log PostgreSQL untuk setiap checkpoint. Pesan log ini mencakup jumlah buffer yang ditulis, waktu yang dihabiskan untuk menulisnya, dan jumlah file WAL yang ditambahkan, dihapus, atau didaur ulang untuk checkpoint tertentu.

  Untuk informasi selengkapnya tentang parameter ini, lihat Error Reporting and Logging dalam dokumentasi PostgreSQL.
- `checkpoint_warning` – Parameter ini menetapkan nilai ambang batas (dalam detik) untuk frekuensi checkpoint yang jika terlampaui, akan menghasilkan peringatan. Secara default, parameter ini tidak diatur di RDS for PostgreSQL. Anda dapat mengatur nilai parameter ini untuk mendapatkan peringatan ketika perubahan basis data di instans DB RDS for PostgreSQL Anda ditulis pada laju yang tidak dapat ditangani oleh ukuran file WAL. Misalnya, Anda mengatur parameter ini ke 30. Jika instans RDS for PostgreSQL Anda perlu menulis perubahan lebih sering daripada setiap 30 detik, peringatan bahwa "checkpoint terjadi terlalu sering" akan dikirim ke log PostgreSQL. Hal ini dapat menunjukkan bahwa nilai `max_wal_size` Anda harus ditingkatkan.

  Untuk informasi selengkapnya, lihat Write Ahead Log dalam dokumentasi PostgreSQL.

Naikkan skala IO

Jenis peristiwa tunggu input/output (IO) ini dapat diatasi dengan menskalakan operasi input/output per detik (IOPS) untuk menyediakan IO yang lebih cepat. Penskalaan IO lebih direkomendasikan daripada penskalaan CPU karena penskalaan CPU dapat menghasilkan lebih banyak pertentangan

IO. Hal ini terjadi karena CPU yang ditingkatkan dapat menangani lebih banyak pekerjaan dan dengan demikian membuat bottleneck IO semakin buruk. Secara umum, kami menyarankan Anda mempertimbangkan untuk menyetel beban kerja Anda sebelum melakukan operasi penskalaan.

Volume log khusus (DLV)

Anda dapat menggunakan volume log khusus (DLV) untuk instans DB yang menggunakan penyimpanan IOPS yang Tersedia (PIOPS) dengan menggunakan konsol Amazon RDS, AWS CLI, atau API Amazon RDS. DLV memindahkan log transaksi database PostgreSQL ke volume penyimpanan yang terpisah dari volume yang berisi tabel database. Lihat informasi yang lebih lengkap di Volume log khusus (DLV).

Lock:advisory

Peristiwa `Lock:advisory` terjadi saat aplikasi PostgreSQL menggunakan kunci untuk mengoordinasi aktivitas di beberapa sesi.

Topik

- Versi mesin yang relevan
- Konteks
- Penyebab
- Tindakan

Versi mesin yang relevan

Informasi peristiwa tunggu ini relevan untuk RDS for PostgreSQL versi 9.6 dan lebih tinggi.

Konteks

Kunci advisory PostgreSQL adalah kunci kooperatif tingkat aplikasi yang secara eksplisit dikunci dan dibuka oleh kode aplikasi pengguna. Aplikasi dapat menggunakan kunci advisory PostgreSQL untuk mengoordinasi aktivitas di beberapa sesi. Tidak seperti kunci biasa tingkat objek atau baris, aplikasi memiliki kontrol penuh atas masa pakai kunci. Untuk informasi selengkapnya, lihat Advisory Locks dalam dokumentasi PostgreSQL.

Kunci advisory dapat dilepaskan sebelum transaksi berakhir atau disimpan oleh sesi di seluruh transaksi. Hal ini tidak berlaku untuk kunci implisit yang diberlakukan sistem, seperti kunci eksklusif akses pada tabel yang diperoleh oleh pernyataan `CREATE INDEX`.

Untuk deskripsi fungsi yang digunakan untuk memperoleh (mengunci) dan melepaskan (membuka kunci) kunci advisory, lihat Advisory Lock Functions dalam dokumentasi PostgreSQL.

Kunci advisory diimplementasikan di atas sistem penguncian PostgreSQL biasa dan terlihat dalam tampilan sistem `pg_locks`.

Penyebab

Jenis kunci ini secara khusus dikendalikan oleh aplikasi yang secara eksplisit menggunakannya. Kunci advisory yang diperoleh untuk setiap baris sebagai bagian dari kueri dapat menyebabkan lonjakan kunci atau penumpukan jangka panjang.

Efek ini terjadi saat kueri dijalankan dengan cara yang memperoleh kunci pada lebih banyak baris daripada yang ditampilkan oleh kueri. Aplikasi pada akhirnya harus melepaskan setiap kunci, tetapi jika kunci diperoleh pada baris yang tidak ditampilkan, maka aplikasi tidak dapat menemukan semua kunci.

Contoh berikut berasal dari Advisory Locks dalam dokumentasi PostgreSQL.

```
SELECT pg_advisory_lock(id) FROM foo WHERE id > 12345 LIMIT 100;
```

Dalam contoh ini, klausa `LIMIT` hanya dapat menghentikan output kueri setelah baris dipilih secara internal dan nilai ID-nya dikunci. Hal ini dapat terjadi secara tiba-tiba saat volume data yang bertambah menyebabkan perencana memilih rencana eksekusi lain yang tidak diuji selama pengembangan. Penumpukan dalam kasus ini terjadi karena aplikasi secara eksplisit memanggil `pg_advisory_unlock` untuk setiap nilai ID yang terkunci. Namun, dalam kasus ini, aplikasi tidak dapat menemukan set kunci yang diperoleh pada baris yang tidak ditampilkan. Karena diperoleh pada tingkat sesi, kunci tidak dilepaskan secara otomatis pada akhir transaksi.

Kemungkinan penyebab lain untuk lonjakan upaya kunci yang diblokir adalah konflik yang tidak diinginkan. Dalam konflik ini, bagian aplikasi yang tidak terkait berbagi ruang ID kunci yang sama secara tidak sengaja.

Tindakan

Tinjau penggunaan kunci advisory oleh aplikasi dan cari tahu secara mendetail di mana dan kapan dalam alur aplikasi setiap jenis kunci advisory diperoleh dan dilepaskan.

Ketahui apakah sesi memperoleh terlalu banyak kunci atau sesi yang berjalan lama tidak melepaskan kunci cukup awal, sehingga mengakibatkan penumpukan kunci yang lambat.

Anda dapat memperbaiki penumpukan kunci tingkat sesi yang lambat dengan mengakhiri sesi menggunakan `pg_terminate_backend(pid)`.

Klien yang menunggu kunci advisory muncul dalam `pg_stat_activity` dengan `wait_event_type=Lock` dan `wait_event=advisory`. Anda dapat memperoleh nilai kunci tertentu dengan mengkueri tampilan sistem `pg_locks` untuk `pid` yang sama, dengan mencari `locktype=advisory` dan `granted=f`.

Anda kemudian dapat mengidentifikasi sesi yang memblokir dengan mengkueri `pg_locks` untuk kunci advisory serupa yang memiliki `granted=t`, seperti ditunjukkan dalam contoh berikut.

```
SELECT blocked_locks.pid AS blocked_pid,
         blocking_locks.pid AS blocking_pid,
         blocked_activity.usename AS blocked_user,
         blocking_activity.usename AS blocking_user,
         now() - blocked_activity.xact_start AS blocked_transaction_duration,
         now() - blocking_activity.xact_start AS blocking_transaction_duration,
         concat(blocked_activity.wait_event_type,':',blocked_activity.wait_event) AS
 blocked_wait_event,
         concat(blocking_activity.wait_event_type,':',blocking_activity.wait_event) AS
 blocking_wait_event,
         blocked_activity.state AS blocked_state,
         blocking_activity.state AS blocking_state,
         blocked_locks.locktype AS blocked_locktype,
         blocking_locks.locktype AS blocking_locktype,
         blocked_activity.query AS blocked_statement,
         blocking_activity.query AS blocking_statement
    FROM pg_catalog.pg_locks blocked_locks
    JOIN pg_catalog.pg_stat_activity blocked_activity ON blocked_activity.pid =
 blocked_locks.pid
    JOIN pg_catalog.pg_locks blocking_locks
        ON blocking_locks.locktype = blocked_locks.locktype
        AND blocking_locks.DATABASE IS NOT DISTINCT FROM blocked_locks.DATABASE
        AND blocking_locks.relation IS NOT DISTINCT FROM blocked_locks.relation
        AND blocking_locks.page IS NOT DISTINCT FROM blocked_locks.page
        AND blocking_locks.tuple IS NOT DISTINCT FROM blocked_locks.tuple
        AND blocking_locks.virtualxid IS NOT DISTINCT FROM blocked_locks.virtualxid
        AND blocking_locks.transactionid IS NOT DISTINCT FROM
 blocked_locks.transactionid
        AND blocking_locks.classid IS NOT DISTINCT FROM blocked_locks.classid
        AND blocking_locks.objid IS NOT DISTINCT FROM blocked_locks.objid
        AND blocking_locks.objsubid IS NOT DISTINCT FROM blocked_locks.objsubid
        AND blocking_locks.pid != blocked_locks.pid
```

```
    JOIN pg_catalog.pg_stat_activity blocking_activity ON blocking_activity.pid =
blocking_locks.pid
    WHERE NOT blocked_locks.GRANTED;
```

Semua fungsi API kunci advisory memiliki dua set argumen, baik satu argumen `bigint` maupun dua argumen `integer`:

- Untuk fungsi API dengan satu argumen `bigint`, 32 bit atas berada dalam `pg_locks.classid` dan 32 bit bawah berada dalam `pg_locks.objid`.
- Untuk fungsi API dengan dua argumen `integer`, argumen pertama adalah `pg_locks.classid` dan argumen kedua adalah `pg_locks.objid`.

Nilai `pg_locks.objsubid` menunjukkan form API yang digunakan: 1 berarti satu argumen `bigint`; 2 berarti dua argumen `integer`.

Lock:extend

Peristiwa `Lock:extend` terjadi saat sebuah proses backend menunggu untuk mengunci relasi agar dapat diperluas, sementara proses lain mengunci relasi tersebut untuk tujuan yang sama.

Topik

Versi mesin yang didukung

Informasi peristiwa tunggu ini didukung untuk semua versi RDS for PostgreSQL.

Konteks

Peristiwa `Lock:extend` menunjukkan bahwa proses backend menunggu untuk memperluas relasi yang dikunci oleh proses backend lain saat proses backend lain ini memperluas relasi tersebut. Karena hanya satu proses pada satu waktu yang dapat memperluas relasi, sistem membuat peristiwa tunggu `Lock:extend`. Operasi `INSERT`, `COPY`, dan `UPDATE` dapat menghasilkan peristiwa ini.

Kemungkinan penyebab peningkatan peristiwa tunggu

Saat peristiwa `Lock:extend` muncul lebih dari biasanya, yang mungkin menunjukkan adanya masalah performa, berikut adalah penyebab umumnya:

Lonjakan penyisipan atau pembaruan konkuren ke tabel yang sama

Mungkin terdapat peningkatan jumlah sesi konkuren dengan kueri yang melakukan penyisipan atau pembaruan pada tabel yang sama.

Bandwidth jaringan tidak cukup

Bandwidth jaringan pada instans DB mungkin tidak cukup untuk kebutuhan komunikasi penyimpanan dari beban kerja saat ini. Hal ini dapat berkontribusi pada latensi penyimpanan yang menyebabkan peningkatan peristiwa `Lock:extend`.

Tindakan

Kami merekomendasikan berbagai tindakan, tergantung pada penyebab peristiwa tunggu Anda.

Topik

- Kurangi penyisipan dan pembaruan konkuren ke relasi yang sama
- Tingkatkan bandwidth jaringan

Kurangi penyisipan dan pembaruan konkuren ke relasi yang sama

Pertama, ketahui apakah terdapat peningkatan pada metrik `tup_inserted` dan `tup_updated` dengan disertai peningkatan pada peristiwa tunggu ini. Jika demikian, periksa relasi yang memiliki pertentangan tinggi untuk operasi penyisipan dan pembaruan. Untuk menentukan hal ini, jalankan kueri tampilan `pg_stat_all_tables` untuk nilai pada bidang `n_tup_ins` dan `n_tup_upd`. Untuk informasi tentang tampilan `pg_stat_all_tables`, lihat pg_stat_all_tables dalam dokumentasi PostgreSQL.

Untuk mendapatkan informasi selengkapnya tentang pemblokiran dan kueri yang diblokir, jalankan kueri `pg_stat_activity` seperti contoh berikut:

```
SELECT
    blocked.pid,
```

```
    blocked.usename,
    blocked.query,
    blocking.pid AS blocking_id,
    blocking.query AS blocking_query,
    blocking.wait_event AS blocking_wait_event,
    blocking.wait_event_type AS blocking_wait_event_type
FROM pg_stat_activity AS blocked
JOIN pg_stat_activity AS blocking ON blocking.pid = ANY(pg_blocking_pids(blocked.pid))
where
blocked.wait_event = 'extend'
and blocked.wait_event_type = 'Lock';

   pid  | usename  |            query             | blocking_id |
            blocking_query                          | blocking_wait_event |
 blocking_wait_event_type
  ------+----------+------------------------------+-------------
+-------------------------------------------------------------------
+---------------------+-------------------------
   7143 |  myuser  | insert into tab1 values (1); |        4600 | INSERT INTO tab1 (a)
 SELECT s FROM generate_series(1,1000000) s; | DataFileExtend      | IO
```

Setelah Anda mengidentifikasi relasi yang berkontribusi pada peningkatan peristiwa `Lock:extend`, gunakan teknik berikut untuk mengurangi pertentangan:

- Cari tahu apakah Anda dapat menggunakan pemartisian untuk mengurangi pertentangan pada tabel yang sama. Memisahkan tuple yang disisipkan atau diperbarui ke dalam partisi yang berbeda dapat mengurangi pertentangan. Untuk informasi tentang partisi, lihat Mengelola partisi PostgreSQL dengan ekstensi pg_partman.
- Jika peristiwa tunggu terutama disebabkan oleh aktivitas pembaruan, pertimbangkan untuk mengurangi nilai faktor pengisian relasi. Hal ini dapat mengurangi permintaan untuk blok baru selama pembaruan. Faktor pengisian adalah parameter penyimpanan untuk tabel yang menentukan jumlah maksimum ruang untuk melakukan packing halaman tabel. Hal ini dinyatakan sebagai persentase dari total ruang untuk sebuah halaman. Untuk informasi selengkapnya tentang parameter fillfactor, lihat CREATE TABLE dalam dokumentasi PostgreSQL.

> **Important**
>
> Kami sangat merekomendasikan untuk menguji sistem jika Anda mengubah faktor pengisian karena mengubah nilai ini dapat berdampak negatif pada performa, tergantung pada beban kerja Anda.

Tingkatkan bandwidth jaringan

Untuk melihat apakah terdapat peningkatan latensi tulis, periksa metrik `WriteLatency` di CloudWatch. Jika ada, gunakan metrik `WriteThroughput` dan `ReadThroughput` Amazon CloudWatch untuk memantau lalu lintas terkait penyimpanan pada klaster DB. Metrik ini dapat membantu Anda menentukan apakah bandwidth jaringan cukup untuk aktivitas penyimpanan beban kerja Anda.

Jika bandwidth jaringan Anda tidak cukup, tingkatkan. Jika instans DB Anda mencapai batas bandwidth jaringan, satu-satunya cara untuk meningkatkan bandwidth adalah meningkatkan ukuran instans DB Anda.

Untuk informasi selengkapnya tentang metrik CloudWatch, lihat Metrik CloudWatch tingkat instans Amazon untuk Amazon RDS. Untuk informasi tentang performa jaringan untuk setiap kelas instans DB, lihat Metrik CloudWatch tingkat instans Amazon untuk Amazon RDS.

Lock:Relation

Peristiwa `Lock:Relation` terjadi saat kueri menunggu untuk memperoleh kunci pada tabel atau tampilan (relasi) yang saat ini dikunci oleh transaksi lain.

Topik

- Versi mesin yang didukung
- Konteks
- Kemungkinan penyebab peningkatan peristiwa tunggu
- Tindakan

Versi mesin yang didukung

Informasi peristiwa tunggu ini didukung untuk semua versi RDS for PostgreSQL.

Konteks

Sebagian besar perintah PostgreSQL secara implisit menggunakan kunci untuk mengontrol akses konkuren ke data dalam tabel. Anda juga dapat menggunakan kunci ini secara eksplisit dalam kode aplikasi Anda dengan perintah `LOCK`. Banyak mode kunci yang tidak kompatibel satu sama lain, dan mode ini dapat memblokir transaksi saat mencoba mengakses objek yang sama. Ketika ini terjadi,

RDS for PostgreSQL akan menghasilkan peristiwa `Lock:Relation`. Berikut adalah beberapa contoh umum:

- Kunci eksklusif seperti `ACCESS EXCLUSIVE` dapat memblokir semua akses konkuren. Operasi bahasa definisi data (DDL) seperti `DROP TABLE`, `TRUNCATE`, `VACUUM FULL`, dan `CLUSTER` memperoleh kunci `ACCESS EXCLUSIVE` secara implisit. `ACCESS EXCLUSIVE` juga merupakan mode kunci default untuk pernyataan `LOCK TABLE` yang tidak menentukan mode secara eksplisit.
- Menggunakan `CREATE INDEX (without CONCURRENT)` pada tabel akan bertentangan dengan pernyataan bahasa manipulasi data (DML) `UPDATE`, `DELETE`, dan `INSERT`, yang memperoleh kunci `ROW EXCLUSIVE`.

Untuk informasi selengkapnya tentang kunci tingkat tabel dan mode kunci yang bertentangan, lihat Explicit Locking dalam dokumentasi PostgreSQL.

Kueri dan transaksi yang memblokir biasanya dapat dibuka blokirnya dengan salah satu cara berikut:

- Kueri yang memblokir – Aplikasi dapat membatalkan kueri atau pengguna dapat mengakhiri proses. Mesin juga dapat memaksa kueri untuk berakhir karena batas waktu pernyataan sesi atau mekanisme deteksi deadlock.
- Transaksi yang memblokir – Transaksi berhenti memblokir saat menjalankan pernyataan `ROLLBACK` atau `COMMIT`. Rollback juga terjadi secara otomatis saat sesi diputus oleh klien atau masalah jaringan, atau diakhiri. Sesi dapat berakhir saat mesin basis data dimatikan, sistem kehabisan memori, dan sebagainya.

Kemungkinan penyebab peningkatan peristiwa tunggu

Saat peristiwa `Lock:Relation` terjadi lebih sering dari biasanya, hal tersebut dapat menunjukkan masalah performa. Penyebab umumnya meliputi yang berikut:

Peningkatan sesi konkuren dengan kunci tabel yang bertentangan

Mungkin ada peningkatan jumlah sesi konkuren dengan kueri yang mengunci tabel yang sama dengan mode kunci yang bertentangan.

Operasi pemeliharaan

Operasi pemeliharaan kondisi seperti `VACUUM` dan `ANALYZE` dapat secara signifikan meningkatkan jumlah kunci yang bertentangan. `VACUUM FULL` memperoleh kunci `ACCESS`

EXCLUSIVE, dan ANALYSE memperoleh kunci SHARE UPDATE EXCLUSIVE. Kedua jenis kunci tersebut dapat menyebabkan peristiwa tunggu Lock:Relation. Operasi pemeliharaan data aplikasi seperti menyegarkan tampilan terwujud juga dapat meningkatkan kueri dan transaksi yang diblokir.

Kunci pada instans pembaca

Mungkin ada pertentangan antara kunci relasi yang dipegang oleh penulis dan pembaca. Saat ini, hanya kunci relasi ACCESS EXCLUSIVE yang direplikasi ke instans pembaca. Namun, kunci relasi ACCESS EXCLUSIVE akan bertentangan dengan kunci relasi ACCESS SHARE yang dipegang oleh pembaca. Hal ini dapat menyebabkan peningkatan peristiwa tunggu relasi kunci pada pembaca.

Tindakan

Kami merekomendasikan berbagai tindakan, tergantung pada penyebab peristiwa tunggu Anda.

Topik

- Kurangi dampak pemblokiran pernyataan SQL
- Minimalkan efek operasi pemeliharaan

Kurangi dampak pemblokiran pernyataan SQL

Untuk mengurangi dampak pemblokiran pernyataan SQL, ubah kode aplikasi Anda jika memungkinkan. Berikut adalah dua teknik umum untuk mengurangi pemblokiran:

- Gunakan opsi NOWAIT – Beberapa perintah SQL, seperti pernyataan SELECT dan LOCK, mendukung opsi ini. Arahan NOWAIT membatalkan kueri permintaan kunci jika kunci tidak dapat segera diperoleh. Teknik ini dapat membantu mencegah sesi yang memblokir menyebabkan penumpukan sesi yang diblokir di belakangnya.

  Misalnya: Transaksi A sedang menunggu kunci yang dipegang oleh transaksi B. Jika B meminta kunci pada tabel yang dikunci oleh transaksi C, transaksi A mungkin diblokir hingga transaksi C selesai. Namun, jika transaksi B menggunakan NOWAIT saat meminta kunci pada C, transaksi ini dapat gagal cepat (fail fast) dan memastikan bahwa transaksi A tidak harus menunggu tanpa batas waktu.
- Gunakan SET lock_timeout – Tetapkan nilai lock_timeout untuk membatasi waktu tunggu pernyataan SQL dalam memperoleh kunci pada relasi. Jika kunci tidak diperoleh dalam batas

waktu yang ditentukan, transaksi yang meminta kunci akan dibatalkan. Tetapkan nilai ini pada tingkat sesi.

Minimalkan efek operasi pemeliharaan

Operasi pemeliharaan seperti `VACUUM` dan `ANALYZE` penting untuk dilakukan. Sebaiknya jangan dinonaktifkan karena peristiwa tunggu `Lock:Relation` berkaitan dengan operasi pemeliharaan ini. Pendekatan berikut dapat meminimalkan efek operasi ini:

- Jalankan operasi pemeliharaan secara manual di luar jam sibuk.
- Untuk mengurangi peristiwa tunggu `Lock:Relation` yang disebabkan oleh tugas autovacuum, lakukan penyetelan autovacuum yang diperlukan. Untuk informasi tentang menyetel autovacuum, lihat Menggunakan autovacuum PostgreSQL di Amazon RDS dalam Panduan Pengguna Amazon RDS.

Lock:transactionid

Peristiwa `Lock:transactionid` terjadi saat transaksi sedang menunggu kunci tingkat baris.

Topik

- Versi mesin yang didukung
- Konteks
- Kemungkinan penyebab peningkatan peristiwa tunggu
- Tindakan

Versi mesin yang didukung

Informasi peristiwa tunggu ini didukung untuk semua versi RDS for PostgreSQL.

Konteks

Peristiwa `Lock:transactionid` terjadi saat transaksi mencoba memperoleh kunci tingkat baris yang telah diberikan untuk transaksi yang sedang berlangsung pada saat bersamaan. Sesi yang menunjukkan peristiwa tunggu `Lock:transactionid` diblokir karena kunci ini. Setelah transaksi yang memblokir berakhir dengan pernyataan `COMMIT` atau `ROLLBACK`, transaksi yang diblokir dapat dilanjutkan.

Semantik kontrol konkurensi multiversi dari RDS for PostgreSQL menjamin bahwa pembaca tidak memblokir penulis dan penulis tidak memblokir pembaca. Agar pertentangan tingkat baris terjadi, transaksi yang memblokir dan diblokir harus mengeluarkan pernyataan yang bertentangan dari jenis berikut:

- `UPDATE`
- `SELECT … FOR UPDATE`
- `SELECT … FOR KEY SHARE`

Pernyataan `SELECT … FOR KEY SHARE` adalah kasus khusus. Basis data menggunakan klausa `FOR KEY SHARE` untuk mengoptimalkan performa integritas referensial. Kunci tingkat baris pada baris dapat memblokir perintah `INSERT`, `UPDATE`, dan `DELETE` pada tabel lain yang mereferensikan baris tersebut.

Kemungkinan penyebab peningkatan peristiwa tunggu

Saat peristiwa ini ditampilkan lebih dari biasanya, penyebabnya biasanya adalah pernyataan `UPDATE`, `SELECT … FOR UPDATE`, atau `SELECT … FOR KEY SHARE` yang dikombinasikan dengan kondisi berikut.

Topik

- Konkurensi tinggi
- Idle pada transaksi
- Transaksi yang berjalan lama

Konkurensi tinggi

Aurora PostgreSQL dapat menggunakan semantik penguncian tingkat baris terperinci. Probabilitas pertentangan tingkat baris meningkat saat kondisi berikut terpenuhi:

- Beberapa beban kerja yang sangat konkuren bersaing untuk baris yang sama.
- Konkurensi meningkat.

Idle pada transaksi

Terkadang kolom `pg_stat_activity.state` menunjukkan nilai `idle in transaction`. Nilai ini ditampilkan untuk sesi yang telah memulai transaksi, tetapi belum mengeluarkan `COMMIT` atau

ROLLBACK. Jika nilai pg_stat_activity.state bukan active, kueri yang ditampilkan pada pg_stat_activity adalah yang paling baru diselesaikan. Sesi yang memblokir tidak secara aktif memproses kueri karena transaksi yang terbuka menahan kunci.

Jika transaksi yang idle memperoleh kunci tingkat baris, transaksi tersebut mungkin mencegah sesi lain memperolehnya. Kondisi ini menyebabkan peristiwa tunggu Lock:transactionid sering terjadi. Untuk mendiagnosis masalah, periksa output dari pg_stat_activity dan pg_locks.

Transaksi yang berjalan lama

Transaksi yang berjalan dalam waktu lama akan mendapatkan kunci untuk waktu lama. Kunci yang lama dipegang ini dapat memblokir transaksi lain sehingga tidak berjalan.

Tindakan

Penguncian baris adalah pertentangan antara pernyataan UPDATE, SELECT … FOR UPDATE, atau SELECT … FOR KEY SHARE. Sebelum mencoba solusi, cari tahu kapan pernyataan ini berjalan pada baris yang sama. Gunakan informasi ini untuk memilih strategi yang dijelaskan pada bagian berikut.

Topik

- Tangani konkurensi tinggi
- Tangani transaksi idle
- Tangani transaksi yang berjalan lama

Tangani konkurensi tinggi

Jika masalahnya adalah konkurensi, coba salah satu teknik berikut:

- Turunkan konkurensi pada aplikasi. Misalnya, kurangi jumlah sesi aktif.
- Terapkan pool koneksi. Untuk mempelajari cara melakukan pooling koneksi dengan Proksi RDS, lihat Menggunakan Proksi Amazon RDS.
- Desain aplikasi atau model data untuk menghindari pertentangan pernyataan UPDATE dan SELECT … FOR UPDATE. Anda juga dapat mengurangi jumlah kunci asing yang diakses oleh pernyataan SELECT … FOR KEY SHARE.

Tangani transaksi idle

Jika `pg_stat_activity.state` menampilkan `idle in transaction`, gunakan strategi berikut:

- Aktifkan autocommit saat memungkinkan. Pendekatan ini mencegah transaksi memblokir transaksi lain sambil menunggu `COMMIT` atau `ROLLBACK`.
- Cari jalur kode yang tidak memiliki `COMMIT`, `ROLLBACK`, atau `END`.
- Pastikan logika penanganan pengecualian pada aplikasi Anda selalu memiliki jalur ke `end of transaction` yang valid.
- Pastikan bahwa aplikasi Anda memproses hasil kueri setelah mengakhiri transaksi dengan `COMMIT` atau `ROLLBACK`.

Tangani transaksi yang berjalan lama

Jika transaksi yang berjalan lama menyebabkan `Lock:transactionid` sering terjadi, coba strategi berikut:

- Cegah kunci baris dari transaksi jangka panjang.
- Batasi panjang kueri dengan menerapkan autocommit jika memungkinkan.

Lock:tuple

Peristiwa `Lock:tuple` terjadi saat proses backend menunggu untuk memperoleh kunci pada tuple.

Topik

Versi mesin yang didukung

Informasi peristiwa tunggu ini didukung untuk semua versi RDS for PostgreSQL.

Konteks

Peristiwa `Lock:tuple` menunjukkan bahwa backend menunggu untuk memperoleh kunci pada tuple, sementara backend lain memegang kunci yang bertentangan pada tuple yang sama. Tabel berikut menggambarkan skenario saat sesi membuat peristiwa `Lock:tuple`.

Wakt	Sesi 1	Sesi 2	Sesi 3
t1	Memulai transaksi.		
t2	Memperbarui baris 1.		
t3		Memperbarui baris 1. Sesi ini mendapatkan kunci eksklusif pada tuple, lalu menunggu sesi 1 untuk melepaskan kunci dengan melakukan komit atau rollback.	
t4			Memperbarui baris 1. Sesi ini menunggu sesi 2 untuk melepaskan kunci eksklusif pada tuple.

Atau Anda dapat mensimulasikan peristiwa tunggu ini dengan menggunakan alat tolok ukur `pgbench`. Konfigurasikan jumlah sesi konkuren yang tinggi untuk memperbarui baris yang sama pada tabel dengan file SQL kustom.

Untuk mempelajari selengkapnya tentang mode kunci yang bertentangan, lihat Explicit Locking dalam dokumentasi PostgreSQL. Untuk mempelajari selengkapnya tentang `pgbench`, lihat pgbench dalam dokumentasi PostgreSQL.

Kemungkinan penyebab peningkatan peristiwa tunggu

Saat peristiwa ini muncul lebih dari biasanya, yang mungkin menunjukkan adanya masalah performa, berikut adalah penyebab umumnya:

- Sejumlah besar sesi konkuren mencoba mendapatkan kunci yang bertentangan untuk tuple yang sama dengan menjalankan pernyataan `UPDATE` atau `DELETE`.

- Beberapa sesi yang sangat konkuren menjalankan pernyataan `SELECT` menggunakan mode kunci `FOR UPDATE` atau `FOR NO KEY UPDATE`.
- Berbagai faktor mendorong aplikasi atau pool koneksi untuk membuka lebih banyak sesi agar menjalankan operasi yang sama. Saat sesi baru mencoba memodifikasi baris yang sama, beban DB dapat melonjak, dan `Lock:tuple` dapat ditampilkan.

Untuk informasi selengkapnya, lihat Row-Level Locks dalam dokumentasi PostgreSQL.

Tindakan

Kami merekomendasikan berbagai tindakan, tergantung pada penyebab peristiwa tunggu Anda.

Topik

- Selidiki logika aplikasi
- Temukan sesi pemblokir
- Kurangi konkurensi saat tinggi
- Pecahkan masalah bottleneck

Selidiki logika aplikasi

Cari tahu apakah sesi pemblokir telah berada pada status `idle in transaction` dalam waktu yang lama. Jika demikian, coba akhiri sesi pemblokir sebagai solusi jangka pendek. Anda dapat menggunakan fungsi `pg_terminate_backend`. Untuk informasi selengkapnya tentang fungsi ini, lihat Server Signaling Functions dalam dokumentasi PostgreSQL.

Untuk solusi jangka panjang, lakukan hal berikut:

- Sesuaikan logika aplikasi.
- Gunakan parameter `idle_in_transaction_session_timeout`. Parameter ini mengakhiri sesi apa pun dengan transaksi terbuka yang telah idle lebih lama dari jumlah waktu yang ditentukan. Untuk informasi selengkapnya, lihat Client Connection Defaults dalam dokumentasi PostgreSQL.
- Gunakan autocommit sebanyak mungkin. Untuk informasi selengkapnya, lihat SET AUTOCOMMIT dalam dokumentasi PostgreSQL.

Temukan sesi pemblokir

Saat peristiwa tunggu `Lock:tuple` terjadi, identifikasi pemblokir dan sesi yang diblokir dengan mencari tahu kunci mana yang bergantung satu sama lain. Untuk informasi selengkapnya, lihat Lock dependency information dalam wiki PostgreSQL.

Contoh berikut mengkueri semua sesi, memfilter `tuple`, dan mengurutkan berdasarkan `wait_time`.

```
SELECT blocked_locks.pid AS blocked_pid,
         blocking_locks.pid AS blocking_pid,
         blocked_activity.usename AS blocked_user,
         blocking_activity.usename AS blocking_user,
         now() - blocked_activity.xact_start AS blocked_transaction_duration,
         now() - blocking_activity.xact_start AS blocking_transaction_duration,
         concat(blocked_activity.wait_event_type,':',blocked_activity.wait_event) AS
 blocked_wait_event,
         concat(blocking_activity.wait_event_type,':',blocking_activity.wait_event) AS
 blocking_wait_event,
         blocked_activity.state AS blocked_state,
         blocking_activity.state AS blocking_state,
         blocked_locks.locktype AS blocked_locktype,
         blocking_locks.locktype AS blocking_locktype,
         blocked_activity.query AS blocked_statement,
         blocking_activity.query AS blocking_statement
    FROM pg_catalog.pg_locks blocked_locks
    JOIN pg_catalog.pg_stat_activity blocked_activity ON blocked_activity.pid =
 blocked_locks.pid
    JOIN pg_catalog.pg_locks blocking_locks
        ON blocking_locks.locktype = blocked_locks.locktype
        AND blocking_locks.DATABASE IS NOT DISTINCT FROM blocked_locks.DATABASE
        AND blocking_locks.relation IS NOT DISTINCT FROM blocked_locks.relation
        AND blocking_locks.page IS NOT DISTINCT FROM blocked_locks.page
        AND blocking_locks.tuple IS NOT DISTINCT FROM blocked_locks.tuple
        AND blocking_locks.virtualxid IS NOT DISTINCT FROM blocked_locks.virtualxid
        AND blocking_locks.transactionid IS NOT DISTINCT FROM
 blocked_locks.transactionid
        AND blocking_locks.classid IS NOT DISTINCT FROM blocked_locks.classid
        AND blocking_locks.objid IS NOT DISTINCT FROM blocked_locks.objid
        AND blocking_locks.objsubid IS NOT DISTINCT FROM blocked_locks.objsubid
        AND blocking_locks.pid != blocked_locks.pid
    JOIN pg_catalog.pg_stat_activity blocking_activity ON blocking_activity.pid =
 blocking_locks.pid
    WHERE NOT blocked_locks.GRANTED;
```

Kurangi konkurensi saat tinggi

Peristiwa `Lock:tuple` mungkin terjadi terus-menerus, terutama pada waktu beban kerja yang sibuk. Dalam situasi ini, pertimbangkan untuk mengurangi konkurensi tinggi untuk baris yang sangat sibuk. Sering kali, hanya beberapa baris mengontrol antrean atau logika Boolean, sehingga membuat baris ini sangat sibuk.

Anda dapat mengurangi konkurensi dengan menggunakan pendekatan yang berbeda berdasarkan kebutuhan bisnis, logika aplikasi, dan jenis beban kerja. Misalnya, Anda dapat melakukan hal berikut:

- Desain ulang tabel dan logika data Anda untuk mengurangi konkurensi tinggi.
- Ubah logika aplikasi untuk mengurangi konkurensi tinggi di tingkat baris.
- Manfaatkan dan desain ulang kueri dengan kunci tingkat baris.
- Gunakan klausa `NOWAIT` dengan operasi percobaan ulang.
- Pertimbangkan untuk menggunakan kontrol konkurensi logika optimistis dan kontrol konkurensi logika penguncian hibrid.
- Pertimbangkan untuk mengubah tingkat isolasi basis data.

Pecahkan masalah bottleneck

`Lock:tuple` dapat terjadi dengan bottleneck seperti "CPU starvation" atau penggunaan maksimum bandwidth Amazon EBS. Untuk mengurangi bottleneck, pertimbangkan pendekatan berikut:

- Naikkan skala jenis kelas instans Anda.
- Optimalkan kueri sarat sumber daya.
- Ubah logika aplikasi.
- Arsipkan data yang jarang diakses.

LWLock:BufferMapping (LWLock:buffer_mapping)

Peristiwa ini terjadi saat sesi menunggu untuk mengaitkan blok data dengan buffer di pool buffer bersama.

> **Note**
>
> Peristiwa ini dinamai `LWLock:BufferMapping` untuk RDS for PostgreSQL versi 13 dan versi yang lebih tinggi. Untuk RDS for PostgreSQL versi 12 dan versi yang lebih lama, peristiwa ini dinamai `LWLock:buffer_mapping`.

Topik

Versi mesin yang didukung

Informasi peristiwa tunggu ini relevan untuk RDS for PostgreSQL versi 9.6 dan lebih tinggi.

Konteks

Pool buffer bersama adalah area memori PostgreSQL yang menampung semua halaman yang sedang atau telah digunakan oleh proses. Ketika suatu proses membutuhkan halaman, proses ini membacakan halaman tersebut ke dalam pool buffer bersama. Parameter `shared_buffers` menetapkan ukuran buffer bersama dan menyediakan area memori untuk menyimpan halaman tabel dan indeks. Jika Anda mengganti parameter ini, pastikan untuk mengaktifkan ulang basis data.

Peristiwa `LWLock:buffer_mapping` tunggu terjadi dalam skenario berikut:

- Sebuah proses mencari tabel buffer untuk halaman dan memperoleh kunci pemetaan buffer bersama.
- Sebuah proses memuat halaman ke dalam pool buffer dan memperoleh kunci pemetaan buffer eksklusif.
- Sebuah proses menghapus halaman dari pool dan memperoleh kunci pemetaan buffer eksklusif.

Penyebab

Ketika peristiwa ini muncul lebih sering dari biasanya, yang mungkin menunjukkan masalah performa, basis data akan melakukan paging masuk dan keluar pada pool buffer bersama. Penyebab umumnya meliputi yang berikut:

- Kueri besar
- Indeks dan tabel yang mengalami bloat
- Pemindaian tabel lengkap
- Ukuran pool bersama yang lebih kecil dari working set

Tindakan

Kami merekomendasikan berbagai tindakan, tergantung pada penyebab peristiwa tunggu Anda.

Topik

- Pantau metrik terkait buffer
- Evaluasi strategi pengindeksan Anda
- Kurangi jumlah buffer yang harus dialokasikan dengan cepat

Pantau metrik terkait buffer

Saat `LWLock:buffer_mapping` menunggu lonjakan, selidiki rasio hit buffer. Anda dapat menggunakan metrik ini untuk mendapatkan pemahaman yang lebih baik tentang apa yang terjadi di cache buffer. Periksa metrik berikut:

`blks_hit`

Metrik penghitung Wawasan Performa ini menunjukkan jumlah blok yang diambil dari pool buffer bersama. Setelah peristiwa tunggu `LWLock:buffer_mapping` muncul, Anda mungkin melihat lonjakan `blks_hit`.

`blks_read`

Metrik penghitung Wawasan Performa ini menunjukkan jumlah blok yang mengharuskan I/O dibacakan ke dalam pool buffer bersama. Anda mungkin melihat lonjakan `blks_read` menjelang peristiwa tunggu `LWLock:buffer_mapping`.

Evaluasi strategi pengindeksan Anda

Untuk mengonfirmasi bahwa strategi pengindeksan Anda tidak menurunkan performa, periksa hal berikut:

Bloat indeks

Pastikan bloat indeks dan tabel tidak menyebabkan halaman yang tidak perlu dibacakan ke buffer bersama. Jika tabel Anda berisi baris yang tidak digunakan, pertimbangkan untuk mengarsipkan datanya dan menghapus baris tersebut dari tabel. Anda kemudian dapat membuat kembali indeks untuk tabel yang diubah ukurannya.

Indeks untuk kueri yang sering digunakan

Untuk menentukan apakah Anda memiliki indeks optimal, pantau metrik mesin DB di Wawasan Performa. Metrik `tup_returned` menunjukkan jumlah baris yang dibaca. Metrik `tup_fetched` menunjukkan jumlah baris yang dikembalikan ke klien. Jika `tup_returned` secara signifikan lebih besar dari `tup_fetched`, data mungkin tidak diindeks dengan benar. Selain itu, statistik tabel Anda mungkin tidak terkini.

Kurangi jumlah buffer yang harus dialokasikan dengan cepat

Untuk mengurangi peristiwa tunggu `LWLock:buffer_mapping`, coba kurangi jumlah buffer yang harus dialokasikan dengan cepat. Salah satu strateginya adalah dengan melakukan operasi batch yang lebih kecil. Anda mungkin dapat memiliki batch yang lebih kecil dengan mempartisi tabel Anda.

LWLock:BufferIO (IPC:BufferIO)

Peristiwa `LWLock:BufferIO` terjadi ketika RDS for PostgreSQL menunggu proses lain untuk menyelesaikan operasi input/output (I/O)-nya ketika secara konkuren mencoba mengakses halaman. Tujuannya adalah agar halaman yang sama dibacakan ke buffer bersama.

Topik

Versi mesin yang relevan

Informasi peristiwa tunggu ini relevan untuk semua versi RDS for PostgreSQL. Untuk RDS for PostgreSQL 12 dan versi sebelumnya peristiwa tunggu ini dinamai lwlock:buffer_io sedangkan di RDS for PostgreSQL versi 13 dinamai lwlock:bufferio. Mulai dari RDS for PostgreSQL versi 14, peristiwa tunggu BufferIO dipindahkan dari jenis peristiwa tunggu `LWLock` ke `IPC` (IPC:BufferIO).

Konteks

Setiap buffer bersama memiliki kunci I/O yang terkait dengan peristiwa tunggu `LWLock:BufferIO`, setiap kali blok (atau halaman) harus diambil di luar pool buffer bersama.

Kunci ini digunakan untuk menangani beberapa sesi yang semuanya memerlukan akses ke blok yang sama. Blok ini harus dibaca dari luar pool buffer bersama, yang ditentukan oleh parameter `shared_buffers`.

Segera setelah halaman dibaca di dalam kumpulan buffer bersama, kunci `LWLock:BufferIO` akan dilepaskan.

> **Note**
>
> Peristiwa tunggu `LWLock:BufferIO` mendahului peristiwa tunggu IO: DataFileRead. Peristiwa tunggu `IO:DataFileRead` terjadi saat data sedang dibaca dari penyimpanan.

Untuk informasi selengkapnya tentang kunci ringan, lihat Gambaran Umum Penguncian.

Penyebab

Penyebab umum peristiwa `LWLock:BufferIO` muncul dalam peristiwa tunggu teratas mencakup yang berikut:

- Beberapa backend atau koneksi mencoba mengakses halaman yang sama yang juga menunggu operasi I/O
- Rasio antara ukuran pool buffer bersama (ditentukan oleh parameter `shared_buffers`) dan jumlah buffer yang dibutuhkan oleh beban kerja saat ini
- Ukuran pool buffer bersama tidak seimbang dengan jumlah halaman yang dikosongkan oleh operasi lain
- Indeks besar atau bloat yang mengharuskan mesin membacakan lebih banyak halaman daripada yang diperlukan ke dalam pool buffer bersama

- Kurangnya indeks yang memaksa mesin DB untuk membaca lebih banyak halaman dari tabel daripada yang diperlukan
- Checkpoint terjadi terlalu sering atau perlu melakukan flushing terlalu banyak halaman yang dimodifikasi
- Lonjakan tiba-tiba untuk koneksi basis data yang mencoba melakukan operasi pada halaman yang sama

Tindakan

Kami merekomendasikan berbagai tindakan tergantung pada penyebab peristiwa tunggu Anda:

- Amati metrik Amazon CloudWatch untuk korelasi antara penurunan tajam pada peristiwa tunggu `BufferCacheHitRatio` dan `LWLock:BufferIO`. Efek ini dapat berarti bahwa Anda memiliki pengaturan buffer bersama kecil. Anda mungkin perlu meningkatkan atau menaikkan skala kelas instans DB Anda. Anda dapat membagi beban kerja Anda menjadi lebih banyak simpul pembaca.
- Setel `max_wal_size` dan `checkpoint_timeout` berdasarkan waktu puncak beban kerja Anda jika Anda melihat `LWLock:BufferIO` bertepatan dengan penurunan metrik `BufferCacheHitRatio`. Kemudian, identifikasi kueri mana yang mungkin menyebabkannya.
- Verifikasi apakah Anda memiliki indeks yang tidak digunakan, lalu hapus.
- Gunakan tabel yang dipartisi (yang juga memiliki indeks yang dipartisi). Dengan melakukan hal ini, Anda dapat menjaga penyusunan ulang indeks tetap rendah dan mengurangi dampaknya.
- Hindari kolom pengindeksan yang tidak perlu.
- Cegah lonjakan koneksi basis data yang tiba-tiba dengan menggunakan pool koneksi.
- Batasi jumlah maksimum koneksi ke basis data sebagai praktik terbaik.

LWLock:buffer_content (BufferContent)

Peristiwa `LWLock:buffer_content` terjadi saat suatu sesi menunggu untuk membaca atau menulis halaman data di memori sementara sesi lain mengunci halaman tersebut untuk penulisan. Di RDS for PostgreSQL 13 dan yang lebih tinggi, peristiwa tunggu ini disebut `BufferContent`.

Topik

- Tindakan

Versi mesin yang didukung

Informasi peristiwa tunggu ini didukung untuk semua versi RDS for PostgreSQL.

Konteks

Untuk membaca atau memanipulasi data, PostgreSQL mengaksesnya melalui buffer memori bersama. Untuk membaca dari buffer, proses mendapatkan kunci ringan (LWLock) pada konten buffer dalam mode bersama. Untuk menulis ke buffer, proses ini mendapatkan kunci tersebut dalam mode eksklusif. Kunci bersama memungkinkan proses lain untuk secara konkuren memperoleh kunci bersama pada konten tersebut. Kunci eksklusif mencegah proses lain mendapatkan jenis kunci apa pun.

Peristiwa `LWLock:buffer_content` (`BufferContent`) menunjukkan bahwa beberapa proses mencoba mendapatkan kunci pada konten buffer tertentu.

Kemungkinan penyebab peningkatan peristiwa tunggu

Saat peristiwa `LWLock:buffer_content` (`BufferContent`) muncul lebih dari biasanya, yang mungkin menunjukkan adanya masalah performa, berikut adalah penyebab umumnya:

Peningkatan pembaruan konkuren ke data yang sama

Mungkin ada peningkatan jumlah sesi konkuren dengan kueri yang memperbarui konten buffer yang sama. Pertentangan ini bisa lebih terlihat pada tabel dengan banyak indeks.

Data beban kerja tidak ada dalam memori

Saat data yang diproses oleh beban kerja aktif tidak ada dalam memori, peristiwa tunggu ini dapat meningkat. Efek ini terjadi karena proses yang memegang kunci dapat mempertahankannya lebih lama saat melakukan operasi I/O disk.

Penggunaan batasan kunci asing yang berlebihan

Batasan kunci asing dapat meningkatkan jumlah waktu saat sebuah proses memegang kunci konten buffer. Efek ini terjadi karena operasi baca memerlukan kunci konten buffer bersama pada kunci yang direferensikan saat kunci tersebut diperbarui.

Tindakan

Kami merekomendasikan berbagai tindakan, tergantung pada penyebab peristiwa tunggu Anda. Anda dapat mengidentifikasi peristiwa `LWLock:buffer_content` (`BufferContent`) dengan menggunakan Wawasan Performa Amazon RDS atau dengan mengkueri tampilan `pg_stat_activity`.

Topik

- Meningkatkan efisiensi dalam memori
- Kurangi penggunaan batasan kunci asing
- Hapus indeks yang tidak digunakan
- Tingkatkan ukuran cache saat menggunakan urutan

Meningkatkan efisiensi dalam memori

Untuk meningkatkan kemungkinan data beban kerja aktif ada di memori, partisi tabel atau naikkan skala kelas instans Anda. Untuk informasi tentang kelas instans DB, lihat Kelas instans DB .

Kurangi penggunaan batasan kunci asing

Selidiki beban kerja yang mengalami jumlah peristiwa tunggu `LWLock:buffer_content` (`BufferContent`) yang tinggi untuk penggunaan batasan kunci asing. Hapus batasan kunci asing yang tidak perlu.

Hapus indeks yang tidak digunakan

Untuk beban kerja yang mengalami jumlah peristiwa tunggu `LWLock:buffer_content` (`BufferContent`) yang tinggi, identifikasi indeks yang tidak digunakan, lalu hapus.

Tingkatkan ukuran cache saat menggunakan urutan

Jika tabel Anda menggunakan urutan, tingkatkan ukuran cache untuk menghapus pertentangan pada halaman urutan dan halaman indeks. Setiap urutan adalah satu halaman dalam memori bersama. Cache yang telah ditentukan adalah per koneksi. Ini mungkin tidak cukup untuk menangani beban kerja ketika banyak sesi konkuren mendapatkan nilai urutan.

LWLock:lock_manager (LWLock:lockmanager)

Peristiwa ini terjadi saat mesin RDS for PostgreSQL mempertahankan area memori kunci bersama untuk mengalokasikan, memeriksa, dan mendealokasikan kunci saat kunci jalur cepat tidak memungkinkan.

Topik

Versi mesin yang didukung

Informasi peristiwa tunggu ini relevan untuk RDS for PostgreSQL versi 9.6 dan lebih tinggi.
Untuk rilis RDS for PostgreSQL yang lebih lama dari versi 13, nama peristiwa tunggu ini adalah `LWLock:lock_manager`. Untuk RDS for PostgreSQL versi 13 dan yang lebih tinggi, nama peristiwa tunggu ini adalah `LWLock:lockmanager`.

Konteks

Saat Anda mengeluarkan pernyataan SQL, RDS for PostgreSQL mencatat kunci untuk melindungi struktur, data, dan integritas basis data Anda selama operasi konkuren. Mesin dapat mencapai tujuan ini menggunakan kunci jalur cepat atau kunci jalur yang tidak cepat. Kunci jalur yang tidak cepat lebih mahal dan menghasilkan lebih banyak overhead daripada kunci jalur cepat.

Penguncian jalur cepat

Untuk mengurangi overhead kunci yang sering diambil dan dilepaskan, tetapi jarang bertentangan, proses backend dapat menggunakan penguncian jalur cepat. Basis data menggunakan mekanisme ini untuk kunci yang memenuhi kriteria berikut:

- Kunci tersebut menggunakan metode kunci DEFAULT.
- Kunci tersebut merepresentasikan kunci pada relasi basis data, bukan relasi bersama.
- Kunci tersebut adalah kunci lemah yang tidak mungkin bertentangan.
- Mesin dapat dengan cepat memverifikasi bahwa tidak ada kunci yang mungkin dapat bertentangan.

Mesin tidak dapat menggunakan penguncian jalur cepat jika salah satu kondisi berikut ini berlaku:

- Kunci tidak memenuhi kriteria di atas.
- Tidak ada lagi slot yang tersedia untuk proses backend.

Untuk menyetel kueri Anda untuk penguncian jalur cepat, Anda dapat menggunakan kueri berikut.

```
SELECT count(*), pid, mode, fastpath
  FROM pg_locks
 WHERE fastpath IS NOT NULL
 GROUP BY 4,3,2
 ORDER BY pid, mode;
 count | pid  |      mode       | fastpath
-------+------+-----------------+----------
16 | 9185 | AccessShareLock | t
336 | 9185 | AccessShareLock | f
1 | 9185 | ExclusiveLock   | t
```

Kueri berikut hanya menunjukkan total di seluruh basis data.

```
SELECT count(*), mode, fastpath
  FROM pg_locks
 WHERE fastpath IS NOT NULL
 GROUP BY 3,2
 ORDER BY mode,1;
count |      mode       | fastpath
-------+-----------------+----------
16 | AccessShareLock | t
337 | AccessShareLock | f
1 | ExclusiveLock   | t
(3 rows)
```

Untuk informasi selengkapnya tentang penguncian jalur cepat, lihat fast path dalam README pengelola kunci PostgreSQL dan pg-locks dalam dokumentasi PostgreSQL.

Contoh masalah penskalaan untuk pengelola kunci

Dalam contoh ini, tabel bernama `purchases` menyimpan data dari rentang waktu lima tahun, yang dipartisi berdasarkan hari. Setiap partisi memiliki dua indeks. Urutan peristiwa berikut terjadi:

1. Anda mengkueri data dari rentang waktu beberapa hari, yang mengharuskan basis data untuk membaca banyak partisi.
2. Basis data membuat entri kunci untuk setiap partisi. Jika indeks partisi adalah bagian dari jalur akses pengoptimisasi, basis data juga akan membuat entri kunci untuk indeks tersebut.
3. Saat jumlah entri kunci yang diminta untuk proses backend yang sama lebih tinggi dari 16, yang merupakan nilai `FP_LOCK_SLOTS_PER_BACKEND`, pengelola kunci menggunakan metode kunci jalur yang tidak cepat.

Aplikasi modern mungkin memiliki ratusan sesi. Jika sesi konkuren mengkueri induk tanpa pemangkasan partisi yang tepat, basis data mungkin membuat ratusan atau bahkan ribuan kunci jalur yang tidak cepat. Biasanya, saat konkurensi ini lebih tinggi dari jumlah vCPU, peristiwa tunggu `LWLock:lock_manager` akan ditampilkan.

> **Note**
>
> Peristiwa tunggu `LWLock:lock_manager` tidak terkait dengan jumlah partisi atau indeks dalam skema basis data. Sebagai gantinya, hal ini terkait dengan jumlah kunci jalur yang tidak cepat yang harus dikontrol oleh basis data.

Kemungkinan penyebab peningkatan peristiwa tunggu

Saat peristiwa tunggu `LWLock:lock_manager` terjadi lebih sering dari biasanya, yang mungkin menunjukkan masalah performa, kemungkinan penyebab lonjakan yang mendadak ini adalah sebagai berikut:

- Sesi aktif konkuren menjalankan kueri yang tidak menggunakan kunci jalur cepat. Sesi ini juga melebihi vCPU maksimum.
- Sejumlah besar sesi aktif konkuren mengakses tabel yang memiliki banyak partisi. Setiap partisi memiliki beberapa indeks.
- Basis data mengalami badai koneksi. Secara default, beberapa aplikasi dan perangkat lunak pool koneksi akan membuat lebih banyak koneksi ketika basis data lambat. Praktik ini memperburuk masalahnya. Setel perangkat lunak pool koneksi Anda sehingga badai koneksi tidak terjadi.
- Sejumlah besar sesi mengkueri tabel induk tanpa memangkas partisi.
- Bahasa definisi data (DL), bahasa manipulasi data (DL), atau perintah pemeliharaan secara khusus mengunci relasi sibuk atau tuple yang sering diakses atau dimodifikasi.

Tindakan

Jika peristiwa tunggu CPU terjadi, hal ini tidak selalu menunjukkan masalah performa. Tanggapi peristiwa ini hanya ketika performa menurun dan peristiwa tunggu ini mendominasi beban DB.

Topik

- Gunakan pemangkasan partisi
- Hapus indeks yang tidak perlu
- Setel kueri Anda untuk penguncian jalur cepat
- Setel peristiwa tunggu lainnya
- Kurangi bottleneck perangkat keras
- Gunakan pooler koneksi
- Upgrade versi RDS for PostgreSQL Anda

Gunakan pemangkasan partisi

Pemangkasan partisi adalah strategi optimisasi kueri untuk tabel yang dipartisi secara deklaratif yang mengecualikan partisi yang tidak dibutuhkan dari pemindaian tabel, sehingga meningkatkan performa. Pemangkasan partisi diaktifkan secara default. Jika dinonaktifkan, aktifkan sebagai berikut.

```
SET enable_partition_pruning = on;
```

Kueri dapat memanfaatkan pemangkasan partisi ketika klausa `WHERE`-nya berisi kolom yang digunakan untuk pembuatan partisi. Untuk informasi selengkapnya, lihat Partition Pruning dalam dokumentasi PostgreSQL.

Hapus indeks yang tidak perlu

Basis data Anda mungkin berisi indeks yang tidak digunakan atau jarang digunakan. Jika demikian, pertimbangkan untuk menghapusnya. Lakukan salah satu dari langkah berikut:

- Pelajari cara menemukan indeks yang tidak perlu dengan membaca Indeks yang Tidak Digunakan di wiki PostgreSQL.
- Jalankan PG Collector. Skrip SQL ini mengumpulkan informasi basis data dan menyajikannya dalam laporan HTML terkonsolidasi. Periksa bagian “Indeks yang tidak digunakan”. Untuk informasi selengkapnya, lihat pg-collector dalam Repositori GitHub Lab AWS.

Setel kueri Anda untuk penguncian jalur cepat

Untuk mengetahui apakah kueri Anda menggunakan penguncian jalur cepat, buat kueri pada kolom `fastpath` dalam tabel `pg_locks`. Jika kueri Anda tidak menggunakan penguncian jalur cepat, coba kurangi jumlah relasi per kueri menjadi kurang dari 16.

Setel peristiwa tunggu lainnya

Jika `LWLock:lock_manager` adalah yang pertama atau kedua dalam daftar peristiwa tunggu teratas, periksa apakah peristiwa tunggu berikut juga ditampilkan pada daftar ini:

- `Lock:Relation`
- `Lock:transactionid`
- `Lock:tuple`

Jika peristiwa di atas ditampilkan pada posisi tinggi dalam daftar, pertimbangkan untuk menyetel peristiwa tunggu ini terlebih dahulu. Peristiwa ini dapat menjadi pendorong untuk `LWLock:lock_manager`.

Kurangi bottleneck perangkat keras

Anda mungkin memiliki bottleneck perangkat keras, seperti kelaparan CPU atau penggunaan maksimum bandwidth Amazon EBS Anda. Jika demikian, maka pertimbangkan untuk mengurangi bottleneck perangkat keras. Pertimbangkan tindakan berikut:

- Naikkan kelas instans Anda.
- Optimalkan kueri yang mengonsumsi CPU dan memori dalam jumlah besar.
- Ubah logika aplikasi Anda.
- Arsipkan data Anda.

Untuk informasi selengkapnya tentang CPU, memori, dan bandwidth jaringan EBS, lihat Jenis Instans Amazon RDS.

Gunakan pooler koneksi

Jika jumlah total koneksi aktif Anda melebihi vCPU maksimum, lebih banyak proses OS akan memerlukan CPU yang melampaui kapasitas yang dapat didukung oleh jenis instans Anda.

Jika demikian, pertimbangkan untuk menggunakan atau menyetel pool koneksi. Untuk informasi selengkapnya tentang vCPU untuk jenis instans Anda, lihat Jenis Instans Amazon RDS.

Untuk informasi selengkapnya tentang pooling koneksi, lihat sumber daya berikut:

- Menggunakan Proksi Amazon RDS
- pgbouncer
- Connection Pools and Data Sources dalam dokumentasi PostgreSQL

Upgrade versi RDS for PostgreSQL Anda

Jika versi RDS for PostgreSQL Anda saat ini lebih rendah dari 12, upgrade ke versi 12 atau lebih tinggi. PostgreSQL versi 12 dan yang lebih baru memiliki mekanisme partisi yang ditingkatkan. Untuk informasi selengkapnya tentang versi 12, lihat Catatan Rilis PostgreSQL 12.0. Untuk informasi selengkapnya tentang meng-upgrade RDS for PostgreSQL, lihat Meningkatkan mesin DB PostgreSQL untuk Amazon RDS.

Timeout:PgSleep

Peristiwa `Timeout:PgSleep` terjadi saat proses server telah memanggil fungsi `pg_sleep` dan menunggu batas waktu tidur berakhir.

Topik

- Versi mesin yang didukung
- Kemungkinan penyebab peningkatan peristiwa tunggu
- Tindakan

Versi mesin yang didukung

Informasi peristiwa tunggu ini didukung untuk semua versi RDS for PostgreSQL.

Kemungkinan penyebab peningkatan peristiwa tunggu

Peristiwa tunggu ini terjadi saat aplikasi, fungsi tersimpan, atau pengguna mengeluarkan pernyataan SQL yang memanggil salah satu fungsi berikut:

- `pg_sleep`
- `pg_sleep_for`

- `pg_sleep_until`

Fungsi tersebut menunda eksekusi sampai jumlah detik yang ditentukan telah berlalu. Misalnya, `SELECT pg_sleep(1)` melakukan penundaan selama 1 detik. Untuk informasi selengkapnya, lihat Delaying Execution dalam dokumentasi PostgreSQL.

Tindakan

Identifikasi pernyataan yang menjalankan fungsi `pg_sleep`. Tentukan apakah penggunaan fungsi tersebut sesuai.

Timeout:VacuumDelay

Peristiwa `Timeout:VacuumDelay` menunjukkan bahwa batas biaya untuk I/O vakum telah terlampaui dan bahwa proses vakum telah dibuat tidur. Operasi vakum berhenti selama durasi yang ditentukan dalam parameter penundaan biaya masing-masing lalu melanjutkan pekerjaannya. Untuk perintah vakum manual, penundaan ditentukan dalam parameter `vacuum_cost_delay`. Untuk daemon autovacuum, penundaan ditentukan dalam `autovacuum_vacuum_cost_delay parameter.`

Topik

Versi mesin yang didukung

Informasi peristiwa tunggu ini didukung untuk semua versi RDS for PostgreSQL.

Konteks

PostgreSQL memiliki daemon autovacuum dan perintah vakum manual. Proses autovacuum “aktif” secara default untuk instans DB RDS for PostgreSQL. Perintah vakum manual digunakan sesuai kebutuhan, misalnya, untuk melakukan purging tabel tuple mati atau menghasilkan statistik baru.

Saat pemvakuman sedang berlangsung, PostgreSQL menggunakan penghitung internal untuk melacak perkiraan biaya seiring sistem melakukan berbagai operasi I/O. Ketika penghitung mencapai

nilai yang ditentukan oleh parameter batas biaya, proses yang melakukan operasi akan tidur selama durasi singkat yang ditentukan dalam parameter penundaan biaya. Kemudian, proses ini akan mengatur ulang penghitung dan melanjutkan operasi.

Proses vakum memiliki parameter yang dapat digunakan untuk mengatur konsumsi sumber daya. Autovacuum dan perintah vakum manual memiliki parameter sendiri untuk menetapkan nilai batas biaya. Keduanya juga memiliki parameter sendiri untuk menentukan penundaan biaya, yaitu jumlah waktu untuk membuat proses vakum menjadi tidur ketika batasnya tercapai. Dengan cara ini, parameter penundaan biaya berfungsi sebagai mekanisme throttling untuk konsumsi sumber daya. Dalam daftar berikut, Anda dapat menemukan deskripsi parameter ini.

Parameter yang mempengaruhi throttling daemon autovacuum

- `autovacuum_vacuum_cost_limit` – Menentukan nilai batas biaya yang akan digunakan dalam operasi vakum otomatis. Jika pengaturan untuk parameter ini ditingkatkan, proses vakum dapat menggunakan lebih banyak sumber daya dan peristiwa tunggu `Timeout:VacuumDelay` akan berkurang.
- `autovacuum_vacuum_cost_delay` – Menentukan nilai penundaan biaya yang akan digunakan dalam operasi vakum otomatis. Nilai default adalah 2 milidetik. Mengatur parameter penundaan ke 0 akan menonaktifkan mekanisme throttling dan dengan demikian, peristiwa tunggu `Timeout:VacuumDelay` tidak akan muncul.

Untuk informasi selengkapnya, lihat Automatic Vacuuming dalam dokumentasi PostgreSQL.

Parameter yang memengaruhi throttling proses vakum manual

- `vacuum_cost_limit` – Ambang batas yang membuat proses pemvakuman menjadi tidur. Secara default, batasnya adalah 200. Angka ini merepresentasikan perkiraan biaya terakumulasi untuk I/O tambahan yang dibutuhkan oleh berbagai sumber daya. Meningkatkan nilai ini akan mengurangi jumlah peristiwa tunggu `Timeout:VacuumDelay`.
- `vacuum_cost_delay` – Jumlah waktu proses vakum tidur ketika batas biaya vakum telah tercapai. Pengaturan default adalah 0, yang berarti fitur ini tidak aktif. Anda dapat mengaturnya ke nilai bilangan bulat untuk menentukan jumlah milidetik untuk mengaktifkan fitur ini, tetapi kami sarankan Anda membiarkannya di pengaturan default.

Untuk informasi selengkapnya tentang parameter `vacuum_cost_delay`, lihat Resource Consumption dalam dokumentasi PostgreSQL.

Untuk mempelajari selengkapnya tentang cara mengonfigurasi dan menggunakan autovacuum dengan RDS for PostgreSQL, lihat Bekerja dengan fitur autovacuum PostgreSQL di Amazon RDS for PostgreSQL.

Kemungkinan penyebab peningkatan peristiwa tunggu

`Timeout:VacuumDelay` dipengaruhi oleh keseimbangan antara pengaturan parameter batas biaya (`vacuum_cost_limit`, `autovacuum_vacuum_cost_limit`) dan parameter penundaan biaya (`vacuum_cost_delay`, `autovacuum_vacuum_cost_delay`) yang mengontrol durasi tidur vakum. Meningkatkan nilai parameter batas biaya akan memungkinkan lebih banyak sumber daya digunakan oleh vakum sebelum dibuat tidur. Tindakan ini akan menghasilkan lebih sedikit peristiwa tunggu `Timeout:VacuumDelay`. Meningkatkan salah satu parameter penundaan akan menyebabkan peristiwa tunggu `Timeout:VacuumDelay` terjadi lebih sering dan berlangsung lebih lama.

Pengaturan parameter `autovacuum_max_workers` juga dapat meningkatkan jumlah `Timeout:VacuumDelay`. Setiap proses pekerja autovacuum tambahan berkontribusi pada mekanisme penghitung internal, dan dengan demikian batasnya dapat tercapai lebih cepat daripada dengan proses pekerja autovacuum tunggal. Karena batas biaya tercapai lebih cepat, penundaan biaya diberlakukan lebih sering, sehingga menghasilkan lebih banyak peristiwa tunggu `Timeout:VacuumDelay`. Untuk informasi selengkapnya, lihat autovacuum_max_workers dalam dokumentasi PostgreSQL.

Objek besar, seperti 500 GB atau lebih, juga meningkatkan peristiwa tunggu ini karena vakum dapat menghabiskan waktu lama untuk menyelesaikan pemrosesan objek besar.

Tindakan

Jika operasi vakum selesai seperti yang diharapkan, tidak diperlukan remediasi. Dengan kata lain, peristiwa tunggu ini belum tentu menunjukkan masalah. Ini menunjukkan bahwa vakum sedang dibuat tidur selama periode waktu yang ditentukan dalam parameter penundaan sehingga sumber daya dapat diterapkan pada proses lain yang perlu diselesaikan.

Jika Anda ingin operasi vakum selesai lebih cepat, Anda dapat menurunkan parameter penundaan. Tindakan ini akan mempersingkat waktu tidur vakum.

Menyeetel RDS for PostgreSQL dengan wawasan proaktif Amazon DevOps Guru

Wawasan proaktif DevOps Guru mendeteksi kondisi pada RDS Anda untuk instans DB RDS for PostgreSQL yang dapat menyebabkan masalah, dan memberi tahu Anda tentangnya sebelum terjadi. DevOps Guru dapat melakukan hal berikut:

- Mencegah banyak masalah umum pada basis data dengan memeriksa silang konfigurasi basis data terhadap pengaturan umum yang direkomendasikan.
- Memberi tahu Anda tentang masalah kritis dalam armada yang, jika dibiarkan tanpa diperiksa, dapat menyebabkan masalah yang lebih besar di kemudian hari.
- Memberi tahu Anda tentang masalah yang baru ditemukan.

Setiap wawasan proaktif berisi analisis penyebab masalah dan rekomendasi untuk tindakan korektif.

Topik

- Basis data telah lama berjalan idle dalam koneksi transaksi

Basis data telah lama berjalan idle dalam koneksi transaksi

Koneksi ke basis berada dalam status `idle in transaction` selama lebih dari 1.800 detik.

Topik

- Versi mesin yang didukung
- Konteks
- Kemungkinan penyebab masalah ini
- Tindakan
- Metrik terkait

Versi mesin yang didukung

Informasi wawasan ini didukung untuk semua versi RDS for PostgreSQL.

Konteks

Transaksi dalam status `idle in transaction` dapat menahan kunci yang memblokir kueri lain. Ini juga dapat mencegah `VACUUM` (termasuk autovacuum) membersihkan baris mati, yang menyebabkan penggelembungan indeks atau tabel atau ID transaksi tumpang tindih.

Kemungkinan penyebab masalah ini

Transaksi yang dimulai dalam sesi interaktif dengan BEGIN atau START TRANSACTION belum berakhir dengan menggunakan perintah COMMIT, ROLLBACK, atau END. Hal ini menyebabkan status transaksi berubah ke `idle in transaction`.

Tindakan

Anda dapat menemukan transaksi idle dengan mengkueri `pg_stat_activity`.

Di klien SQL Anda, jalankan kueri berikut untuk mencantumkan semua koneksi dalam status `idle in transaction` dan mengurutkannya berdasarkan durasi:

```
SELECT now() - state_change as idle_in_transaction_duration, now() - xact_start as
 xact_duration,*
FROM  pg_stat_activity
WHERE state  = 'idle in transaction'
AND   xact_start is not null
ORDER BY 1 DESC;
```

Kami merekomendasikan tindakan yang berbeda bergantung pada penyebab wawasan Anda.

Topik

- Transaksi akhir
- Mengakhiri koneksi
- Konfigurasikan parameter idle_in_transaction_session_timeout
- Memeriksa status AUTOCOMMIT
- Memeriksa logika transaksi dalam kode aplikasi Anda

Transaksi akhir

Ketika Anda memulai transaksi dalam sesi interaktif dengan BEGIN atau START TRANSACTION, status akan berubah ke `idle in transaction`. Status ini tidak akan berubah hingga Anda

mengakhiri transaksi dengan mengeluarkan perintah COMMIT, ROLLBACK, END atau memutuskan koneksi sepenuhnya untuk meluncurkan kembali transaksi.

Mengakhiri koneksi

Akhiri koneksi dengan transaksi idle menggunakan kueri berikut:

```
SELECT pg_terminate_backend(pid);
```

pid adalah ID proses koneksi.

Konfigurasikan parameter idle_in_transaction_session_timeout

Konfigurasikan parameter `idle_in_transaction_session_timeout` di grup parameter baru. Dengan mengonfigurasi parameter ini, intervensi manual tidak diperlukan untuk mengakhiri status idle yang lama dalam transaksi. Untuk informasi lebih lanjut tentang parameter ini, lihat dokumentasi PostgreSQL.

Pesan berikut akan dilaporkan dalam file log PostgreSQL setelah koneksi dihentikan jika transaksi berada dalam status idle_in_transaction yang lebih lama dari waktu yang ditentukan.

```
FATAL: terminating connection due to idle in transaction timeout
```

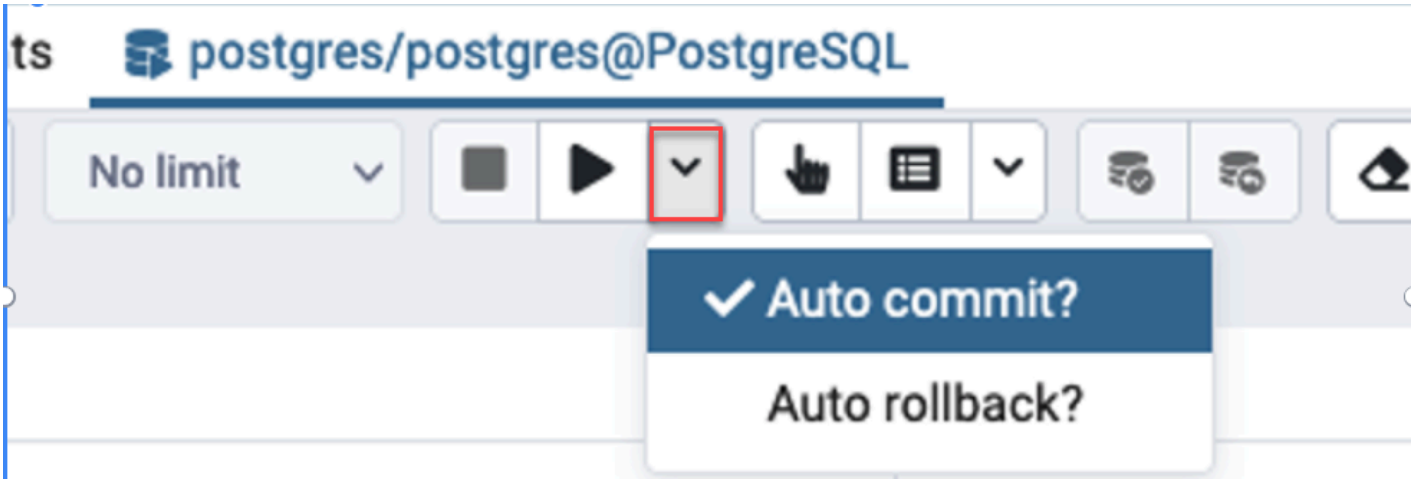
Memeriksa status AUTOCOMMIT

AUTOCOMMIT diaktifkan secara default. Namun, jika dinonaktifkan secara tidak sengaja di klien, pastikan Anda mengaktifkannya kembali.

- Di klien psql Anda, jalankan perintah berikut:

  ```
  postgres=> \set AUTOCOMMIT on
  ```

- Di pgadmin, aktifkan dengan memilih opsi AUTOCOMMIT dari tanda panah bawah.



Memeriksa logika transaksi dalam kode aplikasi Anda

Selidiki logika aplikasi Anda untuk kemungkinan masalah. Pertimbangkan tindakan berikut:

- Periksa apakah JDBC auto commit diatur ke true dalam aplikasi Anda. Pertimbangkan juga untuk menggunakan perintah `COMMIT` eksplisit dalam kode Anda.
- Periksa logika penanganan kesalahan untuk mengetahui apakah transaksi ditutup setelah kesalahan terjadi.
- Periksa apakah aplikasi Anda membutuhkan waktu lama untuk memproses baris yang ditampilkan oleh kueri saat transaksi terbuka. Jika demikian, pertimbangkan untuk mengodekan aplikasi guna menutup transaksi sebelum memproses baris.
- Periksa apakah transaksi berisi banyak operasi yang berjalan lama. Jika demikian, bagi satu transaksi menjadi beberapa transaksi.

Metrik terkait

Metrik PI berikut terkait dengan wawasan ini:

- idle_in_transaction_count - Jumlah sesi dalam status `idle in transaction`.
- idle_in_transaction_max_time - Durasi transaksi yang berjalan paling lama dalam status `idle in transaction`.

Menggunakan ekstensi PostgreSQL dengan Amazon RDS for PostgreSQL

Anda dapat memperluas fungsionalitas PostgreSQL dengan menginstal berbagai ekstensi dan modul. Misalnya, untuk bekerja dengan data spasial Anda dapat menginstal dan menggunakan ekstensi PostGIS. Untuk informasi selengkapnya, lihat Mengelola data spasial dengan ekstensi PostGIS. Sebagai contoh lain, jika Anda ingin meningkatkan entri data untuk tabel yang sangat besar, Anda dapat mempertimbangkan untuk mempartisi data Anda dengan menggunakan ekstensi `pg_partman`. Untuk mempelajari selengkapnya, lihat Mengelola partisi PostgreSQL dengan ekstensi pg_partman.

> **Note**
>
> Pada RDS for PostgreSQL 14.5, RDS for PostgreSQL mendukung Trusted Language Extensions for PostgreSQL. Fitur ini diimplementasikan sebagai ekstensi `pg_tle`, yang dapat Anda tambahkan ke instans DB RDS for PostgreSQL Anda. Dengan menggunakan ekstensi ini, pengembang dapat membuat ekstensi PostgreSQL mereka sendiri di lingkungan yang aman yang menyederhanakan persyaratan penyiapan dan konfigurasi. Untuk informasi selengkapnya, lihat Bekerja dengan Ekstensi Bahasa Tepercaya untuk PostgreSQL.

Dalam beberapa kasus, daripada menginstal ekstensi, Anda dapat menambahkan modul tertentu ke daftar `shared_preload_libraries` dalam grup parameter DB khusus instans DB RDS For PostgreSQL. Biasanya, grup parameter klaster DB default hanya memuat `pg_stat_statements`, tetapi beberapa modul lain tersedia untuk ditambahkan ke daftar. Misalnya, Anda dapat menambahkan kemampuan penjadwalan dengan menambahkan modul `pg_cron`, seperti yang dijelaskan dalam Menjadwalkan pemeliharaan dengan ekstensi pg_cron PostgreSQL. Sebagai contoh lain, Anda dapat men-log rencana eksekusi kueri dengan memuat modul `auto_explain`. Untuk mempelajari lebih lanjut, lihat Mencatat rencana eksekusi kueri di pusat AWS pengetahuan.

Bergantung pada versi RDS for PostgreSQL Anda, menginstall ekstensi mungkin memerlukan izin `rds_superuser`, sebagai berikut:

- Untuk RDS for PostgreSQL versi 12 dan versi sebelumnya, menginstal ekstensi yang memerlukan hak istimewa `rds_superuser`.

- Untuk RDS for PostgreSQL versi 13 dan versi yang lebih tinggi, pengguna (peran) dengan izin membuat pada instans basis data tertentu yang dapat menginstal dan menggunakan ekstensi tepercaya apa pun. Untuk daftar ekstensi tepercaya, lihat Ekstensi terpercaya PostgreSQL.

Anda juga dapat menentukan dengan tepat ekstensi mana yang dapat diinstal pada instans DB RDS for PostgreSQL, dengan mencantumkannya dalam parameter `rds.allowed_extensions`. Untuk informasi selengkapnya, lihat Membatasi penginstalan ekstensi PostgreSQL.

Untuk mempelajari lebih lanjut tentang peran `rds_superuser` tersebut, lihat Memahami peran dan izin PostgreSQL.

Topik

- Menggunakan fungsi dari ekstensi orafce
- Mengelola partisi PostgreSQL dengan ekstensi pg_partman
- Menggunakan pgAudit untuk membuat log aktivitas basis data
- Menjadwalkan pemeliharaan dengan ekstensi pg_cron PostgreSQL
- Menggunakan pglogical untuk menyinkronkan data di seluruh instans
- Menggunakan pgactive untuk mendukung replikasi aktif-aktif
- Mengurangi bloat dalam tabel dan indeks dengan ekstensi pg_repack
- Meningkatkan dan menggunakan ekstensi PLV8
- Menggunakan PL/Rust untuk menulis fungsi PostgreSQL dalam bahasa Rust
- Mengelola data spasial dengan ekstensi PostGIS

Menggunakan fungsi dari ekstensi orafce

Ekstensi Orafce menyediakan fungsi dan operator yang meniru subset fungsi dan paket dari basis data Oracle. Ekstensi orafce memudahkan Anda untuk mem-port aplikasi Oracle ke PostgreSQL. RDS for PostgreSQL versi 9.6.6 dan yang lebih tinggi mendukung ekstensi ini. Untuk informasi lebih lanjut tentang orafce, lihat orafce di. GitHub

> **Note**
>
> RDS for PostgreSQL tidak mendukung paket `utl_file` yang merupakan bagian dari ekstensi orafce. Karena fungsi skema `utl_file` menyediakan operasi baca dan tulis pada

file teks sistem operasi, yang membutuhkan akses superuser ke host yang mendasarinya. Sebagai layanan terkelola, RDS for PostgreSQL tidak menyediakan akses host.

Menggunakan ekstensi orafce

1. Hubungkan ke instans DB dengan nama pengguna utama yang Anda gunakan untuk membuat instans DB.

   Jika Anda ingin mengaktifkan orafce untuk basis data yang berbeda dalam instans DB yang sama, gunakan perintah psql `/c dbname`. Dengan menggunakan perintah ini, Anda dapat mengubah dari basis data utama setelah memulai koneksi.
2. Nyalakan ekstensi orafce dengan pernyataan `CREATE EXTENSION`.

   ```
   CREATE EXTENSION orafce;
   ```
3. Transfer kepemilikan skema oracle ke peran rds_superuser dengan pernyataan `ALTER SCHEMA`.

   ```
   ALTER SCHEMA oracle OWNER TO rds_superuser;
   ```

   Jika Anda ingin melihat daftar pemilik untuk skema oracle, gunakan perintah psql `\dn`.

Mengelola partisi PostgreSQL dengan ekstensi pg_partman

Partisi tabel PostgreSQL menyediakan kerangka kerja untuk penanganan input data dan laporan performa tinggi. Gunakan partisi untuk basis data yang memerlukan input data dalam jumlah besar dengan cepat. Partisi juga menyediakan kueri tabel besar yang lebih cepat. Partisi membantu memelihara data tanpa memengaruhi instans basis data karena sumber daya I/O yang diperlukan lebih sedikit.

Dengan partisi, Anda dapat membagi data menjadi beberapa bagian berukuran kustom untuk diproses. Misalnya, Anda dapat membagi data deret waktu ke dalam berbagai rentang seperti per jam, harian, mingguan, bulanan, triwulanan, tahunan, kustom, atau kombinasinya. Untuk contoh data deret waktu, jika Anda membagi tabel berdasarkan jam, setiap partisi akan berisi data per satu jam. Jika Anda membagi tabel deret waktu berdasarkan hari, partisi akan berisi data per hari, dan seterusnya. Kunci partisi mengontrol ukuran partisi.

Saat Anda menggunakan perintah SQL `INSERT` atau `UPDATE` pada tabel yang dipartisi, mesin basis data merutekan data ke partisi yang sesuai. Partisi tabel PostgreSQL yang menyimpan data adalah tabel turunan dari tabel utama.

Selama pembacaan kueri basis data, pengoptimal PostgreSQL memeriksa klausul `WHERE` pada kueri dan, jika memungkinkan, mengarahkan pemindaian basis data hanya untuk partisi yang relevan.

Mulai versi 10, PostgreSQL menggunakan partisi deklaratif untuk mengimplementasikan partisi tabel. Ini juga dikenal sebagai partisi PostgreSQL asli. Sebelum PostgreSQL versi 10, Anda menggunakan pemicu untuk mengimplementasikan partisi.

Partisi tabel PostgreSQL menyediakan fitur berikut:

- Pembuatan partisi baru setiap saat.
- Rentang partisi bervariasi.
- Partisi yang dapat dilepas dan dapat dipasang kembali menggunakan pernyataan bahasa definisi data (DDL).

  Sebagai contoh, partisi yang dapat dilepas berguna untuk menghapus data historis dari partisi utama, tetapi menyimpan data historis untuk analisis.
- Partisi baru mewarisi properti tabel basis data induk, termasuk yang berikut ini:
  - Indeks
  - Kunci primer, yang harus berisi kolom kunci partisi

  - Kunci asing
  - Batasan pemeriksaan
  - Referensi
- Membuat indeks untuk seluruh tabel atau partisi tertentu.

Anda tidak dapat mengubah skema partisi individual. Namun, Anda dapat mengubah tabel induk (seperti menambahkan kolom baru), yang disebarkan ke partisi.

Topik

- Ikhtisar ekstensi pg_partman PostgreSQL
- Mengaktifkan ekstensi pg_partman
- Mengonfigurasi partisi menggunakan fungsi create_parent
- Mengonfigurasi pemeliharaan partisi menggunakan fungsi run_maintenance_proc

Ikhtisar ekstensi pg_partman PostgreSQL

Anda dapat menggunakan ekstensi `pg_partman` PostgreSQL untuk mengotomatiskan pembuatan dan pemeliharaan partisi tabel. Untuk informasi umum selengkapnya, lihat Manajer Partisi PG dalam dokumentasi `pg_partman`.

> **Note**
>
> Ekstensi `pg_partman` didukung pada RDS for PostgreSQL versi 12.5 dan yang lebih tinggi.

Alih-alih membuat setiap partisi secara manual, Anda dapat mengonfigurasi `pg_partman` dengan pengaturan berikut:

- Tabel yang akan dipartisi
- Jenis partisi
- Kunci partisi
- Granularitas partisi
- Opsi pra-pembuatan dan manajemen partisi

Setelah membuat tabel yang dipartisi PostgreSQL, daftarkan dengan `pg_partman` dengan memanggil fungsi `create_parent`. Tindakan ini akan membuat partisi yang diperlukan berdasarkan parameter yang Anda teruskan ke fungsi.

Ekstensi `pg_partman` juga menyediakan fungsi `run_maintenance_proc`, yang dapat Anda panggil sesuai jadwal untuk secara otomatis mengelola partisi. Untuk memastikan bahwa partisi yang tepat dibuat sesuai kebutuhan, jadwalkan fungsi ini untuk berjalan secara berkala (seperti per jam). Anda juga dapat memastikan bahwa partisi secara otomatis dibatalkan.

Mengaktifkan ekstensi pg_partman

Jika Anda memiliki beberapa basis data di dalam instans DB PostgreSQL yang partisinya ingin Anda kelola, aktifkan ekstensi `pg_partman` secara terpisah untuk setiap basis data. Untuk mengaktifkan ekstensi `pg_partman` untuk basis data tertentu, buat skema pemeliharaan partisi, kemudian buat ekstensi `pg_partman` seperti berikut.

```
CREATE SCHEMA partman;
CREATE EXTENSION pg_partman WITH SCHEMA partman;
```

> **Note**
>
> Untuk membuat ekstensi `pg_partman`, pastikan Anda memiliki hak istimewa `rds_superuser`.

Jika Anda menerima kesalahan seperti berikut, berikan hak istimewa `rds_superuser` untuk akun tersebut atau gunakan akun pengguna super Anda.

```
ERROR: permission denied to create extension "pg_partman"
HINT: Must be superuser to create this extension.
```

Untuk memberikan hak istimewa `rds_superuser`, hubungkan dengan akun pengguna super Anda dan jalankan perintah berikut.

```
GRANT rds_superuser TO user-or-role;
```

Untuk contoh yang menunjukkan penggunaan ekstensi pg_partman, kita gunakan contoh tabel dan partisi basis data berikut. Basis data ini menggunakan tabel yang dipartisi berdasarkan stempel

waktu. Skema `data_mart` berisi tabel bernama `events` dengan kolom bernama `created_at`. Pengaturan berikut disertakan dalam tabel `events`:

- Kunci primer `event_id` dan `created_at`, yang harus memiliki kolom yang digunakan untuk memandu partisi.
- Batasan pemeriksaan `ck_valid_operation` guna menerapkan nilai untuk kolom tabel `operation`.
- Dua kunci asing, yang salah satunya (`fk_orga_membership`)) menunjuk ke tabel eksternal `organization` dan kunci lainnya (`fk_parent_event_id`) adalah kunci asing referensi mandiri.
- Dua indeks, yang salah satunya (`idx_org_id`) untuk kunci asing dan indeks lainnya (`idx_event_type`) untuk jenis peristiwa.

Pernyataan DDL berikut membuat objek ini, yang secara otomatis disertakan pada setiap partisi.

```
CREATE SCHEMA data_mart;
CREATE TABLE data_mart.organization ( org_id BIGSERIAL,
        org_name TEXT,
        CONSTRAINT pk_organization PRIMARY KEY (org_id)
    );

CREATE TABLE data_mart.events(
        event_id        BIGSERIAL,
        operation       CHAR(1),
        value           FLOAT(24),
        parent_event_id BIGINT,
        event_type      VARCHAR(25),
        org_id          BIGSERIAL,
        created_at      timestamp,
        CONSTRAINT pk_data_mart_event PRIMARY KEY (event_id, created_at),
        CONSTRAINT ck_valid_operation CHECK (operation = 'C' OR operation = 'D'),
        CONSTRAINT fk_orga_membership
            FOREIGN KEY(org_id)
            REFERENCES data_mart.organization (org_id),
        CONSTRAINT fk_parent_event_id
            FOREIGN KEY(parent_event_id, created_at)
            REFERENCES data_mart.events (event_id,created_at)
    ) PARTITION BY RANGE (created_at);

CREATE INDEX idx_org_id     ON  data_mart.events(org_id);
CREATE INDEX idx_event_type ON  data_mart.events(event_type);
```

Mengonfigurasi partisi menggunakan fungsi create_parent

Setelah mengaktifkan ekstensi `pg_partman`, gunakan fungsi `create_parent` untuk mengonfigurasi partisi di dalam skema pemeliharaan partisi. Contoh berikut menggunakan contoh tabel `events` yang dibuat di Mengaktifkan ekstensi pg_partman. Panggil fungsi `create_parent` seperti berikut.

```
SELECT partman.create_parent( p_parent_table => 'data_mart.events',
 p_control => 'created_at',
 p_type => 'native',
 p_interval=> 'daily',
 p_premake => 30);
```

Parameternya adalah sebagai berikut:

- `p_parent_table` – Tabel induk yang dipartisi. Tabel ini harus sudah ada dan sepenuhnya memenuhi syarat, termasuk skemanya.
- `p_control` – Kolom yang menjadi dasar pembuatan partisi. Jenis data harus bilangan bulat atau berbasis waktu.
- `p_type` – Jenisnya adalah `'native'` atau `'partman'`. Anda biasanya menggunakan jenis `native` untuk peningkatan performa dan fleksibilitas. Jenis `partman` bergantung pada warisan.
- `p_interval` – Interval waktu atau rentang bilangan bulat untuk setiap partisi. Contoh nilainya termasuk `daily`, per jam, dan sebagainya.
- `p_premake` – Jumlah partisi yang akan dibuat terlebih dahulu untuk mendukung sisipan baru.

Untuk keterangan lengkap tentang fungsi `create_parent`, lihat Fungsi Pembuatan dalam dokumentasi `pg_partman`.

Mengonfigurasi pemeliharaan partisi menggunakan fungsi run_maintenance_proc

Anda dapat menjalankan operasi pemeliharaan partisi untuk secara otomatis membuat partisi baru, melepaskan partisi, atau menghapus partisi lama. Pemeliharaan partisi bergantung pada fungsi `run_maintenance_proc` pada ekstensi `pg_partman` dan `pg_cron`, yang memulai penjadwal internal. Penjadwal `pg_cron` secara otomatis mengeksekusi pernyataan, fungsi, dan prosedur SQL yang ditetapkan dalam basis data Anda.

Contoh berikut menggunakan contoh tabel `events` yang dibuat di Mengaktifkan ekstensi pg_partman untuk mengatur operasi pemeliharaan partisi agar berjalan secara otomatis. Sebagai prasyarat, tambahkan `pg_cron` ke parameter `shared_preload_libraries` dalam grup parameter instans DB.

```
CREATE EXTENSION pg_cron;

UPDATE partman.part_config
SET infinite_time_partitions = true,
    retention = '3 months',
    retention_keep_table=true
WHERE parent_table = 'data_mart.events';
SELECT cron.schedule('@hourly', $$CALL partman.run_maintenance_proc()$$);
```

Berikut penjelasan langkah demi langkah untuk contoh sebelumnya:

1. Modifikasi grup parameter yang terkait dengan instans DB Anda dan tambahkan `pg_cron` ke nilai parameter `shared_preload_libraries`. Untuk menerapkan perubahan, instans DB harus dimulai ulang. Untuk informasi selengkapnya, lihat Memodifikasi parameter dalam grup parameter DB.
2. Jalankan perintah `CREATE EXTENSION pg_cron;` menggunakan akun yang memiliki izin `rds_superuser`. Tindakan ini akan mengaktifkan ekstensi `pg_cron`. Untuk informasi selengkapnya, lihat Menjadwalkan pemeliharaan dengan ekstensi pg_cron PostgreSQL.
3. Jalankan perintah `UPDATE partman.part_config` guna menyesuaikan pengaturan `pg_partman` untuk tabel `data_mart.events`.
4. Jalankan perintah `SET . . .` untuk mengonfigurasi tabel `data_mart.events` dengan klausul berikut:
   a. `infinite_time_partitions = true,` – Mengonfigurasi tabel untuk dapat secara otomatis membuat partisi baru tanpa batas.
   b. `retention = '3 months',` – Mengonfigurasi tabel agar memiliki retensi maksimum tiga bulan.
   c. `retention_keep_table=true` – Mengonfigurasi tabel agar ketika periode retensi sudah habis, tabel tidak akan dihapus secara otomatis. Sebaliknya, partisi yang lebih lama dari periode retensi hanya dilepaskan dari tabel induk.
5. Jalankan perintah `SELECT cron.schedule . . .` untuk membuat panggilan fungsi `pg_cron`. Panggilan ini menetapkan frekuensi penjadwal menjalankan prosedur pemeliharaan `pg_partman`, `partman.run_maintenance_proc`. Untuk contoh ini, prosedur berjalan setiap jam.

Untuk keterangan lengkap tentang fungsi `run_maintenance_proc`, lihat Fungsi Pemeliharaan dalam dokumentasi `pg_partman`.

Menggunakan pgAudit untuk membuat log aktivitas basis data

Lembaga keuangan, lembaga pemerintah, dan banyak industri perlu menyimpan log audit untuk memenuhi persyaratan peraturan. Dengan menggunakan ekstensi PostgreSQL Audit (pgAudit) dengan instans DB RDS for PostgreSQL, Anda dapat menangkap catatan terperinci yang biasanya dibutuhkan oleh auditor atau untuk memenuhi persyaratan peraturan. Misalnya, Anda dapat mengatur ekstensi pgAudit untuk melacak perubahan yang dibuat pada basis data dan tabel tertentu, untuk merekam pengguna yang membuat perubahan, dan banyak detail lainnya.

Ekstensi pgAudit dibangun di atas fungsionalitas infrastruktur pencatatan log PostgreSQL asli dengan memperluas pesan log dengan lebih detail. Dengan kata lain, Anda menggunakan pendekatan yang sama untuk melihat log audit Anda seperti yang Anda lakukan untuk melihat pesan log apa pun. Untuk informasi selengkapnya tentang pencatatan log PostgreSQL, lihat File log basis data RDS for PostgreSQL.

Ekstensi pgAudit menyunting data sensitif seperti kata sandi cleartext dari log. Jika instans DB RDS for PostgreSQL Anda dikonfigurasi untuk mencatat pernyataan bahasa manipulasi data (DHTML) seperti yang dijelaskan dalam Mengaktifkan pengelogan kueri untuk instans DB RDS for PostgreSQL, Anda dapat menghindari masalah kata sandi cleartext dengan menggunakan ekstensi PostgreSQL Audit.

Anda dapat mengonfigurasi audit pada basis data instans Anda dengan tingkat kekhususan yang tinggi. Anda dapat mengaudit semua basis data dan semua pengguna. Atau, Anda dapat memilih untuk mengaudit hanya basis data tertentu, pengguna, dan objek lainnya. Anda juga dapat secara eksplisit mengecualikan pengguna dan basis data tertentu agar tidak diaudit. Untuk informasi selengkapnya, lihat Mengecualikan pengguna atau basis data dari pencatatan log audit.

Mengingat jumlah detail yang dapat ditangkap, kami menyarankan jika Anda menggunakan pgAudit, Anda memantau konsumsi penyimpanan Anda.

Ekstensi pgAudit didukung pada semua Versi RDS for PostgreSQL. Untuk daftar versi pgAudit yang didukung oleh versi RDS for PostgreSQL yang tersedia, lihat Version ekstensi untuk Amazon RDS for PostgreSQL di Catatan rilis Amazon RDS for PostgreSQL.

Topik

- Menyiapkan ekstensi pgAdit
- Audit objek basis data
- Mengecualikan pengguna atau basis data dari pencatatan log audit

- Referensi untuk ekstensi pgAudit

Menyiapkan ekstensi pgAdit

Untuk menyiapkan ekstensi pgAudit pada instans DB RDS for PostgreSQL, Anda terlebih dahulu menambahkan pgAudit ke pustaka bersama pada grup parameter DB khusus untuk instans RDS DB for PostgreSQL. Untuk informasi cara membuat grup parameter DB khusus, lihat Bekerja dengan grup parameter. Selanjutnya, Anda menginstal ekstensi pgAudit. Terakhir, Anda menentukan basis data dan objek yang ingin Anda audit. Prosedur di bagian ini menunjukkan caranya kepada Anda. Anda dapat menggunakan AWS Management Console atau AWS CLI.

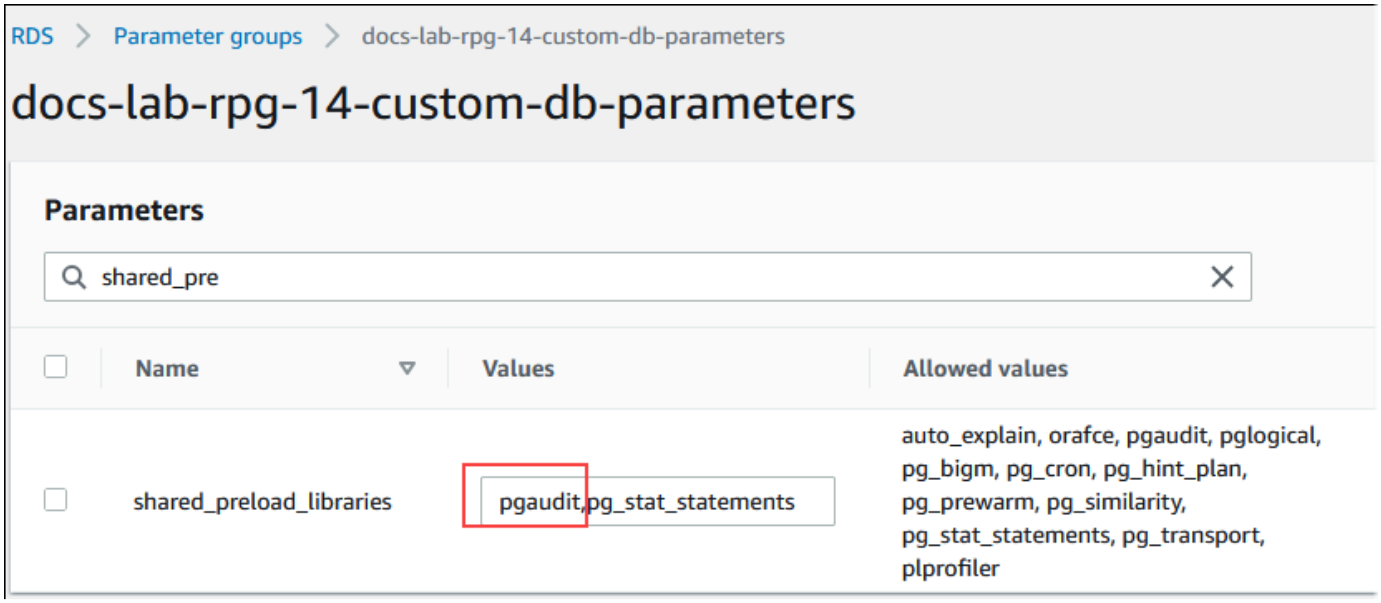
Anda harus memiliki izin sebagai peran `rds_superuser` untuk melakukan semua tugas ini.

Langkah-langkah berikut mengasumsikan bahwa instans DB RDS for PostgreSQL Anda dikaitkan dengan grup parameter DB khusus.

Konsol

Mengatur ekstensi pgAudit

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih instans DB RDS for PostgreSQL Anda.
3. Buka tab Konfigurasi untuk Instans DB RDS for PostgreSQL. Di antara detail Instans, temukan tautan Grup parameter.
4. Pilih tautan untuk membuka parameter kustom yang terkait dengan Instans DB RDS for PostgreSQL.
5. Di kolom pencarian Parameter, ketik `shared_pre` untuk menemukan parameter `shared_preload_libraries`.
6. Pilih Edit parameter untuk mengakses nilai properti.
7. Tambahkan `pgaudit` ke daftar di kolom Nilai. Gunakan koma untuk memisahkan item dalam daftar nilai.



8. Boot ulang instans DB RDS for PostgreSQL Anda sehingga perubahan Anda pada parameter `shared_preload_libraries` berlaku.
9. Ketika instans tersedia, verifikasi bahwa pgAudit yang telah diinisialisasi. Gunakan `psql` untuk terhubung ke , instans DB RDS for PostgreSQL, kemudian jalankan perintah berikut.

```
SHOW shared_preload_libraries;
shared_preload_libraries
--------------------------
rdsutils,pgaudit
(1 row)
```

10. Dengan pgAudit yang diinisialisasi, Anda sekarang dapat membuat ekstensi. Anda perlu membuat ekstensi setelah menginisialisasi pustaka karena ekstensi `pgaudit` menginstal pemicu peristiwa untuk mengaudit pernyataan bahasa definisi data (DDL).

```
CREATE EXTENSION pgaudit;
```

11. Tutup sesi `psql`.

```
labdb=> \q
```

12. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
13. Temukan parameter `pgaudit.log` dalam daftar lalu atur ke nilai yang sesuai untuk kasus penggunaan Anda. Misalnya, menyetel parameter `pgaudit.log` ke `write` seperti yang

ditunjukkan pada gambar berikut menangkap sisipan, pembaruan, penghapusan, dan beberapa jenis lainnya yang berubah pada log.

Anda juga dapat memilih salah satu nilai berikut untuk parameter `pgaudit.log`.

- none – Ini adalah default. Tidak ada perubahan basis data yang dibuat log.
- semua – Semua dibuat log (baca, tulis, fungsi, peran, ddl, lain-lain).
- ddl – Membuat log semua pernyataan bahasa definisi data (DDL) yang tidak disertakan dalam kelas `ROLE`.
- fungsi – Membuat log panggilan fungsi dan blok `DO`.
- lain-lain – Membuat log berbagai perintah, seperti `DISCARD`, `FETCH`, `CHECKPOINT`, `VACUUM`, dan `SET`.
- baca – Membuat log `SELECT` dan `COPY` saat sumbernya adalah relasi (seperti tabel) atau kueri.
- peran – Membuat log pernyataan yang terkait dengan peran dan hak istimewa, seperti `GRANT`, `REVOKE`, `CREATE ROLE`, `ALTER ROLE`, dan `DROP ROLE`.
- write – Membuat log `INSERT`, `UPDATE`, `DELETE`, `TRUNCATE`, dan `COPY` ketika tujuan adalah relasi (tabel).

14. Pilih Simpan perubahan
15. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
16. Pilih instans DB RDS for PostgreSQL dari daftar Basis Data untuk memilihnya, lalu pilih Boot ulang dari menu Tindakan.

AWS CLI

Menyiapkan pgAudit

Untuk mengatur PGAudit menggunakanAWS CLI, Anda memanggil modify-db-parameter-groupoperasi untuk mengubah parameter log audit di grup parameter kustom Anda, seperti yang ditunjukkan dalam prosedur berikut.

1. Gunakan perintah AWS CLI berikut untuk menambah `pgaudit` ke parameter `shared_preload_libraries`.

```
aws rds modify-db-parameter-group \
   --db-parameter-group-name custom-param-group-name \
   --parameters
 "ParameterName=shared_preload_libraries,ParameterValue=pgaudit,ApplyMethod=pending-
reboot" \
   --region aws-region
```

2. Gunakan perintah AWS CLI berikut untuk melakukan boot ulang instans DB RDS for PostgreSQL sehingga pustaka pgaudit dapat diinisialisasi.

```
aws rds reboot-db-instance \
    --db-instance-identifier your-instance \
    --region aws-region
```

3. Ketika instans tersedia, Anda dapat memverifikasi bahwa `pgaudit` telah diinisialisasi. Gunakan `psql` untuk terhubung ke , instans DB RDS for PostgreSQL, kemudian jalankan perintah berikut.

```
SHOW shared_preload_libraries;
shared_preload_libraries
--------------------------
rdsutils,pgaudit
(1 row)
```

   Dengan pgAudit yang diinisialisasi, Anda sekarang dapat membuat ekstensi.

```
CREATE EXTENSION pgaudit;
```

4. Tutup sesi `psql` sehingga Anda dapat menggunakan AWS CLI.

```
labdb=> \q
```

5. Gunakan perintah AWS CLI berikut untuk menentukan kelas pernyataan yang ingin dibuat log oleh sesi audit pencatatan log. Contoh menetapkan parameter `pgaudit.log` ke `write`, yang menangkap sisipan, pembaruan, dan penghapusan ke log.

```
aws rds modify-db-parameter-group \
   --db-parameter-group-name custom-param-group-name \
   --parameters
 "ParameterName=pgaudit.log,ParameterValue=write,ApplyMethod=pending-reboot" \
   --region aws-region
```

   Anda juga dapat memilih salah satu nilai berikut untuk parameter `pgaudit.log`.

   - none – Ini adalah default. Tidak ada perubahan basis data yang dibuat log.
   - semua – Semua dibuat log (baca, tulis, fungsi, peran, ddl, lain-lain).
   - ddl – Membuat log semua pernyataan bahasa definisi data (DDL) yang tidak disertakan dalam kelas `ROLE`.
   - fungsi – Membuat log panggilan fungsi dan blok `DO`.
   - lain-lain – Membuat log berbagai perintah, seperti `DISCARD`, `FETCH`, `CHECKPOINT`, `VACUUM`, dan `SET`.
   - baca – Membuat log `SELECT` dan `COPY` saat sumbernya adalah relasi (seperti tabel) atau kueri.
   - peran – Membuat log pernyataan yang terkait dengan peran dan hak istimewa, seperti `GRANT`, `REVOKE`, `CREATE ROLE`, `ALTER ROLE`, dan `DROP ROLE`.
   - write – Membuat log `INSERT`, `UPDATE`, `DELETE`, `TRUNCATE`, dan `COPY` ketika tujuan adalah relasi (tabel).

   Boot ulang menggunakan perintah berikut. AWS CLI

```
aws rds reboot-db-instance \
    --db-instance-identifier your-instance \
    --region aws-region
```

Audit objek basis data

Dengan pgAudit yang telah disiapkan di instans DB RDS for PostgreSQL Anda dan dikonfigurasi untuk kebutuhan Anda, informasi lebih detail akan ditangkap dalam pembuatan log PostgreSQL.

Misalnya, sementara konfigurasi pencatatan log PostgreSQL default mengidentifikasi tanggal dan waktu perubahan dibuat dalam tabel basis data, dengan ekstensi pgAudit entri log yang dapat menyertakan skema, pengguna yang membuat perubahan, dan detail lainnya tergantung pada bagaimana parameter ekstensi dikonfigurasi. Anda dapat menyiapkan audit untuk melacak perubahan dengan cara berikut ini.

- Untuk setiap sesi, oleh pengguna. Untuk tingkat sesi, Anda dapat menangkap teks perintah yang sepenuhnya memenuhi syarat.
- Untuk setiap objek, oleh pengguna dan basis data.

Kemampuan audit objek diaktifkan saat Anda membuat peran `rds_pgaudit` pada sistem Anda lalu akan menambahkan peran ini ke parameter `pgaudit.role` dalam grup parameter parameter khusus Anda. Secara default, parameter `pgaudit.role` tidak disetel dan satu-satunya nilai yang diizinkan adalah `rds_pgaudit`. Langkah-langkah berikut mengasumsikan bahwa `pgaudit` telah diinisialisasi dan bahwa Anda telah membuat ekstensi `pgaudit` dengan mengikuti prosedur di Menyiapkan ekstensi pgAdit.

```
2022-10-07 23:36:51 UTC:52.95.4.10(14410):postgres@labdb:[1374]:LOG: statement: SELECT feedback, s.sentiment,s.confidence
FROM support,aws_comprehend.detect_sentiment(feedback, 'en') s
ORDER BY s.confidence DESC;
2022-10-07 23:36:51 UTC:52.95.4.10(14410):postgres@labdb:[1374]:LOG: AUDIT: SESSION,2,1,READ,SELECT,TABLE,public.support,"SELECT
feedback, s.sentiment,s.confidence
FROM support,aws_comprehend.detect_sentiment(feedback, 'en') s
ORDER BY s.confidence DESC;",<none>
2022-10-07 23:36:51 UTC:52.95.4.10(14410):postgres@labdb:[1374]:LOG: QUERY STATISTICS
2022-10-07 23:36:51 UTC:52.95.4.10(14410):postgres@labdb:[1374]:DETAIL: ! system usage stats:
! 0.009494 s user, 0.007442 s system, 0.141985 s elapsed
! [0.022327 s user, 0.007442 s system total]
```

Seperti yang ditunjukkan dalam contoh ini, baris "LOG: AUDIT: SESSION" memberikan informasi tabel dan skemanya, di antara detail lainnya.

Menyiapkan audit objek

1. Gunakan `psql` untuk terhubung ke instans DB RDS for PostgreSQL.

   ```
   psql --host=your-instance-name.aws-region.rds.amazonaws.com --port=5432 --
   username=postgrespostgres --password --dbname=labdb
   ```

2. Buat peran basis data yang dinamai `rds_pgaudit` dengan menggunakan perintah berikut.

   ```
   labdb=> CREATE ROLE rds_pgaudit;
   CREATE ROLE
   ```

```
labdb=>
```

3. Tutup sesi `psql`.

```
labdb=> \q
```

Dalam beberapa langkah berikutnya, gunakan AWS CLI untuk memodifikasi parameter log audit di grup parameter khusus Anda.

4. Gunakan perintah AWS CLI berikut untuk mengatur parameter `pgaudit.role` ke `rds_pgaudit`. Secara default, parameter ini kosong, dan `rds_pgaudit` merupakan satu-satunya nilai yang diizinkan.

```
aws rds modify-db-parameter-group \
   --db-parameter-group-name custom-param-group-name \
   --parameters
 "ParameterName=pgaudit.role,ParameterValue=rds_pgaudit,ApplyMethod=pending-reboot"
 \
   --region aws-region
```

5. Gunakan perintah AWS CLI berikut untuk melakukan boot ulang instans DB RDS for PostgreSQL sehingga perubahan Anda pada parameter berlaku.

```
aws rds reboot-db-instance \
    --db-instance-identifier your-instance \
    --region aws-region
```

6. Jalankan perintah berikut untuk mengonfirmasi bahwa `pgaudit.role` diatur ke`rds_pgaudit`.

```
SHOW pgaudit.role;
pgaudit.role
------------------
rds_pgaudit
```

Untuk menguji pencatatan log pgAudit, Anda dapat menjalankan beberapa contoh perintah yang ingin Anda audit. Misalnya, Anda dapat menjalankan perintah berikut.

```
CREATE TABLE t1 (id int);
GRANT SELECT ON t1 TO rds_pgaudit;
SELECT * FROM t1;
id
```

```
----
(0 rows)
```

Log basis data harus berisi entri yang serupa dengan entri berikut.

```
...
2017-06-12 19:09:49 UTC:...:rds_test@postgres:[11701]:LOG: AUDIT:
OBJECT,1,1,READ,SELECT,TABLE,public.t1,select * from t1;
...
```

Untuk informasi tentang melihat log, lihat Memantau file log Amazon RDS.

Untuk mempelajari lebih lanjut tentang ekstensi PGAudit, lihat PGAudit aktif. GitHub

Mengecualikan pengguna atau basis data dari pencatatan log audit

Seperti dibahas dalam File log basis data RDS for PostgreSQL, log PostgreSQL menghabiskan ruang penyimpanan. Menggunakan ekstensi pgAudit dapat menambah volume data yang dikumpulkan di log Anda ke berbagai tingkat, tergantung pada perubahan yang Anda lacak. Anda mungkin tidak perlu mengaudit setiap pengguna atau basis data di Anda. Instans DB RDS for PostgreSQL.

Untuk meminimalkan dampak pada penyimpanan Anda dan untuk menghindari pengambilan catatan audit yang tidak perlu, Anda dapat mengecualikan pengguna dan basis data agar tidak diaudit. Anda juga dapat mengubah pencatatan log dalam sesi tertentu. Contoh berikut menunjukkan caranya kepada Anda.

> **Note**
>
> Pengaturan parameter pada tingkat sesi lebih diutamakan daripada pengaturan dalam grup parameter DB khusus untuk instans DB RDS for PostgreSQL. Jika Anda tidak ingin pengguna basis data melewati pengaturan konfigurasi pencatatan audit Anda, pastikan untuk mengubah izin mereka.

Misalkan instans DB RDS for PostgreSQL Anda dikonfigurasi untuk mengaudit tingkat aktivitas yang sama untuk semua pengguna dan basis data. Anda kemudian memutuskan bahwa Anda tidak ingin mengaudit pengguna `myuser`. Anda dapat mematikan audit `myuser` dengan perintah SQL berikut.

```
ALTER USER myuser SET pgaudit.log TO 'NONE';
```

Kemudian, Anda dapat menggunakan kueri berikut untuk memeriksa kolom `user_specific_settings` untuk `pgaudit.log` agar mengonfirmasi bahwa parameter diatur ke `NONE`.

```
SELECT
    usename AS user_name,
    useconfig AS user_specific_settings
FROM
    pg_user
WHERE
    usename = 'myuser';
```

Anda akan melihat output seperti berikut ini.

```
 user_name | user_specific_settings
-----------+------------------------
 myuser    | {pgaudit.log=NONE}
(1 row)
```

Anda dapat mematikan pencatatan log untuk pengguna tertentu di tengah-tengah sesi mereka dengan basis data menggunakan perintah berikut.

```
ALTER USER myuser IN DATABASE mydatabase SET pgaudit.log TO 'none';
```

Gunakan kueri berikut untuk memeriksa kolom pengaturan untuk pgaudit.log untuk kombinasi pengguna dan basis data tertentu.

```
SELECT
    usename AS "user_name",
    datname AS "database_name",
    pg_catalog.array_to_string(setconfig, E'\n') AS "settings"
FROM
    pg_catalog.pg_db_role_setting s
    LEFT JOIN pg_catalog.pg_database d ON d.oid = setdatabase
    LEFT JOIN pg_catalog.pg_user r ON r.usesysid = setrole
WHERE
    usename = 'myuser'
    AND datname = 'mydatabase'
ORDER BY
    1,
```

```
    2;
```

Anda akan melihat output yang mirip dengan berikut ini.

```
  user_name | database_name |     settings
-----------+---------------+------------------
 myuser    | mydatabase    | pgaudit.log=none
(1 row)
```

Setelah menonaktifkan audit `myuser`, Anda memutuskan bahwa Anda tidak ingin melacak perubahan ke `mydatabase`. Anda mematikan audit untuk basis data spesifik tersebut menggunakan perintah berikut.

```
ALTER DATABASE mydatabase SET pgaudit.log to 'NONE';
```

Kemudian, gunakan kueri berikut untuk memeriksa kolom database_specific_settings untuk mengonfirmasi bahwa pgaudit.log disetel ke NONE.

```
SELECT
a.datname AS database_name,
b.setconfig AS database_specific_settings
FROM
pg_database a
FULL JOIN pg_db_role_setting b ON a.oid = b.setdatabase
WHERE
a.datname = 'mydatabase';
```

Anda akan melihat output seperti berikut ini.

```
 database_name | database_specific_settings
---------------+----------------------------
 mydatabase    | {pgaudit.log=NONE}
(1 row)
```

Untuk mengembalikan pengaturan ke pengaturan default untuk myuser, gunakan perintah berikut:

```
ALTER USER myuser RESET pgaudit.log;
```

Untuk mengembalikan pengaturan ke pengaturan default untuk basis data, gunakan perintah berikut ini.

```
ALTER DATABASE mydatabase RESET pgaudit.log;
```

Untuk mengatur ulang pengguna dan basis data ke pengaturan default, gunakan perintah berikut.

```
ALTER USER myuser IN DATABASE mydatabase RESET pgaudit.log;
```

Anda juga dapat menangkap peristiwa tertentu ke log dengan menyetel `pgaudit.log` ke salah satu nilai lain yang diizinkan untuk parameter `pgaudit.log`. Untuk informasi selengkapnya, lihat Daftar pengaturan yang diizinkan untuk parameter `pgaudit.log`.

```
ALTER USER myuser SET pgaudit.log TO 'read';
ALTER DATABASE mydatabase SET pgaudit.log TO 'function';
ALTER USER myuser IN DATABASE mydatabase SET pgaudit.log TO 'read,function'
```

Referensi untuk ekstensi pgAudit

Anda dapat menentukan tingkat detail yang Anda inginkan untuk log audit Anda dengan mengubah satu atau beberapa parameter yang tercantum di bagian ini.

Mengatur perilaku pgAudit

Anda dapat mengontrol pencatatan audit dengan mengubah satu atau beberapa parameter yang tercantum dalam tabel berikut.

Parameter	Deskripsi
`pgaudit.log`	Menentukan kelas pernyataan yang akan dicatat oleh sesi audit pencatatan log. Nilai yang diizinkan termasuk ddl, fungsi, misc, baca, peran, tulis, tidak ada, semua. Untuk informasi selengkap nya, lihat Daftar pengaturan yang diizinkan untuk parameter `pgaudit.log` .
`pgaudit.log_catalog`	Saat diaktifkan (disetel ke 1), tambahkan pernyataan ke jejak audit jika semua relasi dalam pernyataan ada di pg_catalog.
`pgaudit.log_level`	Menentukan tingkat log untuk digunakan untuk entri log. Nilai yang diizinkan: debug5, debug4, debug3, debug2, debug1, info, notifikasi, peringatan, log

Parameter	Deskripsi
`pgaudit.log_parameter`	Ketika diaktifkan (diatur ke 1), parameter yang diteruskan dengan pernyataan ditangkap dalam log audit.
`pgaudit.log_relation`	Saat diaktifkan (disetel ke 1), log audit untuk sesi membuat entri log terpisah untuk setiap relasi (TABEL, TAMPILAN, dan sebagainya) yang direferensikan dalam pernyataan SELECT atau DML.
`pgaudit.log_statement_once`	Menentukan apakah pencatatan log akan mencakup teks pernyataan dan parameter dengan entri log pertama untuk kombinasi pernyataan/subpernyataan atau dengan setiap entri.
`pgaudit.role`	Menentukan peran utama yang akan digunakan untuk pencatatan log audit objek. Satu-satunya entri yang diizinkan adalah `rds_pgaudit` .

Daftar pengaturan yang diizinkan untuk parameter **`pgaudit.log`**

Nilai	Deskripsi
tidak ada	Ini adalah default. Tidak ada perubahan basis data yang dibuat log.
semua	Semuanya dibuat log (baca, tulis, fungsi, peran, ddl, misc).
ddl	Membuat log semua pernyataan bahasa definisi data (DDL) yang tidak disertakan dalam kelas `ROLE`.
fungsi	Log berfungsi panggilan dan blok `DO`.
misc	Log berbagai perintah, seperti `DISCARD`, `FETCH`, `CHECKPOINT` , `VACUUM` dan `SET`.
baca	Log `SELECT` dan `COPY` ketika sumbernya adalah relasi (seperti tabel) atau kueri.

Nilai	Deskripsi
peran	Log pernyataan yang terkait dengan peran dan hak istimewa, seperti `GRANT`, `REVOKE`, `CREATE ROLE`, `ALTER ROLE`, dan `DROP ROLE`.
tulis	Log `INSERT`, `UPDATE`, `DELETE`, `TRUNCATE`, dan `COPY` ketika tujuan adalah relasi (tabel).

Untuk mencatat beberapa jenis peristiwa dengan audit sesi, gunakan daftar yang dipisahkan koma. Untuk membuat log semua jenis acara, atur `pgaudit.log` ke `ALL`. Boot ulang instans DB Anda untuk menerapkan perubahan.

Dengan objek audit, Anda dapat memperbaiki pencatatan log audit untuk bekerja dengan relasi spesifik. Misalnya, Anda dapat menentukan bahwa Anda ingin pencatatan log audit untuk operasi `READ` di satu tabel atau beberapa.

Menjadwalkan pemeliharaan dengan ekstensi pg_cron PostgreSQL

Anda dapat menggunakan ekstensi `pg_cron` PostgreSQL untuk menjadwalkan perintah pemeliharaan dalam basis data PostgreSQL. Untuk informasi selengkapnya tentang ekstensi, lihat Apa itu pg_cron? dalam dokumentasi pg_cron.

Ekstensi `pg_cron` didukung pada mesin RDS for PostgreSQL versi 12.5 dan yang lebih tinggi.

Untuk mempelajari selengkapnya tentang penggunaan `pg_cron`, lihat Menjadwalkan pekerjaan dengan pg_cron di RDS for PostgreSQL atau basis data Edisi yang kompatibel dengan Aurora PostgreSQL.

Topik

Menyiapkan ekstensi pg_cron

Siapkan ekstensi `pg_cron` sebagai berikut:

1. Ubah grup parameter kustom yang terkait dengan instans DB PostgreSQL Anda dengan menambahkan `pg_cron` ke nilai parameter `shared_preload_libraries`.
   - Jika instans DB RDS for PostgreSQL menggunakan parameter `rds.allowed_extensions` untuk secara eksplisit mencantumkan ekstensi yang dapat diinstal, Anda perlu menambahkan ekstensi `pg_cron` ke daftar. Hanya versi RDS for PostgreSQL tertentu yang mendukung parameter `rds.allowed_extensions`. Secara default, semua ekstensi yang tersedia diizinkan. Untuk informasi selengkapnya, lihat Membatasi penginstalan ekstensi PostgreSQL.

   Mulai ulang instans DB PostgreSQL agar perubahan pada grup parameter dapat diterapkan. Untuk mempelajari selengkapnya tentang bekerja menggunakan grup parameter, lihat Memodifikasi parameter dalam grup parameter DB.
2. Setelah instans DB PostgreSQL dimulai ulang, jalankan perintah berikut menggunakan akun yang memiliki izin `rds_superuser`. Misalnya, jika Anda menggunakan pengaturan default saat membuat instans DB RDS for PostgreSQL, sambungkan sebagai pengguna `postgres` dan buat ekstensi.

```
CREATE EXTENSION pg_cron;
```

Penjadwal pg_cron diatur dalam basis data PostgreSQL default bernama postgres. Objek pg_cron dibuat dalam basis data postgres ini dan semua tindakan penjadwalan berjalan dalam basis data ini.

3. Anda dapat menggunakan pengaturan default, atau Anda dapat menjadwalkan pekerjaan untuk berjalan di basis data lain dalam instans DB PostgreSQL Anda. Untuk menjadwalkan pekerjaan untuk basis data lain dalam instans DB PostgreSQL Anda, lihat contoh di Menjadwalkan pekerjaan cron untuk basis data selain basis data default.

Memberikan izin pengguna basis data untuk menggunakan pg_cron

Menginstal ekstensi pg_cron membutuhkan hak istimewa rds_superuser. Namun, izin untuk menggunakan pg_cron dapat diberikan (oleh anggota grup/peran rds_superuser) kepada pengguna basis data lain, sehingga mereka dapat menjadwalkan pekerjaannya sendiri. Sebaiknya Anda memberikan izin untuk skema cron hanya sesuai kebutuhan jika skema tersebut meningkatkan operasi di lingkungan produksi Anda.

Untuk memberikan izin pengguna basis data dalam skema cron, jalankan perintah berikut:

```
postgres=> GRANT USAGE ON SCHEMA cron TO db-user;
```

Perintah ini memberikan izin *db-user* untuk mengakses skema cron untuk menjadwalkan pekerjaan cron untuk objek yang izin aksesnya mereka miliki. Jika pengguna basis data tidak memiliki izin, tugas akan gagal setelah memposting pesan kesalahan ke file postgresql.log, seperti yang ditunjukkan berikut:

```
2020-12-08 16:41:00 UTC::@:[30647]:ERROR: permission denied for table table-name
2020-12-08 16:41:00 UTC::@:[27071]:LOG: background worker "pg_cron" (PID 30647) exited
 with exit code 1
```

Dengan kata lain, pastikan bahwa pengguna database yang diberikan izin pada cron skema juga memiliki izin pada objek (tabel, skema, dan sebagainya) yang mereka rencanakan untuk dijadwalkan.

Detail pekerjaan cron dan keberhasilan atau kegagalannya juga ditangkap dalam cron.job_run_details tabel. Untuk informasi selengkapnya, lihat Tabel untuk menjadwalkan pekerjaan dan menangkap status .

Menjadwalkan pekerjaan pg_cron

Bagian berikut menunjukkan bagaimana Anda dapat menjadwalkan berbagai tugas manajemen menggunakan pekerjaan pg_cron.

> **Note**
>
> Saat Anda membuat pekerjaan pg_cron, periksa apakah pengaturan `max_worker_processes` lebih besar dari jumlah `cron.max_running_jobs`. Pekerjaan pg_cron akan gagal jika kehabisan proses pekerja latar belakang. Jumlah default pekerjaan pg_cron adalah 5. Untuk informasi selengkapnya, lihat Parameter untuk mengelola ekstensi pg_cron.

Topik

- Mengosongkan tabel
- Membersihkan tabel riwayat pg_cron
- Kesalahan pengelogan ke file postgresql.log saja
- Menjadwalkan pekerjaan cron untuk basis data selain basis data default

Mengosongkan tabel

Pengosongan otomatis menangani pemeliharaan vakum untuk kebanyakan kasus. Namun, Anda mungkin ingin menjadwalkan pengosongan tabel tertentu di waktu yang Anda pilih.

Lihat juga Bekerja dengan fitur autovacuum PostgreSQL di Amazon RDS for PostgreSQL.

Berikut adalah contoh penggunaan fungsi `cron.schedule` untuk menyiapkan pekerjaan untuk menggunakan `VACUUM FREEZE` pada tabel tertentu setiap hari pukul 22.00 (GMT).

```
SELECT cron.schedule('manual vacuum', '0 22 * * *', 'VACUUM FREEZE pgbench_accounts');
 schedule
----------
1
(1 row)
```

Setelah contoh sebelumnya berjalan, Anda dapat memeriksa riwayat di tabel `cron.job_run_details` seperti berikut.

```
postgres=> SELECT * FROM cron.job_run_details;
jobid  | runid | job_pid | database | username | command                        |
 status    | return_message | start_time                    | end_time
-------+-------+---------+----------+----------+--------------------------------
+-----------+---------------+-------------------------------
+-------------------------------
 1     | 1     | 3395    | postgres | adminuser| vacuum freeze pgbench_accounts
 | succeeded | VACUUM         | 2020-12-04 21:10:00.050386+00 | 2020-12-04
 21:10:00.072028+00
(1 row)
```

Berikut ini adalah query dari `cron.job_run_details` tabel untuk melihat pekerjaan gagal.

```
postgres=> SELECT * FROM cron.job_run_details WHERE status = 'failed';
jobid | runid | job_pid | database | username | command                         | status
 | return_message                                  | start_time                    |
 end_time
------+-------+---------+----------+----------+---------------------------------+--------
+-------------------------------------------------+-------------------------------
+-------------------------------
 5    | 4     | 30339   | postgres | adminuser| vacuum freeze pgbench_account | failed
 | ERROR: relation "pgbench_account" does not exist | 2020-12-04 21:48:00.015145+00 |
 2020-12-04 21:48:00.029567+00
(1 row)
```

Untuk informasi selengkapnya, lihat Tabel untuk menjadwalkan pekerjaan dan menangkap status .

Membersihkan tabel riwayat pg_cron

Tabel `cron.job_run_details` berisi riwayat pekerjaan cron yang bisa menjadi sangat besar dari waktu ke waktu. Sebaiknya Anda menjadwalkan pekerjaan yang membersihkan tabel ini. Misalnya, menyimpan entri yang bernilai setara seminggu mungkin cukup untuk tujuan pemecahan masalah.

Contoh berikut menggunakan fungsi cron.schedule untuk menjadwalkan pekerjaan yang berjalan setiap hari di tengah malam untuk membersihkan tabel `cron.job_run_details`. Pekerjaan yang disimpan hanya selama tujuh hari terakhir. Gunakan akun `rds_superuser` untuk menjadwalkan pekerjaan seperti berikut.

```
SELECT cron.schedule('0 0 * * *', $$DELETE
    FROM cron.job_run_details
    WHERE end_time < now() - interval '7 days'$$);
```

Untuk informasi selengkapnya, lihat Tabel untuk menjadwalkan pekerjaan dan menangkap status .

Kesalahan pengelogan ke file postgresql.log saja

Untuk mencegah penulisan ke tabel `cron.job_run_details`, ubah grup parameter yang terkait dengan instans DB PostgreSQL dan atur parameter `cron.log_run` ke nonaktif. Ekstensi `pg_cron` tidak lagi menulis ke tabel dan menangkap kesalahan ke file `postgresql.log` saja. Untuk informasi selengkapnya, lihat Memodifikasi parameter dalam grup parameter DB.

Gunakan perintah berikut untuk memeriksa nilai parameter `cron.log_run`.

```
postgres=> SHOW cron.log_run;
```

Untuk informasi selengkapnya, lihat Parameter untuk mengelola ekstensi pg_cron.

Menjadwalkan pekerjaan cron untuk basis data selain basis data default

Metadata untuk `pg_cron` semua disimpan dalam basis data default PostgreSQL bernama `postgres`. Karena pekerja latar belakang digunakan untuk menjalankan pekerjaan pemeliharaan cron, Anda dapat menjadwalkan pekerjaan di salah satu basis data Anda dalam instans DB PostgreSQL:

1. Dalam basis data cron, jadwalkan pekerjaan seperti yang biasa Anda lakukan menggunakan cron.schedule.

   ```
   postgres=> SELECT cron.schedule('database1 manual vacuum', '29 03 * * *', 'vacuum
    freeze test_table');
   ```

2. Sebagai pengguna dengan peran `rds_superuser`, perbarui kolom basis data untuk pekerjaan yang baru saja Anda buat agar berjalan di basis data lain dalam instans DB PostgreSQL Anda.

   ```
   postgres=> UPDATE cron.job SET database = 'database1' WHERE jobid = 106;
   ```

3. Verifikasi dengan membuat kueri tabel `cron.job`.

   ```
   postgres=> SELECT * FROM cron.job;
   jobid | schedule    | command                          | nodename  | nodeport |
    database | username  | active | jobname
   ------+-------------+----------------------------------+-----------+----------
   +----------+-----------+--------+------------------------
   ```

```
106   | 29 03 * * * | vacuum freeze test_table       | localhost | 8192     |
 database1| adminuser | t      | database1 manual vacuum
  1   | 59 23 * * * | vacuum freeze pgbench_accounts | localhost | 8192     |
 postgres | adminuser | t      | manual vacuum
(2 rows)
```

> **Note**
>
> Dalam beberapa situasi, Anda mungkin menambahkan pekerjaan cron yang ingin Anda jalankan di basis data yang berbeda. Dalam kasus tersebut, pekerjaan mungkin mencoba untuk dijalankan dalam basis data default (`postgres`) sebelum Anda memperbarui kolom basis data yang benar. Jika nama pengguna memiliki izin, berarti pekerjaan berhasil dijalankan di basis data default.

Referensi untuk ekstensi pg_cron

Anda dapat menggunakan parameter, fungsi, dan tabel berikut dengan ekstensi `pg_cron`. Untuk informasi selengkapnya, lihat Apa itu pg_cron? dalam dokumentasi pg_cron.

Topik

- Parameter untuk mengelola ekstensi pg_cron
- Referensi fungsi: cron.schedule
- Referensi fungsi: cron.unschedule
- Tabel untuk menjadwalkan pekerjaan dan menangkap status

Parameter untuk mengelola ekstensi pg_cron

Berikut adalah daftar parameter yang mengontrol perilaku ekstensi `pg_cron`.

Parameter	Deskripsi
cron.database_name	Basis data tempat metadata `pg_cron` disimpan.
`cron.host`	Nama host untuk terhubung ke PostgreSQL. Anda tidak dapat mengubah nilai ini.

Parameter	Deskripsi
`cron.log_run`	Catat setiap pekerjaan yang berjalan di tabel `job_run_details` . Nilainya adalah `on` atau `off`. Untuk informasi selengkapnya, lihat Tabel untuk menjadwalkan pekerjaan dan menangkap status .
`cron.log_statement`	Catat semua pernyataan cron sebelum menjalankannya. Nilainya adalah `on` atau `off`.
`cron.max_running_jobs`	Jumlah maksimum pekerjaan yang dapat dijalankan secara bersamaan.
`cron.use_background_workers`	Gunakan pekerja latar belakang, bukan sesi klien. Anda tidak dapat mengubah nilai ini.

Gunakan perintah SQL berikut untuk menampilkan parameter ini dan nilainya.

```
postgres=> SELECT name, setting, short_desc FROM pg_settings WHERE name LIKE 'cron.%'
 ORDER BY name;
```

Referensi fungsi: cron.schedule

Fungsi ini menjadwalkan pekerjaan cron. Pada mulanya, pekerjaan dijadwalkan di basis data `postgres` default. Fungsi tersebut menampilkan nilai `bigint` yang mewakili ID pekerjaan. Untuk menjadwalkan pekerjaan agar berjalan di basis data lain dalam instans DB PostgreSQL Anda, lihat contohnya di Menjadwalkan pekerjaan cron untuk basis data selain basis data default.

Fungsi ini memiliki dua format sintaks.

Sintaksis

```
cron.schedule (job_name,
    schedule,
    command
);

cron.schedule (schedule,
    command
```

```
);
```

Parameter

Parameter	Deskripsi
`job_name`	Nama pekerjaan cron.
`schedule`	Teks yang menunjukkan jadwal untuk pekerjaan cron. Formatnya adalah format cron standar.
`command`	Teks perintah yang akan dijalankan.

Contoh-contoh

```
postgres=> SELECT cron.schedule ('test','0 10 * * *', 'VACUUM pgbench_history');
 schedule
----------
      145
(1 row)

postgres=> SELECT cron.schedule ('0 15 * * *', 'VACUUM pgbench_accounts');
 schedule
----------
      146
(1 row)
```

Referensi fungsi: cron.unschedule

Fungsi ini menghapus pekerjaan cron. Anda dapat menentukan `job_name` atau `job_id`. Suatu kebijakan memastikan bahwa Anda adalah pemilik guna menghapus jadwal untuk pekerjaan. Fungsi ini menampilkan Boolean yang menunjukkan keberhasilan atau kegagalan.

Fungsi tersebut memiliki format sintaks berikut.

Sintaksis

```
cron.unschedule (job_id);
```

```
cron.unschedule (job_name);
```

Parameter

Parameter	Deskripsi
`job_id`	ID pekerjaan yang ditampilkan dari fungsi `cron.schedule` saat pekerjaan cron dijadwalkan.
`job_name`	Nama pekerjaan cron yang dijadwalkan dengan fungsi `cron.schedule` .

Contoh-contoh

```
postgres=> SELECT cron.unschedule(108);
 unschedule
------------
 t
(1 row)

postgres=> SELECT cron.unschedule('test');
 unschedule
------------
 t
(1 row)
```

Tabel untuk menjadwalkan pekerjaan dan menangkap status

Tabel berikut digunakan untuk menjadwalkan pekerjaan cron dan mencatat bagaimana pekerjaan tersebut diselesaikan.

Tabel	Deskripsi
`cron.job`	Berisi metadata tentang setiap pekerjaan yang dijadwalkan. Sebagian besar interaksi dengan tabel ini akan dilakukan menggunakan fungsi `cron.schedule` dan `cron.unschedule` .

Tabel	Deskripsi
	Important Sebaiknya Anda tidak memberikan pembaruan atau memasukkan hak istimewa secara langsung ke tabel ini. Dengan begitu, pengguna dapat memperbarui kolom `username` agar berjalan sebagai `rds-superuser` .
`cron.job_run_details`	Berisi informasi historis tentang pekerjaan terjadwal terdahulu yang dijalankan. Hal ini berguna untuk menyelidiki status, pesan kembali, dan waktu mulai dan akhir dari pekerjaan yang berjalan. **Note** Untuk mencegah tabel ini berkembang tanpa batas waktu, bersihkan secara teratur. Sebagai contoh, lihat Membersihkan tabel riwayat pg_cron.

Menggunakan pglogical untuk menyinkronkan data di seluruh instans

Semua versi RDS for PostgreSQL yang tersedia saat ini mendukung ekstensi `pglogical`. Ekstensi pglogical mendahului fitur replikasi logis yang mirip secara fungsional yang diperkenalkan oleh PostgreSQL di versi 10. Untuk informasi selengkapnya, lihat Melakukan replikasi logis untuk Amazon RDS for PostgreSQL.

Ekstensi `pglogical` ini mendukung replikasi logis antara dua atau lebih Instans DB RDS for PostgreSQL. Ini juga mendukung replikasi antara versi PostgreSQL yang berbeda, dan antara basis data yang berjalan pada instans RDS for PostgreSQL DB dan klaster DB Aurora PostgreSQL. Ekstensi `pglogical` menggunakan model berlangganan penerbitan untuk mereplikasi perubahan pada tabel dan objek lain, seperti urutan, dari penerbit ke pelanggan. Itu bergantung pada slot replikasi untuk memastikan bahwa perubahan disinkronkan dari simpul penerbit ke simpul pelanggan, didefinisikan sebagai berikut.

- Simpul penerbit adalah instans DB RDS for PostgreSQL yang merupakan sumber data yang akan direplikas ke simpul lain. Simpul penerbit mendefinisikan tabel yang akan direplikasi dalam kumpulan publikasi.
- Simpul pelanggan adalah instans DB RDS for PostgreSQL yang menerima pembaruan WAL dari penerbit. Pelanggan membuat langganan untuk terhubung ke penerbit dan mendapatkan data WAL yang diterjemahkan. Ketika pelanggan membuat langganan, slot replikasi dibuat pada simpul penerbit.

Berikut ini, Anda dapat menemukan informasi tentang cara mengatur ekstensi `pglogical`.

Topik

- Persyaratan dan batasan untuk ekstensi pglogis
- Menyiapkan ekstensi pglogical
- Menyiapkan replikasi logis untuk instans DB RDS for PostgreSQL
- Membangun kembali replikasi logis setelah peningkatan besar
- Mengelola slot replikasi logis untuk RDS for PostgreSQL
- Referensi parameter untuk ekstensi pglogical

Persyaratan dan batasan untuk ekstensi pglogis

Semua rilis RDS for PostgreSQL yang tersedia saat ini untuk mendukung ekstensi `pglogical`.

Baik simpul penerbit maupun simpul pelanggan harus disiapkan untuk replikasi logis.

Tabel yang ingin Anda replikasi dari pelanggan ke penerbit harus memiliki nama yang sama dan skema yang sama. Tabel ini juga harus berisi kolom yang sama, dan kolom harus menggunakan tipe data yang sama. Tabel penerbit dan pelanggan harus memiliki kunci primer yang sama. Kami menyarankan Anda hanya menggunakan PRIMARY KEY sebagai kendala unik.

Tabel pada simpul pelanggan dapat memiliki kendala yang lebih permisif daripada yang ada di simpul penerbit untuk kendala CHECK dan kendala NOT NULL.

Ekstensi `pglogical` ini menyediakan fitur seperti replikasi dua arah yang tidak didukung oleh fitur replikasi logis yang dibangun ke dalam PostgreSQL (versi 10 dan lebih tinggi). Untuk informasi lebih lanjut, lihat PostgreSQL bi-directional replication using pglogical.

Menyiapkan ekstensi pglogical

Untuk menyiapkan ekstensi `pglogical` pada instans DB RDS for PostgreSQL, Anda menambahkan `pglogical` ke pustaka bersama pada grup parameter DB khusus untuk instans DB RDS for PostgreSQL. Anda juga perlu mengatur nilai parameter `rds.logical_replication` ke `1`, untuk mengaktifkan penguraian kode logis. Akhirnya, Anda membuat ekstensi di basis data. Anda dapat menggunakan AWS Management Console atau AWS CLI untuk tugas-tugas ini.

Anda harus memiliki izin sebagai peran `rds_superuser` untuk melakukan tugas-tugas ini.

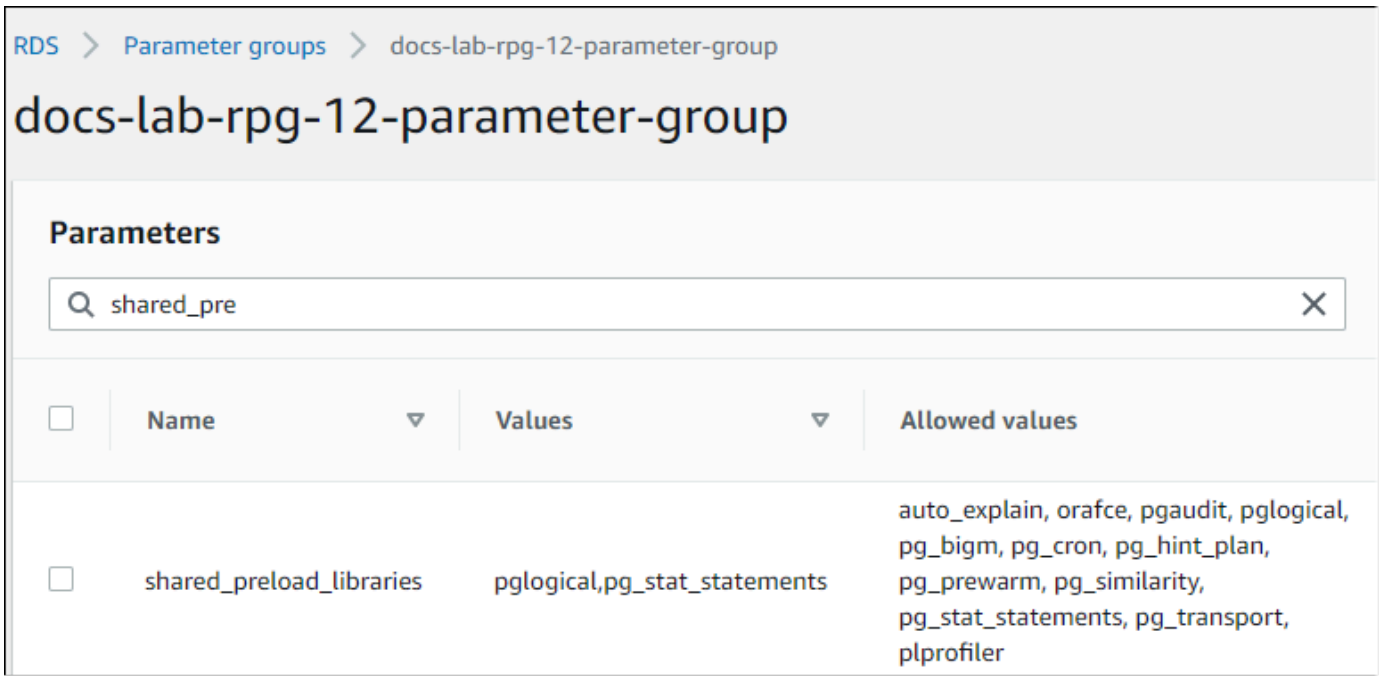
Langkah-langkah berikut mengasumsikan bahwa instans DB RDS for PostgreSQL Anda dikaitkan dengan grup parameter DB khusus. Untuk informasi cara membuat grup parameter DB khusus, lihat Bekerja dengan grup parameter.

Konsol

Menyiapkan ekstensi pglogical

1. Masuk ke AWS Management Console lalu buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih instans DB RDS for PostgreSQL Anda.
3. Buka tab Konfigurasi untuk Instans DB RDS for PostgreSQL. Di antara detail Instans, temukan tautan Grup parameter.
4. Pilih tautan untuk membuka parameter kustom yang terkait dengan Instans DB RDS for PostgreSQL.

5. Di kolom pencarian Parameter, ketik `shared_pre` untuk menemukan parameter `shared_preload_libraries`.
6. Pilih Edit parameter untuk mengakses nilai properti.
7. Tambahkan `pglogical` ke daftar di kolom Nilai. Gunakan koma untuk memisahkan item dalam daftar nilai.



8. Temukan parameter `rds.logical_replication` dan atur ke `1`, untuk mengaktifkan replikasi logis.
9. Boot ulang instans DB RDS for PostgreSQL DB Anda sehingga perubahan Anda akan berlaku.
10. Ketika instans tersedia, Anda dapat menggunakan `psql` (atau pgAdmin) untuk terhubung ke instans DB RDS for PostgreSQL.

```
psql --host=111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password --dbname=labdb
```

11. Untuk memverifikasi bahwa pglogical diinisialisasi, jalankan perintah berikut.

```
SHOW shared_preload_libraries;
shared_preload_libraries
--------------------------
rdsutils,pglogical
(1 row)
```

12. Verifikasi pengaturan yang memungkinkan penguraian kode logis, sebagai berikut.

```
SHOW wal_level;
wal_level
-----------
 logical
(1 row)
```

13. Buat ekstensi, sebagai berikut.

```
CREATE EXTENSION pglogical;
EXTENSION CREATED
```

14. Pilih Simpan perubahan
15. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
16. Pilih instans DB RDS for PostgreSQL dari daftar Basis Data untuk memilihnya, lalu pilih Boot ulang dari menu Tindakan.

AWS CLI

Menyiapkan ekstensi pglogical

Untuk mengatur pglogical menggunakanAWS CLI, Anda memanggil modify-db-parameter-groupoperasi untuk memodifikasi parameter tertentu dalam grup parameter kustom Anda seperti yang ditunjukkan dalam prosedur berikut.

1. Gunakan perintah AWS CLI berikut untuk menambah `pglogical` ke parameter `shared_preload_libraries`.

```
aws rds modify-db-parameter-group \
   --db-parameter-group-name custom-param-group-name \
   --parameters
 "ParameterName=shared_preload_libraries,ParameterValue=pglogical,ApplyMethod=pending-
reboot" \
   --region aws-region
```

2. Gunakan perintah AWS CLI berikut untuk mengatur `rds.logical_replication` ke 1 untuk mengaktifkan kemampuan penguraian kode logis untuk Instans DB RDS for PostgreSQL.

```
aws rds modify-db-parameter-group \
   --db-parameter-group-name custom-param-group-name \
```

```
   --parameters
 "ParameterName=rds.logical_replication,ParameterValue=1,ApplyMethod=pending-
reboot" \
   --region aws-region
```

3. Gunakan AWS CLI perintah berikut untuk melakukan boot ulang instans DB RDS for PostgreSQL sehingga pustaka pglogical dapat diinisialisasi.

```
aws rds reboot-db-instance \
    --db-instance-identifier your-instance \
    --region aws-region
```

4. Ketika instans tersedia, gunakan `psql` untuk terhubung ke instans DB RDS for PostgreSQL.

```
psql --host=111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password --dbname=labdb
```

5. Buat ekstensi, sebagai berikut.

```
CREATE EXTENSION pglogical;
EXTENSION CREATED
```

6. Boot ulang menggunakan perintah berikut. AWS CLI

```
aws rds reboot-db-instance \
    --db-instance-identifier your-instance \
    --region aws-region
```

Menyiapkan replikasi logis untuk instans DB RDS for PostgreSQL

Prosedur berikut menunjukkan cara memulai replikasi logis antara dua instans DB RDS for PostgreSQL. Langkah-langkah mengasumsikan bahwa sumber (penerbit) dan target (pelanggan) memiliki ekstensi `pglogical` yang disiapkan seperti yang dijelaskan dalam Menyiapkan ekstensi pglogical.

Untuk membuat simpul penerbit dan menentukan tabel untuk direplikasi

Langkah-langkah ini mengasumsikan bahwa instans DB RDS for PostgreSQL memiliki basis data yang memiliki satu atau beberapa tabel yang ingin direplikasi ke simpul lain. Anda perlu membuat ulang struktur tabel dari penerbit pada pelanggan, jadi pertama-tama, dapatkan struktur tabel jika

perlu. Anda dapat melakukannya dengan menggunakan `\d` *`tablename`* metacommand `psql` dan kemudian membuat tabel yang sama pada instans pelanggan. Prosedur berikut membuat tabel contoh pada penerbit (sumber) untuk tujuan demonstrasi.

1. Gunakan `psql` untuk terhubung ke instans yang memiliki tabel yang ingin Anda gunakan sebagai sumber untuk pelanggan.

   ```
   psql --host=source-instance.aws-region.rds.amazonaws.com --port=5432 --
   username=postgres --password --dbname=labdb
   ```

   Jika Anda tidak memiliki tabel yang ingin Anda tiru, Anda dapat membuat tabel contoh sebagai berikut.

   a. Buat contoh tabel menggunakan pernyataan SQL berikut.

      ```
      CREATE TABLE docs_lab_table (a int PRIMARY KEY);
      ```

   b. Mengisi tabel dengan data yang dihasilkan dengan menggunakan pernyataan SQL berikut.

      ```
      INSERT INTO docs_lab_table VALUES (generate_series(1,5000));
      INSERT 0 5000
      ```

   c. Verifikasi bahwa data ada dalam tabel dengan menggunakan pernyataan SQL berikut.

      ```
      SELECT count(*) FROM docs_lab_table;
      ```

2. Identifikasi instans DB RDS for PostgreSQL ini sebagai simpul penerbit, sebagai berikut.

   ```
   SELECT pglogical.create_node(
       node_name := 'docs_lab_provider',
       dsn := 'host=source-instance.aws-region.rds.amazonaws.com port=5432
    dbname=labdb');
    create_node
   -------------
      3410995529
   (1 row)
   ```

3. Tambahkan tabel yang ingin Anda replikasi ke set replikasi default. Untuk informasi set replikasi selengkapnya, lihat Replication sets dalam dokumentasi pglogical.

```
SELECT pglogical.replication_set_add_table('default', 'docs_lab_table', 'true',
 NULL, NULL);
 replication_set_add_table
  --------------------------
  t
  (1 row)
```

Penyiapan simpul penerbit selesai. Anda sekarang dapat menyiapkan simpul pelanggan untuk menerima pembaruan dari penerbit.

Untuk menyiapkan simpul pelanggan dan membuat langganan untuk menerima pembaruan

Langkah-langkah ini mengasumsikan bahwa instans DB RDS for PostgreSQL telah disiapkan dengan ekstensi `pglogical`. Untuk informasi selengkapnya, lihat Menyiapkan ekstensi pglogical.

1. Gunakan `psql` untuk terhubung ke instans yang ingin Anda terima pembaruan dari penerbit.

   ```
   psql --host=target-instance.aws-region.rds.amazonaws.com --port=5432 --
   username=postgres --password --dbname=labdb
   ```

2. Pada pelanggan instans DB RDS for PostgreSQL, buat tabel yang sama yang ada pada penerbit. Untuk contoh ini, tabelnya adalah `docs_lab_table`. Anda dapat membuat tabel sebagai berikut.

   ```
   CREATE TABLE docs_lab_table (a int PRIMARY KEY);
   ```

3. Verifikasi bahwa tabel ini kosong.

   ```
   SELECT count(*) FROM docs_lab_table;
    count
   -------
     0
   (1 row)
   ```

4. Identifikasi instans DB RDS for PostgreSQL ini sebagai simpul pelanggan, sebagai berikut.

   ```
   SELECT pglogical.create_node(
       node_name := 'docs_lab_target',
       dsn := 'host=target-instance.aws-region.rds.amazonaws.com port=5432
    sslmode=require dbname=labdb user=postgres password=********');
   ```

```
 create_node
-------------
   2182738256
(1 row)
```

5. Buat langganan.

```
SELECT pglogical.create_subscription(
   subscription_name := 'docs_lab_subscription',
   provider_dsn := 'host=source-instance.aws-region.rds.amazonaws.com port=5432
 sslmode=require dbname=labdb user=postgres password=*******',
   replication_sets := ARRAY['default'],
   synchronize_data := true,
   forward_origins := '{}' );
 create_subscription
---------------------
1038357190
(1 row)
```

Ketika Anda menyelesaikan langkah ini, data dari tabel pada penerbit dibuat dalam tabel pada pelanggan. Anda dapat memverifikasi bahwa ini telah terjadi dengan menggunakan query SQL berikut.

```
SELECT count(*) FROM docs_lab_table;
 count
-------
  5000
(1 row)
```

Dari titik ini ke depan, perubahan yang dilakukan pada tabel pada penerbit direplikasi ke tabel pada pelanggan.

Membangun kembali replikasi logis setelah peningkatan besar

Sebelum Anda dapat melakukan peningkatan versi utama dari instans DB RDS for PostgreSQL DB yang disiapkan sebagai simpul penerbit untuk replikasi logis, Anda harus menghapus semua slot replikasi, bahkan yang tidak aktif. Kami menyarankan Anda untuk mengalihkan sementara transaksi basis data dari simpul penerbit, menghapus sementara slot replikasi, meningkatkan instans DB RDS foe PostgreSQL, dan kemudian membangun kembali dan memulai ulang replikasi.

Slot replikasi hanya dihosting di simpul penerbit. Simpul pelanggan RDS for PostgreSQL dalam skenario replikasi logis tidak memiliki slot untuk dihapus sementara, tetapi tidak dapat ditingkatkan ke versi utama saat ditetapkan sebagai simpul pelanggan dengan berlangganan penerbit. Sebelum meningkatkan simpul pelanggan RDS for PostgreSQL, hapus sementara langganan dan simpul. Untuk informasi selengkapnya, lihat Mengelola slot replikasi logis untuk RDS for PostgreSQL.

Menentukan bahwa replikasi logis telah terganggu

Anda dapat menentukan bahwa proses replikasi telah terganggu dengan menanyakan simpul penerbit atau simpul pelanggan, sebagai berikut.

Untuk memeriksa simpul penerbit

- Gunakan `psql` untuk terhubung ke simpul penerbit, dan kemudian kueri fungsi `pg_replication_slots`. Perhatikan nilai di kolom aktif. Biasanya, ini akan kembali `t` (benar), menunjukkan bahwa replikasi aktif. Jika kueri kembali `f` (salah), ini merupakan indikasi bahwa replikasi ke pelanggan telah berhenti.

```
SELECT slot_name,plugin,slot_type,active FROM pg_replication_slots;
                    slot_name              |      plugin      | slot_type | active
-------------------------------------------+------------------+-----------+--------
 pgl_labdb_docs_labcb4fa94_docs_lab3de412c | pglogical_output | logical   | f
(1 row)
```

Untuk memeriksa simpul pelanggan

Pada simpul pelanggan, Anda dapat memeriksa status replikasi dengan tiga cara berbeda.

- Lihatlah log PostgreSQL pada simpul pelanggan untuk menemukan pesan kegagalan. Log dapat mengidentifikasi kegagalan dengan pesan yang menyertakan kode keluar 1, seperti yang ditunjukkan berikut.

```
2022-07-06 16:17:03 UTC::@:[7361]:LOG: background worker "pglogical apply
 16404:2880255011" (PID 14610) exited with exit code 1
2022-07-06 16:19:44 UTC::@:[7361]:LOG: background worker "pglogical apply
 16404:2880255011" (PID 21783) exited with exit code 1
```

- Kueri fungsi `pg_replication_origin`. Hubungkan ke basis data pada simpul pelanggan menggunakan `psql` dan query fungsi `pg_replication_origin`, sebagai berikut.

```
SELECT * FROM pg_replication_origin;
 roident | roname
---------+--------
(0 rows)
```

Kumpulan hasil kosong berarti replikasi telah terganggu. Umumnya, Anda akan melihat output seperti berikut ini.

```
   roident |                       roname
  ---------+-----------------------------------------------------
         1 | pgl_labdb_docs_labcb4fa94_docs_lab3de412c
  (1 row)
```

- Kueri fungsi `pglogical.show_subscription_status` seperti yang ditunjukkan pada contoh berikut.

```
SELECT subscription_name,status,slot_name FROM pglogical.show_subscription_status();
     subscription_name | status |              slot_name
---====---------------+--------+-----------------------------------
 docs_lab_subscription | down   | pgl_labdb_docs_labcb4fa94_docs_lab3de412c
(1 row)
```

Output ini menunjukkan bahwa replikasi telah terganggu. Statusnya adalah down. Biasanya, output menunjukkan status sebagai `replicating`.

Jika proses replikasi logis Anda telah terganggu, Anda dapat membangun kembali replikasi dengan mengikuti langkah-langkah ini.

Untuk membangun kembali replikasi logis antara simpul penerbit dan pelanggan

Untuk membangun kembali replikasi, pertama-tama Anda memutuskan sambungan pelanggan dari simpul penerbit dan kemudian membangun kembali langganan, seperti yang diuraikan dalam langkah-langkah ini.

1. Hubungkan ke simpul pelanggan menggunakan `psql` sebagai berikut.

```
psql --host=222222222222.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password --dbname=labdb
```

2. Nonaktifkan langganan dengan menggunakan fungsi `pglogical.alter_subscription_disable`.

```
SELECT pglogical.alter_subscription_disable('docs_lab_subscription',true);
 alter_subscription_disable
----------------------------
 t
(1 row)
```

3. Dapatkan identifier simpul penerbit dengan menanyakan `pg_replication_origin`, sebagai berikut.

```
SELECT * FROM pg_replication_origin;
 roident |               roname
---------+--------------------------------------
       1 | pgl_labdb_docs_labcb4fa94_docs_lab3de412c
(1 row)
```

4. Gunakan respons dari langkah sebelumnya dengan perintah `pg_replication_origin_create` untuk menetapkan pengenal yang dapat digunakan oleh langganan saat dibuat kembali.

```
SELECT pg_replication_origin_create('pgl_labdb_docs_labcb4fa94_docs_lab3de412c');
  pg_replication_origin_create
------------------------------
                             1
(1 row)
```

5. Nyalakan langganan dengan meneruskan namanya dengan status `true`, seperti yang ditunjukkan dalam contoh berikut.

```
SELECT pglogical.alter_subscription_enable('docs_lab_subscription',true);
  alter_subscription_enable
---------------------------
 t
(1 row)
```

Periksa status simpul. Statusnya harus `replicating` seperti yang ditunjukkan dalam contoh ini.

```
SELECT subscription_name,status,slot_name
```

```
  FROM pglogical.show_subscription_status();
            subscription_name |   status    |              slot_name
------------------------------+-------------+-----------------------------------
 docs_lab_subscription         | replicating |
 pgl_labdb_docs_lab98f517b_docs_lab3de412c
(1 row)
```

Periksa status slot replikasi pelanggan pada simpul penerbit. Kolom `active` slot harus kembali `t` (benar), menunjukkan bahwa replikasi telah dibuat kembali.

```
SELECT slot_name,plugin,slot_type,active
  FROM pg_replication_slots;
                   slot_name              |      plugin      | slot_type | active
------------------------------------------+------------------+-----------+--------
 pgl_labdb_docs_lab98f517b_docs_lab3de412c | pglogical_output | logical   | t
(1 row)
```

Mengelola slot replikasi logis untuk RDS for PostgreSQL

Sebelum Anda dapat melakukan peningkatan versi utama dari instans DB RDS for PostgreSQL yang disiapkan sebagai simpul penerbit untuk replikasi logis, Anda harus menghapus semua slot replikasi, bahkan yang tidak aktif. Proses pra-pemeriksaan peningkatan versi utama akan memberi tahu Anda bahwa peningkatan tidak dapat dilanjutkan sampai slot dihapuskan sementara.

Untuk menghapus sementara slot dari instans DB RDS for PostgreSQL Anda, pertama-tama hapus sementara langganan dan kemudian hapus sementara slotnya.

Untuk mengidentifikasi slot replikasi yang dibuat menggunakan ekstensi `pglogical`, masuk ke setiap basis data dan dapatkan nama simpul. Saat Anda menanyakan simpul pelanggan, Anda mendapatkan penerbit dan simpul pelanggan dalam output, seperti yang ditunjukkan dalam contoh ini.

```
SELECT * FROM pglogical.node;
node_id   |     node_name
------------+------------------
 2182738256 | docs_lab_target
 3410995529 | docs_lab_provider
(2 rows)
```

Anda bisa mendapatkan detail tentang langganan dengan kueri berikut.

```
SELECT sub_name,sub_slot_name,sub_target
  FROM pglogical.subscription;
 sub_name |         sub_slot_name          | sub_target
----------+--------------------------------+------------
  docs_lab_subscription     | pgl_labdb_docs_labcb4fa94_docs_lab3de412c | 2182738256
(1 row)
```

Anda sekarang dapat menghapus sementara langganan, sebagai berikut.

```
SELECT pglogical.drop_subscription(subscription_name := 'docs_lab_subscription');
 drop_subscription
-------------------
                 1
(1 row)
```

Setelah menghapus sementara langganan, Anda dapat menghapus simpul.

```
SELECT pglogical.drop_node(node_name := 'docs-lab-subscriber');
 drop_node
-----------
 t
(1 row)
```

Anda dapat memverifikasi bahwa simpul tidak ada lagi, sebagai berikut.

```
SELECT * FROM pglogical.node;
 node_id | node_name
---------+-----------
(0 rows)
```

Referensi parameter untuk ekstensi pglogical

Dalam tabel Anda dapat menemukan parameter yang terkait dengan ekstensi `pglogical`. Parameter seperti `pglogical.conflict_log_level` dan `pglogical.conflict_resolution` digunakan untuk menangani konflik pembaruan. Konflik dapat muncul ketika perubahan dilakukan secara lokal ke tabel yang sama yang berlangganan perubahan dari penerbit. Konflik juga dapat terjadi selama berbagai skenario, seperti replikasi dua arah atau ketika beberapa pelanggan mereplikasi dari penerbit yang sama. Untuk informasi lebih lanjut, lihat PostgreSQL bi-directional replication using pglogical.

Parameter	Deskripsi
pglogical.batch_inserts	Melakukan penyisipan batch jika memungkinkan. Tidak diatur secara default. Ubah ke '1' untuk menghidupkan, '0' untuk mematikan.
pglogical.conflict_log_level	Menetapkan tingkat log yang digunakan untuk mencatat log konflik yang diselesaikan. Nilai string yang didukung adalah debug5, debug4, debug3, debug2, debug1, info, notice, warning, error, log, fatal, panic.
pglogical.conflict_resolution	Menetapkan metode untuk digunakan untuk menyelesa ikan konflik ketika konflik dapat diselesaikan. Nilai string yang didukung adalah kesalahan, apply_remote, keep_local, last_update_wins, first_update_wins.
pglogical.extra_connection_ options	Opsi koneksi untuk ditambahkan ke semua koneksi simpul peer.
pglogical.synchronous_commit	Nilai komit sinkron spesifik pglogical
pglogical.use_spi	Gunakan SPI (antarmuka pemrograman server) alih-alih API tingkat rendah untuk menerapkan perubahan. Atur ke '1' untuk menghidupkan, '0' untuk mematikan. Untuk informasi SPI selengkapnya, lihat Server Programming Interface dalam dokumentasi PostgreSQL.

Menggunakan pgactive untuk mendukung replikasi aktif-aktif

Ekstensi `pgactive` menggunakan replikasi aktif-aktif untuk mendukung dan mengoordinasikan operasi penulisan pada beberapa RDS untuk basis data PostgreSQL. Amazon RDS for `pgactive` PostgreSQL mendukung ekstensi pada versi berikut:

- RDS untuk PostgreSQL 16.1 dan versi 16 yang lebih tinggi
- RDS untuk PostgreSQL 15.4-R2 dan versi 15 yang lebih tinggi
- RDS untuk PostgreSQL 14.10 dan versi 14 yang lebih tinggi
- RDS untuk PostgreSQL 13.13 dan versi 13 yang lebih tinggi
- RDS untuk PostgreSQL 12.17 dan versi 12 yang lebih tinggi
- RDS untuk PostgreSQL 11.22

> **Note**
>
> Ketika ada operasi tulis pada lebih dari satu basis data dalam konfigurasi replikasi, konflik mungkin terjadi. Untuk informasi selengkapnya, lihat Menangani konflik dalam replikasi aktif-aktif

Topik

Menginisialisasi kemampuan ekstensi pgactive

Untuk menginisialisasi kemampuan ekstensi `pgactive` pada instans DB RDS for PostgreSQL, tetapkan nilai parameter `rds.enable_pgactive` ke `1` dan kemudian buat ekstensi dalam basis

data. Melakukannya secara otomatis menyalakan parameter `rds.logical_replication` dan `track_commit_timestamp` lalu menetapkan nilai `wal_level` ke `logical`.

Anda harus memiliki izin sebagai peran `rds_superuser` untuk melakukan tugas-tugas ini.

Anda dapat menggunakan AWS Management Console atau AWS CLI untuk membuat RDS yang diperlukan untuk instance PostgreSQL DB. Langkah-langkah berikut mengasumsikan bahwa instans DB RDS for PostgreSQL Anda dikaitkan dengan grup parameter DB khusus. Untuk informasi tentang cara membuat grup parameter DB kusto, lihat Bekerja dengan grup parameter.

Konsol

Untuk menginisialisasi kemampuan ekstensi pgactive

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih instans DB RDS for PostgreSQL.
3. Buka tab Konfigurasi untuk instans DB RDS for PostgreSQL. Dalam detail instans, temukan tautan grup parameter instans DB.
4. Pilih tautan untuk membuka parameter khusus yang terkait dengan instans DB RDS for PostgreSQL.
5. Temukan parameter `rds.enable_pgactive`, lalu atur ke 1 untuk menginisialisasi kemampuan `pgactive`.
6. Pilih Simpan perubahan.
7. Dalam panel navigasi yang ada pada konsol Amazon RDS, pilih Basis Data.
8. Pilih instans DB RDS for PostgreSQL, kemudian pilih Boot ulang dari menu Tindakan.
9. Konfirmasikan boot ulang instans DB sehingga perubahan Anda berlaku.
10. Ketika instans tersedia, Anda dapat menggunakan `psql` klien PostgreSQL lainnya agar terhubung ke instans DB RDS for PostgreSQL.

    Contoh berikut mengasumsikan bahwa instans DB RDS for PostgreSQL memiliki basis data default bernama *`postgres`*.

    ```
    psql --host=mydb.111122223333.aws-region.rds.amazonaws.com --port=5432 --
    username=master username --password --dbname=postgres
    ```

11. Untuk memverifikasi bahwa pgactive telah diinisialisasi, jalankan perintah berikut.

```
postgres=>SELECT setting ~ 'pgactive'
FROM pg_catalog.pg_settings
WHERE name = 'shared_preload_libraries';
```

Jika `pgactive` berada di `shared_preload_libraries`, perintah sebelumnya akan mengembalikan yang berikut:

```
?column?
----------
 t
```

12. Buat ekstensi, sebagai berikut.

```
postgres=> CREATE EXTENSION pgactive;
```

AWS CLI

Untuk menginisialisasi kemampuan ekstensi pgactive

Untuk menginisialisasi `pgactive` penggunaan AWS CLI, panggil modify-db-parameter-groupoperasi untuk memodifikasi parameter tertentu dalam grup parameter kustom Anda seperti yang ditunjukkan dalam prosedur berikut.

1. Gunakan AWS CLI perintah berikut untuk mengatur `rds.enable_pgactive` untuk menginisialisasi `pgactive` kemampuan 1 untuk RDS untuk PostgreSQL DB instance.

```
postgres=>aws rds modify-db-parameter-group \
   --db-parameter-group-name custom-param-group-name \
   --parameters
 "ParameterName=rds.enable_pgactive,ParameterValue=1,ApplyMethod=pending-reboot" \
   --region aws-region
```

2. Gunakan AWS CLI perintah berikut untuk me-reboot RDS untuk PostgreSQL DB instance sehingga perpustakaan diinisialisasi. `pgactive`

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

```
    --region aws-region
```

3. Ketika instans tersedia, gunakan `psql` untuk terhubung ke instans DB RDS for PostgreSQL.

```
psql --host=mydb.111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=master user --password --dbname=postgres
```

4. Buat ekstensi, sebagai berikut.

```
postgres=> CREATE EXTENSION pgactive;
```

Menyiapkan replikasi aktif-aktif untuk instans DB RDS for PostgreSQL

Prosedur berikut menunjukkan cara memulai replikasi aktif-aktif antara dua instans DB RDS for PostgreSQL yang menjalankan PostgreSQL 15.4 atau lebih tinggi di wilayah yang sama. Untuk menjalankan contoh ketersediaan tinggi beberapa wilayah, Anda perlu menerapkan instans Amazon RDS for PostgreSQL di dua wilayah berbeda dan menyiapkan VPC Peering. Untuk informasi selengkapnya, lihat VPC peering.

> **Note**
>
> Mengirim lalu lintas antar beberapa wilayah dapat menimbulkan biaya tambahan.

Langkah-langkah ini mengasumsikan bahwa instans DB RDS for PostgreSQL telah disiapkan dengan ekstensi `pgactive`. Untuk informasi selengkapnya, lihat Menginisialisasi kemampuan ekstensi pgactive.

Mengonfigurasi instans DB RDS for PostgreSQL dengan ekstensi **`pgactive`**

Contoh berikut menggambarkan cara grup `pgactive` dibuat, bersama dengan langkah-langkah lain yang diperlukan untuk membuat ekstensi `pgactive` pada instans DB RDS for PostgreSQL.

1. Gunakan `psql` atau alat klien lain agar terhubung ke instans DB RDS for PostgreSQL.

```
psql --host=firstinstance.111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=master username --password --dbname=postgres
```

2. Buat basis data pada instans RDS for PostgreSQL menggunakan perintah berikut:

```
postgres=> CREATE DATABASE app;
```

3. Alihkan koneksi ke basis data baru menggunakan perintah berikut:

```
\c app
```

4. Untuk memeriksa apakah parameter `shared_preload_libraries` berisi `pgactive`, jalankan perintah berikut:

```
app=>SELECT setting ~ 'pgactive' FROM pg_catalog.pg_settings WHERE name =
 'shared_preload_libraries';
```

```
 ?column?
----------
 t
```

5. Buat dan isi tabel menggunakan pernyataan SQL berikut:

   a. Buat contoh tabel menggunakan pernyataan SQL berikut.

   ```
   app=> CREATE SCHEMA inventory;
   CREATE TABLE inventory.products (
   id int PRIMARY KEY, product_name text NOT NULL,
   created_at timestamptz NOT NULL DEFAULT CURRENT_TIMESTAMP);
   ```

   b. Mengisi tabel dengan beberapa contoh data yang dihasilkan dengan menggunakan pernyataan SQL berikut.

   ```
   app=> INSERT INTO inventory.products (id, product_name)
   VALUES (1, 'soap'), (2, 'shampoo'), (3, 'conditioner');
   ```

   c. Verifikasi bahwa data ada dalam tabel dengan menggunakan pernyataan SQL berikut.

   ```
    app=>SELECT count(*) FROM inventory.products;

    count
   -------
    3
   ```

6. Buat ekstensi `pgactive` pada basis data yang ada.

```
app=> CREATE EXTENSION pgactive;
```

7. Buat dan inisialisasi grup pgactive menggunakan perintah berikut:

```
app=> SELECT pgactive.pgactive_create_group(
    node_name := 'node1-app',
    node_dsn := 'dbname=app host=firstinstance.111122223333.aws-
region.rds.amazonaws.com user=master username password=PASSWORD');
```

node1-app adalah nama yang Anda tetapkan untuk mengidentifikasi simpul secara unik dalam grup `pgactive`.

> **Note**
>
> Untuk melakukan langkah ini dengan sukses pada instans DB yang dapat diakses publik, Anda harus mengaktifkan parameter `rds.custom_dns_resolution` dengan menyetelnya ke 1.

8. Untuk memeriksa apakah instans DB sudah siap, gunakan perintah berikut ini:

```
app=> SELECT pgactive.pgactive_wait_for_node_ready();
```

Jika perintah berhasil, Anda dapat melihat output sebagai berikut:

```

pgactive_wait_for_node_ready
-----------------------------
(1 row)
```

Mengonfigurasi instans RDS for PostgreSQL kedua dan bergabung ke grup **`pgactive`**

Contoh berikut menggambarkan cara instans DB RDS for PostgreSQL bergabung ke grup `pgactive`, bersama dengan langkah-langkah lain yang diperlukan untuk membuat ekstensi `pgactive` pada instans DB.

Langkah-langkah ini mengasumsikan bahwa instans DB RDS for PostgreSQL lainnya telah disiapkan dengan ekstensi `pgactive`. Untuk informasi selengkapnya, lihat Menginisialisasi kemampuan ekstensi pgactive.

1. Gunakan `psql` untuk terhubung ke instans yang ingin Anda terima pembaruan dari penerbit.

```
psql --host=secondinstance.111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=master username --password --dbname=postgres
```

2. Buat basis data pada instans DB RDS for PostgreSQL kedua menggunakan perintah berikut:

```
postgres=> CREATE DATABASE app;
```

3. Alihkan koneksi ke basis data baru menggunakan perintah berikut:

```
\c app
```

4. Buat ekstensi `pgactive` pada basis data yang ada.

```
app=> CREATE EXTENSION pgactive;
```

5. Bergabunglah dengan instans DB kedua RDS for PostgreSQL ke grup `pgactive` sebagai berikut.

```
app=> SELECT pgactive.pgactive_join_group(
node_name := 'node2-app',
node_dsn := 'dbname=app host=secondinstance.111122223333.aws-
region.rds.amazonaws.com user=master username password=PASSWORD',
join_using_dsn := 'dbname=app host=firstinstance.111122223333.aws-
region.rds.amazonaws.com user=postgres password=PASSWORD');
```

   node2-app adalah nama yang Anda tetapkan untuk mengidentifikasi simpul secara unik dalam grup `pgactive`.

6. Untuk memeriksa apakah instans DB sudah siap, gunakan perintah berikut ini:

```
app=> SELECT pgactive.pgactive_wait_for_node_ready();
```

   Jika perintah berhasil, Anda dapat melihat output sebagai berikut:

```
pgactive_wait_for_node_ready
-----------------------------
(1 row)
```

Jika basis data RDS for PostgreSQL pertama relatif besar, Anda dapat melihat `pgactive.pgactive_wait_for_node_ready()` yang merilis laporan kemajuan operasi pemulihan. Output akan terlihat serupa dengan yang berikut ini:

```
NOTICE:  restoring database 'app', 6% of 7483 MB complete
NOTICE:  restoring database 'app', 42% of 7483 MB complete
NOTICE:  restoring database 'app', 77% of 7483 MB complete
NOTICE:  restoring database 'app', 98% of 7483 MB complete
NOTICE:  successfully restored database 'app' from node node1-app in
 00:04:12.274956
 pgactive_wait_for_node_ready
-----------------------------
(1 row)
```

Dari titik ini ke depan, `pgactive` menyingkronkan data antara dua instans DB.

7. Anda dapat menggunakan perintah berikut untuk memverifikasi apakah basis data instans DB kedua memiliki data:

```
app=> SELECT count(*) FROM inventory.products;
```

Jika data berhasil disinkronkan, Anda akan melihat output sebagai berikut:

```

 count
-------
 3
```

8. Jalankan perintah berikut ini untuk memasukkan nilai baru:

```
app=> INSERT INTO inventory.products (id, product_name) VALUES ('lotion');
```

9. Hubungkan ke basis data instans DB pertama dan jalankan kueri berikut:

```
app=> SELECT count(*) FROM inventory.products;
```

Jika replikasi aktif-aktif diinisialisasi, output serupa dengan berikut ini:

```
count
-------
 4
```

Melepas dan menghapus instans DB dari grup **pgactive**

Anda dapat melepas dan menghapus instans DB dari grup pgactive menggunakan langkah-langkah berikut:

1. Anda dapat melepas instans DB kedua dari instans DB pertama menggunakan perintah berikut:

   ```
   app=> SELECT * FROM pgactive.pgactive_detach_nodes(ARRAY['node2-app']);
   ```

2. Menghapus ekstensi pgactive dari instans DB kedua menggunakan perintah berikut:

   ```
   app=> SELECT * FROM pgactive.pgactive_remove();
   ```

   Untuk menghapus ekstensi secara paksa:

   ```
   app=> SELECT * FROM pgactive.pgactive_remove(true);
   ```

3. Hapus sementara ekstensi menggunakan perintah berikut ini:

   ```
   app=> DROP EXTENSION pgactive;
   ```

Menangani konflik dalam replikasi aktif-aktif

Ekstensi pgactive bekerja per basis data dan bukan per klaster. Setiap instans DB yang menggunakan pgactive adalah instans independen dan dapat menerima perubahan data dari sumber apa pun. Ketika perubahan dikirim ke instans DB, PostgreSQL mengkomitmennya secara lokal dan kemudian menggunakan pgactive untuk mereplikasi perubahan secara asinkron ke instans DB lainnya. Ketika dua instans DB PostgreSQL memperbarui catatan yang sama pada waktu yang hampir bersamaan, konflik dapat terjadi.

Ekstensi pgactive menyediakan mekanisme untuk deteksi konflik dan resolusi otomatis. Ini akan melacak stempel waktu ketika transaksi dilakukan pada kedua instans DB dan secara otomatis

menerapkan perubahan dengan stempel waktu terbaru. Ekstensi `pgactive` juga melakukan log ketika konflik terjadi dalam tabel `pgactive.pgactive_conflict_history`.

`pgactive.pgactive_conflict_history`Akan terus tumbuh. Anda mungkin ingin menentukan kebijakan pembersihan. Ini dapat dilakukan dengan menghapus beberapa catatan secara teratur atau mendefinisikan skema partisi untuk hubungan ini (dan kemudian melepaskan, menjatuhkan, memotong partisi yang menarik). Untuk menerapkan kebijakan pembersihan secara teratur, salah satu opsi adalah menggunakan `pg_cron` ekstensi. Lihat informasi berikut dari contoh untuk tabel `pg_cron` riwayat, Penjadwalan pemeliharaan dengan ekstensi PostgreSQL pg_cron.

Menangani urutan dalam replikasi aktif-aktif

Sebuah instans DB RDS for PostgreSQL dengan ekstensi `pgactive` menggunakan dua mekanisme urutan yang berbeda untuk menghasilkan nilai unik.

Urutan Global

Untuk menggunakan urutan global, buat urutan lokal dengan pernyataan `CREATE SEQUENCE`. Gunakan `pgactive.pgactive_snowflake_id_nextval(seqname)` alih-alih `usingnextval(seqname)` untuk mendapatkan nilai unik berikutnya dari urutan.

Contoh berikut membuat urutan global:

```
postgres=> CREATE TABLE gstest (
      id bigint primary key,
      parrot text
    );
```

```
postgres=>CREATE SEQUENCE gstest_id_seq OWNED BY gstest.id;
```

```
postgres=> ALTER TABLE gstest \
      ALTER COLUMN id SET DEFAULT \
      pgactive.pgactive_snowflake_id_nextval('gstest_id_seq');
```

Urutan yang dipartisi

Dalam urutan split-step atau partisi, urutan PostgreSQL normal digunakan pada setiap simpul. Setiap urutan bertambah dengan jumlah yang sama dan dimulai pada offset yang berbeda. Misalnya, dengan langkah 100, simpul 1 menghasilkan urutan sebagai 101, 201, 301, dan seterusnya dan

simpul 2 menghasilkan urutan sebagai 102, 202, 302, dan seterusnya. Skema ini bekerja dengan baik bahkan jika simpul tidak dapat berkomunikasi untuk waktu yang lama, tetapi mengharuskan perancang menentukan jumlah simpul maksimum saat membuat skema dan memerlukan konfigurasi per-simpul. Kesalahan dapat dengan mudah menyebabkan urutan yang tumpang tindih.

Hal ini relatif mudah untuk mengonfigurasi pendekatan ini dengan `pgactive` dengan membuat urutan yang diinginkan pada simpul sebagai berikut:

```
CREATE TABLE some_table (generated_value bigint primary key);
```

```
postgres=> CREATE SEQUENCE some_seq INCREMENT 100 OWNED BY some_table.generated_value;
```

```
postgres=> ALTER TABLE some_table ALTER COLUMN generated_value SET DEFAULT
 nextval('some_seq');
```

Kemudian panggil `setval` setiap simpul untuk memberikan nilai awal offset yang berbeda sebagai berikut.

```
postgres=>
-- On node 1
SELECT setval('some_seq', 1);

-- On node 2
SELECT setval('some_seq', 2);
```

Referensi parameter untuk ekstensi pglactive

Anda dapat menggunakan kueri berikut untuk melihat semua parameter yang terkait dengan ekstensi `pgactive`.

```
postgres=> SELECT * FROM pg_settings WHERE name LIKE 'pgactive.%';
```

Mengukur kelambatan replikasi di antara anggota pgaktif

Anda dapat menggunakan kueri berikut untuk melihat lag replikasi di antara `pgactive` anggota. Jalankan kueri ini di setiap `pgactive` node untuk mendapatkan gambaran lengkap.

```
      postgres=# SELECT *, (last_applied_xact_at - last_applied_xact_committs) AS lag
 FROM pgactive.pgactive_node_slots;
-{ RECORD 1 ]----------------
+-----------------------------------------------------------------
node_name                    | node2-app
slot_name                    | pgactive_5_7332551165694385385_0_5__
slot_restart_lsn             | 0/1A898A8
slot_confirmed_lsn           | 0/1A898E0
walsender_active             | t
walsender_pid                | 69022
sent_lsn                     | 0/1A898E0
write_lsn                    | 0/1A898E0
flush_lsn                    | 0/1A898E0
replay_lsn                   | 0/1A898E0
last_sent_xact_id            | 746
last_sent_xact_committs      | 2024-02-06 18:04:22.430376+00
last_sent_xact_at            | 2024-02-06 18:04:22.431359+00
last_applied_xact_id         | 746
last_applied_xact_committs   | 2024-02-06 18:04:22.430376+00
last_applied_xact_at         | 2024-02-06 18:04:52.452465+00
lag                          | 00:00:30.022089
```

Batasan untuk ekstensi pgactive

- Semua tabel memerlukan Kunci Primary, jika bukan Pembaruan dan Hapus tidak akan diperbolehkan. Nilai di kolom Kunci Primary tidak boleh diperbarui.
- Urutan mungkin memiliki celah dan terkadang mungkin tidak mengikuti perintah. Urutan tidak direplikasi. Untuk informasi selengkapnya, lihat Menangani urutan dalam replikasi aktif-aktif.
- DDL dan objek besar tidak direplikasi.
- Indeks unik sekunder dapat menyebabkan divergensi data.
- Kolasi harus identik pada semua simpul dalam grup.
- Penyeimbang beban di seluruh simpul adalah anti-pola.
- Transaksi besar dapat menyebabkan kelambatan replikasi.

Mengurangi bloat dalam tabel dan indeks dengan ekstensi pg_repack

Anda dapat menggunakan `pg_repack` ekstensi untuk menghapus bloat dari tabel dan indeks sebagai alternatif. `VACUUM FULL` Ekstensi didukung pada RDS for PostgreSQL versi 9.6.3 dan yang lebih tinggi. Untuk informasi selengkapnya tentang `pg_repack` ekstensi dan pengemasan ulang tabel lengkap, lihat dokumentasi GitHub proyek.

Tidak seperti`VACUUM FULL`, `pg_repack` ekstensi memerlukan kunci eksklusif (AccessExclusiveLock) hanya untuk waktu yang singkat selama operasi membangun kembali tabel dalam kasus berikut:

- Pembuatan awal tabel log - Tabel log dibuat untuk merekam perubahan yang terjadi selama salinan awal data, seperti yang ditunjukkan pada contoh berikut:

```
postgres=>\dt+ repack.log_*
List of relations
-[ RECORD 1 ]-+----------
Schema        | repack
Name          | log_16490
Type          | table
Owner         | postgres
Persistence   | permanent
Access method | heap
Size          | 65 MB
Description   |
```

- swap-and-drop Fase terakhir.

Untuk sisa operasi pembangunan kembali, hanya perlu `ACCESS SHARE` kunci pada tabel asli untuk menyalin baris dari itu ke tabel baru. Ini membantu operasi INSERT, UPDATE, dan DELETE untuk melanjutkan seperti biasa.

Rekomendasi

Rekomendasi berikut berlaku saat Anda menghapus bloat dari tabel dan indeks menggunakan ekstensi: `pg_repack`

- Lakukan pengemasan ulang selama jam non-bisnis atau melalui jendela pemeliharaan untuk meminimalkan dampaknya terhadap kinerja aktivitas database lainnya.

- Pantau sesi pemblokiran selama aktivitas membangun kembali dan memastikan bahwa tidak ada aktivitas di tabel asli yang berpotensi memblokir`pg_repack`, khususnya selama swap-and-drop fase akhir ketika memerlukan kunci eksklusif pada tabel asli. Untuk informasi selengkapnya, lihat Mengidentifikasi apa yang memblokir kueri.

  Ketika Anda melihat sesi pemblokiran, Anda dapat menghentikannya menggunakan perintah berikut setelah mempertimbangkan dengan cermat. Ini membantu dalam kelanjutan `pg_repack` untuk menyelesaikan pembangunan kembali:

  ```
  SELECT pg_terminate_backend(pid);
  ```

- Saat menerapkan perubahan yang masih harus dibayar dari tabel `pg_repack's` log pada sistem dengan tingkat transaksi yang sangat tinggi, proses penerapan mungkin tidak dapat mengikuti tingkat perubahan. Dalam kasus seperti itu, tidak `pg_repack` akan dapat menyelesaikan proses penerapan. Untuk informasi selengkapnya, lihat Memantau tabel baru selama pengemasan ulang. Jika indeks sangat membengkak, solusi alternatif adalah melakukan pengemasan ulang indeks saja. Ini juga membantu siklus pembersihan indeks VACUUM untuk menyelesaikan lebih cepat.

  Anda dapat melewati fase pembersihan indeks menggunakan VACUUM manual dari PostgreSQL versi 12, dan dilewati secara otomatis selama autovacuum darurat dari PostgreSQL versi 14. Ini membantu VACUUM menyelesaikan lebih cepat tanpa menghilangkan kembung indeks dan hanya dimaksudkan untuk situasi darurat seperti mencegah VACUUM sampul. Untuk informasi selengkapnya, lihat Menghindari kembung dalam indeks di Panduan Pengguna Amazon Aurora.

Prasyarat

- Tabel harus memiliki PRIMARY KEY atau not-null UNIQUE kendala.
- Versi ekstensi harus sama untuk klien dan server.
- Pastikan bahwa instance RDS memiliki `FreeStorageSpace` lebih dari ukuran total tabel tanpa kembung. Sebagai contoh, pertimbangkan ukuran total tabel termasuk TOAST dan indeks sebagai 2TB, dan total kembung dalam tabel sebagai 1TB. Yang dibutuhkan `FreeStorageSpace` harus lebih dari nilai yang dikembalikan oleh perhitungan berikut:

  `2TB (Table size) - 1TB (Table bloat) = 1TB`

  Anda dapat menggunakan kueri berikut untuk memeriksa ukuran total tabel dan gunakan `pgstattuple` untuk mendapatkan kembung. Untuk informasi selengkapnya, lihat Mendiagnosis tabel dan indeks kembung di Panduan Pengguna Amazon Aurora

```
SELECT pg_size_pretty(pg_total_relation_size('table_name')) AS total_table_size;
```

Ruang ini direklamasi setelah selesainya kegiatan.

- Pastikan instans RDS memiliki kapasitas komputasi dan IO yang cukup untuk menangani operasi pengemasan ulang. Anda dapat mempertimbangkan untuk meningkatkan kelas instance untuk keseimbangan kinerja yang optimal.

Untuk menggunakan **`pg_repack`** ekstensi

1. Instal `pg_repack` ekstensi pada RDS Anda untuk PostgreSQL DB instance dengan menjalankan perintah berikut.

   ```
   CREATE EXTENSION pg_repack;
   ```

2. Jalankan perintah berikut untuk memberikan akses tulis ke tabel log sementara yang dibuat oleh`pg_repack`.

   ```
   ALTER DEFAULT PRIVILEGES IN SCHEMA repack GRANT INSERT ON TABLES TO PUBLIC;
   ALTER DEFAULT PRIVILEGES IN SCHEMA repack GRANT USAGE, SELECT ON SEQUENCES TO
    PUBLIC;
   ```

3. Connect ke database menggunakan utilitas `pg_repack` klien. Gunakan akun yang memiliki hak istimewa `rds_superuser`. Sebagai contoh, asumsikan bahwa peran `rds_test` memiliki hak istimewa `rds_superuser`. Sintaks berikut melakukan `pg_repack` untuk tabel lengkap termasuk semua indeks tabel dalam database. `postgres`

   ```
   pg_repack -h db-instance-name.111122223333.aws-region.rds.amazonaws.com -U rds_test
    -k postgres
   ```

   > **Note**
   >
   > Anda harus terhubung menggunakan opsi -k. Opsi -a tidak didukung.

   Respons dari `pg_repack` klien memberikan informasi pada tabel pada instance DB yang dikemas ulang.

```
INFO: repacking table "pgbench_tellers"
INFO: repacking table "pgbench_accounts"
INFO: repacking table "pgbench_branches"
```

4. Sintaks berikut menampilkan ulang tabel tunggal `orders` termasuk indeks dalam database. `postgres`

```
pg_repack -h db-instance-name.111122223333.aws-region.rds.amazonaws.com -U rds_test
 --table orders -k postgres
```

Sintaks berikut hanya menampilkan indeks untuk `orders` tabel dalam database. `postgres`

```
pg_repack -h db-instance-name.111122223333.aws-region.rds.amazonaws.com -U rds_test
 --table orders --only-indexes -k postgres
```

Memantau tabel baru selama pengemasan ulang

- Ukuran database ditingkatkan dengan ukuran total tabel dikurangi kembung, hingga swap-and-drop fase repack. Anda dapat memantau laju pertumbuhan ukuran database, menghitung kecepatan pengemasan ulang, dan memperkirakan secara kasar waktu yang diperlukan untuk menyelesaikan transfer data awal.

  Sebagai contoh, pertimbangkan ukuran total tabel sebagai 2TB, ukuran database sebagai 4TB, dan total kembung dalam tabel sebagai 1TB. Nilai ukuran total database yang dikembalikan oleh perhitungan pada akhir operasi repack adalah sebagai berikut:

  `2TB (Table size) + 4 TB (Database size) - 1TB (Table bloat) = 5TB`

  Anda dapat memperkirakan secara kasar kecepatan operasi pengemasan ulang dengan mengambil sampel laju pertumbuhan dalam byte antara dua titik waktu. Jika tingkat pertumbuhan 1GB per menit, dibutuhkan 1000 menit atau 16,6 jam kira-kira untuk menyelesaikan operasi pembuatan tabel awal. Selain pembuatan tabel awal, `pg_repack` juga perlu menerapkan perubahan yang masih harus dibayar. Waktu yang dibutuhkan tergantung pada tingkat penerapan perubahan yang sedang berlangsung ditambah perubahan yang masih harus dibayar.

> **Note**
>
> Anda dapat menggunakan `pgstattuple` ekstensi untuk menghitung kembung dalam tabel. Untuk informasi selengkapnya, lihat pgstattuple.

- Jumlah baris dalam tabel `pg_repack's` log, di bawah skema repack mewakili volume perubahan yang menunggu untuk diterapkan ke tabel baru setelah pemuatan awal.

  Anda dapat memeriksa tabel `pg_repack's` log `pg_stat_all_tables` untuk memantau perubahan yang diterapkan pada tabel baru. `pg_stat_all_tables.n_live_tup`menunjukkan jumlah catatan yang tertunda untuk diterapkan ke tabel baru. Untuk informasi selengkapnya, lihat pg_stat_all_tables.

```
postgres=>SELECT relname,n_live_tup FROM pg_stat_all_tables WHERE schemaname =
 'repack' AND relname ILIKE '%log%';

-[ RECORD 1 ]---------
relname    | log_16490
n_live_tup | 2000000
```

- Anda dapat menggunakan `pg_stat_statements` ekstensi untuk mengetahui waktu yang dibutuhkan oleh setiap langkah dalam operasi pengemasan ulang. Ini sangat membantu dalam persiapan untuk menerapkan operasi pengemasan ulang yang sama di lingkungan produksi. Anda dapat menyesuaikan `LIMIT` klausa untuk memperluas output lebih lanjut.

```
postgres=>SELECT
     SUBSTR(query, 1, 100) query,
     round((round(total_exec_time::numeric, 6) / 1000 / 60),4)
 total_exec_time_in_minutes
 FROM
     pg_stat_statements
 WHERE
     query ILIKE '%repack%'
 ORDER BY
     total_exec_time DESC LIMIT 5;

 query                                                                |
 total_exec_time_in_minutes
```

```
------------------------------------------------------------------------
+---------------------------
 CREATE UNIQUE INDEX index_16493 ON repack.table_16490 USING btree (a) |
        6.8627
 INSERT INTO repack.table_16490 SELECT a FROM ONLY public.t1           |
        6.4150
 SELECT repack.repack_apply($1, $2, $3, $4, $5, $6)                    |
        0.5395
 SELECT repack.repack_drop($1, $2)                                     |
        0.0004
 SELECT repack.repack_swap($1)                                         |
        0.0004
(5 rows)
```

Pengepakan ulang sepenuhnya merupakan out-of-place operasi sehingga tabel asli tidak terpengaruh dan kami tidak mengantisipasi tantangan tak terduga yang memerlukan pemulihan tabel asli. Jika repack gagal secara tak terduga, Anda harus memeriksa penyebab kesalahan dan menyelesaikannya.

Setelah masalah teratasi, jatuhkan dan buat ulang `pg_repack` ekstensi di database tempat tabel ada, dan coba lagi langkahnya. `pg_repack` Selain itu, ketersediaan sumber daya komputasi dan aksesibilitas tabel secara bersamaan memainkan peran penting dalam penyelesaian operasi pengemasan ulang secara tepat waktu.

Meningkatkan dan menggunakan ekstensi PLV8

PLV8 adalah ekstensi bahasa Javascript tepercaya untuk PostgreSQL. Anda dapat menggunakannya untuk prosedur tersimpan, pemicu, dan kode prosedural lainnya yang dapat dipanggil dari SQL. Ekstensi bahasa ini didukung oleh semua rilis PostgreSQL saat ini.

Jika Anda menggunakan PLV8 dan meningkatkan PostgreSQL ke versi PLV8 yang baru, Anda segera memanfaatkan ekstensi baru. Lakukan langkah-langkah berikut untuk menyinkronkan metadata katalog Anda dengan versi baru PLV8. Langkah-langkah ini opsional, tetapi kami sangat menyarankan Anda menyelesaikannya untuk menghindari peringatan ketidakcocokan metadata.

Proses peningkatan akan menghapus sementara semua fungsi PLV8 Anda yang ada. Oleh karena itu, kami menyarankan Anda membuat snapshot dari instans DB RDS for PostgreSQL Anda sebelum meningkatkan. Untuk informasi selengkapnya, lihat Membuat snapshot DB untuk instans DB Single-AZ.

Untuk menyinkronkan metadata katalog Anda dengan versi baru PLV8

1. Verifikasi bahwa Anda perlu memperbarui. Untuk melakukannya, jalankan perintah berikut saat terhubung dengan instans Anda.

   ```
   SELECT * FROM pg_available_extensions WHERE name IN ('plv8','plls','plcoffee');
   ```

   Jika hasil Anda berisi nilai untuk versi terinstal yang lebih rendah dari versi default, lanjutkan dengan prosedur ini untuk memperbarui ekstensi Anda. Misalnya, kumpulan hasil berikut menunjukkan bahwa Anda harus memperbarui.

   ```
   name    | default_version | installed_version |                    comment
   --------+-----------------+-------------------
   +-------------------------------------------------
   plls    | 2.1.0           | 1.5.3             | PL/LiveScript (v8) trusted
    procedural language
   plcoffee| 2.1.0           | 1.5.3             | PL/CoffeeScript (v8) trusted
    procedural language
   plv8    | 2.1.0           | 1.5.3             | PL/JavaScript (v8) trusted
    procedural language
   (3 rows)
   ```

2. Buat snapshot instans DB RDS for PostgreSQL Anda jika Anda belum melakukannya. Anda dapat melanjutkan dengan langkah-langkah berikut saat snapshot sedang dibuat.

3. Dapatkan hitungan jumlah fungsi PLV8 dalam instans DB Anda sehingga Anda dapat memvalidasi bahwa semuanya sudah siap setelah peningkatan. Misalnya, kueri SQL berikut mengembalikan jumlah fungsi yang ditulis dalam plv8, plcoffee, and plls.

```
SELECT proname, nspname, lanname
FROM pg_proc p, pg_language l, pg_namespace n
WHERE p.prolang = l.oid
AND n.oid = p.pronamespace
AND lanname IN ('plv8','plcoffee','plls');
```

4. Gunakan pg_dump untuk membuat file dump hanya untuk skema. Misalnya, buat file di mesin klien Anda di direktori `/tmp`.

```
./pg_dump -Fc --schema-only -U master postgres >/tmp/test.dmp
```

   Contoh ini menggunakan hal berikut:

   - `-Fc` – Format khusus
   - --schema-only – Buang perintah yang diperlukan untuk membuat skema (fungsi dalam kasus ini)
   - `-U` – Nama pengguna utama RDS
   - `database` – Nama untuk basis data di instans DB Anda

   Untuk informasi selengkapnya, lihat pg_dump dalam dokumentasi PostgreSQL.

5. Ekstrak pernyataan DDL "CREATE FUNCTION" yang ada di berkas dump. Contoh berikut menggunakan perintah `grep` untuk mengekstrak pernyataan DDL yang menciptakan fungsi dan menyimpannya ke file. Anda menggunakan ini dalam langkah-langkah berikutnya untuk membuat ulang fungsi.

```
./pg_restore -l /tmp/test.dmp | grep FUNCTION > /tmp/function_list/
```

   Untuk informasi selengkapnya pada pg_restore, lihat pg_restore dalam dokumentasi PostgreSQL.

6. Hapus sementara fungsi dan ekstensi. Contoh berikut menghapus sementara objek berbasis PLV8 apa pun. Opsi kaskade memastikan bahwa ketergantungan apa pun dapat dihapus sementara.

```
DROP EXTENSION plv8 CASCADE;
```

Jika instans PostgreSQL Anda berisi objek berdasarkan plcoffee atau plls, ulangi langkah ini untuk ekstensi tersebut.

7. Buat ekstensi. Contoh berikut untuk membuat ekstensi plv8, plcoffee, dan plls.

```
CREATE EXTENSION plv8;
CREATE EXTENSION plcoffee;
CREATE EXTENSION plls;
```

8. Buat fungsi menggunakan file dump dan file "driver".

   Contoh berikut membuat ulang fungsi yang Anda ekstrak sebelumnya.

```
./pg_restore -U master -d postgres -Fc -L /tmp/function_list /tmp/test.dmp
```

9. Verifikasi bahwa semua fungsi Anda telah dibuat ulang dengan menggunakan kueri berikut.

```
SELECT * FROM pg_available_extensions WHERE name IN ('plv8','plls','plcoffee');
```

   PLV8 versi 2 menambahkan baris tambahan berikut ke set hasil Anda:

```
    proname    |  nspname   | lanname
---------------+------------+----------
 plv8_version  | pg_catalog | plv8
```

Menggunakan PL/Rust untuk menulis fungsi PostgreSQL dalam bahasa Rust

PL/Rust adalah ekstensi bahasa Rust tepercaya untuk PostgreSQL. Anda dapat menggunakannya untuk prosedur tersimpan, fungsi, dan kode prosedural lainnya yang dapat dipanggil dari SQL. Ekstensi bahasa PL/Rust tersedia dalam versi berikut:

- RDS untuk PostgreSQL 16.1 dan versi 16 yang lebih tinggi
- RDS for PostgreSQL 15.2-R2 dan versi 15 yang lebih tinggi
- RDS for PostgreSQL 14.9 dan versi 14 yang lebih tinggi

- RDS for PostgreSQL 13.12 dan versi 13 yang lebih tinggi

Untuk informasi lebih lanjut, lihat PL/Rust on. GitHub

Topik

- Menyiapkan PL/Rust
- Membuat fungsi dengan PL/Rust
- Menggunakan crate dengan PL/Rust
- Batasan PL/Rust

Menyiapkan PL/Rust

Untuk menginstal ekstensi plrust pada instans DB Anda, tambahkan plrust ke parameter `shared_preload_libraries` dalam grup parameter DB yang terkait dengan instans DB Anda. Dengan ekstensi plrust yang terinstall, Anda dapat membuat fungsi.

Untuk mengubah parameter `shared_preload_libraries`, instans DB Anda harus berkaitan dengan grup parameter khusus. Untuk informasi tentang cara membuat grup parameter DB kusto, lihat Bekerja dengan grup parameter.

Anda dapat menginstal ekstensi plrust menggunakan AWS Management Console atau. AWS CLI

Langkah-langkah berikut mengasumsikan bahwa instans Anda dikaitkan dengan grup parameter DB khusus.

Konsol

Install ekstensi plrust di parameter **`shared_preload_libraries`**

Lakukan langkah-langkah berikut menggunakan akun yang merupakan anggota grup `rds_superuser` (peran).

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data.
3. Pilih nama instans DB Anda untuk menampilkan detailnya.
4. Buka tab Konfigurasi untuk instans DB Anda dan temukan tautan grup parameter instans DB.

5. Pilih tautan untuk membuka parameter khusus yang terkait dengan instans DB.
6. Di bidang pencarian Parameter, ketik `shared_pre` untuk menemukan parameter **`shared_preload_libraries`**.
7. Pilih Edit parameter untuk mengakses nilai properti.
8. Tambahkan plrust ke daftar di bidang Nilai. Gunakan koma untuk memisahkan item dalam daftar nilai.
9. Boot ulang instans DB sehingga perubahan Anda pada parameter`shared_preload_libraries` akan berlaku. Boot ulang awal mungkin memerlukan waktu tambahan untuk menyelesaikannya.
10. Ketika instans tersedia, verifikasi bahwa plrust telah diinisialisasi. Gunakan `psql` untuk terhubung ke instans DB, kemudian jalankan perintah berikut.

```
SHOW shared_preload_libraries;
```

Outputnya semestinya mirip dengan yang berikut:

```
shared_preload_libraries
-------------------------
rdsutils,plrust
(1 row)
```

AWS CLI

Install ekstensi plrust di parameter shared_preload_libraries

Lakukan langkah-langkah berikut menggunakan akun yang merupakan anggota grup `rds_superuser` (peran).

1. Gunakan modify-db-parameter-group AWS CLI perintah untuk menambahkan plrust ke `shared_preload_libraries` parameter.

```
aws rds modify-db-parameter-group \
   --db-parameter-group-name custom-param-group-name \
   --parameters
 "ParameterName=shared_preload_libraries,ParameterValue=plrust,ApplyMethod=pending-
reboot" \
   --region aws-region
```

2. Gunakan reboot-db-instance AWS CLI perintah untuk me-reboot instance DB dan menginisialisasi pustaka plrust. Boot ulang awal mungkin memerlukan waktu tambahan untuk menyelesaikannya.

```
aws rds reboot-db-instance \
    --db-instance-identifier your-instance \
    --region aws-region
```

3. Ketika instans tersedia, Anda dapat memverifikasi bahwa plrust telah diinisialisasi. Gunakan `psql` untuk terhubung ke instans DB, kemudian jalankan perintah berikut.

```
SHOW shared_preload_libraries;
```

Outputnya semestinya mirip dengan yang berikut:

```
shared_preload_libraries
--------------------------
rdsutils,plrust
(1 row)
```

Membuat fungsi dengan PL/Rust

PL/Rust akan mengkompilasi fungsi sebagai pustaka dinamis, memuatnya, dan menjalankannya.

Fungsi Rust berikut menyaring kelipatan dari array.

```
postgres=> CREATE LANGUAGE plrust;
CREATE EXTENSION
```

```
CREATE OR REPLACE FUNCTION filter_multiples(a BIGINT[], multiple BIGINT) RETURNS
 BIGINT[]
    IMMUTABLE STRICT
    LANGUAGE PLRUST AS
$$
    Ok(Some(a.into_iter().filter(|x| x.unwrap() % multiple != 0).collect()))
$$;

WITH gen_values AS (
SELECT ARRAY(SELECT * FROM generate_series(1,100)) as arr)
SELECT filter_multiples(arr, 3)
```

```
from gen_values;
```

Menggunakan crate dengan PL/Rust

Dimulai dengan Amazon RDS untuk PostgreSQL versi 15.4, 14.9, dan 13.12, PL/Rust mendukung peti berikut:

- `aes`
- `ctr`
- `rand`

Dimulai dengan RDS untuk PostgreSQL versi 15.5-R2, 14.10-R2, dan 13.13-R2, PL/Rust mendukung dua peti tambahan:

- `croaring-rs`
- `num-bigint`

Hanya fitur default yang didukung untuk crate ini. Versi RDS for PostgreSQL baru mungkin berisi versi crate terbaru, dan crate versi lama yang mungkin tidak lagi didukung lagi.

Ikuti praktik terbaik untuk melakukan peningkatan versi utama untuk menguji apakah fungsi PL/Rust Anda kompatibel dengan versi utama yang baru. Untuk informasi selengkapnya, lihat blog Best practices for upgrading Amazon RDS to major and minor versions of PostgreSQL serta Upgrading the PostgreSQL DB engine for Amazon RDS di Panduan Pengguna Amazon RDS.

Contoh penggunaan dependensi saat membuat fungsi PL/Rust tersedia di Gunakan dependensi.

Batasan PL/Rust

Secara default, pengguna basis data tidak dapat menggunakan PL/Rust. Untuk menyediakan akses ke PL/Rust, hubungkan sebagai pengguna dengan hak istimewa rds_superuser, lalu jalankan perintah berikut:

```
postgres=> GRANT USAGE ON LANGUAGE PLRUST TO user;
```

Mengelola data spasial dengan ekstensi PostGIS

PostGIS adalah ekstensi dari PostgreSQL untuk menyimpan dan mengelola informasi spasial. Untuk mempelajari PostGIS selengkapnya, lihat PostGIS.net.

Dimulai dengan versi 10.5, PostgreSQL mendukung pustaka libprotobuf 1.3.0 yang digunakan oleh PostGIS untuk menangani data petak vektor kotak peta.

Menyiapkan ekstensi PostGIS membutuhkan hak istimewa `rds_superuser`. Sebaiknya Anda membuat pengguna (peran) untuk mengelola ekstensi PostGIS dan data spasial. Ekstensi PostGIS dan komponen terkaitnya akan menambahkan ribuan fungsi ke PostgreSQL. Pertimbangkan untuk membuat ekstensi PostGIS dalam skema sendiri jika berterima untuk kasus penggunaan Anda. Contoh berikut menunjukkan cara menginstal ekstensi dalam basis data sendiri, tetapi ini tidak diperlukan.

Topik

Langkah 1: Membuat pengguna (peran) untuk mengelola ekstensi PostGIS

Pertama, hubungkan ke instans DB RDS for PostgreSQL sebagai pengguna yang memiliki hak istimewa `rds_superuser`. Jika Anda menyimpan nama default saat menyiapkan instans, Anda akan terhubung sebagai `postgres`.

```
psql --host=111122223333.aws-region.rds.amazonaws.com --port=5432 --username=postgres
 --password
```

Buat peran terpisah (pengguna) untuk mengelola ekstensi PostGIS.

```
postgres=>  CREATE ROLE gis_admin LOGIN PASSWORD 'change_me';
```

```
CREATE ROLE
```

Beri hak istimewa `rds_superuser` ke peran ini agar dapat menginstal ekstensi.

```
postgres=> GRANT rds_superuser TO gis_admin;
GRANT
```

Buat basis data agar digunakan untuk artefak PostGIS. Langkah ini bersifat opsional. Bisa juga dangan membuat skema di basis data pengguna untuk ekstensi PostGIS, tetapi langkah ini juga tidak diperlukan.

```
postgres=> CREATE DATABASE lab_gis;
CREATE DATABASE
```

Beri semua hak istimewa `gis_admin` pada basis data `lab_gis`.

```
postgres=> GRANT ALL PRIVILEGES ON DATABASE lab_gis TO gis_admin;
GRANT
```

Keluar dari sesi lalu hubungkan kembali ke instans DB RDS for PostgreSQL sebagai `gis_admin`.

```
postgres=> psql --host=111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=gis_admin --password --dbname=lab_gis
Password for user gis_admin:...
lab_gis=>
```

Lanjutkan menyiapkan ekstensi seperti yang dijelaskan pada langkah selanjutnya.

Langkah 2: Memuat ekstensi PostGIS

Ekstensi PostGIS mencakup beberapa ekstensi terkait yang bekerja sama untuk menyediakan fungsionalitas geospasial. Anda mungkin tidak memerlukan semua ekstensi yang dibuat pada langkah ini, bergantung pada kasus penggunaan Anda.

Gunakan pernyataan `CREATE EXTENSION` untuk memuat ekstensi PostGIS.

```
CREATE EXTENSION postgis;
CREATE EXTENSION
CREATE EXTENSION postgis_raster;
CREATE EXTENSION
CREATE EXTENSION fuzzystrmatch;
```

```
CREATE EXTENSION
CREATE EXTENSION postgis_tiger_geocoder;
CREATE EXTENSION
CREATE EXTENSION postgis_topology;
CREATE EXTENSION
CREATE EXTENSION address_standardizer_data_us;
CREATE EXTENSION
```

Anda dapat memverifikasi hasil dengan menjalankan kueri SQL yang ditunjukkan dalam contoh berikut yang mencantumkan ekstensi beserta pemiliknya.

```
SELECT n.nspname AS "Name",
  pg_catalog.pg_get_userbyid(n.nspowner) AS "Owner"
  FROM pg_catalog.pg_namespace n
  WHERE n.nspname !~ '^pg_' AND n.nspname <> 'information_schema'
  ORDER BY 1;
List of schemas
     Name     |   Owner
--------------+-----------
 public       | postgres
 tiger        | rdsadmin
 tiger_data   | rdsadmin
 topology     | rdsadmin
(4 rows)
```

Langkah 3: Mentransfer kepemilikan ekstensi

Gunakan pernyataan ALTER SCHEMA untuk mentransfer kepemilikan skema ke peran `gis_admin`.

```
ALTER SCHEMA tiger OWNER TO gis_admin;
ALTER SCHEMA
ALTER SCHEMA tiger_data OWNER TO gis_admin;
ALTER SCHEMA
ALTER SCHEMA topology OWNER TO gis_admin;
ALTER SCHEMA
```

Anda dapat mengonfirmasi perubahan kepemilikan dengan menjalankan kueri SQL berikut. Bisa juga dengan menggunakan metacommand \dn dari baris perintah psql.

```
SELECT n.nspname AS "Name",
  pg_catalog.pg_get_userbyid(n.nspowner) AS "Owner"
```

```
  FROM pg_catalog.pg_namespace n
  WHERE n.nspname !~ '^pg_' AND n.nspname <> 'information_schema'
  ORDER BY 1;

      List of schemas
    Name     |     Owner
-------------+--------------
 public      | postgres
 tiger       | gis_admin
 tiger_data  | gis_admin
 topology    | gis_admin
(4 rows)
```

Langkah 4: Mentransfer kepemilikan objek PostGIS

Gunakan fungsi berikut untuk mentransfer kepemilikan objek PostGIS ke peran `gis_admin`. Jalankan pernyataan berikut dari perintah psql untuk membuat fungsinya.

```
CREATE FUNCTION exec(text) returns text language plpgsql volatile AS $f$ BEGIN EXECUTE
 $1; RETURN $1; END; $f$;
CREATE FUNCTION
```

Selanjutnya, jalankan kueri berikut untuk menjalankan fungsi `exec` yang nantinya akan menjalankan pernyataan dan mengubah izin.

```
SELECT exec('ALTER TABLE ' || quote_ident(s.nspname) || '.' || quote_ident(s.relname)
 || ' OWNER TO gis_admin;')
  FROM (
    SELECT nspname, relname
    FROM pg_class c JOIN pg_namespace n ON (c.relnamespace = n.oid)
    WHERE nspname in ('tiger','topology') AND
    relkind IN ('r','S','v') ORDER BY relkind = 'S')
s;
```

Langkah 5: Menguji ekstensi

Untuk menghindari kebutuhan menentukan nama skema, tambahkan skema `tiger` ke jalur pencarian Anda menggunakan perintah berikut.

```
SET search_path=public,tiger;
SET
```

Uji skema `tiger` menggunakan pernyataan SELECT berikut.

```
SELECT address, streetname, streettypeabbrev, zip
 FROM normalize_address('1 Devonshire Place, Boston, MA 02109') AS na;
address | streetname | streettypeabbrev |  zip
---------+------------+------------------+-------
       1 | Devonshire | Pl               | 02109
(1 row)
```

Untuk mempelajari ekstensi ini selengkapnya, lihat Tiger Geocode dalam dokumentasi PostGIS.

Uji akses ke skema `topology` menggunakan pernyataan `SELECT` berikut. Tindakan ini akan memanggil fungsi `createtopology` untuk mendaftarkan objek topologi baru (my_new_topo) dengan pengidentifikasi referensi spasial yang ditentukan (26986) dan toleransi default (0,5). Untuk mempelajari selengkapnya, lihat CreateTopology dalam dokumentasi PostGIS.

```
SELECT topology.createtopology('my_new_topo',26986,0.5);
 createtopology
----------------
              1
(1 row)
```

Langkah 6: Meningkatkan ekstensi PostGIS

Setiap rilis baru PostgreSQL mendukung satu atau beberapa versi ekstensi PostGIS yang kompatibel dengan rilis tersebut. Meningkatkan mesin PostgreSQL ke versi baru tidak otomatis akan meningkatkan ekstensi PostGIS. Sebelum meningkatkan mesin PostgreSQL, Anda biasanya harus meningkatkan PostGIS ke versi terbaru yang tersedia untuk versi PostgreSQL saat ini. Untuk mengetahui detailnya, lihat Versi ekstensi PostGIS.

Setelah meningkatkan mesin PostgreSQL, Anda kemudian harus meningkatkan ekstensi PostGIS lagi ke versi yang didukung untuk versi mesin PostgreSQL yang baru ditingkatkan. Untuk informasi cara meningkatkan mesin PostgreSQL selengkapnya, lihat Cara melakukan peningkatan versi mayor.

Anda dapat memeriksa pembaruan versi ekstensi PostGIS yang tersedia di instans DB RDS for PostgreSQL kapan saja. Untuk melakukannya, jalankan perintah berikut. Fungsi ini tersedia dengan PostGIS 2.5.0 dan versi yang lebih baru.

```
SELECT postGIS_extensions_upgrade();
```

Jika aplikasi Anda tidak mendukung versi PostGIS terbaru, Anda dapat menginstal versi PostGIS lama yang tersedia di versi utama Anda sebagai berikut.

```
CREATE EXTENSION postgis VERSION "2.5.5";
```

Jika ingin meningkatkan ke versi PostGIS tertentu dari versi lama, Anda juga dapat menggunakan perintah berikut.

```
ALTER EXTENSION postgis UPDATE TO "2.5.5";
```

Bergantung pada versi yang Anda tingkatkan, Anda mungkin perlu menggunakan fungsi ini lagi. Hasil dari menjalankan fungsi pertama akan menentukan perlu atau tidaknya fungsi peningkatan tambahan. Misalnya, ini adalah kasus peningkatan dari PostGIS 2 ke PostGIS 3. Untuk informasi selengkapnya, lihat Meningkatkan PostGIS 2 ke PostGIS 3.

Jika meningkatkan ekstensi ini untuk mempersiapkan peningkatan versi utama dari mesin PostgreSQL, Anda dapat melanjutkan tugas awal lainnya. Untuk informasi selengkapnya, lihat Cara melakukan peningkatan versi mayor.

Versi ekstensi PostGIS

Sebaiknya Anda menginstal versi semua ekstensi seperti PostGIS seperti yang tercantum di Versi ekstensi untuk Amazon RDS for PostgreSQL dalam Catatan Rilis Amazon RDS for PostgreSQL. Untuk mendapatkan daftar versi yang tersedia dalam rilis Anda, gunakan perintah berikut.

```
SELECT * FROM pg_available_extension_versions WHERE name='postgis';
```

Anda dapat menemukan informasi versi di bagian berikut dalam Catatan Rilis Amazon RDS for PostgreSQL:

- Ekstensi PostgreSQL versi 15 didukung di Amazon RDS
- Ekstensi PostgreSQL versi 14 didukung di Amazon RDS
- Ekstensi PostgreSQL versi 13 didukung di Amazon RDS
- Ekstensi PostgreSQL versi 12 didukung di Amazon RDS
- Ekstensi PostgreSQL versi 11 didukung di Amazon RDS
- Ekstensi PostgreSQL versi 10 didukung di Amazon RDS

- Ekstensi PostgreSQL versi 9.6.x yang didukung di Amazon RDS

Meningkatkan PostGIS 2 ke PostGIS 3

Dimulai dengan versi 3.0, fungsi raster PostGIS sekarang merupakan ekstensi terpisah, `postgis_raster`. Ekstensi ini memiliki jalur instalasi dan peningkatan sendiri. Ekstensi ini menghapus lusinan fungsi, tipe data, dan artefak lain yang diperlukan untuk pemrosesan gambar raster dari ekstensi `postgis` inti. Itu berarti bahwa jika kasus penggunaan Anda tidak memerlukan pemrosesan raster, Anda tidak perlu menginstal ekstensi `postgis_raster`.

Dalam contoh peningkatan berikut, perintah peningkatan pertama mengekstrak fungsionalitas raster ke dalam ekstensi `postgis_raster`. Selanjutnya, perintah peningkatan kedua akan diperlukan untuk meningkatkan `postgres_raster` ke versi baru.

Untuk meningkatkan dari PostGIS 2 ke PostGIS 3

1. Identifikasi versi default PostGIS yang tersedia untuk versi PostgreSQL di instans DB RDS for PostgreSQL. Untuk melakukannya, masukkan kueri berikut.

```
SELECT * FROM pg_available_extensions
    WHERE default_version > installed_version;
  name   | default_version | installed_version |                          comment
---------+-----------------+-------------------
+------------------------------------------------------------
 postgis | 3.1.4           | 2.3.7             | PostGIS geometry and geography
 spatial types and functions
(1 row)
```

2. Identifikasi versi PostGIS yang diinstal di setiap basis data di instans DB RDS for PostgreSQL. Dengan kata lain, kueri setiap basis data pengguna sebagai berikut.

```
SELECT
    e.extname AS "Name",
    e.extversion AS "Version",
    n.nspname AS "Schema",
    c.description AS "Description"
FROM
    pg_catalog.pg_extension e
    LEFT JOIN pg_catalog.pg_namespace n ON n.oid = e.extnamespace
    LEFT JOIN pg_catalog.pg_description c ON c.objoid = e.oid
        AND c.classoid = 'pg_catalog.pg_extension'::pg_catalog.regclass
```

```
WHERE
    e.extname LIKE '%postgis%'
ORDER BY
    1;
  Name   | Version | Schema |                            Description
---------+---------+--------
+------------------------------------------------------------------------
 postgis | 2.3.7   | public | PostGIS geometry, geography, and raster spatial types
 and functions
(1 row)
```

Ketidakcocokan antara versi default (PostGIS 3.1.4) dan versi yang diinstal (PostGIS 2.3.7) ini mengindikasikan bahwa Anda perlu meningkatkan ekstensi PostGIS.

```
ALTER EXTENSION postgis UPDATE;
ALTER EXTENSION
WARNING: unpackaging raster
WARNING: PostGIS Raster functionality has been unpackaged
```

3. Jalankan kueri berikut untuk memverifikasi bahwa fungsi raster sekarang telah berada dalam paketnya sendiri.

```
SELECT
    probin,
    count(*)
FROM
    pg_proc
WHERE
    probin LIKE '%postgis%'
GROUP BY
    probin;
          probin          | count
--------------------------+-------
 $libdir/rtpostgis-2.3    | 107
 $libdir/postgis-3        | 487
(2 rows)
```

Hasilnya menunjukkan bahwa masih ada perbedaan antar-versi. Fungsi PostGIS adalah versi 3 (postgis-3), sedangkan fungsi raster (rtpostgis) adalah versi 2 (rtpostgis-2.3). Untuk menyelesaikan peningkatan, Anda dapat menjalankan perintah peningkatan lagi sebagai berikut.

```
postgres=> SELECT postgis_extensions_upgrade();
```

Anda dapat dengan aman mengabaikan kesalahan peringatan. Jalankan lagi kueri berikut untuk memverifikasi bahwa peningkatan telah selesai. Peningkatan selesai ketika PostGIS dan semua ekstensi terkait tidak ditandai sebagai harus ditingkatkan.

```
SELECT postgis_full_version();
```

4. Gunakan kueri berikut untuk melihat proses peningkatan yang telah selesai dan ekstensi yang dikemas secara terpisah, lalu verifikasi bahwa versinya telah sesuai.

```
SELECT
    e.extname AS "Name",
    e.extversion AS "Version",
    n.nspname AS "Schema",
    c.description AS "Description"
FROM
    pg_catalog.pg_extension e
    LEFT JOIN pg_catalog.pg_namespace n ON n.oid = e.extnamespace
    LEFT JOIN pg_catalog.pg_description c ON c.objoid = e.oid
        AND c.classoid = 'pg_catalog.pg_extension'::pg_catalog.regclass
WHERE
    e.extname LIKE '%postgis%'
ORDER BY
    1;
      Name      | Version | Schema |                            Description
----------------+---------+--------
+------------------------------------------------------------------------
 postgis        | 3.1.5   | public | PostGIS geometry, geography, and raster
 spatial types and functions
 postgis_raster | 3.1.5   | public | PostGIS raster types and functions
(2 rows)
```

Output menunjukkan bahwa ekstensi PostGIS 2 telah ditingkatkan ke PostGIS 3, serta ekstensi `postgis` dan ekstensi `postgis_raster` yang sekarang terpisah adalah versi 3.1.5.

Setelah peningkatan ini selesai, Anda dapat menghapus ekstensi seperti di bawah ini jika Anda tidak berencana menggunakan fungsionalitas raster.

```
DROP EXTENSION postgis_raster;
```

Bekerja dengan pembungkus data asing yang didukung untuk Amazon RDS for PostgreSQL

Pembungkus data asing (FDW) adalah jenis ekstensi khusus yang menyediakan akses ke data eksternal. Misalnya, ekstensi `oracle_fdw` memungkinkan klaster DB RDS for PostgreSQL bekerja dengan basis data Oracle. Sebagai contoh lain, dengan menggunakan ekstensi `postgres_fdw` native PostgreSQL, Anda dapat mengakses data yang disimpan dalam instans DB PostgreSQL di luar instans DB RDS for PostgreSQL Anda.

Selanjutnya, Anda dapat menemukan informasi beberapa pembungkus data asing PostgreSQL yang didukung.

Topik

Menggunakan ekstensi log_fdw untuk mengakses log DB menggunakan SQL

instans DB RDS for PosgreSQL mendukung ekstensi `log_fdw` yang dapat Anda gunakan untuk mengakses log mesin basis data menggunakan antarmuka SQL. Ekstensi `log_fdw` ini menyediakan dua fungsi yang memudahkan pembuatan tabel asing untuk log basis data:

- `list_postgres_log_files` – Mencantumkan file di direktori log basis data dan ukuran file dalam bita.
- `create_foreign_table_for_log_file(table_name text, server_name text, log_file_name text)` – Membangun tabel asing untuk file yang ditentukan dalam basis data saat ini.

Semua fungsi yang dibuat oleh `log_fdw` dimiliki oleh `rds_superuser`. Anggota peran `rds_superuser` dapat memberikan akses ke fungsi ini kepada pengguna basis data lain.

Secara default, file log dihasilkan oleh Amazon RDS dalam format `stderr` (kesalahan standar), seperti yang ditentukan dalam parameter `log_destination`. Hanya ada dua opsi untuk parameter ini, yaitu `stderr` dan `csvlog` (nilai yang dipisahkan koma, CSV). Jika Anda menambahkan opsi `csvlog` ke parameter, Amazon RDS akan menghasilkan kedua log `stderr` dan `csvlog`. Penambahan opsi ini dapat memengaruhi kapasitas penyimpanan di klaster DB sehingga Anda perlu mengetahui parameter lain yang memengaruhi penanganan log. Untuk informasi selengkapnya, lihat Mengatur tujuan log (`stderr`, `csvlog`).

Salah satu manfaat dari menghasilkan log `csvlog` adalah bahwa ekstensi `log_fdw` memungkinkan Anda membangun tabel asing dengan data yang terbagi rapi menjadi beberapa kolom. Untuk melakukan ini, instans Anda harus dikaitkan dengan grup parameter DB khusus sehingga Anda dapat mengubah pengaturan `log_destination`. Untuk informasi cara melakukan ini selengkapnya, lihat Bekerja dengan parameter pada instans DB RDS for PostgreSQL.

Contoh berikut mengasumsikan bahwa parameter `log_destination` mencakup `cvslog`.

Untuk menggunakan ekstensi log_fdw

1. Instal ekstensi `log_fdw`.

```
postgres=> CREATE EXTENSION log_fdw;
CREATE EXTENSION
```

2. Buat server log sebagai pembungkus data asing.

```
postgres=> CREATE SERVER log_server FOREIGN DATA WRAPPER log_fdw;
CREATE SERVER
```

3. Pilih semua dari daftar file log.

```
postgres=> SELECT * FROM list_postgres_log_files() ORDER BY 1;
```

   Respons sampel adalah sebagai berikut.

```
          file_name           | file_size_bytes
------------------------------+-----------------
 postgresql.log.2023-08-09-22.csv |            1111
 postgresql.log.2023-08-09-23.csv |            1172
 postgresql.log.2023-08-10-00.csv |            1744
 postgresql.log.2023-08-10-01.csv |            1102
```

```
(4 rows)
```

4. Membuat tabel dengan satu kolom 'log_entry' untuk file yang dipilih.

```
postgres=> SELECT create_foreign_table_for_log_file('my_postgres_error_log',
     'log_server', 'postgresql.log.2023-08-09-22.csv');
```

Respons tidak memberikan detail selain memberitahukan bahwa sekarang tabel telah ada.

```
----------------------------------
(1 row)
```

5. Pilih contoh file log. Kode berikut mengambil waktu log dan deskripsi pesan kesalahan.

```
postgres=> SELECT log_time, message FROM my_postgres_error_log ORDER BY 1;
```

Respons sampel adalah sebagai berikut.

```
             log_time             |                                 message
----------------------------------
+--------------------------------------------------------------------------------
Tue Aug 09 15:45:18.172 2023 PDT | ending log output to stderr
Tue Aug 09 15:45:18.175 2023 PDT | database system was interrupted; last known up
 at 2023-08-09 22:43:34 UTC
Tue Aug 09 15:45:18.223 2023 PDT | checkpoint record is at 0/90002E0
Tue Aug 09 15:45:18.223 2023 PDT | redo record is at 0/90002A8; shutdown FALSE
Tue Aug 09 15:45:18.223 2023 PDT | next transaction ID: 0/1879; next OID: 24578
Tue Aug 09 15:45:18.223 2023 PDT | next MultiXactId: 1; next MultiXactOffset: 0
Tue Aug 09 15:45:18.223 2023 PDT | oldest unfrozen transaction ID: 1822, in
 database 1
(7 rows)
```

Menggunakan ekstensi postgres_fdw untuk mengakses data eksternal

Anda dapat mengakses data dalam tabel di server basis data jarak jauh menggunakan ekstensi postgres_fdw. Jika Anda mengatur koneksi jarak jauh dari instans DB PostgreSQL, akses juga akan tersedia untuk replika baca Anda.

Untuk menggunakan postgres_fdw agar dapat mengakses server basis data jarak jauh

1. Instal ekstensi postgres_fdw.

```
CREATE EXTENSION postgres_fdw;
```

2. Buat server data asing menggunakan CREATE SERVER.

```
CREATE SERVER foreign_server
FOREIGN DATA WRAPPER postgres_fdw
OPTIONS (host 'xxx.xx.xxx.xx', port '5432', dbname 'foreign_db');
```

3. Buat pemetaan pengguna untuk mengidentifikasi peran yang akan digunakan di server jarak jauh.

```
CREATE USER MAPPING FOR local_user
SERVER foreign_server
OPTIONS (user 'foreign_user', password 'password');
```

4. Buat tabel yang memetakan ke tabel pada server jarak jauh.

```
CREATE FOREIGN TABLE foreign_table (
        id integer NOT NULL,
        data text)
SERVER foreign_server
OPTIONS (schema_name 'some_schema', table_name 'some_table');
```

Bekerja dengan basis data MySQL menggunakan ekstensi mysql_fdw

Untuk mengakses basis data yang kompatibel dengan MySQL dari instans DB RDS for PostgreSQL, Anda dapat menginstal dan menggunakan ekstensi `mysql_fdw`. Pembungkus data asing ini memungkinkan Anda bekerja dengan RDS fo MySQL, Aurora MySQL, MariaDB, dan basis data MySQL lainnya yang kompatibel. Koneksi dari instans DB RDS for PostgreSQL ke basis data MySQL dienkripsi secara optimal, bergantung pada konfigurasi klien dan server. Namun, Anda dapat menerapkan enkripsi jika ingin. Untuk informasi selengkapnya, lihat Menggunakan enkripsi bergerak dengan ekstensi.

Ekstensi `mysql_fdw` didukung di Amazon RDS for PostgreSQL versi 14.2, 13.6,, dan rilis yang lebih baru. Ekstensi ini mendukung tugas memilih, menyisipkan, memperbarui, dan menghapus dari DB RDS for PostgreSQL ke tabel di instans basis data yang kompatibel dengan MySQL.

Topik

Menyiapkan basis data RDS for PostgreSQL untuk menggunaan ekstensi mysql_fdw.

Menyiapkan ekstensi `mysql_fdw` di instans DB RDS for PostgreSQL melibatkan pemuatan ekstensi di instans DB lalu membuat koneksi yang mengarah ke instans DB MySQL. Untuk tugas tersebut, Anda harus memiliki detail instans DB MySQL berikut:

- Nama host atau titik akhir. Untuk instans DB RDS for MySQL, Anda dapat menemukan titik akhir menggunakan Konsol. Pilih tab Konektivitas & keamanan dan lihat di bagian “Titik akhir dan port”.
- Nomor port. Nomor port default untuk MySQL adalah 3306.
- Nama basis data. Pengidentifikasi DB.

Anda juga perlu memberikan akses di grup keamanan atau daftar kontrol akses (ACL) untuk port MySQL, 3306. Baik instans DB RDS for PostgreSQL dan instans DB RDS for MySQL memerlukan akses ke port 3306. Jika akses tidak dikonfigurasi dengan benar, Anda akan melihat pesan kesalahan yang mirip dengan yang ada di bawah ini saat mencoba menghubungkan ke tabel yang kompatibel dengan MySQL:

```
ERROR: failed to connect to MySQL: Can't connect to MySQL server on 'hostname.aws-
region.rds.amazonaws.com:3306' (110)
```

Dalam prosedur berikut, Anda (sebagai akun `rds_superuser`) akan membuat server asing. Selanjutnya Anda akan memberikan akses ke server asing untuk pengguna tertentu. Pengguna ini kemudian akan membuat pemetaan mereka sendiri ke akun pengguna MySQL yang sesuai untuk bekerja dengan instans DB MySQL.

Untuk menggunakan mysql_fdw agar dapat mengakses server basis data MySQL

1. Hubungkan ke instans DB PostgreSQL Anda menggunakan akun yang memiliki peran `rds_superuser`. Jika Anda menerima default ketika membuat instans DB RDS for PostgreSQL, nama pengguna adalah `postgres`, dan Anda dapat terhubung menggunakan alat baris perintah `psql` sebagai berikut:

```
psql --host=your-DB-instance.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password
```

2. Instal ekstensi `mysql_fdw` sebagai berikut:

```
postgres=> CREATE EXTENSION mysql_fdw;
CREATE EXTENSION
```

Setelah ekstensi diinstal di instans DB RDS for PostgreSQL, Anda dapat menyiapkan server asing yang menyediakan koneksi ke basis data MySQL.

Untuk membuat server asing

Lakukan tugas-tugas ini di instans DB RDS for PostgreSQL. Langkah-langkah ini mengasumsikan bahwa Anda terhubung sebagai pengguna dengan hak istimewa `rds_superuser`, seperti `postgres`.

1. Membuat server asing di instans DB RDS for PostgreSQL:

```
postgres=> CREATE SERVER mysql-db FOREIGN DATA WRAPPER mysql_fdw OPTIONS (host 'db-
name.111122223333.aws-region.rds.amazonaws.com', port '3306');
CREATE SERVER
```

2. Berikan akses pengguna yang sesuai ke server asing. Pengguna yang diberi akses harus pengguna non-administrator, yaitu pengguna tanpa peran `rds_superuser`.

```
postgres=> GRANT USAGE ON FOREIGN SERVER mysql-db to user1;
GRANT
```

Pengguna PostgreSQL membuat dan mengelola koneksi mereka ke basis data MySQL melalui server asing.

Contoh: Bekerja dengan basis data RDS for MySQL dari RDS for PostgreSQL

Misalkan Anda memiliki tabel sederhana di instans DB RDS for PostgreSQL. Pengguna RDS for PostgreSQL Anda ingin mengueri item (`SELECT`), `INSERT`, `UPDATE`, dan `DELETE` pada tabel tersebut. Asumsikan bahwa ekstensi `mysql_fdw` dibuat di RDS telah Anda untuk instans DB PostgreSQL, seperti yang dijelaskan dalam prosedur sebelumnya. Setelah terhubung ke instans DB

RDS for PostgreSQL sebagai pengguna yang memiliki hak istimewa `rds_superuser`, Anda dapat melanjutkan langkah-langkah berikut.

1. Di instans DB RDS for PostgreSQL, buat server asing:

```
test=> CREATE SERVER mysqldb FOREIGN DATA WRAPPER mysql_fdw OPTIONS (host 'your-DB.aws-region.rds.amazonaws.com', port '3306');
CREATE SERVER
```

2. Berikan penggunaan kepada pengguna yang tidak memiliki izin `rds_superuser`, misalnya `user1`:

```
test=> GRANT USAGE ON FOREIGN SERVER mysqldb TO user1;
GRANT
```

3. Hubungkan sebagai *`user1`*, lalu buat pemetaan ke pengguna MySQL:

```
test=> CREATE USER MAPPING FOR user1 SERVER mysqldb OPTIONS (username 'myuser',
 password 'mypassword');
CREATE USER MAPPING
```

4. Buat tabel asing yang ditautkan ke tabel MySQL:

```
test=> CREATE FOREIGN TABLE mytab (a int, b text) SERVER mysqldb OPTIONS (dbname
 'test', table_name '');
CREATE FOREIGN TABLE
```

5. Jalankan kueri sederhana pada tabel asing:

```
test=> SELECT * FROM mytab;
a |   b
---+-------
1 | apple
(1 row)
```

6. Anda dapat menambahkan, mengubah, dan menghapus data dari tabel MySQL. Sebagai contoh:

```
test=> INSERT INTO mytab values (2, 'mango');
INSERT 0 1
```

Jalankan lagi kueri `SELECT` untuk melihat hasilnya:

```
test=> SELECT * FROM mytab ORDER BY 1;
 a |   b
---+-------
1 | apple
2 | mango
(2 rows)
```

Menggunakan enkripsi bergerak dengan ekstensi

Koneksi ke MySQL dari RDS for PostgreSQL menggunakan enkripsi bergerak (TLS/SSL) secara default. Namun, koneksi akan kembali ke non-enkripsi ketika konfigurasi klien dan server berbeda. Anda dapat menerapkan enkripsi untuk semua koneksi keluar dengan menentukan opsi `REQUIRE SSL` pada akun pengguna RDS for MySQL. Pendekatan yang sama ini juga berfungsi untuk akun pengguna MariaDB dan Aurora MySQL.

Untuk akun pengguna MySQL yang dikonfigurasi ke `REQUIRE SSL`, upaya koneksi akan gagal jika koneksi aman tidak dapat dibuat.

Untuk menerapkan enkripsi akun pengguna basis data MySQL yang sudah ada, Anda dapat menggunakan perintah `ALTER USER`. Sintaks dapat berbeda-beda, bergantung pada versi MySQL, seperti yang ditunjukkan pada tabel berikut. Untuk informasi selengkapnya, lihat ALTER USER di Manual Referensi MySQL.

MySQL 5.7, MySQL 8.0	MySQL 5.6
`ALTER USER 'user'@'%' REQUIRE SSL;`	`GRANT USAGE ON *.* to 'user'@'%' REQUIRE SSL;`

Untuk informasi ekstensi `mysql_fdw` selengkapnya, lihat dokumentasi mysql_fdw.

Bekerja dengan basis data Oracle menggunakan ekstensi oracle_fdw

Untuk mengakses basis data Oracle dari instans DB RDS for PostgreSQL, Anda dapat menginstal dan menggunakan ekstensi `oracle_fdw`. Ekstensi ini adalah pembungkus data asing untuk basis data Oracle. Untuk mempelajari ekstensi ini selengkapnya, lihat dokumentasi oracle_fdw.

Ekstensi `oracle_fdw` didukung di RDS for PostgreSQL 12.7, 13.3, dan versi yang lebih baru.

Topik

- Mengaktifkan ekstensi oracle_fdw
- Contoh: Menggunakan server asing yang terhubung ke basis data Amazon RDS for Oracle.
- Bekerja dengan enkripsi bergerak
- Memahami tampilan dan izin pg_user_mappings

Mengaktifkan ekstensi oracle_fdw

Untuk menggunakan ekstensi oracle_fdw, lakukan prosedur berikut.

Untuk mengaktifkan ekstensi oracle_fdw

- Jalankan perintah berikut menggunakan akun yang memiliki izin `rds_superuser`.

```
CREATE EXTENSION oracle_fdw;
```

Contoh: Menggunakan server asing yang terhubung ke basis data Amazon RDS for Oracle.

Contoh berikut ini menunjukkan penggunaan server asing yang terhubung ke basis data Amazon RDS for Oracle.

Untuk membuat server asing yang ditautkan ke basis data RDS for Oracle

1. Perhatikan hel-hal berikut ini di instans DB RDS for Oracle:
   - Titik Akhir
   - Port
   - Nama basis data
2. Membuat server asing.

```
test=> CREATE SERVER oradb FOREIGN DATA WRAPPER oracle_fdw OPTIONS (dbserver
 '//endpoint:port/DB_name');
CREATE SERVER
```

3. Berikan penggunaan kepada pengguna yang tidak memiliki hak istimewa `rds_superuser`, misalnya `user1`.

```
test=> GRANT USAGE ON FOREIGN SERVER oradb TO user1;
GRANT
```

4. Hubungkan sebagai `user1`, lalu buat pemetaan ke pengguna Oracle.

```
test=> CREATE USER MAPPING FOR user1 SERVER oradb OPTIONS (user 'oracleuser',
 password 'mypassword');
CREATE USER MAPPING
```

5. Buat tabel asing yang ditautkan ke tabel Oracle.

```
test=> CREATE FOREIGN TABLE mytab (a int) SERVER oradb OPTIONS (table 'MYTABLE');
CREATE FOREIGN TABLE
```

6. Kueri tabel asing.

```
test=>  SELECT * FROM mytab;
a
---
1
(1 row)
```

Jika kueri melaporkan kesalahan berikut, periksa grup keamanan dan daftar kontrol akses (ACL) untuk memastikan bahwa kedua instans dapat berkomunikasi.

```
ERROR: connection for foreign table "mytab" cannot be established
DETAIL: ORA-12170: TNS:Connect timeout occurred
```

Bekerja dengan enkripsi bergerak

Enkripsi PostgreSQL-to-Oracle bergerak didasarkan pada kombinasi parameter konfigurasi klien dan server. Untuk contoh menggunakan Oracle 21c, lihat About the Values for Negotiating Encryption and Integrity dalam dokumentasi Oracle. Klien yang digunakan untuk oracle_fdw di Amazon RDS dikonfigurasi dengan `ACCEPTED`, artinya enkripsi bergantung pada konfigurasi server basis data Oracle.

Jika basis data Anda menggunakan RDS for Oracle, lihat Enkripsi jaringan native Oracle untuk mengonfigurasi enkripsi.

Memahami tampilan dan izin pg_user_mappings

Katalog PostgreSQL `pg_user_mapping` menyimpan pemetaan dari pengguna RDS for PostgreSQL ke pengguna di server data asing (jarak jauh). Akses ke katalog dibatasi, tetapi Anda menggunakan tampilan `pg_user_mappings` untuk melihat pemetaan. Di bawah ini, Anda dapat menemukan contoh yang menunjukkan bagaimana izin berlaku dengan contoh basis data Oracle, tetapi informasi ini berlaku lebih umum untuk pembungkus data asing mana pun.

Pada output berikut, Anda dapat menemukan peran dan izin yang dipetakan ke tiga pengguna contoh yang berbeda. Pengguna `rdssu1` dan `rdssu2` adalah anggota dari peran `rds_superuser`, sedangkan `user1` tidak. Contoh menggunakan metacommand `psql` `\du` untuk daftar peran yang sudah ada.

```
test=>  \du
                                                    List of roles
    Role name    |                         Attributes                         |
 Member of
-----------------+------------------------------------------------------------
+-----------------------------------------------------------
 rdssu1          |                                                            |
 {rds_superuser}
 rdssu2          |                                                            |
 {rds_superuser}
 user1           |                                                            | {}
```

Semua pengguna, termasuk pengguna yang memiliki hak istimewa `rds_superuser`, diizinkan untuk melihat pemetaan pengguna (`umoptions`) mereka sendiri di tabel `pg_user_mappings`. Seperti yang ditunjukkan dalam contoh berikut, ketika `rdssu1` mencoba untuk mendapatkan semua pemetaan pengguna, kesalahan muncul meskipun hak istimewa `rdssu1` `rds_superuser`:

```
test=> SELECT * FROM pg_user_mapping;
ERROR: permission denied for table pg_user_mapping
```

Berikut ini adalah beberapa contoh.

```
test=> SET SESSION AUTHORIZATION rdssu1;
```

```
SET
test=> SELECT * FROM pg_user_mappings;
 umid  | srvid | srvname | umuser | usename    |            umoptions
-------+-------+---------+--------+------------+---------------------------------
 16414 | 16411 | oradb   |  16412 | user1      |
 16423 | 16411 | oradb   |  16421 | rdssu1     | {user=oracleuser,password=mypwd}
 16424 | 16411 | oradb   |  16422 | rdssu2     |
 (3 rows)

test=> SET SESSION AUTHORIZATION rdssu2;
SET
test=> SELECT * FROM pg_user_mappings;
 umid  | srvid | srvname | umuser | usename    |            umoptions
-------+-------+---------+--------+------------+---------------------------------
 16414 | 16411 | oradb   |  16412 | user1      |
 16423 | 16411 | oradb   |  16421 | rdssu1     |
 16424 | 16411 | oradb   |  16422 | rdssu2     | {user=oracleuser,password=mypwd}
 (3 rows)

test=> SET SESSION AUTHORIZATION user1;
SET
test=> SELECT * FROM pg_user_mappings;
 umid  | srvid | srvname | umuser | usename    |           umoptions
-------+-------+---------+--------+------------+-------------------------------
 16414 | 16411 | oradb   |  16412 | user1      | {user=oracleuser,password=mypwd}
 16423 | 16411 | oradb   |  16421 | rdssu1     |
 16424 | 16411 | oradb   |  16422 | rdssu2     |
 (3 rows)
```

Karena perbedaan implementasi antara `information_schema._pg_user_mappings` dan `pg_catalog.pg_user_mappings`, maka `rds_superuser` yang dibuat secara manual memerlukan izin tambahan untuk dapat melihat kata sandi di `pg_catalog.pg_user_mappings`.

`rds_superuser` tidak memerlukan izin tambahan untuk dapat melihat kata sandi di `information_schema._pg_user_mappings`.

Pengguna yang tidak memiliki peran `rds_superuser` dapat melihat kata sandi `pg_user_mappings` hanya dalam kondisi berikut:

- Pengguna saat ini adalah pengguna yang dipetakan dan memiliki server atau memegang hak istimewa `USAGE` atas server tersebut.
- Pengguna saat ini adalah pemilik server dan pemetaannya untuk `PUBLIC`.

Bekerja dengan basis data SQL Server menggunakan ekstensi tds_fdw

Anda dapat menggunakan ekstensi `tds_fdw` PostgreSQL untuk mengakses basis data yang mendukung protokol aliran data tabular (TDS), seperti basis data Sybase dan Microsoft SQL Server. Pembungkus data asing ini memungkinkan Anda terhubung dari instans DB RDS for PostgreSQL ke basis data yang menggunakan protokol TDS, seperti Amazon RDS for Microsoft SQL Server. Untuk informasi selengkapnya, lihat dokumentasi tds-fdw/tds_fdw di GitHub.

Ekstensi `tds_fdw` ini didukung di Amazon RDS for PostgreSQL versi 14.2, 13.6, dan rilis yang lebih baru.

Menyiapkan DB Aurora PostgreSQL untuk menggunakan ekstensi tds_fdw

Dalam prosedur berikut, Anda dapat menemukan contoh pengaturan dan penggunaan `tds_fdw` dengan instans DB RDS for PostgreSQ. Sebelum dapat terhubung ke basis data SQL Server menggunakan `tds_fdw`, Anda harus mendapatkan detail berikut untuk instans:

- Nama host atau titik akhir. Untuk instans DB RDS for SQL Server, Anda dapat menemukan titik akhir menggunakan Konsol. Pilih tab Konektivitas & keamanan dan lihat di bagian “Titik akhir dan port”.
- Nomor port. Nomor port default untuk Microsoft SQL Server adalah 1433.
- Nama basis data. Pengidentifikasi DB.

Anda juga perlu menyediakan akses di grup keamanan atau daftar kontrol akses (ACL) untuk port SQL Server, 1433. Baik instans DB RDS for PostgreSQL dan instans DB RDS for SQL Server memerlukan akses ke port 1433. Jika akses tidak dikonfigurasi dengan benar, Anda akan melihat pesan kesalahan seperti yang ada di bawah ini ketika mencoba mengueri Microsoft SQL Server:

```
ERROR: DB-Library error: DB #: 20009, DB Msg: Unable to connect:
Adaptive Server is unavailable or does not exist (mssql2019.aws-
region.rds.amazonaws.com), OS #: 0, OS Msg: Success, Level: 9
```

Untuk menggunakan tds_fdw agar terhubung ke basis data SQL Server

1. Hubungkan ke instans DB PostgreSQL menggunakan akun yang memiliki peran `rds_superuser`:

```
psql --host=your-DB-instance.aws-region.rds.amazonaws.com --port=5432 --
username=test --password
```

2. Instal ekstensi `tds_fdw`:

```
test=> CREATE EXTENSION tds_fdw;
CREATE EXTENSION
```

Setelah ekstensi diinstal di instans DB RDS for PostgreSQL, Anda dapat menyiapkan server asing.

Untuk membuat server asing

Lakukan tugas-tugas ini di instans DB RDS for PostgreSQL menggunakan akun yang memiliki hak istimewa `rds_superuser`.

1. Membuat server asing di instans DB RDS for PostgreSQL:

```
test=> CREATE SERVER sqlserverdb FOREIGN DATA WRAPPER tds_fdw OPTIONS
 (servername 'mssql2019.aws-region.rds.amazonaws.com', port '1433', database
 'tds_fdw_testing');
CREATE SERVER
```

   Untuk mengakses data non-ASCII di sisi SQLServer, buat tautan server dengan opsi character_set di instans DB RDS for PostgreSQL:

```
test=> CREATE SERVER sqlserverdb FOREIGN DATA WRAPPER tds_fdw OPTIONS (servername
 'mssql2019.aws-region.rds.amazonaws.com', port '1433', database 'tds_fdw_testing',
 character_set 'UTF-8');
CREATE SERVER
```

2. Berikan izin kepada pengguna yang tidak memiliki hak istimewa peran `rds_superuser`, misalnya `user1`:

```
test=> GRANT USAGE ON FOREIGN SERVER sqlserverdb TO user1;
```

3. Hubungkan sebagai user1 lalu buat pemetaan ke pengguna SQL Server:

```
test=> CREATE USER MAPPING FOR user1 SERVER sqlserverdb OPTIONS (username
 'sqlserveruser', password 'password');
```

```
CREATE USER MAPPING
```

4. Buat tabel asing yang ditautkan ke tabel SQL Server:

```
test=> CREATE FOREIGN TABLE mytab (a int) SERVER sqlserverdb OPTIONS (table
 'MYTABLE');
CREATE FOREIGN TABLE
```

5. Kueri tabel asing:

```
test=> SELECT * FROM mytab;
 a
---
 1
(1 row)
```

Menggunakan enkripsi bergerak untuk koneksi

Koneksi dari RDS for PostgreSQL ke SQL Server menggunakan enkripsi bergerak (TLS/SSL) bergantung pada konfigurasi basis data SQL Server. Jika SQL Server tidak dikonfigurasi untuk enkripsi, klien RDS for PostgreSQL yang membuat permintaan ke basis data SQL Server akan dikembalikan ke status tidak terenkripsi.

Anda dapat menerapkan enkripsi untuk koneksi ke instans DB RDS for SQL Server dengan menyetel parameter `rds.force_ssl`. Untuk mempelajari caranya, lihat Memaksa koneksi ke instans DB untuk menggunakan SSL. Untuk informasi konfigurasi SSL/TLS untuk RDS for SQL Server selengkapnya, lihat Menggunakan SSL dengan instans DB Microsoft SQL Server.

Bekerja dengan Ekstensi Bahasa Tepercaya untuk PostgreSQL

Ekstensi Bahasa Tepercaya untuk PostgreSQL adalah kit pengembangan sumber terbuka untuk mendesain ekstensi PostgreSQL. Ini memungkinkan Anda untuk membangun ekstensi PostgreSQL performa tinggi dan menjalankannya dengan aman di instans DB RDS for PostgreSQL. Dengan menggunakan Ekstensi Bahasa Tepercaya (TLE) untuk PostgreSQL, Anda dapat membuat ekstensi PostgreSQL yang mengikuti pendekatan terdokumentasi untuk memperluas fungsionalitas PostgreSQL. Lihat informasi selengkapnya, lihat Mengemas Objek Terkait ke dalam Ekstensi dalam dokumentasi PostgreSQL.

Salah satu manfaat utama TLE adalah Anda dapat menggunakannya di lingkungan yang tidak menyediakan akses ke sistem file yang mendasari instans PostgreSQL. Sebelumnya, penginstalan ekstensi baru memerlukan akses ke sistem file. TLE menghilangkan kendala ini. Ini menyediakan lingkungan pengembangan untuk membuat ekstensi baru untuk basis data PostgreSQL apa pun, termasuk yang berjalan di instans DB RDS for PostgreSQL.

TLE dirancang untuk mencegah akses ke sumber daya yang tidak aman untuk ekstensi yang Anda buat menggunakan TLE. Lingkungan runtime-nya membatasi dampak kerusakan ekstensi apa pun ke koneksi basis data tunggal. TLE juga memberi administrator basis data kontrol terperinci atas siapa saja yang dapat menginstal ekstensi, dan memberikan model izin untuk menjalankannya.

TLE didukung pada versi RDS for PostgreSQL berikut:

- Versi 15.2 dan versi 15 yang lebih tinggi.
- Versi 14.5 dan versi 14 yang lebih tinggi.
- Versi 13.12 dan versi 13 yang lebih tinggi.

Lingkungan pengembangan dan runtime Ekstensi Bahasa Tepercaya dikemas sebagai ekstensi PostgreSQL `pg_tle`, versi 1.0.1. Ini mendukung pembuatan ekstensi di JavaScript, Perl, Tcl, PL/PGSQL, dan SQL. Anda menginstal ekstensi `pg_tle` di instans RDS for PostgreSQL dengan cara yang sama seperti Anda menginstal ekstensi PostgreSQL lainnya. Setelah `pg_tle` disiapkan, developer dapat menggunakannya untuk membuat ekstensi PostgreSQL baru, yang dikenal sebagai ekstensi TLE.

Dalam topik berikut, Anda dapat menemukan informasi tentang cara mengatur Ekstensi Bahasa Tepercaya dan cara memulai membuat ekstensi TLE Anda sendiri.

Topik

Terminologi

Untuk membantu Anda lebih memahami Ekstensi Bahasa Tepercaya, lihat glosarium berikut untuk istilah yang digunakan dalam topik ini.

Ekstensi Bahasa Tepercaya untuk PostgreSQL

> Ekstensi Bahasa Tepercaya untuk PostgreSQL adalah nama resmi kit pengembangan sumber terbuka yang dikemas sebagai ekstensi `pg_tle`. Ini dapat digunakan pada sistem PostgreSQL apa pun. Untuk informasi selengkapnya, lihat aws/pg_tle di. GitHub

Ekstensi Bahasa Terpercaya

> Ekstensi Bahasa Tepercaya adalah singkatan dari Ekstensi Bahasa Tepercaya untuk PostgreSQL. Nama singkat ini dan singkatannya (TLE) juga digunakan dalam dokumentasi ini.

bahasa tepercaya

> Bahasa tepercaya adalah bahasa pemrograman atau skrip yang memiliki atribut keamanan tertentu. Misalnya, bahasa tepercaya biasanya membatasi akses ke sistem file, dan membatasi penggunaan properti jaringan tertentu. Kit pengembangan TLE dirancang untuk mendukung bahasa tepercaya. PostgreSQL mendukung beberapa bahasa yang berbeda yang digunakan untuk membuat ekstensi tepercaya atau tidak tepercaya. Sebagai contoh, lihat PL/Perl

Tepercaya dan Tidak Tepercaya dalam dokumentasi PostgreSQL. Saat Anda membuat ekstensi menggunakan Ekstensi Bahasa Tepercaya, ekstensi tersebut akan menggunakan mekanisme bahasa tepercaya secara inheren.

Ekstensi TLE

Ekstensi TLE adalah ekstensi PostgreSQL yang dibuat dengan menggunakan kit pengembangan Ekstensi Bahasa Tepercaya (TLE).

Persyaratan untuk menggunakan Ekstensi Bahasa Tepercaya untuk PostgreSQL

Berikut ini adalah persyaratan untuk menyiapkan dan menggunakan kit pengembangan TLE.

- Versi RDS for PostgreSQL – Ekstensi Bahasa Tepercaya didukung pada RDS for PostgreSQL versi 13.12 dan versi 13 yang lebih tinggi, 14.5 dan versi 14 yang lebih tinggi, dan 15.2 dan versi yang lebih tinggi saja.
  - Jika Anda perlu mengupgrade instans RDS for PostgreSQL, lihat Meningkatkan mesin DB PostgreSQL untuk Amazon RDS.
  - Jika belum memiliki instans DB Amazon RDS yang menjalankan PostgreSQL, Anda dapat membuatnya. Untuk informasi selengkapnya, lihat Instans DB RDS for PostgreSQL, lihat Membuat dan menghubungkan ke instans DB PostgreSQL.
- Memerlukan hak istimewa **`rds_superuser`** – Untuk menyiapkan dan mengonfigurasi ekstensi `pg_tle`, peran pengguna basis data Anda harus memiliki izin peran `rds_superuser`. Secara default, peran ini diberikan kepada `postgres` pengguna yang membuat Instans DB RDS for PostgreSQL.
- Memerlukan grup parameter DB kustom - instans DB RDS for PostgreSQL Anda harus dikonfigurasi dengan grup parameter DB kustom.
  - Jika instans DB RDS for PostgreSQL Anda tidak dikonfigurasi dengan grup parameter DB kustom, Anda harus membuatnya dan mengaitkannya dengan instans DB RDS for PostgreSQL Anda. Untuk ringkasan langkah-langkahnya, lihat Membuat dan menerapkan grup parameter DB kustom.
  - Jika instans DB RDS for PostgreSQL Anda sudah dikonfigurasi menggunakan grup parameter DB kustom, Anda dapat menyiapkan Ekstensi Bahasa Tepercaya. Untuk detailnya, lihat Menyiapkan Ekstensi Bahasa Tepercaya di instans DB RDS for PostgreSQL Anda.

Membuat dan menerapkan grup parameter DB kustom

Gunakan langkah-langkah berikut untuk membuat grup parameter DB kustom dan mengonfigurasi instans DB RDS for PostgreSQL untuk menggunakannya.

Konsol

Untuk membuat grup parameter DB kustom dan menggunakannya dengan instans DB RDS for PostgreSQL

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Pilih grup Parameter dari menu Amazon RDS.
3. Pilih Buat grup parameter.
4. Di halaman Detail grup parameter, masukkan informasi berikut.
   - Untuk Keluarga grup Parameter, pilih postgres14.
   - Untuk Jenis, pilih Grup Parameter DB.
   - Untuk Nama grup, berikan grup parameter Anda nama yang bermakna dalam konteks operasi Anda.
   - Untuk Deskripsi, masukkan deskripsi yang berguna sehingga orang lain di tim Anda dapat dengan mudah menemukannya.
5. Pilih Buat. Grup parameter DB kustom Anda dibuat di Wilayah AWS. Anda sekarang dapat mengubah instans DB RDS for PostgreSQL untuk menggunakannya dengan mengikuti langkah selanjutnya.
6. Pilih Basis data dari menu Amazon RDS.
7. Pilih instans DB RDS for PostgreSQL yang ingin digunakan dengan TLE dari yang tercantum tersebut, lalu pilih Ubah.
8. Di halaman Halaman Ubah pengaturan instans DB, cari Opsi Basis Data di bagian Konfigurasi tambahan dan pilih grup parameter DB kustom Anda dari pemilih.
9. Pilih Lanjutkan untuk menyimpan perubahan.
10. Pilih Terapkan langsung sehingga Anda dapat melanjutkan penyiapan instans DB RDS for PostgreSQL untuk menggunakan TLE.

Untuk melanjutkan penyiapan sistem untuk Ekstensi Bahasa Tepercaya, lihat Menyiapkan Ekstensi Bahasa Tepercaya di instans DB RDS for PostgreSQL Anda.

Untuk informasi selengkapnya tentang cara menggunakan Grup parameter DB, lihat Bekerja dengan grup parameter DB dalam instance DB.

AWS CLI

Anda dapat menghindari penentuan argumen `--region` saat menggunakan perintah CLI dengan mengonfigurasi AWS CLI dengan Wilayah AWS default. Untuk informasi selengkapnya, lihat Dasar-dasar konfigurasi di Panduan Pengguna AWS Command Line Interface.

Untuk membuat grup parameter DB kustom dan menggunakannya dengan instans DB RDS for PostgreSQL

1. Gunakan create-db-parameter-groupAWS CLIperintah untuk membuat grup parameter DB kustom berdasarkan untuk Anda. Wilayah AWS

   Untuk Linux, macOS, atau Unix:

   ```
   aws rds create-db-parameter-group \
     --region aws-region \
     --db-parameter-group-name custom-params-for-pg-tle \
     --db-parameter-group-family postgres14 \
     --description "My custom DB parameter group for Trusted Language Extensions"
   ```

   Untuk Windows:

   ```
   aws rds create-db-parameter-group ^
     --region aws-region ^
     --db-parameter-group-name custom-params-for-pg-tle ^
     --db-parameter-group-family postgres14 ^
     --description "My custom DB parameter group for Trusted Language Extensions"
   ```

   Grup parameter DB kustom Anda tersedia di Wilayah AWS Anda, sehingga Anda dapat mengubah instans DB RDS for PostgreSQL untuk menggunakannya.

2. Gunakan modify-db-instanceAWS CLIperintah untuk menerapkan grup parameter DB kustom Anda ke . RDS Anda untuk instance PostgreSQL DB. Perintah ini langsung me-reboot instans yang aktif.

   Untuk Linux, macOS, atau Unix:

   ```
   aws rds modify-db-instance \
   ```

```
  --region aws-region \
  --db-instance-identifier your-instance-name \
  --db-parameter-group-name custom-params-for-pg-tle \
  --apply-immediately
```

Untuk Windows:

```
aws rds modify-db-instance ^
  --region aws-region ^
  --db-instance-identifier your-instance-name ^
  --db-parameter-group-name custom-params-for-pg-tle ^
  --apply-immediately
```

Untuk melanjutkan penyiapan sistem untuk Ekstensi Bahasa Tepercaya, lihat Menyiapkan Ekstensi Bahasa Tepercaya di instans DB RDS for PostgreSQL Anda.

Untuk informasi selengkapnya, lihat Bekerja dengan grup parameter.

Menyiapkan Ekstensi Bahasa Tepercaya di instans DB RDS for PostgreSQL Anda

Langkah-langkah berikut mengasumsikan bahwa instans DB RDS for PostgreSQL Anda dikaitkan dengan grup parameter DB kustom. Anda dapat menggunakan AWS Management Console atau AWS CLI untuk langkah ini.

Saat menyiapkan Ekstensi Bahasa Tepercaya di instans DB RDS for PostgreSQL, Anda menginstalnya di basis data tertentu untuk digunakan oleh pengguna basis data yang memiliki izin pada basis data tersebut.

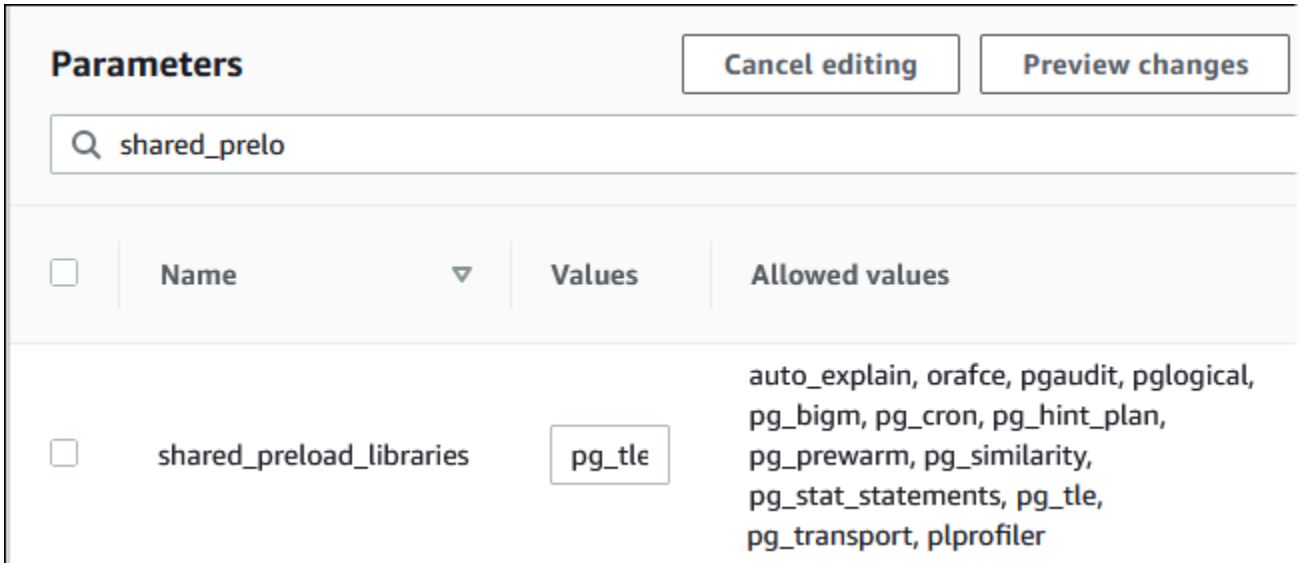
Konsol

Untuk menyiapkan Ekstensi Bahasa Tepercaya

Lakukan langkah-langkah berikut menggunakan akun yang merupakan anggota grup `rds_superuser` (peran).

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih instans DB RDS for PostgreSQL Anda.

3. Buka tab Konfigurasi untuk Instans DB RDS for PostgreSQL. Di antara detail Instans, temukan tautan Grup parameter.
4. Pilih tautan untuk membuka parameter kustom yang terkait dengan Instans DB RDS for PostgreSQL.
5. Di kolom pencarian Parameter, ketik `shared_pre` untuk menemukan parameter `shared_preload_libraries`.
6. Pilih Edit parameter untuk mengakses nilai properti.
7. Tambahkan `pg_tle` ke daftar di kolom Nilai. Gunakan koma untuk memisahkan item dalam daftar nilai.



8. Reboot instans DB RDS for PostgreSQL sehingga perubahan pada parameter `shared_preload_libraries` dapat diterapkan.
9. Ketika instans tersedia, verifikasi bahwa `pg_tle` telah diinisialisasi. Gunakan `psql` untuk terhubung ke instans DB RDS for PostgreSQL, lalu jalankan perintah berikut.

```
SHOW shared_preload_libraries;
shared_preload_libraries
--------------------------
rdsutils,pg_tle
(1 row)
```

10. Dengan ekstensi `pg_tle` yang diinisialisasi, Anda kini dapat membuat ekstensi.

```
CREATE EXTENSION pg_tle;
```

Anda dapat memverifikasi bahwa ekstensi diinstal dengan menggunakan metacommand `psql` berikut.

```
labdb=> \dx
                          List of installed extensions
  Name   | Version |   Schema   |                Description
---------+---------+------------+-------------------------------------------
 pg_tle  | 1.0.1   | pgtle      | Trusted-Language Extensions for PostgreSQL
 plpgsql | 1.0     | pg_catalog | PL/pgSQL procedural language
```

11. Berikan peran `pgtle_admin` ke nama pengguna utama yang Anda buat untuk instans DB RDS for PostgreSQL jika Anda menyiapkannya. Jika Anda menerima opsi default-nya, berarti nilainya `postgres`.

```
labdb=> GRANT pgtle_admin TO postgres;
GRANT ROLE
```

Anda dapat memverifikasi bahwa pemberian telah terjadi dengan menggunakan metacommand `psql` seperti yang ditunjukkan pada contoh berikut. Hanya peran `pgtle_admin` dan `postgres` yang ditampilkan dalam output. Untuk informasi selengkapnya, lihat Memahami peran rds_superuser.

```
labdb=> \du
                          List of roles
    Role name    |           Attributes            |               Member of
-----------------+---------------------------------
+-----------------------------------
pgtle_admin     | Cannot login                     | {}
postgres        | Create role, Create DB          +| {rds_superuser,pgtle_admin}
                | Password valid until infinity    |...
```

12. Tutup sesi `psql` menggunakan metacommand `\q`.

```
\q
```

Untuk mulai membuat ekstensi TLE, lihat Contoh: Membuat ekstensi bahasa tepercaya menggunakan SQL.

AWS CLI

Anda dapat menghindari penentuan argumen `--region` saat menggunakan perintah CLI dengan mengonfigurasi AWS CLI dengan Wilayah AWS default. Untuk informasi selengkapnya, lihat Dasar-dasar konfigurasi di Panduan Pengguna AWS Command Line Interface.

Untuk menyiapkan Ekstensi Bahasa Tepercaya

1. Gunakan modify-db-parameter-groupAWS CLIperintah untuk `pg_tle` menambah `shared_preload_libraries` parameter.

```
aws rds modify-db-parameter-group \
   --db-parameter-group-name custom-param-group-name \
   --parameters
 "ParameterName=shared_preload_libraries,ParameterValue=pg_tle,ApplyMethod=pending-
reboot" \
   --region aws-region
```

2. Gunakan reboot-db-instanceAWS CLIperintah untuk me-reboot dan menginisialisasi perpustakaan. `pg_tle`

```
aws rds reboot-db-instance \
    --db-instance-identifier your-instance \
    --region aws-region
```

3. Ketika instans tersedia, verifikasi bahwa `pg_tle` telah diinisialisasi. Gunakan `psql` untuk terhubung ke instans DB RDS for PostgreSQL, lalu jalankan perintah berikut.

```
SHOW shared_preload_libraries;
shared_preload_libraries
--------------------------
rdsutils,pg_tle
(1 row)
```

   Dengan `pg_tle` diinisialisasi, Anda sekarang dapat membuat ekstensi.

```
CREATE EXTENSION pg_tle;
```

4. Berikan peran `pgtle_admin` ke nama pengguna utama yang Anda buat untuk instans DB RDS for PostgreSQL jika Anda menyiapkannya. Jika Anda menerima opsi default-nya, berarti nilainya `postgres`.

```
GRANT pgtle_admin TO postgres;
GRANT ROLE
```

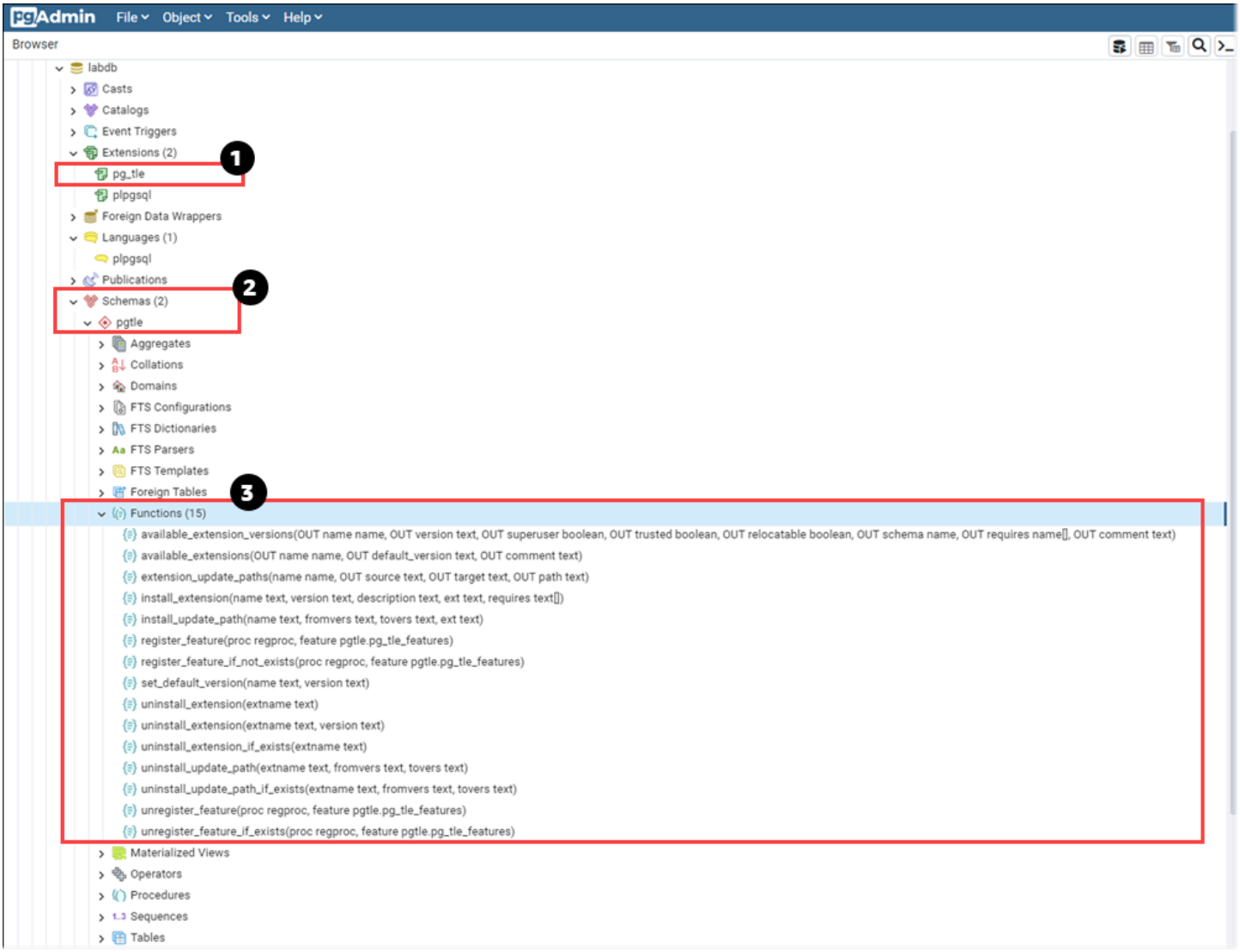
5. Tutup sesi `psql` seperti berikut.

```
labdb=> \q
```

Untuk mulai membuat ekstensi TLE, lihat Contoh: Membuat ekstensi bahasa tepercaya menggunakan SQL.

Ikhtisar Ekstensi Bahasa Tepercaya untuk PostgreSQL

Ekstensi Bahasa Tepercaya untuk PostgreSQL adalah ekstensi PostgreSQL yang Anda instal di instans DB RDS for PostgreSQL dengan cara yang sama seperti Anda menyiapkan ekstensi PostgreSQL lainnya. Pada gambar contoh basis data berikut alat klien pgAdmin, Anda dapat melihat beberapa komponen yang terdiri atas ekstensi `pg_tle`.



Anda dapat melihat detail berikut.

1. Kit pengembangan Ekstensi Bahasa Tepercaya (TLE) untuk PostgreSQL dikemas sebagai ekstensi `pg_tle`. Dengan demikian, `pg_tle` ditambahkan ke ekstensi yang tersedia untuk basis data tempat kit ini diinstal.
2. TLE memiliki skemanya sendiri, `pgtle`. Skema ini berisi fungsi pembantu (3) untuk menginstal dan mengelola ekstensi yang Anda buat.
3. TLE dilengkapi dengan banyak fungsi pembantu untuk menginstal, mendaftarkan, dan mengelola ekstensi Anda. Untuk mempelajari selengkapnya tentang fungsi ini, lihat Referensi fungsi untuk Trusted Language Extensions for PostgreSQL.

Komponen lain dari ekstensi `pg_tle` meliputi:

- Peran **`pgtle_admin`** – Peran `pgtle_admin` dibuat jika ekstensi `pg_tle` diinstal. Peran ini diistimewakan dan harus diperlakukan semestinya. Sebaiknya Anda mengikuti prinsip hak akses paling rendah saat memberikan peran `pgtle_admin` kepada pengguna basis data. Dengan kata lain, berikan peran `pgtle_admin` hanya kepada pengguna basis data yang diizinkan untuk membuat, menginstal, dan mengelola ekstensi TLE baru, seperti `postgres`.
- Tabel **`pgtle.feature_info`** – Tabel `pgtle.feature_info` adalah tabel yang dilindungi yang berisi informasi tentang TLE, hook, serta prosedur dan fungsi tersimpan kustom yang mereka gunakan. Jika Anda memiliki hak istimewa `pgtle_admin`, gunakan fungsi Ekstensi Bahasa Tepercaya berikut untuk menambahkan dan memperbarui informasi tersebut dalam tabel.
  - pgtle.register_feature
  - pgtle.register_feature_if_not_exists
  - pgtle.unregister_feature
  - pgtle.unregister_feature_if_exists

Membuat ekstensi TLE untuk RDS for PostgreSQL

Anda dapat menginstal ekstensi apa pun yang Anda buat dengan TLE di setiap instans DB RDS for PostgreSQL yang ekstensi `pg_tle`-nya telah diinstal. Ekstensi `pg_tle` dicakup ke basis data PostgreSQL tempat ekstensi diinstal. Ekstensi yang Anda buat menggunakan TLE dicakup ke basis data yang sama.

Gunakan berbagai fungsi `pgtle` untuk menginstal kode yang membentuk ekstensi TLE Anda. Semua fungsi Ekstensi Bahasa Tepercaya berikut memerlukan peran `pgtle_admin`.

- pgtle.install_extension
- pgtle.install_update_path
- pgtle.register_feature
- pgtle.register_feature_if_not_exists
- pgtle.set_default_version
- pgtle.uninstall_extension(nama)
- pgtle.uninstall_extension (nama, versi)
- pgtle.uninstall_extension_if_exists
- pgtle.uninstall_update_path
- pgtle.uninstall_update_path_if_exists

- pgtle.unregister_feature
- pgtle.unregister_feature_if_exists

Contoh: Membuat ekstensi bahasa tepercaya menggunakan SQL

Contoh berikut menunjukkan cara membuat ekstensi TLE bernama `pg_distance` yang berisi beberapa fungsi SQL untuk menghitung jarak menggunakan berbagai formula. Dalam daftar, Anda dapat menemukan fungsi untuk menghitung jarak Manhattan dan fungsi untuk menghitung jarak Euclidean. Untuk informasi selengkapnya tentang perbedaan antara formula ini, lihat Geometri taksi dan Geometri Euclidean di Wikipedia.

Anda dapat menggunakan contoh ini di instans DB RDS for PostgreSQL Anda sendiri jika Anda memiliki ekstensi `pg_tle` yang diatur seperti yang dijelaskan Menyiapkan Ekstensi Bahasa Tepercaya di instans DB RDS for PostgreSQL Anda.

> **Note**
>
> Untuk mengikuti prosedur ini, Anda harus memiliki hak istimewa peran `pgtle_admin`.

Untuk membuat contoh ekstensi TLE

Langkah-langkah berikut menggunakan contoh basis data bernama `labdb`. Basis data ini milik pengguna utama `postgres`. Peran `postgres` juga memiliki izin peran `pgtle_admin`.

1. Gunakan `psql` untuk terhubung ke . Instans DB RDS for PostgreSQL.

   ```
   psql --host=db-instance-123456789012.aws-region.rds.amazonaws.com
   --port=5432 --username=postgres --password --dbname=labdb
   ```

2. Buat ekstensi TLE bernama `pg_distance` dengan menyalin kode berikut dan menempelkannya ke konsol sesi `psql` Anda.

   ```
   SELECT pgtle.install_extension
   (
    'pg_distance',
    '0.1',
     'Distance functions for two points',
   $_pg_tle_$
       CREATE FUNCTION dist(x1 float8, y1 float8, x2 float8, y2 float8, norm int)
   ```

```
    RETURNS float8
    AS $$
      SELECT (abs(x2 - x1) ^ norm + abs(y2 - y1) ^ norm) ^ (1::float8 / norm);
    $$ LANGUAGE SQL;

    CREATE FUNCTION manhattan_dist(x1 float8, y1 float8, x2 float8, y2 float8)
    RETURNS float8
    AS $$
      SELECT dist(x1, y1, x2, y2, 1);
    $$ LANGUAGE SQL;

    CREATE FUNCTION euclidean_dist(x1 float8, y1 float8, x2 float8, y2 float8)
    RETURNS float8
    AS $$
      SELECT dist(x1, y1, x2, y2, 2);
    $$ LANGUAGE SQL;
$_pg_tle_$
);
```

Anda akan melihat output seperti berikut.

```
install_extension
---------------
 t
(1 row)
```

Artefak yang membentuk ekstensi `pg_distance` sekarang diinstal di basis data Anda. Artefak ini mencakup file kontrol dan kode untuk ekstensi, yang merupakan item yang harus ada sehingga ekstensi dapat dibuat menggunakan perintah `CREATE EXTENSION`. Dengan kata lain, Anda masih perlu membuat ekstensi agar fungsinya tersedia bagi pengguna basis data.

3. Untuk membuat ekstensi, gunakan perintah `CREATE EXTENSION` seperti yang Anda lakukan untuk ekstensi lainnya. Seperti ekstensi lainnya, pengguna basis data harus memiliki izin `CREATE` dalam basis data.

```
CREATE EXTENSION pg_distance;
```

4. Untuk menguji ekstensi TLE `pg_distance`, Anda dapat menggunakannya untuk menghitung Jarak Manhattan antara empat titik.

```
labdb=> SELECT manhattan_dist(1, 1, 5, 5);
```

```
8
```

Untuk menghitung Jarak Euclidean antara kumpulan titik yang sama, Anda dapat menggunakan berikut.

```
labdb=> SELECT euclidean_dist(1, 1, 5, 5);
5.656854249492381
```

Ekstensi `pg_distance` memuat fungsi dalam basis data dan membuatnya tersedia bagi setiap pengguna dengan izin pada basis data.

Mengubah ekstensi TLE

Untuk meningkatkan performa kueri untuk fungsi yang dikemas dalam ekstensi TLE ini, tambahkan dua atribut PostgreSQL berikut ke spesifikasinya.

- `IMMUTABLE` – Atribut `IMMUTABLE` memastikan bahwa pengoptimal kueri dapat menggunakan pengoptimalan untuk meningkatkan waktu respons kueri. Untuk informasi selengkapnya, lihat Kategori Volatilitas Fungsi dalam dokumentasi PostgreSQL.
- `PARALLEL SAFE` – Atribut `PARALLEL SAFE` adalah atribut lain yang memungkinkan PostgreSQL menjalankan fungsi dalam mode paralel. Untuk informasi selengkapnya, lihat CREATE FUNCTION dalam dokumentasi PostgreSQL.

Dalam contoh berikut, Anda dapat melihat bagaimana fungsi `pgtle.install_update_path` digunakan untuk menambahkan atribut ini ke setiap fungsi guna membuat ekstensi TLE `pg_distance` versi `0.2`. Untuk informasi selengkapnya tentang fungsi ini, lihat pgtle.install_update_path. Anda harus memiliki peran `pgtle_admin` untuk melakukan tugas ini.

Untuk memperbarui ekstensi TLE yang ada dan menentukan versi default

1. Hubungkan ke instans DB RDS for PostgreSQL Anda menggunakan `psql` atau alat klien lainnya, seperti pgAdmin.

   ```
   psql --host=db-instance-123456789012.aws-region.rds.amazonaws.com
   --port=5432 --username=postgres --password --dbname=labdb
   ```

2. Buat ekstensi TLE yang ada dengan menyalin kode berikut dan menempelkannya ke konsol sesi `psql` Anda.

```
SELECT pgtle.install_update_path
(
 'pg_distance',
 '0.1',
 '0.2',
$_pg_tle_$
    CREATE OR REPLACE FUNCTION dist(x1 float8, y1 float8, x2 float8, y2 float8,
 norm int)
    RETURNS float8
    AS $$
      SELECT (abs(x2 - x1) ^ norm + abs(y2 - y1) ^ norm) ^ (1::float8 / norm);
    $$ LANGUAGE SQL IMMUTABLE PARALLEL SAFE;

    CREATE OR REPLACE FUNCTION manhattan_dist(x1 float8, y1 float8, x2 float8, y2
 float8)
    RETURNS float8
    AS $$
      SELECT dist(x1, y1, x2, y2, 1);
    $$ LANGUAGE SQL IMMUTABLE PARALLEL SAFE;

    CREATE OR REPLACE FUNCTION euclidean_dist(x1 float8, y1 float8, x2 float8, y2
 float8)
    RETURNS float8
    AS $$
      SELECT dist(x1, y1, x2, y2, 2);
    $$ LANGUAGE SQL IMMUTABLE PARALLEL SAFE;
$_pg_tle_$
);
```

Anda akan melihat hasil yang mirip dengan berikut ini.

```
install_update_path
---------------------
 t
(1 row)
```

Anda dapat menjadikan versi ekstensi ini sebagai versi default, sehingga pengguna basis data tidak perlu menentukan versi saat mereka membuat atau memperbarui ekstensi di basis data mereka.

3. Untuk menentukan bahwa versi modifikasi (versi 0.2) ekstensi TLE Anda adalah versi default, gunakan fungsi `pgtle.set_default_version` seperti yang ditunjukkan pada contoh berikut.

```
SELECT pgtle.set_default_version('pg_distance', '0.2');
```

   Untuk informasi selengkapnya tentang fungsi ini, lihat pgtle.set_default_version.

4. Dengan kode yang diterapkan, Anda dapat memperbarui ekstensi TLE yang diinstal seperti biasa, dengan menggunakan perintah `ALTER EXTENSION ... UPDATE`, seperti yang ditunjukkan di sini:

```
ALTER EXTENSION pg_distance UPDATE;
```

Menghapus ekstensi TLE dari basis data

Anda dapat menghapus ekstensi TLE Anda dengan menggunakan perintah `DROP EXTENSION` dengan cara yang sama seperti yang Anda lakukan untuk ekstensi PostgreSQL lainnya. Menghapus ekstensi tidak menghapus file penginstalan yang membentuk ekstensi, yang memungkinkan pengguna membuat ulang ekstensi. Untuk menghapus ekstensi dan file penginstalannya, lakukan proses dua langkah berikut.

Untuk menghilangkan ekstensi TLE dan menghapus file penginstalannya

1. Gunakan `psql` atau alat klien lain untuk terhubung ke instans DB RDS for PostgreSQL.

```
psql --host=.111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password --dbname=dbname
```

2. Hilangkan ekstensi seperti yang Anda lakukan pada ekstensi PostgreSQL.

```
DROP EXTENSION your-TLE-extension
```

   Misalnya, jika Anda membuat ekstensi `pg_distance` seperti yang dijelaskan dalamContoh: Membuat ekstensi bahasa tepercaya menggunakan SQL, Anda dapat menghilangkan ekstensi sebagai berikut.

```
DROP EXTENSION pg_distance;
```

Anda melihat output yang mengonfirmasi bahwa ekstensi telah dihilangkan, sebagai berikut.

```
DROP EXTENSION
```

Pada titik ini, ekstensi tidak lagi aktif dalam basis data. Namun, file penginstalan dan file kontrolnya masih tersedia di basis data, sehingga pengguna basis data dapat membuat ekstensi lagi jika mereka menghendaki.

- Jika ingin membiarkan file ekstensi tetap utuh sehingga pengguna basis data dapat membuat ekstensi TLE, Anda dapat berhenti di sini.
- Jika Anda ingin menghapus semua file yang membentuk ekstensi, lanjutkan ke langkah berikutnya.

3. Untuk menghapus semua file penginstalan untuk ekstensi Anda, gunakan fungsi `pgtle.uninstall_extension`. Fungsi ini menghapus semua kode dan file kontrol untuk ekstensi Anda.

```
SELECT pgtle.uninstall_extension('your-tle-extension-name');
```

Misalnya, untuk menghapus semua file penginstalan `pg_distance`, gunakan perintah berikut.

```
SELECT pgtle.uninstall_extension('pg_distance');
 uninstall_extension
---------------------
 t
(1 row)
```

Meng-uninstal Ekstensi Bahasa Tepercaya untuk PostgreSQL

Jika tidak ingin lagi membuat ekstensi TLE Anda sendiri menggunakan TLE, Anda dapat menghilangkan ekstensi `pg_tle` dan menghapus semua artefak. Tindakan ini mencakup menghilangkan ekstensi TLE apa pun dalam basis data dan menghilangkan skema `pgtle`.

Untuk menghilangkan ekstensi **`pg_tle`** dan skema dari basis data

1. Gunakan `psql` atau alat klien lain untuk terhubung ke instans DB RDS for PostgreSQL.

```
psql --host=.111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password --dbname=dbname
```

2. Hilangkan ekstensi pg_tle dari basis data. Jika basis data memiliki ekstensi TLE Anda sendiri yang masih berjalan di basis data, Anda juga harus menghilangkan ekstensi tersebut. Untuk melakukannya, Anda dapat menggunakan kata kunci CASCADE, seperti yang ditunjukkan berikut ini.

   ```
   DROP EXTENSION pg_tle CASCADE;
   ```

   Jika ekstensi pg_tle masih tidak aktif dalam basis data, Anda tidak perlu menggunakan kata kunci CASCADE.

3. Hilangkan skema pgtle. Tindakan ini akan menghapus semua fungsi manajemen dari basis data.

   ```
   DROP SCHEMA pgtle CASCADE;
   ```

   Perintah ini menampilkan hasil berikut setelah proses selesai.

   ```
   DROP SCHEMA
   ```

   Ekstensi pg_tle, skema dan fungsinya, serta semua artefak dihapus. Untuk membuat ekstensi baru menggunakan TLE, lanjutkan proses penyiapan lagi. Untuk informasi selengkapnya, lihat Menyiapkan Ekstensi Bahasa Tepercaya di instans DB RDS for PostgreSQL Anda.

Menggunakan hook PostgreSQL dengan ekstensi TLE

Hook adalah mekanisme callback yang tersedia di PostgreSQL yang memungkinkan pengembang memanggil fungsi kustom atau rutinitas lainnya selama operasi basis data reguler. Kit pengembangan TLE mendukung hook PostgreSQL sehingga Anda dapat mengintegrasikan fungsi kustom dengan perilaku PostgreSQL saat runtime. Misalnya, Anda dapat menggunakan hook untuk mengaitkan proses autentikasi dengan kode kustom Anda sendiri, atau mengubah proses perencanaan dan eksekusi kueri untuk kebutuhan spesifik Anda.

Ekstensi TLE Anda dapat menggunakan hook. Jika hook cakupannya global, ini berlaku di semua basis data. Oleh karena itu, jika ekstensi TLE Anda menggunakan hook global, Anda perlu membuat ekstensi TLE di semua basis data yang dapat diakses pengguna.

Saat menggunakan ekstensi `pg_tle` untuk membuat Ekstensi Bahasa Tepercaya Anda sendiri, Anda dapat menggunakan hook yang tersedia dari SQL API untuk membangun fungsi ekstensi. Anda harus mendaftarkan hook apa pun di `pg_tle`. Untuk beberapa hook, Anda mungkin juga perlu mengatur berbagai parameter konfigurasi. Misalnya, hook pemeriksaan `passcode` dapat diatur ke aktif, nonaktif, wajib. Untuk informasi selengkapnya tentang persyaratan khusus untuk hook `pg_tle` yang tersedia, lihat Referensi hook untuk Trusted Language Extensions for PostgreSQL.

Contoh: Membuat ekstensi yang menggunakan hook PostgreSQL

Contoh yang dibahas di bagian ini menggunakan hook PostgreSQL untuk memeriksa kata sandi yang diberikan selama operasi SQL tertentu dan mencegah pengguna basis data mengatur kata sandi mereka ke salah satu basis data yang tercantum dalam tabel `password_check.bad_passwords`. Tabel berisi sepuluh besar pilihan kata sandi yang paling umum digunakan, tetapi mudah dipecahkan.

Untuk menyiapkan contoh ini di instans DB RDS for PostgreSQL, Anda harus sudah menginstal Ekstensi Bahasa Tepercaya. Untuk detailnya, lihat Menyiapkan Ekstensi Bahasa Tepercaya di instans DB RDS for PostgreSQL Anda.

Untuk menyiapkan contoh hook pemeriksaan kata sandi

1. Gunakan `psql` untuk terhubung ke . Instans DB RDS for PostgreSQL.

```
psql --host=db-instance-123456789012.aws-region.rds.amazonaws.com
--port=5432 --username=postgres --password --dbname=labdb
```

2. Salin kode dari Daftar kode hook pemeriksaan kata sandi dan tempel ke basis data Anda.

```
SELECT pgtle.install_extension (
  'my_password_check_rules',
  '1.0',
  'Do not let users use the 10 most commonly used passwords',
$_pgtle_$
  CREATE SCHEMA password_check;
  REVOKE ALL ON SCHEMA password_check FROM PUBLIC;
  GRANT USAGE ON SCHEMA password_check TO PUBLIC;
```

```
 CREATE TABLE password_check.bad_passwords (plaintext) AS
 VALUES
   ('123456'),
   ('password'),
   ('12345678'),
   ('qwerty'),
   ('123456789'),
   ('12345'),
   ('1234'),
   ('111111'),
   ('1234567'),
   ('dragon');
 CREATE UNIQUE INDEX ON password_check.bad_passwords (plaintext);

 CREATE FUNCTION password_check.passcheck_hook(username text, password text,
password_type pgtle.password_types, valid_until timestamptz, valid_null boolean)
 RETURNS void AS $$
   DECLARE
     invalid bool := false;
   BEGIN
     IF password_type = 'PASSWORD_TYPE_MD5' THEN
       SELECT EXISTS(
         SELECT 1
         FROM password_check.bad_passwords bp
         WHERE ('md5' || md5(bp.plaintext || username)) = password
       ) INTO invalid;
       IF invalid THEN
         RAISE EXCEPTION 'Cannot use passwords from the common password
dictionary';
       END IF;
     ELSIF password_type = 'PASSWORD_TYPE_PLAINTEXT' THEN
       SELECT EXISTS(
         SELECT 1
         FROM password_check.bad_passwords bp
         WHERE bp.plaintext = password
       ) INTO invalid;
       IF invalid THEN
         RAISE EXCEPTION 'Cannot use passwords from the common common password
dictionary';
       END IF;
     END IF;
   END
 $$ LANGUAGE plpgsql SECURITY DEFINER;
```

```
  GRANT EXECUTE ON FUNCTION password_check.passcheck_hook TO PUBLIC;

  SELECT pgtle.register_feature('password_check.passcheck_hook', 'passcheck');
$_pgtle_$
);
```

Setelah ekstensi dimuat ke basis data, Anda akan melihat output seperti berikut.

```
 install_extension
-------------------
 t
(1 row)
```

3. Jika masih terhubung ke basis data, Anda kini dapat membuat ekstensi.

```
CREATE EXTENSION my_password_check_rules;
```

4. Anda dapat mengonfirmasi bahwa ekstensi telah dibuat dalam basis data dengan menggunakan metacommand `psql` berikut.

```
\dx
                        List of installed extensions
          Name           | Version |   Schema   |
 Description
-------------------------+---------+------------
+-------------------------------------------------------------
 my_password_check_rules | 1.0     | public     | Prevent use of any of the top-ten
 most common bad passwords
 pg_tle                  | 1.0.1   | pgtle      | Trusted-Language Extensions for
 PostgreSQL
 plpgsql                 | 1.0     | pg_catalog | PL/pgSQL procedural language
(3 rows)
```

5. Buka sesi terminal lain yang ingin digunakan dengan AWS CLI. Anda perlu mengubah grup parameter DB kustom untuk mengaktifkan hook pemeriksaan kata sandi. Untuk melakukannya, gunakan perintah [modify-db-parameter-group](#)CLI seperti yang ditunjukkan pada contoh berikut.

```
aws rds modify-db-parameter-group \
    --region aws-region \
    --db-parameter-group-name your-custom-parameter-group \
```

```
    --parameters
 "ParameterName=pgtle.enable_password_check,ParameterValue=on,ApplyMethod=immediate"
```

Jika parameter berhasil dihidupkan, Anda akan melihat output seperti berikut.

```
(
    "DBParameterGroupName": "docs-lab-parameters-for-tle"
}
```

Mungkin diperlukan waktu beberapa menit agar perubahan pada pengaturan grup parameter diterapkan. Akan tetapi, parameter ini bersifat dinamis, jadi Anda tidak perlu memulai ulang instans DB RDS for PostgreSQL DB untuk menerapkan pengaturan.

6. Buka sesi `psql` dan kirim kueri ke basis data untuk memverifikasi bahwa hook password_check telah diaktifkan.

```
labdb=> SHOW pgtle.enable_password_check;
pgtle.enable_password_check
----------------------------
on
(1 row)
```

Hook password_check kini aktif. Anda dapat mengujinya dengan membuat peran baru dan menggunakan salah satu kata sandi yang buruk, seperti pada contoh berikut.

```
CREATE ROLE test_role PASSWORD 'password';
ERROR:  Cannot use passwords from the common password dictionary
CONTEXT:  PL/pgSQL function
 password_check.passcheck_hook(text,text,pgtle.password_types,timestamp with time
 zone,boolean) line 21 at RAISE
SQL statement "SELECT password_check.passcheck_hook(
    $1::pg_catalog.text,
    $2::pg_catalog.text,
    $3::pgtle.password_types,
    $4::pg_catalog.timestamptz,
    $5::pg_catalog.bool)"
```

Output ini telah diformat agar mudah dibaca.

Contoh berikut menunjukkan bahwa perilaku `\password` metacommand interaktif `pgsql` juga dipengaruhi oleh hook password_check.

```
postgres=> SET password_encryption TO 'md5';
SET
postgres=> \password
Enter new password for user "postgres":*****
Enter it again:*****
ERROR:  Cannot use passwords from the common password dictionary
CONTEXT:  PL/pgSQL function
 password_check.passcheck_hook(text,text,pgtle.password_types,timestamp with time
 zone,boolean) line 12 at RAISE
SQL statement "SELECT password_check.passcheck_hook($1::pg_catalog.text,
 $2::pg_catalog.text, $3::pgtle.password_types, $4::pg_catalog.timestamptz,
 $5::pg_catalog.bool)"
```

Anda dapat menghilangkan ekstensi TLE ini dan meng-uninstal file sumbernya jika menghendaki. Untuk informasi selengkapnya, lihat Menghapus ekstensi TLE dari basis data.

Daftar kode hook pemeriksaan kata sandi

Kode contoh yang ditampilkan di sini menentukan spesifikasi untuk ekstensi TLE `my_password_check_rules`. Jika kode ini disalin dan ditempelkan ke basis data, kode untuk ekstensi `my_password_check_rules` akan dimuat ke dalam basis data, dan hook `password_check` akan didaftarkan untuk digunakan oleh ekstensi.

```
SELECT pgtle.install_extension (
  'my_password_check_rules',
  '1.0',
  'Do not let users use the 10 most commonly used passwords',
$_pgtle_$
  CREATE SCHEMA password_check;
  REVOKE ALL ON SCHEMA password_check FROM PUBLIC;
  GRANT USAGE ON SCHEMA password_check TO PUBLIC;

  CREATE TABLE password_check.bad_passwords (plaintext) AS
  VALUES
    ('123456'),
    ('password'),
    ('12345678'),
    ('qwerty'),
    ('123456789'),
```

```
    ('12345'),
    ('1234'),
    ('111111'),
    ('1234567'),
    ('dragon');
  CREATE UNIQUE INDEX ON password_check.bad_passwords (plaintext);

  CREATE FUNCTION password_check.passcheck_hook(username text, password text,
 password_type pgtle.password_types, valid_until timestamptz, valid_null boolean)
  RETURNS void AS $$
    DECLARE
      invalid bool := false;
    BEGIN
      IF password_type = 'PASSWORD_TYPE_MD5' THEN
        SELECT EXISTS(
          SELECT 1
          FROM password_check.bad_passwords bp
          WHERE ('md5' || md5(bp.plaintext || username)) = password
        ) INTO invalid;
        IF invalid THEN
          RAISE EXCEPTION 'Cannot use passwords from the common password dictionary';
        END IF;
      ELSIF password_type = 'PASSWORD_TYPE_PLAINTEXT' THEN
        SELECT EXISTS(
          SELECT 1
          FROM password_check.bad_passwords bp
          WHERE bp.plaintext = password
        ) INTO invalid;
        IF invalid THEN
          RAISE EXCEPTION 'Cannot use passwords from the common common password
 dictionary';
        END IF;
      END IF;
    END
  $$ LANGUAGE plpgsql SECURITY DEFINER;

  GRANT EXECUTE ON FUNCTION password_check.passcheck_hook TO PUBLIC;

  SELECT pgtle.register_feature('password_check.passcheck_hook', 'passcheck');
$_pgtle_$
);
```

Menggunakan Jenis Data Kustom di TLE

PostgreSQL mendukung perintah untuk mendaftarkan jenis dasar baru (juga dikenal sebagai jenis skalar) untuk secara efisien menangani struktur data yang kompleks dalam basis data Anda. Jenis dasar memungkinkan Anda menyesuaikan bagaimana data disimpan secara internal, dan cara mengonversinya ke dan dari representasi tekstual eksternal. Jenis data khusus ini sangat membantu saat memperluas PostgreSQL untuk mendukung domain fungsional di mana jenis bawaan seperti angka atau teks tidak dapat menyediakan semantik pencarian yang memadai.

RDS for PostgreSQL memungkinkan Anda membuat jenis data kustom dalam ekstensi bahasa tepercaya dan menentukan fungsi yang mendukung operasi SQL dan indeks untuk jenis data baru ini. Jenis data kustom tersedia untuk versi berikut:

- RDS for PostgreSQL 15.4 dan versi 15 yang lebih tinggi
- RDS for PostgreSQL 14.9 dan versi 14 yang lebih tinggi
- RDS for PostgreSQL 13.12 dan versi 13 yang lebih tinggi

Untuk informasi selengkapnya, lihat Jenis Dasar Bahasa Tepercaya.

Referensi fungsi untuk Trusted Language Extensions for PostgreSQL

Lihat dokumentasi referensi berikut tentang fungsi yang tersedia di Trusted Language Extensions for PostgreSQL. Gunakan fungsi-fungsi ini untuk menginstal, mendaftarkan, memperbarui, dan mengelola ekstensi TLE, yaitu ekstensi PostgreSQL yang Anda kembangkan menggunakan kit pengembangan Trusted Language Extensions.

Topik

- pgtle.available_extensions
- pgtle.available_extension_versions
- pgtle.extension_update_paths
- pgtle.install_extension
- pgtle.install_update_path
- pgtle.register_feature
- pgtle.register_feature_if_not_exists
- pgtle.set_default_version

- pgtle.uninstall_extension(nama)
- pgtle.uninstall_extension (nama, versi)
- pgtle.uninstall_extension_if_exists
- pgtle.uninstall_update_path
- pgtle.uninstall_update_path_if_exists
- pgtle.unregister_feature
- pgtle.unregister_feature_if_exists

pgtle.available_extensions

Fungsi `pgtle.available_extensions` adalah fungsi pengembalian set. Fungsi ini mengembalikan semua ekstensi TLE yang tersedia dalam basis data. Setiap baris yang dikembalikan berisi informasi tentang satu ekstensi TLE.

Prototipe fungsi

```
pgtle.available_extensions()
```

Peran

Tidak ada.

Argumen

Tidak ada.

Output

- `name` – Nama ekstensi TLE.
- `default_version` – Versi ekstensi TLE yang akan digunakan ketika `CREATE EXTENSION` dipanggil tanpa menentukan versi.
- `description` – Penjelasan lebih mendetail tentang ekstensi TLE.

Contoh penggunaan

```
SELECT * FROM pgtle.available_extensions();
```

pgtle.available_extension_versions

Fungsi `available_extension_versions` adalah fungsi pengembalian set. Fungsi ini mengembalikan daftar semua ekstensi TLE yang tersedia dan versinya. Setiap baris berisi informasi tentang versi tertentu dari ekstensi TLE yang diberikan, termasuk apakah versi tersebut memerlukan peran tertentu.

Prototipe fungsi

```
pgtle.available_extension_versions()
```

Peran

Tidak ada.

Argumen

Tidak ada.

Output

- `name` – Nama ekstensi TLE.
- `version` – Versi ekstensi TLE.
- `superuser` – Nilai ini selalu `false` untuk ekstensi TLE Anda. Izin yang diperlukan untuk membuat ekstensi TLE atau memperbaruinya sama dengan izin untuk membuat objek lain dalam basis data tertentu.
- `trusted` – Nilai ini selalu `false` untuk ekstensi TLE.
- `relocatable` – Nilai ini selalu `false` untuk ekstensi TLE.
- `schema` – Menentukan nama skema yang menginstal ekstensi TLE.
- `requires` – Array yang berisi nama ekstensi lain yang dibutuhkan oleh ekstensi TLE ini.
- `description` – Deskripsi mendetail tentang ekstensi TLE.

Lihat informasi selengkapnya tentang nilai output di Packaging Related Objects into an Extension > Extension Files dalam dokumentasi PostgreSQL.

Contoh penggunaan

```
SELECT * FROM pgtle.available_extension_versions();
```

pgtle.extension_update_paths

Fungsi `extension_update_paths` adalah fungsi pengembalian set. Fungsi ini mengembalikan daftar semua jalur pembaruan yang memungkinkan untuk ekstensi TLE. Setiap baris menyertakan peningkatan atau penurunan versi yang tersedia untuk ekstensi TLE tersebut.

Prototipe fungsi

```
pgtle.extension_update_paths(name)
```

Peran

Tidak ada.

Argumen

`name` – Nama ekstensi TLE untuk mendapatkan jalur peningkatan.

Output

- `source` – Versi sumber untuk pembaruan.
- `target` – Versi target untuk pembaruan.
- `path` – Jalur peningkatan yang digunakan untuk memperbarui ekstensi TLE dari versi `source` ke versi `target`, misalnya, `0.1--0.2`.

Contoh penggunaan

```
SELECT * FROM pgtle.extension_update_paths('your-TLE');
```

pgtle.install_extension

Fungsi `install_extension` memungkinkan Anda menginstal artefak yang membentuk ekstensi TLE Anda di basis data. Selanjutnya, ekstensi ini dapat dibuat menggunakan perintah `CREATE EXTENSION`.

Prototipe fungsi

```
pgtle.install_extension(name text, version text, description text, ext text, requires
 text[] DEFAULT NULL::text[])
```

Peran

Tidak ada.

Argumen

- `name` – Nama ekstensi TLE. Nilai ini digunakan saat memanggil `CREATE EXTENSION`.
- `version` – Versi ekstensi TLE.
- `description` – Penjelasan mendetail tentang ekstensi TLE. Deskripsi ini ditampilkan di kolom `comment` pada `pgtle.available_extensions()`.
- `ext` – Isi ekstensi TLE. Nilai ini berisi objek seperti fungsi.
- `requires` – Parameter opsional yang menentukan dependensi untuk ekstensi TLE ini. Ekstensi `pg_tle` secara otomatis ditambahkan sebagai dependensi.

Banyak dari argumen ini sama dengan yang disertakan dalam file kontrol ekstensi untuk menginstal ekstensi PostgreSQL pada sistem file instans PostgreSQL. Untuk informasi selengkapnya, lihat File Ekstensi dalam Mengemas Objek Terkait menjadi Ekstensi dalam dokumentasi PostgreSQL.

Output

Fungsi ini akan mengembalikan `OK` jika berhasil, dan `NULL` jika terjadi kesalahan.

- `OK` – Ekstensi TLE telah berhasil diinstal di basis data.
- `NULL` – Ekstensi TLE belum berhasil diinstal di basis data.

Contoh penggunaan

```
SELECT pgtle.install_extension(
 'pg_tle_test',
 '0.1',
 'My first pg_tle extension',
$_pgtle_$
  CREATE FUNCTION my_test()
  RETURNS INT
  AS $$
    SELECT 42;
  $$ LANGUAGE SQL IMMUTABLE;
$_pgtle_$
);
```

pgtle.install_update_path

Fungsi `install_update_path` menyediakan jalur pembaruan antara dua versi ekstensi TLE yang berbeda. Dengan fungsi ini, pengguna dapat memperbarui versi ekstensi TLE menggunakan sintaks `ALTER EXTENSION ... UPDATE`.

Prototipe fungsi

```
pgtle.install_update_path(name text, fromvers text, tovers text, ext text)
```

Peran

`pgtle_admin`

Argumen

- `name` – Nama ekstensi TLE. Nilai ini digunakan saat memanggil `CREATE EXTENSION`.
- `fromvers` – Versi sumber ekstensi TLE untuk peningkatan.
- `tovers` – Versi tujuan ekstensi TLE untuk peningkatan.
- `ext` – Isi pembaruan. Nilai ini berisi objek seperti fungsi.

Output

Tidak ada.

Contoh penggunaan

```
SELECT pgtle.install_update_path('pg_tle_test', '0.1', '0.2',
  $_pgtle_$
    CREATE OR REPLACE FUNCTION my_test()
    RETURNS INT
    AS $$
      SELECT 21;
    $$ LANGUAGE SQL IMMUTABLE;
  $_pgtle_$
);
```

pgtle.register_feature

Fungsi `register_feature` menambahkan fitur PostgreSQL internal yang ditentukan ke tabel `pgtle.feature_info`. Hook PostgreSQL adalah contoh dari fitur PostgreSQL internal. Kit

pengembangan Trusted Language Extensions mendukung penggunaan hook PostgreSQL. Saat ini, fungsi ini mendukung fitur berikut.

- `passcheck` – Mendaftarkan hook pemeriksaan kata sandi dengan prosedur atau fungsi Anda yang menyesuaikan perilaku pemeriksaan kata sandi PostgreSQL.

Prototipe fungsi

```
pgtle.register_feature(proc regproc, feature pg_tle_feature)
```

Peran

`pgtle_admin`

Argumen

- `proc` – Nama prosedur atau fungsi yang disimpan untuk digunakan dengan fitur tersebut.
- `feature` – Nama fitur `pg_tle` (seperti `passcheck`) untuk mendaftar dengan fungsi.

Output

Tidak ada.

Contoh penggunaan

```
SELECT pgtle.register_feature('pw_hook', 'passcheck');
```

pgtle.register_feature_if_not_exists

Fungsi `pgtle.register_feature_if_not_exists` menambahkan fitur PostgreSQL yang ditentukan ke tabel `pgtle.feature_info` dan mengidentifikasi ekstensi TLE atau prosedur atau fungsi lain yang menggunakan fitur tersebut. Untuk informasi selengkapnya tentang hook dan Trusted Language Extensions, lihat Menggunakan hook PostgreSQL dengan ekstensi TLE.

Prototipe fungsi

```
pgtle.register_feature_if_not_exists(proc regproc, feature pg_tle_feature)
```

Peran

`pgtle_admin`

Argumen

- `proc` – Nama prosedur atau fungsi tersimpan yang berisi logika (kode) untuk digunakan sebagai fitur bagi ekstensi TLE Anda. Misalnya, kode `pw_hook`.
- `feature` – Nama fitur PostgreSQL yang perlu didaftarkan untuk fungsi TLE. Saat ini, satu-satunya fitur yang tersedia adalah hook `passcheck`. Untuk informasi selengkapnya, lihat Hook pemeriksaan kata sandi (passcheck).

Output

Mengembalikan `true` setelah mendaftarkan fitur untuk ekstensi yang ditentukan. Mengembalikan `false` jika fitur sudah terdaftar.

Contoh penggunaan

```
SELECT pgtle.register_feature_if_not_exists('pw_hook', 'passcheck');
```

pgtle.set_default_version

Fungsi `set_default_version` memungkinkan Anda menentukan `default_version` untuk ekstensi TLE Anda. Anda dapat menggunakan fungsi ini untuk menentukan jalur peningkatan dan menetapkan versi sebagai default untuk ekstensi TLE Anda. Ketika pengguna basis data menentukan ekstensi TLE Anda dalam perintah `CREATE EXTENSION` dan `ALTER EXTENSION ... UPDATE`, versi ekstensi TLE Anda ini dibuat dalam basis data untuk pengguna tersebut.

Fungsi ini mengembalikan `true` jika berhasil. Jika ekstensi TLE yang ditentukan dalam argumen `name` tidak ada, fungsi akan mengembalikan pesan kesalahan. Demikian pula, jika `version` ekstensi TLE tidak ada, fungsi akan mengembalikan pesan kesalahan.

Prototipe fungsi

```
pgtle.set_default_version(name text, version text)
```

Peran

`pgtle_admin`

Argumen

- `name` – Nama ekstensi TLE. Nilai ini digunakan saat memanggil `CREATE EXTENSION`.
- `version` – Versi ekstensi TLE untuk mengatur default.

Output

- `true` – Saat pengaturan versi default berhasil, fungsi mengembalikan `true`.
- `ERROR` – Mengembalikan pesan kesalahan jika ekstensi TLE dengan nama atau versi tertentu tidak ada.

Contoh penggunaan

```
SELECT * FROM pgtle.set_default_version('my-extension', '1.1');
```

pgtle.uninstall_extension(nama)

Fungsi `uninstall_extension` menghapus semua versi ekstensi TLE dari basis data. Fungsi ini mencegah panggilan `CREATE EXTENSION` mendatang untuk menginstal ekstensi TLE. Jika ekstensi TLE tidak ada dalam basis data, pesan kesalahan akan muncul.

Fungsi `uninstall_extension` tidak akan menghapus ekstensi TLE yang sedang aktif dalam basis data. Untuk menghapus ekstensi TLE yang sedang aktif, Anda perlu memanggil `DROP EXTENSION` secara eksplisit.

Prototipe fungsi

```
pgtle.uninstall_extension(extname text)
```

Peran

`pgtle_admin`

Argumen

- `extname` – Nama ekstensi TLE yang akan dihapus instalasinya. Nama ini sama dengan yang digunakan `CREATE EXTENSION` untuk memuat ekstensi TLE untuk digunakan dalam basis data tertentu.

Output

Tidak ada.

Contoh penggunaan

```
SELECT * FROM pgtle.uninstall_extension('pg_tle_test');
```

pgtle.uninstall_extension (nama, versi)

Fungsi `uninstall_extension(name, version)` menghapus versi ekstensi TLE tertentu dari basis data. Fungsi ini mencegah `CREATE EXTENSION` dan `ALTER EXTENSION` menginstal atau memperbarui ekstensi TLE ke versi yang ditentukan. Fungsi ini juga menghapus semua jalur pembaruan untuk ekstensi TLE tertentu. Fungsi ini tidak akan menghapus instalasi ekstensi TLE jika ekstensi tersebut saat ini aktif dalam basis data. Anda harus secara eksplisit memanggil `DROP EXTENSION` untuk menghapus ekstensi TLE. Untuk menghapus semua versi ekstensi TLE, lihat pgtle.uninstall_extension(nama).

Prototipe fungsi

```
pgtle.uninstall_extension(extname text, version text)
```

Peran

`pgtle_admin`

Argumen

- `extname` – Nama ekstensi TLE. Nilai ini digunakan saat memanggil `CREATE EXTENSION`.
- `version` – Versi ekstensi TLE yang akan dihapus dari basis data.

Output

Tidak ada.

Contoh penggunaan

```
SELECT * FROM pgtle.uninstall_extension('pg_tle_test', '0.2');
```

pgtle.uninstall_extension_if_exists

Fungsi `uninstall_extension_if_exists` menghapus semua versi ekstensi TLE dari basis data tertentu. Jika ekstensi TLE tidak ada, fungsi akan melakukan pengembalian secara diam-diam (tidak ada pesan kesalahan yang muncul). Jika ekstensi yang ditentukan sedang aktif dalam basis data, fungsi ini tidak akan menghapusnya. Anda harus secara eksplisit memanggil `DROP EXTENSION` untuk menghapus ekstensi TLE sebelum menggunakan fungsi ini untuk menghapus instalasi artefaknya.

Prototipe fungsi

```
pgtle.uninstall_extension_if_exists(extname text)
```

Peran

`pgtle_admin`

Argumen

- `extname` – Nama ekstensi TLE. Nilai ini digunakan saat memanggil `CREATE EXTENSION`.

Output

Fungsi `uninstall_extension_if_exists` mengembalikan `true` setelah menghapus instalasi ekstensi yang ditentukan. Jika ekstensi yang ditentukan tidak ada, fungsi akan mengembalikan `false`.

- `true` – Mengembalikan `true` setelah menghapus instalasi ekstensi TLE.
- `false` – Mengembalikan `false` ketika ekstensi TLE tidak ada dalam basis data.

Contoh penggunaan

```
SELECT * FROM pgtle.uninstall_extension_if_exists('pg_tle_test');
```

pgtle.uninstall_update_path

Fungsi `uninstall_update_path` menghapus jalur pembaruan yang ditentukan dari ekstensi TLE. Fungsi ini membuat `ALTER EXTENSION ... UPDATE TO` tidak dapat menggunakan ekstensi ini sebagai jalur pembaruan.

Jika ekstensi TLE sedang digunakan oleh salah satu versi di jalur pembaruan ini, ekstensi TLE akan tetap ada di basis data.

Jika jalur pembaruan yang ditentukan tidak ada, fungsi akan memunculkan kesalahan.

Prototipe fungsi

```
pgtle.uninstall_update_path(extname text, fromvers text, tovers text)
```

Peran

`pgtle_admin`

Argumen

- `extname` – Nama ekstensi TLE. Nilai ini digunakan saat memanggil `CREATE EXTENSION`.
- `fromvers` – Versi sumber ekstensi TLE yang digunakan pada jalur pembaruan.
- `tovers` – Versi tujuan ekstensi TLE yang digunakan pada jalur pembaruan.

Output

Tidak ada.

Contoh penggunaan

```
SELECT * FROM pgtle.uninstall_update_path('pg_tle_test', '0.1', '0.2');
```

pgtle.uninstall_update_path_if_exists

Fungsi `uninstall_update_path_if_exists` mirip dengan `uninstall_update_path` yang menghapus jalur pembaruan yang ditentukan dari ekstensi TLE. Namun, jika jalur pembaruan tidak ada, fungsi ini tidak memunculkan pesan kesalahan. Sebaliknya, fungsi mengembalikan `false`.

Prototipe fungsi

```
pgtle.uninstall_update_path_if_exists(extname text, fromvers text, tovers text)
```

Peran

`pgtle_admin`

Argumen

- `extname` – Nama ekstensi TLE. Nilai ini digunakan saat memanggil `CREATE EXTENSION`.
- `fromvers` – Versi sumber ekstensi TLE yang digunakan pada jalur pembaruan.
- `tovers` – Versi tujuan ekstensi TLE yang digunakan pada jalur pembaruan.

Output

- `true` – Fungsi telah berhasil memperbarui jalur untuk ekstensi TLE.
- `false` – Fungsi tidak dapat memperbarui jalur untuk ekstensi TLE.

Contoh penggunaan

```
SELECT * FROM pgtle.uninstall_update_path_if_exists('pg_tle_test', '0.1', '0.2');
```

pgtle.unregister_feature

Fungsi `unregister_feature` menyediakan cara untuk menghapus fungsi yang terdaftar untuk menggunakan fitur `pg_tle`, seperti hook. Untuk informasi tentang pendaftaran fitur, lihat pgtle.register_feature.

Prototipe fungsi

```
pgtle.unregister_feature(proc regproc, feature pg_tle_features)
```

Peran

`pgtle_admin`

Argumen

- `proc` – Nama fungsi tersimpan untuk mendaftar dengan fitur `pg_tle`.
- `feature` – Nama fitur `pg_tle` untuk mendaftar dengan fungsi. Misalnya, `passcheck` adalah fitur yang dapat didaftarkan untuk digunakan oleh ekstensi bahasa tepercaya yang Anda kembangkan. Untuk informasi selengkapnya, lihat Hook pemeriksaan kata sandi (passcheck).

Output

Tidak ada.

Contoh penggunaan

```
SELECT * FROM pgtle.unregister_feature('pw_hook', 'passcheck');
```

pgtle.unregister_feature_if_exists

Fungsi `unregister_feature` menyediakan cara untuk menghapus fungsi yang terdaftar untuk menggunakan fitur `pg_tle`, seperti hook. Untuk informasi selengkapnya, lihat Menggunakan hook PostgreSQL dengan ekstensi TLE. Mengembalikan `true` setelah berhasil membatalkan pendaftaran fitur. Mengembalikan `false` jika fitur tidak terdaftar.

Untuk mengetahui informasi tentang pendaftaran fitur `pg_tle` untuk ekstensi TLE Anda, lihat pgtle.register_feature.

Prototipe fungsi

```
pgtle.unregister_feature_if_exists('proc regproc', 'feature pg_tle_features')
```

Peran

`pgtle_admin`

Argumen

- `proc` – Nama fungsi tersimpan yang terdaftar untuk menyertakan fitur `pg_tle`.
- `feature` – Nama fitur `pg_tle` yang terdaftar dengan ekstensi bahasa tepercaya.

Output

Mengembalikan `true` atau `false`, sebagai berikut.

- `true` – Fungsi telah berhasil membatalkan pendaftaran fitur dari ekstensi.
- `false` – Fungsi tidak dapat membatalkan pendaftaran fitur dari ekstensi TLE.

Contoh penggunaan

```
SELECT * FROM pgtle.unregister_feature_if_exists('pw_hook', 'passcheck');
```

Referensi hook untuk Trusted Language Extensions for PostgreSQL

Trusted Language Extensions for PostgreSQL mendukung hook PostgreSQL. Hook adalah mekanisme panggilan balik internal yang tersedia bagi pengembang untuk memperluas fungsionalitas inti PostgreSQL. Dengan hook, pengembang dapat mengimplementasikan fungsi atau prosedurnya sendiri untuk digunakan dalam berbagai operasi basis data, sehingga mengubah perilaku PostgreSQL dalam beberapa cara. Misalnya, Anda dapat menggunakan hook `passcheck` untuk menyesuaikan cara PostgreSQL menangani kata sandi yang diberikan saat membuat atau mengubah kata sandi untuk pengguna (peran).

Lihat dokumentasi berikut untuk mempelajari hook yang tersedia untuk ekstensi TLE Anda.

Topik

- Hook pemeriksaan kata sandi (passcheck)

Hook pemeriksaan kata sandi (passcheck)

Hook `passcheck` digunakan untuk menyesuaikan perilaku PostgreSQL selama proses pemeriksaan kata sandi untuk perintah SQL dan metacommand `psql` berikut.

- `CREATE ROLE` *`username`* `...PASSWORD` – Untuk informasi selengkapnya, lihat CREATE ROLE dalam dokumentasi PostgreSQL.
- `ALTER ROLE` *`username`*`...PASSWORD` – Untuk informasi selengkapnya, lihat ALTER ROLE dalam dokumentasi PostgreSQL.
- `\password` *`username`* – Metacommand `psql` interaktif ini secara aman mengubah kata sandi untuk pengguna yang ditentukan dengan hashing kata sandi sebelum menggunakan sintaks `ALTER ROLE ... PASSWORD` secara transparan. Metacommand adalah pembungkus aman untuk perintah `ALTER ROLE ... PASSWORD`, sehingga hook dapat diterapkan untuk perilaku metacommand `psql`.

Sebagai contoh, lihat Daftar kode hook pemeriksaan kata sandi.

Prototipe fungsi

```
passcheck_hook(username text, password text, password_type pgtle.password_types,
 valid_until timestamptz, valid_null boolean)
```

Argumen

Fungsi hook `passcheck` memiliki argumen berikut.

- `username` – Nama (sebagai teks) dari peran (nama pengguna) yang mengatur kata sandi.
- `password` – Kata sandi yang di-hash atau teks biasa. Kata sandi yang dimasukkan harus sesuai dengan jenis yang ditentukan dalam `password_type`.
- `password_type` – Menentukan format `pgtle.password_type` kata sandi. Format ini dapat berupa salah satu opsi berikut.
  - `PASSWORD_TYPE_PLAINTEXT` – Kata sandi teks biasa.
  - `PASSWORD_TYPE_MD5` – Kata sandi yang telah di-hash menggunakan algoritma MD5 (message digest 5).
  - `PASSWORD_TYPE_SCRAM_SHA_256` – Kata sandi yang telah di-hash menggunakan algoritma SCRAM-SHA-256.
- `valid_until` – Menentukan waktu kapan kata sandi menjadi tidak valid. Argumen ini opsional. Jika Anda menggunakan argumen ini, tentukan waktu sebagai nilai `timestamptz`.
- `valid_null` – Jika Boolean ini diatur ke `true`, opsi `valid_until` diatur ke `NULL`.

Konfigurasi

Fungsi `pgtle.enable_password_check` mengontrol apakah hook passcheck aktif. Hook passcheck memiliki tiga opsi pengaturan.

- `off` – Menonaktifkan hook pemeriksaan kata sandi `passcheck`. Ini adalah nilai default.
- on – Mengaktifkan hook pemeriksaan kata sandi `passcode` sehingga kata sandi diperiksa berdasarkan tabel.
- `require` – Mewajibkan penentuan hook pemeriksaan kata sandi.

Catatan penggunaan

Untuk mengaktifkan atau menonaktifkan hook `passcheck`, Anda perlu memodifikasi grup parameter DB kustom untuk instans DB RDS for PostgreSQL Anda.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-parameter-group \
    --region aws-region \
    --db-parameter-group-name your-custom-parameter-group \
    --parameters
 "ParameterName=pgtle.enable_password_check,ParameterValue=on,ApplyMethod=immediate"
```

Untuk Windows:

```
aws rds modify-db-parameter-group ^
    --region aws-region ^
    --db-parameter-group-name your-custom-parameter-group ^
    --parameters
 "ParameterName=pgtle.enable_password_check,ParameterValue=on,ApplyMethod=immediate"
```

Contoh kode untuk Amazon RDS menggunakan AWS SDK

Contoh kode berikut menunjukkan cara menggunakan Amazon RDS dengan AWS perangkat pengembangan perangkat lunak (SDK).

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Meskipun tindakan menunjukkan cara memanggil setiap fungsi layanan, Anda dapat melihat tindakan dalam konteks pada skenario yang terkait dan contoh lintas layanan.

Skenario adalah contoh kode yang menunjukkan cara untuk menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan yang sama.

Contoh lintas layanan adalah contoh aplikasi yang bekerja di beberapa Layanan AWS.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga berisi informasi tentang cara memulai dan detail tentang versi SDK sebelumnya.

Memulai

Halo Amazon RDS

Contoh kode berikut menunjukkan cara memulai menggunakan Amazon RDS.

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
using System;
using System.Threading.Tasks;
using Amazon.RDS;
using Amazon.RDS.Model;

namespace RDSActions;
```

```
public static class HelloRds
{
    static async Task Main(string[] args)
    {
        var rdsClient = new AmazonRDSClient();

        Console.WriteLine($"Hello Amazon RDS! Following are some of your DB
 instances:");
        Console.WriteLine();

        // You can use await and any of the async methods to get a response.
        // Let's get the first twenty DB instances.
        var response = await rdsClient.DescribeDBInstancesAsync(
            new DescribeDBInstancesRequest()
            {
                MaxRecords = 20 // Must be between 20 and 100.
            });

        foreach (var instance in response.DBInstances)
        {
            Console.WriteLine($"\tDB name: {instance.DBName}");
            Console.WriteLine($"\tArn: {instance.DBInstanceArn}");
            Console.WriteLine($"\tIdentifier: {instance.DBInstanceIdentifier}");
            Console.WriteLine();
        }
    }
}
```

- Lihat detail API di DescribeDBInstances dalam Referensi API AWS SDK for .NET .

C++

SDK for C++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

Kode untuk file CMake MakeLists C.txt.

```
# Set the minimum required version of CMake for this project.
cmake_minimum_required(VERSION 3.13)

# Set the AWS service components used by this project.
set(SERVICE_COMPONENTS rds)

# Set this project's name.
project("hello_rds")

# Set the C++ standard to use to build this target.
# At least C++ 11 is required for the AWS SDK for C++.
set(CMAKE_CXX_STANDARD 11)

# Use the MSVC variable to determine if this is a Windows build.
set(WINDOWS_BUILD ${MSVC})

if (WINDOWS_BUILD) # Set the location where CMake can find the installed
 libraries for the AWS SDK.
    string(REPLACE ";" "/aws-cpp-sdk-all;" SYSTEM_MODULE_PATH
 "${CMAKE_SYSTEM_PREFIX_PATH}/aws-cpp-sdk-all")
    list(APPEND CMAKE_PREFIX_PATH ${SYSTEM_MODULE_PATH})
endif ()

# Find the AWS SDK for C++ package.
find_package(AWSSDK REQUIRED COMPONENTS ${SERVICE_COMPONENTS})

if (WINDOWS_BUILD)
     # Copy relevant AWS SDK for C++ libraries into the current binary directory
 for running and debugging.

     # set(BIN_SUB_DIR "/Debug") # If you are building from the command line, you
 may need to uncomment this
                                     # and set the proper subdirectory to the
 executables' location.

     AWSSDK_CPY_DYN_LIBS(SERVICE_COMPONENTS ""
 ${CMAKE_CURRENT_BINARY_DIR}${BIN_SUB_DIR})
endif ()

add_executable(${PROJECT_NAME}
        hello_rds.cpp)

target_link_libraries(${PROJECT_NAME}
```

```
        ${AWSSDK_LINK_LIBRARIES})
```

Kode untuk file sumber hello_rds.cpp.

```
#include <aws/core/Aws.h>
#include <aws/rds/RDSClient.h>
#include <aws/rds/model/DescribeDBInstancesRequest.h>
#include <iostream>

/*
 *  A "Hello Rds" starter application which initializes an Amazon Relational
 Database Service (Amazon RDS) client and
 *  describes the Amazon RDS instances.
 *
 *  main function
 *
 *  Usage: 'hello_rds'
 *
 */

int main(int argc, char **argv) {
    Aws::SDKOptions options;
    // Optionally change the log level for debugging.
//   options.loggingOptions.logLevel = Utils::Logging::LogLevel::Debug;
    Aws::InitAPI(options); // Should only be called once.
    int result = 0;
    {
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

        Aws::RDS::RDSClient rdsClient(clientConfig);
        Aws::String marker;
        std::vector<Aws::String> instanceDBIDs;

        do {
            Aws::RDS::Model::DescribeDBInstancesRequest request;

            if (!marker.empty()) {
                request.SetMarker(marker);
            }
```

```
            Aws::RDS::Model::DescribeDBInstancesOutcome outcome =
                    rdsClient.DescribeDBInstances(request);

            if (outcome.IsSuccess()) {
                for (auto &instance: outcome.GetResult().GetDBInstances()) {
                    instanceDBIDs.push_back(instance.GetDBInstanceIdentifier());
                }
                marker = outcome.GetResult().GetMarker();
            } else {
                result = 1;
                std::cerr << "Error with RDS::DescribeDBInstances. "
                          << outcome.GetError().GetMessage()
                          << std::endl;
                break;
            }
        } while (!marker.empty());

        std::cout << instanceDBIDs.size() << " RDS instances found." <<
 std::endl;
        for (auto &instanceDBID: instanceDBIDs) {
            std::cout << "   Instance: " << instanceDBID << std::endl;
        }
    }

    Aws::ShutdownAPI(options); // Should only be called once.
    return result;
}
```

- Lihat detail API di DescribeDBInstances dalam Referensi API AWS SDK for C++ .

Go

SDK for 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/rds"
)

// main uses the AWS SDK for Go V2 to create an Amazon Relational Database
 Service (Amazon RDS)
// client and list up to 20 DB instances in your account.
// This example uses the default settings specified in your shared credentials
// and config files.
func main() {
 sdkConfig, err := config.LoadDefaultConfig(context.TODO())
 if err != nil {
  fmt.Println("Couldn't load default configuration. Have you set up your AWS
 account?")
  fmt.Println(err)
  return
 }
 rdsClient := rds.NewFromConfig(sdkConfig)
 const maxInstances = 20
 fmt.Printf("Let's list up to %v DB instances.\n", maxInstances)
 output, err := rdsClient.DescribeDBInstances(context.TODO(),
  &rds.DescribeDBInstancesInput{MaxRecords: aws.Int32(maxInstances)})
 if err != nil {
  fmt.Printf("Couldn't list DB instances: %v\n", err)
  return
 }
 if len(output.DBInstances) == 0 {
  fmt.Println("No DB instances found.")
 } else {
  for _, instance := range output.DBInstances {
   fmt.Printf("DB instance %v has database %v.\n",
 *instance.DBInstanceIdentifier,
    *instance.DBName)
  }
 }
}
```

- Lihat detail API di DescribeDBInstances dalam Referensi API AWS SDK for Go .

Java

SDK for 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.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.DescribeDbInstancesResponse;
import software.amazon.awssdk.services.rds.model.DBInstance;
import software.amazon.awssdk.services.rds.model.RdsException;
import java.util.List;

/**
 * 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 DescribeDBInstances {

    public static void main(String[] args) {
        Region region = Region.US_EAST_1;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();

        describeInstances(rdsClient);
        rdsClient.close();
    }
```

```
    public static void describeInstances(RdsClient rdsClient) {
        try {
            DescribeDbInstancesResponse response =
 rdsClient.describeDBInstances();
            List<DBInstance> instanceList = response.dbInstances();
            for (DBInstance instance : instanceList) {
                System.out.println("Instance ARN is: " +
 instance.dbInstanceArn());
                System.out.println("The Engine is " + instance.engine());
                System.out.println("Connection endpoint is" +
 instance.endpoint().address());
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
}
```

- Lihat detail API di DescribeDBInstances dalam Referensi API AWS SDK for Java 2.x .

Contoh kode

- Tindakan untuk Amazon RDS menggunakan AWS SDK
  - Membuat instans Amazon RDS DB menggunakan SDK AWS
  - Membuat grup parameter Amazon RDS DB menggunakan SDK AWS
  - Membuat snapshot instans Amazon RDS DB menggunakan SDK AWS
  - Membuat token otentikasi untuk autentikasi IAM ke database Amazon RDS menggunakan SDK AWS
  - Menghapus instans Amazon RDS DB menggunakan SDK AWS
  - Menghapus grup parameter Amazon RDS DB menggunakan SDK AWS
  - Jelaskan instans Amazon RDS DB menggunakan SDK AWS
  - Jelaskan grup parameter Amazon RDS DB menggunakan SDK AWS
  - Jelaskan versi mesin database Amazon RDS menggunakan SDK AWS
  - Jelaskan opsi untuk instans Amazon RDS DB menggunakan SDK AWS
  - Jelaskan parameter dalam grup parameter Amazon RDS DB menggunakan SDK AWS

Tindakan untuk Amazon RDS menggunakan AWS SDK

Contoh kode berikut menunjukkan cara melakukan tindakan Amazon RDS individual dengan AWS SDK. Kutipan ini memanggil API Amazon RDS dan merupakan kutipan kode dari program yang lebih besar yang harus dijalankan dalam konteks. Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan instruksi untuk mengatur dan menjalankan kode.

Contoh berikut hanya mencakup tindakan yang paling umum digunakan. Untuk daftar selengkapnya, lihat Referensi API Amazon Relational Database Service (Amazon RDS).

Contoh-contoh

- Membuat instans Amazon RDS DB menggunakan SDK AWS
- Membuat grup parameter Amazon RDS DB menggunakan SDK AWS
- Membuat snapshot instans Amazon RDS DB menggunakan SDK AWS
- Membuat token otentikasi untuk autentikasi IAM ke database Amazon RDS menggunakan SDK AWS
- Menghapus instans Amazon RDS DB menggunakan SDK AWS
- Menghapus grup parameter Amazon RDS DB menggunakan SDK AWS
- Jelaskan instans Amazon RDS DB menggunakan SDK AWS
- Jelaskan grup parameter Amazon RDS DB menggunakan SDK AWS

- Jelaskan versi mesin database Amazon RDS menggunakan SDK AWS
- Jelaskan opsi untuk instans Amazon RDS DB menggunakan SDK AWS
- Jelaskan parameter dalam grup parameter Amazon RDS DB menggunakan SDK AWS
- Jelaskan snapshot instans Amazon RDS DB menggunakan SDK AWS
- Memodifikasi instans Amazon RDS DB menggunakan SDK AWS
- Reboot instans Amazon RDS DB menggunakan SDK AWS
- Mengambil atribut yang dimiliki akun Amazon RDS menggunakan SDK AWS
- Memperbarui parameter dalam grup parameter Amazon RDS DB menggunakan SDK AWS

Membuat instans Amazon RDS DB menggunakan SDK AWS

Contoh-contoh kode berikut menunjukkan cara membuat instans basis data Amazon RDS dan menunggunya sampai tersedia.

Contoh-contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan di dalam konteks. Anda dapat melihat tindakan ini dalam konteks pada contoh kode berikut:

- Memulai instans basis data

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    /// <summary>
    /// Create an RDS DB instance with a particular set of properties. Use the
action DescribeDBInstancesAsync
    /// to determine when the DB instance is ready to use.
    /// </summary>
    /// <param name="dbName">Name for the DB instance.</param>
```

```
   /// <param name="dbInstanceIdentifier">DB instance identifier.</param>
   /// <param name="parameterGroupName">DB parameter group to associate with the
instance.</param>
   /// <param name="dbEngine">The engine for the DB instance.</param>
   /// <param name="dbEngineVersion">Version for the DB instance.</param>
   /// <param name="instanceClass">Class for the DB instance.</param>
   /// <param name="allocatedStorage">The amount of storage in gibibytes (GiB)
to allocate to the DB instance.</param>
   /// <param name="adminName">Admin user name.</param>
   /// <param name="adminPassword">Admin user password.</param>
   /// <returns>DB instance object.</returns>
   public async Task<DBInstance> CreateDBInstance(string dbName, string
dbInstanceIdentifier,
       string parameterGroupName, string dbEngine, string dbEngineVersion,
       string instanceClass, int allocatedStorage, string adminName, string
adminPassword)
   {
       var response = await _amazonRDS.CreateDBInstanceAsync(
           new CreateDBInstanceRequest()
           {
               DBName = dbName,
               DBInstanceIdentifier = dbInstanceIdentifier,
               DBParameterGroupName = parameterGroupName,
               Engine = dbEngine,
               EngineVersion = dbEngineVersion,
               DBInstanceClass = instanceClass,
               AllocatedStorage = allocatedStorage,
               MasterUsername = adminName,
               MasterUserPassword = adminPassword
           });

       return response.DBInstance;
   }
```

- Lihat detail API di CreateDBInstance dalam Referensi API AWS SDK for .NET .

C++

SDK for C++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    Aws::Client::ClientConfiguration clientConfig;
    // Optional: Set to the AWS Region (overrides config file).
    // clientConfig.region = "us-east-1";

Aws::RDS::RDSClient client(clientConfig);

    Aws::RDS::Model::CreateDBInstanceRequest request;
    request.SetDBName(DB_NAME);
    request.SetDBInstanceIdentifier(DB_INSTANCE_IDENTIFIER);
    request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
    request.SetEngine(engineVersion.GetEngine());
    request.SetEngineVersion(engineVersion.GetEngineVersion());
    request.SetDBInstanceClass(dbInstanceClass);
    request.SetStorageType(DB_STORAGE_TYPE);
    request.SetAllocatedStorage(DB_ALLOCATED_STORAGE);
    request.SetMasterUsername(administratorName);
    request.SetMasterUserPassword(administratorPassword);

    Aws::RDS::Model::CreateDBInstanceOutcome outcome =
            client.CreateDBInstance(request);

    if (outcome.IsSuccess()) {
        std::cout << "The DB instance creation has started."
                  << std::endl;
    }
    else {
        std::cerr << "Error with RDS::CreateDBInstance. "
                  << outcome.GetError().GetMessage()
                  << std::endl;
        cleanUpResources(PARAMETER_GROUP_NAME, "", client);
        return false;
    }
```

- Lihat detail API di CreateDBInstance dalam Referensi API AWS SDK for C++ .

CLI

AWS CLI

Untuk membuat instance DB

`create-db-instance`Contoh berikut menggunakan opsi yang diperlukan untuk meluncurkan instans DB baru.

```
aws rds create-db-instance \
    --db-instance-identifier test-mysql-instance \
    --db-instance-class db.t3.micro \
    --engine mysql \
    --master-username admin \
    --master-user-password secret99 \
    --allocated-storage 20
```

Output:

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "test-mysql-instance",
        "DBInstanceClass": "db.t3.micro",
        "Engine": "mysql",
        "DBInstanceStatus": "creating",
        "MasterUsername": "admin",
        "AllocatedStorage": 20,
        "PreferredBackupWindow": "12:55-13:25",
        "BackupRetentionPeriod": 1,
        "DBSecurityGroups": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-12345abc",
                "Status": "active"
            }
        ],
        "DBParameterGroups": [
            {
```

```
                "DBParameterGroupName": "default.mysql5.7",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "DBSubnetGroup": {
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default",
            "VpcId": "vpc-2ff2ff2f",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-########",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2c"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-########",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2d"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-########",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2a"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-########",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2b"
                    },
                    "SubnetStatus": "Active"
                }
            ]
        },
        "PreferredMaintenanceWindow": "sun:08:07-sun:08:37",
        "PendingModifiedValues": {
            "MasterUserPassword": "****"
        },
```

```
        "MultiAZ": false,
        "EngineVersion": "5.7.22",
        "AutoMinorVersionUpgrade": true,
        "ReadReplicaDBInstanceIdentifiers": [],
        "LicenseModel": "general-public-license",
        "OptionGroupMemberships": [
            {
                "OptionGroupName": "default:mysql-5-7",
                "Status": "in-sync"
            }
        ],
        "PubliclyAccessible": true,
        "StorageType": "gp2",
        "DbInstancePort": 0,
        "StorageEncrypted": false,
        "DbiResourceId": "db-5555EXAMPLE44444444EXAMPLE",
        "CACertificateIdentifier": "rds-ca-2019",
        "DomainMemberships": [],
        "CopyTagsToSnapshot": false,
        "MonitoringInterval": 0,
        "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:test-mysql-
instance",
        "IAMDatabaseAuthenticationEnabled": false,
        "PerformanceInsightsEnabled": false,
        "DeletionProtection": false,
        "AssociatedRoles": []
    }
}
```

Untuk informasi selengkapnya, lihat Membuat Instans Amazon RDS DB di Panduan Pengguna Amazon RDS.

- Untuk detail API, lihat createdBInstance di Referensi Perintah.AWS CLI

Go

SDK for Go V2

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// CreateInstance creates a DB instance.
func (instances *DbInstances) CreateInstance(instanceName string, dbName string,
 dbEngine string, dbEngineVersion string, parameterGroupName string,
 dbInstanceClass string,
 storageType string, allocatedStorage int32, adminName string, adminPassword
 string) (
 *types.DBInstance, error) {
 output, err := instances.RdsClient.CreateDBInstance(context.TODO(),
 &rds.CreateDBInstanceInput{
  DBInstanceIdentifier: aws.String(instanceName),
  DBName:               aws.String(dbName),
  DBParameterGroupName: aws.String(parameterGroupName),
  Engine:               aws.String(dbEngine),
  EngineVersion:        aws.String(dbEngineVersion),
  DBInstanceClass:      aws.String(dbInstanceClass),
  StorageType:          aws.String(storageType),
  AllocatedStorage:     aws.Int32(allocatedStorage),
  MasterUsername:       aws.String(adminName),
  MasterUserPassword:   aws.String(adminPassword),
 })
 if err != nil {
  log.Printf("Couldn't create instance %v: %v\n", instanceName, err)
  return nil, err
 } else {
  return output.DBInstance, nil
 }
}
```

- Lihat detail API di CreateDBInstance dalam Referensi API AWS SDK for Go .

Java

SDK for Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
import com.google.gson.Gson;
import
 software.amazon.awssdk.auth.credentials.EnvironmentVariableCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.DescribeDbInstancesRequest;
import software.amazon.awssdk.services.rds.model.CreateDbInstanceRequest;
import software.amazon.awssdk.services.rds.model.CreateDbInstanceResponse;
import software.amazon.awssdk.services.rds.model.RdsException;
import software.amazon.awssdk.services.rds.model.DescribeDbInstancesResponse;
import software.amazon.awssdk.services.rds.model.DBInstance;
import software.amazon.awssdk.services.secretsmanager.SecretsManagerClient;
import
 software.amazon.awssdk.services.secretsmanager.model.GetSecretValueRequest;
import
 software.amazon.awssdk.services.secretsmanager.model.GetSecretValueResponse;

import java.util.List;

/**
 * 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, this example will not work. For more details, see:
 *
```

```
 * https://docs.aws.amazon.com/secretsmanager/latest/userguide/integrating_how-
services-use-secrets_RS.html
 *
 *
 */

public class CreateDBInstance {
    public static long sleepTime = 20;

    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <dbInstanceIdentifier> <dbName> <secretName>

                Where:
                    dbInstanceIdentifier - The database instance identifier.\s
                    dbName - The database name.\s
                    secretName - The name of the AWS Secrets Manager secret that
 contains the database credentials."
                """;

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }

        String dbInstanceIdentifier = args[0];
        String dbName = args[1];
        String secretName = args[2];
        Gson gson = new Gson();
        User user = gson.fromJson(String.valueOf(getSecretValues(secretName)),
 User.class);
        Region region = Region.US_WEST_2;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();

        createDatabaseInstance(rdsClient, dbInstanceIdentifier, dbName,
 user.getUsername(), user.getPassword());
        waitForInstanceReady(rdsClient, dbInstanceIdentifier);
        rdsClient.close();
    }
```

```
   private static SecretsManagerClient getSecretClient() {
       Region region = Region.US_WEST_2;
       return SecretsManagerClient.builder()
               .region(region)

.credentialsProvider(EnvironmentVariableCredentialsProvider.create())
               .build();
   }

   private static String getSecretValues(String secretName) {
       SecretsManagerClient secretClient = getSecretClient();
       GetSecretValueRequest valueRequest = GetSecretValueRequest.builder()
               .secretId(secretName)
               .build();

       GetSecretValueResponse valueResponse =
secretClient.getSecretValue(valueRequest);
       return valueResponse.secretString();
   }

   public static void createDatabaseInstance(RdsClient rdsClient,
           String dbInstanceIdentifier,
           String dbName,
           String userName,
           String userPassword) {

       try {
           CreateDbInstanceRequest instanceRequest =
CreateDbInstanceRequest.builder()
                   .dbInstanceIdentifier(dbInstanceIdentifier)
                   .allocatedStorage(100)
                   .dbName(dbName)
                   .engine("mysql")
                   .dbInstanceClass("db.m4.large")
                   .engineVersion("8.0")
                   .storageType("standard")
                   .masterUsername(userName)
                   .masterUserPassword(userPassword)
                   .build();

           CreateDbInstanceResponse response =
rdsClient.createDBInstance(instanceRequest);
           System.out.print("The status is " +
response.dbInstance().dbInstanceStatus());
```

```
        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    // Waits until the database instance is available.
    public static void waitForInstanceReady(RdsClient rdsClient, String
dbInstanceIdentifier) {
        boolean instanceReady = false;
        String instanceReadyStr;
        System.out.println("Waiting for instance to become available.");
        try {
            DescribeDbInstancesRequest instanceRequest =
DescribeDbInstancesRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .build();

            // Loop until the cluster is ready.
            while (!instanceReady) {
                DescribeDbInstancesResponse response =
rdsClient.describeDBInstances(instanceRequest);
                List<DBInstance> instanceList = response.dbInstances();
                for (DBInstance instance : instanceList) {
                    instanceReadyStr = instance.dbInstanceStatus();
                    if (instanceReadyStr.contains("available"))
                        instanceReady = true;
                    else {
                        System.out.print(".");
                        Thread.sleep(sleepTime * 1000);
                    }
                }
            }
            System.out.println("Database instance is available!");

        } catch (RdsException | InterruptedException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Lihat detail API di CreateDBInstance dalam Referensi API AWS SDK for Java 2.x .

Kotlin

SDK for Kotlin

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
suspend fun createDatabaseInstance(
    dbInstanceIdentifierVal: String?,
    dbNamedbVal: String?,
    masterUsernameVal: String?,
    masterUserPasswordVal: String?
) {
    val instanceRequest = CreateDbInstanceRequest {
        dbInstanceIdentifier = dbInstanceIdentifierVal
        allocatedStorage = 100
        dbName = dbNamedbVal
        engine = "mysql"
        dbInstanceClass = "db.m4.large"
        engineVersion = "8.0"
        storageType = "standard"
        masterUsername = masterUsernameVal
        masterUserPassword = masterUserPasswordVal
    }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.createDbInstance(instanceRequest)
        print("The status is ${response.dbInstance?.dbInstanceStatus}")
    }
}

// Waits until the database instance is available.
suspend fun waitForInstanceReady(dbInstanceIdentifierVal: String?) {
    val sleepTime: Long = 20
    var instanceReady = false
    var instanceReadyStr = ""
    println("Waiting for instance to become available.")

    val instanceRequest = DescribeDbInstancesRequest {
```

```
        dbInstanceIdentifier = dbInstanceIdentifierVal
    }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        while (!instanceReady) {
            val response = rdsClient.describeDbInstances(instanceRequest)
            val instanceList = response.dbInstances
            if (instanceList != null) {
                for (instance in instanceList) {
                    instanceReadyStr = instance.dbInstanceStatus.toString()
                    if (instanceReadyStr.contains("available")) {
                        instanceReady = true
                    } else {
                        println("...$instanceReadyStr")
                        delay(sleepTime * 1000)
                    }
                }
            }
        }
        println("Database instance is available!")
    }
}
```

- Lihat detail API di CreateDBInstance dalam Referensi API AWS SDK for Kotlin.

PHP

SDK for PHP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
require __DIR__ . '/vendor/autoload.php';

use Aws\Exception\AwsException;
```

```
$rdsClient = new Aws\Rds\RdsClient([
    'region' => 'us-east-2'
]);

$dbIdentifier = '<<{{db-identifier}}>>';
$dbClass = 'db.t2.micro';
$storage = 5;
$engine = 'MySQL';
$username = 'MyUser';
$password = 'MyPassword';

try {
    $result = $rdsClient->createDBInstance([
        'DBInstanceIdentifier' => $dbIdentifier,
        'DBInstanceClass' => $dbClass,
        'AllocatedStorage' => $storage,
        'Engine' => $engine,
        'MasterUsername' => $username,
        'MasterUserPassword' => $password,
    ]);
    var_dump($result);
} catch (AwsException $e) {
    echo $e->getMessage();
    echo "\n";
}
```

- Lihat detail API di CreateDBInstance dalam Referensi API AWS SDK for PHP .

Python

SDK for Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def create_db_instance(
        self,
        db_name,
        instance_id,
        parameter_group_name,
        db_engine,
        db_engine_version,
        instance_class,
        storage_type,
        allocated_storage,
        admin_name,
        admin_password,
    ):
        """
        Creates a DB instance.

        :param db_name: The name of the database that is created in the DB
 instance.
        :param instance_id: The ID to give the newly created DB instance.
        :param parameter_group_name: A parameter group to associate with the DB
 instance.
        :param db_engine: The database engine of a database to create in the DB
 instance.
        :param db_engine_version: The engine version for the created database.
```

```
        :param instance_class: The DB instance class for the newly created DB
instance.
        :param storage_type: The storage type of the DB instance.
        :param allocated_storage: The amount of storage allocated on the DB
instance, in GiBs.
        :param admin_name: The name of the admin user for the created database.
        :param admin_password: The admin password for the created database.
        :return: Data about the newly created DB instance.
        """
        try:
            response = self.rds_client.create_db_instance(
                DBName=db_name,
                DBInstanceIdentifier=instance_id,
                DBParameterGroupName=parameter_group_name,
                Engine=db_engine,
                EngineVersion=db_engine_version,
                DBInstanceClass=instance_class,
                StorageType=storage_type,
                AllocatedStorage=allocated_storage,
                MasterUsername=admin_name,
                MasterUserPassword=admin_password,
            )
            db_inst = response["DBInstance"]
        except ClientError as err:
            logger.error(
                "Couldn't create DB instance %s. Here's why: %s: %s",
                instance_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return db_inst
```

- Lihat detail API di CreateDBInstance dalam Referensi API AWS SDK for Python (Boto3).

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang cara memulai dan detail versi-versi SDK sebelumnya.

Membuat grup parameter Amazon RDS DB menggunakan SDK AWS

Contoh-contoh kode berikut menunjukkan cara membuat grup parameter basis data Amazon RDS.

Contoh-contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan di dalam konteks. Anda dapat melihat tindakan ini dalam konteks pada contoh kode berikut:

- Memulai instans basis data

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   /// <summary>
   /// Create a new DB parameter group. Use the action
DescribeDBParameterGroupsAsync
   /// to determine when the DB parameter group is ready to use.
   /// </summary>
   /// <param name="name">Name of the DB parameter group.</param>
   /// <param name="family">Family of the DB parameter group.</param>
   /// <param name="description">Description of the DB parameter group.</param>
   /// <returns>The new DB parameter group.</returns>
   public async Task<DBParameterGroup> CreateDBParameterGroup(
       string name, string family, string description)
   {
       var response = await _amazonRDS.CreateDBParameterGroupAsync(
           new CreateDBParameterGroupRequest()
           {
               DBParameterGroupName = name,
               DBParameterGroupFamily = family,
               Description = description
           });
       return response.DBParameterGroup;
   }
```

- Untuk detail API, lihat CreateDB ParameterGroup di AWS SDK for .NET Referensi API.

C++

SDK for C++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    Aws::Client::ClientConfiguration clientConfig;
    // Optional: Set to the AWS Region (overrides config file).
    // clientConfig.region = "us-east-1";

Aws::RDS::RDSClient client(clientConfig);

    Aws::RDS::Model::CreateDBParameterGroupRequest request;
    request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
    request.SetDBParameterGroupFamily(dbParameterGroupFamily);
    request.SetDescription("Example parameter group.");

    Aws::RDS::Model::CreateDBParameterGroupOutcome outcome =
            client.CreateDBParameterGroup(request);

    if (outcome.IsSuccess()) {
        std::cout << "The DB parameter group was successfully created."
                  << std::endl;
    }
    else {
        std::cerr << "Error with RDS::CreateDBParameterGroup. "
                  << outcome.GetError().GetMessage()
                  << std::endl;
        return false;
    }
```

- Untuk detail API, lihat CreateDB ParameterGroup di AWS SDK for C++ Referensi API.

CLI

AWS CLI

Untuk membuat grup parameter DB

`create-db-parameter-group`Contoh berikut membuat grup parameter DB.

```
aws rds create-db-parameter-group \
    --db-parameter-group-name mydbparametergroup \
    --db-parameter-group-family MySQL5.6 \
    --description "My new parameter group"
```

Output:

```
{
    "DBParameterGroup": {
        "DBParameterGroupName": "mydbparametergroup",
        "DBParameterGroupFamily": "mysql5.6",
        "Description": "My new parameter group",
        "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:mydbparametergroup"
    }
}
```

Untuk informasi selengkapnya, lihat Membuat Grup Parameter DB di Panduan Pengguna Amazon RDS.

- Untuk detail API, lihat CreateDB ParameterGroup di AWS CLI Referensi Perintah.

Go

SDK for Go V2

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// CreateParameterGroup creates a DB parameter group that is based on the
 specified
// parameter group family.
func (instances *DbInstances) CreateParameterGroup(
 parameterGroupName string, parameterGroupFamily string, description string) (
 *types.DBParameterGroup, error) {

 output, err := instances.RdsClient.CreateDBParameterGroup(context.TODO(),
  &rds.CreateDBParameterGroupInput{
   DBParameterGroupName:   aws.String(parameterGroupName),
   DBParameterGroupFamily: aws.String(parameterGroupFamily),
   Description:            aws.String(description),
  })
 if err != nil {
  log.Printf("Couldn't create parameter group %v: %v\n", parameterGroupName, err)
  return nil, err
 } else {
  return output.DBParameterGroup, err
 }
}
```

- Untuk detail API, lihat CreateDB ParameterGroup di AWS SDK for Go Referensi API.

Java

SDK for Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   public static void createDBParameterGroup(RdsClient rdsClient, String
dbGroupName, String dbParameterGroupFamily) {
       try {
           CreateDbParameterGroupRequest groupRequest =
CreateDbParameterGroupRequest.builder()
                   .dbParameterGroupName(dbGroupName)
                   .dbParameterGroupFamily(dbParameterGroupFamily)
                   .description("Created by using the AWS SDK for Java")
                   .build();

           CreateDbParameterGroupResponse response =
rdsClient.createDBParameterGroup(groupRequest);
           System.out.println("The group name is " +
response.dbParameterGroup().dbParameterGroupName());

       } catch (RdsException e) {
           System.out.println(e.getLocalizedMessage());
           System.exit(1);
       }
   }
```

- Untuk detail API, lihat CreateDB ParameterGroup di AWS SDK for Java 2.x Referensi API.

Python

SDK for Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
```

```
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def create_parameter_group(
        self, parameter_group_name, parameter_group_family, description
    ):
        """
        Creates a DB parameter group that is based on the specified parameter
group
        family.

        :param parameter_group_name: The name of the newly created parameter
group.
        :param parameter_group_family: The family that is used as the basis of
the new
                                       parameter group.
        :param description: A description given to the parameter group.
        :return: Data about the newly created parameter group.
        """
        try:
            response = self.rds_client.create_db_parameter_group(
                DBParameterGroupName=parameter_group_name,
                DBParameterGroupFamily=parameter_group_family,
                Description=description,
            )
        except ClientError as err:
            logger.error(
                "Couldn't create parameter group %s. Here's why: %s: %s",
                parameter_group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return response
```

- Untuk detail API, lihat CreateDB ParameterGroup di AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang cara memulai dan detail versi-versi SDK sebelumnya.

Membuat snapshot instans Amazon RDS DB menggunakan SDK AWS

Contoh-contoh kode berikut menunjukkan cara membuat cuplikan instans basis data Amazon RDS.

Contoh-contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan di dalam konteks. Anda dapat melihat tindakan ini dalam konteks pada contoh kode berikut:

- Memulai instans basis data

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   /// <summary>
   /// Create a snapshot of a DB instance.
   /// </summary>
   /// <param name="dbInstanceIdentifier">DB instance identifier.</param>
   /// <param name="snapshotIdentifier">Identifier for the snapshot.</param>
   /// <returns>DB snapshot object.</returns>
   public async Task<DBSnapshot> CreateDBSnapshot(string dbInstanceIdentifier,
string snapshotIdentifier)
   {
       var response = await _amazonRDS.CreateDBSnapshotAsync(
```

```
            new CreateDBSnapshotRequest()
            {
                DBSnapshotIdentifier = snapshotIdentifier,
                DBInstanceIdentifier = dbInstanceIdentifier
            });

        return response.DBSnapshot;
    }
```

- Lihat detail API di CreateDBSnapshot dalam Referensi API AWS SDK for .NET .

C++

SDK for C++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);

            Aws::RDS::Model::CreateDBSnapshotRequest request;
            request.SetDBInstanceIdentifier(DB_INSTANCE_IDENTIFIER);
            request.SetDBSnapshotIdentifier(snapshotID);

            Aws::RDS::Model::CreateDBSnapshotOutcome outcome =
                    client.CreateDBSnapshot(request);

            if (outcome.IsSuccess()) {
                std::cout << "Snapshot creation has started."
                          << std::endl;
            }
            else {
                std::cerr << "Error with RDS::CreateDBSnapshot. "
```

```
                        << outcome.GetError().GetMessage()
                        << std::endl;
                cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
 client);
                return false;
            }
```

- Lihat detail API di CreateDBSnapshot dalam Referensi API AWS SDK for C++ .

CLI

AWS CLI

Untuk membuat snapshot DB

`create-db-snapshot`Contoh berikut membuat snapshot DB.

```
aws rds create-db-snapshot \
    --db-instance-identifier database-mysql \
    --db-snapshot-identifier mydbsnapshot
```

Output:

```
{
    "DBSnapshot": {
        "DBSnapshotIdentifier": "mydbsnapshot",
        "DBInstanceIdentifier": "database-mysql",
        "Engine": "mysql",
        "AllocatedStorage": 100,
        "Status": "creating",
        "Port": 3306,
        "AvailabilityZone": "us-east-1b",
        "VpcId": "vpc-6594f31c",
        "InstanceCreateTime": "2019-04-30T15:45:53.663Z",
        "MasterUsername": "admin",
        "EngineVersion": "5.6.40",
        "LicenseModel": "general-public-license",
        "SnapshotType": "manual",
        "Iops": 1000,
        "OptionGroupName": "default:mysql-5-6",
        "PercentProgress": 0,
```

```
        "StorageType": "io1",
        "Encrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/
AKIAIOSFODNN7EXAMPLE",
        "DBSnapshotArn": "arn:aws:rds:us-
east-1:123456789012:snapshot:mydbsnapshot",
        "IAMDatabaseAuthenticationEnabled": false,
        "ProcessorFeatures": [],
        "DbiResourceId": "db-AKIAIOSFODNN7EXAMPLE"
    }
}
```

Untuk informasi selengkapnya, lihat Membuat Snapshot DB di Panduan Pengguna Amazon RDS.

- Untuk detail API, lihat CreatedBSnapshot di Referensi Perintah.AWS CLI

Go

SDK for Go V2

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// CreateSnapshot creates a snapshot of a DB instance.
func (instances *DbInstances) CreateSnapshot(instanceName string, snapshotName
 string) (
 *types.DBSnapshot, error) {
 output, err := instances.RdsClient.CreateDBSnapshot(context.TODO(),
 &rds.CreateDBSnapshotInput{
  DBInstanceIdentifier: aws.String(instanceName),
  DBSnapshotIdentifier: aws.String(snapshotName),
```

```
 })
 if err != nil {
  log.Printf("Couldn't create snapshot %v: %v\n", snapshotName, err)
  return nil, err
 } else {
  return output.DBSnapshot, nil
 }
}
```

- Lihat detail API di CreateDBSnapshot dalam Referensi API AWS SDK for Go .

Java

SDK for Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   // Create an Amazon RDS snapshot.
   public static void createSnapshot(RdsClient rdsClient, String
dbInstanceIdentifier, String dbSnapshotIdentifier) {
       try {
           CreateDbSnapshotRequest snapshotRequest =
CreateDbSnapshotRequest.builder()
                   .dbInstanceIdentifier(dbInstanceIdentifier)
                   .dbSnapshotIdentifier(dbSnapshotIdentifier)
                   .build();

           CreateDbSnapshotResponse response =
rdsClient.createDBSnapshot(snapshotRequest);
           System.out.println("The Snapshot id is " +
response.dbSnapshot().dbiResourceId());

       } catch (RdsException e) {
           System.out.println(e.getLocalizedMessage());
           System.exit(1);
       }
```

```
    }
```

- Lihat detail API di CreateDBSnapshot dalam Referensi API AWS SDK for Java 2.x .

PHP

SDK for PHP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
require __DIR__ . '/vendor/autoload.php';

use Aws\Exception\AwsException;



$rdsClient = new Aws\Rds\RdsClient([
    'region' => 'us-east-2'
]);

$dbIdentifier = '<<{{db-identifier}}>>';
$snapshotName = '<<{{backup_2018_12_25}}>>';

try {
    $result = $rdsClient->createDBSnapshot([
        'DBInstanceIdentifier' => $dbIdentifier,
        'DBSnapshotIdentifier' => $snapshotName,
    ]);
    var_dump($result);
} catch (AwsException $e) {
    echo $e->getMessage();
    echo "\n";
}
```

- Lihat detail API di CreateDBSnapshot dalam Referensi API AWS SDK for PHP .

Python

SDK for Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def create_snapshot(self, snapshot_id, instance_id):
        """
        Creates a snapshot of a DB instance.

        :param snapshot_id: The ID to give the created snapshot.
        :param instance_id: The ID of the DB instance to snapshot.
        :return: Data about the newly created snapshot.
        """
        try:
            response = self.rds_client.create_db_snapshot(
                DBSnapshotIdentifier=snapshot_id,
 DBInstanceIdentifier=instance_id
```

```
            )
            snapshot = response["DBSnapshot"]
        except ClientError as err:
            logger.error(
                "Couldn't create snapshot of %s. Here's why: %s: %s",
                instance_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return snapshot
```

- Lihat detail API di CreateDBSnapshot dalam Referensi API AWS SDK for Python (Boto3).

Ruby

SDK for Ruby

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
require "aws-sdk-rds"  # v2: require 'aws-sdk'

# Create a snapshot for an Amazon Relational Database Service (Amazon RDS)
# DB instance.
#
# @param rds_resource [Aws::RDS::Resource] The resource containing SDK logic.
# @param db_instance_name [String] The name of the Amazon RDS DB instance.
# @return [Aws::RDS::DBSnapshot, nil] The snapshot created, or nil if error.
def create_snapshot(rds_resource, db_instance_name)
  id = "snapshot-#{rand(10**6)}"
  db_instance = rds_resource.db_instance(db_instance_name)
  db_instance.create_snapshot({
                                db_snapshot_identifier: id
                              })
rescue Aws::Errors::ServiceError => e
```

```
  puts "Couldn't create DB instance snapshot #{id}:\n #{e.message}"
end
```

- Lihat detail API di CreateDBSnapshot dalam Referensi API AWS SDK for Ruby .

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang cara memulai dan detail versi-versi SDK sebelumnya.

Membuat token otentikasi untuk autentikasi IAM ke database Amazon RDS menggunakan SDK AWS

Contoh kode berikut menunjukkan cara membuat token autentikasi untuk autentikasi IAM.

Java

SDK for Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

Gunakan RdsUtilitieskelas untuk menghasilkan token otentikasi.

```
public class GenerateRDSAuthToken {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <dbInstanceIdentifier> <masterUsername>

                Where:
                    dbInstanceIdentifier - The database instance identifier.\s
                    masterUsername - The master user name.\s
                """;

        if (args.length != 2) {
```

```
            System.out.println(usage);
            System.exit(1);
        }

        String dbInstanceIdentifier = args[0];
        String masterUsername = args[1];
        Region region = Region.US_WEST_2;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();

        String token = getAuthToken(rdsClient, dbInstanceIdentifier,
 masterUsername);
        System.out.println("The token response is " + token);
    }

    public static String getAuthToken(RdsClient rdsClient, String
 dbInstanceIdentifier, String masterUsername) {

        RdsUtilities utilities = rdsClient.utilities();
        try {
            GenerateAuthenticationTokenRequest tokenRequest =
 GenerateAuthenticationTokenRequest.builder()
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .username(masterUsername)
                    .port(3306)
                    .hostname(dbInstanceIdentifier)
                    .build();

            return utilities.generateAuthenticationToken(tokenRequest);

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
        return "";
    }
}
```

- Untuk detail API, lihat GeneraTerds AuthToken di Referensi AWS SDK for Java 2.x API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang cara memulai dan detail versi-versi SDK sebelumnya.

Menghapus instans Amazon RDS DB menggunakan SDK AWS

Contoh-contoh kode berikut menunjukkan cara menghapus instans basis data Amazon RDS.

Contoh-contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan di dalam konteks. Anda dapat melihat tindakan ini dalam konteks pada contoh kode berikut:

- Memulai instans basis data

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
/// <summary>
/// Delete a particular DB instance.
/// </summary>
/// <param name="dbInstanceIdentifier">DB instance identifier.</param>
/// <returns>DB instance object.</returns>
public async Task<DBInstance> DeleteDBInstance(string dbInstanceIdentifier)
{
    var response = await _amazonRDS.DeleteDBInstanceAsync(
        new DeleteDBInstanceRequest()
        {
            DBInstanceIdentifier = dbInstanceIdentifier,
            SkipFinalSnapshot = true,
            DeleteAutomatedBackups = true
        });

    return response.DBInstance;
}
```

- Lihat detail API di DeleteDBInstance dalam Referensi API AWS SDK for .NET .

C++

SDK for C++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    Aws::Client::ClientConfiguration clientConfig;
    // Optional: Set to the AWS Region (overrides config file).
    // clientConfig.region = "us-east-1";

Aws::RDS::RDSClient client(clientConfig);

        Aws::RDS::Model::DeleteDBInstanceRequest request;
        request.SetDBInstanceIdentifier(dbInstanceIdentifier);
        request.SetSkipFinalSnapshot(true);
        request.SetDeleteAutomatedBackups(true);

        Aws::RDS::Model::DeleteDBInstanceOutcome outcome =
                client.DeleteDBInstance(request);

        if (outcome.IsSuccess()) {
            std::cout << "DB instance deletion has started."
                      << std::endl;
        }
        else {
            std::cerr << "Error with RDS::DeleteDBInstance. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            result = false;
        }
```

- Lihat detail API di DeleteDBInstance dalam Referensi API AWS SDK for C++ .

CLI

AWS CLI

Untuk menghapus instans DB

`delete-db-instance`Contoh berikut menghapus instance DB tertentu setelah membuat snapshot DB akhir bernama. `test-instance-final-snap`

```
aws rds delete-db-instance \
    --db-instance-identifier test-instance \
    --final-db-snapshot-identifier test-instance-final-snap
```

Output:

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "test-instance",
        "DBInstanceStatus": "deleting",
        ...some output truncated...
    }
}
```

- Untuk detail API, lihat DeletedBInstance di Referensi Perintah AWS CLI .

Go

SDK for Go V2

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
```

```
}



// DeleteInstance deletes a DB instance.
func (instances *DbInstances) DeleteInstance(instanceName string) error {
 _, err := instances.RdsClient.DeleteDBInstance(context.TODO(),
 &rds.DeleteDBInstanceInput{
  DBInstanceIdentifier:   aws.String(instanceName),
  SkipFinalSnapshot:      true,
  DeleteAutomatedBackups: aws.Bool(true),
 })
 if err != nil {
  log.Printf("Couldn't delete instance %v: %v\n", instanceName, err)
  return err
 } else {
  return nil
 }
}
```

- Lihat detail API di DeleteDBInstance dalam Referensi API AWS SDK for Go .

Java

SDK for 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.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.DeleteDbInstanceRequest;
import software.amazon.awssdk.services.rds.model.DeleteDbInstanceResponse;
import software.amazon.awssdk.services.rds.model.RdsException;

/**
 * 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 DeleteDBInstance {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <dbInstanceIdentifier>\s

                Where:
                    dbInstanceIdentifier - The database instance identifier\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String dbInstanceIdentifier = args[0];
        Region region = Region.US_WEST_2;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();

        deleteDatabaseInstance(rdsClient, dbInstanceIdentifier);
        rdsClient.close();
    }

    public static void deleteDatabaseInstance(RdsClient rdsClient, String
 dbInstanceIdentifier) {
        try {
            DeleteDbInstanceRequest deleteDbInstanceRequest =
 DeleteDbInstanceRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .deleteAutomatedBackups(true)
                    .skipFinalSnapshot(true)
                    .build();
```

```
            DeleteDbInstanceResponse response =
 rdsClient.deleteDBInstance(deleteDbInstanceRequest);
            System.out.print("The status of the database is " +
 response.dbInstance().dbInstanceStatus());

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
}
```

- Lihat detail API di DeleteDBInstance dalam Referensi API AWS SDK for Java 2.x .

Kotlin

SDK for Kotlin

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
suspend fun deleteDatabaseInstance(dbInstanceIdentifierVal: String?) {

    val deleteDbInstanceRequest = DeleteDbInstanceRequest {
        dbInstanceIdentifier = dbInstanceIdentifierVal
        deleteAutomatedBackups = true
        skipFinalSnapshot = true
    }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.deleteDbInstance(deleteDbInstanceRequest)
        print("The status of the database is
 ${response.dbInstance?.dbInstanceStatus}")
    }
}
```

- Lihat detail API di DeleteDBInstance dalam Referensi API AWS SDK for Kotlin.

PHP

SDK for PHP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
require __DIR__ . '/vendor/autoload.php';

use Aws\Exception\AwsException;


//Create an RDSClient
$rdsClient = new Aws\Rds\RdsClient([
    'region' => 'us-east-1'
]);

$dbIdentifier = '<<{{db-identifier}}>>';

try {
    $result = $rdsClient->deleteDBInstance([
        'DBInstanceIdentifier' => $dbIdentifier,
    ]);
    var_dump($result);
} catch (AwsException $e) {
    echo $e->getMessage();
    echo "\n";
}
```

- Lihat detail API di DeleteDBInstance dalam Referensi API AWS SDK for PHP .

Python

SDK for Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def delete_db_instance(self, instance_id):
        """
        Deletes a DB instance.

        :param instance_id: The ID of the DB instance to delete.
        :return: Data about the deleted DB instance.
        """
        try:
            response = self.rds_client.delete_db_instance(
                DBInstanceIdentifier=instance_id,
                SkipFinalSnapshot=True,
                DeleteAutomatedBackups=True,
            )
            db_inst = response["DBInstance"]
```

```
        except ClientError as err:
            logger.error(
                "Couldn't delete DB instance %s. Here's why: %s: %s",
                instance_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return db_inst
```

- Lihat detail API di DeleteDBInstance dalam Referensi API AWS SDK for Python (Boto3).

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang cara memulai dan detail versi-versi SDK sebelumnya.

Menghapus grup parameter Amazon RDS DB menggunakan SDK AWS

Contoh-contoh kode berikut menunjukkan cara menghapus grup parameter basis data Amazon RDS.

Contoh-contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan di dalam konteks. Anda dapat melihat tindakan ini dalam konteks pada contoh kode berikut:

- Memulai instans basis data

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    /// <summary>
```

```
    /// Delete a DB parameter group. The group cannot be a default DB parameter
group
    /// or be associated with any DB instances.
    /// </summary>
    /// <param name="name">Name of the DB parameter group.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteDBParameterGroup(string name)
    {
        var response = await _amazonRDS.DeleteDBParameterGroupAsync(
            new DeleteDBParameterGroupRequest()
            {
                DBParameterGroupName = name,
            });
        return response.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Untuk detail API, lihat DeleteDB ParameterGroup di Referensi AWS SDK for .NET API.

C++

SDK for C++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);

        Aws::RDS::Model::DeleteDBParameterGroupRequest request;
        request.SetDBParameterGroupName(parameterGroupName);

        Aws::RDS::Model::DeleteDBParameterGroupOutcome outcome =
                client.DeleteDBParameterGroup(request);
```

```
        if (outcome.IsSuccess()) {
            std::cout << "The DB parameter group was successfully deleted."
                      << std::endl;
        }
        else {
            std::cerr << "Error with RDS::DeleteDBParameterGroup. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            result = false;
        }
```

- Untuk detail API, lihat DeleteDB ParameterGroup di Referensi AWS SDK for C++ API.

CLI

AWS CLI

Untuk menghapus grup parameter DB

commmandContoh berikut menghapus grup parameter DB.

```
aws rds delete-db-parameter-group \
    --db-parameter-group-name mydbparametergroup
```

Perintah ini tidak menghasilkan output.

Untuk informasi selengkapnya, lihat Bekerja dengan Grup Parameter DB di Panduan Pengguna Amazon RDS.

- Untuk detail API, lihat DeleteDB ParameterGroup di Referensi AWS CLI Perintah.

Go

SDK for Go V2

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// DeleteParameterGroup deletes the named DB parameter group.
func (instances *DbInstances) DeleteParameterGroup(parameterGroupName string)
 error {
 _, err := instances.RdsClient.DeleteDBParameterGroup(context.TODO(),
  &rds.DeleteDBParameterGroupInput{
   DBParameterGroupName: aws.String(parameterGroupName),
  })
 if err != nil {
  log.Printf("Couldn't delete parameter group %v: %v\n", parameterGroupName, err)
  return err
 } else {
  return nil
 }
}
```

- Untuk detail API, lihat DeleteDB ParameterGroup di Referensi AWS SDK for Go API.

Java

SDK for Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    // Delete the parameter group after database has been deleted.
    // An exception is thrown if you attempt to delete the para group while
 database
    // exists.
```

```
   public static void deleteParaGroup(RdsClient rdsClient, String dbGroupName,
String dbARN)
           throws InterruptedException {
       try {
           boolean isDataDel = false;
           boolean didFind;
           String instanceARN;

           // Make sure that the database has been deleted.
           while (!isDataDel) {
               DescribeDbInstancesResponse response =
rdsClient.describeDBInstances();
               List<DBInstance> instanceList = response.dbInstances();
               int listSize = instanceList.size();
               didFind = false;
               int index = 1;
               for (DBInstance instance : instanceList) {
                   instanceARN = instance.dbInstanceArn();
                   if (instanceARN.compareTo(dbARN) == 0) {
                       System.out.println(dbARN + " still exists");
                       didFind = true;
                   }
                   if ((index == listSize) && (!didFind)) {
                       // Went through the entire list and did not find the
database ARN.
                       isDataDel = true;
                   }
                   Thread.sleep(sleepTime * 1000);
                   index++;
               }
           }

           // Delete the para group.
           DeleteDbParameterGroupRequest parameterGroupRequest =
DeleteDbParameterGroupRequest.builder()
                   .dbParameterGroupName(dbGroupName)
                   .build();

           rdsClient.deleteDBParameterGroup(parameterGroupRequest);
           System.out.println(dbGroupName + " was deleted.");

       } catch (RdsException e) {
           System.out.println(e.getLocalizedMessage());
           System.exit(1);
```

```
        }
    }
```

- Untuk detail API, lihat DeleteDB ParameterGroup di Referensi AWS SDK for Java 2.x API.

Python

SDK for Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def delete_parameter_group(self, parameter_group_name):
        """
        Deletes a DB parameter group.

        :param parameter_group_name: The name of the parameter group to delete.
        :return: Data about the parameter group.
        """
        try:
```

```
            self.rds_client.delete_db_parameter_group(
                DBParameterGroupName=parameter_group_name
            )
        except ClientError as err:
            logger.error(
                "Couldn't delete parameter group %s. Here's why: %s: %s",
                parameter_group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Untuk detail API, lihat DeleteDB ParameterGroup di AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang cara memulai dan detail versi-versi SDK sebelumnya.

Jelaskan instans Amazon RDS DB menggunakan SDK AWS

Contoh-contoh kode berikut menunjukkan cara menjelaskan instans basis data Amazon RDS.

Contoh-contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan di dalam konteks. Anda dapat melihat tindakan ini dalam konteks pada contoh kode berikut:

- Memulai instans basis data

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    /// <summary>
    /// Returns a list of DB instances.
    /// </summary>
    /// <param name="dbInstanceIdentifier">Optional name of a specific DB
instance.</param>
    /// <returns>List of DB instances.</returns>
    public async Task<List<DBInstance>> DescribeDBInstances(string
dbInstanceIdentifier = null)
    {
        var results = new List<DBInstance>();
        var instancesPaginator = _amazonRDS.Paginators.DescribeDBInstances(
            new DescribeDBInstancesRequest
            {
                DBInstanceIdentifier = dbInstanceIdentifier
            });
        // Get the entire list using the paginator.
        await foreach (var instances in instancesPaginator.DBInstances)
        {
            results.Add(instances);
        }
        return results;
    }
```

- Lihat detail API di DescribeDBInstances dalam Referensi API AWS SDK for .NET .

C++

SDK for C++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";
```

```
    Aws::RDS::RDSClient client(clientConfig);


//! Routine which gets a DB instance description.
/*!
 \sa describeDBInstance()
 \param dbInstanceIdentifier: A DB instance identifier.
 \param instanceResult: The 'DBInstance' object containing the description.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::describeDBInstance(const Aws::String &dbInstanceIdentifier,
                                     Aws::RDS::Model::DBInstance &instanceResult,
                                     const Aws::RDS::RDSClient &client) {
    Aws::RDS::Model::DescribeDBInstancesRequest request;
    request.SetDBInstanceIdentifier(dbInstanceIdentifier);

    Aws::RDS::Model::DescribeDBInstancesOutcome outcome =
            client.DescribeDBInstances(request);

    bool result = true;
    if (outcome.IsSuccess()) {
        instanceResult = outcome.GetResult().GetDBInstances()[0];
    }
    else if (outcome.GetError().GetErrorType() !=
             Aws::RDS::RDSErrors::D_B_INSTANCE_NOT_FOUND_FAULT) {
        result = false;
        std::cerr << "Error with RDS::DescribeDBInstances. "
                  << outcome.GetError().GetMessage()
                  << std::endl;
    }
        // This example does not log an error if the DB instance does not exist.
        // Instead, instanceResult is set to empty.
    else {
        instanceResult = Aws::RDS::Model::DBInstance();
    }

    return result;
}
```

- Lihat detail API di DescribeDBInstances dalam Referensi API AWS SDK for C++ .

CLI

AWS CLI

Untuk menggambarkan instance DB

`describe-db-instances`Contoh berikut mengambil rincian tentang instans DB tertentu.

```
aws rds describe-db-instances \
    --db-instance-identifier mydbinstancecf
```

Output:

```
{
    "DBInstances": [
        {
            "DBInstanceIdentifier": "mydbinstancecf",
            "DBInstanceClass": "db.t3.small",
            "Engine": "mysql",
            "DBInstanceStatus": "available",
            "MasterUsername": "masterawsuser",
            "Endpoint": {
                "Address": "mydbinstancecf.abcexample.us-
east-1.rds.amazonaws.com",
                "Port": 3306,
                "HostedZoneId": "Z2R2ITUGPM61AM"
            },
            ...some output truncated...
        }
    ]
}
```

- Untuk detail API, lihat DescribedBInstances di Referensi Perintah.AWS CLI

Go

SDK for Go V2

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// GetInstance gets data about a DB instance.
func (instances *DbInstances) GetInstance(instanceName string) (
 *types.DBInstance, error) {
 output, err := instances.RdsClient.DescribeDBInstances(context.TODO(),
  &rds.DescribeDBInstancesInput{
   DBInstanceIdentifier: aws.String(instanceName),
  })
 if err != nil {
  var notFoundError *types.DBInstanceNotFoundFault
  if errors.As(err, &notFoundError) {
   log.Printf("DB instance %v does not exist.\n", instanceName)
   err = nil
  } else {
   log.Printf("Couldn't get instance %v: %v\n", instanceName, err)
  }
  return nil, err
 } else {
  return &output.DBInstances[0], nil
 }
}
```

- Lihat detail API di DescribeDBInstances dalam Referensi API AWS SDK for Go .

Java

SDK for 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.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.DescribeDbInstancesResponse;
import software.amazon.awssdk.services.rds.model.DBInstance;
import software.amazon.awssdk.services.rds.model.RdsException;
import java.util.List;

/**
 * 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 DescribeDBInstances {

    public static void main(String[] args) {
        Region region = Region.US_EAST_1;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();

        describeInstances(rdsClient);
        rdsClient.close();
    }

    public static void describeInstances(RdsClient rdsClient) {
        try {
            DescribeDbInstancesResponse response =
 rdsClient.describeDBInstances();
```

```
            List<DBInstance> instanceList = response.dbInstances();
            for (DBInstance instance : instanceList) {
                System.out.println("Instance ARN is: " +
 instance.dbInstanceArn());
                System.out.println("The Engine is " + instance.engine());
                System.out.println("Connection endpoint is" +
 instance.endpoint().address());
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
}
```

- Lihat detail API di DescribeDBInstances dalam Referensi API AWS SDK for Java 2.x .

Kotlin

SDK for Kotlin

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
suspend fun describeInstances() {

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.describeDbInstances(DescribeDbInstancesRequest
 {})
        response.dbInstances?.forEach { instance ->
            println("Instance Identifier is ${instance.dbInstanceIdentifier}")
            println("The Engine is ${instance.engine}")
            println("Connection endpoint is ${instance.endpoint?.address}")
        }
    }
}
```

- Lihat detail API di DescribeDBInstances dalam Referensi API AWS SDK for Kotlin.

PHP

SDK for PHP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
require __DIR__ . '/vendor/autoload.php';

use Aws\Exception\AwsException;


//Create an RDSClient
$rdsClient = new Aws\Rds\RdsClient([
    'region' => 'us-east-2'
]);

try {
    $result = $rdsClient->describeDBInstances();
    foreach ($result['DBInstances'] as $instance) {
        print('<p>DB Identifier: ' . $instance['DBInstanceIdentifier']);
        print('<br />Endpoint: ' . $instance['Endpoint']["Address"]
            . ':' . $instance['Endpoint']["Port"]);
        print('<br />Current Status: ' . $instance["DBInstanceStatus"]);
        print('</p>');
    }
    print(" Raw Result ");
    var_dump($result);
} catch (AwsException $e) {
    echo $e->getMessage();
    echo "\n";
}
```

- Lihat detail API di DescribeDBInstances dalam Referensi API AWS SDK for PHP .

Python

SDK for Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def get_db_instance(self, instance_id):
        """
        Gets data about a DB instance.

        :param instance_id: The ID of the DB instance to retrieve.
        :return: The retrieved DB instance.
        """
        try:
            response = self.rds_client.describe_db_instances(
                DBInstanceIdentifier=instance_id
```

```
            )
            db_inst = response["DBInstances"][0]
        except ClientError as err:
            if err.response["Error"]["Code"] == "DBInstanceNotFound":
                logger.info("Instance %s does not exist.", instance_id)
            else:
                logger.error(
                    "Couldn't get DB instance %s. Here's why: %s: %s",
                    instance_id,
                    err.response["Error"]["Code"],
                    err.response["Error"]["Message"],
                )
                raise
        else:
            return db_inst
```

- Lihat detail API di DescribeDBInstances dalam Referensi API AWS SDK for Python (Boto3).

Ruby

SDK for Ruby

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
require "aws-sdk-rds"  # v2: require 'aws-sdk'

# List all Amazon Relational Database Service (Amazon RDS) DB instances.
#
# @param rds_resource [Aws::RDS::Resource] An SDK for Ruby Amazon RDS resource.
# @return [Array, nil] List of all DB instances, or nil if error.
def list_instances(rds_resource)
  db_instances = []
  rds_resource.db_instances.each do |i|
    db_instances.append({
                          "name": i.id,
                          "status": i.db_instance_status
```

```
                        })
  end
  db_instances
rescue Aws::Errors::ServiceError => e
  puts "Couldn't list instances:\n#{e.message}"
end
```

- Lihat detail API di DescribeDBInstances dalam Referensi API AWS SDK for Ruby .

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang cara memulai dan detail versi-versi SDK sebelumnya.

Jelaskan grup parameter Amazon RDS DB menggunakan SDK AWS

Contoh-contoh kode berikut menunjukkan cara menjelaskan grup parameter basis data Amazon RDS.

Contoh-contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan di dalam konteks. Anda dapat melihat tindakan ini dalam konteks pada contoh kode berikut:

- Memulai instans basis data

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    /// <summary>
    /// Get descriptions of DB parameter groups.
    /// </summary>
    /// <param name="name">Optional name of the DB parameter group to describe.</
param>
```

```
    /// <returns>The list of DB parameter group descriptions.</returns>
    public async Task<List<DBParameterGroup>> DescribeDBParameterGroups(string
name = null)
    {
        var response = await _amazonRDS.DescribeDBParameterGroupsAsync(
            new DescribeDBParameterGroupsRequest()
            {
                DBParameterGroupName = name
            });
        return response.DBParameterGroups;
    }
```

- Untuk detail API, lihat DescribeDB ParameterGroups di Referensi AWS SDK for .NET API.

C++

SDK for C++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);

        Aws::RDS::Model::DescribeDBParameterGroupsRequest request;
        request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);

        Aws::RDS::Model::DescribeDBParameterGroupsOutcome outcome =
                client.DescribeDBParameterGroups(request);

        if (outcome.IsSuccess()) {
            std::cout << "DB parameter group named '" <<
                      PARAMETER_GROUP_NAME << "' already exists." << std::endl;
```

```
            dbParameterGroupFamily = outcome.GetResult().GetDBParameterGroups()
[0].GetDBParameterGroupFamily();
        }

        else {
            std::cerr << "Error with RDS::DescribeDBParameterGroups. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
```

- Untuk detail API, lihat DescribeDB ParameterGroups di Referensi AWS SDK for C++ API.

CLI

AWS CLI

Untuk menggambarkan grup parameter DB Anda

`describe-db-parameter-groups`Contoh berikut mengambil rincian tentang grup parameter DB Anda.

```
aws rds describe-db-parameter-groups
```

Output:

```
{
    "DBParameterGroups": [
        {
            "DBParameterGroupName": "default.aurora-mysql5.7",
            "DBParameterGroupFamily": "aurora-mysql5.7",
            "Description": "Default parameter group for aurora-mysql5.7",
            "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.aurora-mysql5.7"
        },
        {
            "DBParameterGroupName": "default.aurora-postgresql9.6",
            "DBParameterGroupFamily": "aurora-postgresql9.6",
            "Description": "Default parameter group for aurora-postgresql9.6",
            "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.aurora-postgresql9.6"
```

```
        },
        {
            "DBParameterGroupName": "default.aurora5.6",
            "DBParameterGroupFamily": "aurora5.6",
            "Description": "Default parameter group for aurora5.6",
            "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.aurora5.6"
        },
        {
            "DBParameterGroupName": "default.mariadb10.1",
            "DBParameterGroupFamily": "mariadb10.1",
            "Description": "Default parameter group for mariadb10.1",
            "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.mariadb10.1"
        },
        ...some output truncated...
    ]
}
```

Untuk informasi selengkapnya, lihat Bekerja dengan Grup Parameter DB di Panduan Pengguna Amazon RDS.

- Untuk detail API, lihat DescribeDB ParameterGroups di Referensi AWS CLI Perintah.

Go

SDK for Go V2

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// GetParameterGroup gets a DB parameter group by name.
```

```
func (instances *DbInstances) GetParameterGroup(parameterGroupName string) (
 *types.DBParameterGroup, error) {
 output, err := instances.RdsClient.DescribeDBParameterGroups(
  context.TODO(), &rds.DescribeDBParameterGroupsInput{
   DBParameterGroupName: aws.String(parameterGroupName),
  })
 if err != nil {
  var notFoundError *types.DBParameterGroupNotFoundFault
  if errors.As(err, &notFoundError) {
   log.Printf("Parameter group %v does not exist.\n", parameterGroupName)
   err = nil
  } else {
   log.Printf("Error getting parameter group %v: %v\n", parameterGroupName, err)
  }
  return nil, err
 } else {
  return &output.DBParameterGroups[0], err
 }
}
```

- Untuk detail API, lihat DescribeDB ParameterGroups di Referensi AWS SDK for Go API.

Java

SDK for Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   public static void describeDbParameterGroups(RdsClient rdsClient, String
dbGroupName) {
       try {
           DescribeDbParameterGroupsRequest groupsRequest =
DescribeDbParameterGroupsRequest.builder()
                   .dbParameterGroupName(dbGroupName)
                   .maxRecords(20)
                   .build();
```

```
            DescribeDbParameterGroupsResponse response =
 rdsClient.describeDBParameterGroups(groupsRequest);
            List<DBParameterGroup> groups = response.dbParameterGroups();
            for (DBParameterGroup group : groups) {
                System.out.println("The group name is " +
 group.dbParameterGroupName());
                System.out.println("The group description is " +
 group.description());
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
```

- Untuk detail API, lihat DescribeDB ParameterGroups di Referensi AWS SDK for Java 2.x API.

Python

SDK for Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
```

```
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def get_parameter_group(self, parameter_group_name):
        """
        Gets a DB parameter group.

        :param parameter_group_name: The name of the parameter group to retrieve.
        :return: The parameter group.
        """
        try:
            response = self.rds_client.describe_db_parameter_groups(
                DBParameterGroupName=parameter_group_name
            )
            parameter_group = response["DBParameterGroups"][0]
        except ClientError as err:
            if err.response["Error"]["Code"] == "DBParameterGroupNotFound":
                logger.info("Parameter group %s does not exist.",
parameter_group_name)
            else:
                logger.error(
                    "Couldn't get parameter group %s. Here's why: %s: %s",
                    parameter_group_name,
                    err.response["Error"]["Code"],
                    err.response["Error"]["Message"],
                )
                raise
        else:
            return parameter_group
```

- Untuk detail API, lihat DescribeDB ParameterGroups di AWS SDK for Python (Boto3) Referensi API.

Ruby

SDK for Ruby

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
require "aws-sdk-rds"  # v2: require 'aws-sdk'

# List all Amazon Relational Database Service (Amazon RDS) parameter groups.
#
# @param rds_resource [Aws::RDS::Resource] An SDK for Ruby Amazon RDS resource.
# @return [Array, nil] List of all parameter groups, or nil if error.
def list_parameter_groups(rds_resource)
  parameter_groups = []
  rds_resource.db_parameter_groups.each do |p|
    parameter_groups.append({
                              "name": p.db_parameter_group_name,
                              "description": p.description
                            })
  end
  parameter_groups
rescue Aws::Errors::ServiceError => e
  puts "Couldn't list parameter groups:\n #{e.message}"
end
```

- Untuk detail API, lihat DescribeDB ParameterGroups di Referensi AWS SDK for Ruby API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang cara memulai dan detail versi-versi SDK sebelumnya.

Jelaskan versi mesin database Amazon RDS menggunakan SDK AWS

Contoh-contoh kode berikut menunjukkan cara mendeskripsikan versi mesin basis data Amazon.

Contoh-contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan di dalam konteks. Anda dapat melihat tindakan ini dalam konteks pada contoh kode berikut:

- Memulai instans basis data

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   /// <summary>
   /// Get a list of DB engine versions for a particular DB engine.
   /// </summary>
   /// <param name="engine">Name of the engine.</param>
   /// <param name="dbParameterGroupFamily">Optional parameter group family
name.</param>
   /// <returns>List of DBEngineVersions.</returns>
   public async Task<List<DBEngineVersion>> DescribeDBEngineVersions(string
engine,
       string dbParameterGroupFamily = null)
   {
       var response = await _amazonRDS.DescribeDBEngineVersionsAsync(
           new DescribeDBEngineVersionsRequest()
           {
               Engine = engine,
               DBParameterGroupFamily = dbParameterGroupFamily
           });
       return response.DBEngineVersions;
   }
```

- Untuk detail API, lihat DescribeDB EngineVersions di Referensi AWS SDK for .NET API.

C++

SDK for C++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);


//! Routine which gets available DB engine versions for an engine name and
//! an optional parameter group family.
/*!
 \sa getDBEngineVersions()
 \param engineName: A DB engine name.
 \param parameterGroupFamily: A parameter group family name, ignored if empty.
 \param engineVersionsResult: Vector of 'DBEngineVersion' objects returned by the
 routine.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::getDBEngineVersions(const Aws::String &engineName,
                                      const Aws::String &parameterGroupFamily,

 Aws::Vector<Aws::RDS::Model::DBEngineVersion> &engineVersionsResult,
                                      const Aws::RDS::RDSClient &client) {
    Aws::RDS::Model::DescribeDBEngineVersionsRequest request;
    request.SetEngine(engineName);
    if (!parameterGroupFamily.empty()) {
        request.SetDBParameterGroupFamily(parameterGroupFamily);
    }

    Aws::RDS::Model::DescribeDBEngineVersionsOutcome outcome =
            client.DescribeDBEngineVersions(request);
```

```
    if (outcome.IsSuccess()) {
        engineVersionsResult = outcome.GetResult().GetDBEngineVersions();
    }
    else {
        std::cerr << "Error with RDS::DescribeDBEngineVersionsRequest. "
                  << outcome.GetError().GetMessage()
                  << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Untuk detail API, lihat DescribeDB EngineVersions di Referensi AWS SDK for C++ API.

CLI

AWS CLI

Untuk menggambarkan versi mesin DB untuk mesin MySQL DB

`describe-db-engine-versions`Contoh berikut menampilkan rincian tentang masing-masing versi mesin DB untuk mesin DB yang ditentukan.

```
aws rds describe-db-engine-versions \
    --engine mysql
```

Output:

```
{
    "DBEngineVersions": [
        {
            "Engine": "mysql",
            "EngineVersion": "5.5.46",
            "DBParameterGroupFamily": "mysql5.5",
            "DBEngineDescription": "MySQL Community Edition",
            "DBEngineVersionDescription": "MySQL 5.5.46",
            "ValidUpgradeTarget": [
                {
                    "Engine": "mysql",
                    "EngineVersion": "5.5.53",
                    "Description": "MySQL 5.5.53",
```

```
                "AutoUpgrade": false,
                "IsMajorVersionUpgrade": false
            },
            {
                "Engine": "mysql",
                "EngineVersion": "5.5.54",
                "Description": "MySQL 5.5.54",
                "AutoUpgrade": false,
                "IsMajorVersionUpgrade": false
            },
            {
                "Engine": "mysql",
                "EngineVersion": "5.5.57",
                "Description": "MySQL 5.5.57",
                "AutoUpgrade": false,
                "IsMajorVersionUpgrade": false
            },
            ...some output truncated...
        ]
    }
```

Untuk informasi selengkapnya, lihat Apa itu Amazon Relational Database Service (Amazon RDS)? di Panduan Pengguna Amazon RDS.

- Untuk detail API, lihat DescribeDB EngineVersions di Referensi AWS CLI Perintah.

Go

SDK for Go V2

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}
```

```
// GetEngineVersions gets database engine versions that are available for the
 specified engine
// and parameter group family.
func (instances *DbInstances) GetEngineVersions(engine string,
 parameterGroupFamily string) (
 []types.DBEngineVersion, error) {
 output, err := instances.RdsClient.DescribeDBEngineVersions(context.TODO(),
  &rds.DescribeDBEngineVersionsInput{
   Engine:                 aws.String(engine),
   DBParameterGroupFamily: aws.String(parameterGroupFamily),
  })
 if err != nil {
  log.Printf("Couldn't get engine versions for %v: %v\n", engine, err)
  return nil, err
 } else {
  return output.DBEngineVersions, nil
 }
}
```

- Untuk detail API, lihat DescribeDB EngineVersions di Referensi AWS SDK for Go API.

Java

SDK for Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    public static void describeDBEngines(RdsClient rdsClient) {
        try {
            DescribeDbEngineVersionsRequest engineVersionsRequest =
 DescribeDbEngineVersionsRequest.builder()
                    .defaultOnly(true)
                    .engine("mysql")
                    .maxRecords(20)
                    .build();
```

```
            DescribeDbEngineVersionsResponse response =
rdsClient.describeDBEngineVersions(engineVersionsRequest);
            List<DBEngineVersion> engines = response.dbEngineVersions();

            // Get all DBEngineVersion objects.
            for (DBEngineVersion engineOb : engines) {
                System.out.println("The name of the DB parameter group family for
the database engine is "
                        + engineOb.dbParameterGroupFamily());
                System.out.println("The name of the database engine " +
engineOb.engine());
                System.out.println("The version number of the database engine " +
engineOb.engineVersion());
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
```

- Untuk detail API, lihat DescribeDB EngineVersions di Referensi AWS SDK for Java 2.x API.

Python

SDK for Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
```

```
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def get_engine_versions(self, engine, parameter_group_family=None):
        """
        Gets database engine versions that are available for the specified engine
        and parameter group family.

        :param engine: The database engine to look up.
        :param parameter_group_family: When specified, restricts the returned
list of
                                       engine versions to those that are
compatible with
                                       this parameter group family.
        :return: The list of database engine versions.
        """
        try:
            kwargs = {"Engine": engine}
            if parameter_group_family is not None:
                kwargs["DBParameterGroupFamily"] = parameter_group_family
            response = self.rds_client.describe_db_engine_versions(**kwargs)
            versions = response["DBEngineVersions"]
        except ClientError as err:
            logger.error(
                "Couldn't get engine versions for %s. Here's why: %s: %s",
                engine,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return versions
```

- Untuk detail API, lihat DescribeDB EngineVersions di AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang cara memulai dan detail versi-versi SDK sebelumnya.

Jelaskan opsi untuk instans Amazon RDS DB menggunakan SDK AWS

Contoh-contoh kode berikut menunjukkan cara menjelaskan opsi untuk instans basis data Amazon RDS.

Contoh-contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan di dalam konteks. Anda dapat melihat tindakan ini dalam konteks pada contoh kode berikut:

- Memulai instans basis data

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    /// <summary>
    /// Get a list of orderable DB instance options for a specific
    /// engine and engine version.
    /// </summary>
    /// <param name="engine">Name of the engine.</param>
    /// <param name="engineVersion">Version of the engine.</param>
    /// <returns>List of OrderableDBInstanceOptions.</returns>
    public async Task<List<OrderableDBInstanceOption>>
DescribeOrderableDBInstanceOptions(string engine, string engineVersion)
    {
        // Use a paginator to get a list of DB instance options.
        var results = new List<OrderableDBInstanceOption>();
```

```
        var paginateInstanceOptions =
_amazonRDS.Paginators.DescribeOrderableDBInstanceOptions(
            new DescribeOrderableDBInstanceOptionsRequest()
            {
                Engine = engine,
                EngineVersion = engineVersion,
            });
        // Get the entire list using the paginator.
        await foreach (var instanceOptions in
paginateInstanceOptions.OrderableDBInstanceOptions)
        {
            results.Add(instanceOptions);
        }
        return results;
    }
```

- Untuk detail API, lihat DescribeOrderableDB InstanceOptions di Referensi AWS SDK for .NET API.

C++

SDK for C++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);


//! Routine which gets available 'micro' DB instance classes, displays the list
//! to the user, and returns the user selection.
/*!
```

```
 \sa chooseMicroDBInstanceClass()
 \param engineName: The DB engine name.
 \param engineVersion: The DB engine version.
 \param dbInstanceClass: String for DB instance class chosen by the user.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::chooseMicroDBInstanceClass(const Aws::String &engine,
                                             const Aws::String &engineVersion,
                                             Aws::String &dbInstanceClass,
                                             const Aws::RDS::RDSClient &client) {
   std::vector<Aws::String> instanceClasses;
   Aws::String marker;
   do {
       Aws::RDS::Model::DescribeOrderableDBInstanceOptionsRequest request;
       request.SetEngine(engine);
       request.SetEngineVersion(engineVersion);
       if (!marker.empty()) {
           request.SetMarker(marker);
       }

       Aws::RDS::Model::DescribeOrderableDBInstanceOptionsOutcome outcome =
               client.DescribeOrderableDBInstanceOptions(request);

       if (outcome.IsSuccess()) {
           const Aws::Vector<Aws::RDS::Model::OrderableDBInstanceOption>
 &options =
                   outcome.GetResult().GetOrderableDBInstanceOptions();
           for (const Aws::RDS::Model::OrderableDBInstanceOption &option:
 options) {
               const Aws::String &instanceClass = option.GetDBInstanceClass();
               if (instanceClass.find("micro") != std::string::npos) {
                   if (std::find(instanceClasses.begin(), instanceClasses.end(),
                                 instanceClass) ==
                       instanceClasses.end()) {
                       instanceClasses.push_back(instanceClass);
                   }
               }
           }
           marker = outcome.GetResult().GetMarker();
       }
       else {
           std::cerr << "Error with RDS::DescribeOrderableDBInstanceOptions. "
                     << outcome.GetError().GetMessage()
```

```
                        << std::endl;
            return false;
        }
    } while (!marker.empty());

    std::cout << "The available micro DB instance classes for your database
 engine are:"
              << std::endl;
    for (int i = 0; i < instanceClasses.size(); ++i) {
        std::cout << "   " << i + 1 << ": " << instanceClasses[i] << std::endl;
    }

    int choice = askQuestionForIntRange(
            "Which micro DB instance class do you want to use? ",
            1, static_cast<int>(instanceClasses.size()));
    dbInstanceClass = instanceClasses[choice - 1];
    return true;
}
```

- Untuk detail API, lihat DescribeOrderableDB InstanceOptions di Referensi AWS SDK for C++ API.

CLI

AWS CLI

Untuk menjelaskan opsi instans DB yang dapat dipesan

`describe-orderable-db-instance-options`Contoh berikut mengambil rincian tentang opsi yang dapat dipesan untuk instance DB yang menjalankan mesin MySQL DB.

```
aws rds describe-orderable-db-instance-options \
    --engine mysql
```

Output:

```
{
    "OrderableDBInstanceOptions": [
        {
            "MinStorageSize": 5,
```

```
            "ReadReplicaCapable": true,
            "MaxStorageSize": 6144,
            "AvailabilityZones": [
                {
                    "Name": "us-east-1a"
                },
                {
                    "Name": "us-east-1b"
                },
                {
                    "Name": "us-east-1c"
                },
                {
                    "Name": "us-east-1d"
                }
            ],
            "SupportsIops": false,
            "AvailableProcessorFeatures": [],
            "MultiAZCapable": true,
            "DBInstanceClass": "db.m1.large",
            "Vpc": true,
            "StorageType": "gp2",
            "LicenseModel": "general-public-license",
            "EngineVersion": "5.5.46",
            "SupportsStorageEncryption": false,
            "SupportsEnhancedMonitoring": true,
            "Engine": "mysql",
            "SupportsIAMDatabaseAuthentication": false,
            "SupportsPerformanceInsights": false
        }
    ]
    ...some output truncated...
}
```

- Untuk detail API, lihat DescribeOrderableDB InstanceOptions di Referensi AWS CLI Perintah.

Go

SDK for Go V2

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// GetOrderableInstances uses a paginator to get DB instance options that can be
 used to create DB instances that are
// compatible with a set of specifications.
func (instances *DbInstances) GetOrderableInstances(engine string, engineVersion
 string) (
 []types.OrderableDBInstanceOption, error) {

 var output *rds.DescribeOrderableDBInstanceOptionsOutput
 var instanceOptions []types.OrderableDBInstanceOption
 var err error
 orderablePaginator :=
 rds.NewDescribeOrderableDBInstanceOptionsPaginator(instances.RdsClient,
  &rds.DescribeOrderableDBInstanceOptionsInput{
   Engine:        aws.String(engine),
   EngineVersion: aws.String(engineVersion),
  })
 for orderablePaginator.HasMorePages() {
  output, err = orderablePaginator.NextPage(context.TODO())
  if err != nil {
   log.Printf("Couldn't get orderable DB instance options: %v\n", err)
   break
  } else {
   instanceOptions = append(instanceOptions,
 output.OrderableDBInstanceOptions...)
  }
```

```
 }
 return instanceOptions, err
}
```

- Untuk detail API, lihat DescribeOrderableDB InstanceOptions di Referensi AWS SDK for Go API.

Java

SDK for Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   // Get a list of allowed engine versions.
   public static void getAllowedEngines(RdsClient rdsClient, String
dbParameterGroupFamily) {
       try {
           DescribeDbEngineVersionsRequest versionsRequest =
DescribeDbEngineVersionsRequest.builder()
                   .dbParameterGroupFamily(dbParameterGroupFamily)
                   .engine("mysql")
                   .build();

           DescribeDbEngineVersionsResponse response =
rdsClient.describeDBEngineVersions(versionsRequest);
           List<DBEngineVersion> dbEngines = response.dbEngineVersions();
           for (DBEngineVersion dbEngine : dbEngines) {
               System.out.println("The engine version is " +
dbEngine.engineVersion());
               System.out.println("The engine description is " +
dbEngine.dbEngineDescription());
           }

       } catch (RdsException e) {
           System.out.println(e.getLocalizedMessage());
```

```
            System.exit(1);
        }
    }
```

- Untuk detail API, lihat DescribeOrderableDB InstanceOptions di Referensi AWS SDK for Java 2.x API.

Python

SDK for Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def get_orderable_instances(self, db_engine, db_engine_version):
        """
        Gets DB instance options that can be used to create DB instances that are
        compatible with a set of specifications.
```

```
        :param db_engine: The database engine that must be supported by the DB
instance.
        :param db_engine_version: The engine version that must be supported by
the DB instance.
        :return: The list of DB instance options that can be used to create a
compatible DB instance.
        """
        try:
            inst_opts = []
            paginator = self.rds_client.get_paginator(
                "describe_orderable_db_instance_options"
            )
            for page in paginator.paginate(
                Engine=db_engine, EngineVersion=db_engine_version
            ):
                inst_opts += page["OrderableDBInstanceOptions"]
        except ClientError as err:
            logger.error(
                "Couldn't get orderable DB instances. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return inst_opts
```

- Untuk detail API, lihat DescribeOrderableDB InstanceOptions di AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang cara memulai dan detail versi-versi SDK sebelumnya.

Jelaskan parameter dalam grup parameter Amazon RDS DB menggunakan SDK AWS

Contoh-contoh kode berikut menunjukkan cara menjelaskan parameter dalam grup parameter basis data Amazon RDS.

Contoh-contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan di dalam konteks. Anda dapat melihat tindakan ini dalam konteks pada contoh kode berikut:

- Memulai instans basis data

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   /// <summary>
   /// Get a list of DB parameters from a specific parameter group.
   /// </summary>
   /// <param name="dbParameterGroupName">Name of a specific DB parameter
group.</param>
   /// <param name="source">Optional source for selecting parameters.</param>
   /// <returns>List of parameter values.</returns>
   public async Task<List<Parameter>> DescribeDBParameters(string
dbParameterGroupName, string source = null)
   {
       var results = new List<Parameter>();
       var paginateParameters = _amazonRDS.Paginators.DescribeDBParameters(
           new DescribeDBParametersRequest()
           {
               DBParameterGroupName = dbParameterGroupName,
               Source = source
           });
       // Get the entire list using the paginator.
       await foreach (var parameters in paginateParameters.Parameters)
       {
           results.Add(parameters);
       }
       return results;
   }
```

- Lihat detail API di DescribeDBParameters dalam Referensi API AWS SDK for .NET .

C++

SDK for C++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);


//! Routine which gets DB parameters using the 'DescribeDBParameters' api.
/*!
 \sa getDBParameters()
 \param parameterGroupName: The name of the parameter group.
 \param namePrefix: Prefix string to filter results by parameter name.
 \param source: A source such as 'user', ignored if empty.
 \param parametersResult: Vector of 'Parameter' objects returned by the routine.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::getDBParameters(const Aws::String &parameterGroupName,
                                  const Aws::String &namePrefix,
                                  const Aws::String &source,
                                  Aws::Vector<Aws::RDS::Model::Parameter>
 &parametersResult,
                                  const Aws::RDS::RDSClient &client) {
    Aws::String marker;
    do {
        Aws::RDS::Model::DescribeDBParametersRequest request;
        request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
        if (!marker.empty()) {
```

```
            request.SetMarker(marker);
        }
        if (!source.empty()) {
            request.SetSource(source);
        }

        Aws::RDS::Model::DescribeDBParametersOutcome outcome =
                client.DescribeDBParameters(request);

        if (outcome.IsSuccess()) {
            const Aws::Vector<Aws::RDS::Model::Parameter> &parameters =
                    outcome.GetResult().GetParameters();
            for (const Aws::RDS::Model::Parameter &parameter: parameters) {
                if (!namePrefix.empty()) {
                    if (parameter.GetParameterName().find(namePrefix) == 0) {
                        parametersResult.push_back(parameter);
                    }
                }
                else {
                    parametersResult.push_back(parameter);
                }
            }

            marker = outcome.GetResult().GetMarker();
        }
        else {
            std::cerr << "Error with RDS::DescribeDBParameters. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
    } while (!marker.empty());

    return true;
}
```

- Lihat detail API di DescribeDBParameters dalam Referensi API AWS SDK for C++ .

CLI

AWS CLI

Untuk menggambarkan parameter dalam kelompok parameter DB

`describe-db-parameters`Contoh berikut mengambil rincian kelompok parameter DB yang ditentukan.

```
aws rds describe-db-parameters \
    --db-parameter-group-name mydbpg
```

Output:

```
{
    "Parameters": [
        {
            "ParameterName": "allow-suspicious-udfs",
            "Description": "Controls whether user-defined functions that have
 only an xxx symbol for the main function can be loaded",
            "Source": "engine-default",
            "ApplyType": "static",
            "DataType": "boolean",
            "AllowedValues": "0,1",
            "IsModifiable": false,
            "ApplyMethod": "pending-reboot"
        },
        {
            "ParameterName": "auto_generate_certs",
            "Description": "Controls whether the server autogenerates SSL key and
 certificate files in the data directory, if they do not already exist.",
            "Source": "engine-default",
            "ApplyType": "static",
            "DataType": "boolean",
            "AllowedValues": "0,1",
            "IsModifiable": false,
            "ApplyMethod": "pending-reboot"
        },
        ...some output truncated...
    ]
}
```

Untuk informasi selengkapnya, lihat Bekerja dengan Grup Parameter DB di Panduan Pengguna Amazon RDS.

- Untuk detail API, lihat AWS CLI DescribedBParameters di Referensi Perintah.

Go

SDK for Go V2

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// GetParameters gets the parameters that are contained in a DB parameter group.
func (instances *DbInstances) GetParameters(parameterGroupName string, source
 string) (
 []types.Parameter, error) {

 var output *rds.DescribeDBParametersOutput
 var params []types.Parameter
 var err error
 parameterPaginator := rds.NewDescribeDBParametersPaginator(instances.RdsClient,
  &rds.DescribeDBParametersInput{
   DBParameterGroupName: aws.String(parameterGroupName),
   Source:               aws.String(source),
  })
 for parameterPaginator.HasMorePages() {
  output, err = parameterPaginator.NextPage(context.TODO())
  if err != nil {
   log.Printf("Couldn't get parameters for %v: %v\n", parameterGroupName, err)
   break
  } else {
   params = append(params, output.Parameters...)
```

```
  }
 }
 return params, err
}
```

- Lihat detail API di DescribeDBParameters dalam Referensi API AWS SDK for Go .

Java

SDK for Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   // Retrieve parameters in the group.
   public static void describeDbParameters(RdsClient rdsClient, String
dbGroupName, int flag) {
       try {
           DescribeDbParametersRequest dbParameterGroupsRequest;
           if (flag == 0) {
               dbParameterGroupsRequest = DescribeDbParametersRequest.builder()
                       .dbParameterGroupName(dbGroupName)
                       .build();
           } else {
               dbParameterGroupsRequest = DescribeDbParametersRequest.builder()
                       .dbParameterGroupName(dbGroupName)
                       .source("user")
                       .build();
           }

           DescribeDbParametersResponse response =
rdsClient.describeDBParameters(dbParameterGroupsRequest);
           List<Parameter> dbParameters = response.parameters();
           String paraName;
           for (Parameter para : dbParameters) {
               // Only print out information about either auto_increment_offset
or
```

```
                // auto_increment_increment.
                paraName = para.parameterName();
                if ((paraName.compareTo("auto_increment_offset") == 0)
                        || (paraName.compareTo("auto_increment_increment ") ==
0)) {
                    System.out.println("*** The parameter name is  " + paraName);
                    System.out.println("*** The parameter value is  " +
para.parameterValue());
                    System.out.println("*** The parameter data type is " +
para.dataType());
                    System.out.println("*** The parameter description is " +
para.description());
                    System.out.println("*** The parameter allowed values  is " +
para.allowedValues());
                }
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
```

- Lihat detail API di DescribeDBParameters dalam Referensi API AWS SDK for Java 2.x .

Python

SDK for Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
```

```
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def get_parameters(self, parameter_group_name, name_prefix="", source=None):
        """
        Gets the parameters that are contained in a DB parameter group.

        :param parameter_group_name: The name of the parameter group to query.
        :param name_prefix: When specified, the retrieved list of parameters is
filtered
                            to contain only parameters that start with this
prefix.
        :param source: When specified, only parameters from this source are
retrieved.
                       For example, a source of 'user' retrieves only parameters
that
                       were set by a user.
        :return: The list of requested parameters.
        """
        try:
            kwargs = {"DBParameterGroupName": parameter_group_name}
            if source is not None:
                kwargs["Source"] = source
            parameters = []
            paginator = self.rds_client.get_paginator("describe_db_parameters")
            for page in paginator.paginate(**kwargs):
                parameters += [
                    p
                    for p in page["Parameters"]
                    if p["ParameterName"].startswith(name_prefix)
                ]
        except ClientError as err:
            logger.error(
                "Couldn't get parameters for %s. Here's why: %s: %s",
                parameter_group_name,
```

```
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return parameters
```

- Lihat detail API di DescribeDBParameters dalam Referensi API AWS SDK for Python (Boto3).

Ruby

SDK for Ruby

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
require "aws-sdk-rds"  # v2: require 'aws-sdk'

# List all Amazon Relational Database Service (Amazon RDS) parameter groups.
#
# @param rds_resource [Aws::RDS::Resource] An SDK for Ruby Amazon RDS resource.
# @return [Array, nil] List of all parameter groups, or nil if error.
def list_parameter_groups(rds_resource)
  parameter_groups = []
  rds_resource.db_parameter_groups.each do |p|
    parameter_groups.append({
                              "name": p.db_parameter_group_name,
                              "description": p.description
                            })
  end
  parameter_groups
rescue Aws::Errors::ServiceError => e
  puts "Couldn't list parameter groups:\n #{e.message}"
end
```

- Lihat detail API di DescribeDBParameters dalam Referensi API AWS SDK for Ruby .

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang cara memulai dan detail versi-versi SDK sebelumnya.

Jelaskan snapshot instans Amazon RDS DB menggunakan SDK AWS

Contoh-contoh kode berikut menunjukkan cara menjelaskan cuplikan instans basis data Amazon RDS.

Contoh-contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan di dalam konteks. Anda dapat melihat tindakan ini dalam konteks pada contoh kode berikut:

- Memulai instans basis data

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   /// <summary>
   /// Return a list of DB snapshots for a particular DB instance.
   /// </summary>
   /// <param name="dbInstanceIdentifier">DB instance identifier.</param>
   /// <returns>List of DB snapshots.</returns>
   public async Task<List<DBSnapshot>> DescribeDBSnapshots(string
dbInstanceIdentifier)
   {
       var results = new List<DBSnapshot>();
       var snapshotsPaginator = _amazonRDS.Paginators.DescribeDBSnapshots(
           new DescribeDBSnapshotsRequest()
           {
               DBInstanceIdentifier = dbInstanceIdentifier
```

```
            });

        // Get the entire list using the paginator.
        await foreach (var snapshots in snapshotsPaginator.DBSnapshots)
        {
            results.Add(snapshots);
        }
        return results;
    }
```

- Lihat detail API di DescribeDBSnapshots dalam Referensi API AWS SDK for .NET .

C++

SDK for C++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);

            Aws::RDS::Model::DescribeDBSnapshotsRequest request;
            request.SetDBSnapshotIdentifier(snapshotID);

            Aws::RDS::Model::DescribeDBSnapshotsOutcome outcome =
                    client.DescribeDBSnapshots(request);

            if (outcome.IsSuccess()) {
                snapshot = outcome.GetResult().GetDBSnapshots()[0];
            }
            else {
                std::cerr << "Error with RDS::DescribeDBSnapshots. "
                          << outcome.GetError().GetMessage()
```

```
                          << std::endl;
                cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
client);
                return false;
            }
```

- Lihat detail API di DescribeDBSnapshots dalam Referensi API AWS SDK for C++ .

CLI

AWS CLI

Contoh 1: Untuk mendeskripsikan snapshot DB untuk instance DB

`describe-db-snapshots`Contoh berikut mengambil rincian snapshot DB untuk instance DB.

```
aws rds describe-db-snapshots \
    --db-snapshot-identifier mydbsnapshot
```

Output:

```
{
    "DBSnapshots": [
        {
            "DBSnapshotIdentifier": "mydbsnapshot",
            "DBInstanceIdentifier": "mysqldb",
            "SnapshotCreateTime": "2018-02-08T22:28:08.598Z",
            "Engine": "mysql",
            "AllocatedStorage": 20,
            "Status": "available",
            "Port": 3306,
            "AvailabilityZone": "us-east-1f",
            "VpcId": "vpc-6594f31c",
            "InstanceCreateTime": "2018-02-08T22:24:55.973Z",
            "MasterUsername": "mysqladmin",
            "EngineVersion": "5.6.37",
            "LicenseModel": "general-public-license",
            "SnapshotType": "manual",
            "OptionGroupName": "default:mysql-5-6",
            "PercentProgress": 100,
```

```
            "StorageType": "gp2",
            "Encrypted": false,
            "DBSnapshotArn": "arn:aws:rds:us-
east-1:123456789012:snapshot:mydbsnapshot",
            "IAMDatabaseAuthenticationEnabled": false,
            "ProcessorFeatures": [],
            "DbiResourceId": "db-AKIAIOSFODNN7EXAMPLE"
        }
    ]
}
```

Untuk informasi selengkapnya, lihat Membuat Snapshot DB di Panduan Pengguna Amazon RDS.

Contoh 2: Untuk menemukan jumlah snapshot manual yang diambil

`describe-db-snapshots`Contoh berikut menggunakan `length` operator dalam `--query` opsi untuk mengembalikan jumlah snapshot manual yang telah diambil di AWS Wilayah tertentu.

```
aws rds describe-db-snapshots \
    --snapshot-type manual \
    --query "length(*[].{DBSnapshots:SnapshotType})" \
    --region eu-central-1
```

Output:

```
35
```

Untuk informasi selengkapnya, lihat Membuat Snapshot DB di Panduan Pengguna Amazon RDS.

- Untuk detail API, lihat DescribedBSnapshots di Referensi Perintah.AWS CLI

Go

SDK for Go V2

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// GetSnapshot gets a DB instance snapshot.
func (instances *DbInstances) GetSnapshot(snapshotName string)
 (*types.DBSnapshot, error) {
 output, err := instances.RdsClient.DescribeDBSnapshots(context.TODO(),
  &rds.DescribeDBSnapshotsInput{
   DBSnapshotIdentifier: aws.String(snapshotName),
  })
 if err != nil {
  log.Printf("Couldn't get snapshot %v: %v\n", snapshotName, err)
  return nil, err
 } else {
  return &output.DBSnapshots[0], nil
 }
}
```

- Lihat detail API di DescribeDBSnapshots dalam Referensi API AWS SDK for Go .

Python

SDK for Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def get_snapshot(self, snapshot_id):
        """
        Gets a DB instance snapshot.

        :param snapshot_id: The ID of the snapshot to retrieve.
        :return: The retrieved snapshot.
        """
        try:
            response = self.rds_client.describe_db_snapshots(
                DBSnapshotIdentifier=snapshot_id
            )
            snapshot = response["DBSnapshots"][0]
        except ClientError as err:
            logger.error(
```

```
                "Couldn't get snapshot %s. Here's why: %s: %s",
                snapshot_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return snapshot
```

- Lihat detail API di DescribeDBSnapshots dalam Referensi API AWS SDK for Python (Boto3).

Ruby

SDK for Ruby

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
require "aws-sdk-rds" # v2: require 'aws-sdk'

# List all Amazon Relational Database Service (Amazon RDS) DB instance
# snapshots.
#
# @param rds_resource [Aws::RDS::Resource] An SDK for Ruby Amazon RDS resource.
# @return instance_snapshots [Array, nil] All instance snapshots, or nil if
 error.
def list_instance_snapshots(rds_resource)
  instance_snapshots = []
  rds_resource.db_snapshots.each do |s|
    instance_snapshots.append({
                                "id": s.snapshot_id,
                                "status": s.status
                              })
  end
  instance_snapshots
```

```
rescue Aws::Errors::ServiceError => e
  puts "Couldn't list instance snapshots:\n #{e.message}"
end
```

- Lihat detail API di DescribeDBSnapshots dalam Referensi API AWS SDK for Ruby .

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang cara memulai dan detail versi-versi SDK sebelumnya.

Memodifikasi instans Amazon RDS DB menggunakan SDK AWS

Contoh kode berikut menunjukkan cara mengubah instans basis data Amazon RDS.

CLI

AWS CLI

Untuk memodifikasi instance DB

`modify-db-instance`Contoh berikut mengaitkan grup opsi dan grup parameter dengan instance Microsoft SQL Server DB yang kompatibel. `--apply-immediately`Parameter menyebabkan grup opsi dan parameter segera dikaitkan, alih-alih menunggu hingga jendela pemeliharaan berikutnya.

```
aws rds modify-db-instance \
    --db-instance-identifier database-2 \
    --option-group-name test-se-2017 \
    --db-parameter-group-name test-sqlserver-se-2017 \
    --apply-immediately
```

Output:

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "database-2",
        "DBInstanceClass": "db.r4.large",
        "Engine": "sqlserver-se",
```

```
        "DBInstanceStatus": "available",

        ...output omitted...

        "DBParameterGroups": [
            {
                "DBParameterGroupName": "test-sqlserver-se-2017",
                "ParameterApplyStatus": "applying"
            }
        ],
        "AvailabilityZone": "us-west-2d",

        ...output omitted...

        "MultiAZ": true,
        "EngineVersion": "14.00.3281.6.v1",
        "AutoMinorVersionUpgrade": false,
        "ReadReplicaDBInstanceIdentifiers": [],
        "LicenseModel": "license-included",
        "OptionGroupMemberships": [
            {
                "OptionGroupName": "test-se-2017",
                "Status": "pending-apply"
            }
        ],
        "CharacterSetName": "SQL_Latin1_General_CP1_CI_AS",
        "SecondaryAvailabilityZone": "us-west-2c",
        "PubliclyAccessible": true,
        "StorageType": "gp2",

        ...output omitted...

        "DeletionProtection": false,
        "AssociatedRoles": [],
        "MaxAllocatedStorage": 1000
    }
}
```

Untuk informasi selengkapnya, lihat Memodifikasi Instans Amazon RDS DB di Panduan Pengguna Amazon RDS.

- Untuk detail API, lihat ModifyDBInstance di Referensi Perintah.AWS CLI

Java

SDK for 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.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.ModifyDbInstanceRequest;
import software.amazon.awssdk.services.rds.model.ModifyDbInstanceResponse;
import software.amazon.awssdk.services.rds.model.RdsException;

/**
 * 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 ModifyDBInstance {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <dbInstanceIdentifier> <dbSnapshotIdentifier>\s
                Where:
                    dbInstanceIdentifier - The database instance identifier.\s
                    masterUserPassword - The updated password that corresponds to
 the master user name.\s
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }
```

```
        String dbInstanceIdentifier = args[0];
        String masterUserPassword = args[1];
        Region region = Region.US_WEST_2;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();

        updateIntance(rdsClient, dbInstanceIdentifier, masterUserPassword);
        rdsClient.close();
    }

    public static void updateIntance(RdsClient rdsClient, String
 dbInstanceIdentifier, String masterUserPassword) {
        try {
            // For a demo - modify the DB instance by modifying the master
 password.
            ModifyDbInstanceRequest modifyDbInstanceRequest =
 ModifyDbInstanceRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .publiclyAccessible(true)
                    .masterUserPassword(masterUserPassword)
                    .build();

            ModifyDbInstanceResponse instanceResponse =
 rdsClient.modifyDBInstance(modifyDbInstanceRequest);
            System.out.print("The ARN of the modified database is: " +
 instanceResponse.dbInstance().dbInstanceArn());

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
}
```

- Lihat detail API di ModifyDBInstance dalam Referensi API AWS SDK for Java 2.x .

Kotlin

SDK for Kotlin

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
suspend fun updateIntance(dbInstanceIdentifierVal: String?,
 masterUserPasswordVal: String?) {

    val request = ModifyDbInstanceRequest {
        dbInstanceIdentifier = dbInstanceIdentifierVal
        publiclyAccessible = true
        masterUserPassword = masterUserPasswordVal
    }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val instanceResponse = rdsClient.modifyDbInstance(request)
        println("The ARN of the modified database is
 ${instanceResponse.dbInstance?.dbInstanceArn}")
    }
}
```

- Lihat detail API di ModifyDBInstance dalam Referensi API AWS SDK for Kotlin.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang cara memulai dan detail versi-versi SDK sebelumnya.

Reboot instans Amazon RDS DB menggunakan SDK AWS

Contoh kode berikut menunjukkan cara me-reboot instans Amazon RDS DB.

CLI

AWS CLI

Untuk me-reboot instance DB

`reboot-db-instance`Contoh berikut memulai reboot dari instance DB yang ditentukan.

```
aws rds reboot-db-instance \
    --db-instance-identifier test-mysql-instance
```

Output:

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "test-mysql-instance",
        "DBInstanceClass": "db.t3.micro",
        "Engine": "mysql",
        "DBInstanceStatus": "rebooting",
        "MasterUsername": "admin",
        "Endpoint": {
            "Address": "test-mysql-instance.############.us-
west-2.rds.amazonaws.com",
            "Port": 3306,
            "HostedZoneId": "Z1PVIF0EXAMPLE"
        },

    ... output omitted...

    }
}
```

Untuk informasi selengkapnya, lihat Mem-boot Ulang Instans DB di Panduan Pengguna Amazon RDS.

- Untuk detail API, lihat RebootdBInstance di Referensi Perintah AWS CLI .

Java

SDK for 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.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.RebootDbInstanceRequest;
import software.amazon.awssdk.services.rds.model.RebootDbInstanceResponse;
import software.amazon.awssdk.services.rds.model.RdsException;

/**
 * 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 RebootDBInstance {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <dbInstanceIdentifier>\s

                Where:
                    dbInstanceIdentifier - The database instance identifier\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String dbInstanceIdentifier = args[0];
```

```
        Region region = Region.US_WEST_2;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();

        rebootInstance(rdsClient, dbInstanceIdentifier);
        rdsClient.close();
    }

    public static void rebootInstance(RdsClient rdsClient, String
 dbInstanceIdentifier) {
        try {
            RebootDbInstanceRequest rebootDbInstanceRequest =
 RebootDbInstanceRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .build();

            RebootDbInstanceResponse instanceResponse =
 rdsClient.rebootDBInstance(rebootDbInstanceRequest);
            System.out.print("The database " +
 instanceResponse.dbInstance().dbInstanceArn() + " was rebooted");

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
}
```

- Lihat detail API di RebootDBInstance dalam Referensi API AWS SDK for Java 2.x .

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang cara memulai dan detail versi-versi SDK sebelumnya.

Mengambil atribut yang dimiliki akun Amazon RDS menggunakan SDK AWS

Contoh-contoh kode berikut menunjukkan cara mengambil atribut-atribut milik akun Amazon RDS.

CLI

AWS CLI

Untuk menggambarkan atribut akun

`describe-account-attributes`Contoh berikut mengambil atribut untuk AWS akun saat ini.

```
aws rds describe-account-attributes
```

Output:

```
{
    "AccountQuotas": [
        {
            "Max": 40,
            "Used": 4,
            "AccountQuotaName": "DBInstances"
        },
        {
            "Max": 40,
            "Used": 0,
            "AccountQuotaName": "ReservedDBInstances"
        },
        {
            "Max": 100000,
            "Used": 40,
            "AccountQuotaName": "AllocatedStorage"
        },
        {
            "Max": 25,
            "Used": 0,
            "AccountQuotaName": "DBSecurityGroups"
        },
        {
            "Max": 20,
            "Used": 0,
            "AccountQuotaName": "AuthorizationsPerDBSecurityGroup"
        },
        {
            "Max": 50,
            "Used": 1,
```

```
            "AccountQuotaName": "DBParameterGroups"
        },
        {
            "Max": 100,
            "Used": 3,
            "AccountQuotaName": "ManualSnapshots"
        },
        {
            "Max": 20,
            "Used": 0,
            "AccountQuotaName": "EventSubscriptions"
        },
        {
            "Max": 50,
            "Used": 1,
            "AccountQuotaName": "DBSubnetGroups"
        },
        {
            "Max": 20,
            "Used": 1,
            "AccountQuotaName": "OptionGroups"
        },
        {
            "Max": 20,
            "Used": 6,
            "AccountQuotaName": "SubnetsPerDBSubnetGroup"
        },
        {
            "Max": 5,
            "Used": 0,
            "AccountQuotaName": "ReadReplicasPerMaster"
        },
        {
            "Max": 40,
            "Used": 1,
            "AccountQuotaName": "DBClusters"
        },
        {
            "Max": 50,
            "Used": 0,
            "AccountQuotaName": "DBClusterParameterGroups"
        },
        {
            "Max": 5,
```

```
            "Used": 0,
            "AccountQuotaName": "DBClusterRoles"
        }
    ]
}
```

- Untuk detail API, lihat DescribeAccountAttributesdi Referensi AWS CLI Perintah.

Java

SDK for 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.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.AccountQuota;
import software.amazon.awssdk.services.rds.model.RdsException;
import
 software.amazon.awssdk.services.rds.model.DescribeAccountAttributesResponse;
import java.util.List;

/**
 * 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 DescribeAccountAttributes {
    public static void main(String[] args) {
        Region region = Region.US_WEST_2;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();
```

```
        getAccountAttributes(rdsClient);
        rdsClient.close();
    }

    public static void getAccountAttributes(RdsClient rdsClient) {
        try {
            DescribeAccountAttributesResponse response =
 rdsClient.describeAccountAttributes();
            List<AccountQuota> quotasList = response.accountQuotas();
            for (AccountQuota quotas : quotasList) {
                System.out.println("Name is: " + quotas.accountQuotaName());
                System.out.println("Max value is " + quotas.max());
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
}
```

- Untuk detail API, lihat DescribeAccountAttributesdi Referensi AWS SDK for Java 2.x API.

Kotlin

SDK for Kotlin

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
suspend fun getAccountAttributes() {

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response =
 rdsClient.describeAccountAttributes(DescribeAccountAttributesRequest {})
        response.accountQuotas?.forEach { quotas ->
            val response = response.accountQuotas
            println("Name is: ${quotas.accountQuotaName}")
```

```
            println("Max value is ${quotas.max}")
        }
    }
}
```

- Untuk detail API, lihat DescribeAccountAttributesdi AWS SDK untuk referensi API Kotlin.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang cara memulai dan detail versi-versi SDK sebelumnya.

Memperbarui parameter dalam grup parameter Amazon RDS DB menggunakan SDK AWS

Contoh-contoh kode berikut menunjukkan cara memperbarui parameter dalam grup parameter basis data Amazon RDS.

Contoh-contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan di dalam konteks. Anda dapat melihat tindakan ini dalam konteks pada contoh kode berikut:

- Memulai instans basis data

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    /// <summary>
    /// Update a DB parameter group. Use the action
DescribeDBParameterGroupsAsync
    /// to determine when the DB parameter group is ready to use.
    /// </summary>
```

```
    /// <param name="name">Name of the DB parameter group.</param>
    /// <param name="parameters">List of parameters. Maximum of 20 per request.</
param>
    /// <returns>The updated DB parameter group name.</returns>
    public async Task<string> ModifyDBParameterGroup(
        string name, List<Parameter> parameters)
    {
        var response = await _amazonRDS.ModifyDBParameterGroupAsync(
            new ModifyDBParameterGroupRequest()
            {
                DBParameterGroupName = name,
                Parameters = parameters,
            });
        return response.DBParameterGroupName;
    }
```

- Untuk detail API, lihat ModifyDB ParameterGroup di AWS SDK for .NET Referensi API.

C++

SDK for C++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);

        Aws::RDS::Model::ModifyDBParameterGroupRequest request;
        request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
        request.SetParameters(updateParameters);

        Aws::RDS::Model::ModifyDBParameterGroupOutcome outcome =
                client.ModifyDBParameterGroup(request);
```

```
        if (outcome.IsSuccess()) {
            std::cout << "The DB parameter group was successfully modified."
                      << std::endl;
        }
        else {
            std::cerr << "Error with RDS::ModifyDBParameterGroup. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
        }
```

- Untuk detail API, lihat ModifyDB ParameterGroup di AWS SDK for C++ Referensi API.

CLI

AWS CLI

Untuk memodifikasi grup parameter DB

`modify-db-parameter-group`Contoh berikut mengubah nilai parameter dalam kelompok `clr enabled` parameter DB. `--apply-immediately`Parameter menyebabkan grup parameter DB segera dimodifikasi, alih-alih menunggu hingga jendela pemeliharaan berikutnya.

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name test-sqlserver-se-2017 \
    --parameters "ParameterName='clr
 enabled',ParameterValue=1,ApplyMethod=immediate"
```

Output:

```
{
    "DBParameterGroupName": "test-sqlserver-se-2017"
}
```

Untuk informasi selengkapnya, lihat Memodifikasi Parameter dalam Grup Parameter DB di Panduan Pengguna Amazon RDS.

- Untuk detail API, lihat ModifyDB ParameterGroup di AWS CLI Referensi Perintah.

Go

SDK for Go V2

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// UpdateParameters updates parameters in a named DB parameter group.
func (instances *DbInstances) UpdateParameters(parameterGroupName string, params
 []types.Parameter) error {
 _, err := instances.RdsClient.ModifyDBParameterGroup(context.TODO(),
  &rds.ModifyDBParameterGroupInput{
   DBParameterGroupName: aws.String(parameterGroupName),
   Parameters:           params,
  })
 if err != nil {
  log.Printf("Couldn't update parameters in %v: %v\n", parameterGroupName, err)
  return err
 } else {
  return nil
 }
}
```

- Untuk detail API, lihat ModifyDB ParameterGroup di AWS SDK for Go Referensi API.

Java

SDK for Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   // Modify auto_increment_offset and auto_increment_increment parameters.
   public static void modifyDBParas(RdsClient rdsClient, String dbGroupName) {
       try {
           Parameter parameter1 = Parameter.builder()
                   .parameterName("auto_increment_offset")
                   .applyMethod("immediate")
                   .parameterValue("5")
                   .build();

           List<Parameter> paraList = new ArrayList<>();
           paraList.add(parameter1);
           ModifyDbParameterGroupRequest groupRequest =
ModifyDbParameterGroupRequest.builder()
                   .dbParameterGroupName(dbGroupName)
                   .parameters(paraList)
                   .build();

           ModifyDbParameterGroupResponse response =
rdsClient.modifyDBParameterGroup(groupRequest);
           System.out.println("The parameter group " +
response.dbParameterGroupName() + " was successfully modified");

       } catch (RdsException e) {
           System.out.println(e.getLocalizedMessage());
           System.exit(1);
       }
   }
```

- Untuk detail API, lihat ModifyDB ParameterGroup di AWS SDK for Java 2.x Referensi API.

Python

SDK for Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def update_parameters(self, parameter_group_name, update_parameters):
        """
        Updates parameters in a custom DB parameter group.

        :param parameter_group_name: The name of the parameter group to update.
        :param update_parameters: The parameters to update in the group.
        :return: Data about the modified parameter group.
        """
        try:
            response = self.rds_client.modify_db_parameter_group(
                DBParameterGroupName=parameter_group_name,
 Parameters=update_parameters
            )
        except ClientError as err:
```

```
            logger.error(
                "Couldn't update parameters in %s. Here's why: %s: %s",
                parameter_group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return response
```

- Untuk detail API, lihat ModifyDB ParameterGroup di AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang cara memulai dan detail versi-versi SDK sebelumnya.

Skenario untuk Amazon RDS menggunakan AWS SDK

Contoh kode berikut menunjukkan cara menerapkan skenario umum di Amazon RDS dengan AWS SDK. Skenario ini menunjukkan cara menyelesaikan tugas tertentu dengan memanggil beberapa fungsi di dalam Amazon RDS. Setiap skenario menyertakan tautan ke GitHub, di mana Anda dapat menemukan petunjuk tentang cara mengatur dan menjalankan kode.

Contoh-contoh

- Memulai instans Amazon RDS DB menggunakan SDK AWS

Memulai instans Amazon RDS DB menggunakan SDK AWS

Contoh-contoh kode berikut menunjukkan cara:

- Membuat grup parameter basis data kustom dan mengatur nilai parameter.
- Membuat instans basis data yang dikonfigurasikan untuk menggunakan grup parameter. Instans basis data juga berisi basis data.
- Mengambil cuplikan instans.
- Menghapus instans dan grup parameter.

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

Jalankan skenario interaktif di penggugah/prompt perintah.

```
/// <summary>
/// Scenario for RDS DB instance example.
/// </summary>
public class RDSInstanceScenario
{
    /*
    Before running this .NET code example, set up your development environment,
 including your credentials.

    This .NET example performs the following tasks:
    1.  Returns a list of the available DB engine families using the
 DescribeDBEngineVersionsAsync method.
    2.  Selects an engine family and creates a custom DB parameter group using
 the CreateDBParameterGroupAsync method.
    3.  Gets the parameter groups using the DescribeDBParameterGroupsAsync
 method.
    4.  Gets parameters in the group using the DescribeDBParameters method.
    5.  Parses and displays parameters in the group.
    6.  Modifies both the auto_increment_offset and auto_increment_increment
 parameters
        using the ModifyDBParameterGroupAsync method.
    7.  Gets and displays the updated parameters using the DescribeDBParameters
 method with a source of "user".
    8.  Gets a list of allowed engine versions using the
 DescribeDBEngineVersionsAsync method.
    9.  Displays and selects from a list of micro instance classes available for
 the selected engine and version.
    10. Creates an RDS DB instance that contains a MySql database and uses the
 parameter group
        using the CreateDBInstanceAsync method.
```

```
   11. Waits for DB instance to be ready using the DescribeDBInstancesAsync
method.
   12. Prints out the connection endpoint string for the new DB instance.
   13. Creates a snapshot of the DB instance using the CreateDBSnapshotAsync
method.
   14. Waits for DB snapshot to be ready using the DescribeDBSnapshots method.
   15. Deletes the DB instance using the DeleteDBInstanceAsync method.
   16. Waits for DB instance to be deleted using the DescribeDbInstances method.
   17. Deletes the parameter group using the DeleteDBParameterGroupAsync.
   */

   private static readonly string sepBar = new('-', 80);
   private static RDSWrapper rdsWrapper = null!;
   private static ILogger logger = null!;
   private static readonly string engine = "mysql";
   static async Task Main(string[] args)
   {
       // Set up dependency injection for the Amazon RDS service.
       using var host = Host.CreateDefaultBuilder(args)
           .ConfigureLogging(logging =>
               logging.AddFilter("System", LogLevel.Debug)
                   .AddFilter<DebugLoggerProvider>("Microsoft",
LogLevel.Information)
                   .AddFilter<ConsoleLoggerProvider>("Microsoft",
LogLevel.Trace))
           .ConfigureServices((_, services) =>
               services.AddAWSService<IAmazonRDS>()
                   .AddTransient<RDSWrapper>()
           )
           .Build();

       logger = LoggerFactory.Create(builder =>
       {
           builder.AddConsole();
       }).CreateLogger<RDSInstanceScenario>();

       rdsWrapper = host.Services.GetRequiredService<RDSWrapper>();

       Console.WriteLine(sepBar);
       Console.WriteLine(
           "Welcome to the Amazon Relational Database Service (Amazon RDS) DB
instance scenario example.");
       Console.WriteLine(sepBar);
```

```
        try
        {
            var parameterGroupFamily = await ChooseParameterGroupFamily();

            var parameterGroup = await
CreateDbParameterGroup(parameterGroupFamily);

            var parameters = await
DescribeParametersInGroup(parameterGroup.DBParameterGroupName,
                new List<string> { "auto_increment_offset",
"auto_increment_increment" });

            await ModifyParameters(parameterGroup.DBParameterGroupName,
parameters);

            await
DescribeUserSourceParameters(parameterGroup.DBParameterGroupName);

            var engineVersionChoice = await
ChooseDbEngineVersion(parameterGroupFamily);

            var instanceChoice = await ChooseDbInstanceClass(engine,
engineVersionChoice.EngineVersion);

            var newInstanceIdentifier = "Example-Instance-" + DateTime.Now.Ticks;

            var newInstance = await CreateRdsNewInstance(parameterGroup, engine,
engineVersionChoice.EngineVersion,
                instanceChoice.DBInstanceClass, newInstanceIdentifier);
            if (newInstance != null)
            {
                DisplayConnectionString(newInstance);

                await CreateSnapshot(newInstance);

                await DeleteRdsInstance(newInstance);
            }

            await DeleteParameterGroup(parameterGroup);

            Console.WriteLine("Scenario complete.");
            Console.WriteLine(sepBar);
        }
        catch (Exception ex)
```

```
        {
            logger.LogError(ex, "There was a problem executing the scenario.");
        }
    }

    /// <summary>
    /// Choose the RDS DB parameter group family from a list of available
options.
    /// </summary>
    /// <returns>The selected parameter group family.</returns>
    public static async Task<string> ChooseParameterGroupFamily()
    {
        Console.WriteLine(sepBar);
        // 1. Get a list of available engines.
        var engines = await rdsWrapper.DescribeDBEngineVersions(engine);

        Console.WriteLine("1. The following is a list of available DB parameter
group families:");
        int i = 1;
        var parameterGroupFamilies = engines.GroupBy(e =>
e.DBParameterGroupFamily).ToList();
        foreach (var parameterGroupFamily in parameterGroupFamilies)
        {
            // List the available parameter group families.
            Console.WriteLine(
                $"\t{i}. Family: {parameterGroupFamily.Key}");
            i++;
        }

        var choiceNumber = 0;
        while (choiceNumber < 1 || choiceNumber > parameterGroupFamilies.Count)
        {
            Console.WriteLine("Select an available DB parameter group family by
entering a number from the list above:");
            var choice = Console.ReadLine();
            Int32.TryParse(choice, out choiceNumber);
        }
        var parameterGroupFamilyChoice = parameterGroupFamilies[choiceNumber -
1];
        Console.WriteLine(sepBar);
        return parameterGroupFamilyChoice.Key;
    }

    /// <summary>
```

```
   /// Create and get information on a DB parameter group.
   /// </summary>
   /// <param name="dbParameterGroupFamily">The DBParameterGroupFamily for the
new DB parameter group.</param>
   /// <returns>The new DBParameterGroup.</returns>
   public static async Task<DBParameterGroup> CreateDbParameterGroup(string
dbParameterGroupFamily)
   {
       Console.WriteLine(sepBar);
       Console.WriteLine($"2. Create new DB parameter group with family
{dbParameterGroupFamily}:");

       var parameterGroup = await rdsWrapper.CreateDBParameterGroup(
           "ExampleParameterGroup-" + DateTime.Now.Ticks,
           dbParameterGroupFamily, "New example parameter group");

       var groupInfo =
           await rdsWrapper.DescribeDBParameterGroups(parameterGroup
               .DBParameterGroupName);

       Console.WriteLine(
           $"3. New DB parameter group: \n\t{groupInfo[0].Description}, \n\tARN
{groupInfo[0].DBParameterGroupArn}");
       Console.WriteLine(sepBar);
       return parameterGroup;
   }

   /// <summary>
   /// Get and describe parameters from a DBParameterGroup.
   /// </summary>
   /// <param name="parameterGroupName">Name of the DBParameterGroup.</param>
   /// <param name="parameterNames">Optional specific names of parameters to
describe.</param>
   /// <returns>The list of requested parameters.</returns>
   public static async Task<List<Parameter>> DescribeParametersInGroup(string
parameterGroupName, List<string>? parameterNames = null)
   {
       Console.WriteLine(sepBar);
       Console.WriteLine("4. Get some parameters from the group.");
       Console.WriteLine(sepBar);

       var parameters =
           await rdsWrapper.DescribeDBParameters(parameterGroupName);
```

```
        var matchingParameters =
            parameters.Where(p => parameterNames == null ||
parameterNames.Contains(p.ParameterName)).ToList();

        Console.WriteLine("5. Parameter information:");
        matchingParameters.ForEach(p =>
            Console.WriteLine(
                $"\n\tParameter: {p.ParameterName}." +
                $"\n\tDescription: {p.Description}." +
                $"\n\tAllowed Values: {p.AllowedValues}." +
                $"\n\tValue: {p.ParameterValue}."));

        Console.WriteLine(sepBar);

        return matchingParameters;
    }

    /// <summary>
    /// Modify a parameter from a DBParameterGroup.
    /// </summary>
    /// <param name="parameterGroupName">Name of the DBParameterGroup.</param>
    /// <param name="parameters">The parameters to modify.</param>
    /// <returns>Async task.</returns>
    public static async Task ModifyParameters(string parameterGroupName,
List<Parameter> parameters)
    {
        Console.WriteLine(sepBar);
        Console.WriteLine("6. Modify some parameters in the group.");

        foreach (var p in parameters)
        {
            if (p.IsModifiable && p.DataType == "integer")
            {
                int newValue = 0;
                while (newValue == 0)
                {
                    Console.WriteLine(
                        $"Enter a new value for {p.ParameterName} from the
allowed values {p.AllowedValues} ");

                    var choice = Console.ReadLine();
                    Int32.TryParse(choice, out newValue);
                }
```

```
                p.ParameterValue = newValue.ToString();
            }
        }

        await rdsWrapper.ModifyDBParameterGroup(parameterGroupName, parameters);

        Console.WriteLine(sepBar);
    }

    /// <summary>
    /// Describe the user source parameters in the group.
    /// </summary>
    /// <param name="parameterGroupName">Name of the DBParameterGroup.</param>
    /// <returns>Async task.</returns>
    public static async Task DescribeUserSourceParameters(string
parameterGroupName)
    {
        Console.WriteLine(sepBar);
        Console.WriteLine("7. Describe user source parameters in the group.");

        var parameters =
            await rdsWrapper.DescribeDBParameters(parameterGroupName, "user");


        parameters.ForEach(p =>
            Console.WriteLine(
                $"\n\tParameter: {p.ParameterName}." +
                $"\n\tDescription: {p.Description}." +
                $"\n\tAllowed Values: {p.AllowedValues}." +
                $"\n\tValue: {p.ParameterValue}."));

        Console.WriteLine(sepBar);
    }


    /// <summary>
    /// Choose a DB engine version.
    /// </summary>
    /// <param name="dbParameterGroupFamily">DB parameter group family for engine
choice.</param>
    /// <returns>The selected engine version.</returns>
    public static async Task<DBEngineVersion> ChooseDbEngineVersion(string
dbParameterGroupFamily)
    {
```

```
        Console.WriteLine(sepBar);
        // Get a list of allowed engines.
        var allowedEngines =
            await rdsWrapper.DescribeDBEngineVersions(engine,
 dbParameterGroupFamily);

        Console.WriteLine($"Available DB engine versions for parameter group
 family {dbParameterGroupFamily}:");
        int i = 1;
        foreach (var version in allowedEngines)
        {
            Console.WriteLine(
                $"\t{i}. Engine: {version.Engine} Version
 {version.EngineVersion}.");
            i++;
        }

        var choiceNumber = 0;
        while (choiceNumber < 1 || choiceNumber > allowedEngines.Count)
        {
            Console.WriteLine("8. Select an available DB engine version by
 entering a number from the list above:");
            var choice = Console.ReadLine();
            Int32.TryParse(choice, out choiceNumber);
        }

        var engineChoice = allowedEngines[choiceNumber - 1];
        Console.WriteLine(sepBar);
        return engineChoice;
    }

    /// <summary>
    /// Choose a DB instance class for a particular engine and engine version.
    /// </summary>
    /// <param name="engine">DB engine for DB instance choice.</param>
    /// <param name="engineVersion">DB engine version for DB instance choice.</
param>
    /// <returns>The selected orderable DB instance option.</returns>
    public static async Task<OrderableDBInstanceOption>
 ChooseDbInstanceClass(string engine, string engineVersion)
    {
        Console.WriteLine(sepBar);
        // Get a list of allowed DB instance classes.
        var allowedInstances =
```

```
            await rdsWrapper.DescribeOrderableDBInstanceOptions(engine,
 engineVersion);

        Console.WriteLine($"8. Available micro DB instance classes for engine
 {engine} and version {engineVersion}:");
        int i = 1;

        // Filter to micro instances for this example.
        allowedInstances = allowedInstances
            .Where(i => i.DBInstanceClass.Contains("micro")).ToList();

        foreach (var instance in allowedInstances)
        {
            Console.WriteLine(
                $"\t{i}. Instance class: {instance.DBInstanceClass} (storage type
 {instance.StorageType})");
            i++;
        }

        var choiceNumber = 0;
        while (choiceNumber < 1 || choiceNumber > allowedInstances.Count)
        {
            Console.WriteLine("9. Select an available DB instance class by
 entering a number from the list above:");
            var choice = Console.ReadLine();
            Int32.TryParse(choice, out choiceNumber);
        }

        var instanceChoice = allowedInstances[choiceNumber - 1];
        Console.WriteLine(sepBar);
        return instanceChoice;
    }

    /// <summary>
    /// Create a new RDS DB instance.
    /// </summary>
    /// <param name="parameterGroup">Parameter group to use for the DB
 instance.</param>
    /// <param name="engineName">Engine to use for the DB instance.</param>
    /// <param name="engineVersion">Engine version to use for the DB instance.</
param>
    /// <param name="instanceClass">Instance class to use for the DB instance.</
param>
```

```
   /// <param name="instanceIdentifier">Instance identifier to use for the DB
instance.</param>
   /// <returns>The new DB instance.</returns>
   public static async Task<DBInstance?> CreateRdsNewInstance(DBParameterGroup
parameterGroup,
       string engineName, string engineVersion, string instanceClass, string
instanceIdentifier)
   {
       Console.WriteLine(sepBar);
       Console.WriteLine($"10. Create a new DB instance with identifier
{instanceIdentifier}.");
       bool isInstanceReady = false;
       DBInstance newInstance;
       var instances = await rdsWrapper.DescribeDBInstances();
       isInstanceReady = instances.FirstOrDefault(i =>
           i.DBInstanceIdentifier == instanceIdentifier)?.DBInstanceStatus ==
"available";

       if (isInstanceReady)
       {
           Console.WriteLine("Instance already created.");
           newInstance = instances.First(i => i.DBInstanceIdentifier ==
instanceIdentifier);
       }
       else
       {
           Console.WriteLine("Please enter an admin user name:");
           var username = Console.ReadLine();

           Console.WriteLine("Please enter an admin password:");
           var password = Console.ReadLine();

           newInstance = await rdsWrapper.CreateDBInstance(
               "ExampleInstance",
               instanceIdentifier,
               parameterGroup.DBParameterGroupName,
               engineName,
               engineVersion,
               instanceClass,
               20,
               username,
               password
           );
```

```
            // 11. Wait for the DB instance to be ready.

            Console.WriteLine("11. Waiting for DB instance to be ready...");
            while (!isInstanceReady)
            {
                instances = await
 rdsWrapper.DescribeDBInstances(instanceIdentifier);
                isInstanceReady = instances.FirstOrDefault()?.DBInstanceStatus ==
 "available";
                newInstance = instances.First();
                Thread.Sleep(30000);
            }
        }

        Console.WriteLine(sepBar);
        return newInstance;
    }

    /// <summary>
    /// Display a connection string for an RDS DB instance.
    /// </summary>
    /// <param name="instance">The DB instance to use to get a connection
 string.</param>
    public static void DisplayConnectionString(DBInstance instance)
    {
        Console.WriteLine(sepBar);
        // Display the connection string.
        Console.WriteLine("12. New DB instance connection string: ");
        Console.WriteLine(
            $"\n{engine} -h {instance.Endpoint.Address} -P
 {instance.Endpoint.Port} "
            + $"-u {instance.MasterUsername} -p [YOUR PASSWORD]\n");

        Console.WriteLine(sepBar);
    }

    /// <summary>
    /// Create a snapshot from an RDS DB instance.
    /// </summary>
    /// <param name="instance">DB instance to use when creating a snapshot.</
param>
    /// <returns>The snapshot object.</returns>
    public static async Task<DBSnapshot> CreateSnapshot(DBInstance instance)
    {
```

```
        Console.WriteLine(sepBar);
        // Create a snapshot.
        Console.WriteLine($"13. Creating snapshot from DB instance
{instance.DBInstanceIdentifier}.");
        var snapshot = await
rdsWrapper.CreateDBSnapshot(instance.DBInstanceIdentifier, "ExampleSnapshot-" +
DateTime.Now.Ticks);

        // Wait for the snapshot to be available
        bool isSnapshotReady = false;

        Console.WriteLine($"14. Waiting for snapshot to be ready...");
        while (!isSnapshotReady)
        {
            var snapshots = await
rdsWrapper.DescribeDBSnapshots(instance.DBInstanceIdentifier);
            isSnapshotReady = snapshots.FirstOrDefault()?.Status == "available";
            snapshot = snapshots.First();
            Thread.Sleep(30000);
        }

        Console.WriteLine(
            $"Snapshot {snapshot.DBSnapshotIdentifier} status is
{snapshot.Status}.");
        Console.WriteLine(sepBar);
        return snapshot;
    }

    /// <summary>
    /// Delete an RDS DB instance.
    /// </summary>
    /// <param name="instance">The DB instance to delete.</param>
    /// <returns>Async task.</returns>
    public static async Task DeleteRdsInstance(DBInstance newInstance)
    {
        Console.WriteLine(sepBar);
        // Delete the DB instance.
        Console.WriteLine($"15. Delete the DB instance
{newInstance.DBInstanceIdentifier}.");
        await rdsWrapper.DeleteDBInstance(newInstance.DBInstanceIdentifier);

        // Wait for the DB instance to delete.
        Console.WriteLine($"16. Waiting for the DB instance to delete...");
        bool isInstanceDeleted = false;
```

```
        while (!isInstanceDeleted)
        {
            var instance = await rdsWrapper.DescribeDBInstances();
            isInstanceDeleted = instance.All(i => i.DBInstanceIdentifier != 
 newInstance.DBInstanceIdentifier);
            Thread.Sleep(30000);
        }

        Console.WriteLine("DB instance deleted.");
        Console.WriteLine(sepBar);
    }

    /// <summary>
    /// Delete a DB parameter group.
    /// </summary>
    /// <param name="parameterGroup">The parameter group to delete.</param>
    /// <returns>Async task.</returns>
    public static async Task DeleteParameterGroup(DBParameterGroup 
 parameterGroup)
    {
        Console.WriteLine(sepBar);
        // Delete the parameter group.
        Console.WriteLine($"17. Delete the DB parameter group 
 {parameterGroup.DBParameterGroupName}.");
        await 
 rdsWrapper.DeleteDBParameterGroup(parameterGroup.DBParameterGroupName);

        Console.WriteLine(sepBar);
    }
```

Metode pembungkus yang digunakan oleh skenario untuk tindakan instans basis data.

```
/// <summary>
/// Wrapper methods to use Amazon Relational Database Service (Amazon RDS) with
 DB instance operations.
/// </summary>
public partial class RDSWrapper
{
    private readonly IAmazonRDS _amazonRDS;
    public RDSWrapper(IAmazonRDS amazonRDS)
```

```
    {
        _amazonRDS = amazonRDS;
    }


    /// <summary>
    /// Get a list of DB engine versions for a particular DB engine.
    /// </summary>
    /// <param name="engine">Name of the engine.</param>
    /// <param name="dbParameterGroupFamily">Optional parameter group family
name.</param>
    /// <returns>List of DBEngineVersions.</returns>
    public async Task<List<DBEngineVersion>> DescribeDBEngineVersions(string
engine,
        string dbParameterGroupFamily = null)
    {
        var response = await _amazonRDS.DescribeDBEngineVersionsAsync(
            new DescribeDBEngineVersionsRequest()
            {
                Engine = engine,
                DBParameterGroupFamily = dbParameterGroupFamily
            });
        return response.DBEngineVersions;
    }



    /// <summary>
    /// Get a list of orderable DB instance options for a specific
    /// engine and engine version.
    /// </summary>
    /// <param name="engine">Name of the engine.</param>
    /// <param name="engineVersion">Version of the engine.</param>
    /// <returns>List of OrderableDBInstanceOptions.</returns>
    public async Task<List<OrderableDBInstanceOption>>
DescribeOrderableDBInstanceOptions(string engine, string engineVersion)
    {
        // Use a paginator to get a list of DB instance options.
        var results = new List<OrderableDBInstanceOption>();
        var paginateInstanceOptions =
_amazonRDS.Paginators.DescribeOrderableDBInstanceOptions(
            new DescribeOrderableDBInstanceOptionsRequest()
            {
                Engine = engine,
```

```
                EngineVersion = engineVersion,
            });
        // Get the entire list using the paginator.
        await foreach (var instanceOptions in
paginateInstanceOptions.OrderableDBInstanceOptions)
        {
            results.Add(instanceOptions);
        }
        return results;
    }



    /// <summary>
    /// Returns a list of DB instances.
    /// </summary>
    /// <param name="dbInstanceIdentifier">Optional name of a specific DB
instance.</param>
    /// <returns>List of DB instances.</returns>
    public async Task<List<DBInstance>> DescribeDBInstances(string
dbInstanceIdentifier = null)
    {
        var results = new List<DBInstance>();
        var instancesPaginator = _amazonRDS.Paginators.DescribeDBInstances(
            new DescribeDBInstancesRequest
            {
                DBInstanceIdentifier = dbInstanceIdentifier
            });
        // Get the entire list using the paginator.
        await foreach (var instances in instancesPaginator.DBInstances)
        {
            results.Add(instances);
        }
        return results;
    }



    /// <summary>
    /// Create an RDS DB instance with a particular set of properties. Use the
action DescribeDBInstancesAsync
    /// to determine when the DB instance is ready to use.
    /// </summary>
    /// <param name="dbName">Name for the DB instance.</param>
```

```
   /// <param name="dbInstanceIdentifier">DB instance identifier.</param>
   /// <param name="parameterGroupName">DB parameter group to associate with the
instance.</param>
   /// <param name="dbEngine">The engine for the DB instance.</param>
   /// <param name="dbEngineVersion">Version for the DB instance.</param>
   /// <param name="instanceClass">Class for the DB instance.</param>
   /// <param name="allocatedStorage">The amount of storage in gibibytes (GiB)
to allocate to the DB instance.</param>
   /// <param name="adminName">Admin user name.</param>
   /// <param name="adminPassword">Admin user password.</param>
   /// <returns>DB instance object.</returns>
   public async Task<DBInstance> CreateDBInstance(string dbName, string
dbInstanceIdentifier,
       string parameterGroupName, string dbEngine, string dbEngineVersion,
       string instanceClass, int allocatedStorage, string adminName, string
adminPassword)
   {
       var response = await _amazonRDS.CreateDBInstanceAsync(
           new CreateDBInstanceRequest()
           {
               DBName = dbName,
               DBInstanceIdentifier = dbInstanceIdentifier,
               DBParameterGroupName = parameterGroupName,
               Engine = dbEngine,
               EngineVersion = dbEngineVersion,
               DBInstanceClass = instanceClass,
               AllocatedStorage = allocatedStorage,
               MasterUsername = adminName,
               MasterUserPassword = adminPassword
           });

       return response.DBInstance;
   }



   /// <summary>
   /// Delete a particular DB instance.
   /// </summary>
   /// <param name="dbInstanceIdentifier">DB instance identifier.</param>
   /// <returns>DB instance object.</returns>
   public async Task<DBInstance> DeleteDBInstance(string dbInstanceIdentifier)
   {
       var response = await _amazonRDS.DeleteDBInstanceAsync(
```

```
            new DeleteDBInstanceRequest()
            {
                DBInstanceIdentifier = dbInstanceIdentifier,
                SkipFinalSnapshot = true,
                DeleteAutomatedBackups = true
            });

        return response.DBInstance;
    }
```

Metode pembungkus yang digunakan oleh skenario untuk grup parameter basis data.

```
/// <summary>
/// Wrapper methods to use Amazon Relational Database Service (Amazon RDS) with
 parameter groups.
/// </summary>
public partial class RDSWrapper
{

    /// <summary>
    /// Get descriptions of DB parameter groups.
    /// </summary>
    /// <param name="name">Optional name of the DB parameter group to describe.</
param>
    /// <returns>The list of DB parameter group descriptions.</returns>
    public async Task<List<DBParameterGroup>> DescribeDBParameterGroups(string
 name = null)
    {
        var response = await _amazonRDS.DescribeDBParameterGroupsAsync(
            new DescribeDBParameterGroupsRequest()
            {
                DBParameterGroupName = name
            });
        return response.DBParameterGroups;
    }



    /// <summary>
```

```
    /// Create a new DB parameter group. Use the action
 DescribeDBParameterGroupsAsync
    /// to determine when the DB parameter group is ready to use.
    /// </summary>
    /// <param name="name">Name of the DB parameter group.</param>
    /// <param name="family">Family of the DB parameter group.</param>
    /// <param name="description">Description of the DB parameter group.</param>
    /// <returns>The new DB parameter group.</returns>
    public async Task<DBParameterGroup> CreateDBParameterGroup(
        string name, string family, string description)
    {
        var response = await _amazonRDS.CreateDBParameterGroupAsync(
            new CreateDBParameterGroupRequest()
            {
                DBParameterGroupName = name,
                DBParameterGroupFamily = family,
                Description = description
            });
        return response.DBParameterGroup;
    }



    /// <summary>
    /// Update a DB parameter group. Use the action
 DescribeDBParameterGroupsAsync
    /// to determine when the DB parameter group is ready to use.
    /// </summary>
    /// <param name="name">Name of the DB parameter group.</param>
    /// <param name="parameters">List of parameters. Maximum of 20 per request.</
param>
    /// <returns>The updated DB parameter group name.</returns>
    public async Task<string> ModifyDBParameterGroup(
        string name, List<Parameter> parameters)
    {
        var response = await _amazonRDS.ModifyDBParameterGroupAsync(
            new ModifyDBParameterGroupRequest()
            {
                DBParameterGroupName = name,
                Parameters = parameters,
            });
        return response.DBParameterGroupName;
    }
```

```
    /// <summary>
    /// Delete a DB parameter group. The group cannot be a default DB parameter
group
    /// or be associated with any DB instances.
    /// </summary>
    /// <param name="name">Name of the DB parameter group.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteDBParameterGroup(string name)
    {
        var response = await _amazonRDS.DeleteDBParameterGroupAsync(
            new DeleteDBParameterGroupRequest()
            {
                DBParameterGroupName = name,
            });
        return response.HttpStatusCode == HttpStatusCode.OK;
    }



    /// <summary>
    /// Get a list of DB parameters from a specific parameter group.
    /// </summary>
    /// <param name="dbParameterGroupName">Name of a specific DB parameter
group.</param>
    /// <param name="source">Optional source for selecting parameters.</param>
    /// <returns>List of parameter values.</returns>
    public async Task<List<Parameter>> DescribeDBParameters(string
dbParameterGroupName, string source = null)
    {
        var results = new List<Parameter>();
        var paginateParameters = _amazonRDS.Paginators.DescribeDBParameters(
            new DescribeDBParametersRequest()
            {
                DBParameterGroupName = dbParameterGroupName,
                Source = source
            });
        // Get the entire list using the paginator.
        await foreach (var parameters in paginateParameters.Parameters)
        {
            results.Add(parameters);
        }
        return results;
```

```
    }
```

Metode pembungkus yang digunakan oleh skenario untuk tindakan cuplikan basis data.

```
/// <summary>
/// Wrapper methods to use Amazon Relational Database Service (Amazon RDS) with
 snapshots.
/// </summary>
public partial class RDSWrapper
{

    /// <summary>
    /// Create a snapshot of a DB instance.
    /// </summary>
    /// <param name="dbInstanceIdentifier">DB instance identifier.</param>
    /// <param name="snapshotIdentifier">Identifier for the snapshot.</param>
    /// <returns>DB snapshot object.</returns>
    public async Task<DBSnapshot> CreateDBSnapshot(string dbInstanceIdentifier,
 string snapshotIdentifier)
    {
        var response = await _amazonRDS.CreateDBSnapshotAsync(
            new CreateDBSnapshotRequest()
            {
                DBSnapshotIdentifier = snapshotIdentifier,
                DBInstanceIdentifier = dbInstanceIdentifier
            });

        return response.DBSnapshot;
    }



    /// <summary>
    /// Return a list of DB snapshots for a particular DB instance.
    /// </summary>
    /// <param name="dbInstanceIdentifier">DB instance identifier.</param>
    /// <returns>List of DB snapshots.</returns>
    public async Task<List<DBSnapshot>> DescribeDBSnapshots(string
 dbInstanceIdentifier)
    {
```

```
        var results = new List<DBSnapshot>();
        var snapshotsPaginator = _amazonRDS.Paginators.DescribeDBSnapshots(
            new DescribeDBSnapshotsRequest()
            {
                DBInstanceIdentifier = dbInstanceIdentifier
            });

        // Get the entire list using the paginator.
        await foreach (var snapshots in snapshotsPaginator.DBSnapshots)
        {
            results.Add(snapshots);
        }
        return results;
    }
```

- Lihat detail API di topik-topik berikut dalam Referensi API AWS SDK for .NET .
  - CreateDBInstance
  - dibuatB ParameterGroup
  - CreateDBSnapshot
  - DeleteDBInstance
  - DihapusB ParameterGroup
  - DijelaskanB EngineVersions
  - DescribeDBInstances
  - DijelaskanB ParameterGroups
  - DescribeDBParameters
  - DescribeDBSnapshots
  - DescribeOrderableDB InstanceOptions
  - ModifyDB ParameterGroup

C++

SDK for C++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

//! Routine which creates an Amazon RDS instance and demonstrates several
 operations
//! on that instance.
/*!
 \sa gettingStartedWithDBInstances()
 \param clientConfiguration: AWS client configuration.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::gettingStartedWithDBInstances(
        const Aws::Client::ClientConfiguration &clientConfig) {
    Aws::RDS::RDSClient client(clientConfig);

    printAsterisksLine();
    std::cout << "Welcome to the Amazon Relational Database Service (Amazon RDS)"
              << std::endl;
    std::cout << "get started with DB instances demo." << std::endl;
    printAsterisksLine();

    std::cout << "Checking for an existing DB parameter group named '" <<
              PARAMETER_GROUP_NAME << "'." << std::endl;
    Aws::String dbParameterGroupFamily("Undefined");
    bool parameterGroupFound = true;
    {
        // 1. Check if the DB parameter group already exists.
        Aws::RDS::Model::DescribeDBParameterGroupsRequest request;
        request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);

        Aws::RDS::Model::DescribeDBParameterGroupsOutcome outcome =
```

```
                client.DescribeDBParameterGroups(request);

        if (outcome.IsSuccess()) {
            std::cout << "DB parameter group named '" <<
                      PARAMETER_GROUP_NAME << "' already exists." << std::endl;
            dbParameterGroupFamily = outcome.GetResult().GetDBParameterGroups()
[0].GetDBParameterGroupFamily();
        }
        else if (outcome.GetError().GetErrorType() ==
                 Aws::RDS::RDSErrors::D_B_PARAMETER_GROUP_NOT_FOUND_FAULT) {
            std::cout << "DB parameter group named '" <<
                      PARAMETER_GROUP_NAME << "' does not exist." << std::endl;
            parameterGroupFound = false;
        }
        else {
            std::cerr << "Error with RDS::DescribeDBParameterGroups. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
    }

    if (!parameterGroupFound) {
        Aws::Vector<Aws::RDS::Model::DBEngineVersion> engineVersions;

        // 2. Get available engine versions for the specified engine.
        if (!getDBEngineVersions(DB_ENGINE, NO_PARAMETER_GROUP_FAMILY,
                                 engineVersions, client)) {
            return false;
        }

        std::cout << "Getting available database engine versions for " <<
 DB_ENGINE
                  << "."
                  << std::endl;
        std::vector<Aws::String> families;
        for (const Aws::RDS::Model::DBEngineVersion &version: engineVersions) {
            Aws::String family = version.GetDBParameterGroupFamily();
            if (std::find(families.begin(), families.end(), family) ==
                families.end()) {
                families.push_back(family);
                std::cout << "  " << families.size() << ": " << family <<
 std::endl;
            }
```

```
        }

        int choice = askQuestionForIntRange("Which family do you want to use? ",
1,
                                            static_cast<int>(families.size()));
        dbParameterGroupFamily = families[choice - 1];
    }
    if (!parameterGroupFound) {
        // 3.  Create a DB parameter group.
        Aws::RDS::Model::CreateDBParameterGroupRequest request;
        request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
        request.SetDBParameterGroupFamily(dbParameterGroupFamily);
        request.SetDescription("Example parameter group.");

        Aws::RDS::Model::CreateDBParameterGroupOutcome outcome =
                client.CreateDBParameterGroup(request);

        if (outcome.IsSuccess()) {
            std::cout << "The DB parameter group was successfully created."
                      << std::endl;
        }
        else {
            std::cerr << "Error with RDS::CreateDBParameterGroup. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
    }

    printAsterisksLine();
    std::cout << "Let's set some parameter values in your parameter group."
              << std::endl;

    Aws::String marker;
    Aws::Vector<Aws::RDS::Model::Parameter> autoIncrementParameters;
    // 4.  Get the parameters in the DB parameter group.
    if (!getDBParameters(PARAMETER_GROUP_NAME, AUTO_INCREMENT_PREFIX, NO_SOURCE,
                         autoIncrementParameters,
                         client)) {
        cleanUpResources(PARAMETER_GROUP_NAME, "", client);
        return false;
    }

    Aws::Vector<Aws::RDS::Model::Parameter> updateParameters;
```

```
    for (Aws::RDS::Model::Parameter &autoIncParameter: autoIncrementParameters) {
        if (autoIncParameter.GetIsModifiable() &&
            (autoIncParameter.GetDataType() == "integer")) {
            std::cout << "The " << autoIncParameter.GetParameterName()
                      << " is described as: " <<
                      autoIncParameter.GetDescription() << "." << std::endl;
            if (autoIncParameter.ParameterValueHasBeenSet()) {
                std::cout << "The current value is "
                          << autoIncParameter.GetParameterValue()
                          << "." << std::endl;
            }
            std::vector<int> splitValues = splitToInts(
                    autoIncParameter.GetAllowedValues(), '-');
            if (splitValues.size() == 2) {
                int newValue = askQuestionForIntRange(
                        Aws::String("Enter a new value in the range ") +
                        autoIncParameter.GetAllowedValues() + ": ",
                        splitValues[0], splitValues[1]);
                autoIncParameter.SetParameterValue(std::to_string(newValue));
                updateParameters.push_back(autoIncParameter);

            }
            else {
                std::cerr << "Error parsing " <<
autoIncParameter.GetAllowedValues()
                          << std::endl;
            }
        }
    }

    {
        // 5.  Modify the auto increment parameters in the group.
        Aws::RDS::Model::ModifyDBParameterGroupRequest request;
        request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
        request.SetParameters(updateParameters);

        Aws::RDS::Model::ModifyDBParameterGroupOutcome outcome =
                client.ModifyDBParameterGroup(request);

        if (outcome.IsSuccess()) {
            std::cout << "The DB parameter group was successfully modified."
                      << std::endl;
        }
```

```
        else {
            std::cerr << "Error with RDS::ModifyDBParameterGroup. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
        }
    }

    std::cout
            << "You can get a list of parameters you've set by specifying a
source of 'user'."
            << std::endl;

    Aws::Vector<Aws::RDS::Model::Parameter> userParameters;
    // 6.  Display the modified parameters in the group.
    if (!getDBParameters(PARAMETER_GROUP_NAME, NO_NAME_PREFIX, "user",
userParameters,
                         client)) {
        cleanUpResources(PARAMETER_GROUP_NAME, "", client);
        return false;
    }

    for (const auto &userParameter: userParameters) {
        std::cout << "  " << userParameter.GetParameterName() << ", " <<
                  userParameter.GetDescription() << ", parameter value - "
                  << userParameter.GetParameterValue() << std::endl;
    }

    printAsterisksLine();
    std::cout << "Checking for an existing DB instance." << std::endl;

    Aws::RDS::Model::DBInstance dbInstance;
    // 7.  Check if the DB instance already exists.
    if (!describeDBInstance(DB_INSTANCE_IDENTIFIER, dbInstance, client)) {
        cleanUpResources(PARAMETER_GROUP_NAME, "", client);
        return false;
    }

    if (dbInstance.DbInstancePortHasBeenSet()) {
        std::cout << "The DB instance already exists." << std::endl;
    }
    else {
        std::cout << "Let's create a DB instance." << std::endl;
        const Aws::String administratorName = askQuestion(
                "Enter an administrator username for the database: ");
```

```
        const Aws::String administratorPassword = askQuestion(
                "Enter a password for the administrator (at least 8 characters):
");
        Aws::Vector<Aws::RDS::Model::DBEngineVersion> engineVersions;

        // 8.  Get a list of available engine versions.
        if (!getDBEngineVersions(DB_ENGINE, dbParameterGroupFamily,
engineVersions,
                                 client)) {
            cleanUpResources(PARAMETER_GROUP_NAME, "", client);
            return false;
        }

        std::cout << "The available engines for your parameter group are:" <<
std::endl;

        int index = 1;
        for (const Aws::RDS::Model::DBEngineVersion &engineVersion:
engineVersions) {
            std::cout << "  " << index << ": " <<
engineVersion.GetEngineVersion()
                      << std::endl;
            ++index;
        }
        int choice = askQuestionForIntRange("Which engine do you want to use? ",
1,

static_cast<int>(engineVersions.size()));
        const Aws::RDS::Model::DBEngineVersion engineVersion =
engineVersions[choice -
                                                                              1];

        Aws::String dbInstanceClass;
        // 9.  Get a list of micro instance classes.
        if (!chooseMicroDBInstanceClass(engineVersion.GetEngine(),
                                        engineVersion.GetEngineVersion(),
                                        dbInstanceClass,
                                        client)) {
            cleanUpResources(PARAMETER_GROUP_NAME, "", client);
            return false;
        }

        std::cout << "Creating a DB instance named '" << DB_INSTANCE_IDENTIFIER
                  << "' and database '" << DB_NAME << "'.\n"
```

```
                << "The DB instance is configured to use your custom parameter
group '"
                << PARAMETER_GROUP_NAME << "',\n"
                << "selected engine version " <<
engineVersion.GetEngineVersion()
                << ",\n"
                << "selected DB instance class '" << dbInstanceClass << "',"
                << " and " << DB_ALLOCATED_STORAGE << " GiB of " <<
DB_STORAGE_TYPE
                << " storage.\nThis typically takes several minutes." <<
std::endl;

       Aws::RDS::Model::CreateDBInstanceRequest request;
       request.SetDBName(DB_NAME);
       request.SetDBInstanceIdentifier(DB_INSTANCE_IDENTIFIER);
       request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
       request.SetEngine(engineVersion.GetEngine());
       request.SetEngineVersion(engineVersion.GetEngineVersion());
       request.SetDBInstanceClass(dbInstanceClass);
       request.SetStorageType(DB_STORAGE_TYPE);
       request.SetAllocatedStorage(DB_ALLOCATED_STORAGE);
       request.SetMasterUsername(administratorName);
       request.SetMasterUserPassword(administratorPassword);

       Aws::RDS::Model::CreateDBInstanceOutcome outcome =
               client.CreateDBInstance(request);

       if (outcome.IsSuccess()) {
           std::cout << "The DB instance creation has started."
                     << std::endl;
       }
       else {
           std::cerr << "Error with RDS::CreateDBInstance. "
                     << outcome.GetError().GetMessage()
                     << std::endl;
           cleanUpResources(PARAMETER_GROUP_NAME, "", client);
           return false;
       }
   }

   std::cout << "Waiting for the DB instance to become available." << std::endl;

   int counter = 0;
   // 11. Wait for the DB instance to become available.
```

```
    do {
        std::this_thread::sleep_for(std::chrono::seconds(1));
        ++counter;
        if (counter > 900) {
            std::cerr << "Wait for instance to become available timed out ofter "
                      << counter
                      << " seconds." << std::endl;
            cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
client);
            return false;
        }

        dbInstance = Aws::RDS::Model::DBInstance();
        if (!describeDBInstance(DB_INSTANCE_IDENTIFIER, dbInstance, client)) {
            cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
client);
            return false;
        }

        if ((counter % 20) == 0) {
            std::cout << "Current DB instance status is '"
                      << dbInstance.GetDBInstanceStatus()
                      << "' after " << counter << " seconds." << std::endl;
        }
    } while (dbInstance.GetDBInstanceStatus() != "available");

    if (dbInstance.GetDBInstanceStatus() == "available") {
        std::cout << "The DB instance has been created." << std::endl;
    }

    printAsterisksLine();

    // 12. Display the connection string that can be used to connect a 'mysql'
shell to the database.
    displayConnection(dbInstance);

    printAsterisksLine();

    if (askYesNoQuestion(
            "Do you want to create a snapshot of your DB instance (y/n)? ")) {
        Aws::String snapshotID(DB_INSTANCE_IDENTIFIER + "-" +
                               Aws::String(Aws::Utils::UUID::RandomUUID()));
        {
```

```
            std::cout << "Creating a snapshot named " << snapshotID << "." <<
std::endl;
            std::cout << "This typically takes a few minutes." << std::endl;

            // 13. Create a snapshot of the DB instance.
            Aws::RDS::Model::CreateDBSnapshotRequest request;
            request.SetDBInstanceIdentifier(DB_INSTANCE_IDENTIFIER);
            request.SetDBSnapshotIdentifier(snapshotID);

            Aws::RDS::Model::CreateDBSnapshotOutcome outcome =
                    client.CreateDBSnapshot(request);

            if (outcome.IsSuccess()) {
                std::cout << "Snapshot creation has started."
                          << std::endl;
            }
            else {
                std::cerr << "Error with RDS::CreateDBSnapshot. "
                          << outcome.GetError().GetMessage()
                          << std::endl;
                cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
client);
                return false;
            }
        }

        std::cout << "Waiting for snapshot to become available." << std::endl;

        Aws::RDS::Model::DBSnapshot snapshot;
        counter = 0;
        do {
            std::this_thread::sleep_for(std::chrono::seconds(1));
            ++counter;
            if (counter > 600) {
                std::cerr << "Wait for snapshot to be available timed out ofter "
                          << counter
                          << " seconds." << std::endl;
                cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
client);
                return false;
            }

            // 14. Wait for the snapshot to become available.
            Aws::RDS::Model::DescribeDBSnapshotsRequest request;
```

```
            request.SetDBSnapshotIdentifier(snapshotID);

            Aws::RDS::Model::DescribeDBSnapshotsOutcome outcome =
                    client.DescribeDBSnapshots(request);

            if (outcome.IsSuccess()) {
                snapshot = outcome.GetResult().GetDBSnapshots()[0];
            }
            else {
                std::cerr << "Error with RDS::DescribeDBSnapshots. "
                          << outcome.GetError().GetMessage()
                          << std::endl;
                cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
 client);
                return false;
            }

            if ((counter % 20) == 0) {
                std::cout << "Current snapshot status is '"
                          << snapshot.GetStatus()
                          << "' after " << counter << " seconds." << std::endl;
            }
        } while (snapshot.GetStatus() != "available");

        if (snapshot.GetStatus() != "available") {
            std::cout << "A snapshot has been created." << std::endl;
        }
    }

    printAsterisksLine();

    bool result = true;
    if (askYesNoQuestion(
            "Do you want to delete the DB instance and parameter group (y/n)? "))
 {
        result = cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
 client);
    }

    return result;
}


//! Routine which gets DB parameters using the 'DescribeDBParameters' api.
```

```
/*!
 \sa getDBParameters()
 \param parameterGroupName: The name of the parameter group.
 \param namePrefix: Prefix string to filter results by parameter name.
 \param source: A source such as 'user', ignored if empty.
 \param parametersResult: Vector of 'Parameter' objects returned by the routine.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::getDBParameters(const Aws::String &parameterGroupName,
                                  const Aws::String &namePrefix,
                                  const Aws::String &source,
                                  Aws::Vector<Aws::RDS::Model::Parameter>
 &parametersResult,
                                  const Aws::RDS::RDSClient &client) {
    Aws::String marker;
    do {
        Aws::RDS::Model::DescribeDBParametersRequest request;
        request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
        if (!marker.empty()) {
            request.SetMarker(marker);
        }
        if (!source.empty()) {
            request.SetSource(source);
        }

        Aws::RDS::Model::DescribeDBParametersOutcome outcome =
                client.DescribeDBParameters(request);

        if (outcome.IsSuccess()) {
            const Aws::Vector<Aws::RDS::Model::Parameter> &parameters =
                    outcome.GetResult().GetParameters();
            for (const Aws::RDS::Model::Parameter &parameter: parameters) {
                if (!namePrefix.empty()) {
                    if (parameter.GetParameterName().find(namePrefix) == 0) {
                        parametersResult.push_back(parameter);
                    }
                }
                else {
                    parametersResult.push_back(parameter);
                }
            }

            marker = outcome.GetResult().GetMarker();
```

```
        }
        else {
            std::cerr << "Error with RDS::DescribeDBParameters. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
    } while (!marker.empty());

    return true;
}


//! Routine which gets available DB engine versions for an engine name and
//! an optional parameter group family.
/*!
 \sa getDBEngineVersions()
 \param engineName: A DB engine name.
 \param parameterGroupFamily: A parameter group family name, ignored if empty.
 \param engineVersionsResult: Vector of 'DBEngineVersion' objects returned by the
 routine.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::getDBEngineVersions(const Aws::String &engineName,
                                      const Aws::String &parameterGroupFamily,

 Aws::Vector<Aws::RDS::Model::DBEngineVersion> &engineVersionsResult,
                                      const Aws::RDS::RDSClient &client) {
    Aws::RDS::Model::DescribeDBEngineVersionsRequest request;
    request.SetEngine(engineName);
    if (!parameterGroupFamily.empty()) {
        request.SetDBParameterGroupFamily(parameterGroupFamily);
    }

    Aws::RDS::Model::DescribeDBEngineVersionsOutcome outcome =
            client.DescribeDBEngineVersions(request);

    if (outcome.IsSuccess()) {
        engineVersionsResult = outcome.GetResult().GetDBEngineVersions();
    }
    else {
        std::cerr << "Error with RDS::DescribeDBEngineVersionsRequest. "
                  << outcome.GetError().GetMessage()
```

```
                  << std::endl;
    }

    return outcome.IsSuccess();
}


//! Routine which gets a DB instance description.
/*!
 \sa describeDBInstance()
 \param dbInstanceIdentifier: A DB instance identifier.
 \param instanceResult: The 'DBInstance' object containing the description.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::describeDBInstance(const Aws::String &dbInstanceIdentifier,
                                     Aws::RDS::Model::DBInstance &instanceResult,
                                     const Aws::RDS::RDSClient &client) {
    Aws::RDS::Model::DescribeDBInstancesRequest request;
    request.SetDBInstanceIdentifier(dbInstanceIdentifier);

    Aws::RDS::Model::DescribeDBInstancesOutcome outcome =
            client.DescribeDBInstances(request);

    bool result = true;
    if (outcome.IsSuccess()) {
        instanceResult = outcome.GetResult().GetDBInstances()[0];
    }
    else if (outcome.GetError().GetErrorType() !=
             Aws::RDS::RDSErrors::D_B_INSTANCE_NOT_FOUND_FAULT) {
        result = false;
        std::cerr << "Error with RDS::DescribeDBInstances. "
                  << outcome.GetError().GetMessage()
                  << std::endl;
    }
        // This example does not log an error if the DB instance does not exist.
        // Instead, instanceResult is set to empty.
    else {
        instanceResult = Aws::RDS::Model::DBInstance();
    }

    return result;
}
```

```
//! Routine which gets available 'micro' DB instance classes, displays the list
//! to the user, and returns the user selection.
/*!
 \sa chooseMicroDBInstanceClass()
 \param engineName: The DB engine name.
 \param engineVersion: The DB engine version.
 \param dbInstanceClass: String for DB instance class chosen by the user.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::chooseMicroDBInstanceClass(const Aws::String &engine,
                                             const Aws::String &engineVersion,
                                             Aws::String &dbInstanceClass,
                                             const Aws::RDS::RDSClient &client) {
    std::vector<Aws::String> instanceClasses;
    Aws::String marker;
    do {
        Aws::RDS::Model::DescribeOrderableDBInstanceOptionsRequest request;
        request.SetEngine(engine);
        request.SetEngineVersion(engineVersion);
        if (!marker.empty()) {
            request.SetMarker(marker);
        }

        Aws::RDS::Model::DescribeOrderableDBInstanceOptionsOutcome outcome =
                client.DescribeOrderableDBInstanceOptions(request);

        if (outcome.IsSuccess()) {
            const Aws::Vector<Aws::RDS::Model::OrderableDBInstanceOption>
 &options =
                    outcome.GetResult().GetOrderableDBInstanceOptions();
            for (const Aws::RDS::Model::OrderableDBInstanceOption &option:
 options) {
                const Aws::String &instanceClass = option.GetDBInstanceClass();
                if (instanceClass.find("micro") != std::string::npos) {
                    if (std::find(instanceClasses.begin(), instanceClasses.end(),
                                  instanceClass) ==
                        instanceClasses.end()) {
                        instanceClasses.push_back(instanceClass);
                    }
                }
            }
            marker = outcome.GetResult().GetMarker();
```

```
        }
        else {
            std::cerr << "Error with RDS::DescribeOrderableDBInstanceOptions. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
    } while (!marker.empty());

    std::cout << "The available micro DB instance classes for your database
 engine are:"
              << std::endl;
    for (int i = 0; i < instanceClasses.size(); ++i) {
        std::cout << "   " << i + 1 << ": " << instanceClasses[i] << std::endl;
    }

    int choice = askQuestionForIntRange(
            "Which micro DB instance class do you want to use? ",
            1, static_cast<int>(instanceClasses.size()));
    dbInstanceClass = instanceClasses[choice - 1];
    return true;
}

//! Routine which deletes resources created by the scenario.
/*!
\sa cleanUpResources()
\param parameterGroupName: A parameter group name, this may be empty.
\param dbInstanceIdentifier: A DB instance identifier, this may be empty.
\param client: 'RDSClient' instance.
\return bool: Successful completion.
*/
bool AwsDoc::RDS::cleanUpResources(const Aws::String &parameterGroupName,
                                   const Aws::String &dbInstanceIdentifier,
                                   const Aws::RDS::RDSClient &client) {
    bool result = true;
    if (!dbInstanceIdentifier.empty()) {
        {
            // 15. Delete the DB instance.
            Aws::RDS::Model::DeleteDBInstanceRequest request;
            request.SetDBInstanceIdentifier(dbInstanceIdentifier);
            request.SetSkipFinalSnapshot(true);
            request.SetDeleteAutomatedBackups(true);

            Aws::RDS::Model::DeleteDBInstanceOutcome outcome =
```

```
                    client.DeleteDBInstance(request);

            if (outcome.IsSuccess()) {
                std::cout << "DB instance deletion has started."
                          << std::endl;
            }
            else {
                std::cerr << "Error with RDS::DeleteDBInstance. "
                          << outcome.GetError().GetMessage()
                          << std::endl;
                result = false;
            }
        }

        std::cout
                << "Waiting for DB instance to delete before deleting the
parameter group."
                << std::endl;
        std::cout << "This may take a while." << std::endl;

        int counter = 0;
        Aws::RDS::Model::DBInstance dbInstance;
        do {
            std::this_thread::sleep_for(std::chrono::seconds(1));
            ++counter;
            if (counter > 800) {
                std::cerr << "Wait for instance to delete timed out ofter " <<
counter
                          << " seconds." << std::endl;
                return false;
            }

            dbInstance = Aws::RDS::Model::DBInstance();
            // 16. Wait for the DB instance to be deleted.
            if (!describeDBInstance(dbInstanceIdentifier, dbInstance, client)) {
                return false;
            }

            if (dbInstance.DBInstanceIdentifierHasBeenSet() && (counter % 20) ==
0) {
                std::cout << "Current DB instance status is '"
                          << dbInstance.GetDBInstanceStatus()
                          << "' after " << counter << " seconds." << std::endl;
            }
```

```
        } while (dbInstance.DBInstanceIdentifierHasBeenSet());
    }

    if (!parameterGroupName.empty()) {
        // 17. Delete the parameter group.
        Aws::RDS::Model::DeleteDBParameterGroupRequest request;
        request.SetDBParameterGroupName(parameterGroupName);

        Aws::RDS::Model::DeleteDBParameterGroupOutcome outcome =
                client.DeleteDBParameterGroup(request);

        if (outcome.IsSuccess()) {
            std::cout << "The DB parameter group was successfully deleted."
                      << std::endl;
        }
        else {
            std::cerr << "Error with RDS::DeleteDBParameterGroup. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            result = false;
        }
    }

    return result;
}
```

- Lihat detail API di topik-topik berikut dalam Referensi API AWS SDK for C++ .
  - CreateDBInstance
  - dibuatB ParameterGroup
  - CreateDBSnapshot
  - DeleteDBInstance
  - DihapusB ParameterGroup
  - DijelaskanB EngineVersions
  - DescribeDBInstances
  - DijelaskanB ParameterGroups
  - DescribeDBParameters
  - DescribeDBSnapshots
  - DescribeOrderableDB InstanceOptions

- ModifyDB ParameterGroup

Go

SDK for Go V2

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

Jalankan skenario interaktif di penggugah/prompt perintah.

```
// GetStartedInstances is an interactive example that shows you how to use the
 AWS SDK for Go
// with Amazon Relation Database Service (Amazon RDS) to do the following:
//
//  1. Create a custom DB parameter group and set parameter values.
//  2. Create a DB instance that is configured to use the parameter group. The DB
 instance
//     also contains a database.
//  3. Take a snapshot of the DB instance.
//  4. Delete the DB instance and parameter group.
type GetStartedInstances struct {
 sdkConfig  aws.Config
 instances  actions.DbInstances
 questioner demotools.IQuestioner
 helper     IScenarioHelper
 isTestRun  bool
}

// NewGetStartedInstances constructs a GetStartedInstances instance from a
 configuration.
// It uses the specified config to get an Amazon RDS
// client and create wrappers for the actions used in the scenario.
func NewGetStartedInstances(sdkConfig aws.Config, questioner
 demotools.IQuestioner,
 helper IScenarioHelper) GetStartedInstances {
 rdsClient := rds.NewFromConfig(sdkConfig)
 return GetStartedInstances{
```

```
  sdkConfig:  sdkConfig,
  instances:  actions.DbInstances{RdsClient: rdsClient},
  questioner: questioner,
  helper:     helper,
 }
}

// Run runs the interactive scenario.
func (scenario GetStartedInstances) Run(dbEngine string, parameterGroupName
 string,
 instanceName string, dbName string) {
 defer func() {
  if r := recover(); r != nil {
   log.Println("Something went wrong with the demo.")
  }
 }()

 log.Println(strings.Repeat("-", 88))
 log.Println("Welcome to the Amazon Relational Database Service (Amazon RDS) DB
 Instance demo.")
 log.Println(strings.Repeat("-", 88))

 parameterGroup := scenario.CreateParameterGroup(dbEngine, parameterGroupName)
 scenario.SetUserParameters(parameterGroupName)
 instance := scenario.CreateInstance(instanceName, dbEngine, dbName,
 parameterGroup)
 scenario.DisplayConnection(instance)
 scenario.CreateSnapshot(instance)
 scenario.Cleanup(instance, parameterGroup)

 log.Println(strings.Repeat("-", 88))
 log.Println("Thanks for watching!")
 log.Println(strings.Repeat("-", 88))
}

// CreateParameterGroup shows how to get available engine versions for a
 specified
// database engine and create a DB parameter group that is compatible with a
// selected engine family.
func (scenario GetStartedInstances) CreateParameterGroup(dbEngine string,
 parameterGroupName string) *types.DBParameterGroup {

 log.Printf("Checking for an existing DB parameter group named %v.\n",
  parameterGroupName)
```

```
 parameterGroup, err := scenario.instances.GetParameterGroup(parameterGroupName)
 if err != nil {
  panic(err)
 }
 if parameterGroup == nil {
  log.Printf("Getting available database engine versions for %v.\n", dbEngine)
  engineVersions, err := scenario.instances.GetEngineVersions(dbEngine, "")
  if err != nil {
   panic(err)
  }

  familySet := map[string]struct{}{}
  for _, family := range engineVersions {
   familySet[*family.DBParameterGroupFamily] = struct{}{}
  }
  var families []string
  for family := range familySet {
   families = append(families, family)
  }
  sort.Strings(families)
  familyIndex := scenario.questioner.AskChoice("Which family do you want to use?
\n", families)
  log.Println("Creating a DB parameter group.")
  _, err = scenario.instances.CreateParameterGroup(
   parameterGroupName, families[familyIndex], "Example parameter group.")
  if err != nil {
   panic(err)
  }
  parameterGroup, err = scenario.instances.GetParameterGroup(parameterGroupName)
  if err != nil {
   panic(err)
  }
 }
 log.Printf("Parameter group %v:\n", *parameterGroup.DBParameterGroupFamily)
 log.Printf("\tName: %v\n", *parameterGroup.DBParameterGroupName)
 log.Printf("\tARN: %v\n", *parameterGroup.DBParameterGroupArn)
 log.Printf("\tFamily: %v\n", *parameterGroup.DBParameterGroupFamily)
 log.Printf("\tDescription: %v\n", *parameterGroup.Description)
 log.Println(strings.Repeat("-", 88))
 return parameterGroup
}

// SetUserParameters shows how to get the parameters contained in a custom
 parameter
```

```
// group and update some of the parameter values in the group.
func (scenario GetStartedInstances) SetUserParameters(parameterGroupName string)
 {
 log.Println("Let's set some parameter values in your parameter group.")
 dbParameters, err := scenario.instances.GetParameters(parameterGroupName, "")
 if err != nil {
  panic(err)
 }
 var updateParams []types.Parameter
 for _, dbParam := range dbParameters {
  if strings.HasPrefix(*dbParam.ParameterName, "auto_increment") &&
   dbParam.IsModifiable && *dbParam.DataType == "integer" {
   log.Printf("The %v parameter is described as:\n\t%v",
    *dbParam.ParameterName, *dbParam.Description)
   rangeSplit := strings.Split(*dbParam.AllowedValues, "-")
   lower, _ := strconv.Atoi(rangeSplit[0])
   upper, _ := strconv.Atoi(rangeSplit[1])
   newValue := scenario.questioner.AskInt(
    fmt.Sprintf("Enter a value between %v and %v:", lower, upper),
    demotools.InIntRange{Lower: lower, Upper: upper})
   dbParam.ParameterValue = aws.String(strconv.Itoa(newValue))
   updateParams = append(updateParams, dbParam)
  }
 }
 err = scenario.instances.UpdateParameters(parameterGroupName, updateParams)
 if err != nil {
  panic(err)
 }
 log.Println("To get a list of parameters that you set previously, specify a
 source of 'user'.")
 userParameters, err := scenario.instances.GetParameters(parameterGroupName,
 "user")
 if err != nil {
  panic(err)
 }
 log.Println("Here are the parameters you set:")
 for _, param := range userParameters {
  log.Printf("\t%v: %v\n", *param.ParameterName, *param.ParameterValue)
 }
 log.Println(strings.Repeat("-", 88))
}

// CreateInstance shows how to create a DB instance that contains a database of a
```

```
// specified type. The database is also configured to use a custom DB parameter
 group.
func (scenario GetStartedInstances) CreateInstance(instanceName string, dbEngine
 string,
 dbName string, parameterGroup *types.DBParameterGroup) *types.DBInstance {

 log.Println("Checking for an existing DB instance.")
 instance, err := scenario.instances.GetInstance(instanceName)
 if err != nil {
  panic(err)
 }
 if instance == nil {
  adminUsername := scenario.questioner.Ask(
   "Enter an administrator username for the database: ", demotools.NotEmpty{})
  adminPassword := scenario.questioner.AskPassword(
   "Enter a password for the administrator (at least 8 characters): ", 7)
  engineVersions, err := scenario.instances.GetEngineVersions(dbEngine,
   *parameterGroup.DBParameterGroupFamily)
  if err != nil {
   panic(err)
  }
  var engineChoices []string
  for _, engine := range engineVersions {
   engineChoices = append(engineChoices, *engine.EngineVersion)
  }
  engineIndex := scenario.questioner.AskChoice(
   "The available engines for your parameter group are:\n", engineChoices)
  engineSelection := engineVersions[engineIndex]
  instOpts, err :=
 scenario.instances.GetOrderableInstances(*engineSelection.Engine,
   *engineSelection.EngineVersion)
  if err != nil {
   panic(err)
  }
  optSet := map[string]struct{}{}
  for _, opt := range instOpts {
   if strings.Contains(*opt.DBInstanceClass, "micro") {
    optSet[*opt.DBInstanceClass] = struct{}{}
   }
  }
  var optChoices []string
  for opt := range optSet {
   optChoices = append(optChoices, opt)
  }
```

```
  sort.Strings(optChoices)
  optIndex := scenario.questioner.AskChoice(
   "The available micro DB instance classes for your database engine are:\n",
 optChoices)
  storageType := "standard"
  allocatedStorage := int32(5)
  log.Printf("Creating a DB instance named %v and database %v.\n"+
   "The DB instance is configured to use your custom parameter group %v,\n"+
   "selected engine %v,\n"+
   "selected DB instance class %v,"+
   "and %v GiB of %v storage.\n"+
   "This typically takes several minutes.",
   instanceName, dbName, *parameterGroup.DBParameterGroupName,
 *engineSelection.EngineVersion,
   optChoices[optIndex], allocatedStorage, storageType)
  instance, err = scenario.instances.CreateInstance(
   instanceName, dbName, *engineSelection.Engine, *engineSelection.EngineVersion,
   *parameterGroup.DBParameterGroupName, optChoices[optIndex], storageType,
   allocatedStorage, adminUsername, adminPassword)
  if err != nil {
   panic(err)
  }
  for *instance.DBInstanceStatus != "available" {
   scenario.helper.Pause(30)
   instance, err = scenario.instances.GetInstance(instanceName)
   if err != nil {
    panic(err)
   }
  }
  log.Println("Instance created and available.")
 }
 log.Println("Instance data:")
 log.Printf("\tDBInstanceIdentifier: %v\n", *instance.DBInstanceIdentifier)
 log.Printf("\tARN: %v\n", *instance.DBInstanceArn)
 log.Printf("\tStatus: %v\n", *instance.DBInstanceStatus)
 log.Printf("\tEngine: %v\n", *instance.Engine)
 log.Printf("\tEngine version: %v\n", *instance.EngineVersion)
 log.Println(strings.Repeat("-", 88))
 return instance
}

// DisplayConnection displays connection information about a DB instance and tips
// on how to connect to it.
```

```
func (scenario GetStartedInstances) DisplayConnection(instance *types.DBInstance)
 {
 log.Println(
  "You can now connect to your database by using your favorite MySQL client.\n" +
   "One way to connect is by using the 'mysql' shell on an Amazon EC2 instance\n"
 +
   "that is running in the same VPC as your DB instance. Pass the endpoint,\n" +
   "port, and administrator username to 'mysql'. Then, enter your password\n" +
   "when prompted:")
 log.Printf("\n\tmysql -h %v -P %v -u %v -p\n",
  *instance.Endpoint.Address, instance.Endpoint.Port, *instance.MasterUsername)
 log.Println("For more information, see the User Guide for RDS:\n" +
  "\thttps://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
CHAP_GettingStarted.CreatingConnecting.MySQL.html#CHAP_GettingStarted.Connecting.MySQL")
 log.Println(strings.Repeat("-", 88))
}

// CreateSnapshot shows how to create a DB instance snapshot and wait until it's
 available.
func (scenario GetStartedInstances) CreateSnapshot(instance *types.DBInstance) {
 if scenario.questioner.AskBool(
  "Do you want to create a snapshot of your DB instance (y/n)? ", "y") {
  snapshotId := fmt.Sprintf("%v-%v", *instance.DBInstanceIdentifier,
 scenario.helper.UniqueId())
  log.Printf("Creating a snapshot named %v. This typically takes a few minutes.
\n", snapshotId)
  snapshot, err :=
 scenario.instances.CreateSnapshot(*instance.DBInstanceIdentifier, snapshotId)
  if err != nil {
   panic(err)
  }
  for *snapshot.Status != "available" {
   scenario.helper.Pause(30)
   snapshot, err = scenario.instances.GetSnapshot(snapshotId)
   if err != nil {
    panic(err)
   }
  }
  log.Println("Snapshot data:")
  log.Printf("\tDBSnapshotIdentifier: %v\n", *snapshot.DBSnapshotIdentifier)
  log.Printf("\tARN: %v\n", *snapshot.DBSnapshotArn)
  log.Printf("\tStatus: %v\n", *snapshot.Status)
  log.Printf("\tEngine: %v\n", *snapshot.Engine)
  log.Printf("\tEngine version: %v\n", *snapshot.EngineVersion)
```

```
  log.Printf("\tDBInstanceIdentifier: %v\n", *snapshot.DBInstanceIdentifier)
  log.Printf("\tSnapshotCreateTime: %v\n", *snapshot.SnapshotCreateTime)
  log.Println(strings.Repeat("-", 88))
 }
}

// Cleanup shows how to clean up a DB instance and DB parameter group.
// Before the DB parameter group can be deleted, all associated DB instances must
 first be deleted.
func (scenario GetStartedInstances) Cleanup(
 instance *types.DBInstance, parameterGroup *types.DBParameterGroup) {

 if scenario.questioner.AskBool(
  "\nDo you want to delete the database instance and parameter group (y/n)? ",
 "y") {
  log.Printf("Deleting database instance %v.\n", *instance.DBInstanceIdentifier)
  err := scenario.instances.DeleteInstance(*instance.DBInstanceIdentifier)
  if err != nil {
   panic(err)
  }
  log.Println(
   "Waiting for the DB instance to delete. This typically takes several
 minutes.")
  for instance != nil {
   scenario.helper.Pause(30)
   instance, err = scenario.instances.GetInstance(*instance.DBInstanceIdentifier)
   if err != nil {
    panic(err)
   }
  }
  log.Printf("Deleting parameter group %v.",
 *parameterGroup.DBParameterGroupName)
  err =
 scenario.instances.DeleteParameterGroup(*parameterGroup.DBParameterGroupName)
  if err != nil {
   panic(err)
  }
 }
}
```

Tentukan fungsi-fungsi yang dipanggil oleh skenario untuk mengelola tindakan Amazon RDS.

```
type DbInstances struct {
 RdsClient *rds.Client
}


// GetParameterGroup gets a DB parameter group by name.
func (instances *DbInstances) GetParameterGroup(parameterGroupName string) (
 *types.DBParameterGroup, error) {
 output, err := instances.RdsClient.DescribeDBParameterGroups(
  context.TODO(), &rds.DescribeDBParameterGroupsInput{
   DBParameterGroupName: aws.String(parameterGroupName),
  })
 if err != nil {
  var notFoundError *types.DBParameterGroupNotFoundFault
  if errors.As(err, &notFoundError) {
   log.Printf("Parameter group %v does not exist.\n", parameterGroupName)
   err = nil
  } else {
   log.Printf("Error getting parameter group %v: %v\n", parameterGroupName, err)
  }
  return nil, err
 } else {
  return &output.DBParameterGroups[0], err
 }
}



// CreateParameterGroup creates a DB parameter group that is based on the
 specified
// parameter group family.
func (instances *DbInstances) CreateParameterGroup(
 parameterGroupName string, parameterGroupFamily string, description string) (
 *types.DBParameterGroup, error) {

 output, err := instances.RdsClient.CreateDBParameterGroup(context.TODO(),
  &rds.CreateDBParameterGroupInput{
   DBParameterGroupName:   aws.String(parameterGroupName),
   DBParameterGroupFamily: aws.String(parameterGroupFamily),
   Description:            aws.String(description),
  })
 if err != nil {
```

```
  log.Printf("Couldn't create parameter group %v: %v\n", parameterGroupName, err)
  return nil, err
 } else {
  return output.DBParameterGroup, err
 }
}



// DeleteParameterGroup deletes the named DB parameter group.
func (instances *DbInstances) DeleteParameterGroup(parameterGroupName string)
 error {
 _, err := instances.RdsClient.DeleteDBParameterGroup(context.TODO(),
  &rds.DeleteDBParameterGroupInput{
   DBParameterGroupName: aws.String(parameterGroupName),
  })
 if err != nil {
  log.Printf("Couldn't delete parameter group %v: %v\n", parameterGroupName, err)
  return err
 } else {
  return nil
 }
}



// GetParameters gets the parameters that are contained in a DB parameter group.
func (instances *DbInstances) GetParameters(parameterGroupName string, source
 string) (
 []types.Parameter, error) {

 var output *rds.DescribeDBParametersOutput
 var params []types.Parameter
 var err error
 parameterPaginator := rds.NewDescribeDBParametersPaginator(instances.RdsClient,
  &rds.DescribeDBParametersInput{
   DBParameterGroupName: aws.String(parameterGroupName),
   Source:               aws.String(source),
  })
 for parameterPaginator.HasMorePages() {
  output, err = parameterPaginator.NextPage(context.TODO())
  if err != nil {
   log.Printf("Couldn't get parameters for %v: %v\n", parameterGroupName, err)
   break
```

```
  } else {
   params = append(params, output.Parameters...)
  }
 }
 return params, err
}



// UpdateParameters updates parameters in a named DB parameter group.
func (instances *DbInstances) UpdateParameters(parameterGroupName string, params
 []types.Parameter) error {
 _, err := instances.RdsClient.ModifyDBParameterGroup(context.TODO(),
  &rds.ModifyDBParameterGroupInput{
   DBParameterGroupName: aws.String(parameterGroupName),
   Parameters:           params,
  })
 if err != nil {
  log.Printf("Couldn't update parameters in %v: %v\n", parameterGroupName, err)
  return err
 } else {
  return nil
 }
}



// CreateSnapshot creates a snapshot of a DB instance.
func (instances *DbInstances) CreateSnapshot(instanceName string, snapshotName
 string) (
 *types.DBSnapshot, error) {
 output, err := instances.RdsClient.CreateDBSnapshot(context.TODO(),
 &rds.CreateDBSnapshotInput{
  DBInstanceIdentifier: aws.String(instanceName),
  DBSnapshotIdentifier: aws.String(snapshotName),
 })
 if err != nil {
  log.Printf("Couldn't create snapshot %v: %v\n", snapshotName, err)
  return nil, err
 } else {
  return output.DBSnapshot, nil
 }
}
```

```
// GetSnapshot gets a DB instance snapshot.
func (instances *DbInstances) GetSnapshot(snapshotName string)
 (*types.DBSnapshot, error) {
 output, err := instances.RdsClient.DescribeDBSnapshots(context.TODO(),
  &rds.DescribeDBSnapshotsInput{
   DBSnapshotIdentifier: aws.String(snapshotName),
  })
 if err != nil {
  log.Printf("Couldn't get snapshot %v: %v\n", snapshotName, err)
  return nil, err
 } else {
  return &output.DBSnapshots[0], nil
 }
}



// CreateInstance creates a DB instance.
func (instances *DbInstances) CreateInstance(instanceName string, dbName string,
 dbEngine string, dbEngineVersion string, parameterGroupName string,
 dbInstanceClass string,
 storageType string, allocatedStorage int32, adminName string, adminPassword
 string) (
 *types.DBInstance, error) {
 output, err := instances.RdsClient.CreateDBInstance(context.TODO(),
 &rds.CreateDBInstanceInput{
  DBInstanceIdentifier: aws.String(instanceName),
  DBName:               aws.String(dbName),
  DBParameterGroupName: aws.String(parameterGroupName),
  Engine:               aws.String(dbEngine),
  EngineVersion:        aws.String(dbEngineVersion),
  DBInstanceClass:      aws.String(dbInstanceClass),
  StorageType:          aws.String(storageType),
  AllocatedStorage:     aws.Int32(allocatedStorage),
  MasterUsername:       aws.String(adminName),
  MasterUserPassword:   aws.String(adminPassword),
 })
 if err != nil {
  log.Printf("Couldn't create instance %v: %v\n", instanceName, err)
  return nil, err
 } else {
  return output.DBInstance, nil
```

```
 }
}



// GetInstance gets data about a DB instance.
func (instances *DbInstances) GetInstance(instanceName string) (
 *types.DBInstance, error) {
 output, err := instances.RdsClient.DescribeDBInstances(context.TODO(),
  &rds.DescribeDBInstancesInput{
   DBInstanceIdentifier: aws.String(instanceName),
  })
 if err != nil {
  var notFoundError *types.DBInstanceNotFoundFault
  if errors.As(err, &notFoundError) {
   log.Printf("DB instance %v does not exist.\n", instanceName)
   err = nil
  } else {
   log.Printf("Couldn't get instance %v: %v\n", instanceName, err)
  }
  return nil, err
 } else {
  return &output.DBInstances[0], nil
 }
}



// DeleteInstance deletes a DB instance.
func (instances *DbInstances) DeleteInstance(instanceName string) error {
 _, err := instances.RdsClient.DeleteDBInstance(context.TODO(),
 &rds.DeleteDBInstanceInput{
  DBInstanceIdentifier:   aws.String(instanceName),
  SkipFinalSnapshot:      true,
  DeleteAutomatedBackups: aws.Bool(true),
 })
 if err != nil {
  log.Printf("Couldn't delete instance %v: %v\n", instanceName, err)
  return err
 } else {
  return nil
 }
}
```

```
// GetEngineVersions gets database engine versions that are available for the
 specified engine
// and parameter group family.
func (instances *DbInstances) GetEngineVersions(engine string,
 parameterGroupFamily string) (
 []types.DBEngineVersion, error) {
 output, err := instances.RdsClient.DescribeDBEngineVersions(context.TODO(),
  &rds.DescribeDBEngineVersionsInput{
   Engine:                 aws.String(engine),
   DBParameterGroupFamily: aws.String(parameterGroupFamily),
  })
 if err != nil {
  log.Printf("Couldn't get engine versions for %v: %v\n", engine, err)
  return nil, err
 } else {
  return output.DBEngineVersions, nil
 }
}



// GetOrderableInstances uses a paginator to get DB instance options that can be
 used to create DB instances that are
// compatible with a set of specifications.
func (instances *DbInstances) GetOrderableInstances(engine string, engineVersion
 string) (
 []types.OrderableDBInstanceOption, error) {

 var output *rds.DescribeOrderableDBInstanceOptionsOutput
 var instanceOptions []types.OrderableDBInstanceOption
 var err error
 orderablePaginator :=
 rds.NewDescribeOrderableDBInstanceOptionsPaginator(instances.RdsClient,
  &rds.DescribeOrderableDBInstanceOptionsInput{
   Engine:        aws.String(engine),
   EngineVersion: aws.String(engineVersion),
  })
 for orderablePaginator.HasMorePages() {
  output, err = orderablePaginator.NextPage(context.TODO())
  if err != nil {
   log.Printf("Couldn't get orderable DB instance options: %v\n", err)
   break
```

```
  } else {
   instanceOptions = append(instanceOptions,
 output.OrderableDBInstanceOptions...)
  }
 }
 return instanceOptions, err
}
```

- Lihat detail API di topik-topik berikut dalam Referensi API AWS SDK for Go .
  - CreateDBInstance
  - dibuatB ParameterGroup
  - CreateDBSnapshot
  - DeleteDBInstance
  - DihapusB ParameterGroup
  - DijelaskanB EngineVersions
  - DescribeDBInstances
  - DijelaskanB ParameterGroups
  - DescribeDBParameters
  - DescribeDBSnapshots
  - DescribeOrderableDB InstanceOptions
  - ModifyDB ParameterGroup

Java

SDK for Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

Menjalankan beberapa operasi.

```
import com.google.gson.Gson;
```

```
import
 software.amazon.awssdk.auth.credentials.EnvironmentVariableCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.CreateDbInstanceRequest;
import software.amazon.awssdk.services.rds.model.CreateDbInstanceResponse;
import software.amazon.awssdk.services.rds.model.CreateDbParameterGroupResponse;
import software.amazon.awssdk.services.rds.model.CreateDbSnapshotRequest;
import software.amazon.awssdk.services.rds.model.CreateDbSnapshotResponse;
import software.amazon.awssdk.services.rds.model.DBEngineVersion;
import software.amazon.awssdk.services.rds.model.DBInstance;
import software.amazon.awssdk.services.rds.model.DBParameterGroup;
import software.amazon.awssdk.services.rds.model.DBSnapshot;
import software.amazon.awssdk.services.rds.model.DeleteDbInstanceRequest;
import software.amazon.awssdk.services.rds.model.DeleteDbInstanceResponse;
import software.amazon.awssdk.services.rds.model.DescribeDbEngineVersionsRequest;
import
 software.amazon.awssdk.services.rds.model.DescribeDbEngineVersionsResponse;
import software.amazon.awssdk.services.rds.model.DescribeDbInstancesRequest;
import software.amazon.awssdk.services.rds.model.DescribeDbInstancesResponse;
import
 software.amazon.awssdk.services.rds.model.DescribeDbParameterGroupsResponse;
import software.amazon.awssdk.services.rds.model.DescribeDbParametersResponse;
import software.amazon.awssdk.services.rds.model.DescribeDbSnapshotsRequest;
import software.amazon.awssdk.services.rds.model.DescribeDbSnapshotsResponse;
import
 software.amazon.awssdk.services.rds.model.DescribeOrderableDbInstanceOptionsResponse;
import software.amazon.awssdk.services.rds.model.ModifyDbParameterGroupResponse;
import software.amazon.awssdk.services.rds.model.OrderableDBInstanceOption;
import software.amazon.awssdk.services.rds.model.Parameter;
import software.amazon.awssdk.services.rds.model.RdsException;
import software.amazon.awssdk.services.rds.model.CreateDbParameterGroupRequest;
import
 software.amazon.awssdk.services.rds.model.DescribeDbParameterGroupsRequest;
import software.amazon.awssdk.services.rds.model.DescribeDbParametersRequest;
import software.amazon.awssdk.services.rds.model.ModifyDbParameterGroupRequest;
import
 software.amazon.awssdk.services.rds.model.DescribeOrderableDbInstanceOptionsRequest;
import software.amazon.awssdk.services.rds.model.DeleteDbParameterGroupRequest;
import software.amazon.awssdk.services.secretsmanager.SecretsManagerClient;
import
 software.amazon.awssdk.services.secretsmanager.model.GetSecretValueRequest;
import
 software.amazon.awssdk.services.secretsmanager.model.GetSecretValueResponse;
```

```
import java.util.ArrayList;
import java.util.List;

/**
 * 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, 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. Returns a list of the available DB engines.
 * 2. Selects an engine family and create a custom DB parameter group.
 * 3. Gets the parameter groups.
 * 4. Gets parameters in the group.
 * 5. Modifies the auto_increment_offset parameter.
 * 6. Gets and displays the updated parameters.
 * 7. Gets a list of allowed engine versions.
 * 8. Gets a list of micro instance classes available for the selected engine.
 * 9. Creates an RDS database instance that contains a MySql database and uses
 * the parameter group.
 * 10. Waits for the DB instance to be ready and prints out the connection
 * endpoint value.
 * 11. Creates a snapshot of the DB instance.
 * 12. Waits for an RDS DB snapshot to be ready.
 * 13. Deletes the RDS DB instance.
 * 14. Deletes the parameter group.
 */
public class RDSScenario {
    public static long sleepTime = 20;
    public static final String DASHES = new String(new char[80]).replace("\0",
 "-");

    public static void main(String[] args) throws InterruptedException {
```

```
        final String usage = """

                Usage:
                    <dbGroupName> <dbParameterGroupFamily> <dbInstanceIdentifier>
<dbName> <dbSnapshotIdentifier> <secretName>

                Where:
                    dbGroupName - The database group name.\s
                    dbParameterGroupFamily - The database parameter group name
(for example, mysql8.0).
                    dbInstanceIdentifier - The database instance identifier\s
                    dbName - The database name.\s
                    dbSnapshotIdentifier - The snapshot identifier.\s
                    secretName - The name of the AWS Secrets Manager secret that
contains the database credentials"
                """;

        if (args.length != 6) {
            System.out.println(usage);
            System.exit(1);
        }

        String dbGroupName = args[0];
        String dbParameterGroupFamily = args[1];
        String dbInstanceIdentifier = args[2];
        String dbName = args[3];
        String dbSnapshotIdentifier = args[4];
        String secretName = args[5];

        Gson gson = new Gson();
        User user = gson.fromJson(String.valueOf(getSecretValues(secretName)),
User.class);
        String masterUsername = user.getUsername();
        String masterUserPassword = user.getPassword();

        Region region = Region.US_WEST_2;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();
        System.out.println(DASHES);
        System.out.println("Welcome to the Amazon RDS example scenario.");
        System.out.println(DASHES);

        System.out.println(DASHES);
```

```
        System.out.println("1. Return a list of the available DB engines");
        describeDBEngines(rdsClient);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("2. Create a custom parameter group");
        createDBParameterGroup(rdsClient, dbGroupName, dbParameterGroupFamily);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("3. Get the parameter group");
        describeDbParameterGroups(rdsClient, dbGroupName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("4. Get the parameters in the group");
        describeDbParameters(rdsClient, dbGroupName, 0);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("5. Modify the auto_increment_offset parameter");
        modifyDBParas(rdsClient, dbGroupName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("6. Display the updated value");
        describeDbParameters(rdsClient, dbGroupName, -1);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("7. Get a list of allowed engine versions");
        getAllowedEngines(rdsClient, dbParameterGroupFamily);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("8. Get a list of micro instance classes available for
the selected engine");
        getMicroInstances(rdsClient);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println(
                "9. Create an RDS database instance that contains a MySql
database and uses the parameter group");
```

```
        String dbARN = createDatabaseInstance(rdsClient, dbGroupName,
dbInstanceIdentifier, dbName, masterUsername,
                masterUserPassword);
        System.out.println("The ARN of the new database is " + dbARN);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("10. Wait for DB instance to be ready");
        waitForInstanceReady(rdsClient, dbInstanceIdentifier);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("11. Create a snapshot of the DB instance");
        createSnapshot(rdsClient, dbInstanceIdentifier, dbSnapshotIdentifier);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("12. Wait for DB snapshot to be ready");
        waitForSnapshotReady(rdsClient, dbInstanceIdentifier,
dbSnapshotIdentifier);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("13. Delete the DB instance");
        deleteDatabaseInstance(rdsClient, dbInstanceIdentifier);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("14. Delete the parameter group");
        deleteParaGroup(rdsClient, dbGroupName, dbARN);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("The Scenario has successfully completed.");
        System.out.println(DASHES);

        rdsClient.close();
    }

    private static SecretsManagerClient getSecretClient() {
        Region region = Region.US_WEST_2;
        return SecretsManagerClient.builder()
                .region(region)
```

```
.credentialsProvider(EnvironmentVariableCredentialsProvider.create())
                .build();
   }

   public static String getSecretValues(String secretName) {
       SecretsManagerClient secretClient = getSecretClient();
       GetSecretValueRequest valueRequest = GetSecretValueRequest.builder()
               .secretId(secretName)
               .build();

       GetSecretValueResponse valueResponse =
secretClient.getSecretValue(valueRequest);
       return valueResponse.secretString();
   }

   // Delete the parameter group after database has been deleted.
   // An exception is thrown if you attempt to delete the para group while
database
   // exists.
   public static void deleteParaGroup(RdsClient rdsClient, String dbGroupName,
String dbARN)
           throws InterruptedException {
       try {
           boolean isDataDel = false;
           boolean didFind;
           String instanceARN;

           // Make sure that the database has been deleted.
           while (!isDataDel) {
               DescribeDbInstancesResponse response =
rdsClient.describeDBInstances();
               List<DBInstance> instanceList = response.dbInstances();
               int listSize = instanceList.size();
               didFind = false;
               int index = 1;
               for (DBInstance instance : instanceList) {
                   instanceARN = instance.dbInstanceArn();
                   if (instanceARN.compareTo(dbARN) == 0) {
                       System.out.println(dbARN + " still exists");
                       didFind = true;
                   }
                   if ((index == listSize) && (!didFind)) {
```

```
                        // Went through the entire list and did not find the
database ARN.
                        isDataDel = true;
                    }
                    Thread.sleep(sleepTime * 1000);
                    index++;
                }
            }

            // Delete the para group.
            DeleteDbParameterGroupRequest parameterGroupRequest =
DeleteDbParameterGroupRequest.builder()
                    .dbParameterGroupName(dbGroupName)
                    .build();

            rdsClient.deleteDBParameterGroup(parameterGroupRequest);
            System.out.println(dbGroupName + " was deleted.");

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    // Delete the DB instance.
    public static void deleteDatabaseInstance(RdsClient rdsClient, String
dbInstanceIdentifier) {
        try {
            DeleteDbInstanceRequest deleteDbInstanceRequest =
DeleteDbInstanceRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .deleteAutomatedBackups(true)
                    .skipFinalSnapshot(true)
                    .build();

            DeleteDbInstanceResponse response =
rdsClient.deleteDBInstance(deleteDbInstanceRequest);
            System.out.print("The status of the database is " +
response.dbInstance().dbInstanceStatus());

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
```

```
    }

    // Waits until the snapshot instance is available.
    public static void waitForSnapshotReady(RdsClient rdsClient, String
dbInstanceIdentifier,
            String dbSnapshotIdentifier) {
        try {
            boolean snapshotReady = false;
            String snapshotReadyStr;
            System.out.println("Waiting for the snapshot to become available.");

            DescribeDbSnapshotsRequest snapshotsRequest =
DescribeDbSnapshotsRequest.builder()
                    .dbSnapshotIdentifier(dbSnapshotIdentifier)
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .build();

            while (!snapshotReady) {
                DescribeDbSnapshotsResponse response =
rdsClient.describeDBSnapshots(snapshotsRequest);
                List<DBSnapshot> snapshotList = response.dbSnapshots();
                for (DBSnapshot snapshot : snapshotList) {
                    snapshotReadyStr = snapshot.status();
                    if (snapshotReadyStr.contains("available")) {
                        snapshotReady = true;
                    } else {
                        System.out.print(".");
                        Thread.sleep(sleepTime * 1000);
                    }
                }
            }

            System.out.println("The Snapshot is available!");
        } catch (RdsException | InterruptedException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    // Create an Amazon RDS snapshot.
    public static void createSnapshot(RdsClient rdsClient, String
dbInstanceIdentifier, String dbSnapshotIdentifier) {
        try {
```

```
            CreateDbSnapshotRequest snapshotRequest =
CreateDbSnapshotRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .dbSnapshotIdentifier(dbSnapshotIdentifier)
                    .build();

            CreateDbSnapshotResponse response =
rdsClient.createDBSnapshot(snapshotRequest);
            System.out.println("The Snapshot id is " +
response.dbSnapshot().dbiResourceId());

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    // Waits until the database instance is available.
    public static void waitForInstanceReady(RdsClient rdsClient, String
dbInstanceIdentifier) {
        boolean instanceReady = false;
        String instanceReadyStr;
        System.out.println("Waiting for instance to become available.");
        try {
            DescribeDbInstancesRequest instanceRequest =
DescribeDbInstancesRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .build();

            String endpoint = "";
            while (!instanceReady) {
                DescribeDbInstancesResponse response =
rdsClient.describeDBInstances(instanceRequest);
                List<DBInstance> instanceList = response.dbInstances();
                for (DBInstance instance : instanceList) {
                    instanceReadyStr = instance.dbInstanceStatus();
                    if (instanceReadyStr.contains("available")) {
                        endpoint = instance.endpoint().address();
                        instanceReady = true;
                    } else {
                        System.out.print(".");
                        Thread.sleep(sleepTime * 1000);
                    }
                }
```

```
            }
            System.out.println("Database instance is available! The connection
endpoint is " + endpoint);

        } catch (RdsException | InterruptedException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    // Create a database instance and return the ARN of the database.
    public static String createDatabaseInstance(RdsClient rdsClient,
            String dbGroupName,
            String dbInstanceIdentifier,
            String dbName,
            String masterUsername,
            String masterUserPassword) {

        try {
            CreateDbInstanceRequest instanceRequest =
CreateDbInstanceRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .allocatedStorage(100)
                    .dbName(dbName)
                    .dbParameterGroupName(dbGroupName)
                    .engine("mysql")
                    .dbInstanceClass("db.m4.large")
                    .engineVersion("8.0")
                    .storageType("standard")
                    .masterUsername(masterUsername)
                    .masterUserPassword(masterUserPassword)
                    .build();

            CreateDbInstanceResponse response =
rdsClient.createDBInstance(instanceRequest);
            System.out.print("The status is " +
response.dbInstance().dbInstanceStatus());
            return response.dbInstance().dbInstanceArn();

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
```

```
        return "";
    }

    // Get a list of micro instances.
    public static void getMicroInstances(RdsClient rdsClient) {
        try {
            DescribeOrderableDbInstanceOptionsRequest dbInstanceOptionsRequest =
DescribeOrderableDbInstanceOptionsRequest
                    .builder()
                    .engine("mysql")
                    .build();

            DescribeOrderableDbInstanceOptionsResponse response = rdsClient

.describeOrderableDBInstanceOptions(dbInstanceOptionsRequest);
            List<OrderableDBInstanceOption> orderableDBInstances =
response.orderableDBInstanceOptions();
            for (OrderableDBInstanceOption dbInstanceOption :
orderableDBInstances) {
                System.out.println("The engine version is " +
dbInstanceOption.engineVersion());
                System.out.println("The engine description is " +
dbInstanceOption.engine());
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    // Get a list of allowed engine versions.
    public static void getAllowedEngines(RdsClient rdsClient, String
dbParameterGroupFamily) {
        try {
            DescribeDbEngineVersionsRequest versionsRequest =
DescribeDbEngineVersionsRequest.builder()
                    .dbParameterGroupFamily(dbParameterGroupFamily)
                    .engine("mysql")
                    .build();

            DescribeDbEngineVersionsResponse response =
rdsClient.describeDBEngineVersions(versionsRequest);
            List<DBEngineVersion> dbEngines = response.dbEngineVersions();
```

```
            for (DBEngineVersion dbEngine : dbEngines) {
                System.out.println("The engine version is " +
dbEngine.engineVersion());
                System.out.println("The engine description is " +
dbEngine.dbEngineDescription());
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    // Modify auto_increment_offset and auto_increment_increment parameters.
    public static void modifyDBParas(RdsClient rdsClient, String dbGroupName) {
        try {
            Parameter parameter1 = Parameter.builder()
                    .parameterName("auto_increment_offset")
                    .applyMethod("immediate")
                    .parameterValue("5")
                    .build();

            List<Parameter> paraList = new ArrayList<>();
            paraList.add(parameter1);
            ModifyDbParameterGroupRequest groupRequest =
ModifyDbParameterGroupRequest.builder()
                    .dbParameterGroupName(dbGroupName)
                    .parameters(paraList)
                    .build();

            ModifyDbParameterGroupResponse response =
rdsClient.modifyDBParameterGroup(groupRequest);
            System.out.println("The parameter group " +
response.dbParameterGroupName() + " was successfully modified");

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    // Retrieve parameters in the group.
    public static void describeDbParameters(RdsClient rdsClient, String
dbGroupName, int flag) {
```

```
        try {
            DescribeDbParametersRequest dbParameterGroupsRequest;
            if (flag == 0) {
                dbParameterGroupsRequest = DescribeDbParametersRequest.builder()
                        .dbParameterGroupName(dbGroupName)
                        .build();
            } else {
                dbParameterGroupsRequest = DescribeDbParametersRequest.builder()
                        .dbParameterGroupName(dbGroupName)
                        .source("user")
                        .build();
            }

            DescribeDbParametersResponse response =
rdsClient.describeDBParameters(dbParameterGroupsRequest);
            List<Parameter> dbParameters = response.parameters();
            String paraName;
            for (Parameter para : dbParameters) {
                // Only print out information about either auto_increment_offset
or
                // auto_increment_increment.
                paraName = para.parameterName();
                if ((paraName.compareTo("auto_increment_offset") == 0)
                        || (paraName.compareTo("auto_increment_increment ") ==
0)) {
                    System.out.println("*** The parameter name is  " + paraName);
                    System.out.println("*** The parameter value is  " +
para.parameterValue());
                    System.out.println("*** The parameter data type is " +
para.dataType());
                    System.out.println("*** The parameter description is " +
para.description());
                    System.out.println("*** The parameter allowed values  is " +
para.allowedValues());
                }
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
```

```
   public static void describeDbParameterGroups(RdsClient rdsClient, String
dbGroupName) {
       try {
           DescribeDbParameterGroupsRequest groupsRequest =
DescribeDbParameterGroupsRequest.builder()
                   .dbParameterGroupName(dbGroupName)
                   .maxRecords(20)
                   .build();

           DescribeDbParameterGroupsResponse response =
rdsClient.describeDBParameterGroups(groupsRequest);
           List<DBParameterGroup> groups = response.dbParameterGroups();
           for (DBParameterGroup group : groups) {
               System.out.println("The group name is " +
group.dbParameterGroupName());
               System.out.println("The group description is " +
group.description());
           }

       } catch (RdsException e) {
           System.out.println(e.getLocalizedMessage());
           System.exit(1);
       }
   }

   public static void createDBParameterGroup(RdsClient rdsClient, String
dbGroupName, String dbParameterGroupFamily) {
       try {
           CreateDbParameterGroupRequest groupRequest =
CreateDbParameterGroupRequest.builder()
                   .dbParameterGroupName(dbGroupName)
                   .dbParameterGroupFamily(dbParameterGroupFamily)
                   .description("Created by using the AWS SDK for Java")
                   .build();

           CreateDbParameterGroupResponse response =
rdsClient.createDBParameterGroup(groupRequest);
           System.out.println("The group name is " +
response.dbParameterGroup().dbParameterGroupName());

       } catch (RdsException e) {
           System.out.println(e.getLocalizedMessage());
           System.exit(1);
       }
```

```
    }

    public static void describeDBEngines(RdsClient rdsClient) {
        try {
            DescribeDbEngineVersionsRequest engineVersionsRequest =
 DescribeDbEngineVersionsRequest.builder()
                    .defaultOnly(true)
                    .engine("mysql")
                    .maxRecords(20)
                    .build();

            DescribeDbEngineVersionsResponse response =
 rdsClient.describeDBEngineVersions(engineVersionsRequest);
            List<DBEngineVersion> engines = response.dbEngineVersions();

            // Get all DBEngineVersion objects.
            for (DBEngineVersion engineOb : engines) {
                System.out.println("The name of the DB parameter group family for
 the database engine is "
                        + engineOb.dbParameterGroupFamily());
                System.out.println("The name of the database engine " +
 engineOb.engine());
                System.out.println("The version number of the database engine " +
 engineOb.engineVersion());
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
}
```

- Lihat detail API di topik-topik berikut dalam Referensi API AWS SDK for Java 2.x .
  - CreateDBInstance
  - dibuatB ParameterGroup
  - CreateDBSnapshot
  - DeleteDBInstance
  - DihapusB ParameterGroup
  - DijelaskanB EngineVersions

- DescribeDBInstances
- DijelaskanB ParameterGroups
- DescribeDBParameters
- DescribeDBSnapshots
- DescribeOrderableDB InstanceOptions
- ModifyDB ParameterGroup

Kotlin

SDK for Kotlin

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
/**
Before running this 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-kotlin/latest/developer-guide/setup.html

This example requires an AWS Secrets Manager secret that contains the database
 credentials. If you do not create a
secret, this example will not work. For more details, see:

https://docs.aws.amazon.com/secretsmanager/latest/userguide/integrating_how-
services-use-secrets_RS.html

This example performs the following tasks:

1. Returns a list of the available DB engines by invoking the
 DescribeDbEngineVersions method.
2. Selects an engine family and create a custom DB parameter group by invoking
 the createDBParameterGroup method.
3. Gets the parameter groups by invoking the DescribeDbParameterGroups method.
```

```
4. Gets parameters in the group by invoking the DescribeDbParameters method.
5. Modifies both the auto_increment_offset and auto_increment_increment
 parameters by invoking the modifyDbParameterGroup method.
6. Gets and displays the updated parameters.
7. Gets a list of allowed engine versions by invoking the
 describeDbEngineVersions method.
8. Gets a list of micro instance classes available for the selected engine.
9. Creates an Amazon Relational Database Service (Amazon RDS) database instance
 that contains a MySQL database and uses the parameter group.
10. Waits for DB instance to be ready and prints out the connection endpoint
 value.
11. Creates a snapshot of the DB instance.
12. Waits for the DB snapshot to be ready.
13. Deletes the DB instance.
14. Deletes the parameter group.
 */

var sleepTime: Long = 20
suspend fun main(args: Array<String>) {
    val usage = """
        Usage:
            <dbGroupName> <dbParameterGroupFamily> <dbInstanceIdentifier>
 <dbName> <dbSnapshotIdentifier><secretName>

        Where:
            dbGroupName - The database group name.
            dbParameterGroupFamily - The database parameter group name.
            dbInstanceIdentifier - The database instance identifier.
            dbName -  The database name.
            dbSnapshotIdentifier - The snapshot identifier.
            secretName - The name of the AWS Secrets Manager secret that contains
 the database credentials.
    """

    if (args.size != 6) {
        println(usage)
        exitProcess(1)
    }

    val dbGroupName = args[0]
    val dbParameterGroupFamily = args[1]
    val dbInstanceIdentifier = args[2]
    val dbName = args[3]
    val dbSnapshotIdentifier = args[4]
```

```
    val secretName = args[5]

    val gson = Gson()
    val user = gson.fromJson(getSecretValues(secretName).toString(),
User::class.java)
    val username = user.username
    val userPassword = user.password

    println("1. Return a list of the available DB engines")
    describeDBEngines()

    println("2. Create a custom parameter group")
    createDBParameterGroup(dbGroupName, dbParameterGroupFamily)

    println("3. Get the parameter groups")
    describeDbParameterGroups(dbGroupName)

    println("4. Get the parameters in the group")
    describeDbParameters(dbGroupName, 0)

    println("5. Modify the auto_increment_offset parameter")
    modifyDBParas(dbGroupName)

    println("6. Display the updated value")
    describeDbParameters(dbGroupName, -1)

    println("7. Get a list of allowed engine versions")
    getAllowedEngines(dbParameterGroupFamily)

    println("8. Get a list of micro instance classes available for the selected
engine")
    getMicroInstances()

    println("9. Create an RDS database instance that contains a MySql database
and uses the parameter group")
    val dbARN = createDatabaseInstance(dbGroupName, dbInstanceIdentifier, dbName,
username, userPassword)
    println("The ARN of the new database is $dbARN")

    println("10. Wait for DB instance to be ready")
    waitForDbInstanceReady(dbInstanceIdentifier)

    println("11. Create a snapshot of the DB instance")
    createDbSnapshot(dbInstanceIdentifier, dbSnapshotIdentifier)
```

```
    println("12. Wait for DB snapshot to be ready")
    waitForSnapshotReady(dbInstanceIdentifier, dbSnapshotIdentifier)

    println("13. Delete the DB instance")
    deleteDbInstance(dbInstanceIdentifier)

    println("14. Delete the parameter group")
    if (dbARN != null) {
        deleteParaGroup(dbGroupName, dbARN)
    }

    println("The Scenario has successfully completed.")
}

suspend fun deleteParaGroup(dbGroupName: String, dbARN: String) {
    var isDataDel = false
    var didFind: Boolean
    var instanceARN: String

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        // Make sure that the database has been deleted.
        while (!isDataDel) {
            val response = rdsClient.describeDbInstances()
            val instanceList = response.dbInstances
            val listSize = instanceList?.size
            isDataDel = false // Reset this value.
            didFind = false // Reset this value.
            var index = 1
            if (instanceList != null) {
                for (instance in instanceList) {
                    instanceARN = instance.dbInstanceArn.toString()
                    if (instanceARN.compareTo(dbARN) == 0) {
                        println("$dbARN still exists")
                        didFind = true
                    }
                    if (index == listSize && !didFind) {
                        // Went through the entire list and did not find the
 database name.
                        isDataDel = true
                    }
                    index++
                }
            }
```

```
        }

        // Delete the para group.
        val parameterGroupRequest = DeleteDbParameterGroupRequest {
            dbParameterGroupName = dbGroupName
        }
        rdsClient.deleteDbParameterGroup(parameterGroupRequest)
        println("$dbGroupName was deleted.")
    }
}

suspend fun deleteDbInstance(dbInstanceIdentifierVal: String) {
    val deleteDbInstanceRequest = DeleteDbInstanceRequest {
        dbInstanceIdentifier = dbInstanceIdentifierVal
        deleteAutomatedBackups = true
        skipFinalSnapshot = true
    }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.deleteDbInstance(deleteDbInstanceRequest)
        print("The status of the database is
 ${response.dbInstance?.dbInstanceStatus}")
    }
}

// Waits until the snapshot instance is available.
suspend fun waitForSnapshotReady(dbInstanceIdentifierVal: String?,
 dbSnapshotIdentifierVal: String?) {
    var snapshotReady = false
    var snapshotReadyStr: String
    println("Waiting for the snapshot to become available.")

    val snapshotsRequest = DescribeDbSnapshotsRequest {
        dbSnapshotIdentifier = dbSnapshotIdentifierVal
        dbInstanceIdentifier = dbInstanceIdentifierVal
    }

    while (!snapshotReady) {
        RdsClient { region = "us-west-2" }.use { rdsClient ->
            val response = rdsClient.describeDbSnapshots(snapshotsRequest)
            val snapshotList: List<DbSnapshot>? = response.dbSnapshots
            if (snapshotList != null) {
                for (snapshot in snapshotList) {
                    snapshotReadyStr = snapshot.status.toString()
```

```
                    if (snapshotReadyStr.contains("available")) {
                        snapshotReady = true
                    } else {
                        print(".")
                        delay(sleepTime * 1000)
                    }
                }
            }
        }
    }
    println("The Snapshot is available!")
}

// Create an Amazon RDS snapshot.
suspend fun createDbSnapshot(dbInstanceIdentifierVal: String?,
 dbSnapshotIdentifierVal: String?) {
    val snapshotRequest = CreateDbSnapshotRequest {
        dbInstanceIdentifier = dbInstanceIdentifierVal
        dbSnapshotIdentifier = dbSnapshotIdentifierVal
    }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.createDbSnapshot(snapshotRequest)
        print("The Snapshot id is ${response.dbSnapshot?.dbiResourceId}")
    }
}

// Waits until the database instance is available.
suspend fun waitForDbInstanceReady(dbInstanceIdentifierVal: String?) {
    var instanceReady = false
    var instanceReadyStr: String
    println("Waiting for instance to become available.")

    val instanceRequest = DescribeDbInstancesRequest {
        dbInstanceIdentifier = dbInstanceIdentifierVal
    }
    var endpoint = ""
    while (!instanceReady) {
        RdsClient { region = "us-west-2" }.use { rdsClient ->
            val response = rdsClient.describeDbInstances(instanceRequest)
            val instanceList = response.dbInstances
            if (instanceList != null) {
                for (instance in instanceList) {
                    instanceReadyStr = instance.dbInstanceStatus.toString()
```

```
                    if (instanceReadyStr.contains("available")) {
                        endpoint = instance.endpoint?.address.toString()
                        instanceReady = true
                    } else {
                        print(".")
                        delay(sleepTime * 1000)
                    }
                }
            }
        }
    }
    println("Database instance is available! The connection endpoint is
 $endpoint")
}

// Create a database instance and return the ARN of the database.
suspend fun createDatabaseInstance(dbGroupNameVal: String?,
 dbInstanceIdentifierVal: String?, dbNameVal: String?, masterUsernameVal:
 String?, masterUserPasswordVal: String?): String? {
    val instanceRequest = CreateDbInstanceRequest {
        dbInstanceIdentifier = dbInstanceIdentifierVal
        allocatedStorage = 100
        dbName = dbNameVal
        dbParameterGroupName = dbGroupNameVal
        engine = "mysql"
        dbInstanceClass = "db.m4.large"
        engineVersion = "8.0"
        storageType = "standard"
        masterUsername = masterUsernameVal
        masterUserPassword = masterUserPasswordVal
    }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.createDbInstance(instanceRequest)
        print("The status is ${response.dbInstance?.dbInstanceStatus}")
        return response.dbInstance?.dbInstanceArn
    }
}

// Get a list of micro instances.
suspend fun getMicroInstances() {
    val dbInstanceOptionsRequest = DescribeOrderableDbInstanceOptionsRequest {
        engine = "mysql"
    }
```

```
    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response =
 rdsClient.describeOrderableDbInstanceOptions(dbInstanceOptionsRequest)
        val orderableDBInstances = response.orderableDbInstanceOptions
        if (orderableDBInstances != null) {
            for (dbInstanceOption in orderableDBInstances) {
                println("The engine version is
 ${dbInstanceOption.engineVersion}")
                println("The engine description is ${dbInstanceOption.engine}")
            }
        }
    }
}

// Get a list of allowed engine versions.
suspend fun getAllowedEngines(dbParameterGroupFamilyVal: String?) {
    val versionsRequest = DescribeDbEngineVersionsRequest {
        dbParameterGroupFamily = dbParameterGroupFamilyVal
        engine = "mysql"
    }
    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.describeDbEngineVersions(versionsRequest)
        val dbEngines: List<DbEngineVersion>? = response.dbEngineVersions
        if (dbEngines != null) {
            for (dbEngine in dbEngines) {
                println("The engine version is ${dbEngine.engineVersion}")
                println("The engine description is
 ${dbEngine.dbEngineDescription}")
            }
        }
    }
}

// Modify the auto_increment_offset parameter.
suspend fun modifyDBParas(dbGroupName: String) {
    val parameter1 = Parameter {
        parameterName = "auto_increment_offset"
        applyMethod = ApplyMethod.Immediate
        parameterValue = "5"
    }

    val paraList: ArrayList<Parameter> = ArrayList()
    paraList.add(parameter1)
    val groupRequest = ModifyDbParameterGroupRequest {
```

```
        dbParameterGroupName = dbGroupName
        parameters = paraList
    }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.modifyDbParameterGroup(groupRequest)
        println("The parameter group ${response.dbParameterGroupName} was
 successfully modified")
    }
}

// Retrieve parameters in the group.
suspend fun describeDbParameters(dbGroupName: String?, flag: Int) {
    val dbParameterGroupsRequest: DescribeDbParametersRequest
    dbParameterGroupsRequest = if (flag == 0) {
        DescribeDbParametersRequest {
            dbParameterGroupName = dbGroupName
        }
    } else {
        DescribeDbParametersRequest {
            dbParameterGroupName = dbGroupName
            source = "user"
        }
    }
    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.describeDbParameters(dbParameterGroupsRequest)
        val dbParameters: List<Parameter>? = response.parameters
        var paraName: String
        if (dbParameters != null) {
            for (para in dbParameters) {
                // Only print out information about either auto_increment_offset
 or auto_increment_increment.
                paraName = para.parameterName.toString()
                if (paraName.compareTo("auto_increment_offset") == 0 ||
 paraName.compareTo("auto_increment_increment ") == 0) {
                    println("*** The parameter name is  $paraName")
                    System.out.println("*** The parameter value is
 ${para.parameterValue}")
                    System.out.println("*** The parameter data type is
 ${para.dataType}")
                    System.out.println("*** The parameter description is
 ${para.description}")
                    System.out.println("*** The parameter allowed values  is
 ${para.allowedValues}")
```

```
                }
            }
        }
    }
}

suspend fun describeDbParameterGroups(dbGroupName: String?) {
    val groupsRequest = DescribeDbParameterGroupsRequest {
        dbParameterGroupName = dbGroupName
        maxRecords = 20
    }
    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.describeDbParameterGroups(groupsRequest)
        val groups = response.dbParameterGroups
        if (groups != null) {
            for (group in groups) {
                println("The group name is ${group.dbParameterGroupName}")
                println("The group description is ${group.description}")
            }
        }
    }
}

// Create a parameter group.
suspend fun createDBParameterGroup(dbGroupName: String?,
 dbParameterGroupFamilyVal: String?) {
    val groupRequest = CreateDbParameterGroupRequest {
        dbParameterGroupName = dbGroupName
        dbParameterGroupFamily = dbParameterGroupFamilyVal
        description = "Created by using the AWS SDK for Kotlin"
    }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.createDbParameterGroup(groupRequest)
        println("The group name is
 ${response.dbParameterGroup?.dbParameterGroupName}")
    }
}

// Returns a list of the available DB engines.
suspend fun describeDBEngines() {
    val engineVersionsRequest = DescribeDbEngineVersionsRequest {
        defaultOnly = true
        engine = "mysql"
```

```
        maxRecords = 20
    }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.describeDbEngineVersions(engineVersionsRequest)
        val engines: List<DbEngineVersion>? = response.dbEngineVersions

        // Get all DbEngineVersion objects.
        if (engines != null) {
            for (engineOb in engines) {
                println("The name of the DB parameter group family for the
 database engine is ${engineOb.dbParameterGroupFamily}.")
                println("The name of the database engine ${engineOb.engine}.")
                println("The version number of the database engine
 ${engineOb.engineVersion}")
            }
        }
    }
}

suspend fun getSecretValues(secretName: String?): String? {
    val valueRequest = GetSecretValueRequest {
        secretId = secretName
    }

    SecretsManagerClient { region = "us-west-2" }.use { secretsClient ->
        val valueResponse = secretsClient.getSecretValue(valueRequest)
        return valueResponse.secretString
    }
}
```

- Lihat detail API di topik-topik berikut dalam Referensi API AWS SDK For Kotlin.
  - CreateDBInstance
  - dibuatB ParameterGroup
  - CreateDBSnapshot
  - DeleteDBInstance
  - DihapusB ParameterGroup
  - DijelaskanB EngineVersions
  - DescribeDBInstances

- DijelaskanB ParameterGroups
- DescribeDBParameters
- DescribeDBSnapshots
- DescribeOrderableDB InstanceOptions
- ModifyDB ParameterGroup

Python

SDK for Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

Jalankan skenario interaktif di penggugah/prompt perintah.

```
class RdsInstanceScenario:
    """Runs a scenario that shows how to get started using Amazon RDS DB
 instances."""

    def __init__(self, instance_wrapper):
        """
        :param instance_wrapper: An object that wraps Amazon RDS DB instance
 actions.
        """
        self.instance_wrapper = instance_wrapper

    def create_parameter_group(self, parameter_group_name, db_engine):
        """
        Shows how to get available engine versions for a specified database
 engine and
        create a DB parameter group that is compatible with a selected engine
 family.

        :param parameter_group_name: The name given to the newly created
 parameter group.
        :param db_engine: The database engine to use as a basis.
        :return: The newly created parameter group.
```

```
        """
        print(
            f"Checking for an existing DB instance parameter group named
{parameter_group_name}."
        )
        parameter_group = self.instance_wrapper.get_parameter_group(
            parameter_group_name
        )
        if parameter_group is None:
            print(f"Getting available database engine versions for {db_engine}.")
            engine_versions =
self.instance_wrapper.get_engine_versions(db_engine)
            families = list({ver["DBParameterGroupFamily"] for ver in
engine_versions})
            family_index = q.choose("Which family do you want to use? ",
families)
            print(f"Creating a parameter group.")
            self.instance_wrapper.create_parameter_group(
                parameter_group_name, families[family_index], "Example parameter
group."
            )
            parameter_group = self.instance_wrapper.get_parameter_group(
                parameter_group_name
            )
        print(f"Parameter group {parameter_group['DBParameterGroupName']}:")
        pp(parameter_group)
        print("-" * 88)
        return parameter_group

    def update_parameters(self, parameter_group_name):
        """
        Shows how to get the parameters contained in a custom parameter group and
        update some of the parameter values in the group.

        :param parameter_group_name: The name of the parameter group to query and
modify.
        """
        print("Let's set some parameter values in your parameter group.")
        auto_inc_parameters = self.instance_wrapper.get_parameters(
            parameter_group_name, name_prefix="auto_increment"
        )
        update_params = []
        for auto_inc in auto_inc_parameters:
            if auto_inc["IsModifiable"] and auto_inc["DataType"] == "integer":
```

```
                print(f"The {auto_inc['ParameterName']} parameter is described
as:")
                print(f"\t{auto_inc['Description']}")
                param_range = auto_inc["AllowedValues"].split("-")
                auto_inc["ParameterValue"] = str(
                    q.ask(
                        f"Enter a value between {param_range[0]} and
{param_range[1]}: ",
                        q.is_int,
                        q.in_range(int(param_range[0]), int(param_range[1])),
                    )
                )
                update_params.append(auto_inc)
        self.instance_wrapper.update_parameters(parameter_group_name,
update_params)
        print(
            "You can get a list of parameters you've set by specifying a source
of 'user'."
        )
        user_parameters = self.instance_wrapper.get_parameters(
            parameter_group_name, source="user"
        )
        pp(user_parameters)
        print("-" * 88)

    def create_instance(self, instance_name, db_name, db_engine,
parameter_group):
        """
        Shows how to create a DB instance that contains a database of a specified
        type and is configured to use a custom DB parameter group.

        :param instance_name: The name given to the newly created DB instance.
        :param db_name: The name given to the created database.
        :param db_engine: The engine of the created database.
        :param parameter_group: The parameter group that is associated with the
DB instance.
        :return: The newly created DB instance.
        """
        print("Checking for an existing DB instance.")
        db_inst = self.instance_wrapper.get_db_instance(instance_name)
        if db_inst is None:
            print("Let's create a DB instance.")
            admin_username = q.ask(
```

```
                "Enter an administrator user name for the database: ",
q.non_empty
            )
            admin_password = q.ask(
                "Enter a password for the administrator (at least 8 characters):
",
                q.non_empty,
            )
            engine_versions = self.instance_wrapper.get_engine_versions(
                db_engine, parameter_group["DBParameterGroupFamily"]
            )
            engine_choices = [ver["EngineVersion"] for ver in engine_versions]
            print("The available engines for your parameter group are:")
            engine_index = q.choose("Which engine do you want to use? ",
engine_choices)
            engine_selection = engine_versions[engine_index]
            print(
                "The available micro DB instance classes for your database engine
are:"
            )
            inst_opts = self.instance_wrapper.get_orderable_instances(
                engine_selection["Engine"], engine_selection["EngineVersion"]
            )
            inst_choices = list(
                {
                    opt["DBInstanceClass"]
                    for opt in inst_opts
                    if "micro" in opt["DBInstanceClass"]
                }
            )
            inst_index = q.choose(
                "Which micro DB instance class do you want to use? ",
inst_choices
            )
            group_name = parameter_group["DBParameterGroupName"]
            storage_type = "standard"
            allocated_storage = 5
            print(
                f"Creating a DB instance named {instance_name} and database
{db_name}.\n"
                f"The DB instance is configured to use your custom parameter
group {group_name},\n"
                f"selected engine {engine_selection['EngineVersion']},\n"
                f"selected DB instance class {inst_choices[inst_index]},"
```

```
                f"and {allocated_storage} GiB of {storage_type} storage.\n"
                f"This typically takes several minutes."
            )
            db_inst = self.instance_wrapper.create_db_instance(
                db_name,
                instance_name,
                group_name,
                engine_selection["Engine"],
                engine_selection["EngineVersion"],
                inst_choices[inst_index],
                storage_type,
                allocated_storage,
                admin_username,
                admin_password,
            )
            while db_inst.get("DBInstanceStatus") != "available":
                wait(10)
                db_inst = self.instance_wrapper.get_db_instance(instance_name)
        print("Instance data:")
        pp(db_inst)
        print("-" * 88)
        return db_inst

    @staticmethod
    def display_connection(db_inst):
        """
        Displays connection information about a DB instance and tips on how to
        connect to it.

        :param db_inst: The DB instance to display.
        """
        print(
            "You can now connect to your database using your favorite MySql
 client.\n"
            "One way to connect is by using the 'mysql' shell on an Amazon EC2
 instance\n"
            "that is running in the same VPC as your DB instance. Pass the
 endpoint,\n"
            "port, and administrator user name to 'mysql' and enter your password
\n"
            "when prompted:\n"
        )
        print(
```

```
            f"\n\tmysql -h {db_inst['Endpoint']['Address']} -P
 {db_inst['Endpoint']['Port']} "
            f"-u {db_inst['MasterUsername']} -p\n"
        )
        print(
            "For more information, see the User Guide for Amazon RDS:\n"
            "\thttps://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
CHAP_GettingStarted.CreatingConnecting.MySQL.html#CHAP_GettingStarted.Connecting.MySQL"
        )
        print("-" * 88)

    def create_snapshot(self, instance_name):
        """
        Shows how to create a DB instance snapshot and wait until it's available.

        :param instance_name: The name of a DB instance to snapshot.
        """
        if q.ask(
            "Do you want to create a snapshot of your DB instance (y/n)? ",
 q.is_yesno
        ):
            snapshot_id = f"{instance_name}-{uuid.uuid4()}"
            print(
                f"Creating a snapshot named {snapshot_id}. This typically takes a
 few minutes."
            )
            snapshot = self.instance_wrapper.create_snapshot(snapshot_id,
 instance_name)
            while snapshot.get("Status") != "available":
                wait(10)
                snapshot = self.instance_wrapper.get_snapshot(snapshot_id)
            pp(snapshot)
            print("-" * 88)

    def cleanup(self, db_inst, parameter_group_name):
        """
        Shows how to clean up a DB instance and parameter group.
        Before the parameter group can be deleted, all associated DB instances
 must first
        be deleted.

        :param db_inst: The DB instance to delete.
        :param parameter_group_name: The DB parameter group to delete.
        """
```

```
        if q.ask(
            "\nDo you want to delete the DB instance and parameter group (y/n)?
",
            q.is_yesno,
        ):
            print(f"Deleting DB instance {db_inst['DBInstanceIdentifier']}.")

self.instance_wrapper.delete_db_instance(db_inst["DBInstanceIdentifier"])
            print(
                "Waiting for the DB instance to delete. This typically takes
several minutes."
            )
            while db_inst is not None:
                wait(10)
                db_inst = self.instance_wrapper.get_db_instance(
                    db_inst["DBInstanceIdentifier"]
                )
            print(f"Deleting parameter group {parameter_group_name}.")
            self.instance_wrapper.delete_parameter_group(parameter_group_name)

    def run_scenario(self, db_engine, parameter_group_name, instance_name,
db_name):
        logging.basicConfig(level=logging.INFO, format="%(levelname)s:
%(message)s")

        print("-" * 88)
        print(
            "Welcome to the Amazon Relational Database Service (Amazon RDS)\n"
            "get started with DB instances demo."
        )
        print("-" * 88)

        parameter_group = self.create_parameter_group(parameter_group_name,
db_engine)
        self.update_parameters(parameter_group_name)
        db_inst = self.create_instance(
            instance_name, db_name, db_engine, parameter_group
        )
        self.display_connection(db_inst)
        self.create_snapshot(instance_name)
        self.cleanup(db_inst, parameter_group_name)

        print("\nThanks for watching!")
        print("-" * 88)
```

```
if __name__ == "__main__":
    try:
        scenario = RdsInstanceScenario(InstanceWrapper.from_client())
        scenario.run_scenario(
            "mysql",
            "doc-example-parameter-group",
            "doc-example-instance",
            "docexampledb",
        )
    except Exception:
        logging.exception("Something went wrong with the demo.")
```

Tentukan fungsi-fungsi yang dipanggil oleh skenario untuk mengelola tindakan Amazon RDS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def get_parameter_group(self, parameter_group_name):
        """
        Gets a DB parameter group.

        :param parameter_group_name: The name of the parameter group to retrieve.
        :return: The parameter group.
        """
        try:
```

```
            response = self.rds_client.describe_db_parameter_groups(
                DBParameterGroupName=parameter_group_name
            )
            parameter_group = response["DBParameterGroups"][0]
        except ClientError as err:
            if err.response["Error"]["Code"] == "DBParameterGroupNotFound":
                logger.info("Parameter group %s does not exist.",
parameter_group_name)
            else:
                logger.error(
                    "Couldn't get parameter group %s. Here's why: %s: %s",
                    parameter_group_name,
                    err.response["Error"]["Code"],
                    err.response["Error"]["Message"],
                )
                raise
        else:
            return parameter_group


    def create_parameter_group(
        self, parameter_group_name, parameter_group_family, description
    ):
        """
        Creates a DB parameter group that is based on the specified parameter
group
        family.

        :param parameter_group_name: The name of the newly created parameter
group.
        :param parameter_group_family: The family that is used as the basis of
the new
                                       parameter group.
        :param description: A description given to the parameter group.
        :return: Data about the newly created parameter group.
        """
        try:
            response = self.rds_client.create_db_parameter_group(
                DBParameterGroupName=parameter_group_name,
                DBParameterGroupFamily=parameter_group_family,
                Description=description,
            )
        except ClientError as err:
            logger.error(
```

```
                "Couldn't create parameter group %s. Here's why: %s: %s",
                parameter_group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return response


    def delete_parameter_group(self, parameter_group_name):
        """
        Deletes a DB parameter group.

        :param parameter_group_name: The name of the parameter group to delete.
        :return: Data about the parameter group.
        """
        try:
            self.rds_client.delete_db_parameter_group(
                DBParameterGroupName=parameter_group_name
            )
        except ClientError as err:
            logger.error(
                "Couldn't delete parameter group %s. Here's why: %s: %s",
                parameter_group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise


    def get_parameters(self, parameter_group_name, name_prefix="", source=None):
        """
        Gets the parameters that are contained in a DB parameter group.

        :param parameter_group_name: The name of the parameter group to query.
        :param name_prefix: When specified, the retrieved list of parameters is
filtered
                            to contain only parameters that start with this
prefix.
        :param source: When specified, only parameters from this source are
retrieved.
                       For example, a source of 'user' retrieves only parameters
that
```

```
                    were set by a user.
        :return: The list of requested parameters.
        """
        try:
            kwargs = {"DBParameterGroupName": parameter_group_name}
            if source is not None:
                kwargs["Source"] = source
            parameters = []
            paginator = self.rds_client.get_paginator("describe_db_parameters")
            for page in paginator.paginate(**kwargs):
                parameters += [
                    p
                    for p in page["Parameters"]
                    if p["ParameterName"].startswith(name_prefix)
                ]
        except ClientError as err:
            logger.error(
                "Couldn't get parameters for %s. Here's why: %s: %s",
                parameter_group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return parameters


    def update_parameters(self, parameter_group_name, update_parameters):
        """
        Updates parameters in a custom DB parameter group.

        :param parameter_group_name: The name of the parameter group to update.
        :param update_parameters: The parameters to update in the group.
        :return: Data about the modified parameter group.
        """
        try:
            response = self.rds_client.modify_db_parameter_group(
                DBParameterGroupName=parameter_group_name,
Parameters=update_parameters
            )
        except ClientError as err:
            logger.error(
                "Couldn't update parameters in %s. Here's why: %s: %s",
                parameter_group_name,
```

```
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return response


    def create_snapshot(self, snapshot_id, instance_id):
        """
        Creates a snapshot of a DB instance.

        :param snapshot_id: The ID to give the created snapshot.
        :param instance_id: The ID of the DB instance to snapshot.
        :return: Data about the newly created snapshot.
        """
        try:
            response = self.rds_client.create_db_snapshot(
                DBSnapshotIdentifier=snapshot_id,
DBInstanceIdentifier=instance_id
            )
            snapshot = response["DBSnapshot"]
        except ClientError as err:
            logger.error(
                "Couldn't create snapshot of %s. Here's why: %s: %s",
                instance_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return snapshot


    def get_snapshot(self, snapshot_id):
        """
        Gets a DB instance snapshot.

        :param snapshot_id: The ID of the snapshot to retrieve.
        :return: The retrieved snapshot.
        """
        try:
            response = self.rds_client.describe_db_snapshots(
                DBSnapshotIdentifier=snapshot_id
```

```
            )
            snapshot = response["DBSnapshots"][0]
        except ClientError as err:
            logger.error(
                "Couldn't get snapshot %s. Here's why: %s: %s",
                snapshot_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return snapshot


    def get_engine_versions(self, engine, parameter_group_family=None):
        """
        Gets database engine versions that are available for the specified engine
        and parameter group family.

        :param engine: The database engine to look up.
        :param parameter_group_family: When specified, restricts the returned
list of
                                       engine versions to those that are
compatible with
                                       this parameter group family.
        :return: The list of database engine versions.
        """
        try:
            kwargs = {"Engine": engine}
            if parameter_group_family is not None:
                kwargs["DBParameterGroupFamily"] = parameter_group_family
            response = self.rds_client.describe_db_engine_versions(**kwargs)
            versions = response["DBEngineVersions"]
        except ClientError as err:
            logger.error(
                "Couldn't get engine versions for %s. Here's why: %s: %s",
                engine,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return versions
```

```
    def get_orderable_instances(self, db_engine, db_engine_version):
        """
        Gets DB instance options that can be used to create DB instances that are
        compatible with a set of specifications.

        :param db_engine: The database engine that must be supported by the DB
instance.
        :param db_engine_version: The engine version that must be supported by
the DB instance.
        :return: The list of DB instance options that can be used to create a
compatible DB instance.
        """
        try:
            inst_opts = []
            paginator = self.rds_client.get_paginator(
                "describe_orderable_db_instance_options"
            )
            for page in paginator.paginate(
                Engine=db_engine, EngineVersion=db_engine_version
            ):
                inst_opts += page["OrderableDBInstanceOptions"]
        except ClientError as err:
            logger.error(
                "Couldn't get orderable DB instances. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return inst_opts


    def get_db_instance(self, instance_id):
        """
        Gets data about a DB instance.

        :param instance_id: The ID of the DB instance to retrieve.
        :return: The retrieved DB instance.
        """
        try:
            response = self.rds_client.describe_db_instances(
                DBInstanceIdentifier=instance_id
            )
```

```
            db_inst = response["DBInstances"][0]
        except ClientError as err:
            if err.response["Error"]["Code"] == "DBInstanceNotFound":
                logger.info("Instance %s does not exist.", instance_id)
            else:
                logger.error(
                    "Couldn't get DB instance %s. Here's why: %s: %s",
                    instance_id,
                    err.response["Error"]["Code"],
                    err.response["Error"]["Message"],
                )
                raise
        else:
            return db_inst


    def create_db_instance(
        self,
        db_name,
        instance_id,
        parameter_group_name,
        db_engine,
        db_engine_version,
        instance_class,
        storage_type,
        allocated_storage,
        admin_name,
        admin_password,
    ):
        """
        Creates a DB instance.

        :param db_name: The name of the database that is created in the DB
instance.
        :param instance_id: The ID to give the newly created DB instance.
        :param parameter_group_name: A parameter group to associate with the DB
instance.
        :param db_engine: The database engine of a database to create in the DB
instance.
        :param db_engine_version: The engine version for the created database.
        :param instance_class: The DB instance class for the newly created DB
instance.
        :param storage_type: The storage type of the DB instance.
```

```
        :param allocated_storage: The amount of storage allocated on the DB
instance, in GiBs.
        :param admin_name: The name of the admin user for the created database.
        :param admin_password: The admin password for the created database.
        :return: Data about the newly created DB instance.
        """
        try:
            response = self.rds_client.create_db_instance(
                DBName=db_name,
                DBInstanceIdentifier=instance_id,
                DBParameterGroupName=parameter_group_name,
                Engine=db_engine,
                EngineVersion=db_engine_version,
                DBInstanceClass=instance_class,
                StorageType=storage_type,
                AllocatedStorage=allocated_storage,
                MasterUsername=admin_name,
                MasterUserPassword=admin_password,
            )
            db_inst = response["DBInstance"]
        except ClientError as err:
            logger.error(
                "Couldn't create DB instance %s. Here's why: %s: %s",
                instance_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return db_inst


    def delete_db_instance(self, instance_id):
        """
        Deletes a DB instance.

        :param instance_id: The ID of the DB instance to delete.
        :return: Data about the deleted DB instance.
        """
        try:
            response = self.rds_client.delete_db_instance(
                DBInstanceIdentifier=instance_id,
                SkipFinalSnapshot=True,
                DeleteAutomatedBackups=True,
```

```
            )
            db_inst = response["DBInstance"]
        except ClientError as err:
            logger.error(
                "Couldn't delete DB instance %s. Here's why: %s: %s",
                instance_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return db_inst
```

- Lihat detail API di topik-topik berikut dalam Referensi API AWS SDK for Python (Boto3).
  - CreateDBInstance
  - dibuatB ParameterGroup
  - CreateDBSnapshot
  - DeleteDBInstance
  - DihapusB ParameterGroup
  - DijelaskanB EngineVersions
  - DescribeDBInstances
  - DijelaskanB ParameterGroups
  - DescribeDBParameters
  - DescribeDBSnapshots
  - DescribeOrderableDB InstanceOptions
  - ModifyDB ParameterGroup

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang cara memulai dan detail versi-versi SDK sebelumnya.

Contoh tanpa server untuk Amazon RDS menggunakan SDK AWS

Contoh kode berikut menunjukkan cara menggunakan Amazon RDS dengan AWS SDK.

Contoh-contoh

- Menghubungkan ke database Amazon RDS dalam fungsi Lambda

Menghubungkan ke database Amazon RDS dalam fungsi Lambda

Contoh kode berikut menunjukkan bagaimana menerapkan fungsi Lambda yang menghubungkan ke database RDS. Fungsi membuat permintaan database sederhana dan mengembalikan hasilnya.

JavaScript

SDK untuk JavaScript (v2)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di repositori contoh Nirserver.

Melaporkan kegagalan item batch Kinesis dengan Lambda menggunakan Javascript.

```
/*
Node.js code here.
*/
// ES6+ example
import { Signer } from "@aws-sdk/rds-signer";
import mysql from 'mysql2/promise';

async function createAuthToken() {
  // Define connection authentication parameters
  const dbinfo = {

    hostname: process.env.ProxyHostName,
    port: process.env.Port,
    username: process.env.DBUserName,
    region: process.env.AWS_REGION,

  }
```

```
  // Create RDS Signer object
  const signer = new Signer(dbinfo);

  // Request authorization token from RDS, specifying the username
  const token = await signer.getAuthToken();
  return token;
}

async function dbOps() {

  // Obtain auth token
  const token = await createAuthToken();
  // Define connection configuration
  let connectionConfig = {
    host: process.env.ProxyHostName,
    user: process.env.DBUserName,
    password: token,
    database: process.env.DBName,
    ssl: 'Amazon RDS'
  }
  // Create the connection to the DB
  const conn = await mysql.createConnection(connectionConfig);
  // Obtain the result of the query
  const [res,] = await conn.execute('select ?+? as sum', [3, 2]);
  return res;

}

export const handler = async (event) => {
  // Execute database flow
  const result = await dbOps();
  // Return result
  return {
    statusCode: 200,
    body: JSON.stringify("The selected sum is: " + result[0].sum)
  }
};
```

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang cara memulai dan detail versi-versi SDK sebelumnya.

Contoh lintas layanan untuk Amazon RDS menggunakan SDK AWS

Contoh aplikasi berikut menggunakan AWS SDK untuk menggabungkan Amazon RDS dengan yang lain. Layanan AWS Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan petunjuk tentang cara mengatur dan menjalankan aplikasi.

Contoh-contoh

- Buat pelacak butir kerja Aurora Nirserver

Buat pelacak butir kerja Aurora Nirserver

Contoh-contoh kode berikut menunjukkan cara membuat aplikasi web yang melacak butir kerja dalam basis data Amazon Aurora Nirserver dan menggunakan Amazon Simple Email Service (Amazon SES) untuk mengirim laporan.

.NET

AWS SDK for .NET

Menunjukkan cara menggunakan AWS SDK for .NET untuk membuat aplikasi web yang melacak item pekerjaan dalam database Amazon Aurora dan laporan email dengan menggunakan Amazon Simple Email Service (Amazon SES). Contoh ini menggunakan sisi depan yang dibangun dengan React.js untuk berinteraksi dengan backend RESTful .NET.

- Integrasikan aplikasi web React dengan AWS layanan.
- Cantumkan, tambahkan, perbarui, dan hapus butir di tabel Aurora.
- Kirim laporan email tentang butir kerja terfilter dengan menggunakan Amazon SES.
- Menyebarkan dan mengelola sumber daya contoh dengan AWS CloudFormation skrip yang disertakan.

Untuk kode sumber lengkap dan instruksi tentang cara mengatur dan menjalankan, lihat contoh lengkapnya di GitHub.

Layanan yang digunakan dalam contoh ini

- Aurora
- Amazon RDS
- Layanan Data Amazon RDS
- Amazon SES

C++

SDK for C++

Menunjukkan cara membuat aplikasi web yang melacak dan melaporkan butir kerja yang tersimpan dalam basis data Amazon Aurora Nirserver.

Untuk kode sumber lengkap dan instruksi tentang cara menyiapkan C++ REST API yang menanyakan data Amazon Aurora Tanpa Server dan untuk digunakan oleh aplikasi React, lihat contoh lengkapnya di. GitHub

Layanan yang digunakan dalam contoh ini

- Aurora
- Amazon RDS
- Layanan Data Amazon RDS
- Amazon SES

Java

SDK for Java 2.x

Menunjukkan cara membuat aplikasi web yang melacak dan melaporkan butir kerja yang tersimpan dalam basis data Amazon RDS.

Untuk kode sumber lengkap dan petunjuk tentang cara menyiapkan Spring REST API yang menanyakan data Amazon Aurora Tanpa Server dan untuk digunakan oleh aplikasi React, lihat contoh lengkapnya di. GitHub

Untuk kode sumber lengkap dan instruksi tentang cara menyiapkan dan menjalankan contoh yang menggunakan JDBC API, lihat contoh lengkapnya di. GitHub

Layanan yang digunakan dalam contoh ini

- Aurora
- Amazon RDS
- Layanan Data Amazon RDS
- Amazon SES

JavaScript

SDK untuk JavaScript (v3)

Menunjukkan cara menggunakan AWS SDK for JavaScript (v3) untuk membuat aplikasi web yang melacak item pekerjaan dalam database Amazon Aurora dan laporan email dengan menggunakan Amazon Simple Email Service (Amazon SES). Contoh ini menggunakan sisi depan yang dibangun dengan React.js untuk berinteraksi dengan backend Express Node.js.

- Integrasikan aplikasi web React.js dengan Layanan AWS.
- Cantumkan, tambahkan, dan perbarui butir di tabel Aurora.
- Kirim laporan email tentang butir kerja terfilter dengan menggunakan Amazon SES.
- Menyebarkan dan mengelola sumber daya contoh dengan AWS CloudFormation skrip yang disertakan.

Untuk kode sumber lengkap dan instruksi tentang cara mengatur dan menjalankan, lihat contoh lengkapnya di GitHub.

Layanan yang digunakan dalam contoh ini

- Aurora
- Amazon RDS
- Layanan Data Amazon RDS
- Amazon SES

Kotlin

SDK for Kotlin

Menunjukkan cara membuat aplikasi web yang melacak dan melaporkan butir kerja yang tersimpan dalam basis data Amazon RDS.

Untuk kode sumber lengkap dan petunjuk tentang cara menyiapkan Spring REST API yang menanyakan data Amazon Aurora Tanpa Server dan untuk digunakan oleh aplikasi React, lihat contoh lengkapnya di. GitHub

Layanan yang digunakan dalam contoh ini

- Aurora
- Amazon RDS
- Layanan Data Amazon RDS
- Amazon SES

PHP

SDK for PHP

Menunjukkan cara menggunakan AWS SDK for PHP untuk membuat aplikasi web yang melacak item pekerjaan dalam database Amazon RDS dan laporan email dengan menggunakan Amazon Simple Email Service (Amazon SES). Contoh ini menggunakan sisi depan yang dibangun dengan React.js untuk berinteraksi dengan backend RESTful PHP.

- Integrasikan aplikasi web React.js dengan AWS layanan.
- Cantumkan, tambahkan, perbarui, dan hapus butir di tabel Amazon RDS.
- Kirim laporan email tentang butir kerja terfilter dengan menggunakan Amazon SES.
- Menyebarkan dan mengelola sumber daya contoh dengan AWS CloudFormation skrip yang disertakan.

Untuk kode sumber lengkap dan instruksi tentang cara mengatur dan menjalankan, lihat contoh lengkapnya di GitHub.

Layanan yang digunakan dalam contoh ini

- Aurora
- Amazon RDS
- Layanan Data Amazon RDS
- Amazon SES

Python

SDK for Python (Boto3)

Menunjukkan cara menggunakan AWS SDK for Python (Boto3) untuk membuat layanan REST yang melacak item pekerjaan di database Amazon Aurora Tanpa Server dan laporan email dengan menggunakan Amazon Simple Email Service (Amazon SES). Contoh ini menggunakan rangka kerja web Flask untuk menangani perutean HTTP dan terintegrasi dengan halaman web React untuk menyajikan aplikasi web yang berfungsi penuh.

- Bangun layanan Flask REST yang terintegrasi dengan. Layanan AWS
- Baca, tulis, dan perbarui butir kerja yang tersimpan dalam basis data Aurora Nirserver.
- Buat AWS Secrets Manager rahasia yang berisi kredensi database dan gunakan untuk mengautentikasi panggilan ke database.
- Gunakan Amazon SES untuk mengirim laporan email tentang butir kerja.

Untuk kode sumber lengkap dan instruksi tentang cara mengatur dan menjalankan, lihat contoh lengkapnya di GitHub.

Layanan yang digunakan dalam contoh ini

- Aurora
- Amazon RDS
- Layanan Data Amazon RDS
- Amazon SES

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang cara memulai dan detail versi-versi SDK sebelumnya.

Keamanan dalam Amazon RDS

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai pelanggan AWS, Anda mendapatkan manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan dari organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara AWS dan Anda. Model tanggung jawab bersama menjelaskan hal ini sebagai keamanan dari cloud dan keamanan di cloud:

- Keamanan cloud – AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan layanan AWS di Cloud AWS. AWS juga memberikan Anda layanan yang dapat digunakan dengan aman. Auditor pihak ketiga secara berkala menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari AWS program kepatuhan. Untuk mempelajari program kepatuhan yang berlaku pada Amazon RDS, lihat layanan AWS dalam cakupan menurut program kepatuhan.
- Keamanan di cloud – Tanggung jawab Anda ditentukan oleh AWS yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, termasuk sensitivitas data, persyaratan perusahaan, serta hukum dan peraturan yang berlaku.

Dokumentasi ini akan membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan Amazon RDS. Topik berikut menunjukkan kepada Anda cara mengonfigurasi Amazon RDS untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga akan mempelajari cara menggunakan layanan AWS lain yang membantu Anda memantau dan mengamankan sumber daya Amazon RDS.

Anda dapat mengelola akses ke sumber daya Amazon RDS dan basis data Anda di instans DB. Metode yang Anda gunakan untuk mengelola akses ditentukan oleh jenis tugas yang harus dilakukan oleh pengguna dengan Amazon RDS:

- Jalankan klaster DB Anda dalam cloud privat virtual (VPC) berdasarkan layanan Amazon VPC untuk kontrol akses jaringan tertinggi. Untuk informasi selengkapnya tentang cara membuat klaster DB di VPC, lihat Amazon VPC dan Amazon RDS.
- Gunakan kebijakan (IAM) AWS Identity and Access Management untuk menetapkan izin yang menentukan siapa yang diizinkan untuk mengelola sumber daya Amazon RDS. Misalnya, Anda dapat menggunakan IAM untuk menentukan siapa yang diizinkan untuk membuat, menjelaskan, memodifikasi, dan menghapus klaster DB, memberi tag pada sumber daya, atau memodifikasi grup keamanan.

- Gunakan grup keamanan untuk mengontrol alamat IP atau instans Amazon EC2 yang dapat terhubung ke basis data Anda di klaster DB. Saat Anda pertama kali membuat klaster DB, firewall-nya mencegah semua akses basis data kecuali melalui aturan yang ditentukan oleh grup keamanan terkait.
- Gunakan koneksi Lapisan Soket Aman (SSL) atau Keamanan Lapisan Pengangkutan (TLS) dengan instans DB yang menjalankan mesin basis data Db2, MySQL, MariaDB, PostgreSQL, Oracle, atau Microsoft SQL Server. Untuk informasi selengkapnya tentang cara menggunakan SSL/TLS dengan klaster DB, lihat .
- Gunakan enkripsi Amazon RDS untuk mengamankan instans DB dan snapshot yang tersimpan. Enkripsi Amazon RDS menggunakan algoritma enkripsi AES-256 standar industri untuk mengenkripsi data Anda di server yang meng-host klaster DB Anda. Untuk informasi selengkapnya, lihat Mengenkripsi sumber daya Amazon RDS.
- Gunakan enkripsi jaringan dan enkripsi data transparan dengan instans DB Oracle; untuk informasi selengkapnya, lihat Enkripsi jaringan asli Oracle dan Enkripsi Data Transparan Oracle
- Gunakan fitur keamanan pada mesin DB Anda untuk mengontrol siapa yang dapat login ke basis data di klaster DB. Fitur ini berfungsi seolah-olah basis data berada di jaringan lokal Anda.

> **Note**
>
> Anda hanya perlu mengonfigurasikan keamanan untuk kasus penggunaan Anda. Anda tidak perlu mengonfigurasi akses keamanan untuk proses yang dikelola Amazon RDS. Ini termasuk membuat cadangan, mereplikasi data antara instans DB utama dan replika baca, dan proses lainnya.

Untuk informasi selengkapnya tentang cara mengelola akses ke sumber daya Amazon RDS dan basis data Anda di klaster DB, lihat topik berikut.

Topik

- Autentikasi basis data dengan Amazon RDS
- Manajemen kata sandi dengan Amazon RDS Aurora dan AWS Secrets Manager
- Perlindungan data di Amazon RDS
- Manajemen identitas dan akses untuk Amazon RDS
- Pencatatan dan pemantauan di Amazon RDS
- Validasi kepatuhan untuk Amazon RDS

- Ketangguhan di Amazon RDS
- Keamanan infrastruktur di Amazon RDS
- API Amazon RDS dan titik akhir VPC antarmuka (AWS PrivateLink)
- Praktik terbaik keamanan untuk Amazon RDS
- Mengontrol akses dengan grup keamanan
- Hak akses akun pengguna master
- Menggunakan peran terkait layanan untuk Amazon RDS
- Amazon VPC dan Amazon RDS

Autentikasi basis data dengan Amazon RDS

Amazon RDS mendukung beberapa cara untuk mengautentikasi pengguna basis data.

Autentikasi kata sandi, Kerberos, dan basis data IAM menggunakan metode autentikasi yang berbeda ke basis data. Oleh karena itu, pengguna tertentu dapat masuk ke basis data dengan menggunakan hanya satu metode autentikasi.

Untuk PostgreSQL, gunakan hanya salah satu dari setelah peran berikut untuk pengguna basis data tertentu:

- Untuk menggunakan autentikasi basis data IAM, tetapkan peran `rds_iam` untuk pengguna.
- Untuk menggunakan autentikasi Kerberos, tetapkan peran `rds_ad` untuk pengguna.
- Untuk menggunakan autentikasi kata sandi, jangan tetapkan peran `rds_iam` atau `rds_ad` untuk pengguna.

Jangan tetapkan kedua peran `rds_iam` dan `rds_ad` untuk pengguna basis data PostgreSQL baik secara langsung maupun tidak langsung dengan akses pemberian bersarang. Jika peran `rds_iam` ditambahkan ke pengguna master, autentikasi IAM diutamakan atas autentikasi kata sandi sehingga pengguna master harus masuk sebagai pengguna IAM.

> **Important**
>
> Sebaiknya jangan gunakan pengguna master secara langsung di aplikasi Anda. Sebagai gantinya, ikuti praktik terbaik penggunaan pengguna basis data yang dibuat dengan privilese minimal yang diperlukan untuk aplikasi Anda.

Topik

Autentikasi kata sandi

Dengan autentikasi kata sandi, basis data Anda melakukan semua administrasi akun pengguna. Anda membuat pengguna dengan pernyataan SQL seperti `CREATE USER`, dengan klausa yang tepat yang diperlukan oleh mesin basis data untuk menentukan kata sandi. Misalnya, di MySQL, pernyataannya adalah `CREATE USER` *`nama`* `IDENTIFIED BY` *`kata sandi`*, sedangkan di PostgreSQL, pernyataannya `CREATE USER` *`nama`* `WITH PASSWORD` *`kata sandi`*.

Dengan autentikasi kata sandi, basis data Anda mengendalikan dan mengautentikasi akun pengguna. Jika mesin basis data memiliki fitur pengelolaan kata sandi kuat, mesin itu dapat meningkatkan keamanan. Autentikasi basis data mungkin lebih mudah dikelola dengan menggunakan autentikasi kata sandi apabila komunitas pengguna Anda kecil. Karena dalam hal ini dihasilkan kata sandi teks jelas, mengintegrasikan dengan AWS Secrets Manager dapat meningkatkan keamanan.

Lihat informasi tentang cara menggunakan Secrets Manager dengan Amazon RDS di Membuat rahasia dasar dan Merotasi rahasia untuk basis data Amazon RDS yang didukung dalam Panduan Pengguna AWS Secrets Manager. Lihat informasi tentang cara mengambil rahasia secara terprogram pada aplikasi kustom Anda di Mengambil nilai rahasia dalam Panduan Pengguna AWS Secrets Manager.

Autentikasi basis data IAM

Anda dapat mengautentikasi instans basis data Anda dengan menggunakan autentikasi basis data AWS Identity and Access Management (IAM). Autentikasi basis data IAM bekerja dengan MySQL dan PostgreSQL. Dengan metode autentikasi ini, Anda tidak perlu menggunakan kata sandi saat menghubungi instans basis data. Sebagai gantinya, Anda menggunakan token autentikasi.

Lihat informasi yang lebih lengkap tentang autentikasi basis data IAM, yang meliputi informasi tentang ketersediaan mesin basis data tertentu, di Autentikasi basis data IAM untuk MariaDB, MySQL, dan PostgreSQL.

Autentikasi Kerberos

Amazon RDS mendukung autentikasi eksternal pengguna basis data dengan menggunakan Kerberos dan Microsoft Active Directory. Kerberos adalah protokol autentikasi jaringan yang menggunakan tiket dan kriptografi kunci simetris untuk menghilangkan kebutuhan mengirim kata sandi melalui jaringan. Kerberos telah tertanam ke dalam Active Directory dan dirancang untuk mengautentikasi pengguna ke sumber daya jaringan, seperti basis data.

Dukungan Amazon RDS untuk Kerberos dan Active Directory memberikan manfaat upaya masuk tunggal dan autentikasi terpusat pengguna basis data. Anda dapat menyimpan kredensial pengguna Anda di Active Directory. Active Directory menyediakan tempat terpusat untuk menyimpan dan mengelola kredensial bagi beberapa instans basis data.

Anda bisa membuat pengguna basis data dapat mengautentikasi instans basis data dengan dua cara. Pengguna dapat menggunakan kredensial yang disimpan di AWS Directory Service for Microsoft Active Directory atau di Active Directory on-premise Anda.

Instans basis data Microsoft SQL Server dan PostgreSQL mendukung hubungan kepercayaan rimba satu dan dua arah. Instans basis data Oracle mendukung hubungan kepercayaan eksternal dan rimba satu atau dua arah. Lihat informasi yang lebih lengkap di Kapan sebaiknya menciptakan hubungan kepercayaan dalam Panduan Administrasi AWS Directory Service.

Lihat informasi tentang autentikasi Kerberos dengan mesin basis data tertentu di:

- Menggunakan AWS Managed Active Directory dengan RDS for SQL Server
- Menggunakan autentikasi Kerberos untuk MySQL
- Mengonfigurasi autentikasi Kerberos untuk Amazon RDS for Oracle
- Menggunakan autentikasi Kerberos dengan Amazon RDS for PostgreSQL

> **Note**
>
> Saat ini, autentikasi Kerberos tidak didukung untuk instans basis data MariaDB.

Manajemen kata sandi dengan Amazon RDS Aurora dan AWS Secrets Manager

Amazon RDS terintegrasi dengan Secrets Manager untuk mengelola kata sandi pengguna utama untuk klaster instans DB dan DB Multi-AZ.

Topik

Batasan untuk integrasi Secrets Manager dengan Amazon RDS

Mengelola kata sandi pengguna utama dengan Secrets Manager tidak didukung untuk fitur berikut:

- Untuk semua mesin DB kecuali RDS for SQL Server, membuat replika baca ketika DB sumber atau klaster DB mengelola kredensil dengan Secrets Manager
- Deployment Blue/Green Amazon RDS
- Amazon RDS Custom
- Switchover Oracle Data Guard
- RDS for Oracle dengan CDB

Ikhtisar mengelola kata sandi pengguna master dengan AWS Secrets Manager

Dengan AWS Secrets Manager, Anda dapat mengganti kredensi hard-code dalam kode Anda, termasuk kata sandi database, dengan panggilan API ke Secrets Manager untuk mengambil rahasia secara terprogram. Untuk mengetahui informasi selengkapnya tentang Secrets Manager, lihat Panduan Pengguna AWS Secrets Manager.

Ketika Anda menyimpan rahasia database di Secrets Manager, Anda akan Akun AWS dikenakan biaya. Untuk informasi tentang harga, lihat AWS Secrets Manager Harga.

Anda dapat menentukan bahwa RDS mengelola kata sandi pengguna utama di Secrets Manager untuk instans DB Amazon RDS atau klaster DB Multi-AZ saat Anda melakukan salah satu operasi berikut:

- Membuat instans DB
- Membuat replika baca klaster DB Multi-AZ
- Mengubah instans DB
- Mengubah klaster DB Multi-AZ
- Memulihkan instans DB dari Amazon S3

Saat Anda menentukan bahwa RDS mengelola kata sandi pengguna utama di Secrets Manager, RDS menghasilkan kata sandi dan menyimpannya dalam Secrets Manager. Anda dapat berinteraksi langsung dengan rahasia untuk mengambil kredensial untuk pengguna utama. Anda juga dapat menentukan kunci yang dikelola pelanggan untuk mengenkripsi rahasia, atau menggunakan kunci KMS default yang disediakan oleh Secrets Manager.

Secara default, RDS mengelola pengaturan untuk rahasia dan merotasi rahasia setiap tujuh hari. Anda dapat mengubah beberapa pengaturan, seperti jadwal rotasi. Jika Anda menghapus instans DB yang mengelola rahasia di Secrets Manager, rahasia dan metadata terkaitnya juga akan dihapus.

Untuk terhubung ke klaster instans atau DB Multi-AZ DB dengan kredensial dalam rahasia, Anda dapat mengambil rahasia dari Secrets Manager. 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

Manfaat mengelola kata sandi pengguna utama dengan Secrets Manager

Mengelola kata sandi pengguna utama RDS dengan Secrets Manager memberikan manfaat berikut:

- RDS secara otomatis menghasilkan kredensial basis data.
- RDS secara otomatis menyimpan dan mengelola kredensil database di. AWS Secrets Manager
- RDS merotasi kredensial basis data secara teratur, tanpa mewajibkan perubahan aplikasi.
- Secrets Manager mengamankan kredensial basis data dari akses manusia dan tampilan teks biasa.
- Secrets Manager memungkinkan pengambilan kredensial basis data rahasia untuk koneksi basis data.
- Secrets Manager memungkinkan kontrol akses terperinci ke kredensial basis data dalam rahasia menggunakan IAM.
- Secara opsional, Anda dapat memisahkan enkripsi basis data dari enkripsi kredensial dengan kunci KMS lainnya.
- Anda dapat menghilangkan rotasi dan manajemen manual kredensial basis data.
- Anda dapat memantau kredensi database dengan mudah dengan dan AWS CloudTrail Amazon. CloudWatch

Untuk informasi selengkapnya tentang manfaat Secrets Manager, lihat Panduan Pengguna AWS Secrets Manager.

Izin yang diperlukan untuk integrasi Secrets Manager

Pengguna harus memiliki izin yang diperlukan untuk melakukan operasi yang terkait dengan integrasi Secrets Manager. Anda dapat membuat kebijakan IAM yang memberikan izin untuk melakukan operasi API tertentu pada sumber daya spefisik yang diperlukan. Anda kemudian dapat melampirkan kebijakan tersebut ke set izin IAM atau peran yang memerlukan izin tersebut. Untuk informasi selengkapnya, lihat Manajemen identitas dan akses untuk Amazon RDS.

Untuk membuat, memodifikasi, atau memulihkan operasi, pengguna yang menentukan bahwa Amazon RDS mengelola kata sandi pengguna utama di Secrets Manager harus memiliki izin untuk melakukan operasi berikut:

- `kms:DescribeKey`
- `secretsmanager:CreateSecret`

- `secretsmanager:TagResource`

Untuk membuat, memodifikasi, atau memulihkan operasi, pengguna yang menentukan kunci yang dikelola pelanggan untuk mengenkripsi rahasia dalam Secrets Manager harus memiliki izin untuk melakukan operasi berikut:

- `kms:Decrypt`
- `kms:GenerateDataKey`
- `kms:CreateGrant`

Untuk mengubah operasi, pengguna yang merotasi kata sandi pengguna utama dalam Secrets Manager harus memiliki izin untuk melakukan operasi berikut:

- `secretsmanager:RotateSecret`

Menegakkan manajemen RDS Aurora sandi pengguna utama di AWS Secrets Manager

Anda dapat menggunakan kunci kondisi IAM untuk menerapkan manajemen RDS kata sandi pengguna utama di AWS Secrets Manager. Kebijakan berikut tidak mengizinkan pengguna untuk membuat atau memulihkan instans DB atau klaster DB kecuali kata sandi pengguna utama dikelola oleh RDS di Secrets Manager.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": ["rds:CreateDBInstance", "rds:CreateDBCluster",
 "rds:RestoreDBInstanceFromS3", "rds:RestoreDBClusterFromS3"],
            "Resource": "*",
            "Condition": {
                "Bool": {
                    "rds:ManageMasterUserPassword": false
                }
            }
        }
    ]
```

```
}
```

> **Note**
>
> Kebijakan ini memberlakukan manajemen kata sandi pada AWS Secrets Manager saat pembuatan. Namun, Anda masih dapat menonaktifkan integrasi Secrets Manager dan mengatur kata sandi utama secara manual dengan mengubah instans.
> Untuk mencegahnya, sertakan `rds:ModifyDBInstance`, `rds:ModifyDBCluster` dalam blok tindakan kebijakan. Perhatikan bahwa tindakan ini akan mencegah pengguna menerapkan perubahan lebih lanjut pada instans yang ada yang Secrets Manager-nya tidak diaktifkan.

Untuk informasi lebih lanjut tentang penggunaan kunci kondisi dalam kebijakan IAM, lihat Kunci kondisi kebijakan untuk Amazon RDS dan Contoh Kebijakan: Menggunakan kunci kondisi.

Mengelola kata sandi pengguna utama untuk instans DB dengan Secrets Manager

Anda dapat mengonfigurasi manajemen RDS kata sandi pengguna utama di Secrets Manager saat Anda melakukan tindakan berikut:

- Membuat instans DB Amazon RDS
- Memodifikasi instans DB Amazon RDS
- Memulihkan cadangan ke instans DB MySQL

Anda dapat menggunakan konsol RDS, API AWS CLI, atau RDS untuk melakukan tindakan ini.

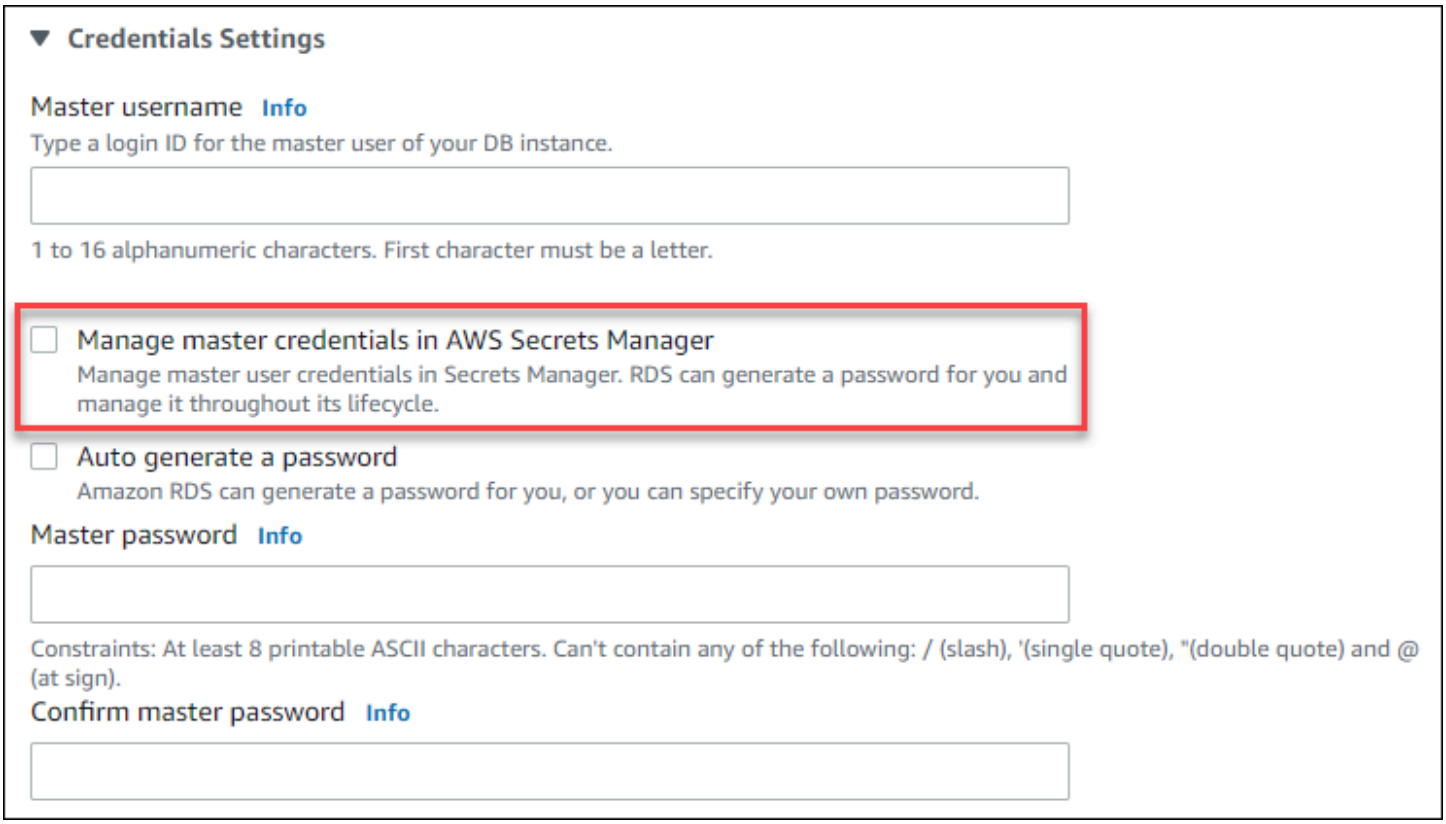
Konsol

Ikuti petunjuk untuk membuat atau mengubah instans DB dengan konsol RDS:

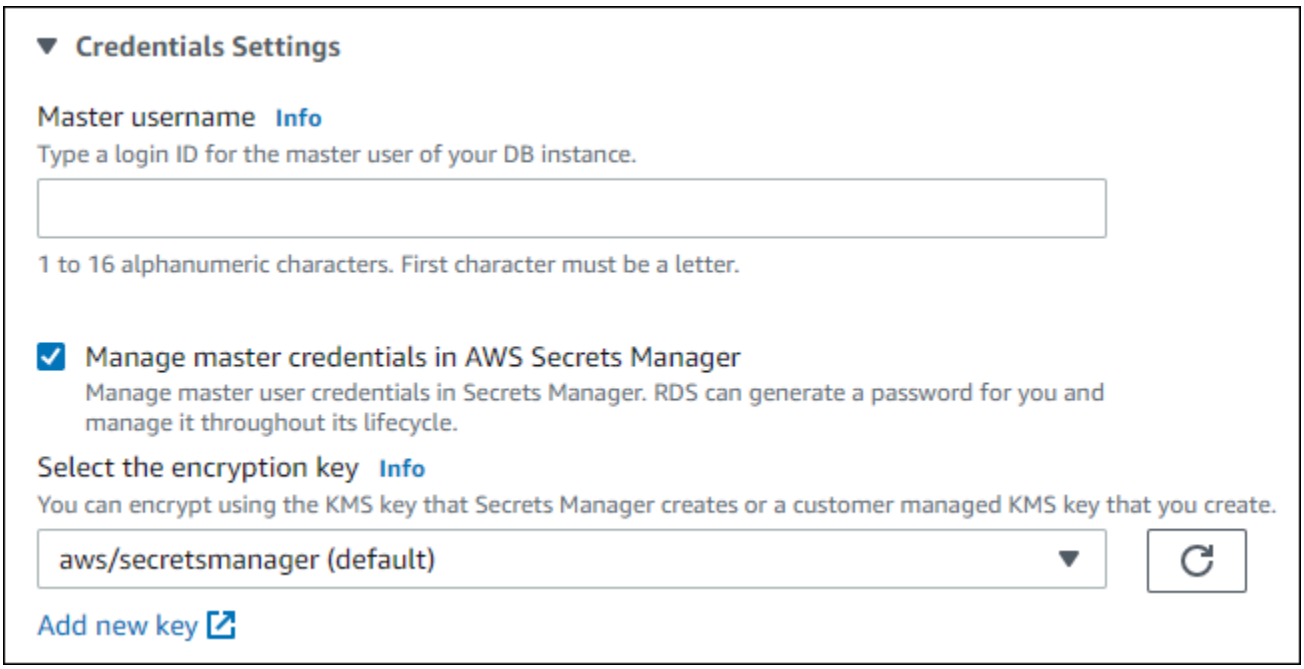
- Membuat instans DB
- Memodifikasi instans DB Amazon RDS
- Mengimpor data dari Amazon S3 ke instans DB MySQL baru

Saat menggunakan konsol RDS untuk melakukan salah satu operasi ini, Anda dapat menentukan bahwa kata sandi pengguna utama dikelola oleh RDS di Secrets Manager. Untuk melakukannya saat Anda membuat atau memulihkan instans DB, pilih Kelola kredensial utama di AWS Secrets Manager dalam Pengaturan kredensial. Saat Anda mengubah instans DB, pilih Kelola kredensial utama di AWS Secrets Manager dalam Pengaturan.

Gambar berikut adalah contoh pengaturan Kelola kredensial utama di AWS Secrets Manager saat Anda membuat atau memulihkan instans DB.



Saat Anda memilih opsi ini, RDS akan menghasilkan kata sandi pengguna utama dan mengelolanya sepanjang siklus pemakaiannya di Secrets Manager.



Anda dapat memilih untuk mengenkripsi rahasia dengan kunci KMS yang disediakan Secrets Manager atau dengan kunci yang dikelola pelanggan yang Anda buat. Setelah RDS mengelola kredensial basis data untuk instans DB, Anda tidak dapat mengubah kunci KMS yang digunakan untuk mengenkripsi rahasia.

Anda dapat memilih pengaturan lain untuk memenuhi kebutuhan Anda. Untuk informasi selengkapnya tentang pengaturan yang tersedia saat Anda membuat instans DB, lihat Pengaturan untuk instans DB. Untuk informasi selengkapnya tentang pengaturan yang tersedia saat Anda mengubah instans DB, lihat Pengaturan untuk instans DB.

AWS CLI

Untuk mengelola kata sandi pengguna master dengan RDS di Secrets Manager, tentukan `--manage-master-user-password` opsi di salah satu AWS CLI perintah berikut:

- create-db-instance
- modify-db-instance
- restore-db-instance-from-s3

Jika Anda memilih opsi `--manage-master-user-password` dalam perintah ini, RDS akan menghasilkan kata sandi pengguna utama dan mengelolanya sepanjang siklus pemakaiannya di Secrets Manager.

Untuk mengenkripsi rahasia, Anda dapat menentukan kunci yang dikelola pelanggan atau menggunakan kunci KMS default yang disediakan oleh Secrets Manager. Gunakan opsi `--master-user-secret-kms-key-id` untuk menentukan kunci yang dikelola pelanggan. Pengidentifikasi kunci AWS KMS adalah kunci ARN, ID kunci, alias ARN, atau nama alias untuk kunci KMS. Untuk menggunakan kunci KMS yang berbeda Akun AWS, tentukan kunci ARN atau alias ARN. Setelah RDS mengelola kredensial basis data untuk instans DB, Anda tidak dapat mengubah kunci KMS yang digunakan untuk mengenkripsi rahasia.

Anda dapat memilih pengaturan lain untuk memenuhi kebutuhan Anda. Untuk informasi selengkapnya tentang pengaturan yang tersedia saat Anda membuat instans DB, lihat Pengaturan untuk instans DB. Untuk informasi selengkapnya tentang pengaturan yang tersedia saat Anda mengubah instans DB, lihat Pengaturan untuk instans DB.

Contoh ini membuat instans DB dan menentukan bahwa RDS mengelola kata sandi pengguna utama di Secrets Manager. Rahasianya dienkripsi menggunakan kunci KMS yang disediakan oleh Secrets Manager.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --engine mysql \
    --engine-version 8.0.30 \
    --db-instance-class db.r5b.large \
    --allocated-storage 200 \
    --manage-master-user-password
```

Untuk Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --engine mysql ^
    --engine-version 8.0.30 ^
    --db-instance-class db.r5b.large ^
    --allocated-storage 200 ^
    --manage-master-user-password
```

API RDS

Untuk menentukan bahwa RDS mengelola kata sandi pengguna utama di Secrets Manager, atur parameter `ManageMasterUserPassword` ke `true` di salah satu operasi API RDS berikut:

- CreateDBInstance
- ModifyDBInstance
- DipulihkanB S3 InstanceFrom

Jika Anda mengatur parameter `ManageMasterUserPassword` ke `true` di salah satu operasi ini, RDS akan menghasilkan kata sandi pengguna utama dan mengelolanya sepanjang siklus pemakaiannya di Secrets Manager.

Untuk mengenkripsi rahasia, Anda dapat menentukan kunci yang dikelola pelanggan atau menggunakan kunci KMS default yang disediakan oleh Secrets Manager. Gunakan parameter `MasterUserSecretKmsKeyId` untuk menentukan kunci yang dikelola pelanggan. Pengidentifikasi kunci AWS KMS adalah kunci ARN, ID kunci, alias ARN, atau nama alias untuk kunci KMS. Untuk menggunakan kunci KMS di Akun AWS yang berbeda, tentukan ARN kunci atau ARN alias. Setelah RDS mengelola kredensial basis data untuk instans DB, Anda tidak dapat mengubah kunci KMS yang digunakan untuk mengenkripsi rahasia.

Mengelola kata sandi pengguna utama untuk klaster DB Multi-AZ dengan Secrets Manager

Anda dapat mengonfigurasi manajemen RDS kata sandi pengguna utama di Secrets Manager saat Anda melakukan tindakan berikut:

- Membuat klaster DB Multi-AZ
- Mengubah klaster basis data Multi-AZ

Anda dapat menggunakan konsol RDS, API AWS CLI, atau RDS untuk melakukan tindakan ini.

Konsol

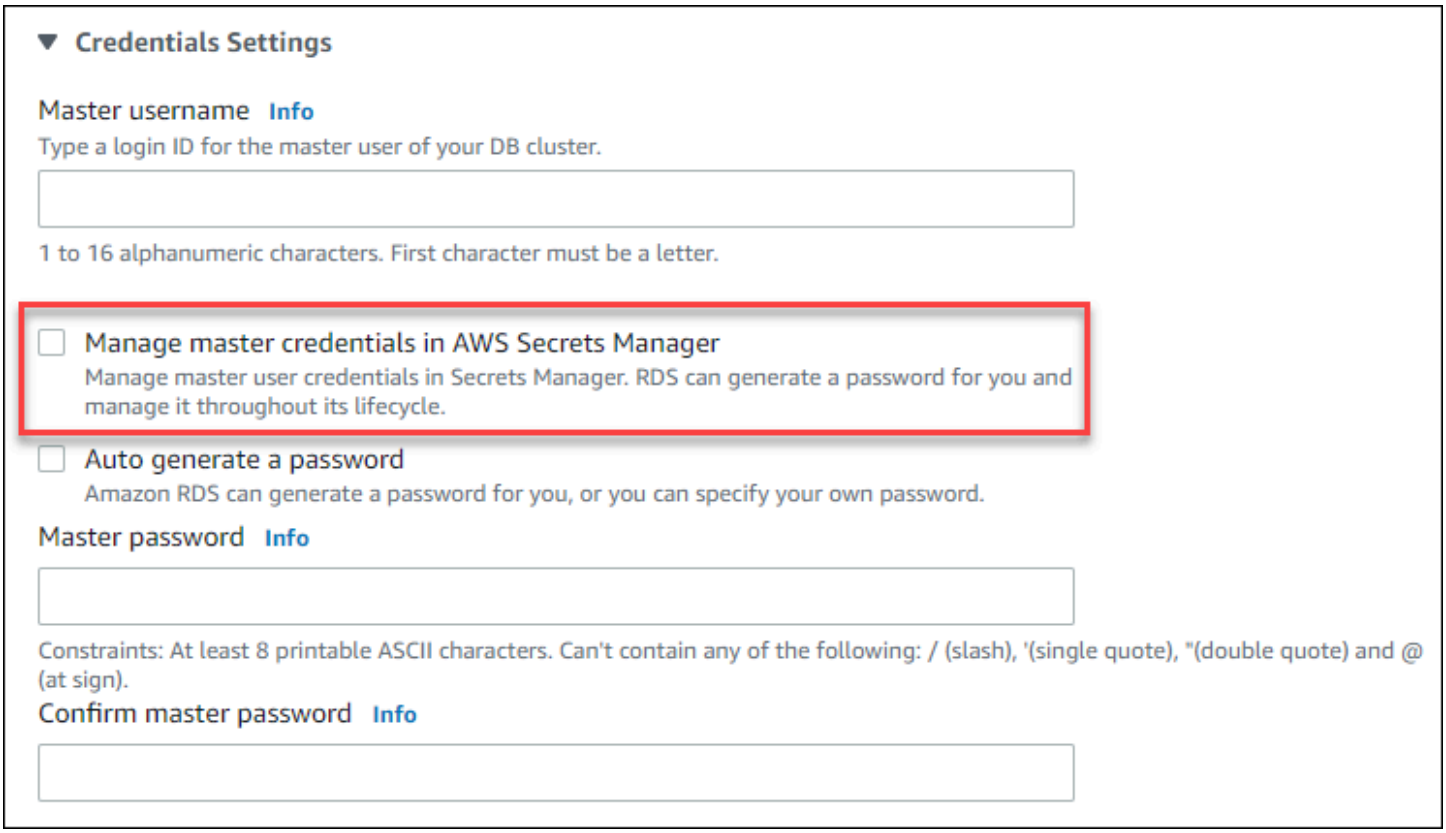
Ikuti petunjuk untuk membuat atau mengubah klaster DB Multi-AZ dengan konsol RDS:

- Membuat klaster DB

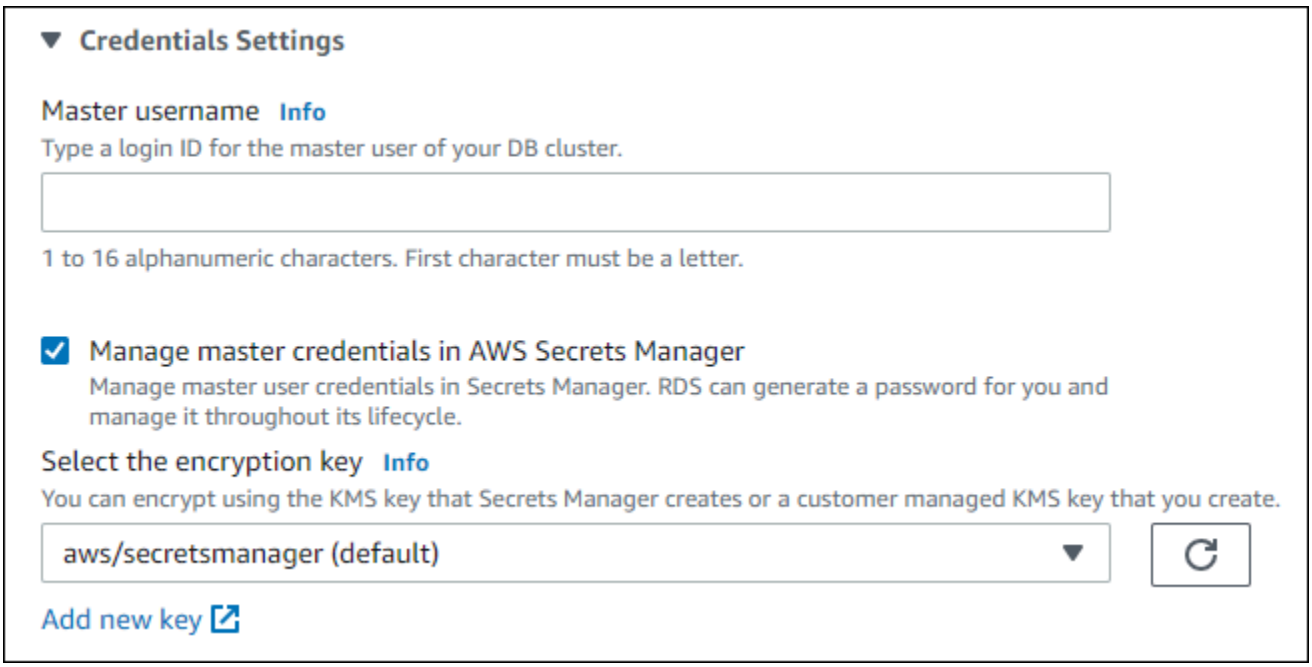
- Mengubah klaster basis data Multi-AZ

Saat menggunakan konsol RDS untuk melakukan salah satu operasi ini, Anda dapat menentukan bahwa kata sandi pengguna utama dikelola oleh RDS di Secrets Manager. Untuk melakukannya saat membuat klaster DB, pilih Kelola kredensial utama di AWS Secrets Manager dalam Pengaturan kredensial. Saat Anda mengubah klaster DB, pilih Kelola kredensial utama di AWS Secrets Manager dalam Pengaturan.

Gambar berikut adalah contoh pengaturan Kelola kredensial utama di AWS Secrets Manager saat Anda membuat klaster DB.



Jika Anda memilih opsi ini, RDS akan menghasilkan kata sandi pengguna utama dan mengelolanya sepanjang siklus pemakaiannya di Secrets Manager.



Anda dapat memilih untuk mengenkripsi rahasia dengan kunci KMS yang disediakan Secrets Manager atau dengan kunci yang dikelola pelanggan yang Anda buat. Setelah RDS mengelola kredensial basis data untuk klaster DB, Anda tidak dapat mengubah kunci KMS yang digunakan untuk mengenkripsi rahasia.

Anda dapat memilih pengaturan lain untuk memenuhi kebutuhan Anda.

Untuk informasi selengkapnya tentang pengaturan yang tersedia saat Anda membuat klaster DB Multi-AZ, lihat Pengaturan untuk membuat klaster DB Multi-AZ. Untuk informasi selengkapnya tentang pengaturan yang tersedia saat Anda mengubah klaster DB Multi-AZ, lihat Setelan untuk mengubah klaster basis data Multi-AZ.

AWS CLI

Untuk menentukan bahwa RDS mengelola kata sandi pengguna utama di Secrets Manager, tentukan opsi `--manage-master-user-password` di salah satu perintah berikut:

- create-db-cluster
- modify-db-cluster

Jika Anda menentukan opsi `--manage-master-user-password` dalam perintah ini, RDS akan menghasilkan kata sandi pengguna utama dan mengelolanya sepanjang siklus pemakaiannya di Secrets Manager.

Untuk mengenkripsi rahasia, Anda dapat menentukan kunci yang dikelola pelanggan atau menggunakan kunci KMS default yang disediakan oleh Secrets Manager. Gunakan opsi `--master-user-secret-kms-key-id` untuk menentukan kunci yang dikelola pelanggan. Pengidentifikasi kunci AWS KMS adalah kunci ARN, ID kunci, alias ARN, atau nama alias untuk kunci KMS. Untuk menggunakan kunci KMS yang berbeda Akun AWS, tentukan kunci ARN atau alias ARN. Setelah RDS mengelola kredensial basis data untuk klaster DB, Anda tidak dapat mengubah kunci KMS yang digunakan untuk mengenkripsi rahasia.

Anda dapat memilih pengaturan lain untuk memenuhi kebutuhan Anda.

Untuk informasi selengkapnya tentang pengaturan yang tersedia saat Anda membuat klaster DB Multi-AZ, lihat Pengaturan untuk membuat klaster DB Multi-AZ. Untuk informasi selengkapnya tentang pengaturan yang tersedia saat Anda mengubah klaster DB Multi-AZ, lihat Setelan untuk mengubah klaster basis data Multi-AZ.

Contoh ini membuat klaster DB Multi-AZ dan menentukan bahwa RDS mengelola kata sandi di Secrets Manager. Rahasianya dienkripsi menggunakan kunci KMS yang disediakan oleh Secrets Manager.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds create-db-cluster \
   --db-cluster-identifier mysql-multi-az-db-cluster \
   --engine mysql \
   --engine-version 8.0.28  \
   --backup-retention-period 1  \
   --allocated-storage 4000 \
   --storage-type io1 \
   --iops 10000 \
   --db-cluster-instance-class db.r6gd.xlarge \
   --manage-master-user-password
```

Untuk Windows:

```
aws rds create-db-cluster ^
   --db-cluster-identifier mysql-multi-az-db-cluster ^
   --engine mysql ^
   --engine-version 8.0.28 ^
   --backup-retention-period 1 ^
   --allocated-storage 4000 ^
```

```
    --storage-type io1 ^
    --iops 10000 ^
    --db-cluster-instance-class db.r6gd.xlarge ^
    --manage-master-user-password
```

API RDS

Untuk menentukan bahwa RDS mengelola kata sandi pengguna utama di Secrets Manager, atur parameter `ManageMasterUserPassword` ke `true` di salah satu operasi berikut:

- CreateDBCluster
- ModifyDBCluster

Jika Anda mengatur parameter `ManageMasterUserPassword` ke `true` di salah satu operasi ini, RDS akan menghasilkan kata sandi pengguna utama dan mengelolanya sepanjang siklus pemakaiannya di Secrets Manager.

Untuk mengenkripsi rahasia, Anda dapat menentukan kunci yang dikelola pelanggan atau menggunakan kunci KMS default yang disediakan oleh Secrets Manager. Gunakan parameter `MasterUserSecretKmsKeyId` untuk menentukan kunci yang dikelola pelanggan. Pengidentifikasi kunci AWS KMS adalah kunci ARN, ID kunci, alias ARN, atau nama alias untuk kunci KMS. Untuk menggunakan kunci KMS di Akun AWS yang berbeda, tentukan ARN kunci atau ARN alias. Setelah RDS mengelola kredensial basis data untuk klaster DB, Anda tidak dapat mengubah kunci KMS yang digunakan untuk mengenkripsi rahasia.

Merotasi rahasia kata sandi pengguna utama untuk instans DB

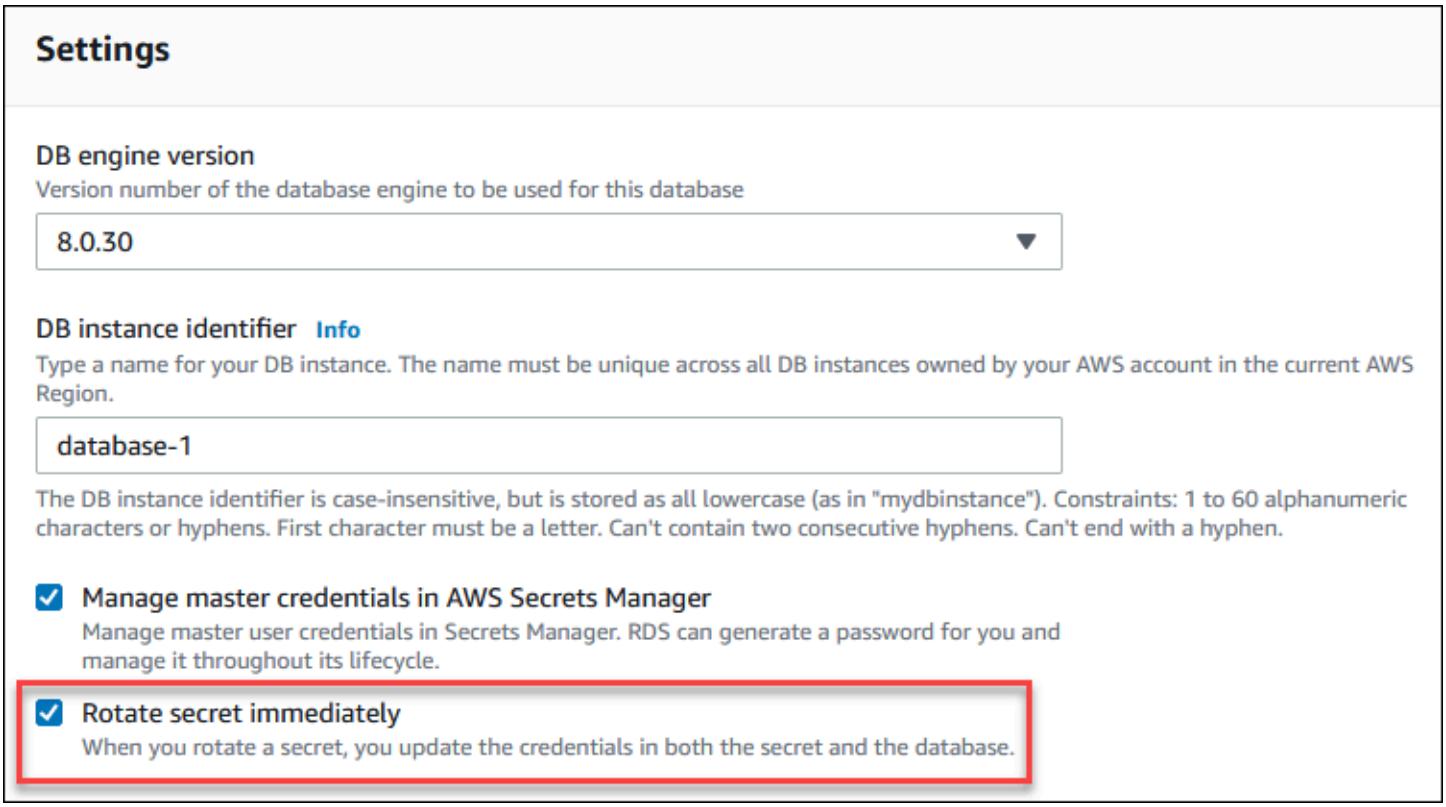
Ketika RDS merotasi rahasia kata sandi pengguna utama, Secrets Manager akan menghasilkan versi rahasia baru untuk rahasia yang sudah ada. Rahasia versi baru berisi kata sandi pengguna utama baru. Amazon RDS mengubah kata sandi pengguna utama untuk instans DB agar sesuai dengan kata sandi versi rahasia baru.

Anda dapat segera merotasi rahasia, alih-alih menunggu rotasi yang dijadwalkan. Untuk merotasi rahasia kata sandi pengguna utama di Secrets Manager, ubah instans DB. Untuk informasi tentang cara memodifikasi instans DB, lihat Memodifikasi instans DB Amazon RDS.

Anda dapat memutar rahasia kata sandi pengguna master segera dengan konsol RDS, API AWS CLI, atau RDS. Kata sandi baru selalu sepanjang 28 karakter dan berisi setidaknya satu karakter huruf besar dan kecil, satu angka, dan satu tanda baca.

Konsol

Untuk merotasi rahasia kata sandi pengguna utama menggunakan konsol RDS, ubah instans DB dan pilih Rotasi rahasia secara langsung di Pengaturan.



Ikuti petunjuk untuk mengubah instans DB dengan konsol RDS di Memodifikasi instans DB Amazon RDS. Anda harus memilih Terapkan langsung di halaman konfirmasi.

AWS CLI

Untuk memutar rahasia kata sandi pengguna master menggunakan AWS CLI, gunakan modify-db-instanceperintah dan tentukan `--rotate-master-user-password` opsi. Anda harus menentukan opsi `--apply-immediately` saat merotasi kata sandi utama.

Contoh ini merotasi rahasia kata sandi pengguna utama.

Example

Untuk Linux, macOS, atau Unix:

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

```
    --rotate-master-user-password \
    --apply-immediately
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --rotate-master-user-password ^
    --apply-immediately
```

API RDS

Anda dapat merotasi rahasia kata sandi pengguna utama menggunakan operasi ModifyDBInstance dan mengatur parameter `RotateMasterUserPassword` ke `true`. Anda harus mengatur parameter `ApplyImmediately` ke `true` saat merotasi kata sandi utama.

Merotasi rahasia kata sandi pengguna utama untuk klaster DB Multi-AZ

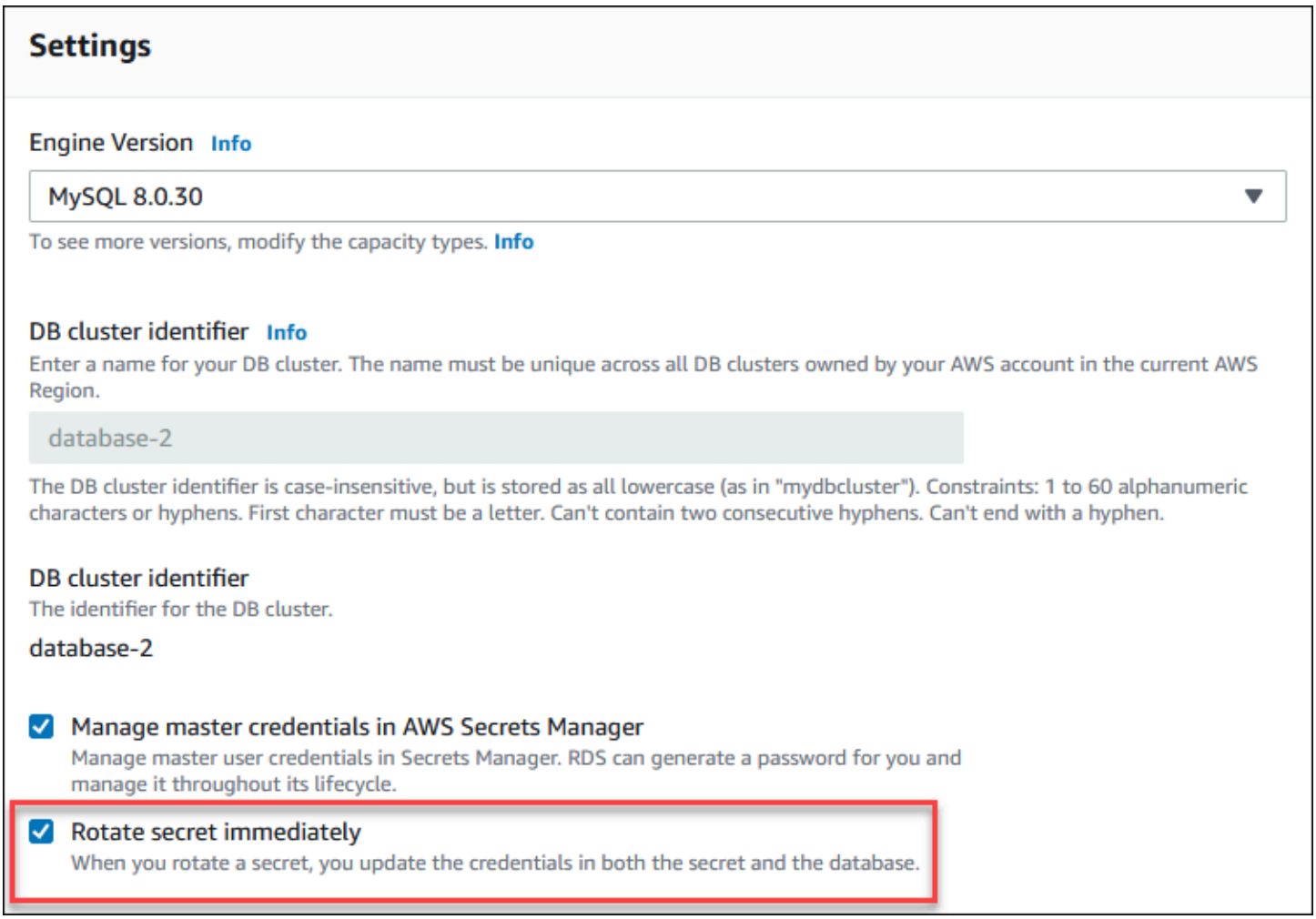
Ketika RDS merotasi rahasia kata sandi pengguna utama, Secrets Manager akan menghasilkan versi rahasia baru untuk rahasia yang sudah ada. Rahasia versi baru berisi kata sandi pengguna utama baru. Amazon RDS mengubah kata sandi pengguna utama untuk klaster DB Multi-AZ agar sesuai dengan kata sandi versi rahasia baru.

Anda dapat segera merotasi rahasia, alih-alih menunggu rotasi yang dijadwalkan. Untuk merotasi rahasia kata sandi pengguna utama di Secrets Manager, ubah klaster DB Multi-AZ. Untuk informasi tentang cara mengubah klaster DB Multi-AZ, lihat Mengubah klaster basis data Multi-AZ.

Anda dapat memutar rahasia kata sandi pengguna master segera dengan konsol RDS, API AWS CLI, atau RDS. Kata sandi baru selalu sepanjang 28 karakter dan berisi setidaknya satu karakter huruf besar dan kecil, satu angka, dan satu tanda baca.

Konsol

Untuk merotasi rahasia kata sandi pengguna utama menggunakan konsol RDS, ubah klaster DB Multi-AZ dan pilih Rotasi rahasia secara langsung di Pengaturan.



Ikuti petunjuk untuk mengubah klaster DB Multi-AZ dengan konsol RDS di Mengubah klaster basis data Multi-AZ. Anda harus memilih Terapkan langsung di halaman konfirmasi.

AWS CLI

Untuk memutar rahasia kata sandi pengguna master menggunakan AWS CLI, gunakan modify-db-clusterperintah dan tentukan `--rotate-master-user-password` opsi. Anda harus menentukan opsi `--apply-immediately` saat merotasi kata sandi utama.

Contoh ini merotasi rahasia kata sandi pengguna utama.

Example

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-cluster \
```

```
    --db-cluster-identifier mydbcluster \
    --rotate-master-user-password \
    --apply-immediately
```

Untuk Windows:

```
aws rds modify-db-cluster ^
    --db-cluster-identifier mydbcluster ^
    --rotate-master-user-password ^
    --apply-immediately
```

API RDS

Anda dapat merotasi rahasia kata sandi pengguna utama menggunakan operasi ModifyDBCluster dan mengatur parameter `RotateMasterUserPassword` ke `true`. Anda harus mengatur parameter `ApplyImmediately` ke `true` saat merotasi kata sandi utama.

Melihat detail tentang rahasia untuk instans DB

Anda dapat mengambil rahasia Anda menggunakan konsol (https://console.aws.amazon.com/secretsmanager/) atau perintah AWS CLI (get-secret-valueSecrets Manager).

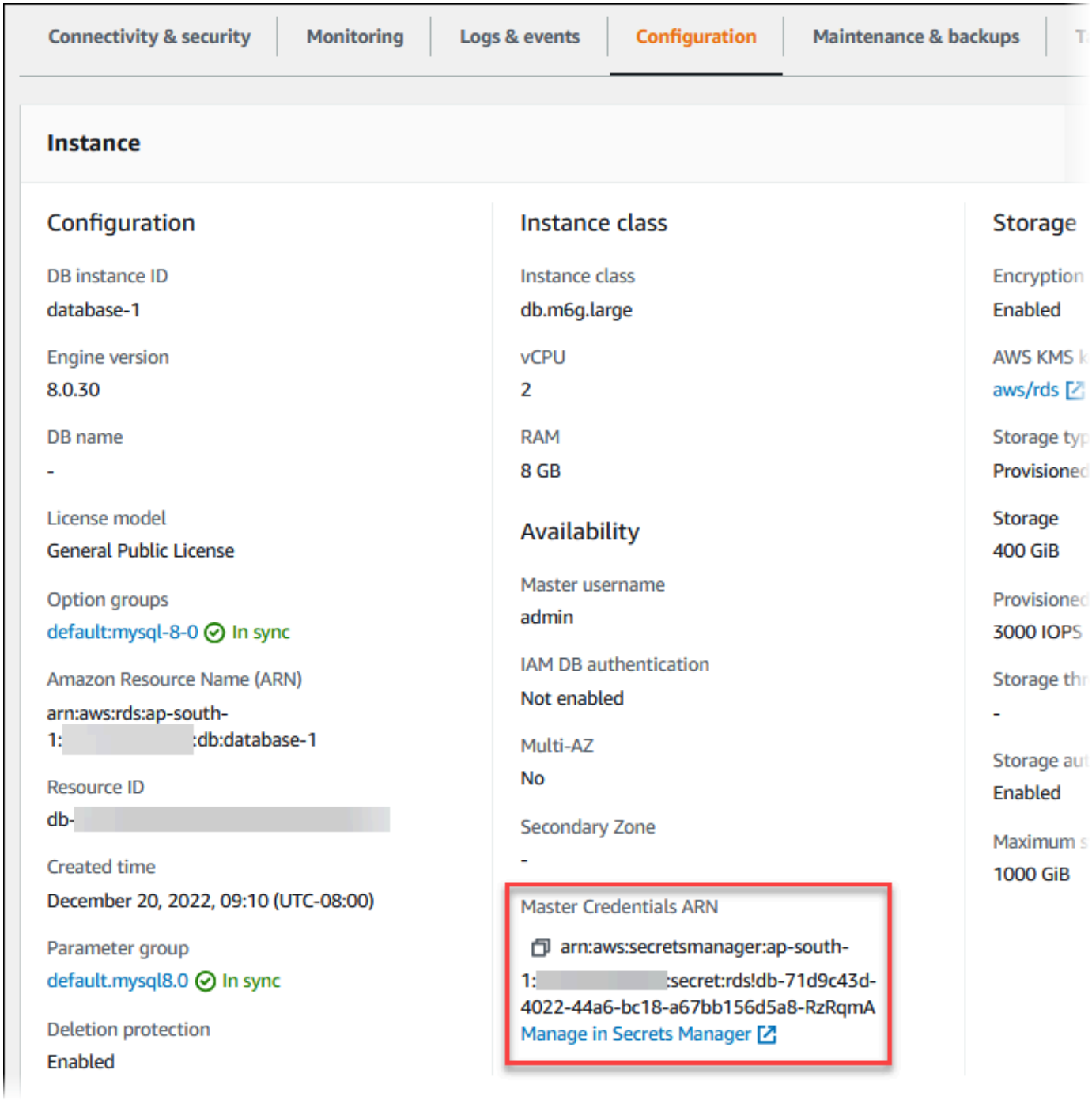
Anda dapat menemukan Amazon Resource Name (ARN) dari rahasia yang dikelola oleh RDS di Secrets Manager dengan konsol RDS, AWS CLI, atau RDS API.

Konsol

Untuk melihat detail tentang rahasia yang dikelola oleh RDS di Secrets Manager

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih basis data.
3. Pilih nama instans DB untuk menampilkan detailnya.
4. Pilih tab Konfigurasi.

   Di ARN Kredensial Utama, Anda dapat melihat ARN rahasia.



Anda dapat mengikuti tautan Mengelola di Secrets Manager untuk melihat dan mengelola rahasia di konsol Secrets Manager.

AWS CLI

Anda dapat menggunakan perintah describe-db-instancesRDS CLI untuk menemukan informasi berikut tentang rahasia yang dikelola oleh RDS di Secrets Manager:

- `SecretArn` – ARN rahasia
- `SecretStatus` – Status rahasia

  Kemungkinan nilai statusnya meliputi:

  - `creating` – Rahasia sedang dibuat.
  - `active` – Rahasia tersedia untuk penggunaan normal dan rotasi.
  - `rotating` – Rahasia sedang dirotasi.
  - `impaired` – Rahasia dapat digunakan untuk mengakses kredensial basis data, tetapi tidak dapat dirotasi. Rahasia mungkin memiliki status ini jika, misalnya, izin diubah sehingga RDS tidak dapat lagi mengakses rahasia atau kunci KMS untuk rahasia tersebut.

    Ketika rahasia memiliki status ini, Anda dapat memperbaiki kondisi yang menyebabkan status tersebut. Jika Anda memperbaiki kondisi yang menyebabkan status, status tersebut tetap `impaired` hingga rotasi berikutnya. Sebagai alternatif, Anda dapat mengubah instans DB untuk menonaktifkan manajemen otomatis kredensial basis data, dan kemudian mengubah instans DB lagi untuk mengaktifkan manajemen otomatis kredensial basis data. Untuk memodifikasi instans DB, gunakan `--manage-master-user-password` opsi dalam modify-db-instanceperintah.
- `KmsKeyId` – ARN kunci KMS yang digunakan untuk mengenkripsi rahasia

Tentukan `--db-instance-identifier` opsi untuk menampilkan output untuk instans DB tertentu. Contoh ini menunjukkan output untuk rahasia yang digunakan oleh instans DB.

Example

```
aws rds describe-db-instances --db-instance-identifier mydbinstance
```

Berikut ini adalah contoh output untuk rahasia:

```
"MasterUserSecret": {
                "SecretArn": "arn:aws:secretsmanager:eu-west-1:123456789012:secret:rds!
db-033d7456-2c96-450d-9d48-f5de3025e51c-xmJRDx",
                "SecretStatus": "active",
                "KmsKeyId": "arn:aws:kms:eu-
west-1:123456789012:key/0987dcba-09fe-87dc-65ba-ab0987654321"
            }
```

Ketika Anda memiliki ARN rahasia, Anda dapat melihat detail tentang rahasia menggunakan perintah get-secret-valueSecrets Manager CLI.

Contoh ini menunjukkan detail untuk rahasia dalam output contoh sebelumnya.

Example

Untuk Linux, macOS, atau Unix:

```
aws secretsmanager get-secret-value \
    --secret-id 'arn:aws:secretsmanager:eu-west-1:123456789012:secret:rds!
db-033d7456-2c96-450d-9d48-f5de3025e51c-xmJRDx'
```

Untuk Windows:

```
aws secretsmanager get-secret-value ^
    --secret-id 'arn:aws:secretsmanager:eu-west-1:123456789012:secret:rds!
db-033d7456-2c96-450d-9d48-f5de3025e51c-xmJRDx'
```

API RDS

Anda dapat melihat ARN, status, dan kunci KMS untuk rahasia yang dikelola oleh RDS di Secrets Manager dengan menggunakan operasi DescribeDBInstances dan mengatur parameter `DBInstanceIdentifier` ke ID instans DB. Detil tentang rahasia disertakan dalam output.

Ketika Anda memiliki ARN rahasia, Anda dapat melihat detail tentang rahasia menggunakan operasi GetSecretValueSecrets Manager.

Melihat detail tentang rahasia untuk klaster DB Multi-AZ

Anda dapat mengambil rahasia Anda menggunakan konsol (https://console.aws.amazon.com/secretsmanager/) atau perintah AWS CLI (get-secret-valueSecrets Manager).

Anda dapat menemukan Amazon Resource Name (ARN) dari rahasia yang dikelola oleh RDS di Secrets Manager dengan konsol RDS AWS CLI,, atau RDS API.

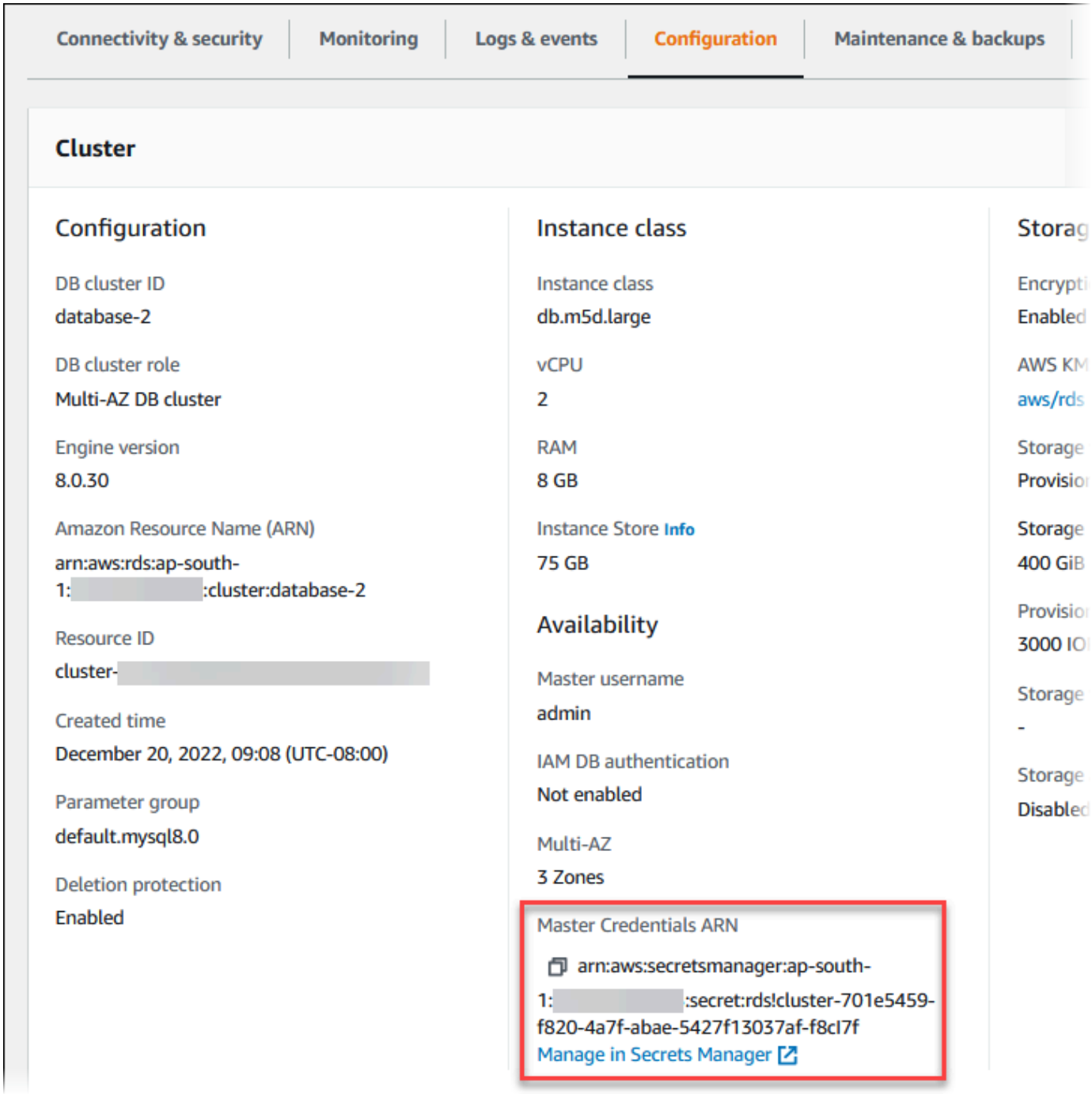
Konsol

Untuk melihat detail tentang rahasia yang dikelola oleh RDS di Secrets Manager

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih basis data.

3. Pilih nama klaster DB Multi-AZ untuk menampilkan detailnya.
4. Pilih tab Konfigurasi.

   Di ARN Kredensial Utama, Anda dapat melihat ARN rahasia.



Anda dapat mengikuti tautan Mengelola di Secrets Manager untuk melihat dan mengelola rahasia di konsol Secrets Manager.

AWS CLI

Anda dapat menggunakan AWS CLI describe-db-clustersperintah RDS untuk menemukan informasi berikut tentang rahasia yang dikelola oleh RDS Aurora Manager:

- `SecretArn` – ARN rahasia
- `SecretStatus` – Status rahasia

  Kemungkinan nilai statusnya meliputi:

  - `creating` – Rahasia sedang dibuat.
  - `active` – Rahasia tersedia untuk penggunaan normal dan rotasi.
  - `rotating` – Rahasia sedang dirotasi.
  - `impaired` – Rahasia dapat digunakan untuk mengakses kredensial basis data, tetapi tidak dapat dirotasi. Rahasia mungkin memiliki status ini jika, misalnya, izin diubah sehingga RDS tidak dapat lagi mengakses rahasia atau kunci KMS untuk rahasia tersebut.

    Ketika rahasia memiliki status ini, Anda dapat memperbaiki kondisi yang menyebabkan status tersebut. Jika Anda memperbaiki kondisi yang menyebabkan status, status tersebut tetap `impaired` hingga rotasi berikutnya. Sebagai alternatif, Anda dapat mengubah klaster DB untuk menonaktifkan manajemen otomatis kredensial basis data, dan kemudian mengubah klaster DB lagi untuk mengaktifkan manajemen otomatis kredensial basis data. Untuk memodifikasi cluster DB, gunakan `--manage-master-user-password` opsi dalam modify-db-clusterperintah.
- `KmsKeyId` – ARN kunci KMS yang digunakan untuk mengenkripsi rahasia

Tentukan opsi `--db-cluster-identifier` untuk menampilkan output untuk klaster DB tertentu. Contoh ini menunjukkan output untuk rahasia yang digunakan oleh klaster DB.

Example

```
aws rds describe-db-clusters --db-cluster-identifier mydbcluster
```

Contoh berikut menunjukkan output untuk rahasia:

```
"MasterUserSecret": {
                "SecretArn": "arn:aws:secretsmanager:eu-west-1:123456789012:secret:rds!
cluster-033d7456-2c96-450d-9d48-f5de3025e51c-xmJRDx",
                "SecretStatus": "active",
```

```
                "KmsKeyId": "arn:aws:kms:eu-
west-1:123456789012:key/0987dcba-09fe-87dc-65ba-ab0987654321"
            }
```

Ketika Anda memiliki ARN rahasia, Anda dapat melihat detail tentang rahasia menggunakan perintah get-secret-valueSecrets Manager CLI.

Contoh ini menunjukkan detail untuk rahasia dalam output contoh sebelumnya.

Example

Untuk Linux, macOS, atau Unix:

```
aws secretsmanager get-secret-value \
    --secret-id 'arn:aws:secretsmanager:eu-west-1:123456789012:secret:rds!
cluster-033d7456-2c96-450d-9d48-f5de3025e51c-xmJRDx'
```

Untuk Windows:

```
aws secretsmanager get-secret-value ^
    --secret-id 'arn:aws:secretsmanager:eu-west-1:123456789012:secret:rds!
cluster-033d7456-2c96-450d-9d48-f5de3025e51c-xmJRDx'
```

API RDS

Anda dapat melihat ARN, status, dan kunci KMS untuk rahasia yang dikelola oleh RDS di Secrets Manager menggunakan operasi RDS DescribeDBClusters dan mengatur parameter `DBClusterIdentifier` ke ID klaster DB. Detil tentang rahasia disertakan dalam output.

Ketika Anda memiliki ARN rahasia, Anda dapat melihat detail tentang rahasia menggunakan operasi GetSecretValueSecrets Manager.

Ketersediaan Wilayah dan versi

Ketersediaan dan dukungan fitur bervariasi di seluruh versi khusus dari setiap mesin basis data dan di seluruh Wilayah AWS. Untuk informasi selengkapnya tentang ketersediaan versi dan Wilayah dengan integrasi Secrets Manager dengan Amazon RDS, lihat Integrasi Secrets Manager.

Perlindungan data di Amazon RDS

Model tanggung jawab bersama AWS berlaku untuk perlindungan data di Amazon Relational Database Service. Sebagaimana diuraikan dalam model ini, AWS bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk memelihara kendali atas isi yang dihost 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, sebaiknya lindungi kredensial Akun AWS dan siapkan untuk masing-masing pengguna 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 sumber daya AWS. Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan pengelogan aktivitas pengguna dengan AWS CloudTrail.
- Gunakan solusi enkripsi AWS, bersama semua kontrol keamanan bawaan dalam Layanan AWS.
- Gunakan layanan keamanan terkelola lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-2 ketika mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi yang lebih lengkap tentang titik akhir FIPS yang tersedia di Standar Pemrosesan Informasi Federal (FIPS) 140-2.

Kami sangat menganjurkan supaya Anda tidak memasukkan informasi rahasia atau sensitif, seperti alamat email pelanggan, ke dalam tag atau bidang teks isian bebas seperti bidang Nama. Hal ini mencakup ketika Anda bekerja dengan Amazon RDS atau Layanan AWS lain dengan menggunakan konsol, API, AWS CLI, atau AWS SDK. Data apa pun yang Anda masukkan ke dalam tag atau bidang teks isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan atau 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.

Topik

Melindungi data dengan menggunakan enkripsi

Anda dapat mengaktifkan enkripsi untuk sumber daya basis data. Anda juga dapat mengenkripsi koneksi ke klaster basis data.

Topik

Mengenkripsi sumber daya Amazon RDS

Amazon RDS dapat mengenkripsi instans DB Amazon RDS Anda. Data yang dienkripsi saat diam termasuk penyimpanan dasar untuk instans DB, pencadangan otomatisnya, replika baca, dan snapshot.

Instans DB terenkripsi Amazon RDS menggunakan algoritma AES-256 standar industri untuk mengenkripsi data Anda di server yang meng-host instans DB Amazon RDS Anda. Setelah data Anda dienkripsi, Amazon RDS menangani autentikasi akses dan dekripsi data Anda secara transparan dengan dampak minimal terhadap performa. Anda tidak perlu memodifikasi aplikasi klien basis data untuk menggunakan enkripsi.

> **Note**
>
> Untuk instans DB terenkripsi dan tidak terenkripsi, data yang sedang transit antara sumber dan replika baca dienkripsi, bahkan saat mereplikasi di seluruh Wilayah. AWS

Topik

- Mengenkripsi instans DB
- Menentukan apakah enkripsi untuk instans DB diaktifkan
- Ketersediaan enkripsi Amazon RDS
- Enkripsi dalam bergerak
-

Ikhtisar mengenkripsi sumber daya Amazon RDS

Instans DB terenkripsi oleh Amazon RDS memberikan lapisan perlindungan data tambahan dengan mengamankan data Anda dari akses yang tidak sah ke penyimpanan yang mendasari. Anda dapat menggunakan enkripsi Amazon RDS untuk meningkatkan perlindungan data di aplikasi Anda yang di-deploy di cloud, dan untuk memenuhi persyaratan kepatuhan enkripsi diam.

Untuk instans DB terenkripsi Amazon RDS, semua log, cadangan, dan snapshot akan dienkripsi. Amazon RDS menggunakan AWS KMS key untuk mengenkripsi sumber daya ini. Untuk informasi selengkapnya tentang kunci KMS, lihat AWS KMS keys di Panduan Developer AWS Key Management Service dan Manajemen AWS KMS key. Jika Anda menyalin snapshot terenkripsi, Anda dapat menggunakan kunci KMS yang berbeda untuk mengenkripsi snapshot target, bukan yang digunakan untuk mengenkripsi snapshot sumber.

Replika baca instans terenkripsi Amazon RDS harus dienkripsi menggunakan kunci KMS yang sama dengan instans DB utama ketika keduanya berada di Wilayah yang sama. AWS Jika instans DB utama dan replika baca berada di AWS Wilayah yang berbeda, Anda mengenkripsi replika baca menggunakan kunci KMS untuk Wilayah tersebut. AWS

Anda dapat menggunakan Kunci yang dikelola AWS, atau Anda dapat membuat kunci yang dikelola pelanggan. Untuk mengelola kunci yang dikelola pelanggan yang digunakan untuk mengenkripsi dan mendekripsi sumber daya Amazon RDS, gunakan AWS Key Management Service (AWS KMS). AWS KMS menggabungkan perangkat keras dan perangkat lunak yang aman dengan ketersediaan tinggi untuk menyediakan sistem manajemen kunci yang diskalakan untuk cloud. Dengan menggunakan AWS KMS, Anda dapat membuat kunci terkelola pelanggan dan menentukan kebijakan yang mengontrol bagaimana kunci yang dikelola pelanggan ini dapat digunakan. AWS KMS mendukung CloudTrail, sehingga Anda dapat mengaudit penggunaan kunci KMS untuk memverifikasi bahwa kunci yang dikelola pelanggan digunakan dengan tepat. Anda dapat menggunakan kunci yang dikelola pelanggan dengan Amazon Aurora dan AWS layanan yang didukung seperti Amazon S3, Amazon EBS, dan Amazon Redshift. Untuk daftar layanan yang terintegrasi AWS KMS, lihat Integrasi AWS Layanan.

Amazon RDS juga mendukung enkripsi sebuah instans DB Oracle atau SQL Server dengan Enkripsi Data Transparan (TDE). TDE dapat digunakan dengan enkripsi RDS saat diam, meskipun menggunakan enkripsi TDE dan RDS saat diam secara bersamaan dapat sedikit memengaruhi kinerja basis data Anda. Anda harus mengelola kunci yang berbeda untuk setiap metode enkripsi. Untuk informasi lebih lanjut tentang TDE, lihat Enkripsi Data Transparan Oracle atau Dukungan untuk Enkripsi Data Transparan di SQL Server.

Mengenkripsi instans DB

Untuk mengenkripsi instans DB baru, pilih Aktifkan enkripsi di konsol Amazon RDS. Untuk informasi tentang pembuatan instans DB, lihat Membuat instans DB Amazon RDS.

Jika Anda menggunakan create-db-instance AWS CLI perintah untuk membuat instance DB terenkripsi, atur parameternya. `--storage-encrypted` Jika Anda menggunakan Operasi API CreateDBInstance, atur parameter `StorageEncrypted` ke true.

Saat Anda membuat instans DB terenkripsi, Anda dapat memilih kunci yang dikelola pelanggan atau Kunci yang dikelola AWS untuk Amazon RDS guna mengenkripsi instans DB Anda. Jika Anda tidak menentukan pengenal kunci untuk kunci yang dikelola pelanggan, Amazon RDS menggunakan Kunci yang dikelola AWS untuk instans DB baru Anda. Amazon RDS membuat RDS Kunci yang dikelola AWS untuk Amazon untuk akun Anda AWS . AWS Akun Anda memiliki RDS Amazon yang berbeda Kunci yang dikelola AWS untuk setiap AWS Wilayah.

Untuk informasi selengkapnya tentang kunci KMS, lihat AWS KMS keys di Panduan DeveloperAWS Key Management Service .

Setelah Anda membuat instans DB terenkripsi, Anda tidak dapat mengubah kunci KMS yang digunakan oleh instans DB tersebut. Oleh karena itu, pastikan untuk menentukan persyaratan kunci KMS Anda sebelum membuat instans DB terenkripsi Anda.

Jika Anda menggunakan AWS CLI `create-db-instance` perintah untuk membuat instans DB terenkripsi dengan kunci yang dikelola pelanggan, setel `--kms-key-id` parameter ke pengidentifikasi kunci apa pun untuk kunci KMS. Jika Anda menggunakan operasi `CreateDBInstance` API Amazon RDS, atur parameter `KmsKeyId` ke pengidentifikasi kunci mana pun untuk kunci KMS. Untuk menggunakan kunci yang dikelola pelanggan di akun AWS yang berbeda, tentukan ARN kunci atau ARN alias.

> **Important**
>
> Amazon RDS dapat kehilangan akses ke kunci KMS untuk instans DB. Misalnya, RDS kehilangan akses ketika kunci KMS tidak diaktifkan, atau ketika akses RDS ke kunci KMS dicabut. Dalam kasus ini, instans DB terenkripsi menjadi berstatus `inaccessible-encryption-credentials-recoverable`. Instans DB tetap berlangsung selama tujuh hari. Jika selama waktu ini Anda memulai instans DB, instans akan memeriksa apakah kunci KMS aktif, dan jika aktif, maka instans DB akan dipulihkan. Mulai ulang instance DB menggunakan AWS CLI perintah start-db-instanceatau AWS Management Console.
> Jika instans DB tidak dipulihkan, maka instans DB tersebut masuk ke status `inaccessible-encryption-credentials` terminal. Dalam hal ini, Anda hanya dapat memulihkan instans DB dari pencadangan. Sebaiknya selalu aktifkan pencadangan instans DB terenkripsi agar data terenkripsi di basis data Anda tidak hilang.

Menentukan apakah enkripsi untuk instans DB diaktifkan

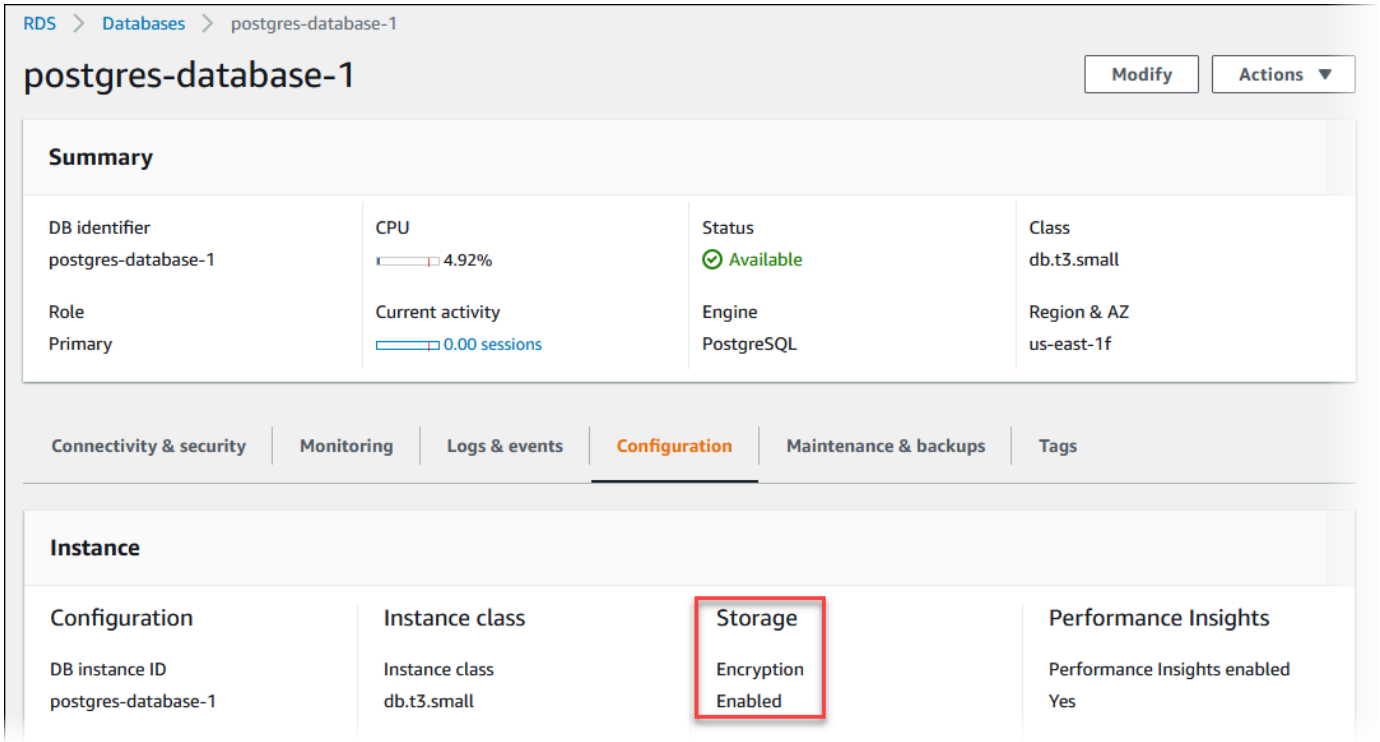
Anda dapat menggunakan AWS Management Console, AWS CLI, atau RDS API untuk menentukan apakah enkripsi saat istirahat diaktifkan untuk instans DB.

Konsol

Untuk menentukan apakah enkripsi diam untuk instans DB diaktifkan atau tidak

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis Data.
3. Pilih nama instans DB yang ingin Anda periksa untuk melihat detailnya.
4. Pilih tab Konfigurasi, dan periksa nilai Enkripsi di bagian Penyimpanan.

   Akan muncul Diaktifkan atau Tidak diaktifkan.



AWS CLI

Untuk menentukan apakah enkripsi saat istirahat diaktifkan untuk instance DB dengan menggunakan AWS CLI, panggil describe-db-instancesperintah dengan opsi berikut:

- `--db-instance-identifier` – Nama instans DB.

Contoh berikut menggunakan kueri untuk mengembalikan salah satu `TRUE` atau `FALSE` mengenai enkripsi saat diam untuk instans DB `mydb`.

Example

```
aws rds describe-db-instances --db-instance-identifier mydb --query "*[].
{StorageEncrypted:StorageEncrypted}" --output text
```

RDS API

Untuk menentukan apakah enkripsi diam untuk instans DB diaktifkan menggunakan Amazon RDS API, panggil operasi DescribeDBInstances dengan parameter berikut:

- `DBInstanceIdentifier` – Nama instans DB.

Ketersediaan enkripsi Amazon RDS

Enkripsi Amazon RDS saat ini tersedia untuk semua mesin basis data dan jenis penyimpanan, kecuali untuk SQL Server Express Edition.

Enkripsi Amazon RDS tersedia untuk sebagian besar kelas instans DB. Tabel berikut mencantumkan kelas instans DB yang tidak mendukung enkripsi Amazon RDS:

Jenis instans	Kelas instans
Tujuan umum (M1)	db.m1.small db.m1.medium db.m1.large db.m1.xlarge
Memori yang dioptimalkan (M2)	db.m2.xlarge db.m2.2xlarge db.m2.4xlarge
Burable (T2)	db.t2.micro

Enkripsi dalam bergerak

AWS menyediakan konektivitas aman dan pribadi antara instans DB dari semua jenis. Selain itu, beberapa tipe instans menggunakan kemampuan offload dari perangkat keras Nitro System yang mendasarinya untuk secara otomatis mengenkripsi lalu lintas dalam transit antar instans. Enkripsi ini menggunakan algoritma Authenticated Encryption with Associated Data (AEAD), dengan enkripsi 256-bit. Tidak ada dampak terhadap performa jaringan. Untuk mendukung enkripsi lalu lintas dalam transit tambahan ini antara instans, persyaratan-persyaratan berikut harus dipenuhi:

- Instans-instans tersebut menggunakan tipe instans berikut:
  - Tujuan umum: M6i, M6iD, M6in, M6idn, M7g
  - Memori yang dioptimalkan: R6i, R6id, R6in, R6idn, R7g, X2idn, X2IEDN, X2IEZN
- Contohnya sama Wilayah AWS.

- Instans-instans tersebut berada dalam VPC yang sama atau VPC yang di-peering yang sama, dan lalu lintas tidak melewati perangkat atau layanan jaringan virtual, seperti penyeimbang beban atau gateway transit.

Batasan berikut ada untuk instans DB dienkripsi dengan Amazon RDS:

- Anda hanya dapat mengenkripsi instans DB Amazon RDS saat Anda membuatnya, bukan setelah instans DB dibuat.

  Namun, karena Anda dapat mengenkripsi salinan snapshot yang tidak dienkripsi, Anda dapat menambahkan enkripsi secara efektif ke instans DB yang tidak terenkripsi. Artinya, Anda dapat membuat snapshot instans DB Anda, lalu membuat salinan terenkripsi dari snapshot tersebut. Anda kemudian dapat memulihkan instans DB dari snapshot terenkripsi, sehingga Anda memiliki salinan terenkripsi dari instans DB asli Anda. Untuk informasi selengkapnya, lihat Menyalin snapshot DB.
- Anda tidak dapat menonaktifkan enkripsi di instans DB terenkripsi.
- Anda tidak dapat membuat snapshot terenkripsi dari instans DB tidak terenkripsi.
- Snapshot instans DB terenkripsi harus dienkripsi menggunakan kunci KMS yang sama seperti instans DB.
- Anda tidak dapat memiliki replika baca terenkripsi dari instans DB yang tidak dienkripsi atau replika baca yang tidak dienkripsi dari instans DB terenkripsi.
- Replika baca terenkripsi harus dienkripsi dengan kunci KMS yang sama dengan instans DB sumber ketika keduanya berada di Wilayah yang sama. AWS
- Anda tidak dapat memulihkan cadangan atau tangkapan layar yang tidak terenkripsi ke instans DB terenkripsi.
- Untuk menyalin snapshot terenkripsi dari satu AWS Wilayah ke wilayah lain, Anda harus menentukan kunci KMS di Wilayah tujuan. AWS Ini karena kunci KMS khusus untuk AWS Wilayah tempat mereka dibuat.

  Snapshot sumber tetap terenkripsi selama proses penyalinan. Amazon RDS menggunakan enkripsi amplop untuk melindungi data selama proses penyalinan. Untuk informasi selengkapnya tentang enkripsi amplop, lihat Enkripsi amplop dalam Panduan Developer AWS Key Management Service .
- Anda tidak dapat menghilangkan enkripsi instans DB yang terenkripsi. Namun, Anda dapat mengekspor data dari instans DB terenkripsi dan mengimpor data ke instans DB tidak terenkripsi.

Manajemen AWS KMS key

Amazon RDS secara otomatis terintegrasi dengan AWS Key Management Service (AWS KMS) untuk manajemen kunci. Amazon RDS menggunakan enkripsi amplop. Untuk informasi selengkapnya tentang enkripsi amplop, lihat Enkripsi amplop di Panduan Developer AWS Key Management Service.

Anda dapat menggunakan dua jenis kunci AWS KMS untuk mengenkripsi instans DB.

- Anda harus membuat kunci yang dikelola pelanggan jika ingin mengontrol kunci KMS sepenuhnya. Untuk informasi selengkapnya tentang kunci yang dikelola pelanggan, lihat Kunci yang dikelola pelanggan di Panduan Developer AWS Key Management Service.

  Anda tidak dapat membagikan snapshot yang telah dienkripsi menggunakan Kunci yang dikelola AWS akun AWS yang membagikan snapshot tersebut.
- Kunci yang dikelola AWS adalah kunci KMS di akun Anda yang dibuat, dikelola, dan digunakan atas nama Anda oleh layanan AWS yang terintegrasi dengan AWS KMS. Secara default, Kunci yang dikelola AWS RDS (`aws/rds`) digunakan untuk enkripsi. Anda tidak dapat mengelola, memutar, atau menghapus Kunci yang dikelola AWS RDS. Untuk informasi selengkapnya tentang Kunci yang dikelola AWS, lihat Kunci yang dikelola AWS di Panduan Developer AWS Key Management Service.

Untuk mengelola kunci KMS yang digunakan untuk instans DB terenkripsi Amazon RDS, gunakan AWS Key Management Service (AWS KMS) di konsol AWS KMS, AWS CLI, atau AWS KMS API. Untuk melihat log audit dari setiap tindakan yang diambil dengan kunci yang dikelola AWS atau pelanggan, gunakan AWS CloudTrail. Untuk informasi selengkapnya tentang rotasi kunci, lihat Merotasi kunci AWS KMS.

> **Important**
>
> Jika Anda menonaktifkan atau mencabut izin ke kunci KMS yang digunakan oleh basis data RDS, RDS akan memasukkan basis data Anda ke dalam status terminal ketika akses ke kunci KMS diperlukan. Perubahan ini dapat dilakukan secara langsung, atau ditangguhkan, tergantung kasus penggunaan yang memerlukan akses ke kunci KMS. Dalam kondisi ini, instans DB tidak lagi tersedia, dan kondisi basis data saat ini tidak dapat dipulihkan. Untuk memulihkan instans DB, Anda harus mengaktifkan kembali akses ke kunci KMS untuk RDS, kemudian memulihkan instans DB dari cadangan terbaru yang tersedia.

Mengotorisasi penggunaan kunci yang dikelola pelanggan

Saat RDS menggunakan kunci yang dikelola pelanggan dalam operasi kriptografi, kunci ini bertindak atas nama pengguna yang membuat atau mengubah sumber daya RDS.

Untuk membuat sumber daya RDS menggunakan kunci yang dikelola pelanggan, pengguna harus memiliki izin untuk memanggil operasi berikut pada kunci yang dikelola pelanggan:

- kms:CreateGrant
- kms:DescribeKey

Anda dapat menentukan izin yang diperlukan ini dalam kebijakan utama, atau dalam kebijakan IAM jika kebijakan kunci memungkinkan hal tersebut.

Anda dapat memperketat kebijakan IAM dalam berbagai cara. Misalnya, jika Anda hanya mengizinkan penggunaan kunci yang dikelola pelanggan untuk permintaan yang berasal dari RDS, Anda dapat menggunakan kunci kondisi kms:ViaService dengan nilai `rds.<region>.amazonaws.com`. Selain itu, Anda juga dapat menggunakan kunci atau nilai di Konteks enkripsi Amazon RDS sebagai kondisi dalam penggunaan kunci yang dikelola pelanggan untuk enkripsi.

Untuk informasi selengkapnya, lihat Mengizinkan pengguna di akun lain untuk menggunakan kunci KMS di Panduan Developer AWS Key Management Service dan Kebijakan kunci di AWS KMS.

Konteks enkripsi Amazon RDS

Saat RDS menggunakan kunci KMS Anda, atau saat Amazon EBS menggunakan kunci KMS atas nama RDS, layanan akan menentukan konteks enkripsi. Konteks enkripsi adalah data terautentikasi tambahan (AAD) yang digunakan oleh AWS KMS untuk memastikan integritas data. Ketika konteks enkripsi ditentukan untuk operasi enkripsi, layanan harus menentukan konteks enkripsi yang sama untuk operasi dekripsi. Jika tidak, dekripsi akan gagal. Konteks enkripsi juga dituliskan ke log AWS CloudTrail untuk membantu Anda memahami alasan penggunaan kunci KMS tertentu. Log CloudTrail Anda mungkin berisi banyak entri yang menjelaskan penggunaan kunci KMS, tetapi konteks enkripsi di setiap entri log dapat membantu Anda menentukan alasan penggunaan tersebut.

Minimal, Amazon RDS selalu menggunakan ID instans DB untuk konteks enkripsi, seperti pada contoh berformat JSON berikut:

```
{ "aws:rds:db-id": "db-CQYSMDPBRZ7BPMH7Y3RTDG5QY" }
```

Konteks enkripsi ini dapat membantu Anda mengidentifikasi instans DB yang digunakan kunci KMS Anda.

Ketika kunci KMS Anda digunakan untuk instans DB dan volume Amazon EBS tertentu, baik ID instans DB maupun ID volume Amazon EBS digunakan untuk konteks enkripsi, seperti pada contoh berformat JSON berikut:

```
{
  "aws:rds:db-id": "db-BRG7VYS3SVIFQW7234EJQOM5RQ",
  "aws:ebs:id": "vol-ad8c6542"
}
```

Anda dapat menggunakan Secure Socket Layer (SSL) atau Transport Layer Security (TLS) dari aplikasi Anda untuk mengenkripsi koneksi ke database yang menjalankan Db2, MariaDB, Microsoft SQL Server, MySQL, Oracle, atau PostgreSQL.

Secara opsional, koneksi SSL/TLS Anda dapat melakukan verifikasi identitas server dengan memvalidasi sertifikat server yang diinstal pada database Anda. Untuk meminta verifikasi identitas server, ikuti proses umum ini:

1. Pilih Otoritas Sertifikat (CA) yang menandatangani sertifikat server DB, untuk basis data Anda. Untuk informasi selengkapnya tentang otoritas sertifikat, lihat Otoritas sertifikat.
2. Unduh paket sertifikat yang akan digunakan saat Anda terhubung ke basis data. Untuk mengunduh paket sertifikat, lihat Bundel sertifikat untuk semua Wilayah AWS dan Bundel sertifikat untuk spesifik Wilayah AWS.

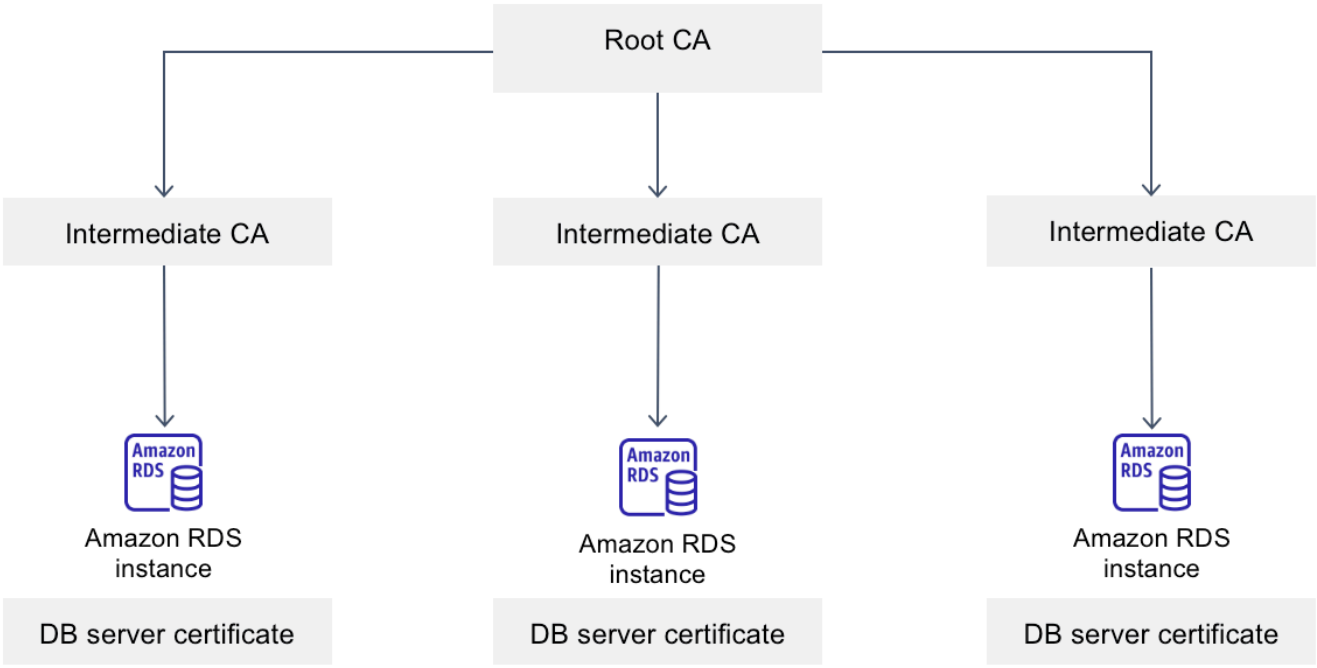
   > **Note**
   >
   > Semua sertifikat hanya tersedia untuk diunduh menggunakan koneksi SSL/TLS.

3. Hubungkan ke basis data menggunakan proses mesin DB Anda untuk menerapkan koneksi SSL/TLS. Setiap mesin DB memiliki prosesnya sendiri untuk menerapkan SSL/TLS. Untuk mempelajari cara menerapkan SSL/TLS untuk basis data Anda, ikuti tautan yang sesuai dengan mesin DB Anda:
   - Menggunakan SSL/TLS dengan instans basis data RDS for Db2
   - Menggunakan SSL/TLS dengan instans basis data MariaDB
   - Menggunakan SSL dengan instans DB Microsoft SQL Server

- Menggunakan SSL/TLS dengan instans DB MySQL
- Menggunakan SSL dengan instans DB RDS for Oracle
- Menggunakan SSL dengan instans DB PostgreSQL

Otoritas sertifikat

Otoritas Sertifikat (CA) adalah sertifikat yang mengidentifikasi CA root di bagian atas rantai sertifikat. CA menandatangani sertifikat server DB, yang diinstal pada setiap instans DB. Sertifikat server DB mengidentifikasi instans DB sebagai server tepercaya.



Amazon RDS menyediakan CA berikut untuk menandatangani sertifikat server DB untuk database.

Otoritas sertifikat (CA)	Deskripsi
rds-ca-2019	Menggunakan otoritas sertifikat dengan algoritma kunci privat RSA 2048 dan algoritma penandata nganan SHA256. CA ini kedaluwarsa pada tahun 2024 dan tidak mendukung rotasi sertifikat server otomatis. Jika Anda menggunakan CA ini dan ingin

Otoritas sertifikat (CA)	Deskripsi
	mempertahankan standar yang sama, kami sarankan Anda beralih ke CA rds-ca-rsa 2048-g1.
rds-ca-rsa2048-g1	Menggunakan otoritas sertifikat dengan algoritma kunci privat RSA 2048 dan algoritma penandata nganan SHA256 di sebagian besar Wilayah AWS. Dalam AWS GovCloud (US) Regions, CA ini menggunakan otoritas sertifikat dengan algoritma kunci pribadi RSA 2048 dan algoritma penandata nganan SHA384. CA ini tetap berlaku lebih lama dari CA rds-ca-2019. CA ini mendukung rotasi sertifikat server otomatis.
rds-ca-rsa4096-g1	Menggunakan otoritas sertifikat dengan algoritma kunci privat RSA 4096 dan algoritma penandata nganan SHA384. CA ini mendukung rotasi sertifikat server otomatis.
rds-ca-ecc384-g1	Menggunakan otoritas sertifikat dengan algoritma kunci privat ECC 384 dan algoritma penandatanganan SHA384. CA ini mendukung rotasi sertifikat server otomatis.

> **Note**
>
> Jika Anda menggunakan AWS CLI, Anda dapat melihat validitas otoritas sertifikat yang tercantum di atas dengan menggunakan deskripsi-sertifikat.

Sertifikat CA ini termasuk dalam paket sertifikat regional dan global. Bila Anda menggunakan rds-ca-rsa 2048-g1, rds-ca-rsa 4096-g1, atau rds-ca-ecc 384-g1 CA dengan database, RDS mengelola sertifikat server DB pada database. RDS merotasi sertifikat server DB secara otomatis sebelum sertifikat ini kedaluwarsa.

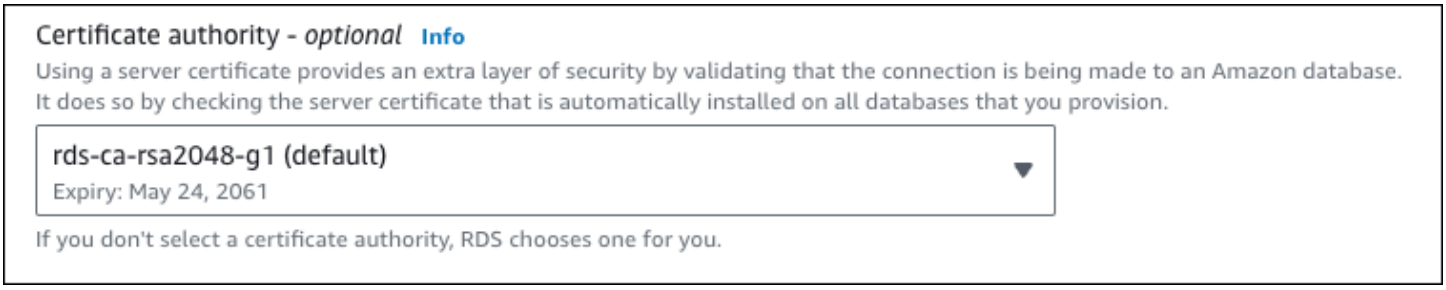
Mengatur CA untuk basis data Anda

Anda dapat mengatur CA untuk basis data saat Anda melakukan tugas berikut:

- Buat instans DB atau cluster DB multi-AZ — Anda dapat mengatur CA saat membuat instans atau cluster DB. Untuk petunjuk, lihat the section called “Membuat instans DB” atau the section called “Membuat klaster DB Multi-AZ”.
- Memodifikasi instans DB atau cluster DB multi-AZ — Anda dapat mengatur CA untuk instans atau cluster DB dengan memodifikasinya. Untuk petunjuk, lihat the section called “Memodifikasi instans DB” atau the section called “Mengubah klaster basis data Multi-AZ”.

> **Note**
>
> CA default diatur ke rds-ca-rsa 2048-g1. Anda dapat mengganti CA default untuk Anda Akun AWS dengan menggunakan perintah modify-certificate.

CA yang tersedia bergantung pada mesin DB dan versi mesin DB. Jika menggunakan AWS Management Console, Anda dapat memilih CA menggunakan pengaturan Otoritas sertifikat, seperti yang ditampilkan pada gambar berikut.



Konsol hanya menampilkan CA yang tersedia untuk mesin DB dan versi mesin DB. Jika Anda menggunakan AWS CLI, Anda dapat mengatur CA untuk instans DB menggunakan modify-db-instanceperintah create-db-instanceor. Anda dapat mengatur CA untuk cluster DB multi-AZ menggunakan modify-db-clusterperintah create-db-clusteror.

Jika Anda menggunakan AWS CLI, Anda dapat melihat CA yang tersedia untuk akun Anda dengan menggunakan perintah deskripsi-sertifikat. Perintah ini juga menunjukkan tanggal kedaluwarsa untuk setiap CA di `ValidTill` dalam output. Anda dapat menemukan CA yang tersedia untuk mesin DB tertentu dan versi mesin DB menggunakan describe-db-engine-versionsperintah.

Contoh berikut menunjukkan CA yang tersedia untuk versi mesin DB RDS for PostgreSQL default.

```
aws rds describe-db-engine-versions --default-only --engine postgres
```

Output Anda akan seperti yang berikut ini. CA yang tersedia tercantum di `SupportedCACertificateIdentifiers`. Output ini juga menunjukkan apakah versi mesin DB mendukung rotasi sertifikat tanpa pengaktifan ulang di `SupportsCertificateRotationWithoutRestart`.

```
{
    "DBEngineVersions": [
        {
            "Engine": "postgres",
            "MajorEngineVersion": "13",
            "EngineVersion": "13.4",
            "DBParameterGroupFamily": "postgres13",
            "DBEngineDescription": "PostgreSQL",
            "DBEngineVersionDescription": "PostgreSQL 13.4-R1",
            "ValidUpgradeTarget": [],
            "SupportsLogExportsToCloudwatchLogs": false,
            "SupportsReadReplica": true,
            "SupportedFeatureNames": [
                "Lambda"
            ],
            "Status": "available",
            "SupportsParallelQuery": false,
            "SupportsGlobalDatabases": false,
            "SupportsBabelfish": false,
            "SupportsCertificateRotationWithoutRestart": true,
            "SupportedCACertificateIdentifiers": [
                "rds-ca-2019",
                "rds-ca-rsa2048-g1",
                "rds-ca-ecc384-g1",
                "rds-ca-rsa4096-g1"
            ]
        }
    ]
}
```

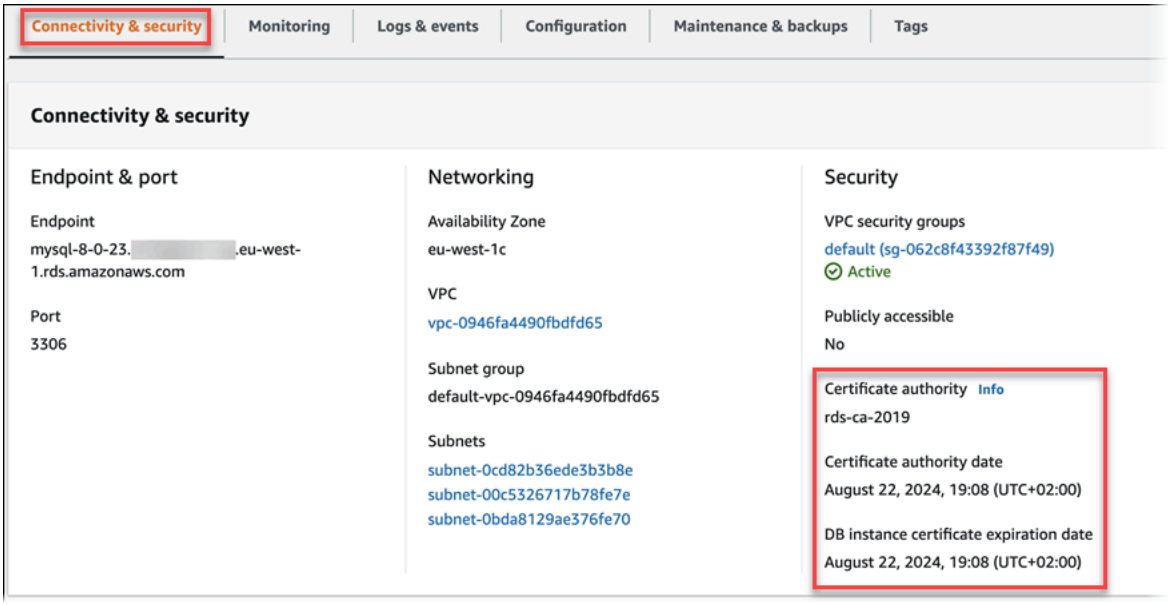
Validitas sertifikat server DB

Validitas sertifikat server DB bergantung pada mesin DB dan versi mesin DB. Jika versi mesin DB mendukung rotasi sertifikat tanpa pengaktifan ulang, validitas sertifikat server DB adalah 1 tahun. Jika tidak, validitasnya adalah 3 tahun.

Untuk informasi selengkapnya tentang rotasi sertifikat server DB, lihat Rotasi sertifikat server otomatis.

Melihat CA untuk instans DB Anda

Anda dapat melihat detail tentang CA untuk database dengan melihat tab Konektivitas & keamanan di konsol, seperti pada gambar berikut.



Jika Anda menggunakan AWS CLI, Anda dapat melihat detail tentang CA untuk instans DB dengan menggunakan describe-db-instancesperintah. Anda dapat melihat detail tentang CA untuk cluster DB multi-AZ dengan menggunakan describe-db-clustersperintah.

Untuk memeriksa konten paket sertifikat CA Anda, gunakan perintah berikut:

```
keytool -printcert -v -file global-bundle.pem
```

Bundel sertifikat untuk semua Wilayah AWS

Untuk mendapatkan bundel sertifikat yang berisi sertifikat perantara dan root untuk semua Wilayah AWS, unduh dari https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem.

Jika aplikasi Anda ada di Micro Windows dan memerlukan file PKCS7, Anda dapat mengunduh paket sertifikat PKCS7. Paket ini berisi sertifikat perantara dan root di https://truststore.pki.rds.amazonaws.com/global/global-bundle.p7b.

> **Note**
>
> Amazon RDS Proxy menggunakan sertifikat dari AWS Certificate Manager (ACM). Jika Anda menggunakan RDS Proxy, Anda tidak perlu mengunduh sertifikat Amazon RDS atau memperbarui aplikasi yang menggunakan koneksi Proxy RDS. Untuk informasi selengkapnya, lihat Penggunaan TLS/SSL dengan Proksi RDS.

Bundel sertifikat untuk spesifik Wilayah AWS

Untuk mendapatkan bundel sertifikat yang berisi sertifikat perantara dan root untuk Wilayah AWS, unduh dari tautan untuk tabel berikut. Wilayah AWS

AWS Wilayah	Paket sertifikat (PEM)	Paket sertifikat (PKCS7)
AS Timur (Virginia Utara)	us-east-1-bundle.pem	us-east-1-bundle.p7b
AS Timur (Ohio)	us-east-2-bundle.pem	us-east-2-bundle.p7b
AS Barat (California Utara)	us-west-1-bundle.pem	us-west-1-bundle.p7b
AS Barat (Oregon)	us-west-2-bundle.pem	us-west-2-bundle.p7b
Afrika (Cape Town)	af-south-1-bundle.pem	af-south-1-bundle.p7b
Asia Pasifik (Hong Kong)	ap-east-1-bundle.pem	ap-east-1-bundle.p7b
Asia Pasifik (Hyderabad)	ap-south-2-bundle.pem	ap-south-2-bundle.p7b
Asia Pasifik (Jakarta)	ap-southeast-3-bundle.pem	ap-southeast-3-bundle.p7b
Asia Pasifik (Melbourne)	ap-southeast-4-bundle.pem	ap-southeast-4-bundle.p7b
Asia Pasifik (Mumbai)	ap-south-1-bundle.pem	ap-south-1-bundle.p7b
Asia Pasifik (Osaka)	ap-northeast-3-bundle.pem	ap-northeast-3-bundle.p7b
Asia Pasifik (Tokyo)	ap-northeast-1-bundle.pem	ap-northeast-1-bundle.p7b
Asia Pasifik (Seoul)	ap-northeast-2-bundle.pem	ap-northeast-2-bundle.p7b

AWS Wilayah	Paket sertifikat (PEM)	Paket sertifikat (PKCS7)
Asia Pasifik (Singapura)	ap-southeast-1-bundle.pem	ap-southeast-1-bundle.p7b
Asia Pasifik (Sydney)	ap-southeast-2-bundle.pem	ap-southeast-2-bundle.p7b
Kanada (Pusat)	ca-central-1-bundle.pem	ca-central-1-bundle.p7b
Kanada Barat (Calgary)	ca-barat-1-bundle.pem	ca-barat-1-bundle.p7b
Eropa (Frankfurt)	eu-central-1-bundle.pem	eu-central-1-bundle.p7b
Eropa (Irlandia)	eu-west-1-bundle.pem	eu-west-1-bundle.p7b
Eropa (London)	eu-west-2-bundle.pem	eu-west-2-bundle.p7b
Eropa (Milan)	eu-south-1-bundle.pem	eu-south-1-bundle.p7b
Eropa (Paris)	eu-west-3-bundle.pem	eu-west-3-bundle.p7b
Eropa (Spanyol)	eu-south-2-bundle.pem	eu-south-2-bundle.p7b
Eropa (Stockholm)	eu-north-1-bundle.pem	eu-north-1-bundle.p7b
Eropa (Zürich)	eu-central-2-bundle.pem	eu-central-2-bundle.p7b
Israel (Tel Aviv)	il-central-1-bundle.pem	il-central-1-bundle.p7b
Timur Tengah (Bahrain)	me-south-1-bundle.pem	me-south-1-bundle.p7b
Timur Tengah (UEA)	me-central-1-bundle.pem	me-central-1-bundle.p7b
Amerika Selatan (Sao Paulo)	sa-east-1-bundle.pem	sa-east-1-bundle.p7b

Sertifikat AWS GovCloud (US)

Untuk mendapatkan bundel sertifikat yang berisi sertifikat perantara dan root untuk AWS GovCloud (US) Region s, unduh dari https://truststore.pki. us-gov-west-1.rds.amazonaws.com/global/global-bundle.pem.

Jika aplikasi Anda berada di Microsoft Windows dan memerlukan file PKCS7, Anda dapat mengunduh paket sertifikat PKCS7. Bundel ini berisi sertifikat perantara dan root di https://truststore.pki. us-gov-west-1.rds.amazonaws.com/global/global-bundle.p7b.

Untuk mendapatkan bundel sertifikat yang berisi sertifikat perantara dan root untuk AWS GovCloud (US) Region, unduh dari tautan untuk tabel berikut. AWS GovCloud (US) Region

AWS GovCloud (US) Region	Paket sertifikat (PEM)	Paket sertifikat (PKCS7)
AWS GovCloud (AS-Timur)	us-gov-east-1-bundel.pem	us-gov-east-1-bundel.p7b
AWS GovCloud (AS-Barat)	us-gov-west-1-bundel.pem	us-gov-west-1-bundel.p7b

Merotasi sertifikat SSL/TLS

Sertifikat Otoritas Sertifikat Amazon RDS rds-ca-2019 akan kedaluwarsa pada Agustus 2024. Jika Anda menggunakan atau berencana untuk menggunakan Secure Sockets Layer (SSL) atau Transport Layer Security (TLS) dengan verifikasi sertifikat untuk terhubung ke instans RDS DB atau cluster DB multi-AZ, pertimbangkan untuk menggunakan salah satu sertifikat rds-ca-rsa CA baru 2048-g1, 4096-g1 atau 384-g1. rds-ca-rsa rds-ca-ecc Jika saat ini Anda tidak menggunakan SSL/TLS dengan verifikasi sertifikat, Anda mungkin masih memiliki sertifikat CA yang kedaluwarsa dan harus memperbaruinya ke sertifikat CA baru jika Anda berencana untuk menggunakan SSL/TLS dengan verifikasi sertifikat untuk terhubung ke basis data RDS Anda.

Ikuti petunjuk ini untuk menyelesaikan pembaruan Anda. Sebelum memperbarui instans DB atau klaster DB multi-AZ untuk menggunakan sertifikat CA baru, pastikan Anda memperbarui klien atau aplikasi yang terhubung ke database RDS Anda.

Amazon RDS menyediakan sertifikat CA baru sebagai praktik terbaik AWS keamanan. Untuk informasi tentang sertifikat baru dan AWS Wilayah yang didukung, lihat.

> **Note**
>
> Amazon RDS Proxy menggunakan sertifikat dari AWS Certificate Manager (ACM). Jika Anda menggunakan RDS Proxy, ketika Anda memutar sertifikat SSL/TLS Anda, Anda tidak perlu memperbarui aplikasi yang menggunakan koneksi Proxy RDS. Untuk informasi selengkapnya, lihat Penggunaan TLS/SSL dengan Proksi RDS.

> **Note**
>
> Jika Anda menggunakan aplikasi Go versi 1.15 dengan instans DB atau cluster DB multi-AZ yang dibuat atau diperbarui ke sertifikat rds-ca-2019 sebelum 28 Juli 2020, Anda harus memperbarui sertifikat lagi. Jalankan `modify-db-instance` perintah untuk instans DB, atau `modify-db-cluster` perintah untuk cluster DB multi-AZ, menggunakan pengidentifikasi sertifikat CA baru. Anda dapat menemukan CA yang tersedia untuk mesin DB dan versi mesin DB tertentu menggunakan perintah `describe-db-engine-versions`. Jika Anda membuat database atau memperbarui sertifikatnya setelah 28 Juli 2020, tidak ada tindakan yang diperlukan. Untuk informasi selengkapnya, lihat Go GitHub issue #39568.

Topik

Memperbarui sertifikat CA Anda dengan memodifikasi instans atau cluster DB

Contoh berikut memperbarui sertifikat CA Anda dari rds-ca-2019 ke 2048-g1. rds-ca-rsa Anda dapat memilih sertifikat yang berbeda. Untuk informasi selengkapnya, lihat Otoritas sertifikat.

Untuk memperbarui sertifikat CA Anda dengan memodifikasi instans atau cluster DB

1. Unduh sertifikat SSL/TLS yang baru seperti yang dijelaskan dalam .
2. Perbarui aplikasi Anda untuk menggunakan sertifikat SSL/TLS baru.

   Metode untuk memperbarui aplikasi untuk sertifikat SSL/TLS baru bergantung pada aplikasi spesifik Anda. Bekerja samalah dengan developer aplikasi Anda untuk memperbarui sertifikat SSL/TLS untuk aplikasi Anda.

   Untuk informasi tentang pemeriksaan koneksi SSL/TLS dan pembaruan aplikasi untuk setiap mesin DB, lihat topik berikut:

   - Memperbarui aplikasi untuk terhubung ke instans MariaDB menggunakan sertifikat SSL/TLS baru

- Memperbarui aplikasi untuk terhubung ke instans DB Microsoft SQL Server menggunakan sertifikat SSL/TLS baru
- Memperbarui aplikasi untuk terhubung ke instans DB MySQL menggunakan sertifikat SSL/TLS baru
- Memperbarui aplikasi untuk terhubung ke instans DB Oracle menggunakan sertifikat SSL/TLS baru
- Memperbarui aplikasi untuk terhubung ke instans DB PostgreSQL menggunakan sertifikat SSL/TLS baru

Untuk contoh skrip yang memperbarui trust store untuk sistem operasi Linux, lihat Contoh skrip untuk mengimpor sertifikat ke trust store Anda.

> **Note**
>
> Paket sertifikat berisi sertifikat untuk CA lama dan baru, sehingga Anda dapat meningkatkan aplikasi Anda dengan aman dan mempertahankan konektivitas selama periode transisi. Jika Anda menggunakan AWS Database Migration Service untuk memigrasikan database ke instans DB atau cluster , sebaiknya gunakan bundel sertifikat untuk memastikan konektivitas selama migrasi.

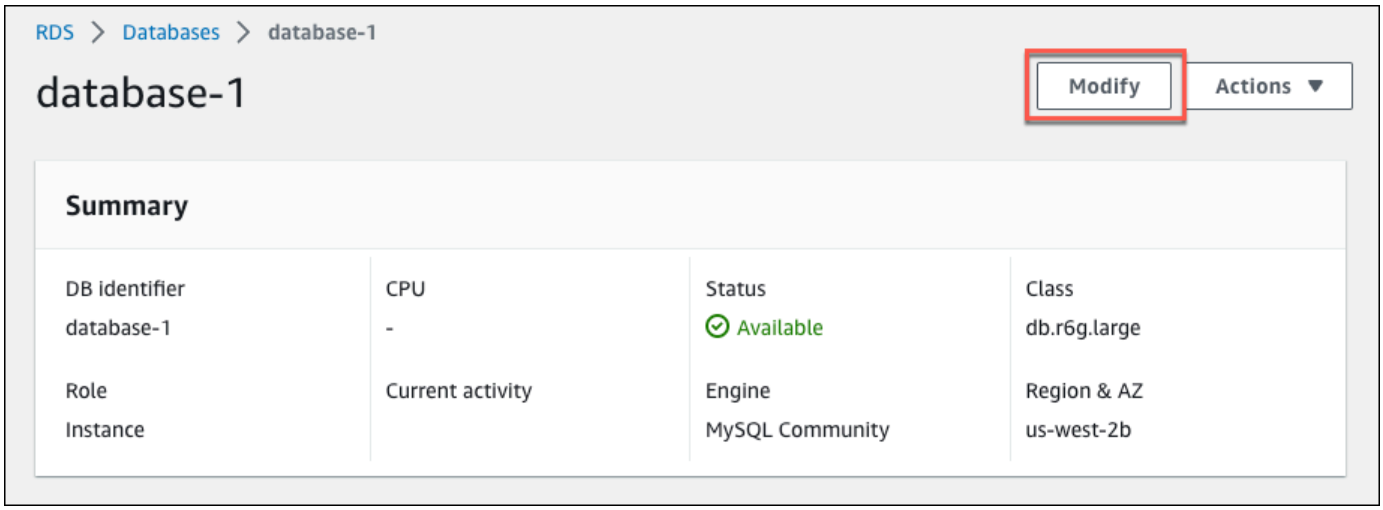
3. Ubah instans DB atau cluster DB multi-AZ untuk mengubah CA dari rds-ca-2019 menjadi 2048-g1. rds-ca-rsa Untuk memeriksa apakah database Anda memerlukan restart untuk memperbarui sertifikat CA, gunakan describe-db-engine-versionsperintah dan periksa `SupportsCertificateRotationWithoutRestart` bendera.

> **Important**
>
> Jika Anda mengalami masalah konektivitas setelah masa berlaku sertifikat berakhir, gunakan opsi terapkan segera dengan menentukan Terapkan segera di konsol atau dengan menentukan opsi `--apply-immediately` menggunakan AWS CLI. Secara default, operasi ini dijadwalkan untuk berjalan selama periode pemeliharaan berikutnya. Untuk mengatur penggantian CA klaster Anda yang berbeda dari CA RDS default, gunakan perintah CLI modify-certificates.

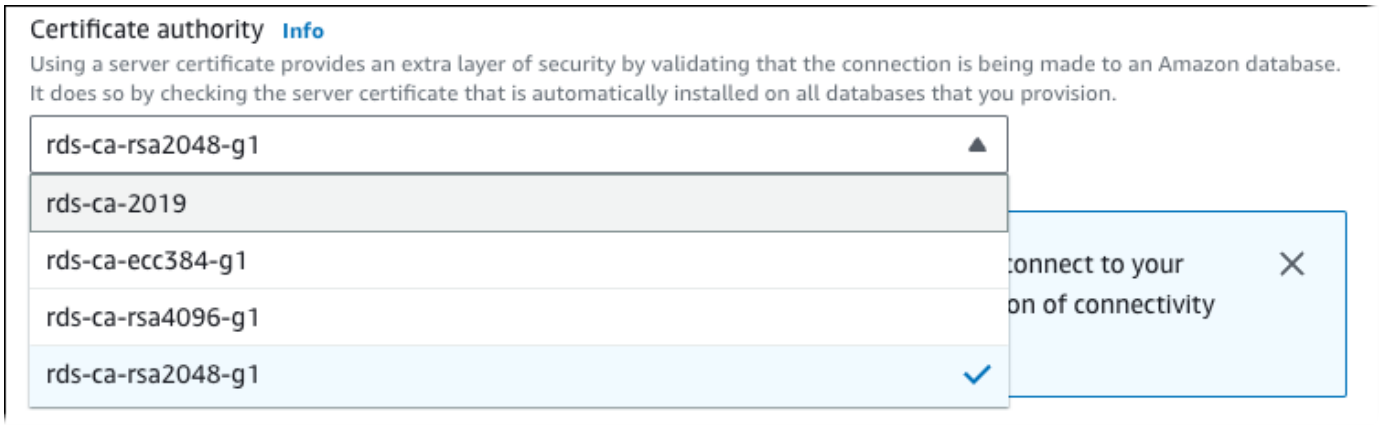
Anda dapat menggunakan AWS Management Console atau AWS CLI untuk mengubah sertifikat CA dari rds-ca-2019 ke rds-ca-rsa2048-g1 untuk instans DB atau cluster DB multi-AZ.

Konsol

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Databases, lalu pilih instans DB atau cluster DB multi-AZ yang ingin Anda modifikasi.
3. Pilih Ubah.



4. Di bagian Konektivitas, pilih rds-ca-rsa2048-g1.



5. Pilih Lanjutkan dan periksa ringkasan modifikasi.
6. Untuk segera menerapkan perubahan, pilih Terapkan segera.
7. Di halaman konfirmasi, tinjau perubahan Anda. Jika benar, pilih Modify DB Instance atau Modify cluster untuk menyimpan perubahan Anda.

> **Important**
>
> Saat Anda menjadwalkan operasi ini, pastikan bahwa Anda telah memperbarui trust store sisi klien sebelumnya.

Atau pilih Kembali untuk mengedit perubahan atau Batalkan untuk membatalkan perubahan Anda.

AWS CLI

Untuk menggunakan AWS CLI untuk mengubah CA dari rds-ca-2019 ke rds-ca-rsa2048-g1 untuk instans DB atau cluster DB multi-AZ, panggil perintah or. modify-db-instancemodify-db-cluster Tentukan instans DB atau pengidentifikasi cluster dan `--ca-certificate-identifier` opsi.

Gunakan `--apply-immediately` parameter untuk segera menerapkan pembaruan. Secara default, operasi ini dijadwalkan untuk berjalan selama periode pemeliharaan berikutnya.

> **Important**
>
> Saat Anda menjadwalkan operasi ini, pastikan bahwa Anda telah memperbarui trust store sisi klien sebelumnya.

Example

contoh DB

Contoh berikut memodifikasi `mydbinstance` dengan menyetel sertifikat CA `kerds-ca-rsa2048-g1`.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --ca-certificate-identifier rds-ca-rsa2048-g1
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --ca-certificate-identifier rds-ca-rsa2048-g1
```

> **Note**
>
> Jika instance Anda memerlukan reboot, Anda dapat menggunakan perintah modify-db-instanceCLI dan menentukan opsi. `--no-certificate-rotation-restart`

Example

Kluster DB multi-AZ

Contoh berikut memodifikasi `mydbcluster` dengan menyetel sertifikat CA `kerds-ca-rsa2048-g1`.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-cluster \
    --db-cluster-identifier mydbcluster \
    --ca-certificate-identifier rds-ca-rsa2048-g1
```

Untuk Windows:

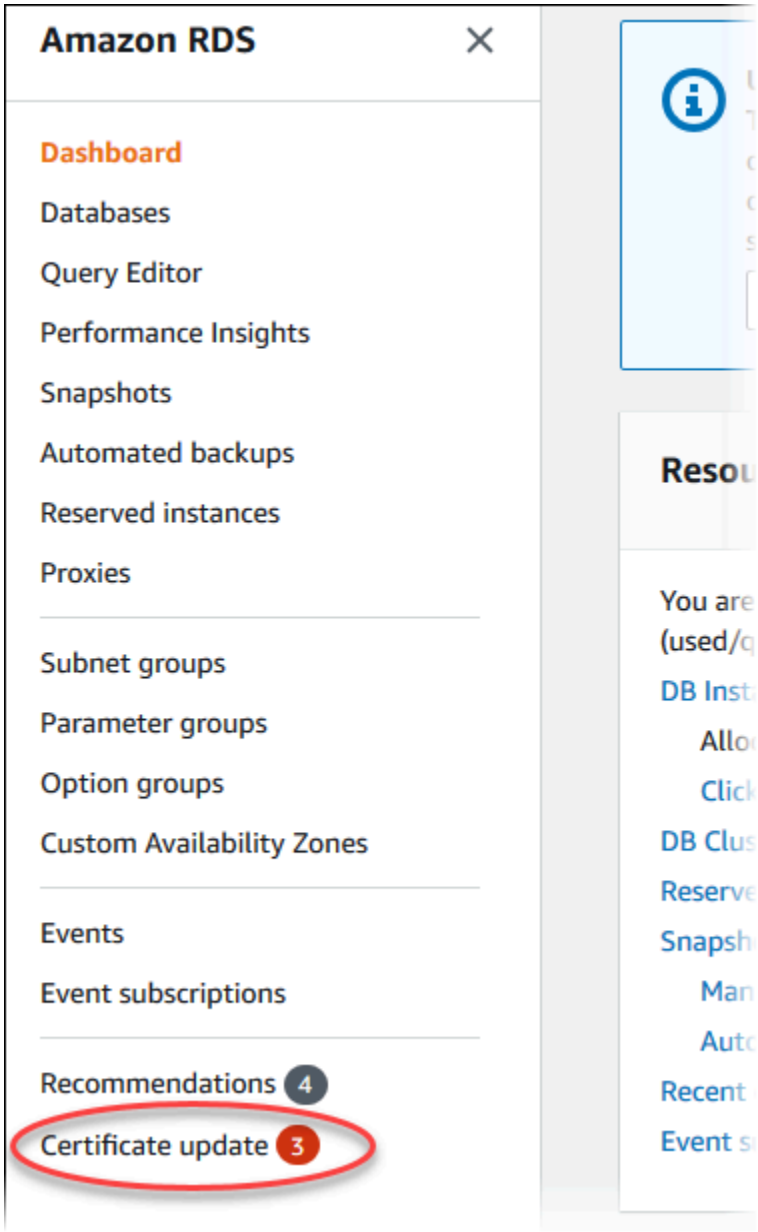
```
aws rds modify-db-cluster ^
    --db-cluster-identifier mydbcluster ^
    --ca-certificate-identifier rds-ca-rsa2048-g1
```

Memperbarui sertifikat CA Anda dengan menerapkan pemeliharaan

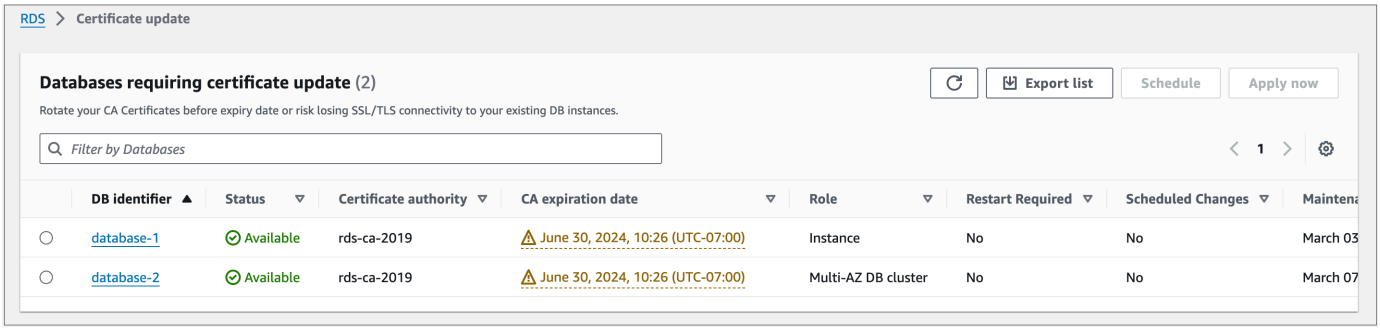
Lakukan langkah-langkah berikut untuk memperbarui sertifikat CA Anda dengan menerapkan pemeliharaan.

Untuk memperbarui sertifikat CA Anda dengan menerapkan pemeliharaan

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Pembaruan sertifikat.



Halaman Basis data yang memerlukan pembaruan sertifikat akan muncul.



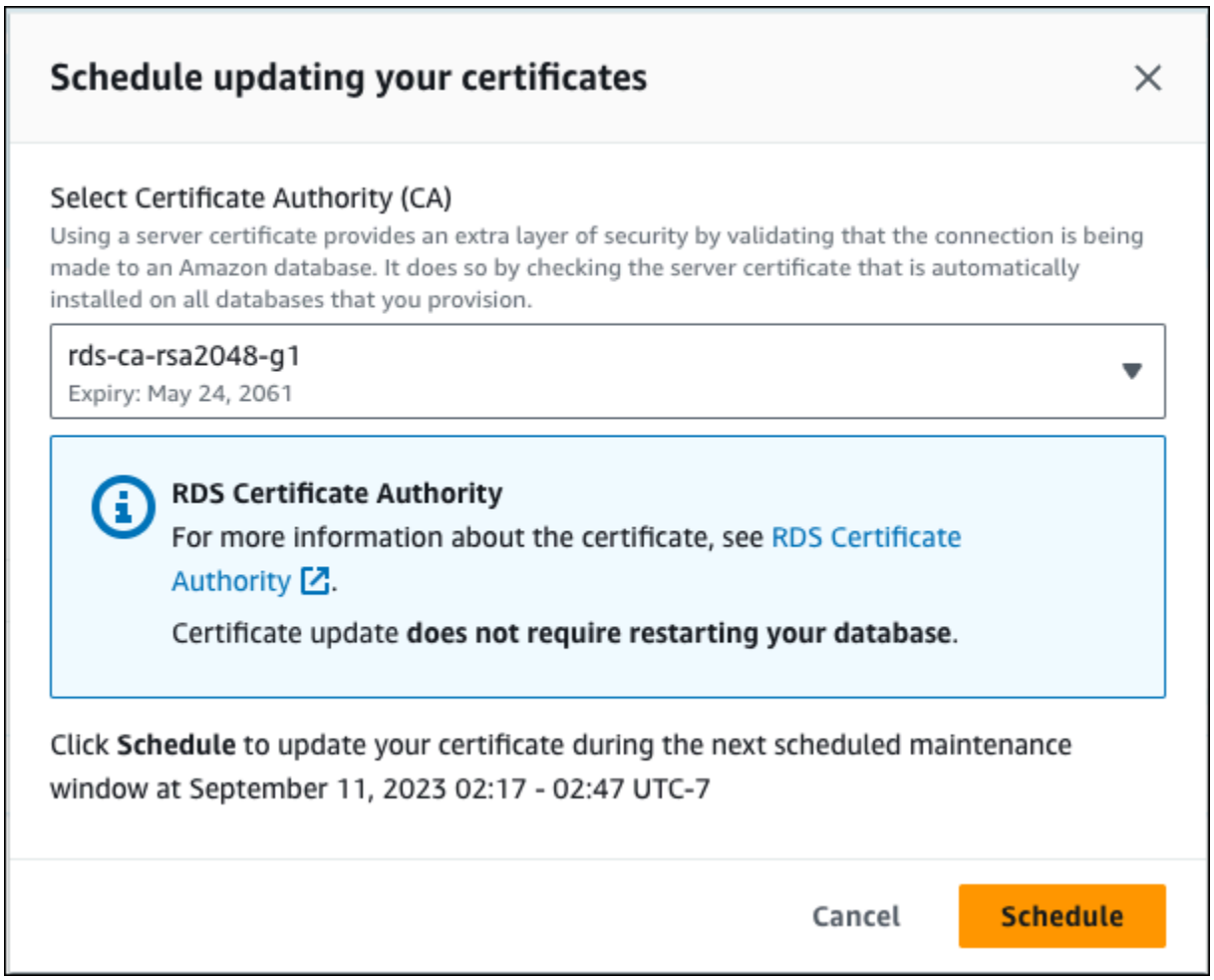
> **Note**
>
> Halaman ini hanya menampilkan instans dan cluster DB untuk saat ini. Wilayah AWS Jika Anda memiliki database di lebih dari satu Wilayah AWS, periksa halaman ini di masing-masing Wilayah AWS untuk melihat semua instance DB dengan sertifikat SSL/TLS lama.

3. Pilih instans DB atau cluster DB multi-AZ yang ingin Anda perbarui.

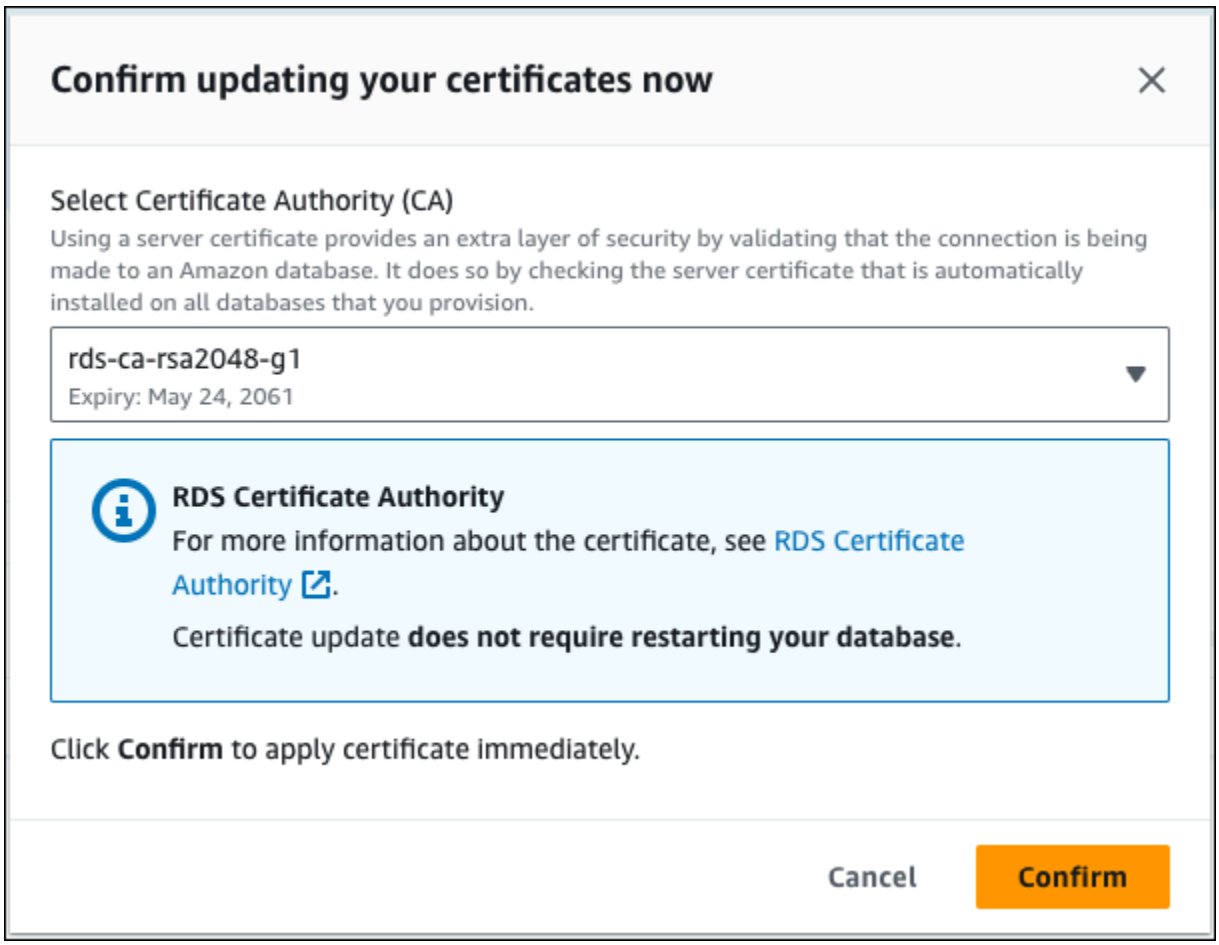
   Anda dapat menjadwalkan rotasi sertifikat untuk periode pemeliharaan berikutnya dengan memilih Jadwal. Segera terapkan rotasi dengan memilih Terapkan sekarang.

   > **Important**
   >
   > Jika Anda mengalami masalah konektivitas setelah sertifikat kedaluwarsa, gunakan opsi Terapkan sekarang.

4. a. Jika Anda memilih Jadwal, Anda akan diminta untuk mengonfirmasi rotasi sertifikat CA. Prompt ini juga menyatakan periode terjadwal untuk pembaruan Anda.



b. Jika Anda memilih Terapkan sekarang, Anda akan diminta untuk mengonfirmasi rotasi sertifikat CA.



> **Important**
>
> Sebelum menjadwalkan rotasi sertifikat CA di basis data Anda, perbarui aplikasi klien yang menggunakan SSL/TLS dan sertifikat server untuk terhubung. Pembaruan ini khusus untuk mesin DB Anda. Setelah Anda memperbarui aplikasi klien ini, Anda dapat mengonfirmasi rotasi sertifikat CA.

Untuk melanjutkan, pilih kotak centang, lalu pilih Konfirmasi.

5. Ulangi langkah 3 dan 4 untuk setiap instans dan cluster DB yang ingin Anda perbarui.

Rotasi sertifikat server otomatis

Jika CA Anda mendukung rotasi sertifikat server otomatis, RDS secara otomatis menangani rotasi sertifikat server DB. RDS menggunakan CA root yang sama untuk rotasi otomatis ini, jadi Anda tidak perlu mengunduh paket CA baru. Lihat Otoritas sertifikat.

Rotasi dan validitas sertifikat server DB Anda bergantung pada mesin DB Anda:

- Jika mesin DB Anda mendukung rotasi tanpa pengaktifan ulang, RDS secara otomatis merotasi sertifikat server DB tanpa memerlukan tindakan apa pun dari Anda. RDS mencoba merotasi sertifikat server DB Anda dalam periode pemeliharaan yang Anda pilih di waktu paruh sertifikat server DB. Sertifikat server DB baru berlaku selama 12 bulan.
- Jika mesin DB Anda tidak mendukung rotasi tanpa pengaktifan ulang, RDS memberi tahu Anda tentang peristiwa pemeliharaan setidaknya 6 bulan sebelum sertifikat server DB kedaluwarsa. Sertifikat server DB baru berlaku selama 36 bulan.

Gunakan describe-db-engine-versionsperintah dan periksa `SupportsCertificateRotationWithoutRestart` bendera untuk mengidentifikasi apakah versi mesin DB mendukung memutar sertifikat tanpa memulai ulang. Untuk informasi selengkapnya, lihat Mengatur CA untuk basis data Anda.

Contoh skrip untuk mengimpor sertifikat ke trust store Anda

Berikut adalah contoh skrip shell yang mengimpor paket sertifikat ke trust store.

Setiap skrip shell menggunakan keytool, yang merupakan bagian dari Java Development Kit (JDK). Untuk informasi tentang cara menginstal JDK, lihat JDK Installation Guide.

Topik

- Contoh skrip untuk mengimpor sertifikat di Linux
- Contoh skrip untuk mengimpor sertifikat di macOS

Contoh skrip untuk mengimpor sertifikat di Linux

Berikut adalah contoh skrip shell yang mengimpor paket sertifikat ke trust store di sistem operasi Linux.

```
mydir=tmp/certs
```

```
if [ ! -e "${mydir}" ]
then
mkdir -p "${mydir}"
fi

truststore=${mydir}/rds-truststore.jks
storepassword=changeit

curl -sS "https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem" >
 ${mydir}/global-bundle.pem
awk 'split_after == 1 {n++;split_after=0} /-----END CERTIFICATE-----/ {split_after=1}
{print > "rds-ca-" n+1 ".pem"}' < ${mydir}/global-bundle.pem

for CERT in rds-ca-*; do
  alias=$(openssl x509 -noout -text -in $CERT | perl -ne 'next unless /Subject:/;
 s/.*(CN=|CN = )//; print')
  echo "Importing $alias"
  keytool -import -file ${CERT} -alias "${alias}" -storepass ${storepassword} -keystore
 ${truststore} -noprompt
  rm $CERT
done

rm ${mydir}/global-bundle.pem

echo "Trust store content is: "

keytool -list -v -keystore "$truststore" -storepass ${storepassword} | grep Alias | cut
 -d " " -f3- | while read alias
do
   expiry=`keytool -list -v -keystore "$truststore" -storepass ${storepassword} -alias
 "${alias}" | grep Valid | perl -ne 'if(/until: (.*?)\n/) { print "$1\n"; }'`
   echo " Certificate ${alias} expires in '$expiry'"
done
```

Contoh skrip untuk mengimpor sertifikat di macOS

Berikut adalah contoh skrip shell yang mengimpor paket sertifikat ke trust store di sistem operasi macOS.

```
mydir=tmp/certs
if [ ! -e "${mydir}" ]
```

```
then
mkdir -p "${mydir}"
fi

truststore=${mydir}/rds-truststore.jks
storepassword=changeit

curl -sS "https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem" >
 ${mydir}/global-bundle.pem
split -p "-----BEGIN CERTIFICATE-----" ${mydir}/global-bundle.pem rds-ca-

for CERT in rds-ca-*; do
  alias=$(openssl x509 -noout -text -in $CERT | perl -ne 'next unless /Subject:/;
 s/.*(CN=|CN = )//; print')
  echo "Importing $alias"
  keytool -import -file ${CERT} -alias "${alias}" -storepass ${storepassword} -keystore
 ${truststore} -noprompt
  rm $CERT
done

rm ${mydir}/global-bundle.pem

echo "Trust store content is: "

keytool -list -v -keystore "$truststore" -storepass ${storepassword} | grep Alias | cut
 -d " " -f3- | while read alias
do
   expiry=`keytool -list -v -keystore "$truststore" -storepass ${storepassword} -alias
 "${alias}" | grep Valid | perl -ne 'if(/until: (.*?)\n/) { print "$1\n"; }'`
   echo " Certificate ${alias} expires in '$expiry'"
done
```

Privasi lalu lintas jaringan internet

Koneksi dilindungi antara Amazon RDS dan aplikasi on-premise dan antara Amazon RDS dan sumber daya AWS lainnya di dalam Kawasan AWS yang sama.

Lalu lintas antara layanan dan aplikasi dan klien on-premise

Anda memiliki dua opsi konektivitas antara jaringan privat dan AWS:

- Koneksi VPN Lokasi-ke-Lokasi AWS. Lihat informasi yang lebih lengkap di Apakah AWS Site-to-Site VPN?
- Koneksi AWS Direct Connect. Lihat informasi yang lebih lengkap di Apakah AWS Direct Connect?

Anda mendapatkan akses ke Amazon RDS melalui jaringan dengan menggunakan operasi API yang diterbitkan AWS. 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.

Selain itu, permintaan harus ditandatangani menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan pengguna utama IAM. Atau Anda dapat menggunakan AWS Security Token Service (AWS STS) untuk menghasilkan kredensial keamanan sementara untuk menandatangani permintaan.

Manajemen identitas dan akses untuk Amazon RDS

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diotorisasi (memiliki izin) untuk menggunakan sumber daya Amazon RDS. IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan di Amazon RDS Amazon .

Pengguna layanan – Jika Anda menggunakan layanan Amazon RDS untuk melakukan pekerjaan, administrator Anda akan memberikan kredensial dan izin yang Anda butuhkan. Saat Anda menggunakan lebih banyak fitur Amazon RDS untuk melakukan pekerjaan, Anda mungkin memerlukan izin tambahan. Memahami bagaimana cara mengelola akses dapat membantu Anda meminta izin yang tepat dari administrator Anda. Jika Anda tidak dapat mengakses fitur di Amazon RDS, lihat Memecahkan masalah identitas dan akses Amazon RDS.

Administrator layanan – Jika Anda bertanggung jawab atas sumber daya Amazon RDS di perusahaan, Anda mungkin memiliki akses penuh ke Amazon RDS. Tugas Anda adalah menentukan fitur Amazon RDS dan sumber daya mana yang dapat diakses karyawan Anda. Kemudian, Anda

harus mengirim permintaan kepada administrator untuk mengubah izin pengguna layanan. Tinjau informasi di halaman ini untuk memahami konsep dasar IAM. Untuk mempelajari selengkapnya tentang bagaimana perusahaan Anda dapat menggunakan IAM dengan Amazon RDS, lihat Cara kerja Amazon RDS dengan IAM.

Administrator – Jika Anda adalah seorang administrator, Anda mungkin ingin mengetahui detail tentang cara menulis kebijakan untuk mengelola akses ke Amazon RDS. Untuk melihat contoh kebijakan berbasis identitas Amazon RDS yang dapat Anda gunakan di IAM, lihat Contoh kebijakan berbasis identitas untuk Amazon RDS.

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensi identitas Anda. Anda harus diautentikasi (masuk ke AWS) sebagai Pengguna root akun AWS, sebagai pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk AWS sebagai identitas federasi dengan menggunakan kredensil yang disediakan melalui sumber identitas. AWS IAM Identity Center Pengguna (IAM Identity Center), autentikasi masuk tunggal perusahaan Anda, dan kredensi Google atau Facebook Anda adalah contoh identitas federasi. Saat Anda masuk sebagai identitas gabungan, administrator Anda sebelumnya menyiapkan federasi identitas menggunakan peran IAM. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil peran.

Bergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau portal AWS akses. Untuk informasi selengkapnya tentang masuk AWS, lihat Cara masuk ke Panduan AWS Sign-In Pengguna Anda Akun AWS.

Jika Anda mengakses AWS secara terprogram, AWS sediakan kit pengembangan perangkat lunak (SDK) dan antarmuka baris perintah (CLI) untuk menandatangani permintaan Anda secara kriptografis dengan menggunakan kredensil Anda. Jika Anda tidak menggunakan AWS alat, Anda harus menandatangani permintaan sendiri. Untuk informasi selengkapnya tentang penggunaan metode yang disarankan untuk menandatangani permintaan sendiri, lihat Menandatangani permintaan AWS API di Panduan Pengguna IAM.

Apa pun metode autentikasi yang digunakan, Anda mungkin diminta untuk menyediakan informasi keamanan tambahan. Misalnya, AWS merekomendasikan agar Anda menggunakan otentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari selengkapnya, lihat Autentikasi multi-faktor dalam Panduan Pengguna AWS IAM Identity Center dan Menggunakan autentikasi multi-faktor (MFA) di AWS dalam Panduan Pengguna IAM.

AWS pengguna root akun

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya di akun. Identitas ini disebut pengguna Akun AWS root dan diakses dengan masuk dengan alamat email dan kata sandi yang Anda gunakan untuk membuat akun. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari Anda. Lindungi kredensial pengguna root Anda dan gunakan kredensial tersebut untuk melakukan tugas yang hanya dapat dilakukan pengguna root. Untuk daftar lengkap tugas yang mengharuskan Anda masuk sebagai pengguna root, lihat Tugas yang memerlukan kredensial pengguna root dalam Panduan Pengguna IAM.

Identitas terfederasi

Sebagai praktik terbaik, mewajibkan pengguna manusia, termasuk pengguna yang memerlukan akses administrator, untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS dengan menggunakan kredensi sementara.

Identitas federasi adalah pengguna dari direktori pengguna perusahaan Anda, penyedia identitas web, direktori Pusat Identitas AWS Directory Service, atau pengguna mana pun yang mengakses Layanan AWS dengan menggunakan kredensil yang disediakan melalui sumber identitas. Ketika identitas federasi mengakses Akun AWS, mereka mengambil peran, dan peran memberikan kredensi sementara.

Untuk manajemen akses terpusat, kami sarankan Anda menggunakan AWS IAM Identity Center. Anda dapat membuat pengguna dan grup di Pusat Identitas IAM, atau Anda dapat menghubungkan dan menyinkronkan ke sekumpulan pengguna dan grup di sumber identitas Anda sendiri untuk digunakan di semua aplikasi Akun AWS dan aplikasi Anda. Untuk informasi tentang Pusat Identitas IAM, lihat Apa itu Pusat Identitas IAM? dalam Panduan Pengguna AWS IAM Identity Center .

Pengguna dan grup IAM

Pengguna IAM adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus untuk satu orang atau aplikasi. Jika memungkinkan, sebaiknya andalkan kredensial sementara daripada membuat pengguna IAM yang memiliki kredensial jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan khusus yang memerlukan kredensial jangka panjang dengan pengguna IAM, kami sarankan Anda merotasi kunci akses. Untuk informasi selengkapnya, lihat Rotasikan kunci akses secara rutin untuk kasus penggunaan yang memerlukan kredensial jangka panjang dalam Panduan Pengguna IAM.

Grup IAM adalah identitas yang menentukan kumpulan pengguna IAM. Anda tidak dapat masuk sebagai grup. Anda dapat menggunakan grup untuk menentukan izin untuk beberapa pengguna sekaligus. Grup membuat izin lebih mudah dikelola untuk sekelompok besar pengguna. Misalnya, Anda dapat memiliki grup yang bernama IAMAdmins dan memberikan izin kepada grup tersebut untuk mengelola sumber daya IAM.

Pengguna berbeda dari peran. Pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran dimaksudkan untuk dapat digunakan oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara. Untuk mempelajari selengkapnya, lihat Kapan harus membuat pengguna IAM (bukan peran) dalam Panduan Pengguna IAM.

Anda dapat mengautentikasi ke instans DB Anda menggunakan autentikasi basis data IAM.

Autentikasi basis data IAM berfungsi dengan mesin DB berikut:

- RDS for MariaDB
- RDS for MySQL
- RDS for PostgreSQL

Untuk informasi selengkapnya tentang cara mengautentikasi ke instans DB Anda menggunakan IAM, lihat Autentikasi basis data IAM untuk MariaDB, MySQL, dan PostgreSQL.

Peran IAM

Peran IAM adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus. Peran ini mirip dengan pengguna, tetapi tidak terkait dengan orang tertentu. Anda dapat mengambil peran IAM untuk sementara AWS Management Console dengan beralih peran. Anda dapat mengambil peran dengan memanggil operasi AWS CLI atau AWS API atau dengan menggunakan URL kustom. Untuk informasi selengkapnya tentang cara menggunakan peran, lihat Menggunakan peran IAM dalam Panduan Pengguna IAM.

Peran IAM dengan kredensial sementara berguna dalam situasi berikut:

- Izin pengguna sementara – Pengguna dapat mengambil peran IAM untuk mendapatkan izin yang berbeda sementara waktu agar dapat melakukan tugas tertentu.
- Akses pengguna terfederasi – Untuk menetapkan izin ke identitas terfederasi, Anda membuat peran dan menentukan izin untuk peran tersebut. Saat identitas terfederasi mengautentikasi, identitas tersebut akan dikaitkan dengan peran dan diberi izin yang ditentukan oleh peran tersebut.

Untuk informasi tentang peran-peran untuk federasi, lihat Membuat peran untuk Penyedia Identitas pihak ketiga dalam Panduan Pengguna IAM. Jika Anda menggunakan Pusat Identitas IAM, Anda perlu mengonfigurasi set izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah identitas tersebut diautentikasi, Pusat Identitas IAM mengorelasikan izin yang diatur ke peran dalam IAM. Untuk informasi tentang rangkaian izin, silakan lihat Rangkaian izin dalam Panduan Pengguna AWS IAM Identity Center .

- Akses lintas akun – Anda dapat menggunakan peran IAM untuk mengizinkan seseorang (prinsipal tepercaya) di akun lain untuk mengakses sumber daya di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, dengan beberapa Layanan AWS, Anda dapat melampirkan kebijakan secara langsung ke sumber daya (alih-alih menggunakan peran sebagai proxy). Untuk mempelajari perbedaan antara kebijakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Bagaimana peran IAM berbeda dari kebijakan berbasis sumber daya dalam Panduan Pengguna IAM.
- Akses lintas layanan — Beberapa Layanan AWS menggunakan fitur lain Layanan AWS. Sebagai contoh, ketika Anda melakukan panggilan dalam suatu layanan, biasanya layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Sebuah layanan mungkin melakukannya menggunakan izin prinsipal yang memanggil, menggunakan peran layanan, atau peran terkait layanan.
  - Sesi akses teruskan — Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Saat Anda menggunakan beberapa layanan, Anda mungkin melakukan tindakan yang kemudian memulai tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan saat membuat permintaan FAS, lihat Teruskan sesi akses.
  - Peran layanan – Peran layanan adalah peran IAM yang diambil oleh layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, memodifikasi, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Membuat peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.
  - Peran terkait layanan — Peran terkait layanan adalah jenis peran layanan yang ditautkan ke. Layanan AWS Layanan dapat menggunakan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.

- Aplikasi yang berjalan di Amazon EC2 — Anda dapat menggunakan peran IAM untuk mengelola kredensi sementara untuk aplikasi yang berjalan pada instans EC2 dan membuat atau permintaan API. AWS CLI AWS Cara ini lebih dianjurkan daripada menyimpan kunci akses dalam instans EC2. Untuk menetapkan AWS peran ke instans EC2 dan membuatnya tersedia untuk semua aplikasinya, Anda membuat profil instance yang dilampirkan ke instance. Profil instans berisi peran dan memungkinkan program yang berjalan di instans EC2 mendapatkan kredensial sementara. Untuk informasi selengkapnya, lihat Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan di instans Amazon EC2 dalam Panduan Pengguna IAM.

Untuk mempelajari apakah harus menggunakan peran IAM, lihat Kapan harus membuat peran IAM (bukan pengguna) dalam Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke identitas atau sumber daya IAM. AWS Kebijakan adalah objek AWS yang, ketika dikaitkan dengan identitas atau sumber daya, menentukan izinnya. AWS mengevaluasi kebijakan ini ketika entitas (pengguna root, pengguna, atau peran IAM) membuat permintaan. Izin dalam kebijakan dapat menentukan permintaan yang diizinkan atau ditolak. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang struktur dan konten dokumen kebijakan JSON, lihat Gambaran umum kebijakan JSON dalam Panduan Pengguna IAM.

Administrator dapat menggunakan kebijakan untuk menentukan siapa yang memiliki akses ke AWS sumber daya, dan tindakan apa yang dapat mereka lakukan pada sumber daya tersebut. Setiap entitas IAM (set izin atau peran) dimulai tanpa izin. Dengan kata lain, secara default, pengguna tidak dapat melakukan apa pun, termasuk mengubah kata sandinya sendiri. Untuk memberikan izin kepada pengguna untuk melakukan sesuatu, administrator harus melampirkan kebijakan izin kepada pengguna. Atau administrator dapat menambahkan pengguna ke grup yang memiliki izin yang dimaksudkan. Ketika administrator memberikan izin untuk grup, semua pengguna dalam grup tersebut akan diberi izin tersebut.

Kebijakan IAM mendefinisikan izin untuk suatu tindakan terlepas dari metode yang Anda gunakan untuk melakukan operasinya. Sebagai contoh, anggap saja Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole`. Pengguna dengan kebijakan tersebut bisa mendapatkan informasi peran dari AWS Management Console, API AWS CLI, atau AWS API.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke identitas, seperti set izin atau peran. Kebijakan ini mengontrol tindakan apa yang bisa dilakukan oleh identitas tersebut, sumber daya yang mana, dan dalam kondisi apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Membuat kebijakan IAM dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan terkelola. Kebijakan inline disematkan secara langsung ke satu set izin atau peran. Kebijakan terkelola adalah kebijakan mandiri yang dapat Anda lampirkan ke beberapa set izin dan peran di AWS akun Anda. Kebijakan AWS terkelola mencakup kebijakan terkelola dan kebijakan yang dikelola pelanggan. Untuk mempelajari cara memilih antara kebijakan terkelola atau kebijakan inline, lihat Memilih antara kebijakan terkelola dan kebijakan inline dalam Panduan Pengguna IAM.

Untuk informasi tentang kebijakan AWS terkelola yang khusus untuk Amazon RDS Aurora, lihat. AWS kebijakan terkelola untuk Amazon RDS

Jenis kebijakan lainnya

AWS mendukung jenis kebijakan tambahan yang kurang umum. Jenis kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda oleh jenis kebijakan yang lebih umum.

- Batasan izin – Batasan izin adalah fitur lanjutan untuk menetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas IAM (set izin atau peran). Anda dapat menetapkan batasan izin untuk suatu entitas. Izin yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas milik entitas dan batasan izin tersebut. Kebijakan berbasis sumber daya yang menentukan set izin atau peran di bidang `Principal` tidak dibatasi oleh batasan izin. Penolakan eksplisit dalam salah satu kebijakan ini membatalkan izin. Untuk informasi selengkapnya tentang batasan izin, lihat Batasan izin untuk entitas IAM dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCP) — SCP adalah kebijakan JSON yang menentukan izin maksimum untuk organisasi atau unit organisasi (OU) di. AWS Organizations AWS Organizations adalah layanan untuk mengelompokkan dan mengelola beberapa AWS akun secara terpusat yang dimiliki bisnis Anda. Jika Anda mengaktifkan semua fitur di sebuah organisasi, maka Anda dapat menerapkan kebijakan kontrol layanan (SCP) ke salah satu atau semua akun Anda. SCP membatasi izin untuk entitas di akun anggota, termasuk masing-masing. Pengguna root akun AWS Untuk informasi selengkapnya tentang Organisasi dan SCP, lihat Cara kerja SCP dalam Panduan Pengguna AWS Organizations .

- Kebijakan sesi – Kebijakan sesi adalah kebijakan lanjutan yang Anda berikan sebagai parameter saat Anda membuat sesi sementara secara programatis untuk peran atau pengguna terfederasi. Izin sesi yang dihasilkan adalah perpotongan antara set izin atau kebijakan berbasis identitas peran dan kebijakan sesi tersebut. Izin juga dapat berasal dari kebijakan berbasis sumber daya. Penolakan eksplisit dalam salah satu kebijakan ini membatalkan izin. Untuk informasi selengkapnya, lihat Kebijakan sesi dalam Panduan Pengguna IAM.

Beberapa jenis kebijakan

Ketika beberapa jenis kebijakan berlaku untuk sebuah permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat Logika evaluasi kebijakan di Panduan Pengguna IAM.

Cara kerja Amazon RDS dengan IAM

Sebelum menggunakan IAM untuk mengelola akses ke Amazon RDS, Anda harus memahami fitur IAM yang dapat digunakan dengan Amazon RDS.

Fitur IAM yang dapat digunakan dengan Amazon RDS

Fitur IAM	Dukungan Amazon RDS
Kebijakan berbasis identitas	Ya
Kebijakan berbasis sumber daya	Tidak
Tindakan kebijakan	Ya
Sumber daya kebijakan	Ya
kunci-kunci persyaratan kebijakan (spesifik layanan)	Ya
ACL	Tidak
Kontrol akses berbasis atribut (ABAC) (tag dalam kebijakan)	Ya
Kredensial sementara	Ya

Fitur IAM	Dukungan Amazon RDS
Teruskan sesi akses	Ya
Peran layanan	Ya
Peran terkait layanan	Ya

Untuk mendapatkan tampilan tingkat tinggi tentang cara Amazon RDS Amazon dan layanan AWS lainnya bekerja dengan IAM, AWS lihat layanan yang bekerja dengan IAM di Panduan Pengguna IAM.

Topik

- Kebijakan berbasis identitas Amazon RDS
- Kebijakan berbasis sumber daya dalam Amazon RDS
- Tindakan kebijakan untuk Amazon RDS
- Sumber daya kebijakan untuk Amazon RDS
- Kunci kondisi kebijakan untuk Amazon RDS
- Daftar kontrol akses (ACL) di Amazon RDS
- Kontrol akses berbasis atribut (ABAC) dalam kebijakan dengan tag Amazon RDS
- Menggunakan kredensial sementara dengan Amazon RDS
-
- Peran layanan untuk Amazon RDS
- Peran terkait layanan untuk Amazon RDS

Kebijakan berbasis identitas Amazon RDS

Mendukung kebijakan berbasis identitas	Ya

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan pengguna dan peran, di sumber daya mana, dan

dengan ketentuan apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Membuat kebijakan IAM dalam Panduan Pengguna IAM.

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan tindakan dan sumber daya yang diizinkan atau ditolak, serta ketentuan terkait jenis tindakan yang diizinkan atau ditolak. Anda tidak dapat menentukan pengguna utama dalam kebijakan berbasis identitas karena kebijakan ini berlaku untuk pengguna atau peran yang dilampiri kebijakan. Untuk mempelajari semua elemen yang dapat digunakan dalam kebijakan JSON, lihat Referensi elemen kebijakan JSON IAM dalam Panduan Pengguna IAM.

Contoh kebijakan berbasis identitas untuk Amazon RDS

Untuk melihat contoh kebijakan berbasis identitas Amazon RDS, lihat Contoh kebijakan berbasis identitas untuk Amazon RDS.

Kebijakan berbasis sumber daya dalam Amazon RDS

Mendukung kebijakan berbasis sumber daya	Tidak

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya yang dilampiri kebijakan tersebut, kebijakan ini menentukan jenis tindakan yang dapat dilakukan oleh pengguna utama tertentu di sumber daya tersebut dan apa ketentuannya. Anda harus menentukan pengguna utama dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan seluruh akun atau entitas IAM di akun lain sebagai pengguna utama dalam kebijakan berbasis sumber daya. Menambahkan pengguna utama lintas akun ke kebijakan berbasis sumber daya bagian dari membangun hubungan kepercayaan. Ketika prinsipal dan sumber daya berbeda Akun AWS, administrator IAM di akun tepercaya juga harus memberikan izin entitas utama (pengguna atau peran) untuk mengakses sumber daya. Izin diberikan dengan melampirkan kebijakan berbasis identitas ke entitas tersebut. Namun, jika kebijakan berbasis sumber daya memberikan akses kepada pengguna utama dalam akun yang sama, kebijakan berbasis identitas lainnya tidak diperlukan. Untuk informasi selengkapnya, lihat Perbedaan peran IAM dengan kebijakan berbasis sumber daya di Panduan Pengguna IAM.

Tindakan kebijakan untuk Amazon RDS

Mendukung tindakan kebijakan	Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, pengguna utama mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Tindakan kebijakan biasanya memiliki nama yang sama dengan operasi AWS API terkait. Ada beberapa pengecualian, misalnya tindakan hanya izin yang tidak memiliki operasi API yang cocok. Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam suatu kebijakan. Tindakan tambahan ini disebut tindakan dependen.

Menyertakan tindakan dalam suatu kebijakan untuk memberikan izin melakukan operasi terkait.

Tindakan kebijakan di Amazon RDS menggunakan awalan berikut sebelum tindakan: `rds:`. Misalnya, untuk memberikan izin kepada seseorang untuk menjelaskan instans DB dengan operasi API `DescribeDBInstances` Amazon RDS, Anda menyertakan tindakan `rds:DescribeDBInstances` dalam kebijakan mereka. Pernyataan kebijakan harus memuat elemen `Action` atau `NotAction`. Amazon RDS menentukan serangkaian tindakannya sendiri yang menjelaskan tugas yang dapat Anda lakukan dengan layanan ini.

Untuk menentukan beberapa tindakan dalam satu pernyataan, pisahkan tindakan dengan koma seperti berikut:

```
"Action": [
      "rds:action1",
      "rds:action2"
```

Anda juga dapat menentukan beberapa tindakan menggunakan wildcard (*). Misalnya, untuk menentukan semua tindakan yang dimulai dengan kata `Describe`, sertakan tindakan berikut.

```
"Action": "rds:Describe*"
```

Untuk melihat daftar tindakan Amazon RDS, lihat Tindakan yang Ditentukan oleh Amazon RDS di Referensi Otorisasi Layanan.

Sumber daya kebijakan untuk Amazon RDS

Mendukung sumber daya kebijakan	Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, pengguna utama mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek atau beberapa objek yang menjadi target penerapan tindakan. Pernyataan harus menyertakan elemen `Resource` atau `NotResource`. Praktik terbaiknya, tentukan sumber daya menggunakan Amazon Resource Name (ARN). Anda dapat melakukan ini untuk tindakan yang mendukung jenis sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, misalnya operasi pencantuman, gunakan wildcard (*) untuk mengindikasikan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

```
"Resource": "*"
```

Sumber daya instans DB memiliki Amazon Resource Name (ARN) berikut.

```
arn:${Partition}:rds:${Region}:${Account}:{ResourceType}/${Resource}
```

Untuk informasi selengkapnya tentang format ARN, lihat Amazon Resource Names (ARN) dan ruang nama AWS layanan.

Misalnya, untuk menentukan instans DB `dbtest` dalam pernyataan Anda, gunakan ARN berikut.

```
"Resource": "arn:aws:rds:us-west-2:123456789012:db:dbtest"
```

Untuk menentukan semua instans DB milik akun tertentu, gunakan wildcard (*).

```
"Resource": "arn:aws:rds:us-east-1:123456789012:db:*"
```

Beberapa operasi API RDS, seperti operasi untuk membuat sumber daya, tidak dapat dilakukan pada sumber daya tertentu. Jika demikian, gunakan wildcard (*).

```
"Resource": "*"
```

Banyak operasi API Amazon RDS menggunakan beberapa sumber daya. Misalnya, `CreateDBInstance` membuat instans DB. Anda dapat menentukan bahwa seorang pengguna harus menggunakan grup keamanan dan grup parameter spesifik saat membuat instans DB. Untuk menentukan beberapa sumber daya dalam satu pernyataan, pisahkan ARN dengan koma.

```
"Resource": [
      "resource1",
      "resource2"
```

Untuk melihat daftar jenis sumber daya Amazon RDS dan ARN-nya, lihat Sumber Daya yang Ditentukan oleh Amazon RDS di Referensi Otorisasi Layanan. Untuk mempelajari jenis tindakan yang dapat Anda tentukan dengan ARN di tiap sumber daya, lihat Tindakan yang Ditentukan oleh Amazon RDS.

Kunci kondisi kebijakan untuk Amazon RDS

Mendukung kunci kondisi kebijakan spesifik layanan	Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, pengguna utama mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Elemen `Condition` (atau blok `Condition`) memungkinkan Anda menentukan kondisi di mana suatu pernyataan akan diterapkan. Elemen `Condition` bersifat opsional. Anda dapat membuat ekspresi kondisional yang menggunakan operator kondisi, misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi dalam kebijakan dengan nilai-nilai yang diminta.

Jika Anda menentukan beberapa elemen `Condition` dalam satu pernyataan, atau beberapa kunci dalam satu elemen `Condition`, AWS akan mengevaluasinya dengan menggunakan operasi `AND` logis. Jika Anda menentukan beberapa nilai untuk satu kunci kondisi, AWS mengevaluasi kondisi menggunakan `OR` operasi logis. Semua kondisi harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan kondisi. Misalnya, Anda dapat memberikan izin kepada pengguna IAM untuk mengakses sumber daya hanya jika izin tersebut

mempunyai tag yang sesuai dengan nama pengguna IAM mereka. Untuk informasi selengkapnya, lihat Elemen kebijakan IAM: variabel dan tag di Panduan Pengguna IAM.

AWS mendukung kunci kondisi global dan kunci kondisi khusus layanan. Untuk melihat semua kunci kondisi AWS global, lihat kunci konteks kondisi AWS global di Panduan Pengguna IAM.

Amazon RDS menentukan set kunci kondisinya sendiri dan juga mendukung penggunaan beberapa kunci kondisi global. Untuk melihat semua kunci kondisi AWS global, lihat kunci konteks kondisi AWS global di Panduan Pengguna IAM.

Semua operasi API RDS mendukung kunci kondisi `aws:RequestedRegion`.

Untuk melihat daftar kunci kondisi Amazon RDS, lihat Kunci Kondisi untuk Amazon RDS di Referensi Otorisasi Layanan. Untuk mempelajari tindakan dan sumber daya yang dapat Anda gunakan dengan kunci kondisi, lihat Tindakan yang Ditentukan oleh Amazon RDS.

Daftar kontrol akses (ACL) di Amazon RDS

Mendukung daftar kontrol akses (ACL)	Tidak

Daftar kontrol akses (ACL) mengontrol pengguna utama (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL sama dengan kebijakan berbasis sumber daya, meskipun tidak menggunakan format dokumen kebijakan JSON.

Kontrol akses berbasis atribut (ABAC) dalam kebijakan dengan tag Amazon RDS

Mendukung tag kontrol akses berbasis atribut (ABAC) dalam kebijakan	Ya

Kontrol akses berbasis atribut (ABAC) adalah strategi otorisasi yang menentukan izin berdasarkan atribut. Dalam AWS, atribut ini disebut tag. Anda dapat melampirkan tag ke entitas IAM (pengguna atau peran) dan ke banyak AWS sumber daya. Pemberian tanda ke entitas dan sumber daya adalah langkah pertama dari ABAC. Kemudian, rancanglah kebijakan ABAC untuk mengizinkan operasi saat tag milik pengguna utama cocok dengan tag yang ada di sumber daya yang ingin diakses.

ABAC sangat berguna di lingkungan yang berkembang dengan cepat dan berguna di situasi di mana pengelolaan kebijakan menjadi rumit.

Untuk mengendalikan akses berdasarkan tag, berikan informasi tentang tag di elemen kondisi dari kebijakan dengan menggunakan kunci kondisi `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, atau `aws:TagKeys`.

Jika sebuah layanan mendukung ketiga kunci kondisi untuk setiap jenis sumber daya, nilainya adalah Ya untuk layanan tersebut. Jika suatu layanan mendukung ketiga kunci kondisi hanya untuk beberapa jenis sumber daya, nilainya adalah Parsial.

Untuk informasi selengkapnya tentang ABAC, lihat Apa itu ABAC? di Panduan Pengguna IAM. Untuk melihat tutorial terkait langkah-langkah penyiapan ABAC, lihat Menggunakan kontrol akses berbasis atribut (ABAC) di Panduan Pengguna IAM.

Untuk informasi selengkapnya cara memberi tag ke sumber daya Amazon RDS, lihat Menentukan kondisi: Menggunakan tag kustom. Untuk melihat contoh kebijakan berbasis identitas untuk membatasi akses ke sumber daya berdasarkan tag pada sumber daya tersebut, lihat Berikan izin untuk tindakan atas suatu sumber daya dengan tag tertentu dengan dua nilai yang berbeda.

Menggunakan kredensial sementara dengan Amazon RDS

Mendukung kredensial sementara	Ya

Beberapa Layanan AWS tidak berfungsi saat Anda masuk menggunakan kredensi sementara. Untuk informasi tambahan, termasuk yang Layanan AWS bekerja dengan kredensi sementara, lihat Layanan AWS yang bekerja dengan IAM di Panduan Pengguna IAM.

Anda menggunakan kredensi sementara jika Anda masuk AWS Management Console menggunakan metode apa pun kecuali nama pengguna dan kata sandi. Misalnya, ketika Anda mengakses AWS menggunakan tautan masuk tunggal (SSO) perusahaan Anda, proses tersebut secara otomatis membuat kredensi sementara. Anda juga akan membuat kredensial sementara secara otomatis saat masuk ke konsol sebagai pengguna dan kemudian beralih peran. Untuk informasi selengkapnya tentang cara beralih peran, lihat Beralih peran (konsol) di Panduan Pengguna IAM.

Anda dapat membuat kredenal sementara secara manual menggunakan API AWS CLI atau AWS . Anda kemudian dapat menggunakan kredensi sementara tersebut untuk mengakses. AWS AWS merekomendasikan agar Anda secara dinamis menghasilkan kredensi sementara alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, lihat Kredensial keamanan sementara di IAM.

Mendukung sesi akses ke depan	Ya

Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Saat Anda menggunakan beberapa layanan, Anda mungkin melakukan tindakan yang kemudian memulai tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan saat membuat permintaan FAS, lihat Meneruskan sesi akses.

Peran layanan untuk Amazon RDS

Mendukung peran layanan	Ya

Peran layanan adalah peran IAM yang diambil oleh layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Membuat peran untuk mendelegasikan izin ke Layanan AWS di Panduan Pengguna IAM.

> **Warning**
>
> Mengubah izin untuk peran layanan dapat mengganggu fungsionalitas Amazon RDS. Edit peran layanan hanya jika Amazon RDS menyediakan panduan untuk melakukannya.

Peran terkait layanan untuk Amazon RDS

Mendukung peran terkait layanan	Ya

Peran terkait layanan adalah jenis peran layanan yang ditautkan ke. Layanan AWS Layanan dapat menggunakan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.

Untuk detail tentang cara menggunakan peran terkait layanan Amazon RDS, lihat Menggunakan peran terkait layanan untuk Amazon RDS.

Contoh kebijakan berbasis identitas untuk Amazon RDS

Secara default, peran dan kumpulan izin tidak memiliki izin untuk membuat atau mengubah sumber daya Amazon RDS. Mereka juga tidak dapat melakukan tugas menggunakan AWS Management Console, AWS CLI, atau AWS API. Administrator harus membuat kebijakan IAM yang memberikan izin kepada peran atau kumpulan izin untuk menjalankan operasi API tertentu pada sumber daya tertentu yang diperlukan. Administrator kemudian dapat melampirkan kebijakan tersebut ke peran atau kumpulan izin yang memerlukan izin tersebut.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM menggunakan contoh dokumen kebijakan JSON ini, lihat Membuat kebijakan pada tab JSON dalam Panduan Pengguna IAM.

Topik

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber daya Amazon RDS yang ada di akun Anda. Tindakan ini dikenai biaya untuk

Akun AWS Anda. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat kebijakan yang dikelola AWS atau kebijakan yang dikelola AWS untuk fungsi pekerjaan di Panduan Pengguna IAM.
- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukan ini dengan menentukan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, juga dikenal sebagai izin hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan IAM untuk menerapkan izin, lihat Kebijakan dan izin di IAM di Panduan Pengguna IAM.
- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti AWS CloudFormation. Untuk informasi selengkapnya, lihat Elemen kebijakan JSON IAM: Syarat di Panduan Pengguna IAM.
- Menggunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda guna memastikan izin yang aman dan berfungsi – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat validasi kebijakan Analizer Akses IAM di Panduan Pengguna IAM.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk mewajibkan MFA saat operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat Mengonfigurasi akses API yang dilindungi MFA di Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.

Menggunakan konsol Amazon RDS

Untuk mengakses konsol Amazon RDS, Anda harus memiliki kumpulan izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang sumber daya Amazon RDS di Anda. Akun AWS Jika Anda membuat kebijakan berbasis identitas yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana mestinya untuk entitas (pengguna atau peran) dengan kebijakan tersebut.

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau AWS API. Sebagai alternatif, hanya izinkan akses ke tindakan yang cocok dengan operasi API yang sedang Anda coba lakukan.

Untuk memastikan bahwa entitas tersebut masih dapat menggunakan konsol Amazon RDS , lampirkan juga kebijakan terkelola AWS berikut ke entitas.

```
AmazonRDSReadOnlyAccess
```

Untuk informasi selengkapnya, lihat Menambahkan izin ke pengguna di Panduan Pengguna IAM.

Izinkan pengguna melihat izin mereka sendiri

Contoh ini menunjukkan cara membuat kebijakan yang mengizinkan para pengguna IAM melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini di konsol atau menggunakan API atau secara terprogram. AWS CLI AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
```

```
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Izinkan pengguna untuk membuat instans DB di akun AWS

Berikut ini adalah contoh kebijakan yang memungkinkan pengguna dengan ID `123456789012` untuk membuat instans DB untuk AWS akun Anda. Kebijakan ini mewajibkan nama instans DB baru dimulai dengan `test`. Instans DB yang baru juga harus menggunakan mesin basis data MySQL dan kelas instans DB `db.t2.micro`. Selain itu, instans DB baru harus menggunakan grup opsi dan grup parameter DB yang dimulai dengan `default`, dan harus menggunakan grup subnet `default`.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AllowCreateDBInstanceOnly",
         "Effect": "Allow",
         "Action": [
            "rds:CreateDBInstance"
         ],
         "Resource": [
            "arn:aws:rds:*:123456789012:db:test*",
            "arn:aws:rds:*:123456789012:og:default*",
            "arn:aws:rds:*:123456789012:pg:default*",
```

```
            "arn:aws:rds:*:123456789012:subgrp:default"
         ],
         "Condition": {
            "StringEquals": {
               "rds:DatabaseEngine": "mysql",
               "rds:DatabaseClass": "db.t2.micro"
            }
         }
      }
   ]
}
```

Kebijakan ini mencakup pernyataan tunggal yang menentukan izin berikut untuk pengguna:

- Kebijakan ini memungkinkan pengguna untuk membuat instans DB menggunakan operasi CreatedBInstance API (ini juga berlaku untuk perintah dan). create-db-instance AWS CLI AWS Management Console
- Elemen `Resource` menentukan bahwa pengguna dapat melakukan tindakan pada atau dengan sumber daya. Anda menentukan sumber daya menggunakan Amazon Resource Name (ARN). ARN ini mencakup nama layanan yang dimiliki sumber daya (`rds`), AWS Wilayah (*menunjukkan wilayah mana pun dalam contoh ini), nomor AWS akun (123456789012adalah nomor akun dalam contoh ini), dan jenis sumber daya. Untuk informasi selengkapnya tentang cara membuat ARN, lihat Bekerja dengan Amazon Resource Name (ARN) di Amazon RDS.

  Elemen `Resource` dalam contoh menentukan batasan kebijakan berikut pada sumber daya untuk pengguna:

  - ID instans DB untuk instans DB baru harus dimulai dengan `test` (misalnya, `testCustomerData1`, `test-region2-data`).
  - Grup opsi untuk instans DB baru harus dimulai dengan `default`.
  - Grup parameter DB opsi untuk instans DB baru harus dimulai dengan `default`.
  - Grup subnet untuk instans DB baru harus berupa grup subnet `default`.
- Elemen `Condition` menentukan bahwa mesin DB harus berupa MySQL dan kelas instans DB harus berupa `db.t2.micro`. Elemen `Condition` menentukan kondisi ketika kebijakan harus diberlakukan. Anda dapat menambahkan izin atau batasan tambahan dengan menggunakan elemen `Condition`. Untuk informasi selengkapnya tentang cara menentukan kondisi, lihat Kunci kondisi kebijakan untuk Amazon RDS. Contoh ini menetapkan kondisi `rds:DatabaseEngine` dan `rds:DatabaseClass`. Untuk informasi tentang nilai kondisi yang valid untuk `rds:DatabaseEngine`, lihat daftar di bagian parameter `Engine` di CreateDBInstance.

Untuk informasi tentang nilai kondisi yang valid untuk `rds:DatabaseClass`, Lihat Mesin DB yang didukung untuk kelas instans DB .

Kebijakan ini tidak menentukan elemen `Principal` karena dalam kebijakan berbasis identitas, Anda tidak menentukan pengguna utama yang mendapatkan izin. Saat Anda menyematkan kebijakan kepada pengguna, pengguna ini menjadi pengguna utama implisit. Saat Anda menyematkan kebijakan izin pada peran IAM, pengguna utama yang diidentifikasi dalam kebijakan kepercayaan peran tersebut akan mendapatkan izin.

Untuk melihat daftar tindakan Amazon RDS, lihat Tindakan yang Ditentukan oleh Amazon RDS di Referensi Otorisasi Layanan.

Izin yang diperlukan untuk menggunakan konsol

Agar pengguna dapat bekerja dengan konsol, pengguna tersebut harus memiliki kumpulan izin minimum. Izin ini memungkinkan pengguna untuk mendeskripsikan sumber daya Amazon RDS Aurora untuk akun AWS mereka dan untuk memberikan informasi terkait lainnya, termasuk keamanan Amazon EC2 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 masih dapat menggunakan konsol, lampirkan juga kebijakan yang dikelola `AmazonRDSReadOnlyAccess` kepada pengguna, sebagaimana dijelaskan dalam Mengelola akses menggunakan kebijakan.

Anda tidak perlu memperbolehkan izin konsol minimum bagi pengguna yang hanya melakukan panggilan ke AWS CLI atau Amazon RDS API.

Kebijakan berikut memberikan akses penuh ke semua sumber daya Amazon RDS untuk akun root: AWS

```
AmazonRDSFullAccess
```

Mengizinkan pengguna melakukan setiap tindakan yang dijelaskan pada sumber daya RDS

Kebijakan izin berikut memberikan izin kepada pengguna untuk menjalankan semua tindakan yang dimulai dengan `Describe`. Tindakan ini menunjukkan informasi tentang sumber daya RDS, seperti instans DB. Karakter wildcard (*) dalam elemen `Resource` menunjukkan bahwa tindakan diperbolehkan untuk semua sumber daya Amazon RDS yang dimiliki akun tersebut.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AllowRDSDescribe",
         "Effect": "Allow",
         "Action": "rds:Describe*",
         "Resource": "*"
      }
   ]
}
```

Mengizinkan pengguna membuat instans DB yang menggunakan grup parameter DB dan grup subnet yang telah ditentukan

Kebijakan izin berikut memberikan izin untuk hanya memperbolehkan pengguna membuat instans DB yang harus menggunakan grup parameter DB `mydbpg` dan grup subnet DB `mydbsubnetgroup`.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "VisualEditor0",
         "Effect": "Allow",
         "Action": "rds:CreateDBInstance",
         "Resource": [
            "arn:aws:rds:*:*:pg:mydbpg",
            "arn:aws:rds:*:*:subgrp:mydbsubnetgroup"
         ]
      }
   ]
}
```

Berikan izin untuk tindakan atas suatu sumber daya dengan tag tertentu dengan dua nilai yang berbeda

Anda dapat menggunakan kondisi dalam kebijakan berbasis identitas untuk mengontrol akses ke sumber daya Amazon RDS berdasarkan tag. Kebijakan berikut memungkinkan izin untuk melakukan operasi `CreateDBSnapshot` API pada instans DB dengan tag `stage` diatur ke `development` atau `test`.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllowAnySnapshotName",
         "Effect":"Allow",
         "Action":[
            "rds:CreateDBSnapshot"
         ],
         "Resource":"arn:aws:rds:*:123456789012:snapshot:*"
      },
      {
         "Sid":"AllowDevTestToCreateSnapshot",
         "Effect":"Allow",
         "Action":[
            "rds:CreateDBSnapshot"
         ],
         "Resource":"arn:aws:rds:*:123456789012:db:*",
         "Condition":{
            "StringEquals":{
                "rds:db-tag/stage":[
                  "development",
                  "test"
               ]
            }
         }
      }
   ]
}
```

Kebijakan berikut memungkinkan izin untuk melakukan operasi `ModifyDBInstance` API pada instans DB dengan tag `stage` diatur ke `development` atau `test`.

```
{
```

```
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllowChangingParameterOptionSecurityGroups",
         "Effect":"Allow",
         "Action":[
            "rds:ModifyDBInstance"
         ],
         "Resource": [
            "arn:aws:rds:*:123456789012:pg:*",
            "arn:aws:rds:*:123456789012:secgrp:*",
            "arn:aws:rds:*:123456789012:og:*"
         ]
      },
      {
         "Sid":"AllowDevTestToModifyInstance",
         "Effect":"Allow",
         "Action":[
            "rds:ModifyDBInstance"
         ],
         "Resource":"arn:aws:rds:*:123456789012:db:*",
         "Condition":{
            "StringEquals":{
                "rds:db-tag/stage":[
                  "development",
                  "test"
               ]
            }
         }
      }
   ]
}
```

Mencegah pengguna menghapus instans DB

Kebijakan izin berikut memberikan izin untuk mencegah pengguna menghapus instans DB tertentu. Misalnya, Anda mungkin ingin menolak kemampuan untuk menghapus instans DB produksi Anda kepada setiap pengguna yang bukan administrator.

```
{
   "Version": "2012-10-17",
```

```
   "Statement": [
      {
         "Sid": "DenyDelete1",
         "Effect": "Deny",
         "Action": "rds:DeleteDBInstance",
         "Resource": "arn:aws:rds:us-west-2:123456789012:db:my-mysql-instance"
      }
   ]
}
```

Menolak semua akses ke sumber daya

Anda juga dapat secara eksplisit menolak akses ke sumber daya. Kebijakan penolakan lebih diutamakan daripada kebijakan yang diizinkan. Kebijakan berikut secara eksplisit menolak kemampuan pengguna untuk mengelola sumber daya:

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Deny",
         "Action": "rds:*",
         "Resource": "arn:aws:rds:us-east-1:123456789012:db:mydb"
      }
   ]
}
```

Contoh Kebijakan: Menggunakan kunci kondisi

Berikut ini adalah contoh cara menggunakan kunci kondisi dalam kebijakan izin IAM Amazon RDS.

Contoh 1: Memberikan izin untuk membuat instans DB yang menggunakan mesin DB spesifik dan tidak berupa Multi-AZ

Kebijakan berikut menggunakan kunci kondisi RDS dan memungkinkan pengguna membuat instans DB yang menggunakan mesin basis data MySQL saja dan tidak menggunakan MultiAZ. Elemen `Condition` menunjukkan persyaratan bahwa mesin basis data adalah MySQL.

```
{
   "Version": "2012-10-17",
   "Statement": [
```

```
      {
         "Sid": "AllowMySQLCreate",
         "Effect": "Allow",
         "Action": "rds:CreateDBInstance",
         "Resource": "*",
         "Condition": {
            "StringEquals": {
               "rds:DatabaseEngine": "mysql"
            },
            "Bool": {
               "rds:MultiAz": false
            }
         }
      }
   ]
}
```

Contoh 2: Secara eksplisit menolak izin untuk membuat instans DB untuk kelas instans DB tertentu dan membuat instans DB yang menggunakan IOPS yang Tersedia

Kebijakan berikut secara eksplisit menolak izin untuk membuat instans DB yang menggunakan kelas instans DB `r3.8xlarge` dan `m4.10xlarge`, yang merupakan kelas instans DB terbesar dan termahal. Kebijakan ini juga mencegah pengguna membuat instans DB yang menggunakan IOPS yang Tersedia, yang menimbulkan biaya tambahan.

Izin yang secara tegas menolak lebih diprioritaskan daripada izin lain yang diberikan. Ini memastikan bahwa identitas tidak akan secara kebetulan mendapatkan izin yang tidak pernah ingin Anda berikan.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "DenyLargeCreate",
         "Effect": "Deny",
         "Action": "rds:CreateDBInstance",
         "Resource": "*",
         "Condition": {
            "StringEquals": {
               "rds:DatabaseClass": [
                  "db.r3.8xlarge",
                  "db.m4.10xlarge"
               ]
```

```
            }
         }
      },
      {
         "Sid": "DenyPIOPSCreate",
         "Effect": "Deny",
         "Action": "rds:CreateDBInstance",
         "Resource": "*",
         "Condition": {
            "NumericNotEquals": {
               "rds:Piops": "0"
            }
         }
      }
   ]
}
```

Contoh 3: Membatasi kumpulan kunci dan nilai tag yang dapat digunakan untuk menandai sumber daya

Kebijakan berikut menggunakan kunci kondisi RDS dan memungkinkan penambahan tag dengan kunci `stage` untuk ditambahkan ke sumber daya dengan nilai `test`, `qa`, dan `production`.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "rds:AddTagsToResource",
            "rds:RemoveTagsFromResource"
         ],
         "Resource": "*",
         "Condition": {
            "streq": {
               "rds:req-tag/stage": [
                  "test",
                  "qa",
                  "production"
               ]
            }
         }
      }
```

```
    ]
}
```

Menentukan kondisi: Menggunakan tag kustom

Amazon RDS mendukung penentuan kondisi dalam kebijakan IAM menggunakan tag kustom.

Sebagai contoh, misalkan Anda menambahkan tag bernama `environment` ke instans DB Anda dengan nilai seperti `beta`, `staging`, `production`, dan sebagainya. Jika melakukannya, Anda dapat membuat kebijakan yang membatasi pengguna tertentu pada instans DB berdasarkan nilai tag `environment`.

> **Note**
>
> ID tag kustom bersifat peka huruf besar-kecil.

Tabel berikut mencantumkan ID tag RDS yang dapat digunakan pada elemen `Condition`.

ID tag RDS	Berlaku untuk
`db-tag`	Instans DB, termasuk replika baca
`snapshot-tag`	Snapshot DB
`ri-tag`	Instans DB terpesan
`og-tag`	Grup opsi DB
`pg-tag`	Grup parameter DB
`subgrp-tag`	Grup subnet DB
`es-tag`	Langganan peristiwa
`cluster-tag`	Klaster DB
`cluster-pg-tag`	Grup parameter klaster DB
`cluster-snapshot-tag`	Snapshot klaster DB

Sintaks untuk kondisi tag kustom adalah sebagai berikut:

```
"Condition":{"StringEquals":{"rds:rds-tag-identifier/tag-name":
["value"]} }
```

Misalnya, elemen `Condition` berikut berlaku untuk instans DB dengan tag bernama `environment` dan nilai tag `production`.

```
"Condition":{"StringEquals":{"rds:db-tag/environment": ["production"]} }
```

Untuk informasi tentang membuat tag, lihat Memberi tag pada sumber daya Amazon RDS.

Important

Jika Anda mengelola akses ke sumber daya RDS Anda menggunakan pemberian tag, sebaiknya Anda mengamankan akses ke tag untuk sumber daya RDS Anda. Anda dapat mengelola akses ke tag dengan membuat kebijakan untuk tindakan `AddTagsToResource` dan `RemoveTagsFromResource`. Misalnya, kebijakan berikut menolak kemampuan pengguna untuk menambahkan atau menghapus tag untuk semua sumber daya. Anda kemudian dapat membuat kebijakan untuk mengizinkan pengguna tertentu menambahkan atau menghapus tag.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"DenyTagUpdates",
         "Effect":"Deny",
         "Action":[
            "rds:AddTagsToResource",
            "rds:RemoveTagsFromResource"
         ],
         "Resource":"*"
      }
   ]
}
```

Untuk melihat daftar tindakan Amazon RDS, lihat Tindakan yang Ditentukan oleh Amazon RDS di Referensi Otorisasi Layanan.

Contoh kebijakan: Menggunakan tag kustom

Contoh berikut menunjukkan cara menggunakan tag kustom dalam kebijakan izin IAM Amazon RDS. Untuk informasi lebih lanjut tentang cara menambahkan tag ke sumber daya Amazon RDS, lihat Bekerja dengan Amazon Resource Name (ARN) di Amazon RDS.

> **Note**
>
> Semua contoh menggunakan wilayah us-west-2 dan berisi ID akun fiktif.

Contoh 1: Memberikan izin untuk tindakan pada sumber daya dengan tag tertentu dengan dua nilai yang berbeda

Kebijakan berikut memungkinkan izin untuk melakukan operasi `CreateDBSnapshot` API pada instans DB dengan tag `stage` diatur ke `development` atau `test`.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllowAnySnapshotName",
         "Effect":"Allow",
         "Action":[
            "rds:CreateDBSnapshot"
         ],
         "Resource":"arn:aws:rds:*:123456789012:snapshot:*"
      },
      {
         "Sid":"AllowDevTestToCreateSnapshot",
         "Effect":"Allow",
         "Action":[
            "rds:CreateDBSnapshot"
         ],
         "Resource":"arn:aws:rds:*:123456789012:db:*",
         "Condition":{
            "StringEquals":{
                "rds:db-tag/stage":[
                  "development",
                  "test"
               ]
            }
```

```
         }
      }
   ]
}
```

Kebijakan berikut memungkinkan izin untuk melakukan operasi `ModifyDBInstance` API pada instans DB dengan tag `stage` diatur ke `development` atau `test`.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllowChangingParameterOptionSecurityGroups",
         "Effect":"Allow",
         "Action":[
            "rds:ModifyDBInstance"
         ],
         "Resource":" [
            "arn:aws:rds:*:123456789012:pg:*",
            "arn:aws:rds:*:123456789012:secgrp:*",
            "arn:aws:rds:*:123456789012:og:*"
         ]
      },
      {
         "Sid":"AllowDevTestToModifyInstance",
         "Effect":"Allow",
         "Action":[
            "rds:ModifyDBInstance"
         ],
         "Resource":"arn:aws:rds:*:123456789012:db:*",
         "Condition":{
            "StringEquals":{
                "rds:db-tag/stage":[
                  "development",
                  "test"
               ]
            }
         }
      }
   ]
}
```

Contoh 2: Secara eksplisit menolak izin untuk membuat instans DB yang menggunakan grup parameter DB yang ditentukan

Kebijakan berikut secara eksplisit menolak izin untuk membuat instans DB yang menggunakan grup parameter DB dengan nilai tag spesifik. Anda dapat menerapkan kebijakan ini jika Anda mengharuskan grup parameter DB yang dibuat pengguna tertentu selalu digunakan saat membuat instans DB. Kebijakan yang menggunakan `Deny` paling sering digunakan untuk membatasi akses yang diberikan oleh kebijakan yang lebih luas.

Izin yang secara tegas menolak lebih diprioritaskan daripada izin lain yang diberikan. Ini memastikan bahwa identitas tidak akan secara kebetulan mendapatkan izin yang tidak pernah ingin Anda berikan.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"DenyProductionCreate",
         "Effect":"Deny",
         "Action":"rds:CreateDBInstance",
         "Resource":"arn:aws:rds:*:123456789012:pg:*",
         "Condition":{
            "StringEquals":{
               "rds:pg-tag/usage":"prod"
            }
         }
      }
   ]
}
```

Contoh 3: Memberikan izin untuk tindakan pada instans DB dengan nama instans yang diawali dengan nama pengguna

Kebijakan berikut memungkinkan izin untuk memanggil API apa pun (kecuali untuk `AddTagsToResource` atau `RemoveTagsFromResource`) pada instans DB yang memiliki nama instans DB yang diawali dengan nama pengguna dan memiliki tag bernama `stage` yang sama dengan `devo` atau yang tidak memiliki tag bernama `stage`.

Baris `Resource` dalam kebijakan mengidentifikasi sumber daya berdasarkan Amazon Resource Name (ARN). Untuk informasi selengkapnya tentang cara menggunakan ARN dengan sumber daya Amazon RDS, lihat Bekerja dengan Amazon Resource Name (ARN) di Amazon RDS.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllowFullDevAccessNoTags",
         "Effect":"Allow",
         "NotAction":[
            "rds:AddTagsToResource",
            "rds:RemoveTagsFromResource"
         ],
         "Resource":"arn:aws:rds:*:123456789012:db:${aws:username}*",
         "Condition":{
            "StringEqualsIfExists":{
               "rds:db-tag/stage":"devo"
            }
         }
      }
   ]
}
```

AWS kebijakan terkelola untuk Amazon RDS

Untuk menambahkan izin ke set dan peran izin, lebih mudah menggunakan kebijakan AWS terkelola daripada menulis kebijakan sendiri. Dibutuhkan waktu dan keahlian untuk membuat kebijakan yang dikelola pelanggan IAM yang hanya memberi tim Anda izin yang mereka butuhkan. Untuk memulai dengan cepat, Anda dapat menggunakan kebijakan AWS terkelola kami. Kebijakan ini mencakup kasus penggunaan umum dan tersedia di Akun AWS Anda. Untuk informasi selengkapnya tentang kebijakan AWS AWS terkelola, lihat kebijakan terkelola di Panduan Pengguna IAM.

Layanan AWS memelihara dan memperbarui kebijakan AWS terkelola. Anda tidak dapat mengubah izin dalam kebijakan AWS terkelola. Layanan terkadang menambahkan izin tambahan ke kebijakan AWS terkelola untuk mendukung fitur baru. Jenis pembaruan ini akan memengaruhi semua identitas (pengguna, grup, dan peran) di mana kebijakan tersebut dilampirkan. Layanan kemungkinan besar akan memperbarui kebijakan AWS terkelola saat fitur baru diluncurkan atau saat operasi baru tersedia. Layanan tidak menghapus izin dari kebijakan AWS terkelola, sehingga pembaruan kebijakan tidak merusak izin yang ada.

Selain itu, AWS mendukung kebijakan terkelola untuk fungsi pekerjaan yang mencakup beberapa layanan. Misalnya, kebijakan `ReadOnlyAccess` AWS terkelola menyediakan akses hanya-baca ke semua Layanan AWS dan sumber daya. Saat layanan meluncurkan fitur baru, AWS tambahkan izin hanya-baca untuk operasi dan sumber daya baru. Untuk melihat daftar dan deskripsi dari kebijakan fungsi tugas, lihat kebijakan yang dikelola AWS untuk fungsi tugas di Panduan Pengguna IAM.

Topik

- AWS kebijakan terkelola: AmazonRDS ReadOnlyAccess
- AWS kebijakan terkelola: AmazonRDS FullAccess
- AWS kebijakan terkelola: AmazonRDS DataFullAccess
- AWS kebijakan terkelola: AmazonRDS EnhancedMonitoringRole
- AWS kebijakan terkelola: AmazonRDS PerformanceInsightsReadOnly
- AWS kebijakan terkelola: AmazonRDS PerformanceInsightsFullAccess
- AWS kebijakan terkelola: AmazonRDS DirectoryServiceAccess
- AWS kebijakan terkelola: AmazonRDS ServiceRolePolicy
- AWS kebijakan terkelola: AmazonRDS CustomServiceRolePolicy
- AWSkebijakan terkelola: Instans AmazonRDSCustom ProfileRolePolicy

AWS kebijakan terkelola: AmazonRDS ReadOnlyAccess

Kebijakan ini memungkinkan akses hanya-baca ke Amazon RDS melalui. AWS Management Console

Detail izin

Kebijakan ini mencakup izin berikut:

- `rds` – Mengizinkan pengguna utama mendeskripsikan sumber daya Amazon RDS dan mencantumkan tag untuk sumber daya Amazon RDS.
- `cloudwatch`— Memungkinkan kepala sekolah untuk mendapatkan statistik metrik Amazon CloudWatch .
- `ec2` – Mengizinkan pengguna utama mendeskripsikan Zona Ketersediaan dan sumber daya jaringan.
- `logs`— Memungkinkan prinsipal untuk menggambarkan aliran CloudWatch log Log dari grup log, dan mendapatkan CloudWatch peristiwa log Log.
- `devops-guru`— Memungkinkan prinsipal untuk mendeskripsikan sumber daya yang memiliki cakupan Amazon DevOps Guru, yang ditentukan baik oleh nama CloudFormation tumpukan atau tag sumber daya.

Untuk informasi selengkapnya tentang kebijakan ini, termasuk dokumen kebijakan JSON, lihat AmazonRDS ReadOnlyAccess di Panduan Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: AmazonRDS FullAccess

Kebijakan ini menyediakan akses penuh ke Amazon RDS melalui. AWS Management Console

Detail izin

Kebijakan ini mencakup izin berikut:

- `rds` – Mengizinkan pengguna utama memiliki akses penuh ke Amazon RDS.
- `application-autoscaling` – Mengizinkan pengguna utama mendeskripsikan dan mengelola target dan kebijakan penskalaan Application Auto Scaling.
- `cloudwatch`— Memungkinkan kepala sekolah mendapatkan statika CloudWatch metrik dan mengelola alarm. CloudWatch

- `ec2` – Mengizinkan pengguna utama mendeskripsikan Zona Ketersediaan dan sumber daya jaringan.
- `logs`— Memungkinkan prinsipal untuk menggambarkan aliran CloudWatch log Log dari grup log, dan mendapatkan CloudWatch peristiwa log Log.
- `outposts`— Memungkinkan prinsipal untuk mendapatkan AWS Outposts jenis instance.
- `pi` – Mengizinkan pengguna utama untuk mendapatkan metrik Wawasan Performa.
- `sns` – Mengizinkan pengguna utama untuk berlangganan dan topik Amazon Simple Notification Service (Amazon SNS), dan menerbitkan pesan Amazon SNS.
- `devops-guru`— Memungkinkan prinsipal untuk mendeskripsikan sumber daya yang memiliki cakupan Amazon DevOps Guru, yang ditentukan baik oleh nama CloudFormation tumpukan atau tag sumber daya.

Untuk informasi selengkapnya tentang kebijakan ini, termasuk dokumen kebijakan JSON, lihat AmazonRDS FullAccess di Panduan Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: AmazonRDS DataFullAccess

Kebijakan ini memungkinkan akses penuh untuk menggunakan Data API dan editor kueri pada Aurora Serverless klaster tertentu Akun AWS. Kebijakan ini memungkinkan Akun AWS untuk mendapatkan nilai rahasia dari AWS Secrets Manager.

Anda dapat melampirkan kebijakan `AmazonRDSDataFullAccess` ke identitas IAM Anda.

Detail izin

Kebijakan ini mencakup izin berikut:

- `dbqms` – Mengizinkan pengguna utama mengakses, menghapus, mendeskripsikan, dan memperbarui kueri. Layanan Metadata Kueri basis data (`dbqms`) adalah layanan khusus internal. Ini memberikan kueri terbaru dan tersimpan Anda untuk editor kueri di AWS Management Console untuk beberapa Layanan AWS, termasuk Amazon RDS.
- `rds-data` – Mengizinkan pengguna utama untuk menjalankan pernyataan SQL pada basis data Aurora Serverless.
- `secretsmanager`— Memungkinkan kepala sekolah untuk mendapatkan nilai rahasia dari. AWS Secrets Manager

Untuk informasi selengkapnya tentang kebijakan ini, termasuk dokumen kebijakan JSON, lihat AmazonRDS DataFullAccess di Panduan Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: AmazonRDS EnhancedMonitoringRole

Kebijakan ini menyediakan akses ke Amazon CloudWatch Logs untuk Amazon RDS Enhanced Monitoring.

Detail izin

Kebijakan ini mencakup izin berikut:

- `logs`— Memungkinkan prinsipal untuk membuat grup CloudWatch log Log dan kebijakan retensi, dan untuk membuat dan mendeskripsikan aliran CloudWatch log log dari grup log. Hal ini juga memungkinkan prinsipal untuk menempatkan dan mendapatkan peristiwa CloudWatch log Log.

Untuk informasi selengkapnya tentang kebijakan ini, termasuk dokumen kebijakan JSON, lihat AmazonRDS EnhancedMonitoringRole di Panduan Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: AmazonRDS PerformanceInsightsReadOnly

Kebijakan ini menyediakan akses hanya-baca ke Wawasan Performa Amazon RDS untuk instans DB Amazon RDS dan klaster DB Amazon Aurora.

Kebijakan ini kini mencakup `Sid` (ID pernyataan) sebagai pengidentifikasi pernyataan kebijakan.

Detail izin

Kebijakan ini mencakup izin berikut:

- `rds` – Mengizinkan pengguna utama mendeskripsikan instans DB Amazon RDS dan klaster DB Amazon Aurora.
- `pi` – Mengizinkan pengguna utama melakukan panggilan ke API Wawasan Performa Amazon RDS dan mengakses metrik Wawasan Performa.

Untuk informasi selengkapnya tentang kebijakan ini, termasuk dokumen kebijakan JSON, lihat AmazonRDS PerformanceInsightsReadOnly di Panduan Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: AmazonRDS PerformanceInsightsFullAccess

Kebijakan ini menyediakan akses penuh ke Wawasan Performa Amazon RDS untuk instans DB Amazon RDS dan klaster DB Amazon Aurora.

Kebijakan ini kini mencakup `Sid` (ID pernyataan) sebagai pengidentifikasi pernyataan kebijakan.

Detail izin

Kebijakan ini mencakup izin berikut:

- `rds` – Mengizinkan pengguna utama mendeskripsikan instans DB Amazon RDS dan klaster DB Amazon Aurora.
- `pi` – Mengizinkan pengguna utama melakukan panggilan ke API Wawasan Performa Amazon RDS, serta membuat, melihat, dan menghapus laporan analisis performa.
- `cloudwatch`— Memungkinkan kepala sekolah untuk membuat daftar semua metrik Amazon, dan mendapatkan CloudWatch data metrik dan statistik.

Untuk informasi selengkapnya tentang kebijakan ini, termasuk dokumen kebijakan JSON, lihat AmazonRDS PerformanceInsightsFullAccess di Panduan Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: AmazonRDS DirectoryServiceAccess

Kebijakan ini mengizinkan Amazon RDS untuk melakukan panggilan ke AWS Directory Service.

Detail izin

Kebijakan ini mencakup izin berikut:

- `ds`— Memungkinkan kepala sekolah untuk mendeskripsikan AWS Directory Service direktori dan mengontrol otorisasi ke direktori. AWS Directory Service

Untuk informasi selengkapnya tentang kebijakan ini, termasuk dokumen kebijakan JSON, lihat AmazonRDS DirectoryServiceAccess di Panduan Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: AmazonRDS ServiceRolePolicy

Anda tidak dapat melampirkan kebijakan `AmazonRDSServiceRolePolicy` ke entitas IAM Anda. Kebijakan ini dilampirkan ke peran tertaut layanan yang memungkinkan Amazon RDS melakukan

tindakan atas nama Anda. Untuk informasi selengkapnya, lihat Izin peran terkait layanan untuk Amazon RDS.

AWS kebijakan terkelola: AmazonRDS CustomServiceRolePolicy

Anda tidak dapat melampirkan kebijakan `AmazonRDSCustomServiceRolePolicy` ke entitas IAM Anda. Kebijakan ini dilampirkan ke peran tertaut layanan yang memungkinkan Amazon RDS melakukan tindakan atas nama Anda. Untuk informasi selengkapnya, lihat Izin peran terkait layanan untuk Amazon RDS Custom.

AWSkebijakan terkelola: Instans AmazonRDSCustom ProfileRolePolicy

Anda tidak harus melampirkan `AmazonRDSCustomInstanceProfileRolePolicy` ke entitas IAM Anda. Ini hanya boleh dilampirkan ke peran profil instans yang digunakan untuk memberikan izin ke instans Amazon RDS Custom DB Anda untuk melakukan berbagai tindakan otomatisasi dan tugas manajemen database. Teruskan profil instans sebagai `custom-iam-instance-profile` parameter selama pembuatan instans Kustom RDS dan RDS Custom mengaitkan profil instance ini ke instans DB Anda.

Detail izin

Kebijakan ini mencakup izin berikut:

- `ssm`,`ssmmessages`, `ec2messages` - Memungkinkan RDS Custom untuk berkomunikasi, menjalankan otomatisasi, dan memelihara agen pada instans DB melalui Systems Manager.
- `ec2`, `s3` - Memungkinkan RDS Custom untuk melakukan operasi pencadangan pada instans DB yang menyediakan kemampuan point-in-time pemulihan.
- `secretsmanager`- Memungkinkan RDS Custom untuk mengelola rahasia spesifik instans DB yang dibuat oleh RDS Custom.
- `cloudwatch`, `logs` - Memungkinkan RDS Custom untuk mengunggah metrik dan log instans DB CloudWatch melalui CloudWatch agen.
- `events`, `sqs` - Memungkinkan RDS Custom untuk mengirim dan menerima informasi status tentang instans DB.
- `kms`- Memungkinkan RDS Custom menggunakan kunci KMS khusus instance untuk melakukan enkripsi rahasia dan objek S3 yang dikelola RDS Custom.

Untuk informasi selengkapnya tentang kebijakan ini, termasuk dokumen kebijakan JSON, lihat Instans AmazonRDSCustom ProfileRolePolicy di Panduan Referensi Kebijakan AWS Terkelola.

Amazon RDS memperbarui kebijakan AWS terkelola

Lihat detail tentang pembaruan kebijakan AWS terkelola untuk Amazon RDS sejak layanan ini mulai melacak perubahan ini. Untuk menerima peringatan otomatis tentang perubahan pada halaman ini, berlangganan umpan RSS di halaman Riwayat dokumen Amazon RDS.

Perubahan	Deskripsi	Tanggal
AWS kebijakan terkelola untuk Amazon RDS – Kebijakan baru	Amazon RDS menambahkan kebijakan terkelola baru yang diberi nama `AmazonRDSCustomInstanceProfileRolePolicy` untuk memungkinkan RDS Custom melakukan tindakan otomatisasi dan tugas manajemen database melalui profil instans EC2. Untuk informasi selengkapnya, lihat AWS kebijakan terkelola untuk Amazon RDS.	Februari 27, 2024
Izin peran terkait layanan untuk Amazon RDS – Pembaruan pada kebijakan yang sudah ada	Amazon RDS menambahkan ID pernyataan baru ke `AmazonRDSServiceRolePolicy` peran `AWSServiceRoleForRDS` terkait layanan. Untuk informasi selengkapnya, lihat Izin peran terkait layanan untuk Amazon RDS.	Januari 19, 2024
AWS kebijakan terkelola untuk Amazon RDS – Pembaruan pada kebijakan yang ada	Kebijakan terkelola `AmazonRDSPerformanceInsightsReadOnly` dan `AmazonRDSPerformanceInsightsFullAcce`	23 Oktober 2023

Perubahan	Deskripsi	Tanggal
	`ss` kini menyertakan `Sid` (ID pernyataan) sebagai pengidentifikasi dalam pernyataan kebijakan. Untuk informasi selengkapnya, lihat AWS kebijakan terkelola : AmazonRDS PerformanceInsightsReadOnly dan AWS kebijakan terkelola : AmazonRDS PerformanceInsightsFullAccess	
Izin peran terkait layanan untuk Amazon RDS – Pembaruan pada kebijakan yang sudah ada	Amazon RDS menambahkan izin baru ke `AmazonRDSCustomServiceRolePolicy` peran terkait layanan `AWSServiceRoleForRDSCustom` . Izin baru ini memungkinkan RDS Custom for Oracle untuk membuat, memodifikasi, dan menghapus EventBridge Aturan Terkelola. Untuk informasi selengkapnya, lihat Izin peran terkait layanan untuk Amazon RDS Custom.	20 September 2023

Perubahan	Deskripsi	Tanggal
AWS kebijakan terkelola untuk Amazon RDS – Pembaruan pada kebijakan yang sudah ada	Amazon RDS menambahkan izin baru ke kebijakan terkelola `AmazonRDSFullAccess` . Izin ini memungkinkan Anda membuat, melihat, dan menghapus laporan analisis performa selama periode waktu tertentu. Untuk informasi selengkapnya tentang konfigurasi kebijakan akses untuk Wawasan Performa, lihat Mengonfigurasi kebijakan akses untuk Wawasan Performa	17 Agustus 2023

Perubahan	Deskripsi	Tanggal
AWS kebijakan terkelola untuk Amazon RDS – Kebijakan baru dan pembaruan pada kebijakan yang sudah ada	Amazon RDS menambahkan izin baru ke kebijakan terkelola `AmazonRDSPerformanceInsightsReadOnly` dan kebijakan terkelola baru bernama `AmazonRDSPerformanceInsightsFullAccess`. Izin ini memungkinkan Anda menganalisis Wawasan Performa selama periode tertentu, melihat hasil analisis beserta rekomendasinya, dan menghapus laporan. Untuk informasi selengkapnya tentang konfigurasi kebijakan akses untuk Wawasan Performa, lihat Mengonfigurasi kebijakan akses untuk Wawasan Performa	16 Agustus 2023
Izin peran terkait layanan untuk Amazon RDS – Pembaruan pada kebijakan yang sudah ada	Amazon RDS menambahkan izin baru ke `AmazonRDSCustomServiceRolePolicy` peran terkait layanan `AWSServiceRoleForRDSCustom`. Izin baru ini memungkinkan RDS Custom for Oracle menggunakan snapshot DB. Untuk informasi selengkapnya, lihat Izin peran terkait layanan untuk Amazon RDS Custom.	23 Juni 2023

Perubahan	Deskripsi	Tanggal
Izin peran terkait layanan untuk Amazon RDS – Pembaruan pada kebijakan yang sudah ada	Amazon RDS menambahkan izin baru ke `AmazonRDSCustomServiceRolePolicy` peran terkait layanan `AWSServiceRoleForRDSCustom`. Izin baru ini memungkinkan RDS Custom for Oracle menggunakan snapshot DB. Untuk informasi selengkapnya, lihat Izin peran terkait layanan untuk Amazon RDS Custom.	23 Juni 2023
Izin peran terkait layanan untuk Amazon RDS – Pembaruan pada kebijakan yang sudah ada	Amazon RDS menambahkan izin baru ke `AmazonRDSCustomServiceRolePolicy` peran terkait layanan `AWSServiceRoleForRDSCustom`. Izin baru ini memungkinkan RDS Custom membuat antarmuka jaringan. Untuk informasi selengkapnya, lihat Izin peran terkait layanan untuk Amazon RDS Custom.	30 Mei 2023

Perubahan	Deskripsi	Tanggal
Izin peran terkait layanan untuk Amazon RDS – Pembaruan pada kebijakan yang sudah ada	Amazon RDS menambahkan izin baru ke `AmazonRDSCustomServiceRolePolicy` peran terkait layanan `AWSServiceRoleForRDSCustom` . Izin baru ini memungkinkan RDS Custom memanggil Amazon EBS untuk memeriksa kuota penyimpanan. Untuk informasi selengkapnya, lihat Izin peran terkait layanan untuk Amazon RDS Custom.	18 April 2023

Perubahan	Deskripsi	Tanggal
Izin peran terkait layanan untuk Amazon RDS – Pembaruan pada kebijakan yang sudah ada	Amazon RDS Custom menambahkan izin baru ke `AmazonRDSCustomServiceRolePolicy` peran terkait layanan `AWSServiceRoleForRDSCustom` untuk berintegrasi dengan Amazon SQS. RDS Custom perlu terintegrasi dengan Amazon SQS untuk membuat dan mengelola antrean SQS di akun pelanggan. Nama antrean SQS mengikuti format `do-not-delete-rds-custom-[identifier]` dan diberi tag dengan `Amazon RDS Custom`. Izin untuk `ec2:CreateSnapshot` juga ditambahkan untuk memungkinkan RDS Custom membuat cadangan volume yang dilampirkan ke instans. Untuk informasi selengkapnya, lihat Izin peran terkait layanan untuk Amazon RDS Custom.	6 April 2023

Perubahan	Deskripsi	Tanggal
AWS kebijakan terkelola untuk Amazon RDS – Pembaruan pada kebijakan yang sudah ada	Amazon RDS menambahkan CloudWatch namespace `ListMetrics` Amazon baru ke dan. `AmazonRDSFullAccess` `AmazonRDSReadOnlyAccess` Nama ruang ini diperlukan agar Amazon RDS dapat mencantumkan metrik penggunaan sumber daya tertentu. Untuk informasi selengkapnya, lihat Ringkasan mengelola izin akses ke CloudWatch sumber daya Anda di Panduan CloudWatch Pengguna Amazon.	4 April 2023
AWS kebijakan terkelola untuk Amazon RDS – Pembaruan pada kebijakan yang sudah ada	Amazon RDS menambahkan izin baru `AmazonRDSFullAccess` dan `AmazonRDSReadOnlyAccess` mengelola kebijakan untuk memungkinkan tampilan temuan Amazon DevOps Guru di konsol RDS. Izin ini diperlukan untuk memungkinkan tampilan temuan DevOps Guru. Untuk informasi selengkapnya, lihat pembaruan Amazon RDS ke kebijakan AWS terkelola.	30 Maret 2023

Perubahan	Deskripsi	Tanggal
Izin peran terkait layanan untuk Amazon RDS – Pembaruan pada kebijakan yang sudah ada	Amazon RDS menambahkan izin baru ke `AmazonRDSServiceRolePolicy` peran `AWSServiceRoleForRDS` terkait layanan untuk integrasi. AWS Secrets Manager RDS perlu berintegrasi dengan Secrets Manager untuk mengelola kata sandi pengguna utama di Secrets Manager. Rahasia menggunakan konvensi penamaan terpesan dan membatasi pembaruan pelanggan. Untuk informasi selengkapnya, lihat Manajemen kata sandi dengan Amazon RDS Aurora dan AWS Secrets Manager.	22 Desember 2022

Perubahan	Deskripsi	Tanggal
Izin peran terkait layanan untuk Amazon RDS – Pembaruan pada kebijakan yang sudah ada	Amazon RDS menambahkan izin baru ke `AmazonRDSCustomServiceRolePolicy` peran terkait layanan `AWSServiceRoleForRDSCustom` . RDS Custom mendukung klaster DB. Izin baru dalam kebijakan ini memungkinkan RDS Custom menelepon Layanan AWS atas nama kluster DB Anda. Untuk informasi selengkapnya, lihat Izin peran terkait layanan untuk Amazon RDS Custom.	9 November 2022

Perubahan	Deskripsi	Tanggal
Izin peran terkait layanan untuk Amazon RDS – Pembaruan pada kebijakan yang sudah ada	Amazon RDS menambahkan izin baru ke peran terkait layanan `AWSServiceRoleForRDS` untuk berintegrasi dengan AWS Secrets Manager. Integrasi dengan Secrets Manager diperlukan agar SQL Server Reporting Services (SSRS) Email dapat digunakan di RDS. SSRS Email membuat rahasia atas nama pelanggan. Rahasia menggunakan konvensi penamaan terpesan dan membatasi pembaruan pelanggan. Untuk informasi selengkapnya, lihat Menggunakan Email SSRS untuk mengirim laporan.	26 Agustus 2022

Perubahan	Deskripsi	Tanggal
Izin peran terkait layanan untuk Amazon RDS – Pembaruan pada kebijakan yang sudah ada	Amazon RDS menambahkan CloudWatch namespace Amazon baru ke for. `AmazonRDSPreviewServiceRolePolicy` `PutMetricData` Nama ruang ini diperlukan agar Amazon RDS dapat menerbitkan metrik penggunaan sumber daya. Untuk informasi selengkapnya, lihat Menggunakan tombol kondisi untuk membatasi akses ke CloudWatch ruang nama di CloudWatch Panduan Pengguna Amazon.	7 Juni 2022

Perubahan	Deskripsi	Tanggal
Izin peran terkait layanan untuk Amazon RDS – Pembaruan pada kebijakan yang sudah ada	Amazon RDS menambahkan CloudWatch namespace Amazon baru ke for. `AmazonRDSBetaServiceRolePolicy` `PutMetricData` Nama ruang ini diperlukan agar Amazon RDS dapat menerbitkan metrik penggunaan sumber daya. Untuk informasi selengkapnya, lihat Menggunakan tombol kondisi untuk membatasi akses ke CloudWatch ruang nama di CloudWatch Panduan Pengguna Amazon.	7 Juni 2022
Izin peran terkait layanan untuk Amazon RDS – Pembaruan pada kebijakan yang sudah ada	Amazon RDS menambahkan CloudWatch namespace Amazon baru ke for. `AWSServiceRoleForRDS` `PutMetricData` Nama ruang ini diperlukan agar Amazon RDS dapat menerbitkan metrik penggunaan sumber daya. Untuk informasi selengkapnya, lihat Menggunakan tombol kondisi untuk membatasi akses ke CloudWatch ruang nama di CloudWatch Panduan Pengguna Amazon.	22 April 2022

Perubahan	Deskripsi	Tanggal
Izin peran terkait layanan untuk Amazon RDS – Pembaruan pada kebijakan yang sudah ada	Amazon RDS menambahkan izin baru ke peran terkait layanan `AWSServiceRoleForRDS` untuk mengelola izin untuk kolam IP milik pelanggan dan tabel rute gateway lokal (LGW-RTB). Izin ini diperlukan agar RDS on Outposts dapat melakukan replikasi multi-AZ di seluruh jaringan lokal Outpost. Untuk informasi selengkapnya, lihat Mengelola deployment Multi-AZ untuk Amazon RDS di AWS Outposts.	19 April 2022
Kebijakan berbasis identitas – Pembaruan pada kebijakan yang sudah ada	Amazon RDS menambahkan izin baru ke kebijakan yang dikelola `AmazonRDSFullAccess` untuk mendeskripsikan izin di LGW-RTB. Izin ini diperlukan untuk menjelaskan izin agar RDS on Outposts dapat melakukan replikasi multi-AZ di seluruh jaringan lokal Outpost. Untuk informasi selengkapnya, lihat Mengelola deployment Multi-AZ untuk Amazon RDS di AWS Outposts.	19 April 2022

Perubahan	Deskripsi	Tanggal
AWS kebijakan terkelola untuk Amazon RDS – Kebijakan baru	Amazon RDS menambahkan kebijakan terkelola baru yang diberi nama `AmazonRDSPerformanceInsightsReadOnly` untuk mengizinkan Amazon RDS memanggil AWS layanan atas nama instans DB Anda. Untuk informasi selengkapnya tentang konfigurasi kebijakan akses untuk Wawasan Performa, lihat Mengonfigurasi kebijakan akses untuk Wawasan Performa	10 Maret 2022
Izin peran terkait layanan untuk Amazon RDS – Pembaruan pada kebijakan yang sudah ada	Amazon RDS menambahkan CloudWatch ruang nama Amazon baru ke for. `AWSServiceRoleForRDS` `PutMetricData` Ruang nama ini diperlukan untuk Amazon DocumentDB (dengan kompatibilitas MongoDB) dan Amazon Neptune untuk menerbitkan metrik. CloudWatch Untuk informasi selengkapnya, lihat Menggunakan tombol kondisi untuk membatasi akses ke CloudWatch ruang nama di CloudWatch Panduan Pengguna Amazon.	4 Maret 2022

Perubahan	Deskripsi	Tanggal
Izin peran terkait layanan untuk Amazon RDS Custom – Kebijakan baru	Amazon RDS menambahkan peran terkait layanan baru bernama `AWSServiceRoleForRDSCustom` agar RDS Custom dapat memanggil Layanan AWS atas nama instans DB Anda.	26 Oktober 2021
Amazon RDS mulai melacak perubahan	Amazon RDS mulai melacak perubahan untuk kebijakan yang AWS dikelola.	26 Oktober 2021

Pencegahan masalah confused deputy lintas layanan

Masalah confused deputy adalah masalah keamanan saat entitas yang tidak memiliki izin untuk melakukan suatu tindakan dapat memaksa entitas yang lebih berhak untuk melakukan tindakan tersebut. Di AWS, peniruan identitas lintas layanan dapat mengakibatkan masalah confused deputy.

Peniruan identitas lintas layanan dapat terjadi ketika satu layanan (layanan panggilan) memanggil layanan lain (layanan yang dipanggil). Layanan panggilan dapat dimanipulasi agar 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 dapat membantu Anda melindungi data untuk semua layanan dengan pengguna utama layanan yang telah diberi akses ke sumber daya di akun Anda. Untuk informasi selengkapnya, lihat Masalah confused deputy di Panduan Pengguna IAM.

Untuk membatasi izin yang diberikan Amazon RDS ke layanan lain untuk sumber daya tertentu, sebaiknya gunakan kunci konteks kondisi global `aws:SourceArn` dan `aws:SourceAccount` di kebijakan sumber daya.

Dalam beberapa kasus, nilai `aws:SourceArn` tidak berisi ID akun, misalnya saat Anda menggunakan Amazon Resource Name (ARN) untuk bucket Amazon S3. Dalam kasus ini, pastikan untuk menggunakan kedua kunci konteks kondisi global untuk membatasi izin. Dalam beberapa kasus, Anda menggunakan kunci konteks kondisi global dan nilai `aws:SourceArn` yang berisi ID akun. Dalam hal ini, pastikan bahwa nilai `aws:SourceAccount` dan akun dalam nilai `aws:SourceArn` menggunakan ID akun yang sama ketika digunakan dalam pernyataan yang sama. Jika Anda ingin hanya satu sumber daya yang akan dikaitkan dengan akses lintas layanan, gunakan `aws:SourceArn`. Jika Anda ingin mengizinkan semua sumber daya di akun AWS yang ditentukan dikaitkan dengan penggunaan lintas layanan, gunakan `aws:SourceAccount`.

Pastikan nilai `aws:SourceArn` adalah ARN untuk jenis sumber daya Amazon RDS. Untuk informasi selengkapnya, lihat Bekerja dengan Amazon Resource Name (ARN) di Amazon RDS.

Cara paling efektif untuk melindungi dari masalah confused deputy adalah dengan menggunakan kunci konteks kondisi global `aws:SourceArn` dengan ARN lengkap sumber daya. Dalam beberapa kasus, Anda mungkin tidak mengetahui ARN lengkap sumber daya atau mungkin Anda menentukan beberapa sumber daya. Dalam hal ini, gunakan kunci kondisi konteks global `aws:SourceArn` dengan wildcard (*) untuk bagian ARN yang tidak diketahui. Contohnya adalah `arn:aws:rds:*:123456789012:*`.

Contoh berikut menunjukkan cara Anda dapat menggunakan `aws:SourceArn` dan kunci konteks kondisi global `aws:SourceAccount` di Amazon RDS untuk mencegah masalah confused deputy.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "rds.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:rds:us-east-1:123456789012:db:mydbinstance"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

Untuk contoh kebijakan lainnya yang menggunakan kunci konteks kondisi global `aws:SourceArn` dan `aws:SourceAccount`, lihat bagian berikut:

- Memberikan izin untuk menerbitkan pemberitahuan ke topik Amazon SNS
- Membuat peran IAM secara manual untuk pencadangan dan pemulihan native
- Mengatur Autentikasi Windows untuk instans DB SQL Server
- Prasyarat untuk mengintegrasikan RDS for SQL Server dengan S3
- Membuat peran IAM secara manual untuk SQL Server Audit
- Mengonfigurasi izin IAM untuk integrasi RDS for Oracle dengan Amazon S3
- Mengatur akses ke bucket Amazon S3 (Impor PostgreSQL)
- Menyiapkan akses ke bucket Amazon S3 (Ekspor PostgreSQL)

Autentikasi basis data IAM untuk MariaDB, MySQL, dan PostgreSQL

Anda dapat mengautentikasi ke klaster DB menggunakan autentikasi basis data AWS Identity and Access Management (IAM). Autentikasi basis data IAM bekerja dengan MariaDB, MySQL, dan PostgreSQL. Dengan metode autentikasi ini, Anda tidak perlu menggunakan kata sandi saat terhubung ke klaster DB. Sebagai gantinya, Anda menggunakan token autentikasi.

Token autentikasi adalah string karakter unik yang dihasilkan Amazon RDS atas permintaan. Token autentikasi dibuat menggunakan AWS Signature Versi 4. Setiap token memiliki masa pakai 15 menit. Anda tidak perlu menyimpan kredensial pengguna dalam basis data, karena autentikasi dikelola secara eksternal menggunakan IAM. Anda juga masih dapat menggunakan autentikasi basis data standar. Token hanya digunakan untuk autentikasi dan tidak memengaruhi sesi setelah dibuat.

Autentikasi basis data IAM memberikan manfaat berikut:

- Lalu lintas jaringan ke dan dari basis data dienkripsi menggunakan Lapisan Soket Aman (SSL) atau Keamanan Lapisan Pengangkutan (TLS). Untuk informasi selengkapnya tentang cara menggunakan SSL/TLS bersama Amazon RDS, lihat .
- Anda dapat menggunakan IAM untuk mengelola akses ke sumber daya basis data Anda secara terpusat, bukan mengelola akses satu per satu pada klaster DB.
- Untuk aplikasi yang berjalan di Amazon EC2, Anda dapat menggunakan kredensial profil khusus untuk instans EC2 untuk mengakses basis data, bukan menggunakan kata sandi, untuk keamanan yang lebih baik.

Secara umum, pertimbangkan untuk menggunakan autentikasi basis data IAM saat aplikasi Anda membuat kurang dari 200 koneksi per detik, dan Anda tidak ingin mengelola nama pengguna dan kata sandi secara langsung dalam kode aplikasi Anda.

Driver JDBC AWS untuk MySQL mendukung autentikasi basis data IAM. Untuk informasi selengkapnya, lihat Autentikasi Database AWS IAM di AWS Driver JDBC untuk repositori MySQL. GitHub

Topik

- Kunci konteks kondisi global AWS yang tidak didukung
- Mengaktifkan dan menonaktifkan autentikasi basis data IAM
- Membuat dan menggunakan kebijakan IAM untuk akses basis data IAM
- Membuat akun basis data menggunakan autentikasi IAM
- Menghubungkan ke instans DB menggunakan autentikasi IAM

Ketersediaan Wilayah dan versi

Ketersediaan dan dukungan fitur bervariasi di berbagai versi khusus dari setiap mesin basis data, dan di seluruh Wilayah AWS. Untuk informasi selengkapnya tentang ketersediaan versi dan Wilayah dengan autentikasi basis data IAM dan Amazon RDS, lihat. Autentikasi basis data IAM

Dukungan CLI dan SDK

Autentikasi basis data IAM tersedia untuk AWS CLI dan untuk SDK AWS khusus bahasa berikut:

- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby

Batasan untuk autentikasi basis data IAM

Saat menggunakan autentikasi basis data IAM, batasan berikut berlaku:

- Jumlah maksimum koneksi per detik untuk klaster DB Anda mungkin akan dibatasi, bergantung pada kelas instans DB-nya dan beban kerja Anda. Otentikasi IAM dapat gagal jika terjadi kehabisan sumber daya selama beban DB puncak.
- Saat ini, autentikasi basis data IAM tidak mendukung kunci konteks kondisi global.

  Untuk informasi selengkapnya tentang kunci konteks kondisi global, lihat AWS kunci konteks kondisi global dalam Panduan Pengguna IAM.

- Untuk PostgreSQL, jika peran IAM (`rds_iam`) ditambahkan ke pengguna (termasuk pengguna master RDS), autentikasi IAM diprioritaskan atas autentikasi kata sandi, sehingga pengguna harus login sebagai pengguna IAM.
- Untuk PostgreSQL, Amazon RDS tidak mendukung pengaktifan metode autentikasi IAM dan Kerberos secara bersamaan.
- Untuk PostgreSQL, Anda tidak dapat menggunakan autentikasi IAM untuk membuat koneksi replikasi.
- Anda tidak dapat menggunakan data DNS Route 53 kustom sebagai pengganti titik akhir klaster DB untuk menghasilkan token autentikasi.

Rekomendasi untuk autentikasi basis data IAM

Kami merekomendasikan hal berikut saat menggunakan autentikasi basis data IAM:

- Gunakan autentikasi basis data IAM saat aplikasi Anda membutuhkan kurang dari 200 koneksi autentikasi basis data IAM baru per detik.

  Mesin basis data yang bekerja dengan Amazon RDS tidak memberlakukan batasan apa pun pada upaya autentikasi per detik. Namun, saat Anda menggunakan autentikasi basis data IAM, aplikasi Anda harus membuat token autentikasi. Aplikasi Anda kemudian menggunakan token tersebut untuk terhubung ke klaster DB. Jika Anda melebihi batas maksimum untuk koneksi baru per detik, maka overhead tambahan dari autentikasi basis data IAM dapat menyebabkan throttling koneksi.

  Pertimbangkan untuk menggunakan penyatuan koneksi di aplikasi Anda untuk memitigasi pembuatan koneksi yang konstan. Cara ini dapat mengurangi overhead dari autentikasi DB IAM dan memungkinkan aplikasi Anda menggunakan kembali koneksi yang ada. Atau, pertimbangkan untuk menggunakan Proksi RDS untuk kasus penggunaan ini. Proksi RDS memiliki biaya tambahan. Lihat Harga Proksi RDS.
- Ukuran token autentikasi basis data IAM bergantung pada banyak hal termasuk jumlah tag IAM, kebijakan layanan IAM, panjang ARN, serta properti basis data dan IAM lainnya. Ukuran minimum token ini umumnya sekitar 1 KB tetapi bisa saja lebih besar. Karena token ini digunakan sebagai kata sandi dalam string koneksi ke basis data yang menggunakan autentikasi IAM, Anda harus memastikan bahwa driver basis data Anda (misalnya, ODBC) dan/atau alat apa pun tidak membatasi atau memotong token ini dikarenakan ukurannya. Token yang terpotong akan menyebabkan kegagalan validasi autentikasi oleh basis data dan IAM.

- Jika Anda menggunakan kredensial temporer saat membuat token autentikasi basis data IAM, kredensial temporer masih harus valid saat menggunakan token autentikasi basis data IAM untuk membuat permintaan koneksi.

Kunci konteks kondisi global AWS yang tidak didukung

Autentikasi basis data IAM tidak mendukung subset kunci konteks kondisi global AWS berikut.

- `aws:Referer`
- `aws:SourceIp`
- `aws:SourceVpc`
- `aws:SourceVpce`
- `aws:UserAgent`
- `aws:VpcSourceIp`

Untuk informasi selengkapnya, lihat kunci konteks kondisi global AWS dalam Panduan Pengguna IAM.

Mengaktifkan dan menonaktifkan autentikasi basis data IAM

Secara default, autentikasi basis data IAM dinonaktifkan di instans DB. Anda dapat mengaktifkan atau menonaktifkan autentikasi basis data IAM menggunakan AWS Management Console, AWS CLI, atau API.

Anda dapat mengaktifkan autentikasi basis data IAM saat Anda melakukan salah satu tindakan berikut:

- Untuk membuat instans DB yang baru dengan autentikasi basis data IAM diaktifkan, lihat Membuat instans DB Amazon RDS.
- Untuk memodifikasi instans DB untuk mengaktifkan autentikasi basis data IAM, lihat Memodifikasi instans DB Amazon RDS.
- Untuk memulihkan instans DB dari snapshot dengan autentikasi basis data IAM diaktifkan, lihat Memulihkan dari snapshot DB.
- Untuk memulihkan instans DB ke titik waktu dengan autentikasi basis data IAM diaktifkan, lihat Memulihkan instans DB dengan waktu yang ditentukan.

Autentikasi IAM untuk instans DB PostgreSQL mengharuskan nilai SSL berupa 1. Anda tidak dapat mengaktifkan autentikasi IAM untuk instans DB PostgreSQL jika nilai SSL adalah 0. Anda tidak dapat mengubah nilai SSL ke 0 jika autentikasi IAM diaktifkan untuk instans DB PostgreSQL.

Konsol

Setiap alur kerja pembuatan atau modifikasi memiliki bagian Autentikasi basis data, tempat Anda dapat mengaktifkan atau menonaktifkan autentikasi basis data IAM. Di bagian tersebut, pilih Kata sandi dan autentikasi basis data IAM untuk mengaktifkan autentikasi basis data IAM.

Untuk mengaktifkan atau menonaktifkan autentikasi basis data IAM untuk instans DB yang ada

1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data.
3. Pilih instans DB yang ingin Anda ubah.

   > **Note**
   >
   > Pastikan bahwa instans DB kompatibel dengan autentikasi IAM. Periksa persyaratan kompatibilitas dalam Ketersediaan Wilayah dan versi.

4. Pilih Ubah.
5. Di bagian Autentikasi basis data, pilih Kata sandi dan autentikasi basis data IAM untuk mengaktifkan autentikasi basis data IAM. Pilih Autentikasi kata sandi atau Kata sandi dan autentikasi Kerberos untuk menonaktifkan autentikasi IAM.
6. Pilih Lanjutkan.
7. Untuk segera menerapkan perubahan, pilih Segera di bagian Penjadwalan modifikasi.
8. Pilih Modifikasi instans DB .

AWS CLI

Untuk membuat instans DB baru dengan autentikasi IAM menggunakan AWS CLI, gunakan perintah `create-db-instance`. Tentukan opsi `--enable-iam-database-authentication`, seperti yang ditunjukkan dalam contoh berikut.

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --db-instance-class db.m3.medium \
```

```
    --engine MySQL \
    --allocated-storage 20 \
    --master-username masterawsuser \
    --manage-master-user-password \
    --enable-iam-database-authentication
```

Untuk memperbarui instans DB yang ada agar memiliki atau tidak memiliki autentikasi IAM, gunakan perintah AWS CLI `modify-db-instance`. Tentukan opsi `--enable-iam-database-authentication` atau `--no-enable-iam-database-authentication`, sesuai kebutuhan.

> **Note**
>
> Pastikan bahwa instans DB kompatibel dengan autentikasi IAM. Periksa persyaratan kompatibilitas dalam Ketersediaan Wilayah dan versi.

Secara default, Amazon RDS melakukan modifikasi selama periode pemeliharaan berikutnya. Jika Anda ingin menggantinya dan mengaktifkan autentikasi DB IAM sesegera mungkin, gunakan parameter `--apply-immediately`.

Contoh berikut menunjukkan cara untuk segera mengaktifkan autentikasi IAM untuk instans DB yang sudah ada.

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --apply-immediately \
    --enable-iam-database-authentication
```

Jika Anda memulihkan sebuah instans DB, gunakan salah satu perintah AWS CLI berikut:

- `restore-db-instance-to-point-in-time`
- `restore-db-instance-from-db-snapshot`

Pengaturan autentikasi basis data IAM akan ditetapkan secara default ke pengaturan untuk snapshot sumber. Untuk mengubah pengaturan ini, atur opsi `--enable-iam-database-authentication` atau `--no-enable-iam-database-authentication` sebagaimana diperlukan.

API RDS

Untuk membuat instans DB baru dengan autentikasi IAM dengan menggunakan API, gunakan operasi API `CreateDBInstance`. Atur parameter `EnableIAMDatabaseAuthentication` ke `true`.

Untuk memperbarui instans DB yang ada agar memiliki autentikasi IAM, gunakan operasi API `ModifyDBInstance`. Atur parameter `EnableIAMDatabaseAuthentication` ke `true` untuk mengaktifkan autentikasi IAM, atau `false` untuk menonaktifkannya.

> **Note**
>
> Pastikan bahwa instans DB kompatibel dengan autentikasi IAM. Periksa persyaratan kompatibilitas dalam Ketersediaan Wilayah dan versi.

Jika Anda memulihkan instans DB, gunakan salah satu operasi API berikut:

- `RestoreDBInstanceFromDBSnapshot`
- `RestoreDBInstanceToPointInTime`

Pengaturan autentikasi basis data IAM akan ditetapkan secara default ke pengaturan untuk snapshot sumber. Untuk mengubah pengaturan ini, atur parameter `EnableIAMDatabaseAuthentication` ke `true` untuk mengaktifkan autentikasi IAM, atau `false` untuk menonaktifkannya.

Membuat dan menggunakan kebijakan IAM untuk akses basis data IAM

Untuk memungkinkan pengguna atau peran terhubung ke instans DB, Anda harus membuat kebijakan IAM. Setelah itu, lampirkan kebijakan tersebut ke set izin atau peran.

> **Note**
>
> Untuk mempelajari selengkapnya tentang kebijakan IAM, lihat Manajemen identitas dan akses untuk Amazon RDS.

Contoh kebijakan berikut memungkinkan pengguna terhubung ke instans DB menggunakan autentikasi basis data IAM.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
             "rds-db:connect"
         ],
         "Resource": [
             "arn:aws:rds-db:us-east-2:1234567890:dbuser:db-ABCDEFGHIJKL01234/db_user"
         ]
      }
   ]
}
```

Important

Pengguna dengan izin administrator dapat mengakses instans DB tanpa izin eksplisit dalam kebijakan IAM. Jika Anda ingin membatasi akses administrator ke instans DB, Anda dapat membuat peran IAM yang dengan izin istimewa yang sesuai dan menetapkannya ke administrator.

Note

Jangan samakan awalan `rds-db:` dengan awalan operasi API RDS lain yang diawali dengan `rds:`. Anda menggunakan awalan `rds-db:` dan tindakan `rds-db:connect` hanya untuk autentikasi basis data IAM. Hal ini tidak berlaku dalam konteks lainnya.

Contoh kebijakan ini mencakup satu pernyataan dengan elemen berikut:

- `Effect` – Tentukan `Allow` untuk memberikan akses ke instans DB. Jika Anda tidak secara eksplisit mengizinkan akses, maka akses ditolak secara default.
- `Action` – Tentukan `rds-db:connect` untuk memungkinkan koneksi ke instans DB.
- `Resource` – Tentukan Amazon Resource Name (ARN) yang menjelaskan satu akun basis data dalam satu instans DB. Format ARN adalah sebagai berikut.

```
arn:aws:rds-db:region:account-id:dbuser:DbiResourceId/db-user-name
```

Dalam format ini, ganti hal berikut:

- *region* adalah Wilayah AWS untuk instans DB. Dalam contoh kebijakan, Wilayah AWS adalah `us-east-2`.
- *account-id* adalah nomor akun AWS untuk instans DB. Dalam contoh kebijakan, nomor akun adalah `1234567890`. Pengguna harus berada di akun yang sama dengan akun untuk instans DB.

  Untuk melakukan akses lintas akun, buat peran IAM dengan kebijakan yang ditunjukkan di atas di akun untuk instans DB dan izinkan akun Anda yang lain untuk mengambil peran tersebut.
- *DbiResourceId* adalah pengidentifikasi untuk instans DB. Pengidentifikasi ini unik untuk Wilayah AWS dan tidak pernah berubah. Dalam contoh kebijakan, pengidentifikasi adalah `db-ABCDEFGHIJKL01234`.

  Untuk menemukan ID sumber daya instans DB AWS Management Console untuk Amazon RDS, pilih instans DB untuk melihat detailnya. Kemudian, pilih tab Konfigurasi. ID Sumber Daya ditampilkan di bagian Konfigurasi.

  Alternatifnya, Anda dapat menggunakan perintah AWS CLI untuk menampilkan daftar pengidentifikasi dan ID sumber daya untuk semua instans DB Anda di Wilayah AWS saat ini, seperti yang ditunjukkan berikut.

  ```
  aws rds describe-db-instances --query "DBInstances[*].
  [DBInstanceIdentifier,DbiResourceId]"
  ```

  Jika Anda menggunakan Amazon Aurora, tentukan `DbClusterResourceId`, bukan `DbiResourceId`. Untuk informasi selengkapnya, lihat Membuat dan menggunakan kebijakan IAM untuk akses basis data IAM dalam Panduan Pengguna Amazon Aurora.

> **Note**
>
> Jika Anda terhubung ke basis data melalui Proksi RDS, tentukan ID sumber daya proksi, seperti `prx-ABCDEFGHIJKL01234`. Untuk informasi tentang menggunakan autentikasi basis data IAM dengan Proksi RDS, lihat Terhubung ke sebuah proksi menggunakan autentikasi IAM.

- *`db-user-name`* adalah nama akun basis data untuk dikaitkan dengan autentikasi IAM. Dalam contoh kebijakan, akun basis data adalah `db_user`.

Anda dapat membuat ARN lain untuk mendukung berbagai pola akses. Kebijakan berikut memungkinkan akses ke dua akun basis data yang berbeda dalam instans DB.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
             "rds-db:connect"
         ],
         "Resource": [
             "arn:aws:rds-db:us-east-2:123456789012:dbuser:db-ABCDEFGHIJKL01234/
jane_doe",
             "arn:aws:rds-db:us-east-2:123456789012:dbuser:db-ABCDEFGHIJKL01234/
mary_roe"
         ]
      }
   ]
}
```

Kebijakan berikut menggunakan karakter “*” untuk mencocokkan semua instans DB dan akun basis data untuk akun AWS dan Wilayah AWS tertentu.

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Effect": "Allow",
            "Action": [
                "rds-db:connect"
            ],
            "Resource": [
                "arn:aws:rds-db:us-east-2:1234567890:dbuser:*/*"
            ]
        }
    ]
}
```

Kebijakan berikut mencocokkan semua instans DB untuk akun AWS dan Wilayah AWS tertentu. Namun, kebijakan ini hanya memberikan akses ke instans DB yang memiliki akun basis data `jane_doe`.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
             "rds-db:connect"
         ],
         "Resource": [
             "arn:aws:rds-db:us-east-2:123456789012:dbuser:*/jane_doe"
         ]
      }
   ]
}
```

Pengguna atau peran hanya memiliki akses ke basis data yang aksesnya dimiliki oleh pengguna basis data tersebut. Misalnya, anggaplah instans DB Anda memiliki basis data bernama dev, dan basis data lain bernama uji. Jika pengguna basis data `jane_doe` hanya memiliki akses ke dev, setiap pengguna atau peran IAM yang mengakses instans DB tersebut dengan pengguna `jane_doe` juga hanya akan memiliki akses ke dev. Pembatasan akses ini juga berlaku untuk objek basis data lain, seperti tabel, tampilan, dan sebagainya.

Administrator harus membuat kebijakan IAM yang memberikan izin pada entitas untuk melakukan operasi API tertentu pada sumber daya yang diperlukan. Administrator kemudian harus melampirkan kebijakan tersebut ke set izin atau peran yang memerlukan izin tersebut. Untuk contoh kebijakan, lihat Contoh kebijakan berbasis identitas untuk Amazon RDS.

Melampirkan kebijakan IAM ke set izin atau peran

Setelah membuat kebijakan IAM untuk memungkinkan autentikasi basis data, Anda perlu melampirkan kebijakan tersebut ke set izin atau peran. Untuk tutorial tentang topik ini, lihat Buat dan lampirkan kebijakan yang dikelola pelanggan pertama Anda dalam Panduan Pengguna IAM.

Saat mengikuti tutorial ini, Anda dapat menggunakan salah satu contoh kebijakan yang ditunjukkan dalam bagian ini sebagai titik awal dan menyesuaikannya dengan kebutuhan Anda. Di akhir tutorial, Anda akan memiliki set izin dengan kebijakan terlampir yang dapat menggunakan tindakan `rds-db:connect`.

> **Note**
>
> Anda dapat memetakan beberapa set izin atau peran ke akun pengguna basis data yang sama. Misalnya, anggaplah kebijakan IAM Anda telah menentukan ARN sumber daya berikut.
>
> ```
> arn:aws:rds-db:us-east-2:123456789012:dbuser:db-12ABC34DEFG5HIJ6KLMNOP78QR/
> jane_doe
> ```
>
> Jika Anda melampirkan kebijakan ke Jane, Bob, dan Diego, maka masing-masing pengguna tersebut dapat terhubung ke klaster DB yang telah ditentukan menggunakan akun basis data `jane_doe`.

Membuat akun basis data menggunakan autentikasi IAM

Dengan autentikasi basis data IAM, Anda tidak perlu menetapkan kata sandi basis data ke akun pengguna yang Anda buat. Jika Anda menghapus pengguna yang dipetakan ke akun basis data, Anda juga harus menghapus akun basis data dengan pernyataan `DROP USER`.

> **Note**
>
> Nama pengguna yang digunakan untuk autentikasi IAM harus sesuai dengan huruf besar/kecil nama pengguna dalam basis data.

Topik

- Menggunakan autentikasi IAM dengan MariaDB dan MySQL
- Menggunakan autentikasi IAM dengan PostgreSQL

Menggunakan autentikasi IAM dengan MariaDB dan MySQL

Dengan MariaDB dan MySQL, autentikasi ditangani oleh `AWSAuthenticationPlugin`—plugin yang disediakan AWS yang berfungsi secara lancar dengan IAM untuk mengautentikasi pengguna Anda. Hubungkan ke klaster DB sebagai pengguna master atau pengguna lain yang dapat membuat pengguna dan memberikan hak akses. Setelah terhubung, berikan pernyataan `CREATE USER`, seperti yang ditunjukkan pada contoh berikut.

```
CREATE USER jane_doe IDENTIFIED WITH AWSAuthenticationPlugin AS 'RDS';
```

Klausa `IDENTIFIED WITH` memungkinkan MariaDB dan MySQL menggunakan `AWSAuthenticationPlugin` untuk mengautentikasi akun basis data (`jane_doe`). Klausa `AS 'RDS'` mengacu pada metode autentikasi. Pastikan nama pengguna basis data yang ditentukan sama dengan sumber daya dalam kebijakan IAM untuk akses basis data IAM. Untuk informasi selengkapnya, lihat Membuat dan menggunakan kebijakan IAM untuk akses basis data IAM.

> **Note**
>
> Jika Anda melihat pesan berikut, artinya plugin yang disediakan AWS tidak tersedia untuk instans DB saat ini.
> `ERROR 1524 (HY000): Plugin 'AWSAuthenticationPlugin' is not loaded`
> Untuk mengatasi kesalahan ini, verifikasikan bahwa Anda menggunakan konfigurasi yang didukung dan bahwa Anda telah mengaktifkan autentikasi basis data IAM di instans DB Anda. Untuk informasi selengkapnya, lihat Ketersediaan Wilayah dan versi dan Mengaktifkan dan menonaktifkan autentikasi basis data IAM.

Setelah membuat akun menggunakan `AWSAuthenticationPlugin`, Anda mengelolanya dengan cara yang sama seperti akun basis data lainnya. Misalnya, Anda dapat memodifikasi hak akses akun dengan pernyataan `GRANT` dan `REVOKE`, atau memodifikasi berbagai atribut akun dengan pernyataan `ALTER USER`.

Lalu lintas jaringan basis data dienkripsi menggunakan SSL/TLS saat menggunakan IAM. Untuk mengizinkan koneksi SSL, ubah akun pengguna dengan perintah berikut.

```
ALTER USER 'jane_doe'@'%' REQUIRE SSL;
```

Menggunakan autentikasi IAM dengan PostgreSQL

Untuk menggunakan autentikasi IAM dengan PostgreSQL, hubungkan ke instans DB sebagai pengguna master atau pengguna lain yang dapat membuat pengguna dan memberikan hak istimewa. Setelah terhubung, buat pengguna basis data lalu berikan peran `rds_iam` kepada pengguna tersebut seperti yang ditunjukkan pada contoh berikut.

```
CREATE USER db_userx;
GRANT rds_iam TO db_userx;
```

Pastikan nama pengguna basis data yang ditentukan sama dengan sumber daya dalam kebijakan IAM untuk akses basis data IAM. Untuk informasi selengkapnya, lihat Membuat dan menggunakan kebijakan IAM untuk akses basis data IAM.

Menghubungkan ke instans DB menggunakan autentikasi IAM

Dengan autentikasi basis data IAM, Anda menggunakan token autentikasi ketika Anda terhubung ke instans DB Anda. Token autentikasi adalah string karakter yang Anda gunakan sebagai pengganti kata sandi. Setelah Anda membuat token autentikasi, token tersebut berlaku selama 15 menit sebelum kedaluwarsa. Jika Anda mencoba terhubung menggunakan token yang kedaluwarsa, permintaan koneksi ditolak.

Setiap token autentikasi harus disertai dengan tanda tangan yang valid, menggunakan AWS Signature versi 4. (Untuk informasi selengkapnya, lihat Proses penandatanganan Signature Versi 4 dalam Referensi Umum AWS.) AWS CLI dan SDK AWS, seperti AWS SDK for Java atau AWS SDK for Python (Boto3), dapat secara otomatis menandatangani setiap token yang Anda buat.

Anda dapat menggunakan token autentikasi saat menghubungkan ke Amazon RDS dari layanan AWS lainnya, seperti AWS Lambda. Dengan menggunakan token, Anda tidak perlu menempatkan

kata sandi dalam kode Anda. Alternatifnya, Anda dapat menggunakan SDK AWS untuk membuat dan menandatangani token autentikasi secara programatis.

Setelah Anda memiliki token autentikasi IAM yang telah ditandatangani, Anda dapat terhubung ke instans DB Amazon RDS. Setelah itu, Anda dapat menemukan cara melakukannya menggunakan alat baris perintah atau SDK AWS, seperti AWS SDK for Java atau AWS SDK for Python (Boto3).

Untuk informasi selengkapnya, lihat postingan blog berikut ini:

- Gunakan autentikasi IAM untuk terhubung dengan SQL Workbench/J ke Aurora MySQL atau Amazon RDS for MySQL
- Menggunakan autentikasi IAM untuk terhubung dengan pgAdmin Amazon Aurora PostgreSQL atau Amazon RDS for PostgreSQL

Prasyarat

Berikut adalah prasyarat untuk menghubungkan ke instans DB menggunakan autentikasi IAM:

- Mengaktifkan dan menonaktifkan autentikasi basis data IAM
- Membuat dan menggunakan kebijakan IAM untuk akses basis data IAM
- Membuat akun basis data menggunakan autentikasi IAM

Topik

- Menghubungkan ke instans DB Anda menggunakan autentikasi IAM dari baris perintah: AWS CLI dan klien mysql
- Menghubungkan ke instans DB Anda menggunakan autentikasi IAM dari baris perintah: AWS CLI dan klien psql
- Menghubungkan ke instans DB Anda menggunakan autentikasi IAM dan AWS SDK for .NET
- Menghubungkan ke instans DB Anda menggunakan autentikasi IAM dan AWS SDK for Go
- Menghubungkan ke instans DB Anda menggunakan autentikasi IAM dan AWS SDK for Java
- Menghubungkan ke instans DB Anda menggunakan autentikasi IAM dan AWS SDK for Python (Boto3)

Menghubungkan ke instans DB Anda menggunakan autentikasi IAM dari baris perintah: AWS CLI dan klien mysql

Anda dapat terhubung dari baris perintah ke instans DB Amazon RDS dengan alat baris perintah AWS CLI dan `mysql` seperti yang dijelaskan berikut ini.

Prasyarat

Berikut adalah prasyarat untuk menghubungkan ke instans DB menggunakan autentikasi IAM:

- Mengaktifkan dan menonaktifkan autentikasi basis data IAM
- Membuat dan menggunakan kebijakan IAM untuk akses basis data IAM
- Membuat akun basis data menggunakan autentikasi IAM

> **Note**
>
> Untuk informasi tentang cara menghubungkan ke basis data menggunakan SQL Workbench/J dengan autentikasi IAM, lihat postingan blog Menggunakan autentikasi IAM untuk terhubung dengan SQL Workbench/J ke Aurora MySQL atau Amazon RDS for MySQL.

Topik

- Membuat token autentikasi IAM
- Menghubungkan ke instans DB

Membuat token autentikasi IAM

Contoh berikut menunjukkan cara mendapatkan token autentikasi yang ditandatangani menggunakan AWS CLI.

```
aws rds generate-db-auth-token \
   --hostname rdsmysql.123456789012.us-west-2.rds.amazonaws.com \
   --port 3306 \
   --region us-west-2 \
   --username jane_doe
```

Dalam contoh, parameternya adalah sebagai berikut:

- `--hostname` – Nama host instans DB yang ingin Anda akses
- `--port` – Nomor port yang digunakan untuk menghubungkan ke instans DB Anda
- `--region` – Wilayah AWS tempat instans DB berjalan
- `--username` – Akun basis data yang ingin Anda akses

Beberapa karakter pertama dari token terlihat seperti berikut.

```
rdsmysql.123456789012.us-west-2.rds.amazonaws.com:3306/?
Action=connect&DBUser=jane_doe&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Expires=900...
```

> **Note**
>
> Anda tidak dapat menggunakan catatan DNS Route 53 kustom sebagai pengganti titik akhir instans DB untuk menghasilkan token autentikasi.

Menghubungkan ke instans DB

Format umum untuk terhubung ditampilkan sebagai berikut.

```
mysql --host=hostName --port=portNumber --ssl-ca=full_path_to_ssl_certificate --enable-
cleartext-plugin --user=userName --password=authToken
```

Parameternya adalah sebagai berikut:

- `--host` – Nama host instans DB yang ingin Anda akses
- `--port` – Nomor port yang digunakan untuk menghubungkan ke instans DB Anda
- `--ssl-ca` – Jalur lengkap ke file sertifikat SSL yang berisi kunci publik

  Untuk informasi selengkapnya tentang dukungan SSL/TLS untuk MariaDB, lihat Menggunakan SSL/TLS dengan instans basis data MariaDB.

  Untuk informasi selengkapnya tentang dukungan SSL/TLS untuk MySQL, lihat Menggunakan SSL/TLS dengan instans DB MySQL.

  Untuk mengunduh sertifikat SSL, lihat .
- `--enable-cleartext-plugin` – Nilai yang menentukan bahwa `AWSAuthenticationPlugin` harus digunakan untuk koneksi ini

Jika Anda menggunakan klien MariaDB, opsi `--enable-cleartext-plugin` tidak diperlukan.

- `--user` – Akun basis data yang ingin Anda akses
- `--password` – Token autentikasi IAM yang ditandatangani

Token autentikasi terdiri atas beberapa ratus karakter. Hal ini dapat sulit ditangani di baris perintah. Salah satu cara untuk mengatasinya adalah dengan menyimpan token ke variabel lingkungan, lalu menggunakan variabel tersebut saat Anda terhubung. Contoh berikut menunjukkan satu cara untuk melakukan solusi ini. Dalam contoh ini, *`/sample_dir/`* adalah jalur lengkap ke file sertifikat SSL yang berisi kunci publik.

```
RDSHOST="mysqldb.123456789012.us-east-1.rds.amazonaws.com"
TOKEN="$(aws rds generate-db-auth-token --hostname $RDSHOST --port 3306 --region us-
west-2 --username jane_doe )"

mysql --host=$RDSHOST --port=3306 --ssl-ca=/sample_dir/global-bundle.pem --enable-
cleartext-plugin --user=jane_doe --password=$TOKEN
```

Saat Anda terhubung menggunakan `AWSAuthenticationPlugin`, koneksi diamankan menggunakan SSL. Untuk memverifikasi hal ini, ketik berikut ini di prompt perintah `mysql>`.

```
show status like 'Ssl%';
```

Baris berikut dalam output menampilkan lebih banyak detail.

```
+---------------+-------------+
| Variable_name | Value

                                                                    |
+---------------+-------------+
| ...           | ...
| Ssl_cipher    | AES256-SHA

                                                                    |
| ...           | ...
| Ssl_version   | TLSv1.1

                                                                    |
| ...           | ...
```

```
+----------------------------+
```

Jika Anda ingin terhubung ke instans DB melalui proksi, lihat Terhubung ke sebuah proksi menggunakan autentikasi IAM.

Menghubungkan ke instans DB Anda menggunakan autentikasi IAM dari baris perintah: AWS CLI dan klien psql

Anda dapat terhubung dari baris perintah ke instans DB Amazon RDS for PostgreSQL dengan AWS CLI dan alat baris perintah psql seperti yang dijelaskan berikut.

Prasyarat

Berikut adalah prasyarat untuk menghubungkan ke instans DB menggunakan autentikasi IAM:

- Mengaktifkan dan menonaktifkan autentikasi basis data IAM
- Membuat dan menggunakan kebijakan IAM untuk akses basis data IAM
- Membuat akun basis data menggunakan autentikasi IAM

> **Note**
>
> Untuk informasi tentang menghubungkan ke basis data Anda menggunakan pgAdmin dengan autentikasi IAM, lihat postingan blog Menggunakan autentikasi IAM untuk terhubung dengan pgAdmin Amazon Aurora PostgreSQL atau Amazon RDS for PostgreSQL.

Topik

- Membuat token autentikasi IAM
- Menghubungkan ke instans Amazon RDS PostgreSQL

Membuat token autentikasi IAM

Token autentikasi terdiri dari ratusan karakter sehingga kemungkinan menjadi sulit ditangani di baris perintah. Salah satu cara untuk mengatasinya adalah dengan menyimpan token ke variabel lingkungan, lalu menggunakan variabel tersebut saat Anda terhubung. Contoh berikut menunjukkan cara menggunakan AWS CLI untuk mendapatkan token autentikasi yang ditandatangani menggunakan perintah `generate-db-auth-token` dan menyimpannya di variabel lingkungan `PGPASSWORD`.

```
export RDSHOST="rdspostgres.123456789012.us-west-2.rds.amazonaws.com"
export PGPASSWORD="$(aws rds generate-db-auth-token --hostname $RDSHOST --port 5432 --
region us-west-2 --username jane_doe )"
```

Dalam contoh, parameter untuk perintah `generate-db-auth-token` adalah sebagai berikut:

- `--hostname` – Nama host instans DB yang ingin Anda akses
- `--port` – Nomor port yang digunakan untuk menghubungkan ke instans DB Anda
- `--region` – Wilayah AWS tempat instans DB berjalan
- `--username` – Akun basis data yang ingin Anda akses

Beberapa karakter pertama dari token yang dihasilkan terlihat seperti berikut.

```
rdspostgres.123456789012.us-west-2.rds.amazonaws.com:5432/?
Action=connect&DBUser=jane_doe&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Expires=900...
```

> **Note**
>
> Anda tidak dapat menggunakan catatan DNS Route 53 kustom sebagai pengganti titik akhir instans DB untuk menghasilkan token autentikasi.

Menghubungkan ke instans Amazon RDS PostgreSQL

Format umum untuk menggunakan psql untuk terhubung ditampilkan sebagai berikut.

```
psql "host=hostName port=portNumber sslmode=verify-full
 sslrootcert=full_path_to_ssl_certificate dbname=DBName user=userName
 password=authToken"
```

Parameternya adalah sebagai berikut:

- `host` – Nama host instans DB yang ingin Anda akses
- `port` – Nomor port yang digunakan untuk menghubungkan ke instans DB Anda
- `sslmode` – Mode SSL yang akan digunakan

Saat Anda menggunakan `sslmode=verify-full`, koneksi SSL memverifikasi titik akhir instans DB di sertifikat SSL.

- `sslrootcert` – Jalur lengkap ke file sertifikat SSL yang berisi kunci publik

  Untuk informasi selengkapnya, lihat Menggunakan SSL dengan instans DB PostgreSQL.

  Untuk mengunduh sertifikat SSL, lihat .
- `dbname` – Basis data yang ingin Anda akses
- `user` – Akun basis data yang ingin Anda akses
- `password` – Token autentikasi IAM yang ditandatangani

> **Note**
>
> Anda tidak dapat menggunakan catatan DNS Route 53 kustom sebagai pengganti titik akhir instans DB untuk menghasilkan token autentikasi.

Contoh berikut menunjukkan penggunaan psql untuk terhubung. Dalam contoh, psql menggunakan variabel lingkungan `RDSHOST` untuk host dan variabel lingkungan `PGPASSWORD` untuk token yang dihasilkan. Selain itu, *`/sample_dir/`* adalah jalur lengkap ke file sertifikat SSL yang berisi kunci publik.

```
export RDSHOST="rdspostgres.123456789012.us-west-2.rds.amazonaws.com"
export PGPASSWORD="$(aws rds generate-db-auth-token --hostname $RDSHOST --port 5432 --
region us-west-2 --username jane_doe )"

psql "host=$RDSHOST port=5432 sslmode=verify-full sslrootcert=/sample_dir/global-
bundle.pem dbname=DBName user=jane_doe password=$PGPASSWORD"
```

Jika Anda ingin terhubung ke instans DB melalui proksi, lihat Terhubung ke sebuah proksi menggunakan autentikasi IAM.

Menghubungkan ke instans DB Anda menggunakan autentikasi IAM dan AWS SDK for .NET

Anda dapat menghubungkan ke instans DB RDS for MariaDB, MySQL, atau PostgreSQL dengan AWS SDK for .NET seperti yang dijelaskan berikut ini.

Prasyarat

Berikut adalah prasyarat untuk menghubungkan ke instans DB menggunakan autentikasi IAM:

- Mengaktifkan dan menonaktifkan autentikasi basis data IAM
- Membuat dan menggunakan kebijakan IAM untuk akses basis data IAM
- Membuat akun basis data menggunakan autentikasi IAM

Contoh

Contoh kode berikut ini menunjukkan cara membuat token autentikasi, lalu menggunakannya untuk terhubung ke instans DB.

Untuk menjalankan contoh kode ini, Anda memerlukan AWS SDK for .NET, yang ada di situs AWS. Paket `AWSSDK.CORE` dan `AWSSDK.RDS` diperlukan. Untuk terhubung ke instans DB, gunakan konektor basis data .NET untuk mesin DB, seperti MySqlConnector for MariaDB atau MySQL, atau Npgsql for PostgreSQL.

Kode ini terhubung ke instans DB MariaDB MySQL. Ubah nilai variabel berikut sesuai kebutuhan:

- `server` – Titik akhir instans DB yang ingin Anda akses
- `user` – Akun basis data yang ingin Anda akses
- `database` – Basis data yang ingin Anda akses
- `port` – Nomor port yang digunakan untuk menghubungkan ke instans DB Anda
- `SslMode` – Mode SSL yang akan digunakan

  Saat Anda menggunakan `SslMode=Required`, koneksi SSL memverifikasi titik akhir instans DB di sertifikat SSL.
- `SslCa` – Jalur lengkap ke sertifikat SSL untuk Amazon RDS

  Untuk mengunduh sertifikat, lihat .

> **Note**
>
> Anda tidak dapat menggunakan catatan DNS Route 53 kustom sebagai pengganti titik akhir instans DB untuk menghasilkan token autentikasi.

```
using System;
```

```
using System.Data;
using MySql.Data;
using MySql.Data.MySqlClient;
using Amazon;

namespace ubuntu
{
  class Program
  {
    static void Main(string[] args)
    {
      var pwd =
 Amazon.RDS.Util.RDSAuthTokenGenerator.GenerateAuthToken(RegionEndpoint.USEast1,
 "mysqldb.123456789012.us-east-1.rds.amazonaws.com", 3306, "jane_doe");
      // for debug only Console.Write("{0}\n", pwd);  //this verifies the token is
 generated

      MySqlConnection conn = new MySqlConnection($"server=mysqldb.123456789012.us-
east-1.rds.amazonaws.com;user=jane_doe;database=mydB;port=3306;password={pwd};SslMode=Required;
      conn.Open();

      // Define a query
      MySqlCommand sampleCommand = new MySqlCommand("SHOW DATABASES;", conn);

      // Execute a query
      MySqlDataReader mysqlDataRdr = sampleCommand.ExecuteReader();

      // Read all rows and output the first column in each row
      while (mysqlDataRdr.Read())
        Console.WriteLine(mysqlDataRdr[0]);

      mysqlDataRdr.Close();
      // Close connection
      conn.Close();
    }
  }
}
```

Kode ini terhubung ke instans DB PostgreSQL.

Ubah nilai variabel berikut sesuai kebutuhan:

- `Server` – Titik akhir instans DB yang ingin Anda akses

- `User ID` – Akun basis data yang ingin Anda akses
- `Database` – Basis data yang ingin Anda akses
- `Port` – Nomor port yang digunakan untuk menghubungkan ke instans DB Anda
- `SSL Mode` – Mode SSL yang akan digunakan

  Saat Anda menggunakan `SSL Mode=Required`, koneksi SSL memverifikasi titik akhir instans DB di sertifikat SSL.
- `Root Certificate` – Jalur lengkap ke sertifikat SSL untuk Amazon RDS

  Untuk mengunduh sertifikat, lihat .

> **Note**
>
> Anda tidak dapat menggunakan catatan DNS Route 53 kustom sebagai pengganti titik akhir instans DB untuk menghasilkan token autentikasi.

```
using System;
using Npgsql;
using Amazon.RDS.Util;

namespace ConsoleApp1
{
    class Program
    {
        static void Main(string[] args)
        {
            var pwd =
 RDSAuthTokenGenerator.GenerateAuthToken("postgresmydb.123456789012.us-
east-1.rds.amazonaws.com", 5432, "jane_doe");
// for debug only Console.Write("{0}\n", pwd);  //this verifies the token is generated

            NpgsqlConnection conn = new
 NpgsqlConnection($"Server=postgresmydb.123456789012.us-east-1.rds.amazonaws.com;User
 Id=jane_doe;Password={pwd};Database=mydb;SSL Mode=Require;Root
 Certificate=full_path_to_ssl_certificate");
            conn.Open();

            // Define a query
```

```
                    NpgsqlCommand cmd = new NpgsqlCommand("select count(*) FROM
 pg_user", conn);

            // Execute a query
            NpgsqlDataReader dr = cmd.ExecuteReader();

            // Read all rows and output the first column in each row
            while (dr.Read())
                Console.Write("{0}\n", dr[0]);

            // Close connection
            conn.Close();
        }
    }
}
```

Jika Anda ingin terhubung ke instans DB melalui proksi, lihat Terhubung ke sebuah proksi menggunakan autentikasi IAM.

Menghubungkan ke instans DB Anda menggunakan autentikasi IAM dan AWS SDK for Go

Anda dapat menghubungkan ke instans DB RDS for MariaDB, MySQL, atau PostgreSQL dengan AWS SDK for Go seperti yang dijelaskan berikut ini.

Prasyarat

Berikut adalah prasyarat untuk menghubungkan ke instans DB menggunakan autentikasi IAM:

- Mengaktifkan dan menonaktifkan autentikasi basis data IAM
- Membuat dan menggunakan kebijakan IAM untuk akses basis data IAM
- Membuat akun basis data menggunakan autentikasi IAM

Contoh

Untuk menjalankan contoh kode ini, Anda memerlukan AWS SDK for Go, yang ada di situs AWS.

Ubah nilai variabel berikut sesuai kebutuhan:

- `dbName` – Basis data yang ingin Anda akses
- `dbUser` – Akun basis data yang ingin Anda akses
- `dbHost` – Titik akhir instans DB yang ingin Anda akses

> **Note**
>
> Anda tidak dapat menggunakan catatan DNS Route 53 kustom sebagai pengganti titik akhir instans DB untuk menghasilkan token autentikasi.

- `dbPort` – Nomor port yang digunakan untuk menghubungkan ke instans DB Anda
- `region` – Wilayah AWS tempat instans DB berjalan

Selain itu, pastikan pustaka yang diimpor dalam kode sampel ada di sistem Anda.

> **Important**
>
> Contoh dalam bagian ini menggunakan kode berikut untuk menyediakan kredensial yang mengakses basis data dari lingkungan lokal:
> `creds := credentials.NewEnvCredentials()`
> Jika Anda mengakses basis data dari layanan AWS, seperti Amazon EC2 atau Amazon ECS, Anda dapat mengganti kode dengan kode berikut:
> `sess := session.Must(session.NewSession())`
> `creds := sess.Config.Credentials`
> Jika Anda membuat perubahan ini, pastikan Anda menambahkan impor berikut:
> `"github.com/aws/aws-sdk-go/aws/session"`

Topik

- Menghubungkan menggunakan autentikasi IAM dan AWS SDK for Go V2
- Menghubungkan menggunakan autentikasi IAM dan AWS SDK for Go V1.

Menghubungkan menggunakan autentikasi IAM dan AWS SDK for Go V2

Anda dapat terhubung ke instans DB menggunakan autentikasi IAM dan AWS SDK for Go V2.

Contoh kode berikut ini menunjukkan cara membuat token autentikasi, lalu menggunakannya untuk terhubung ke instans DB.

Kode ini terhubung ke instans DB MariaDB MySQL.

```
package main
```

```
import (
     "context"
     "database/sql"
     "fmt"

     "github.com/aws/aws-sdk-go-v2/config"
     "github.com/aws/aws-sdk-go-v2/feature/rds/auth"
     _ "github.com/go-sql-driver/mysql"
)

func main() {

     var dbName string = "DatabaseName"
     var dbUser string = "DatabaseUser"
     var dbHost string = "mysqldb.123456789012.us-east-1.rds.amazonaws.com"
     var dbPort int = 3306
     var dbEndpoint string = fmt.Sprintf("%s:%d", dbHost, dbPort)
     var region string = "us-east-1"

    cfg, err := config.LoadDefaultConfig(context.TODO())
    if err != nil {
     panic("configuration error: " + err.Error())
    }

    authenticationToken, err := auth.BuildAuthToken(
     context.TODO(), dbEndpoint, region, dbUser, cfg.Credentials)
    if err != nil {
     panic("failed to create authentication token: " + err.Error())
    }

    dsn := fmt.Sprintf("%s:%s@tcp(%s)/%s?tls=true&allowCleartextPasswords=true",
        dbUser, authenticationToken, dbEndpoint, dbName,
    )

    db, err := sql.Open("mysql", dsn)
    if err != nil {
        panic(err)
    }

    err = db.Ping()
    if err != nil {
        panic(err)
    }
```

```
}
```

Kode ini terhubung ke instans DB PostgreSQL.

```
package main

import (
     "context"
     "database/sql"
     "fmt"

     "github.com/aws/aws-sdk-go-v2/config"
     "github.com/aws/aws-sdk-go-v2/feature/rds/auth"
     _ "github.com/lib/pq"
)

func main() {

     var dbName string = "DatabaseName"
     var dbUser string = "DatabaseUser"
     var dbHost string = "postgresmydb.123456789012.us-east-1.rds.amazonaws.com"
     var dbPort int = 5432
     var dbEndpoint string = fmt.Sprintf("%s:%d", dbHost, dbPort)
     var region string = "us-east-1"

    cfg, err := config.LoadDefaultConfig(context.TODO())
    if err != nil {
     panic("configuration error: " + err.Error())
    }

    authenticationToken, err := auth.BuildAuthToken(
     context.TODO(), dbEndpoint, region, dbUser, cfg.Credentials)
    if err != nil {
     panic("failed to create authentication token: " + err.Error())
    }

    dsn := fmt.Sprintf("host=%s port=%d user=%s password=%s dbname=%s",
        dbHost, dbPort, dbUser, authenticationToken, dbName,
    )

    db, err := sql.Open("postgres", dsn)
    if err != nil {
        panic(err)
```

```
    }

    err = db.Ping()
    if err != nil {
        panic(err)
    }
}
```

Jika Anda ingin terhubung ke instans DB melalui proksi, lihat Terhubung ke sebuah proksi menggunakan autentikasi IAM.

Menghubungkan menggunakan autentikasi IAM dan AWS SDK for Go V1.

Anda dapat terhubung ke instans DB menggunakan autentikasi IAM dan AWS SDK for Go V1

Contoh kode berikut ini menunjukkan cara membuat token autentikasi, lalu menggunakannya untuk terhubung ke instans DB.

Kode ini terhubung ke instans DB MariaDB MySQL.

```
package main

import (
    "database/sql"
    "fmt"
    "log"

    "github.com/aws/aws-sdk-go/aws/credentials"
    "github.com/aws/aws-sdk-go/service/rds/rdsutils"
    _ "github.com/go-sql-driver/mysql"
)

func main() {
    dbName := "app"
    dbUser := "jane_doe"
    dbHost := "mysqldb.123456789012.us-east-1.rds.amazonaws.com"
    dbPort := 3306
    dbEndpoint := fmt.Sprintf("%s:%d", dbHost, dbPort)
    region := "us-east-1"

    creds := credentials.NewEnvCredentials()
    authToken, err := rdsutils.BuildAuthToken(dbEndpoint, region, dbUser, creds)
    if err != nil {
```

```
        panic(err)
    }

    dsn := fmt.Sprintf("%s:%s@tcp(%s)/%s?tls=true&allowCleartextPasswords=true",
        dbUser, authToken, dbEndpoint, dbName,
    )

    db, err := sql.Open("mysql", dsn)
    if err != nil {
        panic(err)
    }

    err = db.Ping()
    if err != nil {
        panic(err)
    }
}
```

Kode ini terhubung ke instans DB PostgreSQL.

```
package main

import (
 "database/sql"
 "fmt"

 "github.com/aws/aws-sdk-go/aws/credentials"
 "github.com/aws/aws-sdk-go/service/rds/rdsutils"
 _ "github.com/lib/pq"
)

func main() {
    dbName := "app"
    dbUser := "jane_doe"
    dbHost := "postgresmydb.123456789012.us-east-1.rds.amazonaws.com"
    dbPort := 5432
    dbEndpoint := fmt.Sprintf("%s:%d", dbHost, dbPort)
    region := "us-east-1"

    creds := credentials.NewEnvCredentials()
    authToken, err := rdsutils.BuildAuthToken(dbEndpoint, region, dbUser, creds)
    if err != nil {
        panic(err)
```

```
    }

    dsn := fmt.Sprintf("host=%s port=%d user=%s password=%s dbname=%s",
        dbHost, dbPort, dbUser, authToken, dbName,
    )

    db, err := sql.Open("postgres", dsn)
    if err != nil {
        panic(err)
    }

    err = db.Ping()
    if err != nil {
        panic(err)
    }
}
```

Jika Anda ingin terhubung ke instans DB melalui proksi, lihat Terhubung ke sebuah proksi menggunakan autentikasi IAM.

Menghubungkan ke instans DB Anda menggunakan autentikasi IAM dan AWS SDK for Java

Anda dapat menghubungkan ke instans DB RDS for MariaDB, MySQL, atau PostgreSQL dengan AWS SDK for Java seperti yang dijelaskan berikut ini.

Prasyarat

Berikut adalah prasyarat untuk menghubungkan ke instans DB menggunakan autentikasi IAM:

- Mengaktifkan dan menonaktifkan autentikasi basis data IAM
- Membuat dan menggunakan kebijakan IAM untuk akses basis data IAM
- Membuat akun basis data menggunakan autentikasi IAM
- Siapkan SDK AWS untuk Java

Topik

- Membuat token autentikasi IAM
- Membuat token autentikasi IAM secara manual
- Menghubungkan ke instans DB

Membuat token autentikasi IAM

Jika Anda menulis program menggunakan AWS SDK for Java, Anda dapat memperoleh token autentikasi yang ditandatangani dengan menggunakan kelas `RdsIamAuthTokenGenerator`. Penggunaan kelas ini mengharuskan Anda untuk memberikan kredensial AWS. Untuk melakukannya, Anda membuat instans kelas `DefaultAWSCredentialsProviderChain`. `DefaultAWSCredentialsProviderChain` menggunakan kunci akses AWS pertama dan kunci rahasia yang ditemukan di rantai penyedia kredensial default. Untuk informasi selengkapnya tentang kunci akses AWS, lihat Mengelola kunci akses untuk pengguna.

> **Note**
>
> Anda tidak dapat menggunakan catatan DNS Route 53 kustom sebagai pengganti titik akhir instans DB untuk menghasilkan token autentikasi.

Setelah membuat instans `RdsIamAuthTokenGenerator`, Anda dapat memanggil metode `getAuthToken` untuk mendapatkan token yang ditandatangani. Berikan Wilayah AWS, nama host, nomor port, dan nama pengguna. Contoh kode berikut menunjukkan cara melakukannya.

```
package com.amazonaws.codesamples;

import com.amazonaws.auth.DefaultAWSCredentialsProviderChain;
import com.amazonaws.services.rds.auth.GetIamAuthTokenRequest;
import com.amazonaws.services.rds.auth.RdsIamAuthTokenGenerator;

public class GenerateRDSAuthToken {

    public static void main(String[] args) {

     String region = "us-west-2";
     String hostname = "rdsmysql.123456789012.us-west-2.rds.amazonaws.com";
     String port = "3306";
     String username = "jane_doe";

     System.out.println(generateAuthToken(region, hostname, port, username));
    }

    static String generateAuthToken(String region, String hostName, String port, String
 username) {
```

```
      RdsIamAuthTokenGenerator generator = RdsIamAuthTokenGenerator.builder()
       .credentials(new DefaultAWSCredentialsProviderChain())
       .region(region)
       .build();

      String authToken = generator.getAuthToken(
       GetIamAuthTokenRequest.builder()
       .hostname(hostName)
       .port(Integer.parseInt(port))
       .userName(username)
       .build());

      return authToken;
     }

}
```

Membuat token autentikasi IAM secara manual

Di Java, cara termudah untuk menghasilkan token autentikasi adalah dengan menggunakan `RdsIamAuthTokenGenerator`. Kelas ini membuat token autentikasi untuk Anda, lalu menandatanganinya menggunakan AWS Signature versi 4. Untuk informasi selengkapnya, lihat Proses penandatanganan Signature versi 4 dalam Referensi Umum AWS.

Namun, Anda juga dapat membuat dan menandatangani token autentikasi secara manual, seperti ditunjukkan dalam contoh kode berikut.

```
package com.amazonaws.codesamples;

import com.amazonaws.SdkClientException;
import com.amazonaws.auth.DefaultAWSCredentialsProviderChain;
import com.amazonaws.auth.SigningAlgorithm;
import com.amazonaws.util.BinaryUtils;
import org.apache.commons.lang3.StringUtils;

import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import java.nio.charset.Charset;
import java.security.MessageDigest;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.SortedMap;
import java.util.TreeMap;
```

```
import static com.amazonaws.auth.internal.SignerConstants.AWS4_TERMINATOR;
import static com.amazonaws.util.StringUtils.UTF8;

public class CreateRDSAuthTokenManually {
    public static String httpMethod = "GET";
    public static String action = "connect";
    public static String canonicalURIParameter = "/";
    public static SortedMap<String, String> canonicalQueryParameters = new TreeMap();
    public static String payload = StringUtils.EMPTY;
    public static String signedHeader = "host";
    public static String algorithm = "AWS4-HMAC-SHA256";
    public static String serviceName = "rds-db";
    public static String requestWithoutSignature;

    public static void main(String[] args) throws Exception {

        String region = "us-west-2";
        String instanceName = "rdsmysql.123456789012.us-west-2.rds.amazonaws.com";
        String port = "3306";
        String username = "jane_doe";

        Date now = new Date();
        String date = new SimpleDateFormat("yyyyMMdd").format(now);
        String dateTimeStamp = new
 SimpleDateFormat("yyyyMMdd'T'HHmmss'Z'").format(now);
        DefaultAWSCredentialsProviderChain creds = new
 DefaultAWSCredentialsProviderChain();
     String awsAccessKey = creds.getCredentials().getAWSAccessKeyId();
     String awsSecretKey = creds.getCredentials().getAWSSecretKey();
        String expiryMinutes = "900";

        System.out.println("Step 1:  Create a canonical request:");
        String canonicalString = createCanonicalString(username, awsAccessKey, date,
 dateTimeStamp, region, expiryMinutes, instanceName, port);
        System.out.println(canonicalString);
        System.out.println();

        System.out.println("Step 2:  Create a string to sign:");
        String stringToSign = createStringToSign(dateTimeStamp, canonicalString,
 awsAccessKey, date, region);
        System.out.println(stringToSign);
        System.out.println();
```

```
        System.out.println("Step 3:  Calculate the signature:");
        String signature = BinaryUtils.toHex(calculateSignature(stringToSign,
newSigningKey(awsSecretKey, date, region, serviceName)));
        System.out.println(signature);
        System.out.println();

        System.out.println("Step 4:  Add the signing info to the request");

        System.out.println(appendSignature(signature));
        System.out.println();

    }

    //Step 1: Create a canonical request date should be in format YYYYMMDD and dateTime
should be in format YYYYMMDDTHHMMSSZ
    public static String createCanonicalString(String user, String accessKey, String
date, String dateTime, String region, String expiryPeriod, String hostName, String
port) throws Exception {
        canonicalQueryParameters.put("Action", action);
        canonicalQueryParameters.put("DBUser", user);
        canonicalQueryParameters.put("X-Amz-Algorithm", "AWS4-HMAC-SHA256");
        canonicalQueryParameters.put("X-Amz-Credential", accessKey + "%2F" + date +
"%2F" + region + "%2F" + serviceName + "%2Faws4_request");
        canonicalQueryParameters.put("X-Amz-Date", dateTime);
        canonicalQueryParameters.put("X-Amz-Expires", expiryPeriod);
        canonicalQueryParameters.put("X-Amz-SignedHeaders", signedHeader);
        String canonicalQueryString = "";
        while(!canonicalQueryParameters.isEmpty()) {
            String currentQueryParameter = canonicalQueryParameters.firstKey();
            String currentQueryParameterValue =
canonicalQueryParameters.remove(currentQueryParameter);
            canonicalQueryString = canonicalQueryString + currentQueryParameter + "=" +
currentQueryParameterValue;
            if (!currentQueryParameter.equals("X-Amz-SignedHeaders")) {
                canonicalQueryString += "&";
            }
        }
        String canonicalHeaders = "host:" + hostName + ":" + port + '\n';
        requestWithoutSignature = hostName + ":" + port + "/?" + canonicalQueryString;

        String hashedPayload = BinaryUtils.toHex(hash(payload));
        return httpMethod + '\n' + canonicalURIParameter + '\n' + canonicalQueryString
+ '\n' + canonicalHeaders + '\n' + signedHeader + '\n' + hashedPayload;
```

```
    }

    //Step 2: Create a string to sign using sig v4
    public static String createStringToSign(String dateTime, String canonicalRequest,
 String accessKey, String date, String region) throws Exception {
        String credentialScope = date + "/" + region + "/" + serviceName + "/
aws4_request";
        return algorithm + '\n' + dateTime + '\n' + credentialScope + '\n' +
 BinaryUtils.toHex(hash(canonicalRequest));

    }

    //Step 3: Calculate signature
    /**
     * Step 3 of the &AWS; Signature version 4 calculation. It involves deriving
     * the signing key and computing the signature. Refer to
     * http://docs.aws.amazon
     * .com/general/latest/gr/sigv4-calculate-signature.html
     */
    public static byte[] calculateSignature(String stringToSign,
                                            byte[] signingKey) {
        return sign(stringToSign.getBytes(Charset.forName("UTF-8")), signingKey,
                SigningAlgorithm.HmacSHA256);
    }

    public static byte[] sign(byte[] data, byte[] key,
                          SigningAlgorithm algorithm) throws SdkClientException {
        try {
            Mac mac = algorithm.getMac();
            mac.init(new SecretKeySpec(key, algorithm.toString()));
            return mac.doFinal(data);
        } catch (Exception e) {
            throw new SdkClientException(
                    "Unable to calculate a request signature: "
                            + e.getMessage(), e);
        }
    }

    public static byte[] newSigningKey(String secretKey,
                                   String dateStamp, String regionName, String
 serviceName) {
        byte[] kSecret = ("AWS4" + secretKey).getBytes(Charset.forName("UTF-8"));
        byte[] kDate = sign(dateStamp, kSecret, SigningAlgorithm.HmacSHA256);
        byte[] kRegion = sign(regionName, kDate, SigningAlgorithm.HmacSHA256);
```

```
        byte[] kService = sign(serviceName, kRegion,
                SigningAlgorithm.HmacSHA256);
        return sign(AWS4_TERMINATOR, kService, SigningAlgorithm.HmacSHA256);
    }

    public static byte[] sign(String stringData, byte[] key,
                        SigningAlgorithm algorithm) throws SdkClientException {
        try {
            byte[] data = stringData.getBytes(UTF8);
            return sign(data, key, algorithm);
        } catch (Exception e) {
            throw new SdkClientException(
                    "Unable to calculate a request signature: "
                            + e.getMessage(), e);
        }
    }

    //Step 4: append the signature
    public static String appendSignature(String signature) {
        return requestWithoutSignature + "&X-Amz-Signature=" + signature;
    }

    public static byte[] hash(String s) throws Exception {
        try {
            MessageDigest md = MessageDigest.getInstance("SHA-256");
            md.update(s.getBytes(UTF8));
            return md.digest();
        } catch (Exception e) {
            throw new SdkClientException(
                    "Unable to compute hash while signing request: "
                            + e.getMessage(), e);
        }
    }
}
```

Menghubungkan ke instans DB

Contoh kode berikut menunjukkan cara membuat token autentikasi, lalu menggunakannya untuk menghubungkan ke instans yang menjalankan MariaDB atau MySQL.

Untuk menjalankan contoh kode ini, Anda memerlukan AWS SDK for Java, yang ada di situs AWS. Selain itu, Anda memerlukan hal berikut:

- MySQL Connector/J. Contoh kode ini diuji dengan `mysql-connector-java-5.1.33-bin.jar`.
- Sertifikat perantara untuk Amazon RDS yang khusus untuk sebuah Wilayah AWS. (Untuk informasi selengkapnya, lihat .) Saat runtime, pemuat kelas mencari sertifikat di direktori yang sama seperti contoh kode Java ini, sehingga pemuat kelas dapat menemukannya.
- Ubah nilai variabel berikut sesuai kebutuhan:
  - `RDS_INSTANCE_HOSTNAME` – Nama host instans DB yang ingin Anda akses.
  - `RDS_INSTANCE_PORT` – Nomor port yang digunakan untuk menghubungkan ke instans DB PostgreSQL Anda.
  - `REGION_NAME` – Wilayah AWS tempat instans DB berjalan.
  - `DB_USER` – Akun basis data yang ingin Anda akses.
  - `SSL_CERTIFICATE` – Sertifikat SSL untuk Amazon RDS yang khusus untuk sebuah Wilayah AWS.

    Untuk mengunduh sertifikat untuk Wilayah AWS Anda, lihat . Tempatkan sertifikat SSL di direktori yang sama dengan file program Java ini, sehingga pemuat kelas dapat menemukan sertifikat saat runtime.

Contoh kode ini memperoleh kredensial AWS dari rantai penyedia kredensial default.

> **Note**
>
> Tentukan kata sandi untuk `DEFAULT_KEY_STORE_PASSWORD` selain prompt yang ditampilkan di sini sebagai praktik terbaik keamanan.

```
package com.amazonaws.samples;

import com.amazonaws.services.rds.auth.RdsIamAuthTokenGenerator;
import com.amazonaws.services.rds.auth.GetIamAuthTokenRequest;
import com.amazonaws.auth.BasicAWSCredentials;
import com.amazonaws.auth.DefaultAWSCredentialsProviderChain;
import com.amazonaws.auth.AWSStaticCredentialsProvider;

import java.io.File;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.security.KeyStore;
```

```
import java.security.cert.CertificateFactory;
import java.security.cert.X509Certificate;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.Properties;

import java.net.URL;

public class IAMDatabaseAuthenticationTester {
    //&AWS; Credentials of the IAM user with policy enabling IAM Database Authenticated
 access to the db by the db user.
    private static final DefaultAWSCredentialsProviderChain creds = new
 DefaultAWSCredentialsProviderChain();
    private static final String AWS_ACCESS_KEY =
 creds.getCredentials().getAWSAccessKeyId();
    private static final String AWS_SECRET_KEY =
 creds.getCredentials().getAWSSecretKey();

    //Configuration parameters for the generation of the IAM Database Authentication
 token
    private static final String RDS_INSTANCE_HOSTNAME = "rdsmysql.123456789012.us-
west-2.rds.amazonaws.com";
    private static final int RDS_INSTANCE_PORT = 3306;
    private static final String REGION_NAME = "us-west-2";
    private static final String DB_USER = "jane_doe";
    private static final String JDBC_URL = "jdbc:mysql://" + RDS_INSTANCE_HOSTNAME +
 ":" + RDS_INSTANCE_PORT;

    private static final String SSL_CERTIFICATE = "rds-ca-2019-us-west-2.pem";

    private static final String KEY_STORE_TYPE = "JKS";
    private static final String KEY_STORE_PROVIDER = "SUN";
    private static final String KEY_STORE_FILE_PREFIX = "sys-connect-via-ssl-test-
cacerts";
    private static final String KEY_STORE_FILE_SUFFIX = ".jks";
    private static final String DEFAULT_KEY_STORE_PASSWORD = "changeit";

    public static void main(String[] args) throws Exception {
        //get the connection
        Connection connection = getDBConnectionUsingIam();
```

```
        //verify the connection is successful
        Statement stmt= connection.createStatement();
        ResultSet rs=stmt.executeQuery("SELECT 'Success!' FROM DUAL;");
        while (rs.next()) {
             String id = rs.getString(1);
            System.out.println(id); //Should print "Success!"
        }

        //close the connection
        stmt.close();
        connection.close();

        clearSslProperties();

    }

    /**
     * This method returns a connection to the db instance authenticated using IAM
Database Authentication
     * @return
     * @throws Exception
     */
    private static Connection getDBConnectionUsingIam() throws Exception {
        setSslProperties();
        return DriverManager.getConnection(JDBC_URL, setMySqlConnectionProperties());
    }

    /**
     * This method sets the mysql connection properties which includes the IAM Database
Authentication token
     * as the password. It also specifies that SSL verification is required.
     * @return
     */
    private static Properties setMySqlConnectionProperties() {
        Properties mysqlConnectionProperties = new Properties();
        mysqlConnectionProperties.setProperty("verifyServerCertificate","true");
        mysqlConnectionProperties.setProperty("useSSL", "true");
        mysqlConnectionProperties.setProperty("user",DB_USER);
        mysqlConnectionProperties.setProperty("password",generateAuthToken());
        return mysqlConnectionProperties;
    }

    /**
     * This method generates the IAM Auth Token.
```

```
     * An example IAM Auth Token would look like follows:
     * btusi123.cmz7kenwo2ye.rds.cn-north-1.amazonaws.com.cn:3306/?
Action=connect&DBUser=iamtestuser&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20171003T010726Z&X-Amz-SignedHeaders=host&X-Amz-Expires=899&X-Amz-
Credential=AKIAPFXHGVDI5RNFO4AQ%2F20171003%2Fcn-north-1%2Frds-db%2Faws4_request&X-Amz-
Signature=f9f45ef96c1f770cdad11a53e33ffa4c3730bc03fdee820cfdf1322eed15483b
     * @return
     */
    private static String generateAuthToken() {
        BasicAWSCredentials awsCredentials = new BasicAWSCredentials(AWS_ACCESS_KEY,
 AWS_SECRET_KEY);

        RdsIamAuthTokenGenerator generator = RdsIamAuthTokenGenerator.builder()
                .credentials(new
 AWSStaticCredentialsProvider(awsCredentials)).region(REGION_NAME).build();
        return generator.getAuthToken(GetIamAuthTokenRequest.builder()

 .hostname(RDS_INSTANCE_HOSTNAME).port(RDS_INSTANCE_PORT).userName(DB_USER).build());
    }

    /**
     * This method sets the SSL properties which specify the key store file, its type
 and password:
     * @throws Exception
     */
    private static void setSslProperties() throws Exception {
        System.setProperty("javax.net.ssl.trustStore", createKeyStoreFile());
        System.setProperty("javax.net.ssl.trustStoreType", KEY_STORE_TYPE);
        System.setProperty("javax.net.ssl.trustStorePassword",
 DEFAULT_KEY_STORE_PASSWORD);
    }

    /**
     * This method returns the path of the Key Store File needed for the SSL
 verification during the IAM Database Authentication to
     * the db instance.
     * @return
     * @throws Exception
     */
    private static String createKeyStoreFile() throws Exception {
        return createKeyStoreFile(createCertificate()).getPath();
    }

    /**
```

```
     *  This method generates the SSL certificate
     * @return
     * @throws Exception
     */
    private static X509Certificate createCertificate() throws Exception {
        CertificateFactory certFactory = CertificateFactory.getInstance("X.509");
        URL url = new File(SSL_CERTIFICATE).toURI().toURL();
        if (url == null) {
            throw new Exception();
        }
        try (InputStream certInputStream = url.openStream()) {
            return (X509Certificate) certFactory.generateCertificate(certInputStream);
        }
    }

    /**
     * This method creates the Key Store File
     * @param rootX509Certificate - the SSL certificate to be stored in the KeyStore
     * @return
     * @throws Exception
     */
    private static File createKeyStoreFile(X509Certificate rootX509Certificate) throws
Exception {
        File keyStoreFile = File.createTempFile(KEY_STORE_FILE_PREFIX,
KEY_STORE_FILE_SUFFIX);
        try (FileOutputStream fos = new FileOutputStream(keyStoreFile.getPath())) {
            KeyStore ks = KeyStore.getInstance(KEY_STORE_TYPE, KEY_STORE_PROVIDER);
            ks.load(null);
            ks.setCertificateEntry("rootCaCertificate", rootX509Certificate);
            ks.store(fos, DEFAULT_KEY_STORE_PASSWORD.toCharArray());
        }
        return keyStoreFile;
    }

    /**
     * This method clears the SSL properties.
     * @throws Exception
     */
    private static void clearSslProperties() throws Exception {
           System.clearProperty("javax.net.ssl.trustStore");
           System.clearProperty("javax.net.ssl.trustStoreType");
           System.clearProperty("javax.net.ssl.trustStorePassword");
    }
```

```
}
```

Jika Anda ingin terhubung ke instans DB melalui proksi, lihat Terhubung ke sebuah proksi menggunakan autentikasi IAM.

Menghubungkan ke instans DB Anda menggunakan autentikasi IAM dan AWS SDK for Python (Boto3)

Anda dapat menghubungkan ke instans DB RDS for MariaDB, MySQL, atau PostgreSQL dengan AWS SDK for Python (Boto3) seperti yang dijelaskan berikut ini.

Prasyarat

Berikut adalah prasyarat untuk menghubungkan ke instans DB menggunakan autentikasi IAM:

- Mengaktifkan dan menonaktifkan autentikasi basis data IAM
- Membuat dan menggunakan kebijakan IAM untuk akses basis data IAM
- Membuat akun basis data menggunakan autentikasi IAM

Selain itu, pastikan pustaka yang diimpor dalam kode sampel ada di sistem Anda.

Contoh

Contoh kode ini menggunakan profil untuk kredensial bersama. Untuk informasi tentang menentukan kredensial, lihat Credentials dalam dokumentasi AWS SDK for Python (Boto3).

Contoh kode berikut ini menunjukkan cara membuat token autentikasi, lalu menggunakannya untuk terhubung ke instans DB.

Untuk menjalankan contoh kode ini, Anda memerlukan AWS SDK for Python (Boto3), yang ada di situs AWS.

Ubah nilai variabel berikut sesuai kebutuhan:

- `ENDPOINT` – Titik akhir instans DB yang ingin Anda akses
- `PORT` – Nomor port yang digunakan untuk menghubungkan ke instans DB Anda
- `USER` – Akun basis data yang ingin Anda akses
- `REGION` – Wilayah AWS tempat instans DB berjalan

- `DBNAME` – Basis data yang ingin Anda akses
- `SSLCERTIFICATE` – Jalur lengkap ke sertifikat SSL untuk Amazon RDS

  Untuk `ssl_ca`, tentukan sertifikat SSL. Untuk mengunduh sertifikat SSL, lihat .

> **Note**
>
> Anda tidak dapat menggunakan catatan DNS Route 53 kustom sebagai pengganti titik akhir instans DB untuk menghasilkan token autentikasi.

Kode ini terhubung ke instans DB MariaDB MySQL.

Sebelum menjalankan kode ini, instal driver PyMySQL dengan mengikuti petunjuk dalam Python Package Index.

```
import pymysql
import sys
import boto3
import os

ENDPOINT="mysqldb.123456789012.us-east-1.rds.amazonaws.com"
PORT="3306"
USER="jane_doe"
REGION="us-east-1"
DBNAME="mydb"
os.environ['LIBMYSQL_ENABLE_CLEARTEXT_PLUGIN'] = '1'

#gets the credentials from .aws/credentials
session = boto3.Session(profile_name='default')
client = session.client('rds')

token = client.generate_db_auth_token(DBHostname=ENDPOINT, Port=PORT, DBUsername=USER,
 Region=REGION)

try:
    conn =  pymysql.connect(host=ENDPOINT, user=USER, passwd=token, port=PORT,
 database=DBNAME, ssl_ca='SSLCERTIFICATE')
    cur = conn.cursor()
    cur.execute("""SELECT now()""")
    query_results = cur.fetchall()
```

```
    print(query_results)
except Exception as e:
    print("Database connection failed due to {}".format(e))
```

Kode ini terhubung ke instans DB PostgreSQL.

Sebelum menjalankan kode ini, instal `psycopg2` dengan mengikuti petunjuk dalam dokumentasi Psycopg.

```
import psycopg2
import sys
import boto3
import os

ENDPOINT="postgresmydb.123456789012.us-east-1.rds.amazonaws.com"
PORT="5432"
USER="jane_doe"
REGION="us-east-1"
DBNAME="mydb"

#gets the credentials from .aws/credentials
session = boto3.Session(profile_name='RDSCreds')
client = session.client('rds')

token = client.generate_db_auth_token(DBHostname=ENDPOINT, Port=PORT, DBUsername=USER,
 Region=REGION)

try:
    conn = psycopg2.connect(host=ENDPOINT, port=PORT, database=DBNAME, user=USER,
 password=token, sslrootcert="SSLCERTIFICATE")
    cur = conn.cursor()
    cur.execute("""SELECT now()""")
    query_results = cur.fetchall()
    print(query_results)
except Exception as e:
    print("Database connection failed due to {}".format(e))
```

Jika Anda ingin terhubung ke instans DB melalui proksi, lihat Terhubung ke sebuah proksi menggunakan autentikasi IAM.

Memecahkan masalah identitas dan akses Amazon RDS

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda hadapi saat bekerja dengan Amazon RDS dan IAM.

Topik

Saya tidak diberi otorisasi untuk melakukan tindakan di Amazon RDS

Jika AWS Management Console memberi tahu bahwa Anda tidak diberi otorisasi untuk melakukan tindakan, Anda harus menghubungi administrator untuk mendapatkan bantuan. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Contoh kesalahan berikut terjadi saat pengguna `mateojackson` mencoba menggunakan konsol untuk melihat detail tentang *`widget`*, tetapi tidak memiliki izin `rds:`*`GetWidget`*.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 rds:GetWidget on resource: my-example-widget
```

Dalam hal ini, Mateo meminta administratornya untuk memperbarui kebijakannya untuk mengizinkan dia mengakses sumber daya *`my-example-widget`* menggunakan tindakan `rds:`*`GetWidget`*.

Saya tidak memiliki izin untuk melakukan iam:PassRole

Jika Anda menerima kesalahan bahwa Anda tidak diberi otorisasi untuk melakukan tindakan `iam:PassRole`, Anda harus menghubungi administrator untuk mendapatkan bantuan. Administrator Anda adalah orang yang memberi Anda kredensial masuk. Minta orang tersebut untuk memperbarui kebijakan Anda agar Anda dapat meneruskan peran ke Amazon RDS.

Beberapa layanan AWS mengizinkan Anda untuk melewati peran yang sudah ada ke layanan tersebut, alih-alih membuat peran layanan atau peran yang ditautkan ke layanan. Untuk melakukan tindakan tersebut, Anda harus memiliki izin untuk memberikan peran pada layanan tersebut.

Contoh kesalahan berikut terjadi saat pengguna bernama `marymajor` mencoba menggunakan konsol untuk melakukan tindakan di Amazon RDS. Namun, tindakan ini mengharuskan layanan

memiliki izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut pada layanan.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dalam kasus ini, Mary meminta administrator untuk memperbarui kebijakannya agar dia dapat melakukan tindakan `iam:PassRole`.

Saya ingin mengizinkan orang di luar akun AWS saya untuk mengakses sumber daya Amazon RDS

Anda dapat membuat peran yang dapat digunakan para pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACL), Anda dapat menggunakan kebijakan tersebut untuk memberi akses kepada orang ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa hal berikut:

- Untuk mempelajari apakah Amazon RDS mendukung fitur ini, lihat Cara kerja Amazon RDS dengan IAM.
- Untuk mempelajari cara memberikan akses ke sumber daya Anda di seluruh akun AWS yang Anda miliki, lihat Memberikan akses kepada pengguna IAM di akun AWS lain yang Anda miliki dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses ke sumber daya Anda ke akun AWS pihak ke tiga, lihat Memberikan akses ke akun AWS milik pihak ketiga dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, silakan lihat Menyediakan akses ke pengguna terautentikasi eksternal (gabungan identitas) di Panduan Pengguna IAM .
- Untuk mempelajari perbedaan antara penggunaan kebijakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, silakan lihat Bagaimana peran IAM berbeda dari kebijakan berbasis sumber daya di Panduan Pengguna IAM.

Pencatatan dan pemantauan di Amazon RDS

Pemantauan adalah bagian penting dari upaya memelihara keandalan, ketersediaan, dan performa Amazon RDS dan solusi AWS Anda. Anda harus mengumpulkan data pemantauan dari semua

bagian solusi AWS agar dapat dengan lebih mudah melakukan debug kegagalan multi-titik jika terjadi. AWS menyediakan beberapa alat untuk memantau sumber daya Amazon RDS Anda dan merespons 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, status harus diubah dan dipertahankan selama jangka waktu tertentu.

Log AWS CloudTrail

CloudTrail menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di Amazon RDS Aurora. CloudTrail menangkap semua panggilan API untuk Amazon RDS Amazon sebagai peristiwa, termasuk panggilan dari konsol dan dari panggilan kode ke operasi Amazon RDS API. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat untuk Amazon RDS Aurora, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan. Untuk informasi selengkapnya, lihat Memantau panggilan API Amazon RDS di AWS CloudTrail.

Pemantauan yang Ditingkatkan

Amazon RDS menyediakan metrik secara waktu nyata untuk sistem operasi (OS) tempat instans DB berjalan. Anda dapat melihat metrik untuk instans DB menggunakan konsol, atau menggunakan output JSON Pemantauan yang Ditingkatkan dari Amazon CloudWatch Logs dalam sistem pemantauan pilihan Anda. Untuk informasi selengkapnya, lihat Memantau metrik OS dengan Pemantauan yang Disempurnakan.

Wawasan Performa Amazon RDS

Wawasan Performa memperluas fitur pemantauan Amazon RDS yang ada untuk menggambarkan performa basis data Anda dan membantu Anda menganalisis masalah yang memengaruhinya. Dengan dasbor Wawasan Performa, Anda dapat memvisualisasikan beban basis data dan memfilter beban berdasarkan waktu tunggu, pernyataan SQL, host, atau pengguna. Untuk informasi selengkapnya, lihat Memantau muatan DB dengan Wawasan Performa di Amazon RDS.

Log Basis Data

Anda dapat melihat, mengunduh, dan melihat log basis data menggunakan AWS Management Console, AWS CLI, atau RDS API. Untuk informasi selengkapnya, lihat Memantau file log Amazon RDS.

Rekomendasi Amazon RDS

Amazon RDS memberikan rekomendasi otomatis untuk sumber daya basis data. Rekomendasi ini memberikan panduan praktik terbaik dengan menganalisis data konfigurasi, penggunaan, dan performa instans DB. Untuk informasi selengkapnya, lihat Melihat dan menanggapi rekomendasi Amazon Aurora RDS.

Notifikasi Peristiwa Amazon RDS

Amazon RDS menggunakan Amazon Simple Notification Service (Amazon SNS) untuk memberikan notifikasi ketika peristiwa Amazon RDS terjadi. Notifikasi ini bisa dalam bentuk apa pun yang didukung oleh Amazon SNS untuk Wilayah AWS, seperti email, pesan teks, atau panggilan ke titik akhir HTTP. Untuk informasi selengkapnya, lihat Menggunakan pemberitahuan peristiwa Amazon RDS.

AWS Trusted Advisor

Trusted Advisor mengacu pada praktik terbaik yang dipelajari dari melayani ratusan ribu pelanggan AWS. Trusted Advisor memeriksa lingkungan AWS Anda lalu membuat rekomendasi ketika ada peluang untuk menghemat uang, meningkatkan ketersediaan dan performa sistem, atau membantu menutup kesenjangan keamanan. Semua pelanggan AWS memiliki akses ke lima pemeriksaan Trusted Advisor. Pelanggan dengan paket dukungan Bisnis atau Perusahaan dapat melihat semua pemeriksaan Trusted Advisor.

Trusted Advisor memiliki pemeriksaan terkait Amazon RDS berikut:

- Instans DB Diam Amazon RDS
- Risiko Akses Grup Keamanan Amazon RDS
- Pencadangan Amazon RDS
- Multi-AZ Amazon RDS

Lihat informasi selengkapnya tentang beberapa pemeriksaan ini di Praktik terbaik (pemeriksaan) Trusted Advisor.

Untuk informasi selengkapnya tentang cara memantau Amazon RDS, lihat Memantau metrik dalam instans Amazon RDS.

Validasi kepatuhan untuk Amazon RDS

Auditor pihak ketiga menilai keamanan dan kepatuhan Amazon RDS sebagai bagian dari sejumlah program kepatuhan AWS. Program ini termasuk SOC, PCI, FedRAMP, HIPAA, dan lainnya.

Untuk daftar layanan AWS yang termasuk dalam cakupan program kepatuhan khusus, lihat Layanan AWS yang masuk dalam cakupan 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 Artifact.

Tanggung jawab kepatuhan Anda saat menggunakan Amazon RDS ditentukan oleh sensitivitas data Anda, tujuan kepatuhan organisasi Anda, serta hukum dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk memudahkan kepatuhan:

- Panduan mulai cepat untuk keamanan dan kepatuhan – Panduan deployment ini membahas pertimbangan arsitektur dan berisi langkah-langkah untuk melakukan deployment lingkungan dasar yang berfokus pada keamanan dan kepatuhan di AWS.
- Membuat rancangan sesuai Keamanan dan Kepatuhan HIPAA di Amazon Web Services – Laporan resmi ini menjelaskan bagaimana perusahaan dapat menggunakan AWS untuk membuat aplikasi sesuai HIPAA.
- Sumber daya kepatuhan AWS – Kumpulan petunjuk pengoperasian dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- AWS Config – Layanan AWS ini menilai sejauh mana konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- AWS Security Hub – Layanan AWS ini memberikan pandangan komprehensif tentang status keamanan Anda di dalam AWS. Security Hub menggunakan kontrol keamanan untuk mengevaluasi sumber daya AWS Anda dan memeriksa kepatuhan Anda terhadap standar industri dan praktik terbaik keamanan. Untuk daftar layanan dan kontrol yang didukung, lihat Referensi kontrol Security Hub.

Ketangguhan di Amazon RDS

Infrastruktur global AWS dibangun di seputar Kawasan dan Zona Ketersediaan AWS. AWS Kawasan menyediakan beberapa Zona Ketersediaan yang terpisah dan terisolasi secara fisik, yang tersambung dengan jejaring jaringan latensi rendah, throughput tinggi, dan sangat redundan. Dengan Zona Ketersediaan, Anda dapat merancang dan mengoperasikan aplikasi dan basis data yang melakukan secara otomatis pinda saat gagal/failover di antara zona-zona tanpa terputus. Zona Ketersediaan lebih sangat tersedia, lebih toleran kesalahan, dan lebih dapat diskalakan daripadai infrastruktur pusat data tunggal atau multi tradisional.

Lihat informasi yang lebih lengkap tentang Kawasan dan Zona Ketersediaan AWSdi infrastruktur global AWS.

Selain infrastruktur global AWS, Amazon RDS menawarkan beberapa fitur untuk membantu mendukung kebutuhan ketangguhan dan pencadangan data Anda.

Pencadangan dan pemulihan

Amazon RDS membuat dan menyimpan cadangan otomatis instans basis data Anda. Amazon RDS membuat cuplikan volume penyimpanan instans basis data Anda, sehingga mencadangkan seluruh instans basis data dan bukan hanya masing-masing basis data.

Amazon RDS membuat cadangan otomatis instans basis data Anda selama jendela pencadangan instans basis data itu. Amazon RDS menyimpan cadangan otomatis instans basis data Anda sesuai dengan periode retensi cadangan yang Anda tentukan. Jika perlu, Anda dapat memulihkan basis data Anda ke sebarang titik waktu selama periode retensi cadangan. Anda juga dapat mencadangkan instans basis data Anda secara manual, dengan membuat secara manual sebuah cuplikan basis data.

Anda dapat membuat instans basis data dengan memulihkan dari cuplikan basis data ini sebagai solusi pemulihan bencana jika instans basis data sumber gagal.

Lihat informasi yang lebih lengkap di Mencadangkan, memulihkan, dan mengekspor data.

Replikasi

Amazon RDS menggunakan fungsionalitas replikasi bawaan mesin basis data MariaDB, MySQL, Oracle, dan PostgreSQL untuk membuat jenis khusus instans basis data yang disebut dengan replika baca dari instans basis data sumber. Pembaruan yang dibuat terhadap instans basis data sumber

disalin secara asinkron ke replika baca. Anda dapat mengurangi beban pada instans basis data sumber dengan mengarahkan kueri baca dari aplikasi Anda ke replika baca. Dengan replika baca, Anda dapat menskalakan ke luar dengan lentur melebihi batas kapasitas instans basis data tunggal untuk beban kerja basis data yang intensif baca. Anda dapat mempromosikan replika baca menjadi instans mandiri sebagai solusi pemulihan bencana jika instans basis data sumber gagal. Untuk beberapa mesin basis data, Amazon RDS juga mendukung opsi-opsi replikasi lain.

Lihat informasi yang lebih lengkap di Menggunakan replika baca instans DB.

Pindah saat gagal/failover

Amazon RDS memberikan ketersediaan tinggi dan dukungan pindah saat gagal/failover untuk instans basis data dengan menggunakan deployment multi-AZ Multi-AZ. Amazon RDS menggunakan beberapa teknologi untuk memberikan dukungan pindah saat gagal/failover ini. Deployment Multi-AZ untuk instans basis data Oracle, PostgreSQL, MySQL, dan MariaDB menggunakan teknologi pindah saat gagal/failover Amazon. Instans basis data SQL Server menggunakan Database Mirroring (DBM) SQL Server.

Lihat informasi yang lebih lengkap di Mengonfigurasi dan mengelola deployment Multi-AZ.

Keamanan infrastruktur di Amazon RDS

Sebagai layanan terkelola, Amazon Relational Database Service dilindungi oleh keamanan jaringan global AWS. Lihat informasi tentang layanan keamanan AWS dan cara AWS melindungi infrastruktur di Keamanan Cloud AWS. Untuk mendesain lingkungan AWS Anda dengan menggunakan praktik terbaik bagi keamanan infrastruktur, lihat Perlindungan Infrastruktur dalam Pilar Keamanan Kerangka Kerja Berarsitektur Baik AWS.

Anda dapat menggunakan panggilan-panggilan API AWS yang diterbitkan untuk mengakses Amazon RDS 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.

Selain itu, permintaan harus ditandatangani dengan menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan pengguna utama IAM. Atau Anda dapat menggunakan AWS Security Token Service (AWS STS) untuk menghasilkan kredensial keamanan sementara untuk menandatangani permintaan.

Selain itu, Amazon RDS menawarkan fitur-fitur untuk membantu mendukung pemeliharaan infrastruktur keamanan.

Grup keamanan

Grup keamanan mengendalikan akses yang dimiliki oleh lalu lintas masuk dan keluar dari instans basis data. Secara bawaan, akses jaringan ke instans basis data dinonaktifkan. Anda dapat menentukan aturan dalam grup keamanan yang memungkinkan akses dari rentang alamat IP, porta, atau grup keamanan. Setelah aturan masuk dikonfigurasikan, aturan yang sama berlaku untuk semua instans basis data yang dikaitkan dengan grup keamanan itu.

Lihat informasi yang lebih lengkap di Mengontrol akses dengan grup keamanan.

Aksesibilitas publik

Saat Anda meluncurkan instans basis data di dalam Cloud Privat Virtual (VPC) berbasis layanan Amazon VPC, Anda dapat mengaktifkan atau menonaktifkan akses publik untuk instans itu. Untuk

menentukan apakah instans basis data yang Anda buat memiliki nama DNS yang terselesaikan ke alamat IP publik, Anda menggunakan parameter Aksesibilitas publik. Dengan menggunakan parameter ini, Anda dapat menetapkan apakah ada akses publik ke instans basis data. Anda dapat mengubah instans basis data untuk mengaktifkan atau menonaktifkan aksesibilitas publik dengan mengubah parameter Aksesibilitas publik.

Lihat informasi yang lebih lengkap di Menyembunyikan klaster DB dalam VPC dari internet.

> **Note**
>
> Jika instans basis data Anda ada dalam VPC tetapi tidak dapat diakses publik, Anda juga dapat menggunakan koneksi VPN Lokasi-ke-Lokasi AWS atau koneksi AWS Direct Connect untuk mengaksesnya dari jaringan privat. Lihat informasi yang lebih lengkap di Privasi lalu lintas jaringan internet.

API Amazon RDS dan titik akhir VPC antarmuka (AWS PrivateLink)

Anda dapat membuat koneksi privat antara titik akhir VPC dan API Amazon RDS dengan membuat titik akhir VPC antarmuka. Titik akhir antarmuka didukung oleh AWS PrivateLink.

AWS PrivateLink memungkinkan Anda mengakses operasi API Amazon RDS secara privat tanpa gateway internet, perangkat NAT, koneksi VPN, atau koneksi AWS Direct Connect. Instans DB dalam VPC Anda tidak memerlukan alamat IP publik untuk berkomunikasi dengan titik akhir API Amazon RDS untuk meluncurkan, memodifikasi, atau menghentikan instans DB . Instans DB Anda juga tidak memerlukan alamat IP publik untuk menggunakan salah satu dari operasi API RDS yang tersedia. Lalu lintas antara VPC Anda dan Amazon RDS tidak keluar dari jaringan Amazon.

Setiap titik akhir antarmuka direpresentasikan oleh satu atau beberapa antarmuka jaringan elastis di subnet Anda. Untuk informasi selengkapnya tentang antarmuka jaringan elastis, lihat Antarmuka jaringan elastis dalam Panduan Pengguna Amazon EC2.

Untuk informasi selengkapnya tentang titik akhir VPC, lihat Titik akhir VPC Antarmuka () di AWS PrivateLink Panduan Pengguna Amazon VPC. Untuk informasi selengkapnya tentang operasi API RDS, lihat Referensi API Amazon RDS.

Anda tidak memerlukan titik akhir VPC antarmuka untuk terhubung ke instans DB. Untuk informasi selengkapnya, lihat Skenario untuk mengakses instans DB di VPC.

Pertimbangan untuk titik akhir VPC

Sebelum Anda menyiapkan titik akhir VPC antarmuka untuk titik akhir API Amazon RDS, pastikan Anda meninjau Properti dan batasan titik akhir antarmuka dalam Panduan Pengguna Amazon VPC.

Semua operasi API RDS yang relevan dengan pengelolaan sumber daya Amazon RDS tersedia dari VPC Anda menggunakan AWS PrivateLink.

Kebijakan titik akhir VPC didukung untuk titik akhir API RDS. Secara default, akses penuh ke operasi API RDS diizinkan melalui titik akhir. Untuk informasi selengkapnya, lihat Mengontrol akses ke layanan dengan titik akhir VPC dalam Panduan Pengguna Amazon VPC.

Ketersediaan

API Amazon RDS saat ini mendukung titik akhir VPC di Wilayah AWS berikut:

- AS Timur (Ohio)
- AS Timur (Virginia Utara)
- AS Barat (California Utara)
- AS Barat (Oregon)
- Afrika (Cape Town)
- Asia Pasifik (Hong Kong)
- Asia Pasifik (Mumbai)
- Asia Pasifik (Osaka)
- Asia Pasifik (Seoul)
- Asia Pasifik (Singapura)
- Asia Pasifik (Sydney)
- Asia Pasifik (Tokyo)
- (Canada (Central)
- Kanada Barat (Calgary)
- China (Beijing)
- Tiongkok (Ningxia)
- Eropa (Frankfurt)
- Eropa (Zurich)
- Eropa (Irlandia)
- Eropa (London)
- Eropa (Paris)
- Eropa (Stockholm)
- Eropa (Milan)
- Israel (Tel Aviv)
- Timur Tengah (Bahrain)
- Amerika Selatan (Sao Paulo)
- AWS GovCloud (AS-Timur)
- AWS GovCloud (AS-Barat)

Membuat titik akhir VPC antarmuka untuk API Amazon RDS

Anda dapat membuat titik akhir VPC untuk API Amazon RDS menggunakan konsol Amazon VPC atau AWS Command Line Interface (AWS CLI). Untuk informasi selengkapnya, lihat Membuat titik akhir antarmuka dalam Panduan Pengguna Amazon VPC.

Buat titik akhir VPC untuk API Amazon RDS menggunakan nama layanan `com.amazonaws.region.rds`.

Kecuali Wilayah AWS di Tiongkok, jika Anda mengaktifkan DNS privat untuk titik akhir, Anda dapat membuat permintaan API ke Amazon RDS dengan titik akhir VPC menggunakan nama DNS default untuk Wilayah AWS, misalnya `rds.us-east-1.amazonaws.com`. Untuk Wilayah AWS Tiongkok (Beijing) dan Tiongkok (Ningxia), Anda dapat membuat permintaan API dengan titik akhir VPC menggunakan `rds-api.cn-north-1.amazonaws.com.cn` dan `rds-api.cn-northwest-1.amazonaws.com.cn`.

Untuk informasi selengkapnya, lihat Mengakses layanan melalui titik akhir antarmuka dalam Panduan Pengguna Amazon VPC.

Membuat kebijakan titik akhir VPC untuk API Amazon RDS

Anda dapat menyisipkan kebijakan titik akhir ke titik akhir VPC yang mengontrol akses ke API Amazon RDS. Kebijakan titik akhir menentukan informasi berikut:

- Prinsipal yang dapat melakukan tindakan.
- Tindakan yang dapat dilakukan.
- Sumber daya yang menjadi target tindakan.

Untuk informasi selengkapnya, lihat Mengontrol akses ke layanan dengan titik akhir VPC dalam Panduan Pengguna Amazon VPC.

Contoh: Kebijakan titik akhir VPC untuk tindakan API Amazon RDS

Berikut ini adalah contoh kebijakan titik akhir untuk API Amazon RDS. Jika dilampirkan ke sebuah titik akhir, kebijakan ini memberikan akses ke tindakan API Amazon RDS untuk semua prinsipal di semua sumber daya.

```
{
   "Statement":[
```

```
      {
         "Principal":"*",
         "Effect":"Allow",
         "Action":[
            "rds:CreateDBInstance",
            "rds:ModifyDBInstance",
            "rds:CreateDBSnapshot"
         ],
         "Resource":"*"
      }
   ]
}
```

Contoh: Kebijakan titik akhir VPC yang menolak semua akses dari akun AWS yang ditentukan

Kebijakan titik akhir VPC berikut menolak semua akses akun AWS `123456789012` ke sumber daya yang menggunakan titik akhir tersebut. Kebijakan ini memungkinkan semua tindakan dari akun lain.

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": "*"
    },
    {
      "Action": "*",
      "Effect": "Deny",
      "Resource": "*",
      "Principal": {
        "AWS": [
          "123456789012"
        ]
      }
   ]
}
```

Praktik terbaik keamanan untuk Amazon RDS

Gunakan akun AWS Identity and Access Management (IAM) untuk mengontrol akses ke operasi API Amazon RDS, khususnya operasi yang membuat, memodifikasi, atau menghapus sumber daya

Amazon RDS. Sumber daya tersebut termasuk instans DB, grup keamanan, dan grup parameter. IAM juga dapat digunakan untuk mengontrol tindakan yang melakukan tindakan administratif umum seperti mencadangkan dan memulihkan instans DB.

- Buat pengguna individual untuk setiap orang yang mengelola sumber daya Amazon RDS, termasuk Anda sendiri. Jangan gunakan kredensial akar AWS untuk mengelola sumber daya Amazon RDS.
- Beri setiap pengguna set izin minimum yang diperlukan untuk melakukan tugas-tugasnya.
- Gunakan grup IAM untuk mengelola izin secara efektif bagi beberapa pengguna.
- Putar kredensial IAM Anda secara rutin.
- Konfigurasikan AWS Secrets Manager untuk memutar rahasia untuk Amazon RDS secara otomatis. Untuk informasi selengkapnya, lihat Memutar rahasia AWS Secrets Manager Anda di Panduan Pengguna AWS Secrets Manager. Anda juga dapat mengambil kredensial dari AWS Secrets Manager secara terprogram. Untuk informasi selengkapnya, lihat Mengambil nilai rahasia di Panduan Pengguna AWS Secrets Manager.

Untuk informasi selengkapnya tentang keamanan Amazon RDS, lihat Keamanan dalam Amazon RDS. Untuk informasi selengkapnya tentang IAM, lihat AWS Identity and Access Management. Untuk informasi tentang praktik terbaik IAM, lihat Praktik terbaik IAM.

AWS Security Hub menggunakan kontrol keamanan untuk mengevaluasi konfigurasi sumber daya dan standar keamanan untuk membantu Anda mematuhi berbagai kerangka kerja kepatuhan. Untuk informasi selengkapnya tentang penggunaan Security Hub guna mengevaluasi sumber daya RDS, lihat Kontrol Amazon Relational Database Service di Panduan Pengguna AWS Security Hub.

Anda dapat memantau penggunaan RDS yang berkaitan dengan praktik terbaik keamanan dengan menggunakan Security Hub. Untuk informasi selengkapnya, lihat Apa yang dimaksud dengan AWS Security Hub?

Gunakan AWS Management Console, AWS CLI, atau API RDS untuk mengubah kata sandi bagi pengguna utama Anda. Jika Anda menggunakan alat lain, seperti klien SQL, untuk mengubah kata sandi pengguna utama, hak istimewa pengguna kemungkinan dapat terhapus secara tidak sengaja.

Mengontrol akses dengan grup keamanan

Grup keamanan VPC mengontrol akses lalu lintas masuk dan keluar dari instans DB. Secara default, akses jaringan untuk instans DB dinonaktifkan. Anda dapat menentukan aturan dalam grup

keamanan yang mengizinkan akses dari rentang alamat IP, port, atau grup keamanan. Setelah aturan masuk dikonfigurasi, aturan yang sama berlaku untuk semua instans DB yang terkait dengan grup keamanan tersebut. Anda dapat menentukan hingga 20 aturan dalam satu grup keamanan.

Ikhtisar grup keamanan VPC

Setiap aturan grup keamanan VPC memungkinkan sumber tertentu untuk mengakses instans DB dalam VPC yang terkait dengan grup keamanan VPC tersebut. Sumbernya dapat berupa rentang alamat (misalnya, 203.0.113.0/24), atau grup keamanan VPC lain. Dengan menentukan grup keamanan VPC sebagai sumber, Anda mengizinkan lalu lintas masuk dari semua instans (biasanya server aplikasi) yang menggunakan grup keamanan VPC sumber. Grup keamanan VPC dapat memiliki aturan yang mengatur lalu lintas masuk dan keluar. Namun, aturan lalu lintas keluar biasanya tidak berlaku untuk instans DB. Aturan lalu lintas keluar hanya berlaku jika instans DB bertindak sebagai klien. Misalnya, aturan lalu lintas keluar berlaku untuk instans DB Oracle dengan tautan basis data keluar. Anda harus menggunakan opsi API Amazon EC2 atau Grup Keamanan pada konsol VPC untuk membuat grup keamanan VPC.

Saat Anda membuat aturan untuk grup keamanan VPC yang memungkinkan akses ke instans di VPC, Anda harus menentukan port untuk setiap rentang alamat yang diizinkan oleh aturan tersebut. Misalnya, jika Anda ingin mengaktifkan akses Secure Shell (SSH) untuk instans di VPC, buat aturan yang mengizinkan akses ke port 22 TCP untuk rentang alamat tertentu.

Anda dapat mengonfigurasi beberapa grup keamanan VPC yang mengizinkan akses ke port yang berbeda untuk instans yang berbeda di VPC Anda. Misalnya, Anda dapat membuat grup keamanan VPC yang memungkinkan akses ke port 80 TCP untuk server web di VPC Anda. Anda kemudian dapat membuat grup keamanan VPC lainnya yang memungkinkan akses ke port 3306 TCP untuk instans DB RDS for MySQL dalam VPC Anda.

Untuk informasi selengkapnya tentang grup keamanan VPC, lihat Grup keamanan di Panduan Pengguna Amazon Virtual Private Cloud.

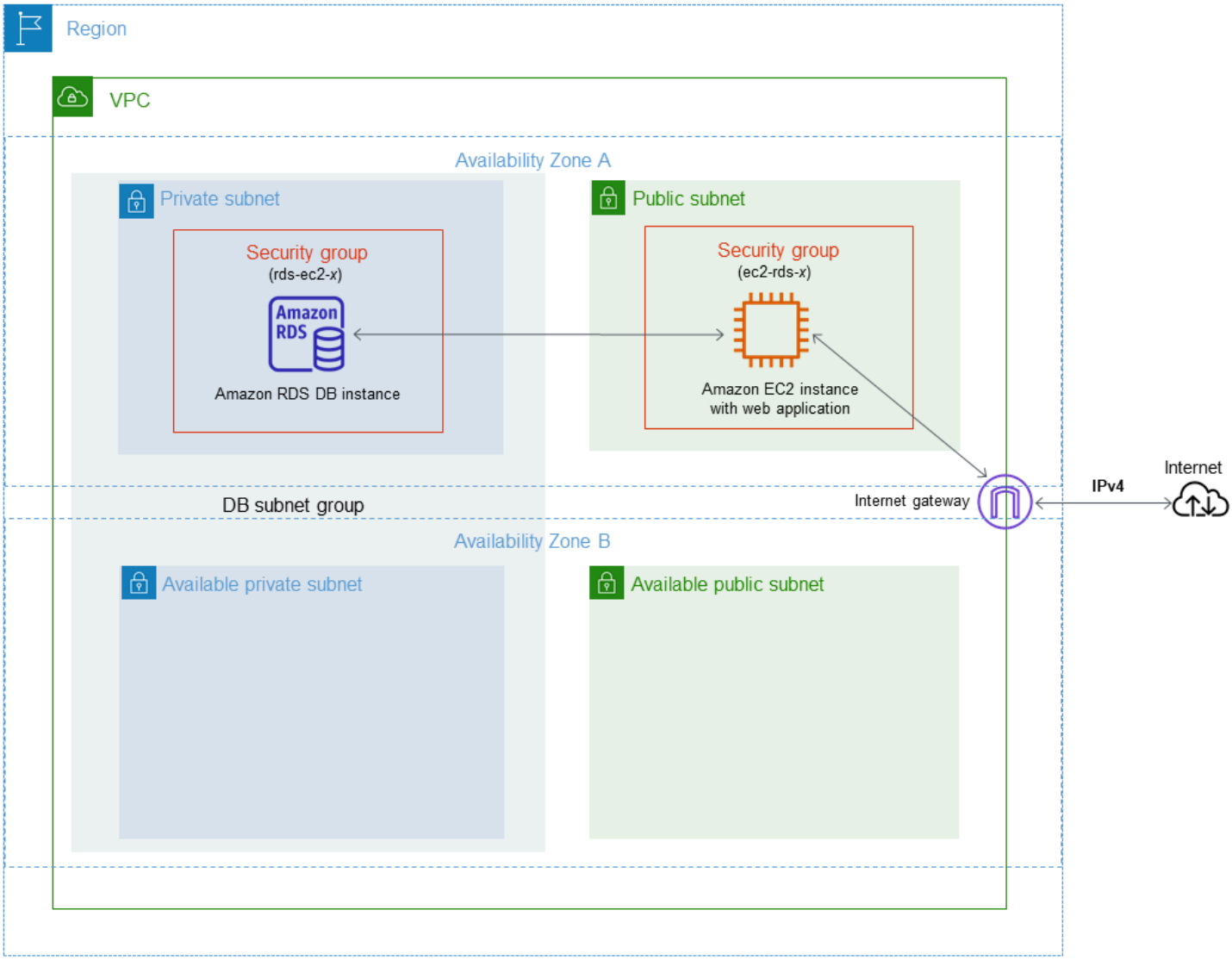
> **Note**
>
> Jika instans DB Anda berada dalam VPC tetapi tidak dapat diakses publik, Anda juga dapat menggunakan koneksi AWS VPN Site-to-Site atau AWS Direct Connect koneksi untuk mengaksesnya dari jaringan pribadi. Untuk informasi selengkapnya, lihat Privasi lalu lintas jaringan internet.

Skenario grup keamanan

Penggunaan umum instans DB di VPC adalah untuk berbagi data dengan server aplikasi yang dijalankan di instans Amazon EC2 dalam VPC yang sama, yang diakses oleh aplikasi klien di luar VPC. Untuk skenario ini, Anda menggunakan halaman RDS dan VPC di AWS Management Console atau operasi API RDS dan EC2 untuk membuat instans dan grup keamanan yang diperlukan:

1. Buat grup keamanan VPC (misalnya, `sg-0123ec2example`) dan tentukan aturan masuk yang menggunakan alamat IP aplikasi klien sebagai sumber. Grup keamanan ini memungkinkan aplikasi klien Anda untuk terhubung ke instans EC2 dalam VPC yang menggunakan grup keamanan ini.
2. Buat instans EC2 untuk aplikasi dan tambahkan instans EC2 ke grup keamanan VPC (`sg-0123ec2example`) yang Anda buat pada langkah sebelumnya.
3. Buat grup keamanan VPC kedua (misalnya, `sg-6789rdsexample`) dan buat aturan baru dengan menentukan grup keamanan VPC yang Anda buat di langkah 1 (`sg-0123ec2example`) sebagai sumbernya.
4. Buat instans DB baru dan tambahkan instans DB tersebut ke grup keamanan VPC (`sg-6789rdsexample`) yang telah Anda buat pada langkah sebelumnya. Saat Anda membuat instans DB, gunakan nomor port yang sama dengan yang ditentukan untuk aturan grup keamanan VPC (`sg-6789rdsexample`) yang Anda buat pada langkah 3.

Diagram berikut menunjukkan skenario ini.



Untuk petunjuk lengkap tentang konfigurasi VPC untuk skenario ini, lihat Tutorial: Membuat VPC untuk digunakan dengan instans DB (khusus IPv4). Untuk informasi selengkapnya tentang penggunaan VPC, lihat Amazon VPC dan Amazon RDS.

Membuat grup keamanan VPC

Anda dapat membuat grup keamanan VPC untuk instans DB menggunakan konsol VPC. Untuk informasi tentang pembuatan grup keamanan, lihat Memberikan akses ke instans DB di VPC Anda dengan membuat grup keamanan dan Grup Keamanan dalam Panduan Pengguna Amazon Virtual Private Cloud.

Mengaitkan grup keamanan dengan instans DB

Anda dapat mengaitkan grup keamanan dengan instans DB menggunakan Modify di konsol RDS, `ModifyDBInstance` Amazon RDS API, atau perintah. `modify-db-instance` AWS CLI

Contoh CLI berikut mengaitkan grup keamanan VPC tertentu dan menghapus grup keamanan DB dari instans DB

```
aws rds modify-db-instance --db-instance-identifier dbName --vpc-security-group-ids sg-ID
```

Untuk informasi tentang modifikasi instans DB, lihat Memodifikasi instans DB Amazon RDS. Untuk pertimbangan grup keamanan saat Anda memulihkan instans DB dari snapshot DB, lihat Pertimbangan grup keamanan.

> **Note**
>
> Konsol RDS menampilkan nama aturan grup keamanan yang berbeda untuk basis data Anda jika nilai Port dikonfigurasi ke nilai non-default.

Hak akses akun pengguna master

Saat Anda membuat instans DB baru, pengguna master default yang Anda gunakan akan mendapatkan hak akses tertentu untuk instans DB tersebut. Anda tidak dapat mengubah nama pengguna master setelah instans DB dibuat.

> **Important**
>
> Kami sangat menyarankan agar Anda tidak menggunakan pengguna master secara langsung di aplikasi Anda. Sebagai gantinya, ikuti praktik terbaik menggunakan pengguna basis data yang dibuat dengan hak akses paling rendah yang diperlukan untuk aplikasi Anda.

> **Note**
>
> Jika Anda secara tidak sengaja menghapus izin bagi pengguna master, Anda dapat memulihkannya dengan memodifikasi instans DB dan mengatur kata sandi pengguna

master yang baru. Untuk informasi selengkapnya tentang cara memodifikasi instans DB, lihat Memodifikasi instans DB Amazon RDS.

Tabel berikut menunjukkan hak akses dan peran basis data yang diperoleh pengguna master untuk masing-masing mesin basis data.

Mesin basis data	Hak akses sistem	Peran basis data
Db2	Pengguna master ditetapkan ke grup `masterdba` dan menetapkan `master_user_role` . `SYSMON`, `DBADM` dengan `DATAACCESS` DAN `ACCCESSCTRL` , `BINDADD`, `CONNECT`, `CREATETAB` , `CREATE_SECURE_OBJECT` , `EXPLAIN`, `IMPLICIT_SCHEMA` , `LOAD`, `SQLADM`, `WLMADM`	`DBA`, `DBA_RESTRICTED` , `DEVELOPER` , `ROLE_NULL ID_PACKAGES` , `ROLE_PROCEDURES` , `ROLE_TABLESPACES`
MySQL dan MariaDB	`SELECT`, `INSERT`, `UPDATE`, `DELETE`, `CREATE`, `DROP`, `RELOAD`, `PROCESS`, `REFERENCES` , `INDEX`, `ALTER`, `SHOW DATABASES` , `CREATE TEMPORARY TABLES`, `LOCK TABLES`, `EXECUTE`, `REPLICATION CLIENT` , `CREATE VIEW`, `SHOW VIEW`, `CREATE ROUTINE`, `ALTER ROUTINE`, `CREATE USER`, `EVENT`, `TRIGGER ON *.* WITH GRANT OPTION`, `REPLICATION SLAVE`	—
PostgreSQL	`CREATE ROLE`, `CREATE DB`, `PASSWORD VALID UNTIL INFINITY`, `CREATE EXTENSION` , `ALTER EXTENSION` , `DROP EXTENSION` , `CREATE TABLESPACE` , `ALTER <OBJECT> OWNER`, `CHECKPOINT` , `PG_CANCEL_BACKEND( )` , `PG_TERMINATE_BACKEND()` , `SELECT PG_STAT_REPLICATION` , `EXECUTE PG_STAT_STATEMENTS_RESET()` , `OWN POSTGRES_FDW_HANDLER()` , `OWN POSTGRES_FDW_VALID`	`RDS_SUPERUSER` Untuk informasi selengkapnya tentang RDS_SUPERUSER, lihat Memahami peran dan izin PostgreSQL.

Mesin basis data	Hak akses sistem	Peran basis data
	ATOR() , OWN POSTGRES_FDW , EXECUTE PG_BUFFERCACHE_PAGES() , SELECT PG_BUFFERCACHE	
Oracle	ALTER DATABASE LINK, ALTER PUBLIC DATABASE LINK, DROP ANY DIRECTORY , EXEMPT ACCESS POLICY, EXEMPT IDENTITY POLICY, GRANT ANY OBJECT PRIVILEGE , RESTRICTED SESSION , EXEMPT REDACTION POLICY	AQ_ADMINISTRATOR_ROLE , AQ_USER_ROLE , CONNECT, CTXAPP, DBA, EXECUTE_CATALOG_ROLE , RECOVERY_CATALOG_OWNER , RESOURCE, SELECT_CATALOG_ROLE
Microsoft SQL Server	ADMINISTER BULK OPERATIONS , ALTER ANY CONNECTION , ALTER ANY CREDENTIAL , ALTER ANY EVENT SESSION, ALTER ANY LINKED SERVER, ALTER ANY LOGIN, ALTER ANY SERVER AUDIT, ALTER ANY SERVER ROLE, ALTER SERVER STATE, ALTER TRACE, CONNECT SQL, CREATE ANY DATABASE, VIEW ANY DATABASE, VIEW ANY DEFINITION , VIEW SERVER STATE, ALTER ON ROLE SQLAgentOperatorRole	DB_OWNER (peran tingkat basis data), PROCESSADMIN (peran tingkat server), SETUPADMIN (peran tingkat server), SQLAgentUserRole (peran tingkat basis data)

Menggunakan peran terkait layanan untuk Amazon RDS

Amazon RDS menggunakan peran terkait layanan AWS Identity and Access Management (IAM). Peran yang terkait layanan adalah jenis peran IAM unik yang ditautkan langsung ke Amazon RDS. Peran terkait layanan telah ditentukan sebelumnya oleh Amazon RDS dan mencakup semua izin yang diperlukan layanan untuk memanggil layanan AWS lainnya atas nama Anda.

Peran terkait layanan memudahkan penggunaan Amazon RDS karena Anda tidak perlu menambahkan izin yang diperlukan secara manual. Amazon RDS menentukan izin peran terkait layanannya, dan kecuali jika ditentukan lain, hanya Amazon RDS yang dapat mengambil perannya. Izin yang ditentukan mencakup kebijakan kepercayaan dan kebijakan izin, serta bahwa kebijakan izin tidak dapat dilampirkan ke entitas IAM lainnya.

Anda dapat menghapus peran-peran tersebut hanya setelah pertama kali menghapus sumber dayanya yang terkait. Cara ini akan melindungi sumber daya Amazon RDS karena Anda tidak dapat menghapus izin untuk mengakses sumber daya tersebut secara tidak sengaja.

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 RDS

Amazon RDS menggunakan peran terkait layanan yang bernama AWSServiceRoleForRDS agar Amazon RDS dapat memanggil layanan AWS atas nama instans DB Anda.

Peran terkait layanan AWSServiceRoleForRDS memercayai layanan berikut untuk mengambil peran:

- `rds.amazonaws.com`

Peran terkait layanan ini memiliki kebijakan izin yang menyertainya bernama `AmazonRDSServiceRolePolicy` yang memberikannya izin untuk beroperasi di akun Anda. Kebijakan izin peran memungkinkan Amazon RDS menyelesaikan tindakan berikut pada sumber daya yang ditentukan:

Untuk informasi selengkapnya tentang kebijakan ini, termasuk dokumen kebijakan JSON, lihat AmazonRDSServiceRolePolicy dalam Panduan Referensi Kebijakan Terkelola AWS.

> **Note**
>
> Anda harus mengonfigurasi izin agar entitas IAM (seperti pengguna, grup, atau peran) dapat membuat, mengedit, atau menghapus peran terkait layanan. Jika Anda menemukan pesan kesalahan berikut:
> Tidak dapat membuat sumber daya. Verifikasi bahwa Anda memiliki izin untuk membuat peran terkait layanan. Jika tidak, tunggu dan coba lagi nanti.
> Pastikan Anda telah mengaktifkan izin berikut:
>
> ```
> {
>     "Action": "iam:CreateServiceLinkedRole",
>     "Effect": "Allow",
>     "Resource": "arn:aws:iam::*:role/aws-service-role/rds.amazonaws.com/
> AWSServiceRoleForRDS",
>     "Condition": {
>         "StringLike": {
>             "iam:AWSServiceName":"rds.amazonaws.com"
>         }
>     }
> }
> ```
>
> Untuk informasi selengkapnya, lihat Izin peran terkait layanan dalam Panduan Pengguna IAM.

Membuat peran terkait layanan untuk Amazon RDS

Anda tidak perlu membuat peran terkait layanan secara manual. Saat Anda membuat instans DB, Amazon RDS membuat peran terkait layanan untuk Anda.

> **Important**
>
> Jika Anda menggunakan layanan Amazon RDS sebelum 1 Desember 2017, saat layanan tersebut mulai mendukung peran terkait layanan, Amazon RDS akan membuat peran AWSServiceRoleForRDS di akun Anda. Untuk mempelajari selengkapnya, lihat Peran baru yang muncul di akun AWS saya.

Jika Anda menghapus peran terkait layanan ini, dan ingin membuatnya lagi, Anda dapat mengulangi proses yang sama untuk membuat kembali peran tersebut di akun Anda. Saat Anda membuat instans DB, Amazon RDS membuat peran terkait layanan untuk Anda.

Mengedit peran terkait layanan untuk Amazon RDS

Amazon RDS tidak mengizinkan Anda untuk mengedit peran terkait layanan AWSServiceRoleForRDS. Setelah membuat peran terkait layanan, Anda tidak dapat mengubah nama peran karena berbagai entitas mungkin merujuk peran tersebut. Namun, Anda dapat mengedit deskripsi peran ini menggunakan IAM. Untuk informasi selengkapnya, lihat Mengedit peran terkait layanan dalam Panduan Pengguna IAM.

Menghapus peran terkait layanan untuk Amazon RDS

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. Namun, Anda harus menghapus semua instans DB Anda sebelum Anda dapat menghapus peran terkait layanan.

Membersihkan peran terkait layanan

Sebelum Anda dapat menggunakan IAM untuk menghapus peran terkait layanan, Anda harus mengonfirmasi terlebih dahulu bahwa peran tersebut tidak memiliki sesi aktif dan menghapus sumber daya yang digunakan oleh peran tersebut.

Untuk memastikan peran terkait layanan memiliki sesi aktif di konsol IAM

1. Masuk ke AWS Management Console lalu buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi konsol IAM, pilih Peran. Kemudian, pilih nama (bukan kotak centang) untuk peran AWSServiceRoleForRDS.
3. Di halaman Ringkasan untuk peran yang dipilih, pilih tab Penasihat Akses.
4. Di tab Penasihat Akses, tinjau aktivitas terbaru untuk peran terkait layanan tersebut.

> **Note**
>
> Jika Anda tidak yakin apakah Amazon RDS menggunakan peran AWSServiceRoleForRDS tersebut, coba hapus peran tersebut. Jika layanan ini menggunakan peran tersebut, peran tidak dapat dihapus dan Anda dapat melihat

Wilayah AWS tempat peran tersebut digunakan. Jika peran tersebut sedang digunakan, Anda harus menunggu hingga sesi ini berakhir sebelum dapat menghapus peran tersebut. Anda tidak dapat mencabut sesi untuk peran terkait layanan.

Jika Anda ingin menghapus peran AWSServiceRoleForRDS, Anda harus terlebih dahulu menghapus semua instans DB Anda.

Menghapus semua instans

Gunakan salah satu dari prosedur ini untuk menghapus setiap instans Anda.

Untuk menghapus sebuah instans (konsol)

1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Basis data.
3. Pilih instans yang ingin Anda hapus.
4. Untuk Tindakan, pilih Hapus.
5. Jika Anda diminta untuk Buat Snapshot akhir?, pilih Ya atau Tidak.
6. Jika Anda memilih Ya pada langkah sebelumnya, untuk Nama snapshot akhir masukkan nama snapshot akhir Anda.
7. Pilih Hapus.

Untuk menghapus sebuah instans (CLI)

Lihat `delete-db-instance` dalam Referensi Perintah AWS CLI.

Untuk menghapus sebuah instans (API)

Lihat `DeleteDBInstance` dalam Referensi API Amazon RDS.

Anda dapat menggunakan konsol IAM, CLI IAM, atau API IAM untuk menghapus peran terkait layanan AWSServiceRoleForRDS. Untuk informasi selengkapnya, lihat Menghapus peran terkait layanan dalam Panduan Pengguna IAM.

Izin peran terkait layanan untuk Amazon RDS Custom

Amazon RDS Custom menggunakan peran terkait layanan yang disebut `AWSServiceRoleForRDSCustom` untuk mengizinkan RDS Custom memanggil AWS layanan atas nama instans DB dan klaster DB Anda.

Peran terkait layanan AWSServiceRoleForRDSCustom memercayai layanan berikut ini untuk menjalankan peran:

- `custom.rds.amazonaws.com`

Peran terkait layanan ini memiliki kebijakan izin yang menyertainya bernama `AmazonRDSCustomServiceRolePolicy` yang memberikannya izin untuk beroperasi di akun Anda. Kebijakan izin peran memungkinkan RDS Custom menyelesaikan tindakan berikut pada sumber daya yang ditentukan:

Untuk informasi selengkapnya tentang kebijakan ini, termasuk dokumen kebijakan JSON, lihat AmazonRDSCustomServiceRolePolicy dalam Panduan Referensi Kebijakan Terkelola AWS.

Membuat, mengedit, atau menghapus peran terkait layanan untuk RDS Custom berfungsi sama seperti untuk Amazon RDS. Untuk informasi selengkapnya, lihat Izin peran terkait layanan untuk Amazon RDS.

> **Note**
>
> Anda harus mengonfigurasi izin agar entitas IAM (seperti pengguna, grup, atau peran) dapat membuat, mengedit, atau menghapus peran terkait layanan. Jika Anda menemukan pesan kesalahan berikut:
> Tidak dapat membuat sumber daya. Verifikasi bahwa Anda memiliki izin untuk membuat peran terkait layanan. Jika tidak, tunggu dan coba lagi nanti.
> Pastikan Anda telah mengaktifkan izin berikut:
>
> ```
> {
>     "Action": "iam:CreateServiceLinkedRole",
>     "Effect": "Allow",
>     "Resource": "arn:aws:iam::*:role/aws-service-role/custom.rds.amazonaws.com/
> AmazonRDSCustomServiceRolePolicy",
>     "Condition": {
>         "StringLike": {
>             "iam:AWSServiceName":"custom.rds.amazonaws.com"
> ```

```
        }
    }
}
```

Untuk informasi selengkapnya, lihat Izin peran terkait layanan dalam Panduan Pengguna IAM.

Amazon VPC dan Amazon RDS

Amazon Virtual Private Cloud (Amazon VPC) memungkinkan Anda meluncurkan sumber daya AWS, seperti instans DB Amazon RDS, ke cloud privat virtual (VPC).

Saat menggunakan VPC, Anda dapat mengontrol lingkungan jaringan virtual Anda. Anda dapat memilih rentang alamat IP Anda sendiri, membuat subnet, serta mengonfigurasi perutean dan daftar kontrol akses. Tidak ada biaya tambahan untuk menjalankan instans DB Anda dalam VPC.

Akun memiliki VPC default. Semua instans DB baru dibuat dalam VPC default kecuali Anda menentukan lain.

Topik

Berikut ini, Anda dapat menemukan pembahasan tentang fungsi VPC yang relevan dengan instans DB Amazon RDS. Untuk informasi selengkapnya tentang Amazon VPC, lihat Panduan Memulai Amazon VPC dan Panduan Pengguna Amazon VPC.

Bekerja dengan klaster DB dalam VPC

Klaster DB Anda berada di cloud privat virtual (VPC). VPC adalah jaringan virtual yang secara logis terisolasi dari jaringan virtual lain di Cloud AWS. Amazon VPC memungkinkan Anda meluncurkan sumber daya AWS, seperti klaster DB Amazon RDS atau instans Amazon EC2, ke VPC. VPC dapat berupa VPC default dari akun Anda atau VPC yang Anda buat. Semua VPC dikaitkan dengan akun AWS Anda.

VPC default memiliki tiga subnet yang dapat digunakan untuk mengisolasi sumber daya di dalam VPC. VPC default juga memiliki gateway internet yang dapat digunakan untuk memberikan akses ke sumber daya di dalam VPC dari luar VPC.

Untuk daftar skenario yang melibatkan klaster DB Amazon RDS di dalam VPC dan di luar VPC, lihat Skenario untuk mengakses instans DB di VPC.

Topik

Dalam tutorial berikut, Anda dapat mempelajari cara membuat VPC yang dapat Anda gunakan untuk skenario Amazon RDS yang umum:

Bekerja dengan klaster DB dalam VPC

Berikut adalah beberapa tips cara bekerja dengan klaster DB dalam VPC:

- VPC Anda harus memiliki minimal dua subnet. Subnet ini harus berada di dua Zona Ketersediaan yang berbeda di Wilayah AWS tempat Anda ingin men-deploy klaster DB. Subnet adalah segmen dari rentang alamat IP VPC yang dapat Anda tentukan dan gunakan untuk mengelompokkan klaster DB berdasarkan kebutuhan keamanan dan operasional Anda.

  Untuk deployment Multi-AZ, menentukan subnet untuk dua atau beberapa Zona Ketersediaan di Wilayah AWS memungkinkan Amazon RDS membuat fungsi siaga baru di Zona Ketersediaan lain sesuai kebutuhan. Pastikan untuk melakukan hal ini bahkan untuk deployment AZ-Tunggal, jika nantinya Anda ingin mengonversinya ke deployment Multi-AZ.

  > **Note**
  >
  > Grup subnet DB untuk Zona Lokal hanya boleh memiliki satu subnet.

- Jika Anda ingin klaster DB dalam VPC dapat diakses publik, pastikan untuk mengaktifkan nama host DNS dan resolusi DNS atribut VPC.
- VPC Anda harus memiliki grup subnet DB yang Anda buat. Anda membuat grup subnet DB dengan menentukan subnet yang telah Anda buat. Amazon RDS memilih subnet dan alamat IP dalam

grup subnet tersebut untuk dikaitkan dengan instans DB Anda. Instans DB menggunakan Zona Ketersediaan yang berisi subnet tersebut.

- VPC Anda harus memiliki grup keamanan VPC yang memungkinkan akses ke klaster DB.

  Untuk informasi selengkapnya, lihat Skenario untuk mengakses instans DB di VPC.

- Blok CIDR di setiap subnet Anda harus cukup besar untuk mengakomodasi alamat IP cadangan untuk Amazon RDS untuk digunakan selama aktivitas pemeliharaan, termasuk failover dan penskalaan komputasi. Misalnya, rentang seperti 10.0.0.0/24 dan 10.0.1.0/24 biasanya memiliki kapasitas cukup besar.
- VPC dapat memiliki atribut penghunian instans, baik yang bersifat default atau khusus. Semua VPC default memiliki atribut penghunian instans yang ditetapkan ke default, dan VPC default dapat mendukung kelas instans DB mana pun.

  Jika Anda memilih untuk menempatkan klaster DB dalam VPC khusus tempat atribut penghunian instans ditetapkan ke khusus, maka kelas instans DB untuk klaster DB Anda harus merupakan salah satu jenis instans khusus Amazon EC2 yang disetujui. Misalnya, instans khusus EC2 r5.large sesuai dengan kelas instans DB db.r5.large. Untuk informasi tentang penghunian instans dalam VPC, lihat Instans khusus dalam Panduan Pengguna Amazon Elastic Compute Cloud.

  Untuk informasi selengkapnya tentang jenis instans yang dapat berada dalam instans khusus, lihat Instans khusus Amazon EC2 di halaman harga EC2.

  > **Note**
  >
  > Jika Anda menetapkan atribut penghunian instans ke khusus untuk klaster DB, hal tersebut tidak menjamin bahwa klaster DB akan berjalan di host khusus.

- Saat grup opsi ditetapkan ke instans DB, grup opsi tersebut dikaitkan dengan VPC instans DB. Penautan ini berarti Anda tidak dapat menggunakan grup opsi yang ditetapkan ke instans DB jika Anda mencoba untuk memulihkan instans DB ke VPC yang berbeda.
- Jika Anda memulihkan instans DB ke VPC yang berbeda, pastikan untuk menetapkan grup opsi default ke instans DB, menetapkan grup opsi yang ditautkan ke VPC tersebut, atau membuat grup opsi baru dan menetapkannya ke instans DB. Dengan opsi yang persisten atau permanen, seperti Oracle TDE, Anda harus membuat grup opsi baru yang mencakup opsi persisten atau permanen saat memulihkan instans DB ke VPC yang berbeda.

Bekerja dengan grup subnet DB

Subnet adalah segmen dari rentang alamat IP VPC yang Anda tetapkan untuk mengelompokkan sumber daya Anda berdasarkan kebutuhan keamanan dan operasional. Grup subnet DB adalah kumpulan subnet (biasanya privat) yang Anda buat dalam VPC dan kemudian Anda tetapkan untuk klaster DB Anda. Dengan menggunakan grup subnet DB, Anda dapat menentukan VPC tertentu saat membuat klaster menggunakan AWS CLI atau API RDS. Jika menggunakan konsol, Anda dapat memilih grup VPC dan subnet yang ingin Anda gunakan.

Setiap grup subnet DB harus memiliki subnet minimal di dua Zona Ketersediaan dalam Wilayah AWS tertentu. Saat membuat klaster DB dalam VPC, Anda memilih grup subnet DB untuk klaster tersebut. Dari grup subnet DB, Amazon RDS memilih subnet dan alamat IP dalam subnet tersebut untuk dikaitkan dengan utama. DB menggunakan Zona Ketersediaan yang berisi subnet.

Jika instans DB utama dalam deployment Multi-AZ gagal, Amazon RDS dapat mempromosikan fungsi siaga yang terkait dan kemudian membuat fungsi siaga baru menggunakan alamat IP subnet di salah satu Zona Ketersediaan lainnya.

Subnet dalam grup subnet DB bersifat publik atau privat. Subnet bersifat umum atau privat, bergantung pada konfigurasi yang Anda tetapkan untuk daftar kontrol akses jaringan (network ACL) dan tabel perutean. Agar klaster DB dapat diakses publik, semua subnet dalam grup subnet DB-nya harus bersifat publik. Jika subnet yang terkait dengan klaster DB yang dapat diakses publik berubah dari publik menjadi pribadi, hal tersebut dapat memengaruhi ketersediaan klaster DB.

Untuk membuat grup subnet DB yang mendukung mode dual-stack, pastikan setiap subnet yang Anda tambahkan ke grup subnet DB memiliki blok CIDR Internet Protocol versi 6 (IPv6) yang terkait dengannya. Untuk informasi selengkapnya, lihat Penentuan alamat IP Amazon RDS dan Bermigrasi ke IPv6 dalam Panduan Pengguna Amazon VPC.

> **Note**
>
> Grup subnet DB untuk Zona Lokal hanya boleh memiliki satu subnet.

Saat Amazon RDS membuat klaster DB dalam VPC, antarmuka jaringan ditetapkan ke klaster DB dengan menggunakan alamat IP dari grup subnet DB. Namun, kami sangat menyarankan agar Anda menggunakan nama Sistem Nama Domain (DNS) untuk terhubung ke klaster DB. Kami merekomendasikan hal ini karena alamat IP pokok berubah selama failover.

> **Note**
>
> Untuk setiap klaster DB yang Anda jalankan dalam VPC, pastikan untuk menyimpan minimal satu alamat di setiap subnet dalam grup subnet DB agar dapat digunakan oleh Amazon RDS untuk tindakan pemulihan.

Subnet bersama

Anda dapat membuat klaster DB dalam VPC bersama.

Beberapa pertimbangan yang perlu diingat saat menggunakan VPC bersama:

- Anda dapat memindahkan klaster DB dari subnet VPC bersama ke subnet VPC privat dan sebaliknya.
- Peserta dalam VPC bersama harus membuat grup keamanan dalam VPC agar mereka dapat membuat klaster DB.
- Pemilik dan peserta dalam VPC bersama dapat mengakses basis data dengan menggunakan kueri SQL. Namun, hanya pembuat sumber daya yang dapat melakukan panggilan API di sumber daya.

Penentuan alamat IP Amazon RDS

Penentuan alamat IP memungkinkan sumber daya dalam VPC Anda berkomunikasi satu sama lain, dan dengan sumber daya melalui internet. Amazon RDS mendukung protokol penentuan alamat IPv4 dan IPv6. Secara default, Amazon RDS dan Amazon VPC menggunakan protokol penentuan alamat IPv4. Anda tidak dapat menonaktifkan perilaku ini. Saat Anda membuat VPC, pastikan untuk menentukan blok CIDR IPv4 (rentang alamat IPv4 privat). Atau, Anda dapat menetapkan blok CIDR IPv6 ke VPC dan subnet Anda, serta menetapkan alamat IPv6 dari blok tersebut ke klaster DB di subnet Anda.

Dukungan untuk protokol IPv6 memperbesar jumlah alamat IP yang didukung. Dengan menggunakan protokol IPv6, Anda memastikan bahwa Anda memiliki ketersediaan alamat yang memadai untuk menghadapi pertumbuhan internet di masa mendatang. Sumber daya RDS baru dan yang sudah ada dapat menggunakan alamat IPv4 dan IPv6 dalam VPC Anda. Mengonfigurasi, mengamankan, dan menerjemahkan lalu lintas jaringan di antara dua protokol yang digunakan di berbagai bagian aplikasi dapat menimbulkan overhead operasional. Anda dapat melakukan

standardisasi pada protokol IPv6 bagi sumber daya Amazon RDS untuk menyederhanakan konfigurasi jaringan Anda.

Topik

- Alamat IPv4
- Alamat IPv6
- Mode dual-stack

Alamat IPv4

Saat membuat VPC, Anda harus menentukan rentang alamat IPv4 untuk VPC dalam bentuk blok CIDR, seperti `10.0.0.0/16`. Grup subnet DB mendefinisikan rentang alamat IP di blok CIDR ini yang dapat digunakan oleh klaster DB. Alamat IP ini bisa bersifat privat atau publik.

Alamat IPv4 privat adalah alamat IP yang tidak dapat diakses melalui internet. Anda dapat menggunakan alamat IPv4 privat untuk komunikasi di antara klaster DB Anda dan sumber daya lainnya, seperti instans Amazon EC2, dalam VPC yang sama. Setiap klaster DB memiliki alamat IP privat untuk komunikasi dalam VPC.

Alamat IP publik adalah alamat IPv4 yang dapat diakses dari internet. Anda dapat menggunakan alamat publik untuk komunikasi di antara klaster DB dan sumber daya di internet, seperti klien SQL. Anda mengontrol apakah klaster DB Anda menerima alamat IP publik.

Untuk tutorial cara membuat VPC hanya dengan alamat IPv4 privat yang dapat Anda gunakan untuk skenario Amazon RDS yang umum, lihat Tutorial: Membuat VPC untuk digunakan dengan instans DB (khusus IPv4).

Alamat IPv6

Atau, Anda dapat memilih mengaitkan blok CIDR IPv6 ke VPC dan subnet, serta menetapkan alamat IPv6 dari blok tersebut ke sumber daya dalam VPC Anda. Setiap alamat IPv6 bersifat unik secara global.

Blok CIDR IPv6 untuk VPC Anda secara otomatis ditetapkan dari kumpulan alamat IPv6 Amazon. Anda tidak dapat memilih sendiri rentang tersebut.

Saat menghubungkan ke alamat IPv6, pastikan kondisi berikut terpenuhi:

- Klien dikonfigurasi, sehingga lalu lintas klien ke basis data melalui IPv6 dimungkinkan.

- Grup keamanan RDS yang digunakan oleh instans DB dikonfigurasi dengan benar, sehingga lalu lintas klien ke basis data melalui IPv6 dimungkinkan.
- Stack sistem operasi klien memungkinkan lalu lintas pada alamat IPv6, dan driver serta pustaka sistem operasi dikonfigurasi untuk memilih titik akhir instans DB default yang benar (yaitu, IPv4 atau IPv6).

Untuk informasi selengkapnya tentang IPv6, lihat Penentuan alamat IP di Panduan Pengguna Amazon VPC.

Mode dual-stack

Ketika klaster DB dapat berkomunikasi melalui protokol penentuan alamat IPv4 dan IPv6, mode dual-stack berarti digunakan. Jadi, sumber daya dapat berkomunikasi dengan klaster DB melalui IPv4, IPv6, atau keduanya. RDS menonaktifkan akses Gateway Internet untuk titik akhir IPv6 dari instans DB mode dual-stack privat. RDS melakukan hal ini untuk memastikan titik akhir IPv6 Anda bersifat privat dan hanya dapat diakses dari dalam VPC Anda.

Topik

- Mode dual-stack dan grup subnet DB
- Bekerja dengan instans DB mode dual-stack
- Memodifikasi klaster DB khusus IPv4 untuk menggunakan mode dual-stack
- Ketersediaan versi dan Wilayah
- Batasan untuk klaster DB jaringan dual-stack

Untuk tutorial cara membuat VPC dengan alamat IPv4 dan IPv6 yang dapat Anda gunakan untuk skenario Amazon RDS yang umum, lihat Tutorial: Membuat VPC untuk digunakan dengan instans DB (mode dual-stack).

Mode dual-stack dan grup subnet DB

Untuk menggunakan mode dual-stack, pastikan setiap subnet dalam grup subnet DB yang Anda kaitkan dengan klaster DB memiliki blok CIDR IPv6 yang terkait dengannya. Anda dapat membuat grup subnet DB baru atau memodifikasi grup subnet DB yang ada untuk memenuhi persyaratan ini. Setelah klaster DB berada dalam mode dual-stack, klien dapat terhubung secara normal. Pastikan firewall keamanan klien dan grup keamanan instans RDS DB dikonfigurasi secara akurat untuk memungkinkan lalu lintas melalui IPv6. Untuk terhubung, klien menggunakan titik akhir instans DB. Aplikasi klien dapat menentukan protokol mana yang lebih disukai saat terhubung ke basis data.

Dalam mode dual-stack, klaster DB mendeteksi protokol jaringan pilihan klien, baik IPv4 maupun IPv6, dan menggunakan protokol tersebut untuk koneksi.

Jika grup subnet DB berhenti mendukung mode dual-stack karena penghapusan subnet atau pembatalan pengaitan CIDR, ada risiko status jaringan menjadi tidak kompatibel untuk instans DB yang terkait dengan grup subnet DB. Selain itu, Anda tidak dapat menggunakan grup subnet DB saat membuat klaster DB mode dual-stack yang baru.

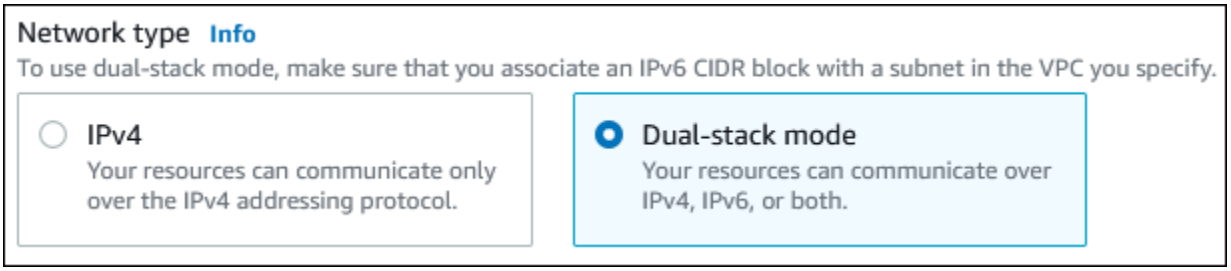
Untuk menentukan apakah grup subnet DB mendukung mode dual-stack dengan menggunakan AWS Management Console, lihat Jenis jaringan pada halaman detail grup subnet DB. Untuk menentukan apakah grup subnet DB mendukung mode dual-stack dengan menggunakanAWS CLI, jalankan describe-db-subnet-groupsperintah dan lihat `SupportedNetworkTypes` di output.

Replika baca dianggap sebagai instans DB independen dan dapat memiliki jenis jaringan yang berbeda dengan instans DB utama. Jika Anda mengubah jenis jaringan untuk instans DB utama replika baca, replika baca tidak akan terpengaruh. Saat memulihkan instans DB, Anda dapat memulihkannya ke jenis jaringan apa pun yang didukung.

Bekerja dengan instans DB mode dual-stack

Saat membuat atau memodifikasi klaster DB, Anda dapat menentukan mode dual-stack agar sumber daya Anda dapat berkomunikasi dengan klaster DB melalui IPv4, IPv6, atau keduanya.

Jika Anda menggunakan AWS Management Console untuk membuat atau memodifikasi instans DB, Anda dapat menentukan mode dual-stack di bagian Jenis jaringan. Gambar berikut menampilkan bagian Jenis jaringan di konsol.



Saat Anda menggunakan AWS CLI untuk membuat atau memodifikasi klaster DB, tetapkan opsi `--network-type` ke `DUAL` untuk menggunakan mode dual-stack. Saat Anda menggunakan API RDS untuk membuat atau memodifikasi klaster DB, tetapkan parameter `NetworkType` ke `DUAL` untuk menggunakan mode dual-stack. Saat Anda memodifikasi jenis jaringan instans DB, mungkin terjadi waktu henti. Jika mode dual-stack tidak didukung oleh versi mesin DB atau grup subnet DB yang ditentukan, kesalahan `NetworkTypeNotSupported` akan ditampilkan.

Untuk informasi selengkapnya tentang cara membuat instans DB, lihat Membuat instans DB Amazon RDS. Untuk informasi selengkapnya tentang cara memodifikasi instans DB, lihat Memodifikasi instans DB Amazon RDS.

Untuk menentukan apakah klaster DB berada di mode dual-stack dengan menggunakan konsol, lihat Jenis jaringan di tab Konektivitas & keamanan untuk klaster DB.

Memodifikasi klaster DB khusus IPv4 untuk menggunakan mode dual-stack

Anda dapat memodifikasi klaster DB khusus IPv4 untuk menggunakan mode dual-stack. Untuk melakukannya, ubah jenis jaringan klaster DB. Modifikasi dapat mengakibatkan waktu henti.

Sebaiknya Anda mengubah jenis jaringan instans DB Amazon RDS selama periode pemeliharaan. Saat ini, pengaturan jenis jaringan instans baru ke mode dual-stack tidak didukung. Anda dapat mengatur jenis jaringan secara manual dengan menggunakan perintah `modify-db-instance` .

Sebelum memodifikasi klaster DB untuk menggunakan mode dual-stack, pastikan grup subnet DB-nya mendukung mode dual-stack. Jika grup subnet DB yang terkait dengan klaster DB tidak mendukung mode dual-stack, tentukan grup subnet DB yang berbeda yang mendukungnya saat Anda memodifikasi klaster DB. Memodifikasi grup subnet DB dari klaster DB dapat menyebabkan waktu henti.

Jika Anda memodifikasi grup subnet DB dari klaster DB sebelum mengubah klaster DB untuk menggunakan mode dual-stack, pastikan grup subnet DB valid untuk klaster DB sebelum dan sesudah perubahan.

Untuk RDS untuk PostgreSQL, RDS untuk MySQL, RDS untuk Oracle, dan RDS untuk instance Single-AZ MariaDB, sebaiknya Anda menjalankan perintah hanya dengan parameter yang disetel untuk mengubah jaringan ke mode dual-stack. modify-db-instance`--network-type`DUAL Menambahkan parameter lain bersama dengan parameter `--network-type` dalam panggilan API yang sama dapat mengakibatkan waktu henti. Untuk memodifikasi beberapa parameter, pastikan modifikasi jenis jaringan berhasil diselesaikan sebelum mengirim permintaan `modify-db-instance` lainnya dengan parameter lain.

Modifikasi tipe jaringan untuk RDS untuk PostgreSQL, RDS untuk MySQL, RDS untuk Oracle, dan RDS untuk instance DB Multi-AZ MariaDB menyebabkan downtime singkat dan memicu failover jika Anda hanya menggunakan parameter atau jika Anda menggabungkan parameter dalam perintah. `--network-type` modify-db-instance

Modifikasi jenis jaringan pada instans DB Multi-AZ atau AZ-Tunggal RDS for SQL Server menyebabkan waktu henti jika Anda hanya menggunakan parameter `--network-type` atau jika

Anda menggabungkan parameter dalam perintah `modify-db-instance`. Modifikasi jenis jaringan menyebabkan failover dalam instans Multi-AZ SQL Server.

Jika Anda tidak dapat terhubung ke klaster DB setelah perubahan, pastikan firewall keamanan klien dan basis data serta tabel rute dikonfigurasi secara akurat agar lalu lintas ke basis data dimungkinkan pada jaringan yang dipilih (baik IPv4 maupun IPv6). Anda mungkin juga perlu memodifikasi parameter sistem operasi, pustaka, atau driver untuk terhubung menggunakan alamat IPv6.

Saat Anda memodifikasi instans DB untuk menggunakan mode dual-stack, tidak akan ada perubahan yang tertunda dari deployment AZ-Tunggal ke deployment Multi-AZ, atau dari deployment Multi-AZ ke deployment AZ-Tunggal.

Anda dapat memodifikasi klaster DB khusus IPv4 untuk menggunakan mode dual-stack

1. Modifikasi grup subnet DB untuk mendukung mode dual-stack, atau buat grup subnet DB yang mendukung mode dual-stack:

   a. Kaitkan blok CIDR IPv6 dengan VPC Anda.

      Untuk mendapatkan petunjuk, lihat Menambahkan blok CIDR IPv6 ke VPC di Panduan Pengguna Amazon VPC.

   b. Tambahkan blok CIDR IPv6 ke semua subnet di grup subnet DB Anda.

      Untuk mendapatkan petunjuk, lihat Menambahkan blok CIDR IPv6 ke subnet di Panduan Pengguna Amazon VPC.

   c. Konfirmasikan bahwa grup subnet DB mendukung mode dual-stack.

      Jika Anda menggunakan AWS Management Console, pilih grup subnet DB, dan pastikan bahwa nilai Jenis jaringan yang didukung adalah Dual, IPv4.

      Jika Anda menggunakanAWS CLI, jalankan describe-db-subnet-groupsperintah, dan pastikan bahwa `SupportedNetworkType` nilai untuk instans DB adalah`Dual, IPv4`.

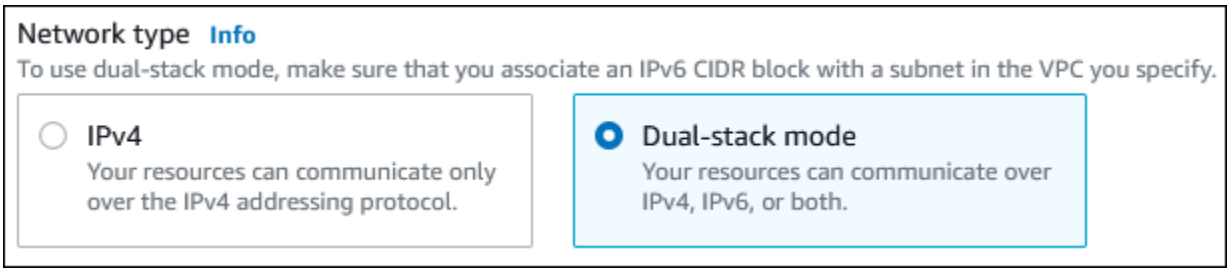
2. Modifikasi grup keamanan yang terkait dengan klaster DB agar IPv6 dapat dihubungkan ke basis data, atau buat grup keamanan baru agar IPv6 dapat terhubung.

   Untuk mendapatkan petunjuk, lihat Aturan grup keamanan di Panduan Pengguna Amazon VPC.

3. Modifikasi klaster DB untuk mendukung mode dual-stack. Untuk melakukannya, tetapkan Jenis jaringan ke Mode dual-stack.

   Jika Anda menggunakan konsol, pastikan pengaturan berikut sudah benar:

- Jenis jaringan – Mode dual-stack



- Grup subnet DB – Grup subnet DB yang telah Anda konfigurasikan pada langkah sebelumnya
- Grup keamanan — Keamanan yang telah Anda konfigurasikan pada langkah sebelumnya

Jika Anda menggunakan AWS CLI, pastikan pengaturan berikut sudah benar:

- `--network-type` – `dual`
- `--db-subnet-group-name` – Grup subnet DB yang telah Anda konfigurasikan pada langkah sebelumnya
- `--vpc-security-group-ids` – Grup keamanan VPC yang telah Anda konfigurasikan pada langkah sebelumnya

Contoh:

```
aws rds modify-db-instance --db-instance-identifier my-instance --network-type
 "DUAL"
```

4. Konfirmasikan bahwa klaster DB mendukung mode dual-stack.

   Jika Anda menggunakan konsol, pilih tab Konektivitas & keamanan untuk klaster DB. Pada tab tersebut, pastikan nilai Jenis jaringan adalah Mode dual-stack.

   Jika Anda menggunakanAWS CLI, jalankan describe-db-instancesperintah, dan pastikan bahwa `NetworkType` nilai untuk instans DB adalah`dual`.

   Jalankan perintah `dig` pada titik akhir instans DB untuk mengidentifikasi alamat IPv6 yang terkait dengannya.

```
dig db-instance-endpoint AAAA
```

Gunakan titik akhir instans DB , bukan alamat IPv6, untuk terhubung ke klaster DB.

Ketersediaan versi dan Wilayah

Ketersediaan dan dukungan fitur bervariasi di berbagai versi khusus dari setiap mesin basis data, dan di seluruh Wilayah AWS. Untuk informasi selengkapnya tentang ketersediaan versi dan Wilayah dengan mode dual-stack, lihat Mode tumpukan ganda.

Batasan untuk klaster DB jaringan dual-stack

Batasan berikut berlaku untuk klaster DB jaringan dual-stack:

- Klaster DB tidak dapat menggunakan protokol IPv6 secara eksklusif. Klaster tersebut dapat menggunakan IPv4 secara eksklusif, atau menggunakan protokol IPv4 dan IPv6 (mode dual-stack).
- Amazon RDS tidak mendukung subnet IPv6 asli.
- Klaster DB yang menggunakan mode dual-stack harus privat. Klaster tersebut tidak dapat diakses publik.
- Mode dual-stack tidak mendukung kelas instans DB db.m3 dan db.r3.
- Untuk RDS for Server SQL, instans DB mode dual-stack yang menggunakan titik akhir pendengar grup ketersediaan Always On AG hanya menampilkan alamat IPv4.
- Anda tidak dapat menggunakan Proksi RDS dengan klaster DB mode dual-stack.
- Anda tidak dapat menggunakan mode dual-stack dengan instans DB RDS di AWS Outposts.
- Anda tidak dapat menggunakan mode dual-stack dengan instans DB di Zona Lokal.

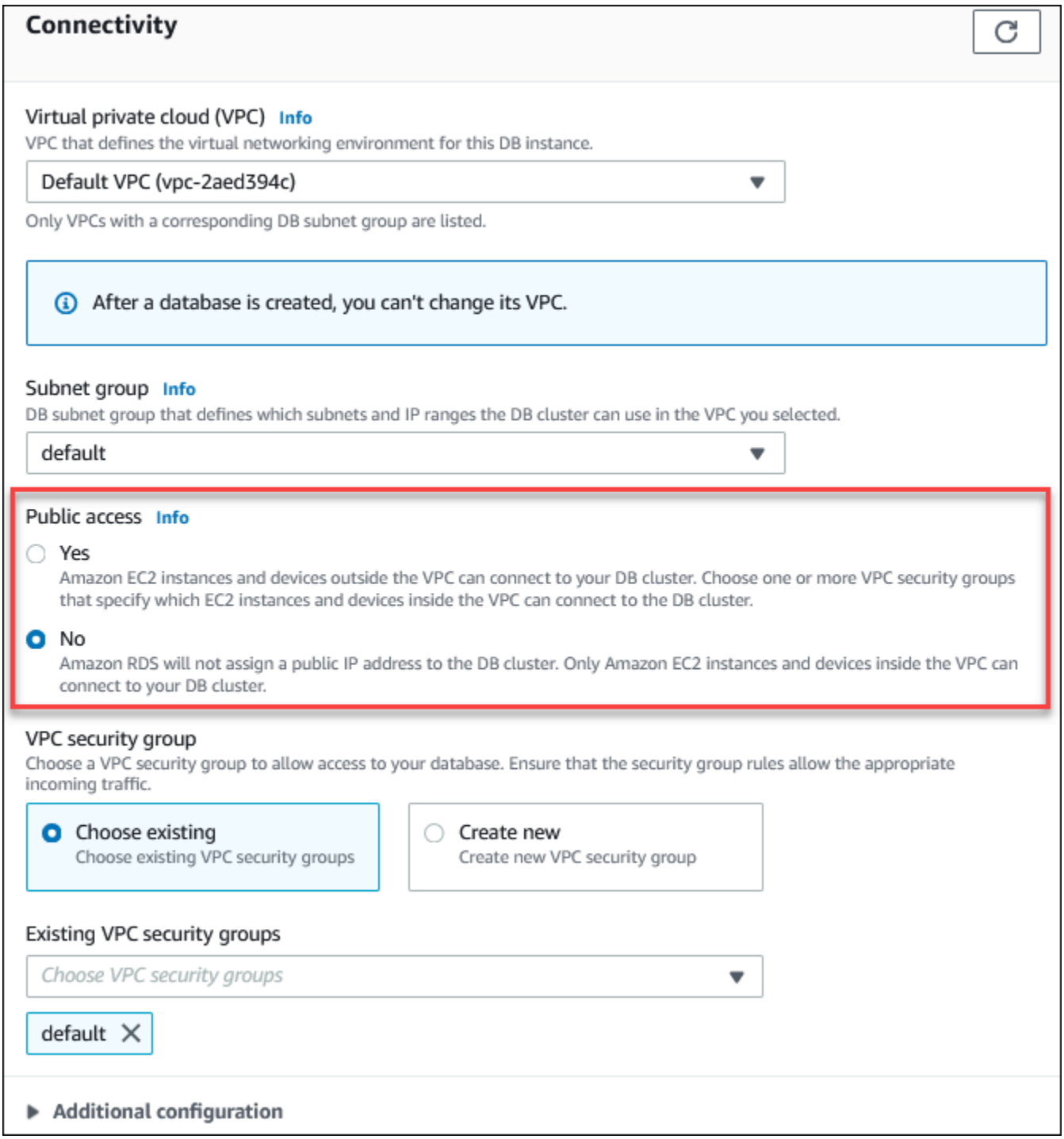
Menyembunyikan klaster DB dalam VPC dari internet

Satu skenario Amazon RDS yang umum adalah memiliki VPC yang di dalamnya ada instans EC2 dengan aplikasi web untuk publik dan klaster DB dengan basis data yang tidak dapat diakses publik. Misalnya, Anda dapat membuat VPC yang memiliki subnet publik dan subnet privat. Instans Amazon EC2 yang berfungsi sebagai server web dapat di-deploy di subnet publik. Klaster DB di-deploy di subnet privat. Dalam deployment seperti itu, hanya server web yang memiliki akses ke klaster DB. Untuk ilustrasi skenario ini, lihat Instans DB dalam VPC yang diakses oleh instans EC2 dalam VPC yang sama.

Saat Anda meluncurkan klaster DB di dalam VPC, klaster DB memiliki alamat IP privat untuk lalu lintas di dalam VPC. Alamat IP privat ini tidak dapat diakses publik. Anda dapat menggunakan opsi

Akses publik untuk menetapkan apakah klaster DB juga memiliki alamat IP publik selain alamat IP privat. Jika klaster DB ditetapkan sebagai dapat diakses publik, titik akhir DNS-nya akan diselesaikan ke alamat IP privat dari dalam VPC. Serta ke alamat IP publik dari luar VPC. Akses ke klaster DB pada akhirnya dikontrol oleh grup keamanan yang digunakannya. Akses publik tersebut tidak diizinkan jika grup keamanan yang ditetapkan ke klaster DB tidak mencakup aturan masuk yang mengizinkannya. Selain itu, agar klaster DB dapat diakses publik, subnet dalam grup subnet DB-nya harus memiliki gateway internet. Lihat informasi yang lebih lengkap di Tidak dapat terhubung ke instans DB Amazon RDS

Anda dapat memodifikasi klaster DB untuk mengaktifkan atau menonaktifkan aksesibilitas publik dengan memodifikasi opsi Akses publik. Ilustrasi berikut ini menunjukkan opsi Akses publik di bagian Konfigurasi konektivitas tambahan. Untuk menetapkan opsi tersebut, buka bagian Konfigurasi konektivitas tambahan di bagian Konektivitas.



Untuk informasi tentang cara memodifikasi instans DB untuk menetapkan opsi Akses publik, lihat Memodifikasi instans DB Amazon RDS.

Membuat klaster DB dalam VPC

Prosedur berikut membantu Anda membuat klaster DB dalam VPC. Untuk menggunakan VPC default, Anda dapat memulai dengan langkah 2, dan menggunakan grup subnet VPC dan DB yang sudah dibuat untuk Anda. Jika ingin membuat VPC tambahan, Anda dapat membuat VPC baru.

> **Note**
>
> Jika Anda ingin klaster DB dalam VPC dapat diakses publik, Anda harus memperbarui informasi DNS untuk VPC dengan mengaktifkan nama host DNS dan resolusi DNS atribut VPC. Untuk informasi tentang cara memperbarui informasi DNS untuk instans VPC, lihat Memperbarui dukungan DNS untuk VPC Anda.

Ikuti langkah-langkah berikut untuk membuat instans DB dalam VPC:

- Langkah 1: Buat VPC
- Langkah 2: Buat grup subnet DB
- Langkah 3: Buat grup keamanan VPC
- Langkah 4: Buat instans DB dalam VPC

Langkah 1: Buat VPC

Buat VPC dengan subnet di minimal dua Zona Ketersediaan. Anda menggunakan subnet ini saat membuat grup subnet DB. Jika Anda memiliki VPC default, subnet secara otomatis dibuat untuk Anda dalam setiap Zona Ketersediaan di Wilayah AWS.

Untuk informasi selengkapnya, lihat Membuat VPC dengan subnet publik dan privat, atau lihat Membuat VPC dalam Panduan Pengguna Amazon VPC.

Langkah 2: Buat grup subnet DB

Grup subnet DB adalah kumpulan subnet (biasanya privat) yang Anda buat untuk VPC dan kemudian ditetapkan untuk klaster DB Anda. Grup subnet DB memungkinkan Anda menetapkan VPC tertentu saat Anda membuat klaster DB menggunakan AWS CLI atau API RDS. Jika menggunakan konsol, Anda dapat memilih grup VPC dan subnet yang ingin digunakan. Setiap grup subnet DB harus memiliki minimal satu subnet di minimal dua Zona Ketersediaan di Wilayah AWS. Sebagai praktik

terbaik, setiap grup subnet DB harus memiliki minimal satu subnet untuk setiap Zona Ketersediaan di Wilayah AWS.

Untuk deployment Multi-AZ, menentukan subnet untuk semua Zona Ketersediaan di Wilayah AWS memungkinkan Amazon RDS membuat replika siaga baru di Zona Ketersediaan lain jika diperlukan. Anda dapat mengikuti praktik terbaik ini bahkan untuk deployment AZ-Tunggal, karena Anda mungkin mengonversinya ke deployment Multi-AZ di masa mendatang.

Agar klaster DB dapat diakses publik, subnet dalam grup subnet DB harus memiliki gateway internet. Untuk informasi selengkapnya tentang gateway internet untuk subnet, lihat Terhubung ke internet menggunakan gateway internet dalam Panduan Pengguna Amazon VPC.

> **Note**
>
> Grup subnet DB untuk Zona Lokal hanya boleh memiliki satu subnet.

Saat membuat klaster DB dalam VPC, Anda dapat memilih grup subnet DB. Amazon RDS memilih subnet dan alamat IP dalam subnet tersebut untuk dikaitkan dengan klaster DB Anda. Jika tidak ada grup subnet DB, Amazon RDS membuat grup subnet default saat Anda membuat klaster DB. Amazon RDS membuat dan mengaitkan Antarmuka Jaringan Elastis ke klaster DB Anda dengan alamat IP tersebut. Klaster DB menggunakan Zona Ketersediaan yang berisi subnet.

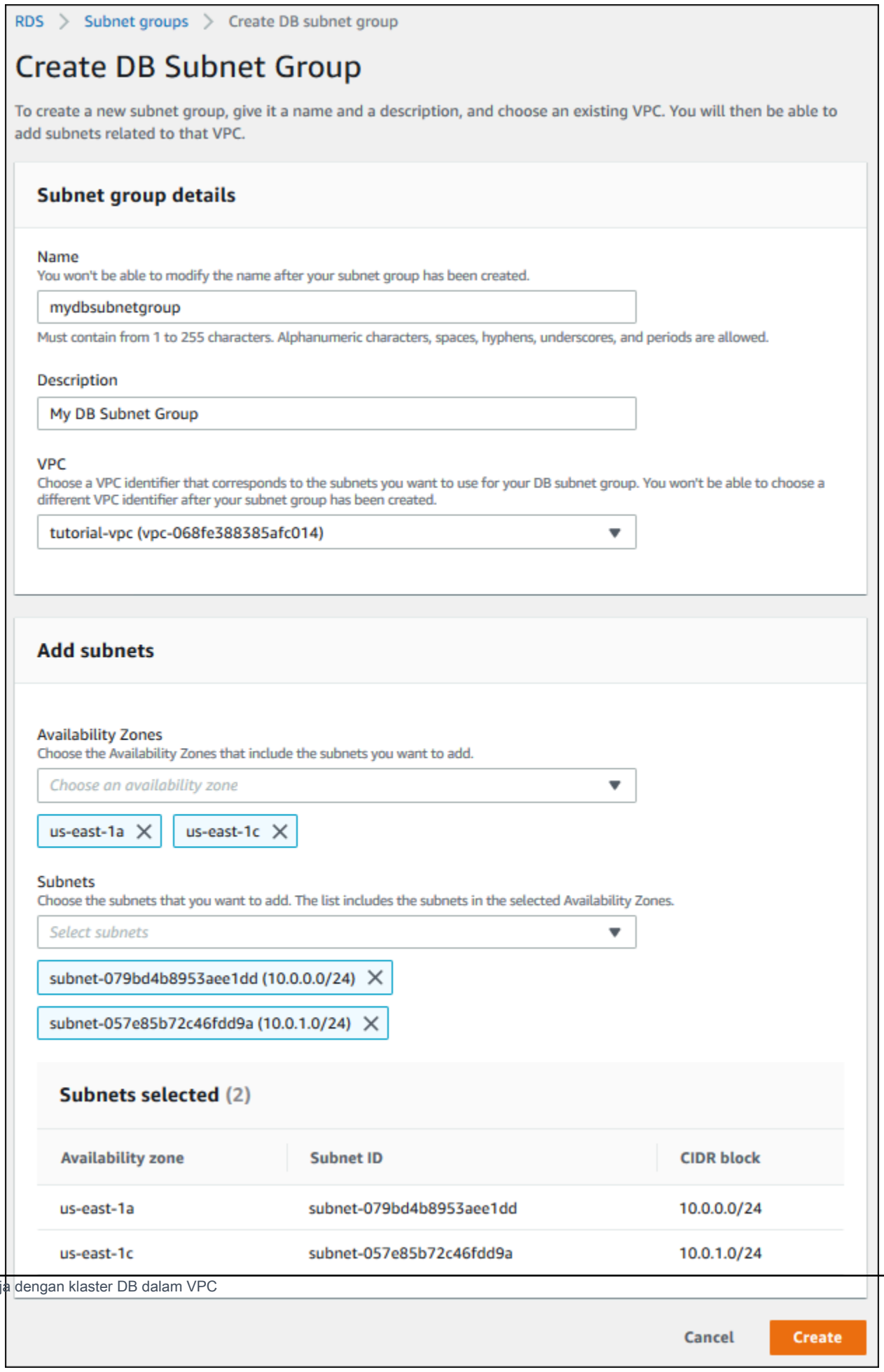
Untuk deployment Multi-AZ, menentukan subnet untuk dua atau beberapa Zona Ketersediaan di Wilayah AWS memungkinkan Amazon RDS membuat fungsi siaga baru di Zona Ketersediaan lain jika diperlukan. Anda harus melakukan hal ini bahkan untuk deployment AZ-Tunggal, jika nantinya Anda ingin mengonversinya ke deployment Multi-AZ.

Pada langkah ini, Anda membuat grup subnet DB dan menambahkan subnet yang telah dibuat untuk VPC Anda.

Untuk membuat grup subnet DB

1. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di panel navigasi, pilih Grup subnet.
3. Pilih Buat Grup Subnet DB.
4. Untuk Nama, ketik nama grup subnet DB Anda.
5. Untuk Deskripsi, ketik deskripsi untuk grup subnet DB Anda.

6. Untuk VPC, pilih VPC default atau VPC yang Anda buat.
7. Di bagian Tambahkan subnet, pilih Zona Ketersediaan yang mencakup subnet dari Zona Ketersediaan, lalu pilih subnet dari Subnet.


RDS > Subnet groups > Create DB subnet group
Create DB Subnet Group
To create a new subnet group, give it a name and a description, and choose an existing VPC. You will then be able to add subnets related to that VPC.
Subnet group details
Name
You won't be able to modify the name after your subnet group has been created.
mydbsubnetgroup
Must contain from 1 to 255 characters. Alphanumeric characters, spaces, hyphens, underscores, and periods are allowed.
Description
My DB Subnet Group
VPC
Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be able to choose a different VPC identifier after your subnet group has been created.
tutorial-vpc (vpc-068fe388385afc014)
Add subnets
Availability Zones
Choose the Availability Zones that include the subnets you want to add.
Choose an availability zone
us-east-1a
us-east-1c
Subnets
Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.
Select subnets
subnet-079bd4b8953aee1dd (10.0.0.0/24)
subnet-057e85b72c46fdd9a (10.0.1.0/24)
Subnets selected (2)
Availability zone
Subnet ID
CIDR block
us-east-1a
subnet-079bd4b8953aee1dd
10.0.0.0/24
us-east-1c
subnet-057e85b72c46fdd9a
10.0.1.0/24
Cancel
Create

> **Note**
>
> Jika telah mengaktifkan Zona Lokal, Anda dapat memilih grup Zona Ketersediaan di halaman Membuat grup subnet DB. Dalam kasus ini, pilih Grup Zona Ketersediaan, Zona Ketersediaan, dan Subnet.

8. Pilih Buat.

   Grup subnet DB baru Anda akan muncul dalam daftar grup subnet DB di konsol RDS. Anda dapat memilih grup subnet DB untuk melihat detail, termasuk semua subnet yang dikaitkan dengan grup, dalam panel detail di bagian bawah jendela.

Langkah 3: Buat grup keamanan VPC

Sebelum membuat klaster DB, Anda dapat membuat grup keamanan VPC untuk dikaitkan dengan klaster DB. Jika belum membuat grup keamanan VPC, Anda dapat menggunakan grup keamanan default saat membuat klaster DB. Untuk petunjuk cara membuat grup keamanan untuk klaster DB, lihat Membuat grup keamanan VPC untuk instans DB privat, atau lihat Mengontrol lalu lintas ke sumber daya menggunakan grup keamanan dalam Panduan Pengguna Amazon VPC.

Langkah 4: Buat instans DB dalam VPC

Pada langkah ini, Anda membuat klaster DB dan menggunakan nama VPC, grup subnet DB, dan grup keamanan VPC yang telah Anda buat pada langkah sebelumnya.

> **Note**
>
> Jika ingin klaster DB dalam VPC dapat diakses publik, Anda harus mengaktifkan nama host DNS dan resolusi DNS atribut VPC. Untuk informasi selengkapnya, lihat Atribut DNS untuk VPC Anda dalam Panduan Pengguna Amazon VPC.

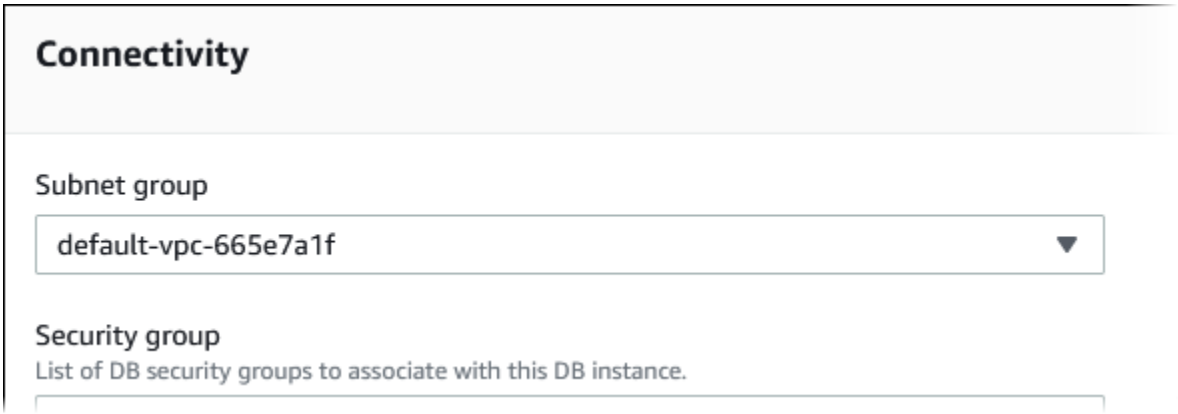
Untuk detail tentang cara membuat klaster DB, lihat Membuat instans DB Amazon RDS.

Saat diminta di bagian Konektivitas, masukkan nama VPC, grup subnet DB, dan grup keamanan VPC.

Memperbarui VPC untuk instans DB

Anda dapat menggunakan AWS Management Console untuk memindahkan instans DB Anda ke VPC yang berbeda.

Untuk informasi tentang mengubah instans DB, lihat Memodifikasi instans DB Amazon RDS. Pada bagian Konektivitas di halaman modifikasi yang ditampilkan berikut ini, masukkan grup subnet DB baru untuk Grup subnet DB. Grup subnet baru harus merupakan grup subnet dalam VPC baru.



Anda tidak dapat mengubah VPC instans DB jika kondisi berikut ini berlaku:

- Instans DB ada di beberapa Zona Ketersediaan. Anda dapat mengonversi instans DB ke Zona Ketersediaan tunggal, memindahkannya ke VPC baru, lalu mengubahnya kembali ke instans DB Multi-AZ. Untuk informasi selengkapnya, lihat Mengonfigurasi dan mengelola deployment Multi-AZ.
- Instans DB memiliki satu atau beberapa replika baca. Anda dapat menghapus replika baca, memindahkan instans DB ke VPC baru, lalu menambahkan replika baca lagi. Untuk informasi selengkapnya, lihat Menggunakan replika baca instans DB.
- Instans DB adalah replika baca. Anda dapat mempromosikan replika baca, lalu memindahkan instans DB mandiri ke VPC baru. Untuk informasi selengkapnya, lihat Mempromosikan replika baca menjadi instans DB mandiri.
- Grup subnet dalam VPC target tidak memiliki subnet di Zona Ketersediaan instans DB. Anda dapat menambahkan subnet di Zona Ketersediaan instans DB ke grup subnet DB, lalu memindahkan instans DB ke VPC baru. Untuk informasi selengkapnya, lihat Bekerja dengan grup subnet DB.

Skenario untuk mengakses instans DB di VPC

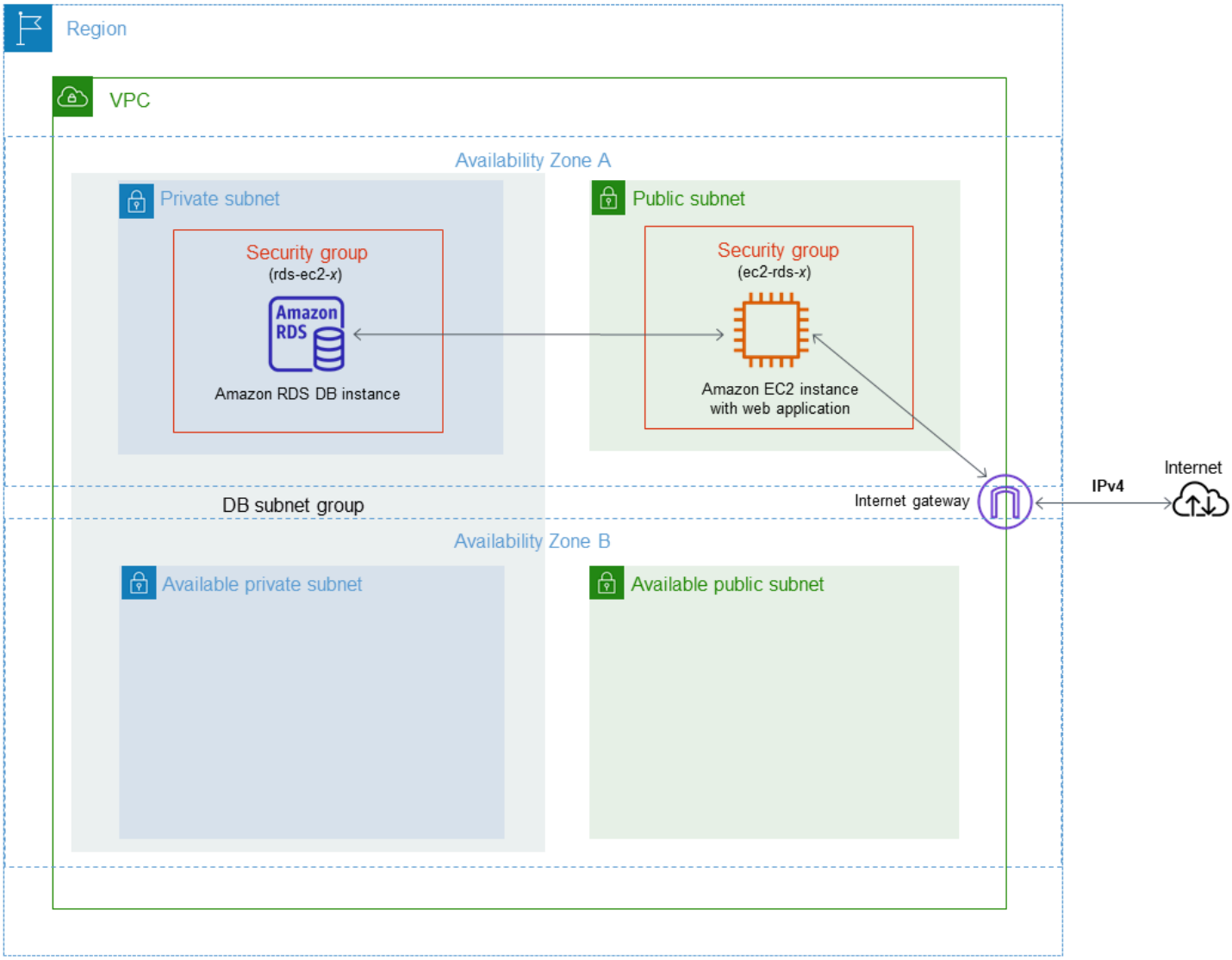
Amazon RDS Aurora mendukung skenario berikut untuk mengakses instans DB di VPC:

- Instans EC2 dalam VPC yang sama
- Instans EC2 dalam VPC yang berbeda
- Aplikasi klien melalui internet
- Jaringan privat

Instans DB dalam VPC yang diakses oleh instans EC2 dalam VPC yang sama

Penggunaan umum instans DB dalam VPC adalah untuk berbagi data dengan server aplikasi yang dijalankan di instans EC2 dalam VPC yang sama.

Diagram berikut menunjukkan skenario ini.



Cara paling sederhana untuk mengelola akses antara instans EC2 dan instans DB di VPC yang sama adalah dengan melakukan tindakan berikut:

- Buat grup keamanan VPC untuk klaster DB Anda. Grup keamanan ini dapat digunakan untuk membatasi akses ke instans DB. Misalnya, Anda dapat membuat aturan kustom untuk grup keamanan ini. Hal ini dapat mengizinkan akses TCP menggunakan port yang Anda tetapkan ke instans DB saat Anda membuatnya dan alamat IP yang Anda gunakan untuk mengakses instans DB untuk pengembangan atau tujuan lainnya.
- Buat grup keamanan VPC untuk memasukkan instans EC2 (server web dan klien) Anda. Grup keamanan ini dapat, jika diperlukan, mengizinkan akses ke instans EC2 dari internet dengan menggunakan tabel perutean VPC. Sebagai contoh, Anda dapat menetapkan aturan pada grup keamanan ini untuk mengizinkan akses TCP ke instans EC2 melalui port 22.
- Buat aturan kustom di grup keamanan untuk klaster DB Anda yang memungkinkan koneksi dari grup keamanan yang Anda buat untuk instans EC2. Aturan ini dapat memungkinkan anggota grup keamanan untuk mengakses klaster DB.

Ada subnet publik dan privat tambahan di Zona Ketersediaan terpisah. Grup subnet DB RDS membutuhkan subnet dalam setidaknya dua Zona Ketersediaan. Subnet tambahan memudahkan Anda untuk beralih ke deployment instans DB Multi-AZ di masa mendatang.

Untuk tutorial yang menunjukkan cara membuat VPC dengan subnet publik dan privat untuk skenario ini, lihat Tutorial: Membuat VPC untuk digunakan dengan instans DB (khusus IPv4).

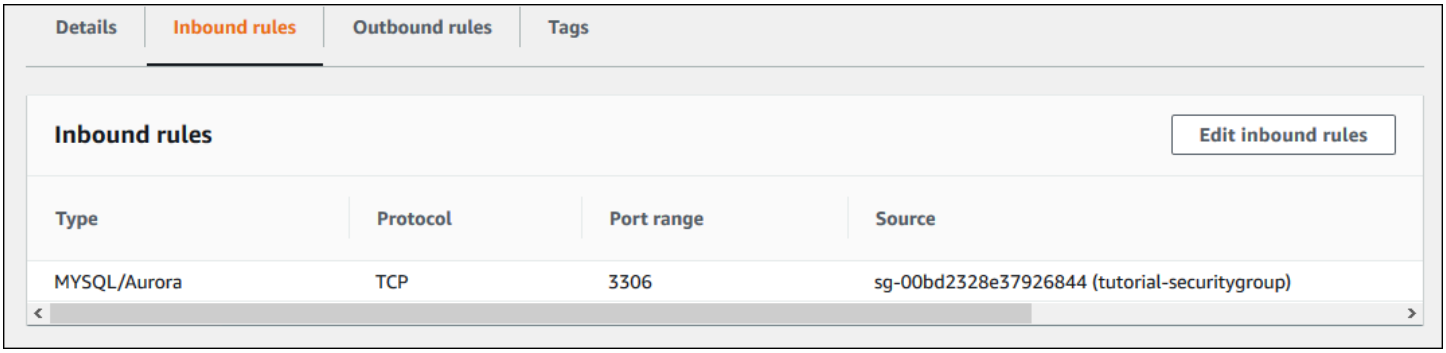
> **Tip**
>
> Anda dapat menyiapkan konektivitas jaringan antara instans Amazon EC2 dan instans DB secara otomatis saat membuat instans DB. Untuk informasi selengkapnya, lihat .

Untuk membuat aturan dalam grup keamanan VPC yang memungkinkan koneksi dari grup keamanan lain, lakukan hal berikut:

1. Masuk ke AWS Management Console lalu buka konsol Amazon VPC di https://console.aws.amazon.com/vpc.
2. Pada panel navigasi, pilih Grup keamanan.

3. Pilih atau buat grup keamanan yang ingin Anda berikan aksesnya ke anggota grup keamanan lain. Dalam skenario sebelumnya, ini adalah grup keamanan yang Anda gunakan untuk instans DB Anda. Pilih tab Aturan masuk, lalu pilih Edit aturan masuk.
4. Di halaman Edit aturan masuk, pilih Tambahkan aturan.
5. Untuk Jenis, pilih entri yang sesuai dengan port yang Anda gunakan saat membuat instans DB Anda, seperti MySQL/Aurora.
6. Di kotak Sumber, mulai ketikkan ID dari grup keamanan, sehingga akan menampilkan daftar grup keamanan yang sesuai. Pilih grup keamanan dengan anggota yang ingin diberi akses ke sumber daya yang dilindungi oleh grup keamanan ini. Dalam skenario sebelumnya, ini adalah grup keamanan yang Anda gunakan untuk instans EC2 Anda.
7. Jika perlu, ulangi langkah-langkah untuk protokol TCP dengan membuat aturan Semua TCP sebagai Jenis dan grup keamanan Anda di kotak Sumber. Jika Anda ingin menggunakan protokol UDP, buat aturan dengan Semua UDP sebagai Jenis dan grup keamanan Anda di Sumber.
8. Pilih Simpan aturan.

Layar berikut menunjukkan aturan masuk dengan grup keamanan sebagai sumbernya.

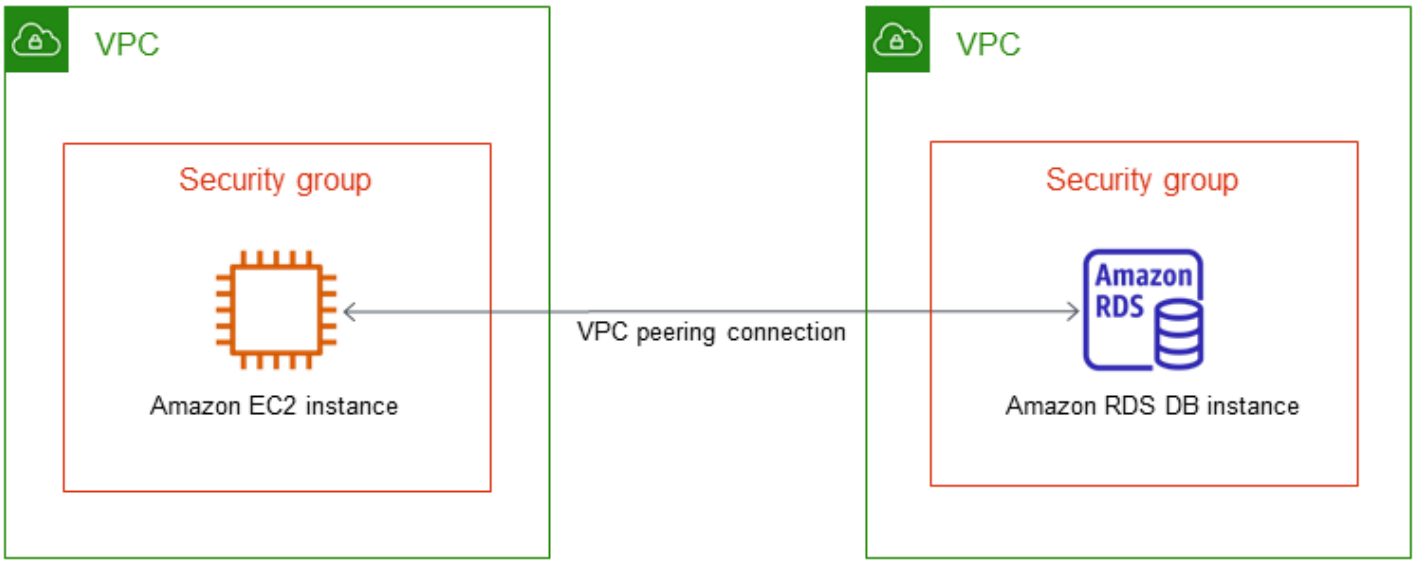


Untuk informasi selengkapnya tentang menghubungkan ke klaster DB dari instans EC2 Anda, lihat Menghubungkan ke instans DB Amazon RDS .

Instans DB dalam VPC yang diakses oleh instans EC2 dalam VPC yang sama

Jika instans DB Anda ada dalam VPC yang berbeda dari instans EC2 yang Anda gunakan untuk mengaksesnya, Anda dapat menggunakan peering VPC untuk mengakses instans DB.

Diagram berikut menunjukkan skenario ini.

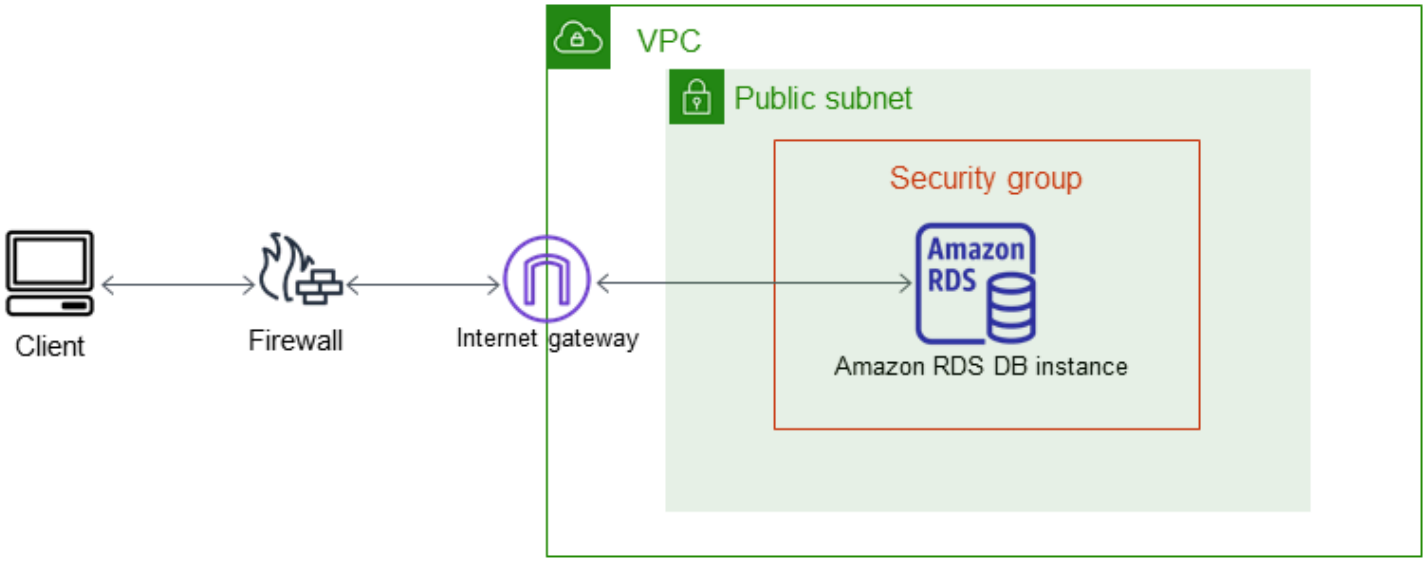


Koneksi peering VPC adalah koneksi jaringan antara dua VPC yang memungkinkan Anda merutekan lalu lintas di antara keduanya menggunakan alamat IP privat. Sumber daya dalam VPC mana pun dapat berkomunikasi satu sama lain seolah-olahnya ada di jaringan yang sama. Anda dapat membuat koneksi peering VPC antara VPC Anda sendiri, dengan VPC di akun AWS lain, atau dengan VPC di Wilayah Wilayah AWS yang berbeda. Untuk mempelajari selengkapnya tentang peering VPC, lihat Peering VPC dalam Panduan Pengguna Amazon Virtual Private Cloud.

Instans DB dalam VPC yang diakses oleh aplikasi klien melalui internet

Untuk mengakses instans DB di VPC dari aplikasi klien melalui internet, Anda dapat mengonfigurasi VPC dengan subnet publik tunggal, dan gateway internet untuk memungkinkan komunikasi melalui internet.

Diagram berikut menunjukkan skenario ini.



Kami merekomendasikan konfigurasi berikut:

- VPC ukuran /16 (misalnya CIDR: 10.0.0.0/16). Ukuran ini menyediakan 65.536 alamat IP privat.
- Subnet dengan ukuran /24 (misalnya CIDR: 10.0.0.0/24). Ukuran ini menyediakan 256 alamat IP privat.
- Instans DB Amazon RDS yang dikaitkan dengan VPC dan subnet. Amazon RDS menetapkan alamat IP dalam subnet ke instans DB Anda.
- Gateway internet yang menghubungkan VPC ke internet dan ke produk AWS lainnya.
- Grup keamanan yang terkait dengan instans DB. Aturan masuk grup keamanan memungkinkan aplikasi klien Anda mengakses instans DB Anda.

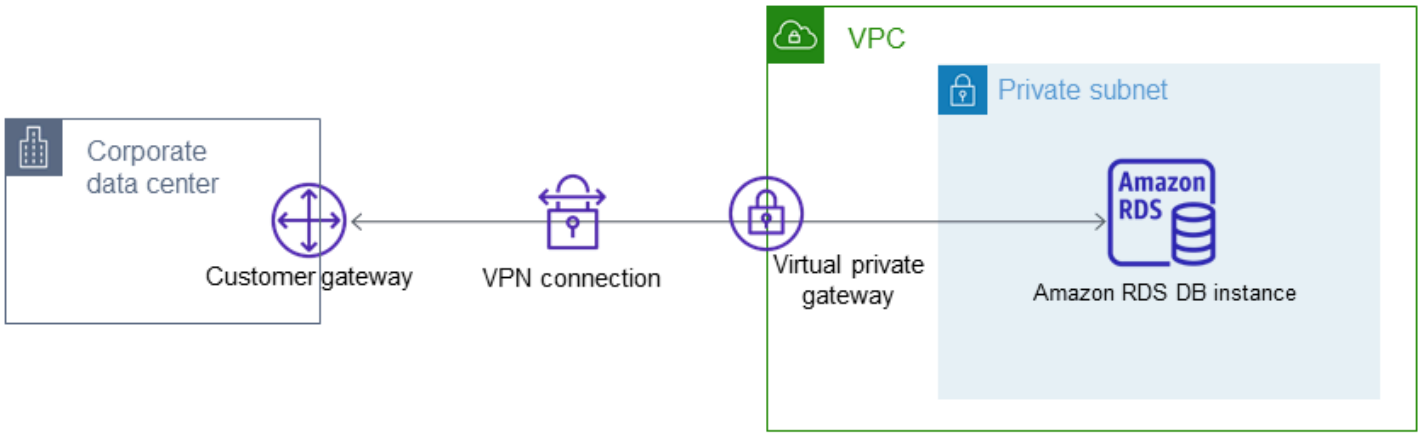
Untuk informasi tentang cara membuat instans DB di VPC, lihat Membuat klaster DB dalam VPC.

Instans DB dalam VPC yang diakses oleh jaringan privat

Jika instans DB Anda tidak dapat diakses secara publik, Anda memiliki opsi berikut untuk mengaksesnya dari jaringan privat:

- Koneksi AWS Site-to-Site VPN. Untuk informasi selengkapnya, lihat Apa itu AWS Site-to-Site VPN?
- Koneksi AWS Direct Connect. Untuk informasi selengkapnya, lihat Apa itu AWS Direct Connect?
- Koneksi AWS Client VPN. Untuk informasi selengkapnya, lihat Apa itu AWS Client VPN?

Diagram berikut menunjukkan skenario dengan koneksi AWS Site-to-Site VPN.

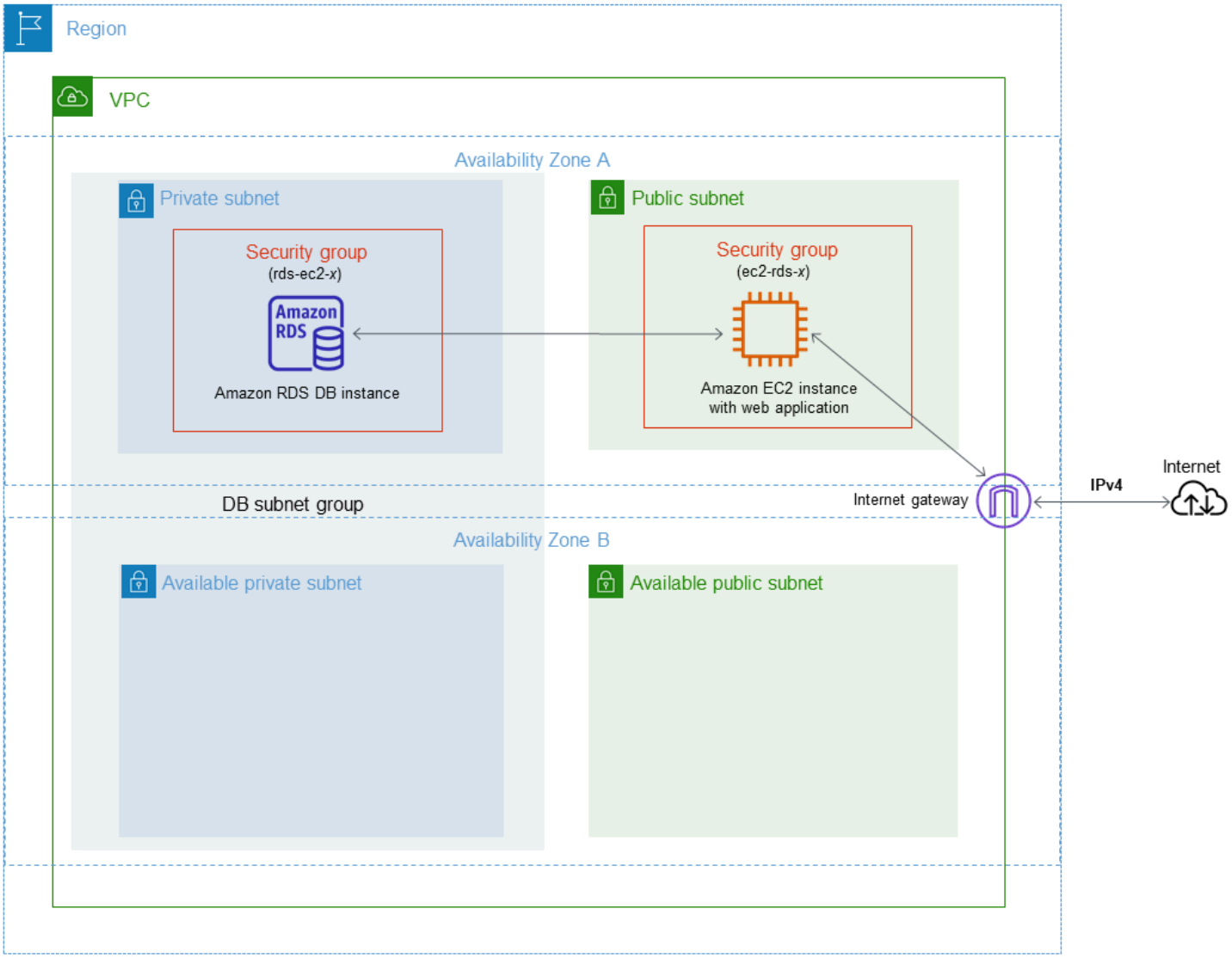


Untuk informasi selengkapnya, lihat Privasi lalu lintas jaringan internet.

Tutorial: Membuat VPC untuk digunakan dengan instans DB (khusus IPv4)

Skenario umum mencakup instans DB cloud privat virtual (VPC) berdasarkan layanan Amazon VPC. VPC ini berbagi data dengan server web yang berjalan di VPC yang sama. Dalam tutorial ini, Anda membuat VPC untuk skenario ini.

Diagram berikut menunjukkan skenario ini. Untuk informasi tentang skenario lain, lihat Skenario untuk mengakses instans DB di VPC.



Instans DB Anda harus tersedia hanya untuk server web Anda, dan bukan untuk internet publik. Dengan demikian, Anda membuat VPC dengan subnet publik maupun privat. Server web di-host di subnet publik, sehingga dapat menjangkau internet publik. Instans DB di-host di subnet privat. Server web dapat terhubung ke instans DB karena di-host dalam VPC yang sama. Tetapi instans DB tidak tersedia untuk internet publik, sehingga memberikan keamanan yang lebih baik.

Tutorial ini mengonfigurasi subnet publik dan privat tambahan di Zona Ketersediaan terpisah. Subnet ini tidak digunakan oleh tutorial. Grup subnet DB RDS membutuhkan subnet, setidaknya di dua Zona Ketersediaan. Subnet tambahan memudahkan untuk beralih ke deployment instans DB Multi-AZ di masa mendatang.

Tutorial ini menjelaskan konfigurasi VPC untuk instans DB Amazon RDS. Untuk tutorial yang menunjukkan cara membuat server web untuk skenario VPC ini, lihat Tutorial: Membuat server web dan instans DB Amazon RDS. Untuk informasi selengkapnya tentang Amazon VPC, lihat Panduan Memulai Amazon VPC dan Panduan Pengguna Amazon VPC.

> **Tip**
>
> Anda dapat mengatur konektivitas jaringan antara instans Amazon EC2 dan instans DB secara otomatis saat membuat instans DB. Konfigurasi jaringan mirip dengan yang dijelaskan dalam tutorial ini. Untuk informasi selengkapnya, lihat Konfigurasikan konektivitas jaringan otomatis dengan instans EC2.

Membuat VPC dengan subnet publik dan privat

Gunakan prosedur berikut untuk membuat VPC dengan subnet publik maupun privat.

Untuk membuat VPC dan subnet

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di sudut kanan atas AWS Management Console, pilih Wilayah untuk membuat VPC Anda. Contoh ini menggunakan Wilayah AS Barat (Oregon).
3. Di sudut kiri atas, pilih Dasbor VPC . Untuk mulai membuat VPC, pilih Buat VPC.
4. Agar Sumber daya dapat dibuat di bagian pengaturan VPC, pilih VPC dan lainnya.
5. Untuk pengaturan VPC, atur nilai-nilai ini:

   - Pembuatan otomatis tag nama – **`tutorial`**
   - Blok CIDR IPv4: – **`10.0.0.0/16`**
   - Blok CIDR IPv6 – Tidak ada Blok CIDR IPv6
   - Penghunian – Default
   - Jumlah Zona Ketersediaan (AZ) – 2
   - Sesuaikan AZ – Pertahankan nilai default.

- Jumlah subnet publik – 2
- Jumlah subnet privat – 2
- Sesuaikan subnet blok CIDR – Pertahankan nilai default.
- Gateway NAT ($) – Tidak ada
- Titik akhir VPC – Tidak ada
- Opsi DNS – Pertahankan nilai default.

> **Note**
>
> Amazon RDS membutuhkan setidaknya dua subnet di dua Zona Ketersediaan yang berbeda untuk mendukung deployment instans DB Multi-AZ. Tutorial ini membuat deployment Single-AZ, tetapi persyaratannya memudahkan konversi ke deployment instans DB Multi-AZ di masa mendatang.

6. Pilih Buat VPC.

Buat grup keamanan VPC untuk server web publik

Berikutnya, Anda membuat grup keamanan untuk akses publik. Untuk terhubung ke instans EC2 publik di VPC Anda, tambahkan aturan masuk ke grup keamanan VPC Anda. Ini memungkinkan lalu lintas untuk terhubung dari internet.

Untuk membuat grup keamanan VPC

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Pilih Dasbor VPC, pilih Grup Keamanan, lalu pilih Buat grup keamanan.
3. Di halaman Buat grup keamanan, atur nilai ini:
   - Nama grup keamanan: **`tutorial-securitygroup`**
   - Deskripsi: **`Tutorial Security Group`**
   - VPC: Pilih VPC yang Anda buat sebelumnya, misalnya: vpc-***`identifier`*** (tutorial-vpc)
4. Tambahkan aturan masuk ke grup keamanan.
   a. Tentukan alamat IP yang akan digunakan untuk terhubung ke instans EC2 di VPC Anda menggunakan Secure Shell (SSH). Untuk menentukan alamat IP publik Anda, Anda dapat

membuka layanan di https://checkip.amazonaws.com di jendela atau tab browser lain. Contoh alamat IP adalah `203.0.113.25/32`.

Dalam banyak kasus, Anda dapat menghubungkan melalui penyedia layanan Internet (ISP) atau dari belakang firewall Anda tanpa alamat IP statis. Jika demikian, temukan rentang alamat IP yang digunakan oleh komputer klien.

> **Warning**
>
> Jika Anda menggunakan `0.0.0.0/0` untuk akses SSH, Anda memungkinkan semua alamat IP untuk mengakses instans publik Anda menggunakan SSH. Hal ini dapat diterima untuk waktu yang singkat di lingkungan pengujian, tetapi tidak aman untuk lingkungan produksi. Dalam produksi, Anda hanya dapat memberikan otorisasi pada alamat IP atau rentang alamat tertentu saja untuk mengakses instans-instans Anda menggunakan SSH.

b. Di bagian Aturan masuk, pilih Tambahkan aturan.

c. Atur nilai berikut untuk aturan masuk baru Anda yang akan mengizinkan akses SSH ke instans Amazon EC2 Anda. Tindakan ini memungkinkan Anda terhubung ke instans Amazon EC2 Anda untuk menginstal server web dan utilitas lainnya. Anda juga terhubung ke instans EC2 Anda untuk mengunggah konten untuk server web Anda.

   - Jenis: **`SSH`**
   - Sumber: Rentang atau alamat IP dari Langkah a, misalnya: **`203.0.113.25/32`**.

d. Pilih Tambahkan aturan.

e. Atur nilai berikut untuk aturan masuk baru yang akan mengizinkan akses HTTP ke server web Anda.

   - Jenis: **`HTTP`**
   - Sumber: **`0.0.0.0/0`**

5. Untuk membuat grup keamanan, pilih Buat grup keamanan.

   Catat ID grup keamanan karena Anda membutuhkannya nanti dalam tutorial ini.

Membuat grup keamanan VPC untuk instans DB privat

Agar instans DB tetap privat, buat grup keamanan kedua untuk akses privat. Untuk terhubung ke instans DB privat di VPC, tambahkan aturan masuk ke grup keamanan VPC Anda yang mengizinkan lalu lintas dari server web Anda saja.

Untuk membuat grup keamanan VPC

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Pilih Dasbor VPC, pilih Grup Keamanan, lalu pilih Buat grup keamanan.
3. Di halaman Buat grup keamanan, atur nilai ini:
   - Nama grup keamanan: **`tutorial-db-securitygroup`**
   - Deskripsi: **`Tutorial DB Instance Security Group`**
   - VPC: Pilih VPC yang Anda buat sebelumnya, misalnya: vpc-***`identifier`*** (tutorial-vpc)
4. Tambahkan aturan masuk ke grup keamanan.
   a. Di bagian Aturan masuk, pilih Tambahkan aturan.
   b. Atur nilai berikut untuk aturan masuk baru Anda yang akan mengizinkan lalu lintas MySQL di port 3306 dari instans Amazon EC2 Anda. Tindakan ini memungkinkan Anda terhubung dari server web Anda ke instans DB Anda. Dengan demikian, Anda dapat menyimpan dan mengambil data dari aplikasi web Anda ke basis data Anda.
      - Jenis: **`MySQL/Aurora`**
      - Sumber: Pengidentifikasi grup keamanantutorial-securitygroup yang Anda buat sebelumnya dalam tutorial ini, misalnya: sg-9edd5cfb.
5. Untuk membuat grup keamanan, pilih Buat grup keamanan.

Membuat grup subnet DB

Grup subnet DB adalah kumpulan subnet yang Anda buat dalam VPC dan yang Anda tetapkan untuk instans DB. Grup subnet DB memungkinkan Anda untuk menentukan VPC tertentu saat membuat instans DB.

Untuk membuat grup subnet DB

1. Identifikasi subnet privat untuk basis data Anda di VPC.

a. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.

b. Pilih Dasbor VPC, lalu pilih Subnet.

c. Perhatikan ID subnet dari subnet bernama tutorial-subnet-private1-us-west-2a dan tutorial-subnet-private2-us-west-2b.

   Anda memerlukan ID subnet saat membuat grup subnet DB.

2. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.

   Pastikan Anda terhubung ke konsol Amazon RDS, bukan konsol Amazon VPC.

3. Di panel navigasi, pilih Grup subnet.

4. Pilih Buat grup subnet DB.

5. Di halaman Buat kelompok subnet DB, atur nilai ini di Detail grup subnet:

   - Nama: **`tutorial-db-subnet-group`**
   - Deskripsi: **`Tutorial DB Subnet Group`**
   - VPC: tutorial-vpc (vpc-***`identifier`***)

6. Di bagian Tambahkan subnet, pilih Zona Ketersediaan dan Subnet.

   Untuk tutorial ini, pilih us-west-2a dan us-west-2b untuk Zona Ketersediaan. Untuk Subnet, pilih subnet privat yang Anda identifikasi pada langkah sebelumnya.

7. Pilih Buat.

   Grup subnet DB baru Anda muncul dalam daftar grup subnet DB di konsol RDS. Anda dapat memilih grup subnet DB untuk melihat detail di panel detail di bagian bawah jendela. Detail ini mencakup semua subnet yang terkait dengan grup.

> **Note**
>
> Jika Anda membuat VPC ini untuk menyelesaikan Tutorial: Membuat server web dan instans DB Amazon RDS, buat instans DB dengan mengikuti petunjuk di Membuat instans DB Amazon RDS.

Menghapus VPC

Setelah membuat VPC dan sumber daya lainnya untuk tutorial ini, Anda dapat menghapusnya jika tidak dibutuhkan lagi.

> **Note**
>
> Jika Anda menambahkan sumber daya di VPC yang Anda buat untuk tutorial ini, Anda mungkin perlu menghapusnya sebelum menghapus VPC. Misalnya, sumber daya ini mungkin menyertakan instans Amazon EC2 atau instansDB Amazon RDS. Untuk informasi selengkapnya, lihat Menghapus VPC di Panduan Pengguna Amazon VPC.

Untuk menghapus VPC dan sumber daya terkait

1. Hapus grup subnet DB.
   a. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
   b. Di panel navigasi, pilih Grup subnet.
   c. Pilih grup subnet DB yang ingin Anda hapus, seperti tutorial-db-subnet-group.
   d. Pilih Hapus, lalu pilih Hapus di jendela konfirmasi.
2. Catat ID VPC.
   a. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
   b. Pilih Dasbor VPC, lalu pilih VPC.
   c. Dalam daftar, identifikasi VPC yang Anda buat, seperti tutorial-vpc.
   d. Catat ID VPC dari VPC yang Anda buat. ID VPC diperlukan di langkah berikutnya.
3. Hapus grup keamanan.
   a. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
   b. Pilih Dasbor VPC, lalu pilih Grup Keamanan.
   c. Pilih grup keamanan untuk instans DB Amazon RDS, seperti tutorial-db-securitygroup.
   d. Untuk Tindakan, pilih Hapus grup keamanan, lalu pilih Hapus di halaman konfirmasi.
   e. Di halaman Grup Keamanan, pilih grup keamanan untuk instans Amazon EC2, seperti tutorial-securitygroup.
   f. Untuk Tindakan, pilih Hapus grup keamanan, lalu pilih Hapus di halaman konfirmasi.

4. Hapus VPC.

    a. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.

    b. Pilih Dasbor VPC, lalu pilih VPC.

    c. Pilih VPC yang ingin Anda hapus, seperti tutorial-vpc.

    d. Untuk Tindakan, pilih Hapus VPC.

    Halaman konfirmasi menunjukkan sumber daya lain yang terkait dengan VPC yang juga akan dihapus, termasuk subnet yang terkait dengannya.

    e. Di halaman konfirmasi, masukkan **`delete`**, lalu pilih Hapus.

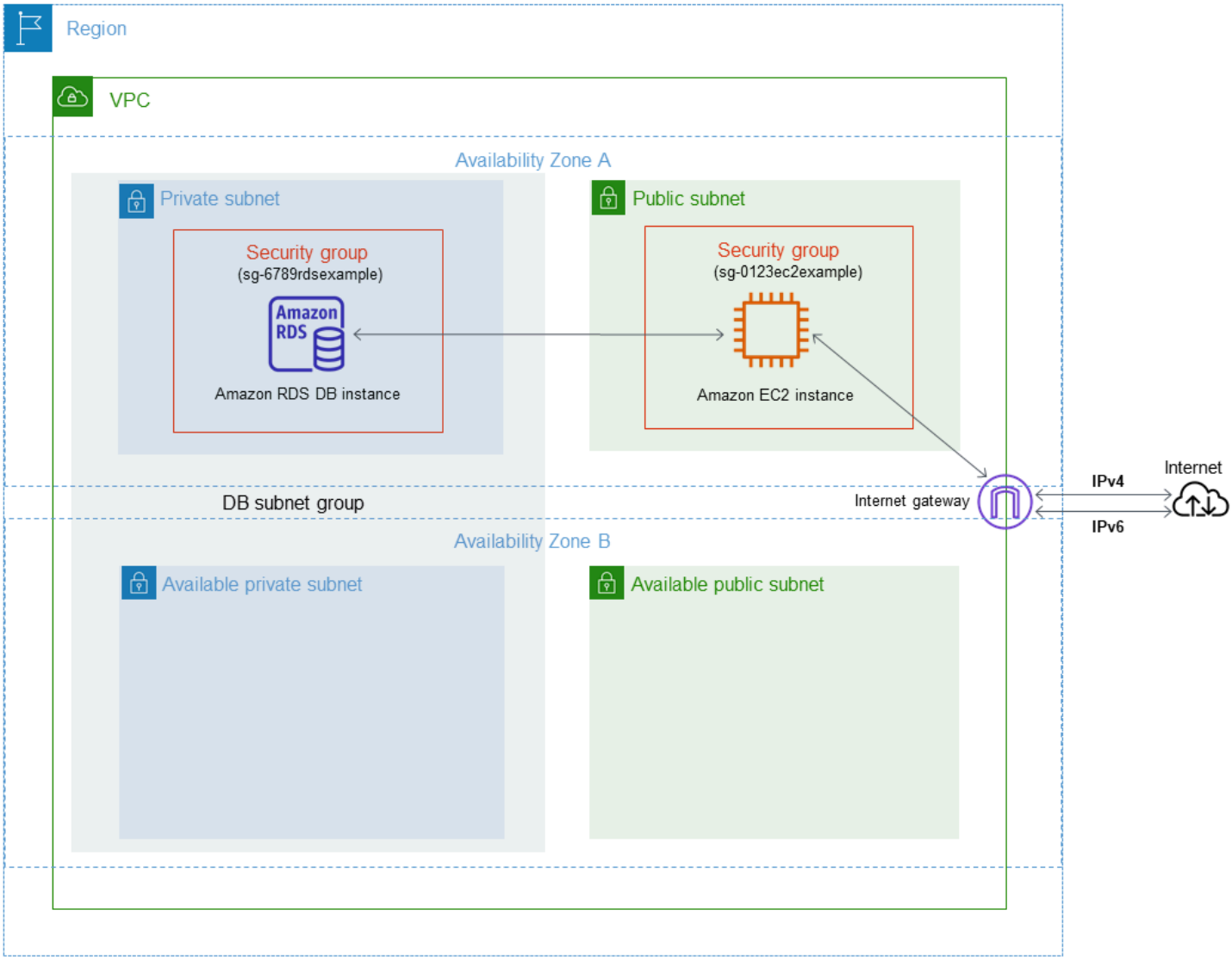
Tutorial: Membuat VPC untuk digunakan dengan instans DB (mode dual-stack)

Skenario umum mencakup instans DB cloud privat virtual (VPC) berdasarkan layanan Amazon VPC. VPC ini membagikan data dengan instans Amazon EC2 publik yang dijalankan di VPC yang sama.

Dalam tutorial ini, Anda akan membuat VPC untuk skenario ini yang berfungsi dengan basis data yang berjalan dalam mode dual-stack. Mode dual-stack untuk mengaktifkan koneksi melalui protokol penentuan alamat IPv6. Untuk informasi selengkapnya tentang penentuan alamat IP, lihat Penentuan alamat IP Amazon RDS.

Instans jaringan dual-stack didukung di sebagian besar wilayah. Untuk informasi selengkapnya, lihat Ketersediaan versi dan Wilayah. Untuk melihat batasan mode dual-stack, lihat Batasan untuk klaster DB jaringan dual-stack.

Diagram berikut menunjukkan skenario ini.



Untuk informasi tentang skenario lain, lihat Skenario untuk mengakses instans DB di VPC.

Instans DB Anda harus tersedia hanya untuk instans Amazon EC2 Anda, dan bukan untuk internet publik. Dengan demikian, Anda membuat VPC dengan subnet publik maupun privat. Instans Amazon EC2 di-hosting di subnet publik agar dapat menjangkau internet publik. Instans DB di-host di subnet privat. Instans Amazon EC2 dapat terhubung ke instans DB karena di-hosting dalam VPC yang sama. Namun, instans DB tidak tersedia untuk internet publik, sehingga memberikan keamanan yang lebih besar.

Tutorial ini mengonfigurasi subnet publik dan privat tambahan di Zona Ketersediaan terpisah. Subnet ini tidak digunakan oleh tutorial. Grup subnet DB RDS membutuhkan subnet, setidaknya di dua Zona Ketersediaan. Subnet tambahan memudahkan untuk beralih ke deployment instans DB Multi-AZ di masa mendatang.

Untuk membuat instans DB yang menggunakan mode dual-stack, tetapkan Mode dual-stack untuk pengaturan Jenis jaringan. Anda juga dapat mengubah instans DB dengan pengaturan yang sama. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS dan Memodifikasi instans DB Amazon RDS.

Tutorial ini menjelaskan konfigurasi VPC untuk instans DB Amazon RDS. Untuk informasi selengkapnya tentang Amazon VPC, lihat Panduan Pengguna Amazon VPC.

Membuat VPC dengan subnet publik dan privat

Gunakan prosedur berikut untuk membuat VPC dengan subnet publik maupun privat.

Untuk membuat VPC dan subnet

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di pojok kanan atas AWS Management Console, pilih Wilayah untuk membuat VPC Anda. Contoh ini menggunakan Wilayah AS Timur (Ohio).
3. Di sudut kiri atas, pilih Dasbor VPC . Untuk mulai membuat VPC, pilih Buat VPC.
4. Agar Sumber daya dapat dibuat di bagian pengaturan VPC, pilih VPC dan lainnya.
5. Untuk Pengaturan VPC lainnya, atur nilai-nilai ini:
   - Pembuatan otomatis tag nama – **`tutorial-dual-stack`**
   - Blok CIDR IPv4: – **`10.0.0.0/16`**
   - Blok CIDR IPv6 – Blok CIDR IPv6 yang disediakan Amazon
   - Penghunian – Default
   - Jumlah Zona Ketersediaan (AZ) – 2
   - Sesuaikan AZ – Pertahankan nilai default.
   - Jumlah subnet publik – 2
   - Jumlah subnet privat – 2
   - Sesuaikan subnet blok CIDR – Pertahankan nilai default.
   - Gateway NAT ($) – Tidak ada
   - Gateway internet khusus egress – Tidak
   - Titik akhir VPC – Tidak ada
   - Opsi DNS – Pertahankan nilai default.

> **Note**
>
> Amazon RDS membutuhkan setidaknya dua subnet di dua Zona Ketersediaan yang berbeda untuk mendukung deployment instans DB Multi-AZ. Tutorial ini membuat deployment Single-AZ, tetapi persyaratannya memudahkan konversi ke deployment instans DB Multi-AZ di masa mendatang.

6. Pilih Buat VPC.

Membuat grup keamanan VPC untuk instans Amazon EC2 publik

Berikutnya, Anda membuat grup keamanan untuk akses publik. Untuk terhubung ke instans EC2 publik di VPC Anda, tambahkan aturan masuk ke grup keamanan VPC Anda yang mengizinkan lalu lintas untuk terhubung dari internet.

Untuk membuat grup keamanan VPC

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Pilih Dasbor VPC, pilih Grup Keamanan, lalu pilih Buat grup keamanan.
3. Di halaman Buat grup keamanan, atur nilai ini:
   - Nama grup keamanan: **`tutorial-dual-stack-securitygroup`**
   - Deskripsi: **`Tutorial Dual-Stack Security Group`**
   - VPC: Pilih VPC yang Anda buat sebelumnya, misalnya: vpc-***`identifier`*** (tutorial–dual-stack-vpc)
4. Tambahkan aturan masuk ke grup keamanan.
   a. Tentukan alamat IP yang akan digunakan untuk terhubung ke instans EC2 di VPC Anda menggunakan Secure Shell (SSH).

      Contoh rentang alamat Protokol Internet versi 4 (IPv4) adalah `203.0.113.25/32`. Contoh rentang alamat Protokol Internet versi 6 (IPv6) adalah `2001:db8:1234:1a00::/64`.

      Dalam banyak kasus, Anda dapat menghubungkan melalui penyedia layanan Internet (ISP) atau dari belakang firewall Anda tanpa alamat IP statis. Jika demikian, temukan rentang alamat IP yang digunakan oleh komputer klien.

> **Warning**
>
> Jika Anda menggunakan `0.0.0.0/0` untuk IPv4 atau `::0` untuk IPv6, Anda memungkinkan semua alamat IP untuk mengakses instans publik Anda menggunakan SSH. Hal ini dapat diterima untuk waktu yang singkat di lingkungan pengujian, tetapi tidak aman untuk lingkungan produksi. Dalam produksi, Anda hanya dapat memberikan otorisasi pada alamat IP atau rentang alamat tertentu saja untuk mengakses instans-instans Anda.

b. Di bagian Aturan masuk, pilih Tambahkan aturan.

c. Atur nilai berikut untuk aturan masuk baru Anda yang akan mengizinkan akses Secure Shell (SSH) ke instans Amazon EC2 Anda. Jika Anda melakukan ini, Anda dapat terhubung ke instans EC2 Anda untuk menginstal klien SQL dan aplikasi lainnya. Tentukan alamat IP agar Anda dapat mengakses instans EC2 Anda:

- Jenis: **SSH**
- Sumber: Rentang atau alamat IP dari langkah a. Contoh alamat IP IPv4 adalah **203.0.113.25/32**. Contoh alamat IP IPv6 adalah **2001:DB8::/32**.

5. Untuk membuat grup keamanan, pilih Buat grup keamanan.

   Catat ID grup keamanan karena Anda membutuhkannya nanti dalam tutorial ini.

Membuat grup keamanan VPC untuk instans DB privat

Agar instans DB tetap privat, buat grup keamanan kedua untuk akses privat. Untuk terhubung ke instans DB di VPC Anda, tambahkan aturan masuk ke grup keamanan VPC Anda. Hal ini mengizinkan lalu lintas dari instans Amazon EC2 Anda saja.

Untuk membuat grup keamanan VPC

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Pilih Dasbor VPC, pilih Grup Keamanan, lalu pilih Buat grup keamanan.
3. Di halaman Buat grup keamanan, atur nilai ini:

   - Nama grup keamanan: **tutorial-dual-stack-db-securitygroup**
   - Deskripsi: **Tutorial Dual-Stack DB Instance Security Group**

- VPC: Pilih VPC yang Anda buat sebelumnya, misalnya: vpc-***identifier*** (tutorial–dual-stack-vpc)

4. Tambahkan aturan masuk ke grup keamanan:

   a. Di bagian Aturan masuk, pilih Tambahkan aturan.

   b. Atur nilai berikut untuk aturan masuk baru Anda yang akan mengizinkan lalu lintas MySQL di port 3306 dari instans Amazon EC2 Anda. Dengan melakukannya, Anda dapat terhubung dari instans EC2 Anda ke instans DB Anda. Melakukan hal ini berarti Anda dapat mengirimkan data dari instans EC2 Anda ke basis data Anda.

      - Jenis: MySQL/Aurora
      - Sumber: Pengidentifikasi grup keamanan tutorial-dual-stack-securitygroup yang Anda buat sebelumnya dalam tutorial ini, misalnya sg-9edd5cfb.

5. Untuk membuat grup keamanan, pilih Buat grup keamanan.

Membuat grup subnet DB

Grup subnet DB adalah kumpulan subnet yang Anda buat dalam VPC dan yang Anda tetapkan untuk instans DB. Dengan menggunakan grup subnet DB, Anda dapat menentukan VPC tertentu saat membuat instans DB. Untuk membuat grup subnet DB yang kompatibel dengan `DUAL`, semua subnet harus kompatibel dengan `DUAL`. Agar kompatibel dengan `DUAL`, subnet harus memiliki CIDR IPv6 yang dikaitkan dengan subnet tersebut.

Untuk membuat grup subnet DB

1. Identifikasi subnet privat untuk basis data Anda di VPC.

   a. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.

   b. Pilih Dasbor VPC, lalu pilih Subnet.

   c. Perhatikan ID subnet dari subnet bernama tutorial-dual-stack-subnet-private1-us-west-2a dan tutorial-dual-stack-subnet-private2-us-west-2b.

      Anda memerlukan ID subnet saat membuat grup subnet DB.

2. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.

   Pastikan Anda terhubung ke konsol Amazon RDS, bukan konsol Amazon VPC.

3. Di panel navigasi, pilih Grup subnet.

4. Pilih Buat grup subnet DB.
5. Di halaman Buat kelompok subnet DB, atur nilai ini di Detail grup subnet:
   - Nama: **`tutorial-dual-stack-db-subnet-group`**
   - Deskripsi: **`Tutorial Dual-Stack DB Subnet Group`**
   - VPC: tutorial-dual-stack-vpc (vpc-***`identifier`***)
6. Di bagian Tambahkan subnet, pilih nilai untuk opsi Zona Ketersediaan dan Subnet.

   Untuk tutorial ini, pilih us-east-2a dan us-east-2b untuk Zona Ketersediaan. Untuk Subnet, pilih subnet privat yang Anda identifikasi pada langkah sebelumnya.
7. Pilih Buat.

Grup subnet DB baru Anda muncul dalam daftar grup subnet DB di konsol RDS. Anda dapat memilih grup subnet DB untuk melihat detailnya. Detail ini termasuk protokol penentuan alamat yang didukung, semua subnet yang terkait dengan grup tersebut, dan jenis jaringan yang didukung oleh grup subnet DB.

Membuat instans Amazon EC2 dalam mode dual-stack

Untuk membuat instans Amazon EC2, ikuti petunjuk dalam Meluncurkan instans menggunakan wizard peluncuran instans baru di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Di halaman Mengonfigurasi Detail Instans, atur nilai-nilai ini dan biarkan nilai lainnya sebagai default:

- Jaringan – Pilih VPC yang ada dengan subnet publik dan privat seperti tutorial-dual-stack-vpc (vpc-*`identifier`*) yang dibuat di Membuat VPC dengan subnet publik dan privat.
- Subnet – Pilih subnet publik yang sudah ada, seperti subnet-***`identifier`*** | tutorial-dual-stack-subnet-public1-us-east-2a | us-east-2a yang dibuat di Membuat grup keamanan VPC untuk instans Amazon EC2 publik.
- IP Publik yang Otomatis Ditetapkan – Pilih Aktifkan.
- IP IPv6 yang Otomatis Ditetapkan – Pilih Aktifkan.
- Firewall (grup keamanan) – Pilih Pilih grup keamanan yang ada.
- Grup keamanan umum – Pilih grup keamanan yang ada, seperti `tutorial-securitygroup` yang dibuat di Membuat grup keamanan VPC untuk instans Amazon EC2 publik. Pastikan grup keamanan yang Anda pilih menyertakan aturan masuk untuk akses Secure Shell (SSH) dan HTTP.

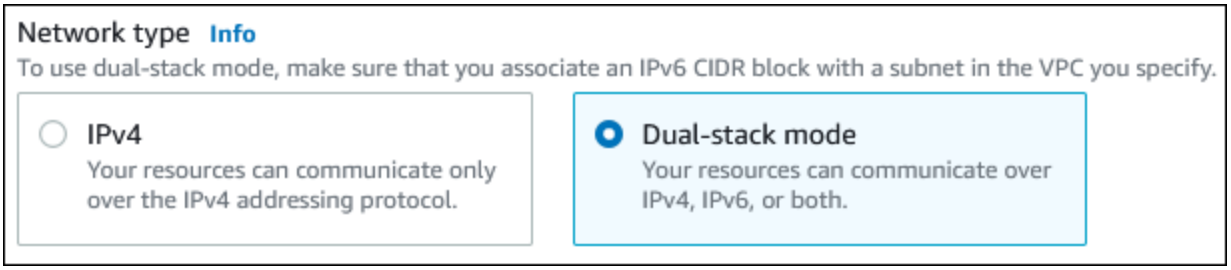
Membuat instans DB dalam mode dual-stack

Pada langkah ini, Anda akan membuat instans DB yang berjalan dalam mode dual-stack.

Membuat instans DB

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
2. Di pojok kanan atas konsol, pilih Wilayah AWS tempat Anda akan membuat instans DB. Contoh ini menggunakan Wilayah AS Timur (Ohio).
3. Di panel navigasi, pilih Basis Data.
4. Pilih Buat basis data.
5. Di halaman Membuat basis data, pastikan opsi Pembuatan standar dipilih, lalu pilih jenis mesin MySQL DB.
6. Di bagian Konektivitas, atur nilai-nilai ini:

   - Jenis jaringan – Pilih Mode dual-stack.

   - Cloud privat virtual (VPC) – Pilih VPC yang ada dengan subnet publik dan privat seperti tutorial-dual-stack-vpc (vpc-*identifier*) yang dibuat di Membuat VPC dengan subnet publik dan privat.

     VPC tersebut harus memiliki subnet di Zona Ketersediaan yang berbeda.

   - Grup subnet DB – Pilih grup subnet DB untuk VPC, seperti tutorial-dual-stack-db-subnet-group yang dibuat di Membuat grup subnet DB.
   - Akses publik – Pilih Tidak.
   - Grup keamanan VPC (firewall) – Pilih Pilih grup keamanan yang ada.
   - Grup keamanan VPC yang ada – Pilih grup keamanan VPC yang sudah ada dan dikonfigurasi untuk akses privat, seperti tutorial-dual-stack-db-securitygroup yang dibuat di Membuat grup keamanan VPC untuk instans DB privat.

Hapus grup keamanan lainnya, seperti grup keamanan default, dengan memilih X yang dikaitkan dengan masing-masing grup keamanan.

- Zona Ketersediaan – Pilih us-west-2a.

  Untuk menghindari lalu lintas AZ, pastikan instans DB dan instans EC2 berada di Zona Ketersediaan yang sama.

7. Untuk bagian yang tersisa, tentukan pengaturan instans DB Anda. Untuk informasi tentang setiap pengaturan, lihat Pengaturan untuk instans DB.

Menghubungkan ke instans Amazon EC2 dan instans DB

Setelah membuat instans Amazon EC2 dan instans DB dalam mode dual-stack, Anda dapat terhubung ke masing-masing instans menggunakan protokol IPv6. Untuk menghubungkan ke instans Amazon EC2 menggunakan protokol IPv6, ikuti petunjuk di Menghubungkan ke instans Linux di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Untuk menghubungkan ke instans DB RDS for MySQL dari instans Amazon EC2, ikuti petunjuk di Menghubungkan ke instans DB MySQL.

Menghapus VPC

Setelah membuat VPC dan sumber daya lainnya untuk tutorial ini, Anda dapat menghapusnya jika tidak dibutuhkan lagi.

Jika Anda menambahkan sumber daya di VPC yang Anda buat untuk tutorial ini, Anda mungkin perlu menghapusnya sebelum menghapus VPC. Contoh sumber dayanya adalah instans Amazon EC2 atau instans DB. Untuk informasi selengkapnya, lihat Menghapus VPC di Panduan Pengguna Amazon VPC.

Untuk menghapus VPC dan sumber daya terkait

1. Hapus grup subnet DB:

   a. Buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
   b. Di panel navigasi, pilih Grup subnet.
   c. Pilih grup subnet DB yang akan dihapus, seperti tutorial-db-subnet-group.
   d. Pilih Hapus, lalu pilih Hapus di jendela konfirmasi.

2. Catat ID VPC:

a. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
b. Pilih Dasbor VPC, lalu pilih VPC.
c. Dalam daftar, identifikasi VPC yang Anda buat, seperti tutorial-dual-stack-vpc.
d. Catat nilai ID VPC dari VPC yang Anda buat. Anda memerlukan ID VPC ini di langkah selanjutnya.

3. Hapus grup keamanan:

a. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
b. Pilih Dasbor VPC, lalu pilih Grup Keamanan.
c. Pilih grup keamanan untuk instans DB Amazon RDS, seperti tutorial-dual-stack-db-securitygroup.
d. Untuk Tindakan, pilih Hapus grup keamanan, lalu pilih Hapus di halaman konfirmasi.
e. Di halaman Grup Keamanan, pilih grup keamanan untuk instans Amazon EC2, seperti tutorial-dual-stack-securitygroup.
f. Untuk Tindakan, pilih Hapus grup keamanan, lalu pilih Hapus di halaman konfirmasi.

4. Hapus gateway NAT:

a. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
b. Pilih Dasbor VPC, lalu pilih Gateway NAT.
c. Pilih gateway NAT dari VPC yang Anda buat. Gunakan ID VPC untuk mengidentifikasi gateway NAT yang benar.
d. Untuk Tindakan, pilih Hapus gateway NAT.
e. Di halaman konfirmasi, masukkan **`delete`**, lalu pilih Hapus.

5. Hapus VPC:

a. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
b. Pilih Dasbor VPC, lalu pilih VPC.
c. Pilih VPC yang ingin Anda hapus, seperti tutorial-dual-stack-vpc.
d. Untuk Tindakan, pilih Hapus VPC.

   Halaman konfirmasi menunjukkan sumber daya lain yang terkait dengan VPC yang juga akan dihapus, termasuk subnet yang terkait dengannya.

e. Di halaman konfirmasi, masukkan **`delete`**, lalu pilih Hapus.

6. Rilis alamat IP Elastis:

    a. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
    b. Pilih Dasbor EC2, lalu pilih IP Elastis.
    c. Pilih alamat IP Elastis yang ingin Anda rilis.
    d. Untuk Tindakan, pilih Rilis alamat IP Elastis.
    e. Di halaman konfirmasi, pilih Rilis.

Memindahkan instans DB yang tidak berada dalam VPC ke VPC

Beberapa instans DB lama pada platform EC2-Classic tidak berada dalam VPC. Jika instans DB Anda tidak berada dalam VPC, Anda dapat menggunakan AWS Management Console untuk memindahkan instans DB Anda dengan mudah ke VPC. Sebelum Anda dapat memindahkan instans DB yang tidak berada dalam VPC ke VPC, Anda harus membuat VPC.

EC2-Classic dipensiunkan pada 15 Agustus 2022. Jika Anda belum bermigrasi dari EC2-Classic ke VPC, kami sarankan Anda bermigrasi sesegera mungkin. Untuk informasi selengkapnya, lihat Migrasi dari EC2-Classic ke VPC dalam Panduan Pengguna Amazon EC2 dan blog EC2-Classic Networking akan Segera Dipensiunkan - Berikut Cara Mempersiapkannya.

Important

Jika Anda adalah pelanggan Amazon RDS baru, jika Anda belum pernah membuat instans DB sebelumnya, atau jika Anda membuat instans DB di Wilayah AWS yang belum pernah Anda gunakan sebelumnya, biasanya Anda berada di platform EC2-VPC dan memiliki VPC default. Untuk informasi tentang menggunakan instans DB dalam VPC, lihat Bekerja dengan klaster DB dalam VPC.

Ikuti langkah-langkah ini untuk membuat VPC untuk instans DB Anda.

- Langkah 1: Buat VPC
- Langkah 2: Buat grup subnet DB
- Langkah 3: Buat grup keamanan VPC

Setelah Anda membuat VPC, ikuti langkah-langkah ini untuk memindahkan instans DB Anda ke VPC.

- Memperbarui VPC untuk instans DB

Kami sangat menyarankan Anda untuk membuat cadangan instans DB segera sebelum migrasi. Tindakan ini akan memastikan bahwa Anda dapat memulihkan data jika migrasi gagal. Untuk informasi selengkapnya, lihat Mencadangkan, memulihkan, dan mengekspor data.

Berikut ini adalah beberapa batasan untuk memindahkan instans DB Anda ke VPC.

- Kelas instans DB generasi sebelumnya – Kelas instans DB generasi sebelumnya mungkin tidak didukung pada platform VPC. Saat memindahkan instans DB ke VPC, pilih kelas instans DB db.m3 atau db.r3. Setelah Anda memindahkan instans DB ke VPC, Anda dapat menskalakan instans DB untuk menggunakan kelas instans DB yang lebih tinggi. Untuk daftar lengkap kelas instans yang didukung VPC, lihat Jenis instans Amazon RDS.
- Multi-AZ – Pemindahan instans DB Multi-AZ yang tidak berada dalam VPC ke VPC tidak didukung saat ini. Untuk memindahkan instans DB Anda ke VPC, pertama-tama ubah instans DB agar menjadi deployment AZ tunggal. Ubah pengaturan Deployment Multi-AZ ke Tidak. Setelah Anda memindahkan instans DB ke VPC, ubah lagi untuk menjadikannya deployment Multi-AZ. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.
- Replika baca – Pemindahan instans DB dengan replika baca yang tidak berada dalam VPC ke VPC tidak didukung saat ini. Untuk memindahkan instans DB Anda ke VPC, pertama-tama hapus semua replika baca. Setelah Anda memindahkan instans DB ke VPC, buat ulang replika baca. Untuk informasi selengkapnya, lihat Menggunakan replika baca instans DB.
- Grup opsi – Jika Anda memindahkan instans DB Anda ke VPC, dan instans DB menggunakan grup opsi kustom, ubah grup opsi yang terkait dengan instans DB Anda. Grup opsi bersifat khusus untuk platform, dan perpindahan ke VPC adalah perubahan pada platform. Untuk menggunakan grup opsi kustom dalam kasus ini, tetapkan grup opsi VPC default ke instans DB, tetapkan grup opsi yang digunakan oleh instans DB lain dalam VPC tempat Anda memindahkan instans DB, atau buat grup opsi baru dan tetapkan ke instans DB. Untuk informasi selengkapnya, lihat Menggunakan grup opsi.

Alternatif untuk memindahkan instans DB yang tidak ada di VPC ke VPC dengan waktu henti minimal

Dengan menggunakan alternatif berikut, Anda dapat memindahkan instans DB yang tidak ada di VPC ke VPC dengan waktu henti minimal. Alternatif ini meminimalkan gangguan pada instans DB sumber dan memungkinkannya melayani lalu lintas pengguna selama migrasi. Namun, waktu yang diperlukan untuk bermigrasi ke VPC akan bervariasi berdasarkan ukuran basis data dan karakteristik beban kerja aktif.

- AWS Database Migration Service (AWS DMS) – AWS DMS memungkinkan migrasi data secara langsung sambil menjaga instans DB sumber tetap beroperasi penuh, tetapi hanya mereplikasi pernyataan DDL terbatas. AWS DMS tidak menyebarkan item seperti indeks, pengguna, hak akses, prosedur tersimpan, dan perubahan basis data lainnya yang tidak terkait langsung dengan data tabel. Selain itu, AWS DMS tidak secara otomatis menggunakan snapshot RDS

untuk pembuatan instans DB awal, yang dapat meningkatkan waktu migrasi. Untuk informasi selengkapnya, lihat AWS Database Migration Service.

- Pemulihan snapshot DB atau pemulihan titik waktu – Anda dapat memindahkan instans DB ke VPC dengan memulihkan snapshot instans DB atau dengan memulihkan instans DB ke suatu titik waktu. Untuk informasi selengkapnya, lihat Memulihkan dari snapshot DB dan Memulihkan instans DB dengan waktu yang ditentukan.

Kuota dan batasan untuk Amazon RDS

Setelah itu, Anda dapat menemukan deskripsi kuota sumber daya dan batasan penamaan untuk Amazon RDS.

Topik

Kuota dalam Amazon RDS

Setiap AWS akun memiliki kuota, untuk setiap AWS Wilayah, pada jumlah sumber daya Amazon RDS yang dapat dibuat. Setelah kuota sumber daya tercapai, panggilan tambahan untuk membuat sumber daya tersebut akan gagal dengan pengecualian.

Tabel berikut mencantumkan sumber daya dan kuota mereka per AWS Wilayah.

Nama	Default	Dapat disesuaikan	Deskripsi
Otorisasi per grup keamanan DB	Setiap Wilayah yang didukung: 20	Tidak	Jumlah otorisasi grup keamanan per grup keamanan DB
Versi mesin kustom	Setiap Wilayah yang didukung: 40	Ya	Jumlah maksimum versi mesin kustom yang diizinkan di akun di Wilayah saat ini
Grup parameter klaster DB	Setiap Wilayah yang didukung: 50	Tidak	Jumlah maksimum grup parameter klaster DB
Klaster DB	Setiap Wilayah yang didukung: 40	Ya	Jumlah maksimum klaster Aurora yang diizinkan di

Nama	Default	Dapat disesu an	Deskripsi
			akun ini dalam Wilayah saat ini
Instans DB	Setiap Wilayah yang didukung: 40	Ya	Jumlah maksimum instans DB yang diizinkan di akun ini dalam Wilayah saat ini
Grup subnet DB	Setiap Wilayah yang didukung: 50	Ya	Jumlah maksimum grup subnet DB
Ukuran konten permintaan HTTP API Data	Setiap Wilayah yang didukung: 4 Megabyte	Tidak	Ukuran maksimum yang diizinkan untuk konten permintaan HTTP.
Pasangan rahasia klaster serentak maksimum API Data	Setiap Wilayah yang didukung: 30	Tidak	Jumlah maksimum pasangan unik cluster dan rahasia DB Tanpa Server Aurora dalam permintaan API Data bersamaan untuk akun dan Wilayah saat ini. AWS

Nama	Default	Dapat disesu an	Deskripsi
Permintaan serentak maksimum API Data	Setiap Wilayah yang didukung: 500	Tidak	Jumlah maksimum permintaan API Data ke klaster DB Aurora Nirserver yang menggunakan rahasia yang sama dan dapat diproses pada saat yang sama. Permintaa n tambahan dimasukka n ke dalam antrean dan diproses setelah permintaan dalam proses selesai.
Ukuran set hasil maksimum API Data	Setiap Wilayah yang didukung: 1 Megabyte	Tidak	Ukuran maksimum set hasil basis data yang dapat dikembalikan oleh API Data.
Ukuran maksimum API Data string respons JSON	Setiap Wilayah yang didukung: 10 Megabyte	Tidak	Ukuran maksimum string respons JSON yang disederhanakan dikembalikan oleh API Data RDS.
Permintaan API Data per detik	Setiap Wilayah yang didukung: 1.000 per detik	Tidak	Jumlah maksimum permintaan ke Data API per detik yang diizinkan di akun ini di AWS Wilayah saat ini.
Langganan peristiwa	Setiap Wilayah yang didukung: 20	Ya	Jumlah maksimum langganan peristiwa

Nama	Default	Dapat disesuaikan	Deskripsi
Peran IAM per klaster DB	Setiap Wilayah yang didukung: 5	Ya	Jumlah maksimum peran IAM yang terkait dengan klaster DB
Peran IAM per instans DB	Setiap Wilayah yang didukung: 5	Ya	Jumlah maksimum peran IAM yang terkait dengan instans DB
Snapshot klaster DB manual	Setiap Wilayah yang didukung: 100	Ya	Jumlah maksimum snapshot klaster DB manual
Snapshot instans DB manual	Setiap Wilayah yang didukung: 100	Ya	Jumlah maksimum snapshot instans DB manual
Grup opsi	Setiap Wilayah yang didukung: 20	Ya	Jumlah maksimum grup opsi
Grup parameter	Setiap Wilayah yang didukung: 50	Ya	Jumlah maksimum grup parameter
Proksi	Setiap Wilayah yang didukung: 20	Ya	Jumlah maksimum proxy yang diizinkan di akun ini di Wilayah saat ini AWS
Replika baca per primer	Setiap Wilayah yang didukung: 15	Ya	Jumlah maksimum replika baca per instans DB primer. Kuota ini tidak dapat disesuaikan untuk Amazon Aurora.

Nama	Default	Dapat disesu an	Deskripsi
Instans DB cadangan	Setiap Wilayah yang didukung: 40	Ya	Jumlah maksimum instans DB cadangan yang diizinkan di akun ini di Wilayah saat ini AWS
Aturan per grup keamanan	Setiap Wilayah yang didukung: 20	Tidak	Jumlah maksimum aturan per grup keamanan DB
Grup keamanan	Setiap Wilayah yang didukung: 25	Ya	Jumlah maksimum aturan grup keamanan DB
Grup keamanan (VPC)	Setiap Wilayah yang didukung: 5	Tidak	Jumlah maksimum grup keamanan DB per Amazon VPC
Subnet per grup subnet DB	Setiap Wilayah yang didukung: 20	Tidak	Jumlah maksimum subnet per grup subnet DB
Tanda per sumber daya	Setiap Wilayah yang didukung: 50	Tidak	Jumlah maksimum tanda per sumber daya Amazon RDS

Nama	Default	Dapat disesuaikan	Deskripsi
Total penyimpanan untuk semua instans DB	Setiap Wilayah yang didukung: 100.000 Gigabyte	Ya	Total penyimpanan maksimum (dalam GB) pada volume EBS untuk semua instans DB Amazon RDS yang ditambahkan bersama. Kuota ini tidak berlaku untuk Amazon Aurora, yang memiliki volume cluster maksimum 128 TiB untuk setiap cluster DB.

Note

Secara default, Anda dapat memiliki hingga total 40 instans DB. Instans DB RDS, instans DB Aurora, instans Amazon Neptune, dan instans Amazon DocumentDB berlaku untuk kuota ini. Batasan berikut berlaku untuk instans DB Amazon RDS:

- 10 untuk setiap edisi SQL Server (Enterprise, Standard, Web, dan Express) dalam model "license-included"
- 10 untuk Oracle dalam model "license-included"
- 40 untuk Db2 di bawah model lisensi "bring-your-own-license" (BYOL)
- 40 untuk MySQL, MariaDB, atau PostgreSQL
- 40 untuk Oracle di bawah model lisensi bring-your-own-license "" (BYOL)

Jika aplikasi Anda memerlukan lebih banyak instans DB, Anda dapat meminta instans DB tambahan dengan membuka Konsol Service Quotas. Di panel navigasi, pilih Layanan AWS . Pilih Amazon Relational Database Service (Amazon RDS), pilih kuota, dan ikuti arahan untuk meminta peningkatan kuota. Untuk informasi selengkapnya, lihat Meminta peningkatan kuota di Panduan Pengguna Service Quotas.

Untuk RDS for Oracle dan RDS for SQL Server, batas replika baca adalah 5 per basis data sumber untuk setiap Wilayah.
Pencadangan yang dikelola oleh AWS Backup dianggap sebagai snapshot DB manual, tetapi tidak dihitung dalam kuota snapshot manual. Untuk selengkapnya AWS Backup, lihat Panduan AWS Backup Pengembang.

Jika Anda menggunakan operasi API RDS dan melebihi kuota default untuk jumlah panggilan per detik, API Amazon RDS akan mengeluarkan kesalahan seperti berikut.

ClientError: Terjadi kesalahan (ThrottlingException) saat memanggil operasi *API_name*: Nilai terlampaui.

Di sini, kurangi jumlah panggilan per detik. Kuota dimaksudkan untuk mencakup sebagian besar kasus penggunaan. Jika batas yang lebih tinggi diperlukan, mintalah peningkatan kuota dengan menghubungi AWS Support. Buka halaman Pusat AWS Support, masuk jika perlu, dan pilih Buat kasus. Pilih Peningkatan batas layanan. Lengkapi dan kirimkan formulir ini.

Note

Kuota ini tidak dapat diubah di konsol Service Quotas Amazon RDS.

Batasan penamaan dalam Amazon RDS

Tabel berikut menjelaskan batasan penamaan dalam Amazon RDS.

Sumber daya atau item	Batasan
Pengidentifikasi instans DB	Pengidentifikasi memiliki batasan penamaan ini: • Harus berisi 1–63 karakter alfanumerik atau tanda hubung. • Karakter pertamanya harus berupa huruf. • Tidak boleh diakhiri dengan satu tanda hubung atau berisi dua tanda hubung berturut-turut. • Harus unik untuk semua instans DB per AWS akun, per AWS Wilayah.

Sumber daya atau item	Batasan
Nama Basis data	Batasan nama basis data berbeda untuk setiap mesin basis data . Untuk informasi selengkapnya, lihat pengaturan yang tersedia saat membuat masing-masing instans DB. **Note** Pendekatan ini tidak berlaku untuk SQL Server. Untuk SQL Server, Anda membuat basis data setelah membuat instans DB.
Nama pengguna utama	Batasan nama pengguna utama berbeda untuk setiap mesin basis data. Untuk informasi selengkapnya, lihat pengaturan yang tersedia saat membuat masing-masing instans DB.
Kata sandi master	Kata sandi untuk pengguna utama basis data dapat mencakup karakter ASCII yang dapat dicetak kecuali /, ', ", @, atau spasi. Untuk Oracle, & adalah batasan karakter tambahan. Kata sandi memiliki jumlah karakter ASCII yang dapat dicetak berikut ini, bergantung pada mesin DB: • Db2: 8–255 • MariaDB dan MySQL: 8–41 • Oracle: 8–30 • SQL Server dan PostgreSQL: 8–128

Sumber daya atau item	Batasan
Nama grup parameter DB	Nama-nama ini memiliki batasan berikut: • Harus berisi 1–255 karakter alfanumerik. • Karakter pertamanya harus berupa huruf. • Tanda hubung diperbolehkan, tetapi nama tidak boleh diakhiri dengan tanda hubung atau berisi dua tanda hubung berturut-turut.
Nama kelompok subnet DB	Nama-nama ini memiliki batasan berikut: • Harus berisi 1–255 karakter. • Karakter alfanumerik, spasi, tanda hubung, garis bawah, dan titik diperbolehkan.

Jumlah maksimum koneksi basis data

Jumlah maksimum koneksi basis data simultan bervariasi berdasarkan jenis mesin DB dan alokasi memori untuk kelas instans DB. Jumlah maksimum koneksi umumnya diatur dalam grup parameter yang terkait dengan instans DB. Pengecualiannya adalah Microsoft SQL Server, yang diatur di properti server untuk instans DB di SQL Server Management Studio (SSMS).

Koneksi basis data mengonsumsi memori. Mengatur salah satu parameter ini terlalu tinggi dapat menyebabkan kondisi memori rendah yang dapat menyebabkan instans DB ditempatkan di status incompatible-parameters. Untuk informasi selengkapnya, lihat Mendiagnosis dan menyelesaikan status parameter yang tidak kompatibel untuk batas memori.

Jika aplikasi Anda sering membuka dan menutup koneksi, atau membiarkan sejumlah besar koneksi berumur panjang tetap terbuka, sebaiknya Anda menggunakan Proksi Amazon RDS. Proksi RDS adalah proksi basis data yang sepenuhnya terkelola dengan ketersediaan tinggi yang menggunakan pooling koneksi untuk berbagi koneksi basis data dengan aman dan efisien. Untuk mempelajari selengkapnya tentang Proksi RDS, lihat Menggunakan Proksi Amazon RDS.

> **Note**
>
> Untuk Oracle, Anda akan mengatur jumlah maksimum proses pengguna serta sesi pengguna dan sistem.

Untuk Db2, Anda tidak dapat mengatur koneksi maksimum. Batasnya adalah 64000.

Koneksi basis data maksimum

Mesin DB	Parameter	Nilai yang diizinkan	Nilai default	Deskripsi
MariaDB dan MySQL	`max_connections`	1–100000	Default untuk semua versi MariaDB dan MySQL kecuali untuk MariaDB versi 10.5 dan 10.6: {DB InstanceClassMemory /12582880} Default untuk MariaDB versi 10.5 dan 10.6: PALING SEDIKIT ({DB InstanceClassMemory /25165760} ,12000) **Note** Dalam kedua kasus tersebut, jika penghitungan nilai default menghasilkan nilai lebih dari 16.000, Amazon RDS menetapkan batas menjadi 16.000 untuk instans DB MariaDB dan MySQL.	Jumlah koneksi klien simultan yang diperbolehkan

Mesin DB	Parameter	Nilai yang diizinkan	Nilai default	Deskripsi
Oracle	`processes`	80–20000	PALING SEDIKIT ({DB InstanceClassMemory /9868951}, 20000)	Proses pengguna
	`sessions`	100–65535	–	Sesi pengguna dan sistem
PostgreSQL	`max_connections`	6–8388607	PALING SEDIKIT ({DB InstanceClassMemory /9531392}, 5000)	Jumlah maksimum koneksi serentak
SQL Server	Jumlah maksimum koneksi serentak	0–32767	0 (tidak terbatas)	Jumlah maksimum koneksi serentak

`DBInstanceClassMemory` dalam byte. Untuk detail tentang bagaimana nilai ini dihitung, lihat Menentukan parameter DB. Karena memori dicadangkan untuk sistem operasi dan proses manajemen RDS, ukuran memori ini lebih kecil dari nilai dalam gibibyte (GiB), seperti yang ditunjukkan di Spesifikasi perangkat keras kelas instans DB .

Misalnya, beberapa kelas instans DB memiliki memori 8 GiB, yaitu 8.589.934.592 byte. Untuk instans DB MySQL yang berjalan pada kelas instans DB dengan memori 8 GiB, seperti db.m7g.large, persamaan yang menggunakan memori total adalah `8589934592/12582880=683`. Namun, variabel `DBInstanceClassMemory` otomatis mengurangi jumlah yang dicadangkan ke sistem operasi dan proses RDS yang mengelola instans DB. Sisa pengurangannya kemudian dibagi 12.582.880. Penghitungan ini menghasilkan sekitar 630 untuk nilai `max_connections` bukan 683. Nilai ini tergantung pada kelas instans DB dan mesin DB.

Ketika instans DB MariaDB atau MySQL berjalan pada kelas instans DB kecil, seperti db.t3.micro atau db.t3.small, total memori yang tersedia rendah. Untuk kelas instans DB ini, RDS menyimpan sebagian besar memori yang tersedia, yang memengaruhi nilai `max_connections`. Misalnya, jumlah koneksi maksimum default untuk instans DB MySQL yang berjalan pada kelas instans DB db.t3.micro adalah sekitar 60. Anda dapat menentukan nilai `max_connections` untuk instans DB MariaDB atau MySQL Anda dengan menghubungkannya dan menjalankan perintah SQL berikut:

```
SHOW GLOBAL VARIABLES LIKE 'max_connections';
```

Batas ukuran file di Amazon RDS

Batas ukuran file berlaku untuk instans DB Amazon RDS tertentu. Untuk informasi selengkapnya, lihat batas khusus mesin berikut:

- Batas ukuran file MariaDB di Amazon RDS
- Batas ukuran file MySQL di Amazon RDS
- Batas ukuran file Oracle di Amazon RDS

Pemecahan Masalah untuk Amazon RDS

Gunakan bagian berikut untuk membantu memecahkan masalah yang Anda miliki dengan instans DB di Amazon RDS dan Amazon Aurora.

Topik

Untuk informasi tentang masalah debug menggunakan API Amazon RDS, lihat Pemecahan masalah aplikasi di Amazon RDS.

Tidak dapat terhubung ke instans DB Amazon RDS

Jika Anda tidak dapat terhubung ke instans DB, berikut adalah penyebab-penyebab umumnya:

- Aturan masuk – Aturan akses yang diberlakukan oleh firewall lokal dan alamat IP yang diizinkan untuk mengakses instans DB mungkin tidak cocok. Kemungkinan besar masalahnya adalah aturan masuk dalam grup keamanan Anda.

  Secara default, instans DB tidak mengizinkan akses. Akses diberikan melalui grup keamanan yang terkait dengan VPC yang mengizinkan lalu lintas masuk dan keluar dari instans DB. Jika perlu, tambahkan aturan masuk dan keluar untuk situasi khusus Anda ke grup keamanan. Anda dapat menentukan alamat IP, rentang alamat IP, atau grup keamanan VPC lainnya.

> **Note**
>
> Saat menambahkan aturan masuk baru, Anda dapat memilih IP Saya untuk Sumber agar dapat mengizinkan akses ke instans DB dari alamat IP yang terdeteksi di browser.

Untuk informasi selengkapnya tentang menyiapkan grup keamanan, lihat Memberikan akses ke instans DB di VPC Anda dengan membuat grup keamanan.

> **Note**
>
> Koneksi klien dari alamat IP di dalam rentang 169.254.0.0/16 tidak diizinkan. Rentang ini adalah Automatic Private IP Addressing Range (APIPA) yang digunakan untuk alamat tautan lokal.

- Aksesibilitas publik – Untuk terhubung ke instans DB dari luar VPC, seperti menggunakan aplikasi klien, instans harus memiliki alamat IP publik yang ditetapkan untuk instans tersebut.

  Agar instans dapat diakses oleh publik, ubah instans dan pilih Ya di bagian Aksesibilitas publik. Untuk informasi selengkapnya, lihat Menyembunyikan klaster DB dalam VPC dari internet.
- Port – Port yang Anda tentukan saat membuat instans DB tidak dapat digunakan untuk mengirim atau menerima komunikasi karena batasan firewall lokal Anda. Untuk menentukan apakah jaringan Anda memungkinkan port tertentu digunakan untuk komunikasi masuk dan keluar, hubungi administrator jaringan Anda.
- Ketersediaan – Untuk instans DB yang baru dibuat, instans DB memiliki status `creating` hingga instans DB siap digunakan. Ketika statusnya berubah menjadi `available`, Anda dapat terhubung ke instans DB. Tergantung pada ukuran instans DB Anda, perlu waktu hingga 20 menit sebelum instans tersedia.
- Gateway internet – Agar instans DB dapat diakses publik, subnet dalam grup subnet DB tersebut harus memiliki gateway internet.

  Mengonfigurasi gateway internet untuk subnet

  1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di https://console.aws.amazon.com/rds/.
  2. Di panel navigasi, pilih Basis Data lalu pilih instans DB.

3. Di tab Konektivitas & keamanan, tuliskan nilai ID VPC di VPC dan subnet ID di Subnet.
4. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
5. Di panel navigasi, pilih Gateway Internet. Pastikan ada gateway internet yang dilampirkan ke VPC Anda. Atau, pilih Buat Gateway Internet untuk membuat gateway internet. Pilih gateway internet, lalu pilih Lampirkan ke VPC dan ikuti arahan untuk melampirkannya ke VPC Anda.
6. Di panel navigasi, pilih Subnet, lalu pilih subnet Anda.
7. Di tab Tabel Rute, pastikan ada rute dengan `0.0.0.0/0` sebagai tujuan dan gateway internet untuk VPC sebagai target.

   Jika Anda terhubung ke instans Anda menggunakan alamat IPv6, pastikan ada rute untuk semua lalu lintas IPv6 (`::/0`) yang mengarah ke gateway internet. Jika tidak, lakukan tindakan berikut:

   a. Pilih ID tabel rute (rtb-xxxxxxxx) untuk menavigasi ke tabel rute.
   b. Di tab Rute, pilih Edit rute. Pilih Tambahkan rute, gunakan `0.0.0.0/0` sebagai tujuan, dan gateway internet sebagai target.

      Untuk IPv6, pilih Tambahkan rute, gunakan `::/0` sebagai tujuan, dan gateway internet sebagai target.
   c. Pilih Simpan rute.

   Selain itu, jika Anda mencoba untuk terhubung ke titik akhir IPv6, pastikan rentang alamat IPv6 klien diizinkan untuk terhubung ke instans DB.

Untuk informasi selengkapnya, lihat Bekerja dengan klaster DB dalam VPC.

Untuk masalah koneksi khusus mesin, lihat topik berikut:

- Memecahkan masalah koneksi ke instans DB SQL Server Anda
- Memecahkan masalah koneksi ke instans Oracle DB Anda
- Memecahkan masalah koneksi ke instans RDS for PostgreSQL Anda
- Koneksi maksimum MySQL dan MariaDB

Menguji koneksi ke instans DB

Anda dapat menguji koneksi ke instans DB menggunakan alat Linux atau Microsoft Windows umum.

Dari terminal Linux atau Unix, Anda dapat menguji koneksi dengan memasukkan hal berikut. Ganti *DB-instance-endpoint* dengan titik akhir dan *port* dengan port instans DB Anda.

```
nc -zv DB-instance-endpoint port
```

Misalnya, hal berikut menunjukkan contoh perintah dan nilai kembali.

```
nc -zv postgresql1.c6c8mn7fake0.us-west-2.rds.amazonaws.com 8299

  Connection to postgresql1.c6c8mn7fake0.us-west-2.rds.amazonaws.com 8299 port [tcp/
vvr-data] succeeded!
```

Pengguna Windows dapat menggunakan Telnet untuk menguji koneksi ke instans DB. Tindakan Telnet tidak didukung selain untuk menguji koneksi. Jika koneksi berhasil, tindakan tidak akan mengembalikan pesan. Jika koneksi gagal, Anda akan menerima pesan kesalahan seperti berikut.

```
C:\>telnet sg-postgresql1.c6c8mntfake0.us-west-2.rds.amazonaws.com 819

  Connecting To sg-postgresql1.c6c8mntfake0.us-west-2.rds.amazonaws.com...Could not
 open
  connection to the host, on port 819: Connect failed
```

Jika tindakan Telnet berhasil, artinya grup keamanan Anda dikonfigurasi dengan benar.

> **Note**
>
> Amazon RDS tidak menerima lalu lintas Internet Control Message Protocol (ICMP), termasuk ping.

Memecahkan masalah autentikasi koneksi

Dalam beberapa kasus, Anda dapat terhubung ke instans DB tetapi mendapatkan kesalahan autentikasi. Dalam kasus ini, Anda mungkin ingin mengatur ulang kata sandi pengguna utama untuk instans DB. Anda dapat melakukan tindakan ini dengan mengubah instans RDS.

Untuk informasi selengkapnya tentang cara mengubah instans DB, lihat Memodifikasi instans DB Amazon RDS.

Masalah keamanan Amazon RDS

Untuk menghindari masalah keamanan, jangan pernah menggunakan nama AWS pengguna dan kata sandi utama Anda untuk akun pengguna. Praktik terbaik adalah menggunakan master Anda Akun AWS untuk membuat pengguna dan menetapkannya ke akun pengguna DB. Anda juga dapat menggunakan akun utama untuk membuat akun pengguna lain, jika perlu.

Untuk informasi tentang membuat pengguna, lihat Membuat pengguna IAM di Akun AWS. Untuk informasi tentang membuat pengguna AWS IAM Identity Center, lihat Mengelola identitas di Pusat Identitas IAM.

Pesan kesalahan “gagal mengambil atribut akun, fungsi konsol tertentu mungkin terganggu.”

Kesalahan ini dapat muncul karena beberapa alasan. Penyebabnya mungkin karena akun Anda kehilangan izin, atau akun Anda belum disiapkan dengan benar. Untuk akun baru, Anda mungkin belum menunggu akun hingga siap digunakan. Untuk akun lama, Anda mungkin tidak memiliki izin dalam kebijakan akses untuk melakukan tindakan tertentu, seperti membuat instans DB. Untuk memecahkan masalah ini, administrator perlu memberikan peran yang diperlukan ke akun Anda. Untuk informasi selengkapnya, lihat dokumentasi IAM.

Memecahkan masalah status jaringan yang tidak kompatibel

Status jaringan yang tidak kompatibel berarti basis datanya mungkin masih dapat diakses di tingkat basis data, tetapi Anda tidak dapat mengubah atau melakukan boot ulang pada basis data tersebut.

Penyebab

Status jaringan instans DB Anda yang tidak kompatibel dapat disebabkan oleh salah satu tindakan berikut:

- Mengubah kelas instans DB.
- Mengubah instans DB untuk menggunakan deployment klaster DB Multi-AZ.
- Mengganti host karena peristiwa pemeliharaan.
- Meluncurkan instans DB pengganti.

- Memulihkan dari pencadangan snapshot.
- Memulai instans DB yang telah dihentikan.

Penyelesaian

Gunakan start-db-instance perintah

Untuk memperbaiki basis data yang berada dalam status jaringan yang tidak kompatibel, ikuti petunjuk ini:

1. Buka https://console.aws.amazon.com/rds/ dan pilih Basis Data dari panel navigasi.
2. Pilih instans DB yang berada dalam status jaringan yang tidak kompatibel dan catat pengidentifikasi instans DB, ID VPC, dan ID subnet dari tab Konektivitas & Keamanan.
3. Gunakan AWS CLI untuk menjalankan `start-db-instance` perintah. Tentukan nilai `--db-instance-identifier`.

> **Note**
>
> Menjalankan perintah ini ketika basis data Anda dalam mode yang tidak kompatibel dapat menyebabkan beberapa waktu henti.
> Perintah `start-db-instance` tidak menyelesaikan masalah ini untuk instans DB RDS for SQL Server.

Status basis data Anda berubah menjadi Tersedia jika perintah berhasil dijalankan.

Jika basis data Anda dimulai ulang, instans DB mungkin menjalankan operasi terakhir yang dijalankan pada instans tersebut sebelum dipindahkan ke status jaringan yang tidak kompatibel. Tindakan ini mungkin memindahkan kembali instans ke status jaringan yang tidak kompatibel.

Jika perintah `start-db-instance` tidak berhasil atau instans berpindah kembali ke status jaringan yang tidak kompatibel, buka halaman Basis Data di konsol RDS dan pilih basis datanya. Arahkan ke bagian Log & peristiwa. Bagian Peristiwa terbaru menampilkan langkah-langkah penyelesaian lebih lanjut untuk diikuti. Pesan-pesan tersebut diklasifikasikan sebagai berikut:

- PEMERIKSAAN SUMBER DAYA INTERNAL: Mungkin ada masalah dengan sumber daya internal Anda.
- PEMERIKSAAN DNS: Periksa penyelesaian DNS dan nama host untuk VPC di konsol VPC.

- Pemeriksaan ENI: antarmuka jaringan elastis (ENI) untuk basis data Anda mungkin tidak ada.
- PEMERIKSAAN GATEWAY: Gateway internet untuk basis data Anda yang tersedia untuk umum tidak dilampirkan ke VPC.
- PEMERIKSAAN IP: Tidak ada alamat IP gratis di subnet Anda.
- PEMERIKSAAN GRUP KEAMANAN: Tidak ada grup keamanan yang dikaitkan dengan basis data Anda atau grup keamanan tidak valid.
- PEMERIKSAAN SUBNET: Tidak ada subnet yang valid di grup subnet DB Anda atau ada masalah dengan subnet Anda.
- PEMERIKSAAN VPC: VPC yang dikaitkan dengan basis data Anda tidak valid.

Lakukan point-in-time pemulihan

Memiliki cadangan (snapshot atau logis) merupakan praktik terbaik, jika basis data Anda memasuki status jaringan yang tidak kompatibel. Lihat Pengantar cadangan. Jika Anda mengaktifkan cadangan otomatis, maka hentikan sementara penulisan apa pun ke database dan lakukan pemulihan. point-in-time

> **Note**
>
> Setelah suatu instans memasuki status jaringan yang tidak kompatibel, instans DB mungkin tidak dapat diakses untuk melakukan pencadangan logis.

Jika Anda tidak mengaktifkan pencadangan otomatis, buat instans DB baru. Kemudian, migrasikan data menggunakan AWS Database Migration Service (AWS DMS), atau dengan menggunakan alat pencadangan dan pemulihan.

Jika ini tidak menyelesaikan masalah, hubungi AWS Support untuk bantuan lebih lanjut.

Mengatur ulang kata sandi pemilik instans DB

Jika Anda terkunci dari instans DB, Anda dapat masuk sebagai pengguna utama. Kemudian, Anda dapat mengatur ulang kredensial untuk pengguna administratif atau peran lainnya. Jika Anda tidak dapat masuk sebagai pengguna utama, pemilik AWS akun dapat mengatur ulang kata sandi pengguna utama. Untuk detail tentang akun administratif atau peran yang mungkin perlu Anda atur ulang, lihat Hak akses akun pengguna master.

Anda dapat mengubah kata sandi instans DB dengan menggunakan konsol Amazon RDS, AWS CLI perintah modify-db-instance, atau dengan menggunakan operasi ModifydBInstance API. Untuk informasi selengkapnya tentang cara mengubah instans DB, lihat Memodifikasi instans DB Amazon RDS.

Penghentian atau boot ulang instans DB Amazon RDS

Penghentian instans DB dapat terjadi ketika instans DB di-boot ulang. Penghentian ini juga dapat terjadi saat instans DB diubah menjadi status yang mencegah akses ke instans tersebut, dan saat basis data di-boot ulang. Boot ulang dapat terjadi saat Anda melakukan boot ulang manual pada instans DB Anda. Boot ulang juga dapat terjadi saat Anda mengubah pengaturan instans DB yang memerlukan boot ulang sebelum dapat diberlakukan.

Boot ulang instans DB terjadi saat Anda mengubah pengaturan yang memerlukan boot ulang, atau ketika Anda secara manual menyebabkan boot ulang. Boot ulang dapat segera terjadi jika Anda mengubah pengaturan dan meminta perubahan segera diberlakukan. Atau, boot ulang dapat terjadi selama jendela pemeliharaan instans DB.

Boot ulang instans DB segera terjadi ketika salah satu hal berikut terjadi:

- Anda mengubah periode retensi pencadangan untuk instans DB dari 0 ke nilai selain nol atau dari nilai selain nol ke 0. Anda mengatur Terapkan Segera ke `true`.
- Anda mengubah kelas instans DB, dan Terapkan Segera diatur menjadi `true`.
- Anda mengubah jenis penyimpanan dari Magnetis (Standar) ke Tujuan Umum (SSD) atau IOPS yang Tersedia (SSD), atau dari IOPS yang Tersedia (SSD) atau Tujuan Umum (SSD) ke Magnetis (Standar).

Boot ulang instans DB terjadi selama jendela pemeliharaan saat salah satu dari hal berikut terjadi:

- Anda mengubah periode retensi pencadangan untuk instans DB dari 0 ke nilai selain nol atau dari nilai selain nol ke 0, dan Terapkan Segera diatur ke `false`.
- Anda mengubah kelas instans DB, dan Terapkan Segera diatur menjadi `false`.

Ketika Anda mengubah parameter statis dalam grup parameter DB, perubahan tersebut tidak diberlakukan hingga instans DB yang dikaitkan dengan grup parameter di-boot ulang. Perubahan ini memerlukan boot ulang manual. Instans DB tidak di-boot ulang secara otomatis selama jendela pemeliharaan.

Untuk melihat tabel yang menunjukkan tindakan instans DB dan dampak dari pengaturan nilai Terapkan Segera, lihat Memodifikasi instans DB Amazon RDS.

Perubahan parameter DB Amazon RDS tidak diberlakukan

Dalam beberapa kasus, Anda mungkin mengubah parameter dalam grup parameter DB, tetapi tidak melihat perubahan akan diberlakukan. Jika demikian, Anda mungkin perlu melakukan boot ulang instans DB yang dikaitkan dengan grup parameter DB. Saat Anda mengubah parameter dinamis, perubahan akan langsung diberlakukan. Ketika Anda mengubah parameter statis, perubahan tersebut tidak diberlakukan hingga Anda melakukan boot ulang instans DB yang dikaitkan dengan grup parameter.

Anda dapat melakukan boot ulang pada instans DB menggunakan konsol RDS. Atau, Anda dapat secara eksplisit memanggil operasi API `RebootDBInstance`. Anda dapat mem-boot ulang tanpa failover jika instans DB ada dalam deployment Multi-AZ. Persyaratan untuk melakukan boot ulang pada instans DB terkait setelah perubahan parameter statis membantu memitigasi risiko kesalahan konfigurasi parameter yang memengaruhi panggilan API. Contohnya adalah memanggil `ModifyDBInstance` untuk mengubah kelas instans DB. Untuk informasi selengkapnya, lihat Memodifikasi parameter dalam grup parameter DB.

Instans DB Amazon RDS kehabisan ruang penyimpanan

Jika instans DB Anda kehabisan ruang penyimpanan, instans tersebut mungkin tidak lagi tersedia. Kami sangat menyarankan agar Anda terus memantau `FreeStorageSpace` metrik yang diterbitkan CloudWatch untuk memastikan bahwa instans DB Anda memiliki ruang penyimpanan gratis yang cukup.

Jika instans basis data Anda kehabisan ruang penyimpanan, statusnya berubah menjadi `storage-full`. Sebagai contoh, panggilan ke operasi API `DescribeDBInstances` untuk instans DB yang telah menghabiskan output penyimpanannya berikut.

```
aws rds describe-db-instances --db-instance-identifier mydbinstance

DBINSTANCE  mydbinstance  2009-12-22T23:06:11.915Z  db.m5.large  mysql8.0  50  sa
storage-full  mydbinstance.clla4j4jgyph.us-east-1.rds.amazonaws.com  3306
us-east-1b  3
 SECGROUP  default  active
 PARAMGRP  default.mysql8.0  in-sync
```

Untuk memulihkan dari skenario ini, tambahkan lebih banyak ruang penyimpanan ke instans Anda menggunakan operasi `ModifyDBInstance` API atau AWS CLI perintah berikut.

Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --allocated-storage 60 \
    --apply-immediately
```

Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --allocated-storage 60 ^
    --apply-immediately
```

```
DBINSTANCE  mydbinstance  2009-12-22T23:06:11.915Z  db.m5.large  mysql8.0  50  sa
storage-full  mydbinstance.clla4j4jgyph.us-east-1.rds.amazonaws.com  3306
us-east-1b  3  60
 SECGROUP  default  active
 PARAMGRP  default.mysql8.0  in-sync
```

Kini, saat Anda menggambarkan instans DB, Anda melihat bahwa instans DB Anda memiliki status `modifying`, yang menunjukkan bahwa penyimpanan sedang diskalakan.

```
aws rds describe-db-instances --db-instance-identifier mydbinstance
```

```
DBINSTANCE  mydbinstance  2009-12-22T23:06:11.915Z  db.m5.large  mysql8.0  50  sa
modifying  mydbinstance.clla4j4jgyph.us-east-1.rds.amazonaws.com
3306  us-east-1b  3  60
 SECGROUP  default  active
 PARAMGRP  default.mysql8.0  in-sync
```

Setelah penskalaan penyimpanan selesai, status instans DB Anda berubah menjadi `available`.

```
aws rds describe-db-instances --db-instance-identifier mydbinstance
```

```
DBINSTANCE  mydbinstance  2009-12-22T23:06:11.915Z  db.m5.large  mysql8.0  60  sa
available  mydbinstance.clla4j4jgyph.us-east-1.rds.amazonaws.com  3306
```

```
us-east-1b  3
 SECGROUP  default  active
 PARAMGRP  default.mysql8.0  in-sync
```

Anda dapat menerima notifikasi ketika ruang penyimpanan Anda habis menggunakan operasi `DescribeEvents`. Misalnya, dalam skenario ini, jika Anda membuat panggilan `DescribeEvents` setelah operasi ini, Anda akan melihat output berikut.

```
aws rds describe-events --source-type db-instance --source-identifier mydbinstance
```

```
2009-12-22T23:44:14.374Z  mydbinstance  Allocated storage has been exhausted db-
instance
2009-12-23T00:14:02.737Z  mydbinstance  Applying modification to allocated storage db-
instance
2009-12-23T00:31:54.764Z  mydbinstance  Finished applying modification to allocated
 storage
```

Kapasitas instans DB tidak cukup untuk Amazon RDS

Kesalahan `InsufficientDBInstanceCapacity` dapat muncul saat Anda mencoba membuat, memulai, atau mengubah instans DB. Kesalahan ini juga dapat muncul saat Anda mencoba memulihkan instans DB dari snapshot DB. Saat kesalahan ini muncul, satu penyebab umumnya adalah kelas instans DB tertentu tidak tersedia di Zona Ketersediaan yang diminta. Anda dapat mencoba salah satu hal berikut untuk memecahkan masalahnya:

- Coba kembali permintaan dengan kelas instans DB yang berbeda.
- Coba kembali permintaan dengan Zona Ketersediaan yang berbeda.
- Coba kembali permintaan tanpa menentukan Zona Ketersediaan eksplisit.

Untuk informasi tentang pemecahan masalah kapasitas instans untuk Amazon EC2, lihat Kapasitas instans tidak mencukupi di Panduan Pengguna Amazon EC2.

Untuk informasi tentang cara mengubah instans DB, lihat Memodifikasi instans DB Amazon RDS.

Masalah memori yang dapat dikosongkan di Amazon RDS

Memori yang dikosongkan adalah total Random Access Memory (RAM) pada instans DB yang dapat dibuat tersedia untuk mesin basis data. Ini adalah jumlah dari memori sistem operasi (OS) yang

kosong serta memori cache halaman dan buffer yang tersedia. Mesin basis data menggunakan sebagian besar memori pada host, tetapi proses OS juga menggunakan sebagian RAM. Memori yang saat ini dialokasikan ke mesin basis data atau digunakan oleh proses OS tidak termasuk dalam memori yang dapat dikosongkan. Ketika mesin basis data kehabisan memori, instans DB dapat menggunakan ruang sementara yang biasanya digunakan untuk buffering dan caching. Seperti disebutkan sebelumnya, ruang sementara ini termasuk dalam memori yang dapat dikosongkan.

Anda menggunakan `FreeableMemory` metrik di Amazon CloudWatch untuk memantau memori yang dapat dibebaskan. Untuk informasi selengkapnya, lihat Ikhtisar metrik pemantauan di Amazon RDS.

Jika instans DB Anda secara konsisten kehabisan memori yang dapat dikosongkan atau menggunakan ruang swap, pertimbangkan untuk meningkatkan ke kelas instans DB yang lebih besar. Untuk informasi selengkapnya, lihat Kelas instans DB .

Anda juga dapat mengubah pengaturan memori. Misalnya, pada RDS for MySQL, Anda dapat menyesuaikan ukuran parameter `innodb_buffer_pool_size`. Parameter ini diatur secara default ke 75 persen memori fisik. Untuk tips pemecahan masalah MySQL lainnya, lihat Bagaimana cara memecahkan masalah memori yang dapat dikosongkan rendah di basis data Amazon RDS for MySQL?

Masalah MySQL dan MariaDB MySQL

Anda dapat mendiagnosis dan memperbaiki masalah dengan instans DB MySQL dan MariaDB.

Topik

- Koneksi maksimum MySQL dan MariaDB
- Mendiagnosis dan menyelesaikan status parameter yang tidak kompatibel untuk batas memori
- Mendiagnosis dan mengatasi jeda di antara replika baca
- Mendiagnosis dan menyelesaikan kegagalan replikasi baca MySQL atau MariaDB
- Membuat pemicu dengan log biner aktif memerlukan hak istimewa SUPER
- Mendiagnosis dan menyelesaikan kegagalan pemulihan point-in-time
- Kesalahan replikasi terhenti
- Pembuatan replika baca gagal atau replikasi rusak dengan kesalahan fatal 1236

Koneksi maksimum MySQL dan MariaDB

Jumlah koneksi maksimum yang diperbolehkan untuk instans DB RDS for MySQL atau RDS for MariaDB didasarkan pada jumlah memori yang tersedia untuk kelas instans DB. Kelas instans DB dengan memori lebih besar akan menghasilkan tersedianya koneksi yang lebih besar. Untuk informasi selengkapnya tentang kelas instans DB, lihat Kelas instans DB .

Batas koneksi untuk instans DB diatur secara default ke kelas instans maksimum DB. Anda dapat membatasi jumlah koneksi bersamaan dengan nilai berapa pun hingga jumlah maksimum koneksi yang diperbolehkan. Gunakan parameter `max_connections` dalam grup parameter untuk instans DB. Untuk informasi lebih lanjut, lihat Jumlah maksimum koneksi basis data dan Bekerja dengan grup parameter.

Anda dapat mengambil jumlah maksimum koneksi yang diperbolehkan untuk instans DB MySQL atau MariaDB dengan menjalankan kueri berikut.

```
SELECT @@max_connections;
```

Anda dapat mengambil jumlah maksimum koneksi yang aktif untuk instans DB MySQL atau MariaDB dengan menjalankan kueri berikut.

```
SHOW STATUS WHERE `variable_name` = 'Threads_connected';
```

Mendiagnosis dan menyelesaikan status parameter yang tidak kompatibel untuk batas memori

Instans DB MariaDB atau MySQL dapat ditempatkan dalam status incompatible-parameters untuk suatu batas memori saat kondisi berikut terpenuhi:

- Instans DB dimulai ulang setidaknya tiga kali dalam satu jam atau setidaknya lima kali dalam satu hari jika status instans DB Tersedia.
- Upaya untuk memulai ulang instans DB gagal karena tindakan pemeliharaan atau proses pemantauan tidak dapat memulai ulang instans DB.
- Penggunaan memori potensial instans DB melebihi 1,2 kali memori yang dialokasikan ke kelas instans DB.

Ketika instans DB dimulai ulang untuk ketiga kalinya dalam satu jam atau untuk kali kelima dalam satu hari, instans tersebut melakukan pemeriksaan penggunaan memori. Pemeriksaan ini membuat perhitungan penggunaan memori potensial instans DB. Nilai yang ditunjukkan oleh perhitungan tersebut adalah jumlah dari nilai berikut:

- Nilai 1 – Jumlah parameter berikut:
  - `innodb_additional_mem_pool_size`
  - `innodb_buffer_pool_size`
  - `innodb_log_buffer_size`
  - `key_buffer_size`
  - `query_cache_size` (Hanya MySQL versi 5.7)
  - `tmp_table_size`
- Nilai 2 – Parameter `max_connections` dikalikan dengan jumlah parameter berikut:
  - `binlog_cache_size`
  - `join_buffer_size`
  - `read_buffer_size`
  - `read_rnd_buffer_size`
  - `sort_buffer_size`
  - `thread_stack`
- Nilai 3 – Jika parameter `performance_schema` diaktifkan, kalikan parameter `max_connections` dengan `257700`.

  Jika parameter `performance_schema` dinonaktifkan, nilai ini adalah nol.

Jadi, nilai yang ditunjukkan oleh perhitungan adalah sebagai berikut:

```
Value 1 + Value 2 + Value 3
```

Ketika nilai ini melebihi 1,2 kali memori yang dialokasikan ke kelas instans DB yang digunakan oleh instans DB, instans DB ditempatkan dalam status incompatible-parameters. Untuk informasi tentang memori yang dialokasikan ke kelas instans DB, lihat Spesifikasi perangkat keras kelas instans DB .

Penghitungan tersebut mengalikan nilai parameter `max_connections` dengan jumlah beberapa parameter. Jika parameter `max_connections` diatur ke nilai yang besar, parameter ini dapat menyebabkan pemeriksaan menunjukkan nilai yang sangat tinggi untuk penggunaan memori

potensial dari instans DB. Dalam kasus ini, pertimbangkan untuk menurunkan nilai parameter `max_connections`.

Untuk mengatasi masalah, selesaikan langkah-langkah berikut:

1. Sesuaikan parameter memori dalam grup parameter DB yang dikaitkan dengan instans DB. Lakukan sedemikian rupa sehingga penggunaan memori potensial lebih rendah dari 1,2 kali memori yang dialokasikan ke kelas instans DB.

   Untuk informasi tentang mengatur parameter, lihat Memodifikasi parameter dalam grup parameter DB.
2. Mulai ulang instans DB.

   Untuk informasi tentang mengatur parameter, lihat Memulai instans DB Amazon RDS yang sebelumnya dihentikan.

Mendiagnosis dan mengatasi jeda di antara replika baca

Setelah Anda membuat replika baca MySQL atau MariaDB dan replikanya tersedia, Amazon RDS pertama-tama mereplikasi perubahan yang dibuat ke instans DB sumber sejak operasi replika baca dimulai. Selama fase ini, waktu jeda replikasi untuk replika baca lebih besar dari 0. Anda dapat memantau jeda waktu ini di Amazon CloudWatch dengan melihat `ReplicaLag`metrik Amazon RDS.

Metrik `ReplicaLag` melaporkan nilai kolom `Seconds_Behind_Master` dari perintah `SHOW REPLICA STATUS` MariaDB atau MySQL. Untuk informasi selengkapnya, lihat SHOW REPLICA STATUS Statement di dokumentasi MySQL.

Saat metrik `ReplicaLag` mencapai 0, replika telah menyamai instans DB sumber. Jika metrik `ReplicaLag` menunjukkan -1, replikasi mungkin tidak aktif. Untuk memecahkan masalah kesalahan replikasi, lihat Mendiagnosis dan menyelesaikan kegagalan replikasi baca MySQL atau MariaDB. Nilai `ReplicaLag` sebesar -1 juga dapat berarti bahwa nilai `Seconds_Behind_Master` tidak dapat ditentukan atau `NULL`.

> **Note**
>
> Versi sebelumnya dari MariaDB dan MySQL menggunakan `SHOW SLAVE STATUS`, bukan `SHOW REPLICA STATUS`. Jika Anda menggunakan versi MariaDB sebelum 10.5 atau versi MySQL sebelum 8.0.23, gunakan `SHOW SLAVE STATUS`.

Metrik `ReplicaLag` menunjukkan -1 saat penghentian jaringan atau saat patch diterapkan selama jendela pemeliharaan. Dalam kasus ini, tunggu konektivitas jaringan hingga dipulihkan atau tunggu jendela pemeliharaan berakhir sebelum Anda memeriksa metrik `ReplicaLag` lagi.

Teknologi replikasi baca MySQL dan MariaDB bersifat asinkron. Oleh karena itu, Anda dapat sesekali mengharapkan peningkatan bagi metrik `BinLogDiskUsage` pada instans DB sumber dan bagi metrik `ReplicaLag` pada replika baca. Misalnya, pertimbangkan situasi saat volume operasi tulis tinggi ke instans DB sumber terjadi secara paralel. Pada saat yang sama, operasi tulis ke replika baca akan diserialkan menggunakan thread I/O tunggal. Situasi tersebut dapat menyebabkan jeda antara instans sumber dan replika baca.

Untuk informasi selengkapnya tentang replika baca dan MySQL, lihat Replication implementation details dalam dokumentasi MySQL. Untuk informasi selengkapnya tentang replika baca dan MariaDB, lihat Replication overview di dokumentasi MariaDB.

Anda dapat mengurangi lag antara pembaruan ke instans DB sumber dan pembaruan berikutnya ke replika baca dengan melakukan hal berikut:

- Atur kelas instans DB dari replika baca agar memiliki ukuran penyimpanan yang sebanding dengan ukuran dari instans DB sumber.
- Pastikan kompatibilitas pengaturan parameter di grup parameter DB yang digunakan oleh instans DB sumber dan replika baca. Untuk informasi selengkapnya dan contoh, lihat diskusi tentang parameter `max_allowed_packet` di bagian berikutnya.
- Nonaktifkan cache kueri. Untuk tabel yang sering diubah, menggunakan cache kueri dapat meningkatkan lag replika karena cache terkunci dan sering disegarkan. Dalam kasus ini, Anda mungkin akan melihat lebih sedikit lag replika jika menonaktifkan cache kueri. Anda dapat menonaktifkan cache kueri dengan mengatur `query_cache_type parameter` ke 0 dalam grup parameter DB untuk instans DB. Untuk informasi selengkapnya tentang cache kueri, lihat Konfigurasi cache kueri.
- Hangatkan kumpulan buffer pada replika baca untuk InnoDB for MySQL atau MariaDB. Misalnya, anggaplah Anda memiliki sejumlah kecil tabel yang sering diperbarui dan Anda menggunakan skema tabel InnoDB atau XtraDB. Dalam kasus ini, dump tabel tersebut pada replika baca. Dengan melakukan hal ini, Anda akan menyebabkan mesin basis data memindai barisan tabel tersebut dari disk, lalu menyimpannya di dalam kumpulan buffer. Pendekatan ini dapat mengurangi jeda replika. Bagian berikut menunjukkan satu contoh.

  Untuk Linux, macOS, atau Unix:

```
PROMPT> mysqldump \
    -h <endpoint> \
    --port=<port> \
    -u=<username> \
    -p <password> \
    database_name table1 table2 > /dev/null
```

Untuk Windows:

```
PROMPT> mysqldump ^
    -h <endpoint> ^
    --port=<port> ^
    -u=<username> ^
    -p <password> ^
    database_name table1 table2 > /dev/null
```

Mendiagnosis dan menyelesaikan kegagalan replikasi baca MySQL atau MariaDB

Amazon RDS memantau status replikasi replika baca Anda. RDS memperbarui bidang Status Replikasi instans replika baca menjadi `Error` jika replikasi berhenti karena alasan apa pun. Anda dapat meninjau detail kesalahan terkait yang dilontarkan oleh mesin MySQL atau MariaDB dengan melihat kolom Kesalahan Replikasi. Peristiwa yang menunjukkan status replika baca juga dihasilkan, termasuk RDS-EVENT-0045, RDS-EVENT-0046, dan RDS-EVENT-0057. Untuk informasi selengkapnya tentang peristiwa dan berlangganan peristiwa, lihat Menggunakan pemberitahuan peristiwa Amazon RDS. Jika pesan kesalahan MySQL muncul, periksa kesalahannya di MySQL error message documentation. Jika pesan kesalahan MariaDB muncul, periksa kesalahannya di MySQL error message documentation.

Situasi umum yang dapat menyebabkan kesalahan replikasi mencakup hal-hal berikut:

- Nilai parameter `max_allowed_packet` untuk replika baca lebih kecil dari parameter `max_allowed_packet` untuk instans DB sumber.

  Parameter `max_allowed_packet` adalah parameter kustom yang dapat Anda atur di grup parameter DB. Parameter `max_allowed_packet` digunakan untuk menentukan ukuran maksimum bahasa manipulasi data (DML) yang dapat dijalankan di basis data. Dalam beberapa

kasus, nilai `max_allowed_packet` untuk instans DB sumber mungkin lebih besar dari nilai `max_allowed_packet` untuk replika baca. Jika demikian, proses replikasi dapat menimbulkan kesalahan dan menghentikan replikasi. Kesalahan yang paling umum adalah `packet bigger than 'max_allowed_packet' bytes`. Anda dapat memperbaiki kesalahan ini dengan mengatur agar replika sumber dan baca menggunakan grup parameter DB yang sama dengan nilai parameter `max_allowed_packet`.

- Menulis ke tabel di replika baca. Jika Anda membuat indeks pada replika baca, parameter `read_only` harus diatur ke 0 untuk membuat indeks. Jika Anda menulis ke tabel di replika baca, tindakan ini dapat memecah replikasi.
- Gunakan mesin penyimpanan nontransaksional seperti MyISAM. Replika baca membutuhkan mesin penyimpanan transaksional. Replikasi hanya didukung untuk mesin penyimpanan berikut: InnoDB for MySQL atau MariaDB.

  Anda dapat mengonversi tabel MyISAM ke InnoDB dengan perintah berikut:

  ```
  alter table <schema>.<table_name> engine=innodb;
  ```
- Gunakan kueri nondeterministik yang tidak aman seperti `SYSDATE()`. Untuk informasi selengkapnya, lihat Determination of safe and unsafe statements in binary logging di dokumentasi MySQL.

Langkah-langkah berikut dapat membantu mengatasi kesalahan replikasi Anda:

- Jika Anda mengalami kesalahan logis dan dapat melewatkan kesalahan tersebut dengan aman, ikuti langkah-langkah yang dijelaskan dalam Melewati kesalahan replikasi saat ini. Instans DB MySQL atau MariaDB Anda harus menjalankan versi yang mencakup prosedur `mysql_rds_skip_repl_error`. Untuk informasi selengkapnya, lihat mysql.rds_skip_repl_error.
- Jika mengalami masalah posisi log biner (binlog), Anda dapat mengubah posisi tayangan ulang replika dengan perintah `mysql_rds_next_master_log`. Instans DB MySQL atau MariaDB Anda harus menjalankan versi yang mendukung perintah `mysql_rds_next_master_log` untuk mengubah posisi tayangan ulang replika. Untuk informasi versi, lihat mysql.rds_next_master_log.
- Anda mungkin mengalami masalah performa sementara karena beban DML yang tinggi. Jika demikian, Anda dapat mengatur parameter `innodb_flush_log_at_trx_commit` ke 2 di grup parameter DB pada replika baca. Dengan melakukan hal ini, Anda dapat membantu replika baca mengejar, meskipun tindakan ini akan mengurangi atomisitas, konsistensi, isolasi, dan daya tahan (ACID) untuk sementara.

- Anda dapat menghapus replika baca dan membuat instans menggunakan pengidentifikasi instans DB yang sama. Jika Anda melakukannya, titik akhir tetap sama dengan replika baca lama Anda.

Jika kesalahan replikasi diperbaiki, Status Replikasi berubah menjadi mereplikasi. Untuk informasi selengkapnya, lihat Pemecahan Masalah batasan replika baca MySQL.

Membuat pemicu dengan log biner aktif memerlukan hak istimewa SUPER

Saat mencoba membuat pemicu di instans DB RDS for MySQL atau RDS for MariaDB, Anda mungkin menerima kesalahan berikut.

```
"You do not have the SUPER privilege and binary logging is enabled"
```

Untuk menggunakan pemicu saat pencatatan log biner diaktifkan memerlukan hak istimewa SUPER, yang dibatasi untuk instans DB RDS for MySQL dan RDS for MariaDB. Anda dapat membuat pemicu saat log biner diaktifkan tanpa hak istimewa SUPER dengan mengatur parameter `log_bin_trust_function_creators` ke true. Untuk mengatur `log_bin_trust_function_creators` menjadi true, buat grup parameter DB baru atau ubah grup parameter DB yang sudah ada.

Anda dapat membuat grup parameter DB baru sehingga Anda dapat membuat pemicu di instans DB RDS for MySQL atau RDS for MariaDB dengan log biner yang aktif. Untuk melakukannya, gunakan perintah CLI berikut. Untuk mengubah grup parameter yang ada, mulailah dengan langkah 2.

Membuat grup parameter baru untuk mengizinkan pemicu dengan log biner yang aktif menggunakan CLI

1. Buat grup parameter baru.

   Untuk Linux, macOS, atau Unix:

   ```
   aws rds create-db-parameter-group \
       --db-parameter-group-name allow-triggers \
       --db-parameter-group-family mysql8.0 \
       --description "parameter group allowing triggers"
   ```

   Untuk Windows:

   ```
   aws rds create-db-parameter-group ^
   ```

```
    --db-parameter-group-name allow-triggers ^
    --db-parameter-group-family mysql8.0 ^
    --description "parameter group allowing triggers"
```

2. Ubah grup parameter DB untuk mengizinkan pemicu.

   Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name allow-triggers \
    --parameters "ParameterName=log_bin_trust_function_creators,
 ParameterValue=true, ApplyMethod=pending-reboot"
```

   Untuk Windows:

```
aws rds modify-db-parameter-group ^
    --db-parameter-group-name allow-triggers ^
    --parameters "ParameterName=log_bin_trust_function_creators,
 ParameterValue=true, ApplyMethod=pending-reboot"
```

3. Ubah instans DB Anda untuk menggunakan grup parameter DB yang baru.

   Untuk Linux, macOS, atau Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --db-parameter-group-name allow-triggers \
    --apply-immediately
```

   Untuk Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --db-parameter-group-name allow-triggers ^
    --apply-immediately
```

4. Agar perubahan diberlakukan, lakukan boot ulang pada instans DB secara manual.

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

Mendiagnosis dan menyelesaikan kegagalan pemulihan point-in-time

Memulihkan Instans DB yang Mencakup Tabel Sementara

Saat mencoba point-in-time memulihkan (PITR) instance MySQL atau MariaDB DB Anda, Anda mungkin mengalami kesalahan berikut.

```
Database instance could not be restored because there has been incompatible database
 activity for restore
functionality. Common examples of incompatible activity include using temporary tables,
 in-memory tables,
or using MyISAM tables. In this case, use of Temporary table was detected.
```

PITR mengandalkan snapshot cadangan dan log biner (binlog) dari MySQL atau MariaDB untuk memulihkan instans DB Anda ke waktu tertentu. Informasi tabel sementara mungkin tidak dapat diandalkan di binlog dan dapat menyebabkan kegagalan PITR. Jika Anda menggunakan tabel sementara di instans DB MySQL atau MariaDB, Anda dapat menurunkan kemungkinan kegagalan PITR. Untuk melakukannya, lakukan pencadangan yang lebih sering. Kegagalan PITR paling mungkin terjadi dalam waktu antara pembuatan tabel sementara dan snapshot cadangan berikutnya.

Memulihkan Instans DB yang Mencakup Tabel Dalam Memori

Anda mungkin mengalami masalah saat memulihkan basis data yang memiliki tabel dalam memori. Tabel dalam-memori dibersihkan selama proses mulai ulang. Sebagai hasilnya, tabel dalam memori Anda mungkin kosong setelah boot ulang. Kami merekomendasikan agar Anda merancang solusi untuk menangani tabel kosong jika proses mulai ulang terjadi saat menggunakan tabel dalam memori. Jika Anda menggunakan tabel dalam memori dengan instans DB yang direplikasi, Anda mungkin perlu membuat replika baca setelah memulai ulang. Hal ini mungkin diperlukan jika replika baca melakukan boot ulang dan tidak dapat memulihkan data dari tabel dalam memori yang kosong.

Untuk informasi selengkapnya tentang pencadangan dan PITR, lihat Pengantar cadangan dan Memulihkan instans DB dengan waktu yang ditentukan.

Kesalahan replikasi terhenti

Ketika memanggil perintah `mysql.rds_skip_repl_error`, Anda mungkin menerima pesan kesalahan yang menyatakan bahwa replikasi mati atau dinonaktifkan.

Pesan kesalahan ini muncul karena replikasi dihentikan dan tidak dapat dimulai ulang.

Jika Anda perlu melewati sejumlah besar kesalahan, lag replikasi dapat meningkat hingga melampaui periode retensi default untuk file log biner. Dalam kasus ini, Anda mungkin mengalami kesalahan fatal karena file log biner dihapus sebelum diputar ulang di replika. Penghapusan ini menyebabkan replikasi berhenti, dan Anda tidak dapat lagi memanggil perintah `mysql.rds_skip_repl_error` untuk melewati kesalahan replikasi.

Anda dapat memitigasi masalah ini dengan meningkatkan jumlah jam penyimpanan file log biner pada sumber replikasi Anda. Setelah meningkatkan waktu retensi binlog, Anda dapat memulai ulang replikasi dan memanggil perintah `mysql.rds_skip_repl_error` sesuai kebutuhan.

Untuk mengatur waktu retensi binlog, gunakan prosedur mysql.rds_set_configuration. Tentukan parameter konfigurasi 'jam retensi binlog' sekaligus jumlah jam untuk menyimpan file binlog di klaster DB, hingga 720 (30 hari). Contoh berikut menetapkan periode penyimpanan file binlog menjadi 48 jam.

```
CALL mysql.rds_set_configuration('binlog retention hours', 48);
```

Pembuatan replika baca gagal atau replikasi rusak dengan kesalahan fatal 1236

Setelah mengubah nilai parameter default untuk instans DB MySQL atau MariaDB, Anda mungkin mengalami salah satu masalah berikut:

- Anda tidak dapat membuat replika baca untuk instans DB.
- Replikasi gagal dengan `fatal error 1236`.

Beberapa nilai parameter default untuk instans DB MySQL dan MariaDB membantu memastikan bahwa basis data tersebut sesuai dengan ACID dan replika baca aman dari crash. Hal ini dilakukan dengan memastikan setiap commit disinkronkan sepenuhnya dengan menulis transaksi ke log biner sebelum di-commit. Mengubah parameter ini dari nilai default untuk meningkatkan performa dapat menyebabkan replikasi gagal ketika transaksi belum ditulis ke log biner.

Untuk mengatasi masalah ini, atur nilai parameter berikut:

- `sync_binlog = 1`
- `innodb_support_xa = 1`
- `innodb_flush_log_at_trx_commit = 1`

Tidak dapat mengatur periode retensi cadangan menjadi 0

Ada beberapa alasan mengapa Anda mungkin perlu mengatur periode retensi cadangan menjadi 0. Misalnya, Anda dapat langsung menonaktifkan pencadangan otomatis dengan mengatur periode retensi menjadi 0.

Dalam beberapa kasus, Anda mungkin menetapkan nilai ke 0 dan menerima pesan yang mengatakan bahwa jangka waktu penyimpanan harus antara 1 dan 35. Dalam kasus ini, periksa untuk memastikan bahwa Anda belum menyiapkan replika baca untuk instans. Replika baca memerlukan cadangan untuk mengelola log replika baca sehingga Anda tidak dapat mengatur periode penyimpanan sebesar 0.

Referensi API Amazon RDS

Selain AWS Management Console dan AWS Command Line Interface (AWS CLI), Amazon RDS juga menyediakan API. Anda dapat menggunakan API untuk mengotomatiskan tugas untuk mengelola instans DB dan objek lain di Amazon RDS.

- Untuk daftar abjad operasi API, lihat Tindakan.
- Untuk daftar abjad jenis data, lihat Jenis data.
- Untuk daftar parameter kueri umum, lihat Parameter umum.
- Untuk deskripsi kode kesalahan, lihat Kesalahan umum.

Untuk informasi selengkapnya tentang AWS CLI, lihat Referensi AWS Command Line Interface untuk Amazon RDS.

Topik

- Menggunakan API Kueri
- Pemecahan masalah aplikasi di Amazon RDS

Menggunakan API Kueri

Bagian berikut membahas secara singkat parameter dan autentikasi permintaan yang digunakan untuk API Kueri.

Untuk informasi umum tentang cara kerja API Kueri, lihat Permintaan kueri dalam Referensi API Amazon EC2.

Parameter kueri

Permintaan berbasis Kueri HTTP adalah permintaan HTTP yang menggunakan kata kerja HTTP GET atau POST dan parameter Kueri yang bernama `Action`.

Setiap permintaan Kueri harus menyertakan beberapa parameter umum untuk menangani autentikasi dan pemilihan tindakan.

Beberapa operasi mengambil daftar parameter. Daftar ini ditentukan menggunakan notasi `param.`*n*. Nilai *n* adalah integer yang dimulai dari 1.

Untuk informasi tentang Wilayah dan titik akhir Amazon RDS, buka Amazon Relational Database Service (RDS) di bagian Wilayah dan Titik Akhir dalam Referensi Umum Amazon Web.

Autentikasi permintaan Kueri

Anda hanya dapat mengirim permintaan Kueri melalui HTTPS, dan Anda harus menyertakan signature di setiap permintaan Kueri. Anda harus menggunakan AWS Signature versi 4 atau Signature versi 2. Untuk informasi selengkapnya, lihat Proses penandatanganan Signature Versi 4 dan Proses penandatanganan Signature versi 2.

Pemecahan masalah aplikasi di Amazon RDS

Amazon RDS memberikan penjelasan tentang kesalahan spesifik dan deskriptif untuk membantu Anda memecahkan masalah saat menangani API Amazon RDS.

Topik

- Kesalahan pengambilan
- Tips penyelesaian masalah

Untuk informasi tentang pemecahan masalah untuk instans DB Amazon RDS, lihat Pemecahan Masalah untuk Amazon RDS.

Kesalahan pengambilan

Biasanya, Anda ingin aplikasi Anda memeriksa apakah permintaan menimbulkan kesalahan sebelum Anda menghabiskan waktu untuk memproses hasil. Cara termudah untuk mengetahui jika terjadi kesalahan adalah dengan mencari simpul `Error` di dalam respons dari API Amazon RDS.

Sintaks XPath menyediakan cara sederhana untuk mencari keberadaan simpul `Error`. Sintaks XPath juga menyediakan cara yang relatif mudah untuk mengambil kode kesalahan dan pesan. Cuplikan kode berikut menggunakan Perl dan modul XML::XPath untuk menentukan apakah kesalahan terjadi selama permintaan. Jika terjadi kesalahan, kode akan mencetak kode kesalahan pertama dan pesan dalam tanggapannya.

```
use XML::XPath;
    my $xp = XML::XPath->new(xml =>$response);
    if ( $xp->find("//Error") )
    {print "There was an error processing your request:\n", " Error code: ",
```

```
    $xp->findvalue("//Error[1]/Code"), "\n", " ",
    $xp->findvalue("//Error[1]/Message"), "\n\n"; }
```

Tips penyelesaian masalah

Sebaiknya lakukan proses berikut untuk mendiagnosis dan menyelesaikan masalah dengan API Amazon RDS.

- Verifikasi bahwa Amazon RDS beroperasi secara normal di Wilayah AWS yang Anda targetkan dengan memeriksa http://status.aws.amazon.com.
- Periksa struktur permintaan Anda.

  Setiap operasi Amazon RDS memiliki halaman referensi di Referensi API Amazon RDS. Periksa ulang bahwa Anda menggunakan parameter dengan benar. Untuk mengetahui kemungkinan kesalahan, lihat contoh permintaan atau skenario pengguna untuk melihat apakah contoh tersebut melakukan operasi serupa.
- Periksa AWS re: Post.

  Amazon RDS memiliki komunitas pengembangan tempat Anda dapat mencari solusi untuk masalah yang dialami orang lain selama ini. Untuk melihat topik, buka AWS re:Post.

Riwayat dokumen

Versi API saat ini: 2014-10-31

Tabel berikut menjelaskan perubahan penting dalam setiap rilis Panduan Pengguna Amazon RDS setelah Mei 2018. Untuk notifikasi tentang pembaruan dokumentasi ini, Anda dapat berlangganan ke umpan RSS.

> **Note**
>
> Anda dapat memfilter fitur Amazon RDS baru di halaman Apa yang Baru dengan Basis Data. Untuk Produk, pilih Amazon RDS. Kemudian, cari menggunakan kata kunci seperti **`RDS Proxy`** atau **`Oracle 2023`**.

Perubahan	Deskripsi	Tanggal
Amazon RDS Custom untuk Oracle mendukung kelas instans DB db.x2iezn	Anda sekarang dapat menggunakan kelas instance db.x2iezn untuk RDS Custom untuk instans Oracle DB. Untuk informasi selengkapnya, lihat Dukungan kelas instans DB untuk RDS Custom for Oracle.	Maret 26, 2024
Amazon RDS mendukung kelas instans db.c6gd untuk cluster DB multi-AZ	Anda sekarang dapat menggunakan kelas instans db.c6gd untuk penerapan cluster DB multi-AZ. Untuk informasi selengkapnya, lihat Ketersediaan kelas instans untuk klaster DB multi-AZ.	Maret 21, 2024
Dukungan Amazon RDS yang Diperluas	Membuat atau memulihkan database RDS untuk MySQL 5.7 atau RDS untuk	Maret 21, 2024

	PostgreSQL 11 sekarang secara otomatis mendaftar kan database tersebut ke Amazon RDS Extended Support sehingga aplikasi Anda yang ada terus berfungsi sebagaimana adanya. Anda dapat memilih keluar dari RDS Extended Support untuk menghindari biaya setelah RDS berakhir tanggal dukungan standar untuk mesin database Anda. Untuk informasi selengkapnya, lihat Menggunakan Dukungan Amazon RDS yang Diperluas.	
RDS untuk integrasi Db2 dengan AWS License Manager	RDS untuk Db2 sekarang terintegrasi dengan. AWS License Manager Jika Anda menggunakan model Bring Your Own License, AWS License Manager integrasi membantu dalam memantau penggunaan lisensi Db2 Anda dalam organisasi Anda. Untuk informasi selengkapnya, lihat Mengintegrasikan dengan AWS License Manager.	Maret 20, 2024

Rotasi sertifikat CA untuk cluster DB multi-AZ	Anda sekarang dapat memutar sertifikat CA untuk cluster DB multi-AZ Anda. Pertimbangkan untuk menggunakan salah satu sertifikat CA baru rds-ca-rsa 2048-g1, rds-ca-rsa 4096-g1, atau 384-g1. rds-ca-ecc Untuk informasi selengkapnya, lihat Memutar sertifikat SSL/TLS Anda.	Maret 6, 2024
Amazon RDS mendukung penyimpanan io2 Block Express	Anda sekarang dapat membuat instance RDS DB yang menggunakan tipe penyimpanan io2 Block Express. Untuk informasi lebih lanjut, lihat penyimpanan io2 Block Express.	Maret 6, 2024
RDS Kustom untuk SQL Server mendukung kelas instans db.r5b dan db.x2iedn DB	Anda sekarang dapat menggunakan kelas instans db.r5b dan db.x2iedn untuk RDS Custom untuk instans SQL Server DB. Untuk informasi selengkapnya, lihat dukungan kelas instans DB untuk RDS Custom for SQL Server.	Maret 4, 2024
RDS Custom for Oracle tersedia di Wilayah Timur Tengah (UEA)	Anda dapat membuat RDS Custom untuk instans Oracle DB di Wilayah Timur Tengah (UEA). Untuk tabel yang menampilkan semua yang didukung Wilayah AWS, lihat RDS Custom for Oracle.	Maret 4, 2024

Kebijakan AWS terkelola baru	Amazon RDS menambahkan kebijakan terkelola baru yang diberi nama `AmazonRDSCustomInstanceProfileRolePolicy` untuk memungkinkan RDS Custom melakukan tindakan otomatisasi dan tugas manajemen database melalui profil instans EC2. Untuk informasi selengkapnya, lihat Pembaruan Amazon RDS terhadap kebijakan terkelola AWS.	Februari 27, 2024
Amazon RDS mendukung MariaDB 10.11.7, 10.6.17, 10.5.24, dan 10.4.33	Anda sekarang dapat membuat instans Amazon RDS DB yang menjalankan MariaDB versi 10.11.7, 10.6.17, 10.5.24, dan 10.4.33. Untuk informasi selengkapnya, lihat Versi MariaDB on Amazon RDS.	Februari 26, 2024
Cluster Amazon RDS Multi-AZ DB mendukung volume penyimpanan Amazon EBS gp3	Cluster DB multi-AZ sekarang mendukung volume EBS berbasis SSD gp3. Untuk informasi lebih lanjut, lihat penyimpanan gp3.	Februari 26, 2024
Dukungan Amazon RDS untuk AWS Secrets Manager di Wilayah Israel (Tel Aviv)	Amazon RDS mendukung Secrets Manager di Wilayah Israel (Tel Aviv). Untuk informasi selengkapnya, lihat Manajemen kata sandi dengan Amazon RDS dan AWS Secrets Manager.	Februari 21, 2024

Amazon RDS untuk Db2 mendukung pencatatan audit	RDS untuk Db2 sekarang mendukung pencatatan audit tingkat database. Saat Anda mengaktifkan pencatatan audit untuk database RDS untuk Db2, Amazon RDS mencatat aktivitas database dan menyimpan log audit di Amazon S3. Untuk informasi selengkapnya, lihat pencatatan audit Db2.	Februari 15, 2024
Dukungan Amazon RDS yang Diperluas	Amazon RDS sekarang secara otomatis mengaktifkan Amazon RDS Extended Support ketika RDS untuk MySQL dan RDS untuk versi mesin utama PostgreSQL di instans DB Anda dan cluster DB multi-AZ mencapai akhir RDS dari tanggal dukungan standar. Untuk informasi selengkapnya, lihat Menggunakan Dukungan Amazon RDS yang Diperluas.	Februari 15, 2024
Amazon RDS mendukung MySQL 8.0.36	Anda sekarang dapat membuat instans Amazon RDS DB yang menjalankan MySQL versi 8.0.36. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	Februari 12, 2024

Amazon RDS mendukung pemeriksaan EBCDIC untuk RDS untuk Db2	Anda sekarang dapat membuat database Db2 yang menggunakan urutan pemeriksaan EBCDIC untuk mengurutkan konten dalam database. Untuk informasi selengkapnya, lihat pemeriksaan EBCDIC untuk database Db2 di Amazon RDS.	Januari 29, 2024
Perbarui ke Sertifikat CA default	Sertifikat CA default diatur `rds-ca-rsa2048-g1` ke . Untuk informasi selengkapnya, lihat Menggunakan SSL/TLS untuk mengenkripsi koneksi ke instans DB.	Januari 26, 2024
Amazon RDS for PostgreSQL mendukung dua peti baru untuk PL/Rust, croaring-rs dan num-bigint	Anda dapat menggunakan dua peti baru di Amazon RDS untuk PostgreSQL. Untuk informasi lebih lanjut, lihat Menggunakan peti dengan PL/karat.	Januari 24, 2024
Amazon RDS untuk PostgreSQL mendukung TLS versi 1.3	Anda dapat menggunakan Transport Layer Security (TLS) versi 1.3 di RDS untuk PostgreSQL. Untuk informasi selengkapnya, lihat Menggunakan SSL dengan instans PostgreSQL DB.	Januari 24, 2024

Kustom RDS untuk SQL Server mendukung Microsoft SQL Server 2022	Anda sekarang dapat membuat RDS Custom untuk instans SQL Server DB yang menggunakan SQL Server 2022. Untuk informasi selengkapnya, lihat Bekerja dengan Kustom RDS untuk SQL Server.	Januari 22, 2024
Memperbarui ke izin kebijakan AWS terkelola	`AmazonRDSServiceRolePolicy` Peran `AWSServiceRoleForRDS` terkait layanan memiliki ID pernyataan baru. Untuk informasi selengkapnya, lihat Pembaruan Amazon RDS terhadap kebijakan terkelola AWS.	Januari 19, 2024
RDS Custom for Oracle mendukung Wilayah Eropa (Paris)	Anda dapat membuat RDS Custom untuk instans Oracle DB di Wilayah Eropa (Paris). Untuk informasi selengkapnya, lihat RDS Custom for Oracle.	Januari 18, 2024
Amazon RDS for MySQL mendukung replikasi multi-sumber	Anda sekarang dapat menggunakan replikasi multi-sumber pada RDS untuk instance MySQL DB. Untuk informasi selengkapnya, lihat Mengonfigurasi replikasi multi-sumber di RDS untuk MySQL.	Januari 16, 2024

Amazon RDS mendukung MySQL 8.2 di lingkungan Pratinjau Database	MySQL 8.2 sekarang tersedia di lingkungan Pratinjau Database di AS Timur (Ohio). Wilayah AWS Untuk informasi selengkapnya, lihat MySQL versi 8.2 di lingkungan Pratinjau Database.	Januari 11, 2024
RDS Proxy tersedia di Wilayah Eropa (Spanyol)	RDS Proxy sekarang tersedia di wilayah Eropa (Spanyol) . Untuk informasi selengkapnya tentang Proksi RDS, lihat Menggunakan Proksi Amazon RDS.	8 Januari 2024
Amazon RDS tersedia di Wilayah Kanada Barat (Calgary)	Amazon RDS sekarang tersedia di Wilayah Kanada Barat (Calgary). Untuk informasi selengkapnya, lihat Wilayah dan Zona Ketersediaan.	Desember 20, 2023
Amazon RDS untuk Db2 mendukung 5.000 pengguna lokal	Anda sekarang dapat menambahkan hingga 5.000 pengguna lokal ke daftar otorisasi. Untuk informasi selengkapnya, lihat rdsadmin.add_user.	Desember 20, 2023

Amazon RDS mendukung melihat dan menanggapi rekomendasi	Rekomendasi Amazon RDS sekarang mencakup rekomendasi reaktif berbasis proaktif dan pembelajaran mesin berbasis ambang batas untuk RDS untuk PostgreSQL. Untuk informasi selengkapnya, lihat Melihat dan menanggapi rekomendasi Amazon RDS.	Desember 19, 2023
Amazon RDS mendukung MariaDB 10.11.6, 10.6.16, 10.5.23, dan 10.4.32	Anda sekarang dapat membuat instans Amazon RDS DB yang menjalankan MariaDB versi 10.11.6, 10.6.16, 10.5.23, dan 10.4.32. Untuk informasi selengkapnya, lihat Versi MariaDB on Amazon RDS.	Desember 12, 2023
Amazon RDS memperkenalkan integrasi nol-ETL dengan Amazon Redshift (pratinjau)	Integrasi nol-ETL memberikan solusi yang dikelola sepenuhnya untuk menyediakan data transaksional di Amazon Redshift dalam hitungan detik setelah ditulis ke RDS untuk instans DB MySQL. Untuk informasi selengkapnya, lihat Menggunakan integrasi nol-ETL Amazon RDS dengan Amazon Redshift (pratinjau).	28 November 2023
Amazon RDS mendukung mesin basis data IBM Db2	Anda sekarang dapat menjalankan mesin basis data IBM Db2 di Amazon RDS. Untuk informasi selengkapnya, lihat Amazon RDS for Db2.	27 November 2023

RDS for PostgreSQL mendukung peningkatan versi mayor ke PostgreSQL 16.1 serta peningkatan versi minor ke 15.5, 14.10, 13.13, 12.17, dan 11.22	Dengan RDS for PostgreSQL, Anda sekarang dapat meningkatkan mesin DB ke versi mayor 16.1 serta meningkatkan versi minor ke 15.5, 14.10, 13.13, 12.17, dan 11.22. Untuk informasi selengkapnya, lihat Meningkatkan mesin DB PostgreSQL untuk Amazon RDS.	17 November 2023
RDS Custom for Oracle mendukung grup opsi	Anda dapat membuat atau memodifikasi grup opsi dan mengaitkannya dengan instans DB RDS Custom for Oracle. Opsi `Timezone` sekarang didukung. Untuk informasi selengkapnya, lihat Menggunakan grup opsi di RDS Custom for Oracle.	17 November 2023
Amazon RDS for MySQL mendukung plugin Grup Replikasi	Anda sekarang dapat mengatur klaster aktif-aktif dengan instans DB RDS for MySQL versi 8.0.35 atau yang lebih tinggi dengan menggunakan plugin Replikasi Grup yang dikembangkan dan dikelola oleh komunitas MySQL. Untuk informasi selengkapnya, lihat Mengonfigurasi klaster aktif-aktif untuk RDS for MySQL.	17 November 2023

Proksi Amazon RDS mendukung RDS for PostgreSQL 16.1	Anda sekarang dapat membuat proksi menggunakan Proksi RDS untuk instans DB RDS for PostgreSQL 16.1. Untuk informasi selengkapnya, lihat Menggunakan Proksi Amazon RDS.	17 November 2023
RDS Custom for SQL Server mendukung Microsoft SQL Server 2019 Developer Edition	Anda dapat membuat instans DB RDS for SQL Server yang menggunakan SQL Server 2019 Developer Edition. Untuk informasi selengkapnya, lihat Bawa Media Anda Sendiri dengan RDS Custom for SQL Server.	16 November 2023
Peningkatan versi minor klaster DB Multi-AZ dengan waktu henti minimal	Saat Anda melakukan peningkatan versi minor klaster DB Multi-AZ, Amazon RDS sekarang meningkatkan instans DB pembaca sebelum instans penulis, sehingga secara signifikan mengurangi waktu henti. Anda dapat mengurangi waktu henti menjadi satu detik atau kurang dengan menggunakan Proksi RDS. Untuk informasi selengkapnya, lihat Meningkatkan versi mesin klaster DB Multi-AZ.	16 November 2023

RDS for SQL Server mendukung Microsoft SQL Server 2022	Anda sekarang dapat membuat instans DB RDS yang menggunakan SQL Server 2022. Untuk informasi selengkapnya, lihat Versi Microsoft SQL Server di Amazon RDS.	15 November 2023
RDS for MySQL mendukung peningkatan snapshot dari versi 5.7 ke 8.0	Anda sekarang dapat meningkatkan versi mesin RDS for MySQL dari versi 5.7 ke versi 8.0. Anda dapat melakukannya dengan menggunakan AWS Management Console, atau `ModifyDBSnapshot` pengoperasian API RDS atau AWS CLI. Untuk informasi selengkapnya, lihat Meningkatkan versi mesin snapshot MySQL DB.	15 November 2023
RDS Custom for SQL Server mendukung pemulihan titik waktu terhadap 1.000 basis data	Anda sekarang dapat membuat hingga 1.000 basis data yang memenuhi syarat untuk pencadangan penuh dan pemulihan titik waktu pada instans DB RDS Custom for SQL Server Anda. Untuk informasi selengkapnya, lihat Memulihkan instans RDS Custom for SQL Server ke suatu titik waktu.	15 November 2023

RDS Custom for SQL Server mendukung penggunaan Kunci Master Layanan	RDS Custom for SQL Server sekarang mendukung penggunaan Kunci Master Layanan (SMK). SMK memungkinkan Anda mengenkripsi objek seperti kredensial, dan menggunakan fitur SQL Server seperti TDE dan enkripsi kolom. Untuk informasi selengkapnya, lihat Menggunakan Kunci Master Layanan dengan RDS Custom for SQL Server.	13 November 2023
Amazon RDS mendukung MySQL 8.1 di lingkungan Pratinjau Basis Data	MySQL 8.1 sekarang tersedia di lingkungan Pratinjau Database di AS Timur (Ohio). Wilayah AWS Untuk informasi selengkapnya, lihat MySQL versi 8.1 di lingkungan Pratinjau Basis Data.	10 November 2023
RDS mendukung MySQL 8.0.35 dan MySQL 5.7.44	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 8.0.35 dan 5.7.44. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	9 November 2023
Proksi RDS mendukung klaster DB Multi-AZ	Proksi RDS sekarang mendukung koneksi ke klaster DB Multi-AZ. Untuk informasi selengkapnya, lihat Menggunakan titik akhir Proksi Amazon RDS.	9 November 2023

RDS Custom for Oracle tersedia di AWS GovCloud (US) Regions	Amazon RDS sekarang tersedia di AWS GovCloud (US) Regions. Untuk informasi selengkapnya, lihat RDS Custom for Oracle.	9 November 2023
Amazon RDS Optimized Writes mendukung kelas instans DB db.m5	Amazon RDS Optimized Writes sekarang mendukung kelas instans DB db.m5. Untuk informasi selengkapnya, lihat Meningkatkan performa penulisan dengan Amazon RDS Optimized Writes for MariaDB dan Meningkatkan performa penulisan dengan Amazon RDS Optimized Writes for MySQL.	9 November 2023
Amazon RDS for Oracle mendukung konfigurasi multi-penghuni arsitektur CDB	Dengan fitur multi-penghuni RDS for Oracle, RDS memberikan arsitektur dan pengalaman multi-penghuni Oracle yang dikelola sepenuhnya untuk basis data Oracle Anda. Anda dapat menggunakan API RDS untuk membuat beberapa PDB, yang disebut basis data penghuni, dalam CDB. RDS menawarkan konfigurasi multi-penghuni arsitektur CDB sebagai alternatif dari konfigurasi penghuni tunggal lama. Untuk informasi selengkapnya, lihat Konfigurasi multi-penghuni arsitektur CDB.	8 November 2023

Amazon RDS mengekspor metrik Performance Insights ke Amazon CloudWatch	Performance Insights memungkinkan Anda mengekspor dasbor metrik yang telah dikonfigurasi sebelumnya atau kustom ke Amazon. CloudWatch Dasbor metrik yang diekspor tersedia untuk dilihat di konsol. CloudWatch Anda juga dapat mengekspor widget metrik Performance Insights yang dipilih dan melihat data metrik di konsol. CloudWatch Untuk informasi selengkapnya, lihat Mengekspor metrik Performance Insights ke. CloudWatch	8 November 2023
Amazon RDS Custom for Oracle memungkinkan Anda meningkatkan sistem operasi pada instans DB	Anda sekarang dapat meningkatkan basis data atau sistem operasi (OS) untuk instans DB RDS Custom for Oracle menggunakan perintah CLI `modify-db-instance` . Untuk informasi selengkapnya, lihat Meningkatkan instans DB untuk Amazon RDS Custom for Oracle.	7 November 2023
Proksi RDS mendukung Protokol yang Diperluas untuk RDS for PostgreSQL	Anda sekarang dapat menjalankan protokol kueri yang diperluas pada instans DB RDS for PostgreSQL. Untuk informasi selengkapnya, lihat Menggunakan Proksi Amazon RDS.	6 November 2023

Dukungan RDS for PostgreSQL untuk Deployment Blue/Green RDS	Anda sekarang dapat membuat deployment blue/green dari instans DB RDS for PostgreSQL. Untuk informasi selengkapnya, lihat Menggunakan Deployment Blue/Green Amazon RDS untuk pembaruan basis data.	26 Oktober 2023
Memperbarui ke kebijakan AWS terkelola	Kebijakan terkelola `AmazonRDSPerformanceInsightsReadOnly` dan `AmazonRDSPerformanceInsightsFullAccess` sekarang menyertakan `Sid` (ID pernyataan) sebagai pengidentifikasi dalam pernyataan kebijakan. Untuk informasi selengkapnya, lihat Pembaruan Amazon RDS terhadap kebijakan terkelola AWS.	23 Oktober 2023
RDS Custom for Oracle mendukung Wilayah Eropa (Milan)	Untuk informasi selengkapnya, lihat RDS Custom for Oracle.	23 Oktober 2023

Aktifkan RDS Optimized Writes pada basis data yang ada	Anda sekarang dapat mengaktifkan RDS Optimized Writes pada instans DB yang ada bahkan jika instans tersebut dibuat dengan versi mesin, kelas instans DB, atau konfigurasi sistem file yang tidak mendukung fitur ini. Untuk informasi selengkap nya, lihat Mengaktifkan RDS Optimized Writes pada basis data yang ada untuk RDS for MySQL, dan Mengaktifkan RDS Optimized Writes pada basis data yang ada untuk RDS for MariaDB.	19 Oktober 2023
Amazon RDS mendukung penggunaan volume log khusus (DLV).	Anda sekarang dapat menggunakan volume log khusus (DLV) RDS for MariaDB, RDS for MySQL, dan RDS for PostgreSQL. DLV ideal untuk basis data dengan penyimpanan besar yang dialokasikan, persyarat an I/O per detik (IOPS) tinggi, atau beban kerja yang sensitif terhadap latensi. Untuk informasi selengkapnya, lihat Menggunakan volume log khusus (DLV).	17 Oktober 2023

Amazon RDS for PostgreSQL, Amazon RDS for MySQL, dan Amazon RDS for MariaDB mendukung kelas instans DB yang baru	Anda dapat membuat instans DB Amazon RDS yang menjalankan PostgreSQL, MySQL, dan MariaDB yang menggunakan kelas instans DB db.m6.in, db.m6idn, db.r6.in, dan db.r6.idn. Untuk informasi selengkapnya, lihat Mesin DB yang didukung untuk semua kelas instans DB yang tersedia.	12 Oktober 2023
Amazon RDS for PostgreSQL mendukung pgactive	Ekstensi pgactive tersedia di Amazon RDS for PostgreSQL. Untuk informasi selengkapnya, lihat Menggunakan ekstensi PostgreSQL dengan Amazon RDS for PostgreSQL.	9 Oktober 2023
RDS Custom for Oracle tersedia di Wilayah Asia Pasifik (Jakarta)	Anda dapat membuat RDS Custom untuk instans Oracle DB di Wilayah Asia Pasifik (Jakarta). Untuk informasi selengkapnya, lihat RDS Custom for Oracle.	5 Oktober 2023
RDS Custom for SQL Server mendukung kolasi tingkat server baru	RDS Custom for SQL Server sekarang mendukung berbagai kolasi server, baik dalam pengodean tradisional maupun UTF-8, untuk lokalitas SQL_Latin, Jepang, Jerman, dan Arab. Untuk informasi selengkapnya, lihat Dukungan kolasi dan karakter untuk instans DB RDS Custom for SQL Server.	26 September 2023

Memperbarui ke izin kebijakan AWS terkelola	Peran `AWSServiceRoleForRDSCustom` terkait layanan memiliki izin baru yang memungkinkan RDS Custom membuat, memodifikasi, dan menghapus Aturan Terkelola. `AmazonRDSCustomServiceRolePolicy` EventBridge Untuk informasi selengkapnya, lihat Pembaruan Amazon RDS terhadap kebijakan terkelola AWS.	20 September 2023
Amazon RDS menerbitkan metrik penghitung Performance Insights ke Amazon CloudWatch	Fungsi matematika metrik DB_PERF_INSIGHTS di konsol CloudWatch memungkinkan Anda melakukan kueri Amazon RDS untuk metrik penghitung Performance Insights. Untuk informasi selengkapnya, lihat Membuat CloudWatch alarm untuk memantau Amazon RDS.	20 September 2023
Wawasan Performa mendukung statistik tingkat penyerapan untuk SQL Server	Jika Anda menggunakan Wawasan Performa, Anda dapat melihat statistik SQL baik pada tingkat pernyataan maupun penyerapan untuk Amazon RDS for SQL Server. Untuk informasi selengkapnya, lihat Menganalisis kueri yang berjalan di SQL Server.	18 September 2023

Amazon RDS for PostgreSQL, Amazon RDS for MySQL, dan Amazon RDS for MariaDB mendukung jenis kelas instans db.m6.id dan db.r6.id	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan PostgreSQL, MySQL, dan MariaDB yang menggunakan jenis kelas instans DB db.m6.id dan db.r6.id yang dioptimalkan untuk memori. Jenis ini menawarkan penyimpanan SSD berbasis NVMe lokal. Untuk informasi selengkapnya, lihat Mesin DB yang didukung untuk semua kelas instans DB yang tersedia.	11 September 2023
Dukungan peningkatan versi mayor untuk klaster DB Multi-AZ RDS for PostgreSQL	Anda sekarang dapat melakukan peningkatan versi mayor klaster DB Multi-AZ RDS for PostgreSQL Anda. Untuk informasi selengkapnya, lihat Meningkatkan versi mesin klaster DB Multi-AZ.	7 September 2023
Amazon RDS mendukung MariaDB 10.11.5, 10.6.15, 10.5.22, dan 10.4.31	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MariaDB versi 10.11.5, 10.6.15, 10.5.22, dan 10.4.31. Untuk informasi selengkapnya, lihat Versi MariaDB on Amazon RDS.	7 September 2023

Dukungan Amazon RDS yang Diperluas	Amazon RDS mengumumkan kemampuan mendatang untuk terus menjalankan versi mesin mayor RDS for MySQL dan RDS for PostgreSQL di instans DB Anda setelah tanggal akhir dukungan standar RDS. Untuk informasi selengkapnya, lihat Menggunakan Dukungan Amazon RDS yang Diperluas.	1 September 2023
RDS Custom mendukung tindakan memulai dan menghentikan instans DB RDS Custom for SQL Server	RDS Custom sekarang mendukung tindakan memulai dan menghentikan instans DB RDS Custom for SQL Server. Untuk informasi selengkapnya, lihat Memulai dan menghentikan instans DB RDS Custom for SQL Server.	31 Agustus 2023
Amazon RDS Optimized Writes mendukung kelas instans DB db.r5	Amazon RDS Optimized Writes sekarang mendukung kelas instans DB db.r5. Untuk informasi selengkapnya, lihat Meningkatkan performa penulisan dengan Amazon RDS Optimized Writes for MariaDB dan Meningkatkan performa penulisan dengan Amazon RDS Optimized Writes for MySQL.	31 Agustus 2023

Amazon RDS for Oracle mendukung peningkatan otomatis file zona waktu untuk CDB	Dengan opsi `TIMEZONE_FILE_AUTOUPGRADE`, Anda dapat meningkatkan file zona waktu saat ini ke versi terbaru pada basis data kontainer (CDB) RDS for Oracle Anda. Untuk informasi selengkapnya, lihat Peningkatan otomatis file zona waktu Oracle.	29 Agustus 2023
Amazon RDS Optimized Writes mendukung kelas instans DB db.m6g dan db.m6i	Amazon RDS Optimized Writes sekarang mendukung kelas instans db.m6g dan db.m6i DB. Untuk informasi selengkapnya, lihat Meningkatkan performa penulisan dengan Amazon RDS Optimized Writes for MariaDB dan Meningkatkan performa penulisan dengan Amazon RDS Optimized Writes for MySQL.	28 Agustus 2023
Amazon RDS mendukung MariaDB 10.11	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MariaDB versi 10.11. Untuk informasi selengkapnya, lihat Versi MariaDB on Amazon RDS.	21 Agustus 2023

Memperbarui ke izin kebijakan AWS terkelola	`AmazonRDSCustomServiceRolePolicy` untuk peran yang ditautkan layanan `AWSServiceRoleForRDSCustom` memiliki izin baru yang memungkinkan RDS Custom membuat antarmuka jaringan. Untuk informasi selengkapnya, lihat Pembaruan Amazon RDS terhadap kebijakan terkelola AWS.	18 Agustus 2023
Memperbarui ke izin kebijakan AWS terkelola	Kebijakan terkelola `AmazonRDSFullAccess` memiliki izin baru yang memungkinkan Anda membuat, melihat, dan menghapus laporan analisis performa untuk jangka waktu tertentu. Untuk informasi selengkapnya, lihat Pembaruan Amazon RDS terhadap kebijakan terkelola AWS.	17 Agustus 2023

Memperbarui ke izin kebijakan AWS terkelola	Penambahan izin baru ke kebijakan terkelola `AmazonRDSPerformanceInsightsReadOnly` dan penambahan kebijakan terkelola baru `AmazonRDSPerformanceInsightsFullAccess` memungkinkan Anda membuat laporan analisis beban DB untuk jangka waktu tertentu. Untuk informasi selengkapnya, lihat Pembaruan Amazon RDS terhadap kebijakan terkelola AWS.	16 Agustus 2023
Amazon RDS mendukung analisis performa untuk jangka waktu tertentu	Wawasan Performa memungkinkan Anda membuat dan melihat laporan analisis performa untuk jangka waktu tertentu. Laporan ini memberikan wawasan yang diidentifikasi dan rekomendasi untuk menyelesaikan masalah performa. Untuk informasi selengkapnya, lihat Menganalisis beban DB untuk jangka waktu tertentu.	16 Agustus 2023

Amazon RDS Custom for Oracle mendukung kelas instans DB db.r5b dan db.x2iedn	Anda sekarang dapat menggunakan kelas instans db.r5b dan db.x2iedn untuk instans DB RDS Custom for Oracle. Untuk informasi selengkapnya, lihat Dukungan kelas instans DB untuk RDS Custom for Oracle.	16 Agustus 2023
Amazon RDS Custom for Oracle mendukung kelas instans DB db.m6i, db.r6i, dan db.t3 DB	Anda sekarang dapat menggunakan kelas instans db.m6i, db.r6i, dan db.t3 untuk instans DB RDS Custom for Oracle. Untuk informasi selengkapnya, lihat Dukungan kelas instans DB untuk RDS Custom for Oracle.	15 Agustus 2023
Amazon RDS for PostgreSQL kini mendukung PostgreSQL versi 16 Beta 3 di lingkungan pratinjau basis data	PostgreSQL versi 16 Beta 3 sekarang tersedia di lingkungan pratinjau database di AS Timur (Ohio). Wilayah AWS Untuk informasi selengkapnya, lihat Menggunakan lingkungan pratinjau basis data.	11 Agustus 2023
Amazon RDS mendukung MySQL 8.0.34 dan 5.7.43	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 8.0.34 dan 5.7.43. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	9 Agustus 2023

RDS for SQL Server mendukung tampilan metrik OS untuk replika siaga	Anda sekarang dapat melihat metrik OS untuk replika siaga RDS for SQL Server. Untuk informasi selengkapnya, lihat Melihat metrik OS di konsol RDS.	3 Agustus 2023
RDS for Oracle mendukung Oracle Data Guard untuk CDB	RDS for Oracle mendukung replika baca Data Guard untuk basis data kontainer (CDB) 19c dan 21c Oracle Database. Anda dapat membuat, mengelola, dan mempromosikan replika baca di CDB, seperti yang Anda dapat lakukan di non-CDB, menggunakan API RDS yang ada. Untuk informasi selengkapnya, lihat Replika baca multi-penghuni.	1 Agustus 2023
Amazon RDS tersedia di Wilayah Israel (Tel Aviv)	Amazon RDS kini tersedia di Wilayah Israel (Tel Aviv). Untuk informasi selengkap nya, lihat Wilayah dan Zona Ketersediaan.	1 Agustus 2023
Amazon RDS mendukung Oracle APEX versi 23.1.v1	Anda dapat menggunakan APEX 23.1.v1 dengan Oracle Database 19c dan yang lebih tinggi. Untuk informasi selengkapnya, lihat Oracle Application Express.	26 Juli 2023

Amazon RDS Custom for Oracle mendukung SID Oracle non-default	Saat Anda membuat instans DB RDS Custom for Oracle menggunakan Oracle Database 19c, Anda dapat menentukan pengidentifikasi sistem Oracle non-default (SID Oracle). Nilai ini juga merupakan nama CDB. Untuk informasi selengkapnya, lihat Pertimbangan arsitektur multi-penghuni.	21 Juli 2023
RDS for SQL Server mendukung Direktori Aktif yang Dikelola Sendiri	Anda sekarang dapat menggunakan Direktori Aktif yang Dikelola Sendiri untuk langsung menggabungkan instans DB RDS for SQL Server Anda ke domain Microsoft Active Directory (AD) Anda. Domain AD yang dikelola sendiri dapat berada di on-premise atau di cloud. Untuk informasi selengkapnya, lihat Menggunakan Direktori Aktif yang Dikelola Sendiri.	7 Juli 2023

Dukungan replikasi logis PostgreSQL untuk klaster DB Multi-AZ	Anda sekarang dapat menggunakan replikasi logis PostgreSQL dengan klaster DB Multi-AZ Anda untuk mereplikasi dan menyinkronkan tabel individual daripada seluruh instans basis data. Untuk informasi selengkapnya, lihat Menggunakan replikasi logis PostgreSQL dengan klaster DB Multi-AZ.	6 Juli 2023
Amazon RDS for PostgreSQL kini mendukung PostgreSQL versi 16 Beta 2 di lingkungan pratinjau basis data	PostgreSQL versi 16 Beta 2 sekarang tersedia di lingkungan pratinjau database di AS Timur (Ohio). Wilayah AWS Untuk informasi selengkapnya, lihat Menggunakan lingkungan pratinjau basis data.	6 Juli 2023
Memperbarui ke izin kebijakan AWS terkelola	`AmazonRDSCustomServiceRolePolicy` untuk peran yang ditautkan layanan `AWSServiceRoleForRDSCustom` memiliki izin baru yang memungkinkan RDS Custom for Oracle menggunakan snapshot. Untuk informasi selengkapnya, lihat Pembaruan Amazon RDS terhadap kebijakan terkelola AWS.	23 Juni 2023

RDS mendukung MariaDB 10.6.14, 10.5.21, dan 10.4.30	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MariaDB versi 10.6.14, 10.5.21, dan 10.4.30. Untuk informasi selengkapnya, lihat Versi MariaDB on Amazon RDS.	22 Juni 2023
RDS mendukung MySQL 8.0.33 dan 5.7.42	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 8.0.33 dan 5.7.42. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	15 Juni 2023
RDS mendukung MariaDB 10.6.13, 10.5.20, 10.4.29, dan 10.3.39	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MariaDB versi 10.6.13, 10.5.20, 10.4.29, dan 10.3.39. Untuk informasi selengkap nya, lihat Versi MariaDB on Amazon RDS.	15 Juni 2023

RDS for Oracle mendukung ruang tabel yang dapat dipindahkan	Anda dapat memigrasikan data dari basis data Oracle on-premise ke instans DB RDS for Oracle menggunakan ruang tabel yang dapat dipindahkan. Teknik ini tidak memerlukan lisensi tambahan dan merupakan teknik migrasi dengan waktu henti paling singkat. Untuk informasi selengkapnya, lihat Bermigrasi menggunakan ruang tabel Oracle yang dapat dipindahkan.	15 Juni 2023
Amazon RDS mendukung Proksi RDS dengan RDS for MariaDB versi 10.6	Anda sekarang dapat membuat Proksi RDS dengan basis data RDS for MariaDB versi 10.6. Untuk informasi selengkapnya tentang Proksi RDS, lihat Menggunakan Proksi Amazon RDS.	15 Juni 2023
RDS Custom for SQL Server mendukung Bawa Media Anda Sendiri (BYOM)	Anda sekarang dapat membuat Versi Mesin Kustom (CEV) menggunakan media SQL Server Anda sendiri. Untuk informasi selengkapnya, lihat Bawa Media Anda Sendiri dengan RDS Custom for SQL Server.	8 Juni 2023

RDS for Oracle dapat mengonversi non-CDB ke CDB Oracle Database 19c	Jika instans DB Anda menjalankan Oracle Database 19c dengan RU April 2021 atau yang lebih tinggi, Anda dapat mengonversi non-CDB ke CDB (basis data kontainer). Setelah Anda mengonversi arsitektur, Anda dapat meningkatkan CDB 19c Anda ke CDB 21c. Langkah ini diperlukan karena Anda tidak dapat meningkatkan basis data Anda dan mengonversi arsitektur menggunakan perintah tunggal. Untuk informasi selengkapnya, lihat Mengonversi non-CDB menjadi CDB RDS for Oracle.	31 Mei 2023
Klaster DB Multi-AZ tersedia di Wilayah Tiongkok	Cluster DB multi-AZ sekarang tersedia di Wilayah AWS China (Beijing) dan China (Ningxia). Untuk informasi selengkapnya, lihat Klaster DB Multi-AZ.	30 Mei 2023

Amazon RDS Optimized Reads mendukung klaster DB Multi-AZ	Amazon RDS Optimized Reads sekarang mendukung klaster DB Multi-AZ. Untuk informasi selengkapnya, lihat Meningkatkan performa kueri untuk RDS for MySQL dengan Amazon RDS Optimized Reads dan Meningkatkan performa kueri untuk RDS for PostgreSQL dengan Amazon RDS Optimized Reads.	30 Mei 2023
RDS Custom for Oracle mendukung Wilayah Asia Pasifik (Jakarta)	Untuk informasi selengkapnya, lihat RDS Custom for Oracle.	29 Mei 2023
Buat replika baca instans DB dengan klaster DB Multi-AZ RDS for PostgreSQL sumber	Anda sekarang dapat membuat replika baca instans DB dengan klaster DB Multi-AZ RDS for PostgreSQL sebagai sumbernya. Sebelumnya, hanya RDS for MySQL yang didukung. Untuk informasi selengkapnya, lihat Membuat replika baca instans DB dari klaster DB Multi-AZ.	24 Mei 2023
Amazon RDS menyediakan gabungan Performance Insights CloudWatch dan metrik di dasbor Performance Insights	Amazon RDS kini menyediakan tampilan konsolidasi Performance Insights CloudWatch dan metrik di dasbor Performance Insights. Untuk informasi selengkapnya, lihat Melihat metrik gabungan di konsol Amazon RDS.	24 Mei 2023

Amazon RDS Optimized Reads tersedia di Wilayah Tiongkok	Amazon RDS Optimized Reads kini tersedia di Wilayah AWS Tiongkok (Beijing) dan Tiongkok (Ningxia). Untuk informasi selengkapnya, lihat Meningkatkan performa kueri untuk RDS for MariaDB dengan Amazon RDS Optimized Reads dan Meningkatkan performa kueri untuk RDS for MySQL dengan Amazon RDS Optimized Reads.	24 April 2023
Dukungan Amazon RDS untuk AWS Secrets Manager di Wilayah China	Amazon RDS mendukung Secrets Manager di Wilayah Tiongkok (Beijing) dan Tiongkok (Ningxia). Untuk informasi selengkapnya, lihat Manajemen kata sandi dengan Amazon RDS dan AWS Secrets Manager.	20 April 2023
RDS Custom for Oracle mendukung penggunaan kembali ID AMI untuk CEV baru	Saat Anda membuat versi mesin kustom (CEV), RDS Custom for Oracle akan ditetapkan secara default ke Amazon Machine Image (AMI) terbaru yang tersedia. Anda sekarang dapat menentukan ID AMI yang digunakan dalam CEV sebelumnya. Untuk informasi selengkapnya, lihat Membuat CEV.	19 April 2023

Amazon RDS mendukung penerbitan peristiwa dengan tag ke pelanggan topik	Pemberitahuan acara Amazon RDS yang dikirim ke Amazon Simple Notification Service (Amazon SNS) atau EventBridge Amazon sekarang berisi tag peristiwa di badan pesan. Tag ini menyediakan data sumber daya yang dipengaruhi oleh peristiwa layanan. Untuk informasi selengkapnya, lihat Tag dan atribut notifikasi peristiwa Amazon RDS.	17 April 2023
Beli instans terpesan untuk klaster DB Multi-AZ	Anda sekarang dapat membeli instans DB terpesan untuk klaster DB Multi-AZ. Untuk informasi selengkapnya, lihat Instans DB terpesan untuk klaster DB Multi-AZ.	12 April 2023
Amazon RDS mendukung kelas instans db.m7g dan db.r7g	Anda sekarang dapat menggunakan kelas instans db.m7g dan db.r7g untuk instans DB RDS for MySQL, RDS for MariaDB, dan RDS for PostgreSQL. Untuk informasi selengkapnya, lihat Mesin DB yang didukung untuk kelas instans DB.	12 April 2023

Pembaruan terhadap izin peran yang ditautkan layanan Amazon RDS Custom	`AmazonRDSCustomServiceRolePolicy` sekarang memberikan izin tambahan untuk memungkinkan RDS Custom for SQL Server menggunakan Amazon SQS dan membuat snapshot. Untuk informasi selengkapnya, lihat Pembaruan terhadap kebijakan terkelola AWS.	6 April 2023
Bermigrasi ke klaster DB Multi-AZ RDS for MySQL menggunakan replika baca	Anda sekarang dapat menggunakan replika baca untuk memigrasikan deployment instans DB AZ Tunggal atau Multi-AZ RDS for MySQL ke deployment klaster DB Multi-AZ RDS for MySQL dengan waktu henti yang lebih singkat. Untuk informasi selengkapnya, lihat Bermigrasi ke klaster DB Multi-AZ menggunakan replika baca.	6 April 2023
Buat replika baca instans DB dari klaster DB Multi-AZ	Anda sekarang dapat membuat replika baca instans DB dari klaster DB Multi-AZ untuk diskalakan di luar kapasitas komputasi klaster sumber. Untuk informasi selengkapnya, lihat Membuat replika baca instans DB dari klaster DB Multi-AZ.	6 April 2023

Amazon RDS Custom for SQL Server mendukung Multi-AZ	Anda dapat membuat deployment Multi-AZ dengan RDS Custom for SQL Server. Untuk informasi selengkapnya, lihat Mengelola deployment Multi-AZ untuk RDS Custom for SQL Server.	6 April 2023
Memperbarui ke izin kebijakan AWS terkelola	`AmazonRDSReadOnlyAccess` Kebijakan `AmazonRDSFullAccess` dan sekarang memberikan izin tambahan untuk memungkinkan tampilan temuan Amazon DevOps Guru di konsol RDS. Untuk informasi selengkapnya, lihat Pembaruan Amazon RDS terhadap kebijakan terkelola AWS.	30 Maret 2023
Amazon RDS mendukung Oracle APEX versi 22.2.v1	Anda dapat menggunakan APEX 22.2.v1 dengan semua versi Oracle Database yang didukung. Untuk informasi selengkapnya, lihat Oracle Application Express.	30 Maret 2023

Amazon DevOps Guru tersedia untuk RDS untuk PostgreSQL	RDS untuk PostgreSQL memberi tahu Anda tentang anomali terbaru yang terdeteksi oleh Amazon Guru. DevOps Halaman detail basis data konsol memberi tahu Anda tentang saat ini dan anomali yang terjadi dalam 24 jam terakhir. DevOpsGur u menerbitkan wawasan proaktif dengan rekomendasi untuk membantu mengatasi masalah dalam RDS Anda untuk database PostgreSQ L sebelum diprediksi terjadi. Untuk informasi selengkapnya, lihat Cara Kerja DevOps Guru for RDS.	30 Maret 2023
RDS Custom mendukung volume penyimpanan gp3 Amazon EBS	RDS Custom for Oracle dan RDS Custom for SQL Server keduanya mendukung volume EBS berbasis SSD io1, gp2, dan gp3. Untuk informasi selengkapnya, lihat Persyarat an umum untuk RDS Custom for Oracle dan Persyaratan umum untuk RDS Custom for SQL Server.	29 Maret 2023

Memperbarui ke izin kebijakan AWS terkelola	`AmazonRDSReadOnlyAccess` Kebijakan `AmazonRDSFullAccess` dan sekarang memberikan izin tambahan ke Amazon. CloudWatch Untuk informasi selengkapnya, lihat Pembaruan Amazon RDS terhadap kebijakan terkelola AWS.	16 Maret 2023
Proksi RDS tersedia di Wilayah Tiongkok	Proksi RDS sekarang tersedia di Wilayah Tiongkok (Beijing) dan Tiongkok (Ningxia). Untuk informasi selengkapnya tentang Proksi RDS, lihat Menggunakan Proksi Amazon RDS.	15 Maret 2023
Proksi RDS tersedia di Wilayah Asia Pasifik (Jakarta)	Sekarang Proksi RDS tersedia di Wilayah Asia Pasifik (Jakarta). Untuk informasi selengkapnya tentang Proksi RDS, lihat Menggunakan Proksi Amazon RDS.	8 Maret 2023
Amazon RDS Optimized Writes meningkatkan performa transaksi penulisan untuk RDS for MariaDB	Anda dapat meningkatkan performa transaksi penulisan untuk instans DB RDS for MariaDB dengan Amazon RDS Optimized Writes. Untuk informasi selengkapnya, lihat Meningkatkan performa penulisan dengan Amazon RDS Optimized Writes for MariaDB.	7 Maret 2023

Amazon RDS for PostgreSQL versi 15.2	Fitur baru di Amazon RDS for PostgreSQL 15.2 mencakup perintah “MERGE” standar SQL untuk kueri SQL bersyarat, peningkatan performa untuk penyortiran dalam memori dan berbasis disk, serta dukungan untuk komit dua fase dan pemfilter an baris/kolom untuk replikasi logis.	27 Februari 2023
RDS Custom for Oracle tersedia di Wilayah Kanada (Pusat) dan Amerika Selatan (Sao Paulo)	Untuk tabel yang menampilkan semua yang didukung Wilayah AWS, lihat RDS Custom for Oracle.	22 Februari 2023
Amazon RDS mendukung cadangan otomatis lintas Wilayah untuk RDS for MariaDB dan RDS for MySQL	Anda sekarang dapat mereplikasi snapshot DB dan log transaksi di antara Wilayah AWS untuk instans DB RDS for MariaDB dan RDS for MySQL. Untuk informasi selengkapnya, lihat Mereplika si cadangan otomatis ke Wilayah AWS lain.	22 Februari 2023

Amazon RDS for Oracle mendukung notifikasi awal tentang peningkatan versi minor otomatis	RDS memberi tahu Anda di awal tentang tanggal ketika versi minor baru untuk mesin RDS for Oracle akan tersedia. RDS mulai menjadwalkan peningkatan otomatis versi minor instans DB RDS for Oracle Anda pada tanggal ketersediaan. Untuk informasi selengkapnya, lihat Sebelum peningkatan otomatis versi minor dijadwalkan.	21 Februari 2023

Amazon RDS for SQL Server mendukung Aliran Aktivitas Basis Data	Anda sekarang dapat memantau instans DB SQL Server menggunakan Aliran Aktivitas Basis Data. Sebuah instans basis data SQL Server memiliki audit server yang dikelola oleh Amazon RDS. Anda dapat menentukan kebijakan untuk mencatat peristiwa server dalam spesifikasi audit server. Anda dapat membuat spesifikasi audit basis data dan menentukan kebijakan untuk mencatat peristiwa basis data. Aliran aktivitas dikumpulkan dan ditransmisikan ke Amazon Kinesis. Dari Kinesis, Anda dapat memantau aliran aktivitas untuk analisis lebih lanjut. Untuk informasi selengkapnya, lihat Memantau Amazon RDS dengan Aliran Aktivitas Basis Data.	15 Februari 2023
RDS mendukung MySQL 8.0.32 dan 5.7.41	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 8.0.32 dan 5.7.41. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	7 Februari 2023

Amazon RDS for Oracle mendukung cipher suite baru untuk SSL	Jika Anda menjalankan Oracle Database 19c atau 21c, Anda dapat menentukan enam cipher suite baru dalam opsi `SSL` untuk RDS for Oracle. Suite ini mendukung FIPS dan memenuhi standar FedRAMP. Untuk informasi selengkap nya, lihat Lapisan Soket Aman Oracle.	3 Februari, 2023
Amazon RDS for Oracle mendukung cipher suite baru untuk Oracle Enterprise Manager	Anda dapat menggunak an empat cipher suite baru yang memenuhi standar FedRAMP untuk opsi `OEM`. Untuk informasi selengkap nya, lihat Oracle Managemen t Agent untuk Enterprise Manager Cloud Control.	3 Februari, 2023
RDS for Oracle mendukung Aliran Aktivitas Basis Data di Asia Pasifik (Hyderabad), Eropa (Spanyol), dan Timur Tengah (UEA)	Untuk informasi selengkap nya, lihat Aliran Aktivitas Basis Data.	27 Januari 2023

Bermigrasi ke klaster DB Multi-AZ RDS for PostgreSQL menggunakan replika baca	Dengan menggunakan replika baca, Anda dapat memigrasikan deployment instans DB AZ Tunggal atau Multi-AZ RDS for PostgreSQL ke deployment klaster DB Multi-AZ RDS for PostgreSQL dengan waktu henti yang lebih singkat. Untuk informasi selengkapnya, lihat Bermigrasi ke klaster DB Multi-AZ menggunakan replika baca.	23 Januari 2023
Amazon RDS tersedia di Wilayah Asia Pasifik (Melbourne)	Amazon RDS kini tersedia di Wilayah Asia Pasifik (Melbourne). Untuk informasi selengkapnya, lihat Wilayah dan Zona Ketersediaan.	23 Januari 2023
RDS for MariaDB mendukung pemberlakuan koneksi SSL/TLS	RDS for MariaDB sekarang mendukung pemberlakuan koneksi SSL/TLS dengan mengatur parameter `require_secure_transport` ke `ON`. Untuk informasi selengkapnya, lihat Mewajibkan SSL/TLS untuk semua koneksi ke instans DB MariaDB.	19 Januari 2023

Amazon RDS Optimized Reads meningkatkan performa kueri untuk RDS for MariaDB	Anda dapat mencapai pemrosesan kueri yang lebih cepat untuk instans DB RDS for MariaDB dengan Amazon RDS Optimized Reads. Untuk informasi selengkapnya, lihat Meningkatkan performa kueri untuk RDS for MariaDB dengan Amazon RDS Optimized Reads.	11 Januari 2023
Pulihkan snapshot klaster DB Multi-AZ ke instans DB	Anda sekarang dapat memulihkan snapshot klaster DB Multi-AZ ke deployment instans DB AZ Tunggal atau Multi-AZ. Untuk informasi selengkapnya, lihat Memulihkan dari snapshot klaster DB Multi-AZ ke instans DB.	10 Januari 2023
Tentukan otoritas sertifikat (CA) selama pembuatan instans DB	Anda sekarang dapat menentukan CA mana yang akan digunakan untuk sertifikat server instans DB selama pembuatan instans DB. Untuk informasi selengkapnya, lihat Otoritas sertifikat.	5 Januari 2023

RDS Custom for SQL Server mendukung versi mesin kustom	Versi mesin kustom (CEV) untuk RDS Custom for SQL Server adalah Amazon Machine Image (AMI) dengan Microsoft SQL Server yang sudah diinstal sebelumnya. Anda memilih Windows AMI Amazon EC2 untuk digunakan sebagai image dasar dan dapat menginstal perangkat lunak lain pada sistem operasi (OS). Anda dapat menyesuai kan konfigurasi OS dan SQL Server untuk memenuhi kebutuhan perusahaan Anda. Untuk informasi selengkap nya, lihat Menggunakan versi mesin kustom untuk RDS Custom for SQL Server.	28 Desember 2022
Gunakan Deployment Blue/ Green Amazon RDS yang tersedia di Wilayah AWS tambahan	Fitur Deployment Blue/Gree n kini tersedia di Wilayah Tiongkok (Beijing) dan Tiongkok (Ningxia). Untuk informasi selengkapnya, lihat Menggunakan Deploymen t Blue/Green Amazon RDS untuk pembaruan basis data.	22 Desember 2022

Pembaruan terhadap izin peran yang ditautkan layanan IAM	ServiceRolePolicy Kebijakan AmazonRDS sekarang memberikan izin tambahan untuk. AWS Secrets Manager Untuk informasi selengkapnya, lihat Pembaruan Amazon RDS terhadap kebijakan terkelola AWS.	22 Desember 2022
Amazon RDS mendukung penggantian nama klaster DB Multi-AZ	Anda sekarang dapat mengganti nama klaster DB Multi-AZ. Untuk informasi selengkapnya, lihat Mengganti nama klaster DB Multi-AZ.	22 Desember 2022
Amazon RDS terintegrasi dengan manajemen kata AWS Secrets Manager sandi	Amazon RDS dapat mengelola kata sandi master pengguna untuk instans DB atau klaster DB Multi-AZ di Secrets Manager. Untuk informasi selengkapnya, lihat Manajemen kata sandi dengan Amazon RDS dan AWS Secrets Manager.	22 Desember 2022
Amazon RDS Optimized Writes mendukung kelas instans DB db.r6g dan db.r6gd	Amazon RDS Optimized Writes sekarang mendukung kelas instans DB db.r6g dan db.r6gd. Untuk informasi selengkapnya, lihat Meningkatkan performa penulisan dengan Amazon RDS Optimized Writes.	22 Desember 2022

Amazon RDS Custom untuk Oracle mendukung yang baru Wilayah AWS	Anda dapat membuat instans DB RDS Custom for Oracle di Wilayah Asia Pasifik (Seoul) dan Asia Pasifik (Osaka). Untuk informasi selengkapnya, lihat RDS Custom for Oracle untuk Oracle.	21 Desember 2022
Amazon RDS AWS Outposts mendukung replika baca	Anda sekarang dapat membuat replika baca dari instans DB MySQL atau PostgreSQL RDS on Outposts. Untuk informasi selengkapnya, lihat Membuat replika baca untuk Amazon RDS on AWS Outposts.	19 Desember 2022
RDS Custom for Oracle mendukung tindakan memodifikasi kelas instans DB	Anda sekarang dapat mengubah kelas instans DB RDS Custom for Oracle Anda. Untuk informasi selengkapnya, lihat Memodifikasi instans DB RDS Custom for Oracle.	16 Desember 2022
RDS for MySQL dan RDS for PostgreSQL mendukung kelas instans DB db.x2iedn	Anda sekarang dapat menggunakan kelas instans DB db.x2iedn untuk instans DB RDS for MySQL dan RDS for PostgreSQL. Untuk informasi selengkapnya, lihat Mesin DB yang didukung untuk kelas instans DB.	14 Desember 2022

Amazon RDS Optimized Writes mendukung kelas instans DB db.x2iedn	Amazon RDS Optimized Writes sekarang mendukung kelas instans DB db.x2iedn. Untuk informasi selengkapnya, lihat Meningkatkan performa penulisan dengan Amazon RDS Optimized Writes.	14 Desember 2022
Amazon RDS mendukung penyalinan grup opsi DB saat menyalin snapshot DB	Anda sekarang dapat menyalin grup opsi Akun AWS sebagai bagian dari permintaan salinan snapshot pada RDS untuk database Oracle. Untuk informasi selengkapnya, lihat Pertimbangan grup opsi.	13 Desember 2022
Amazon RDS mendukung Proksi RDS dengan RDS for PostgreSQL versi 14	Anda sekarang dapat membuat Proksi RDS dengan basis data RDS for PostgreSQL versi 14. Untuk informasi selengkapnya tentang Proksi RDS, lihat Menggunakan Proksi Amazon RDS.	13 Desember 2022
Amazon RDS for Oracle mendukung kelas instans db.x2idn, db.x2iedn, dan db.x2iezn	Anda sekarang dapat menggunakan kelas instans db.x2idn, db.x2iedn, dan db.x2iezn untuk instans DB Amazon RDS for Oracle. Untuk informasi selengkapnya, lihat Mesin DB yang didukung untuk kelas instans DB dan Kelas instans RDS for Oracle yang didukung.	12 Desember 2022

Instans DB RDS for PostgreSQL mendukung Trusted Language Extensions for PostgreSQL	Trusted Language Extensions for PostgreSQL adalah kit pengembangan sumber terbuka yang memungkinkan Anda membangun ekstensi PostgreSQL beperforma tinggi dan menjalankannya dengan aman di instans DB RDS for PostgreSQL Anda. Untuk informasi selengkapnya, lihat Menggunakan Trusted Language Extensions for PostgreSQL.	30 November 2022
Gunakan Deployment Blue/Green Amazon RDS untuk pembaruan basis data	Anda dapat membuat perubahan pada instans DB di lingkungan pementasan dan menguji perubahan tanpa memengaruhi instans DB produksi Anda. Ketika Anda siap, Anda dapat mempromosikan lingkungan pementasan menjadi lingkungan produksi baru, dengan waktu henti minimal. Untuk informasi selengkapnya, lihat Menggunakan Deployment Blue/Green Amazon RDS untuk pembaruan basis data.	27 November 2022

Amazon RDS Optimized Writes meningkatkan performa transaksi penulisan untuk RDS for MySQL	Anda dapat meningkatkan performa transaksi penulisan untuk instans DB RDS for MySQL dengan Amazon RDS Optimized Writes. Untuk informasi selengkapnya, lihat Meningkatkan performa penulisan dengan Amazon RDS Optimized Writes for MySQL.	27 November 2022
Amazon RDS Optimized Reads meningkatkan performa kueri untuk RDS for MySQL	Anda dapat mencapai pemrosesan kueri yang lebih cepat untuk instans DB RDS for MySQL dengan Amazon RDS Optimized Reads. Untuk informasi selengkapnya, lihat Meningkatkan performa kueri dengan Amazon RDS Optimized Reads.	27 November 2022
Amazon RDS tersedia di Wilayah Asia Pasifik (Hyderabad)	Amazon RDS kini tersedia di Wilayah Asia Pasifik (Hyderabad). Untuk informasi selengkapnya, lihat Wilayah dan Zona Ketersediaan.	22 November 2022
RDS mendukung MariaDB 10.6.11, 10.5.18, 10.4.27, dan 10.3.37	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MariaDB versi 10.6.11, 10.5.18, 10.4.27, dan 10.3.37. Untuk informasi selengkap nya, lihat Versi MariaDB on Amazon RDS.	18 November 2022

RDS Custom for Oracle mendukung pengaturan parameter penginstalan non-default dalam versi mesin kustom (CEV)	Saat Anda membuat CEV, Anda dapat mengatur nilai non-default untuk basis Oracle, beranda Oracle, nama dan ID pengguna UNIX, serta nama dan ID grup UNIX. Dengan cara ini, Anda mendapatkan lebih banyak kontrol atas penginstalan basis data di instans DB RDS Custom for Oracle Anda. Untuk informasi selengkapnya, lihat Mempersiapkan manifes CEV.	18 November 2022
Amazon RDS mendukung Oracle APEX versi 22.1.v1	Anda dapat menggunakan APEX 22.1.v1 dengan semua versi Oracle Database yang didukung. Untuk informasi selengkapnya, lihat Oracle Application Express.	18 November 2022
RDS for SQL Server mendukung replika baca lintas Wilayah	Anda sekarang dapat membuat replika baca lintas Wilayah untuk meningkat kan kemampuan pemulihan bencana, mengurangi latensi baca aplikasi, dan menurunka n beban kerja baca dari instans DB primer. Untuk informasi selengkapnya, lihat Membuat replika baca secara berbeda Wilayah AWS.	16 November 2022

Amazon RDS tersedia di Wilayah Eropa (Spanyol)	Amazon RDS kini tersedia di Wilayah Eropa (Spanyol) . Untuk informasi selengkapnya, lihat Wilayah dan Zona Ketersediaan.	16 November 2022
RDS for SQL Server mendukung server tertaut untuk basis data Oracle	Anda sekarang dapat membuat server tertaut untuk mengakses basis data Oracle eksternal untuk membaca data dan menjalankan perintah SQL. Untuk informasi selengkapnya, lihat Server Tertaut dengan Oracle OLEDB dengan RDS for SQL Server.	15 November 2022
RDS Custom for Oracle mendukung Oracle Multitenant	Anda dapat membuat instans DB RDS Custom for Oracle sebagai basis data kontainer (CDB). Setelah pembuatan , CDB berisi root CDB, seed PDB, dan satu PDB. Anda dapat menambahkan lebih banyak PDB secara manual menggunakan Oracle SQL. Untuk informasi selengkapnya, lihat Gambaran umum arsitektur Amazon RDS Custom for Oracle.	15 November 2022

Amazon RDS for Oracle mendukung integrasi Amazon EFS	Jika Anda menambahkan opsi `EFS_INTEGRATION` ke grup opsi Anda, Anda dapat mentransfer file antara instans DB RDS for Oracle dan sistem file Amazon EFS. Untuk informasi selengkapnya, lihat Amazon EFS.	15 November 2022
RDS mendukung MySQL 8.0.31 dan 5.7.40	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 8.0.31 dan 5.7.40. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	10 November 2022
Amazon RDS tersedia di Wilayah Eropa (Zürich)	Amazon RDS kini tersedia di Wilayah Eropa (Zürich). Untuk informasi selengkapnya, lihat Wilayah dan Zona Ketersediaan.	9 November 2022
Akses ke cadangan log transaksi kini tersedia untuk RDS for SQL Server	Anda sekarang dapat melihat dan menyalin cadangan log transaksi basis data ke Amazon S3 bucket. Untuk informasi selengkapnya, lihat Akses ke cadangan log transaksi.	7 November 2022
Cluster DB multi-AZ didukung dalam tambahan Wilayah AWS	Cluster DB multi-AZ sekarang tersedia dalam tambahan. Wilayah AWS Untuk informasi selengkapnya, lihat Klaster DB Multi-AZ.	4 November 2022

Amazon RDS mendukung penyimpanan gp3	Anda sekarang dapat membuat instans DB Amazon RDS yang menggunakan volume penyimpanan SSD Tujuan Umum (gp3) Amazon EBS, yang memungkinkan Anda menyesuaikan performa penyimpanan secara independen dari kapasitas penyimpanan. Untuk informasi selengkapnya, lihat Penyimpanan SSD Tujuan Umum.	4 November 2022
Amazon RDS mendukung peristiwa baru untuk pembaruan sistem operasi	Amazon RDS sekarang mendukung peristiwa instans DB baru, RDS-EVENT-0230, dalam kategori peristiwa patching keamanan. Peristiwa baru ini memberi tahu Anda ketika pembaruan sistem operasi tersedia untuk instans DB Anda. Untuk informasi selengkapnya, lihat Memantau peristiwa Amazon RDS dan Menggunakan pembaruan sistem operasi.	28 Oktober 2022

Amazon RDS for Oracle mendukung kelas instans memori yang dioptimalkan r5b yang telah dikonfigurasi sebelumnya	Kelas instans DB Oracle db.r5b dioptimalkan untuk beban kerja yang memerlukan memori, penyimpanan, dan I/O tambahan per vCPU. Misalnya, db.r5b.4xlarge.tpc2.mem2x memiliki multithreading aktif dan menyediakan memori dua kali lebih banyak daripada db.r5b.4xlarge. Untuk informasi selengkapnya, lihat Kelas instans RDS for Oracle.	27 Oktober 2022
Amazon RDS mendukung 15 replika baca untuk instans DB RDS for MariaDB, RDS for MySQL, dan RDS for PostgreSQL	Anda sekarang dapat membuat hingga 15 replika baca untuk instans DB RDS for MariaDB, RDS for MySQL, dan RDS for PostgreSQL. Untuk informasi tentang replika baca, lihat Menggunakan replika baca.	20 Oktober 2022
Amazon RDS for PostgreSQL kini mendukung PostgreSQL versi 15 RC 3 di lingkungan pratinjau basis data	PostgreSQL versi 15 Beta 3 sekarang tersedia di lingkungan pratinjau database di AS Timur (Ohio). Wilayah AWS Untuk informasi selengkapnya, lihat Menggunakan lingkungan pratinjau basis data.	18 Oktober 2022

Amazon RDS mendukung pengaturan konektivitas secara otomatis antara basis data RDS dan instans EC2	Anda dapat menggunakan AWS Management Console untuk mengatur konektivitas antara instans RDS DB yang ada atau cluster DB multi-AZ dan instans EC2. Untuk informasi selengkapnya, lihat Menghubungkan instans EC2 dan basis data RDS secara otomatis.	14 Oktober 2022
AWS Driver JDBC untuk PostgreSQL umumnya tersedia	Driver AWS JDBC untuk PostgreSQL adalah driver klien yang dirancang untuk RDS untuk PostgreSQ L. AWS JDBC Driver for PostgreSQL sekarang tersedia secara umum. Untuk informasi selengkapnya, lihat Menghubungkan dengan Driver AWS JDBC untuk PostgreSQL.	6 Oktober 2022
Amazon RDS for Oracle mendukung Oracle APEX versi 21.2.v1	APEX 21.2 menyertakan patch 33420059. Untuk informasi selengkapnya, lihat Persyarat an versi APEX.	3 Oktober 2022
RDS mendukung MySQL 5.7.39	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 5.7.39. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	29 September 2022

RDS mendukung MariaDB 10.6.10	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MariaDB versi 10.6.10. Untuk informasi selengkapnya, lihat Versi MariaDB on Amazon RDS.	29 September 2022
Proksi RDS mendukung RDS for SQL Server	Anda sekarang dapat membuat Proksi RDS untuk instans DB RDS yang menjalankan Microsoft SQL Server versi 2014 atau lebih tinggi. Untuk informasi selengkapnya tentang Proksi RDS, lihat Menggunakan Proksi Amazon RDS.	19 September 2022
RDS mendukung MariaDB 10.5.17, 10.4.26, dan 10.3.36	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MariaDB versi 10.5.17, 10.4.26, dan 10.3.36. Untuk informasi selengkapnya, lihat Versi MariaDB on Amazon RDS.	15 September 2022

Amazon RDS for Oracle mendukung penyimpanan instans lokal untuk data sementara	Anda sekarang dapat meluncurkan Amazon RDS for Oracle di jenis instans db.r5d dan db.m5d Amazon EC2 dengan ruang tabel sementara dan Database Smart Flash Cache (cache flash) dikonfigurasi untuk menggunakan penyimpanan instans. Dengan menyimpan data sementara secara lokal, Anda dapat mencapai latensi baca dan tulis yang lebih rendah jika dibandingkan dengan penyimpanan standar berdasarkan Amazon EBS. Untuk informasi selengkapnya, lihat Menyimpan data Oracle sementara di penyimpanan instans.	14 September 2022
Wawasan Performa menampilkan 25 kueri SQL teratas	Di dasbor Wawasan Performa, tab SQL Teratas menampilkan 25 kueri SQL yang paling berkontribusi pada beban DB. Untuk informasi selengkapnya, lihat Gambaran umum tab SQL Teratas.	13 September 2022
RDS mendukung MySQL 8.0.30	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 8.0.30. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	9 September 2022

Amazon RDS tersedia di Wilayah Timur Tengah (UEA)	Amazon RDS kini tersedia di Wilayah Timur Tengah (UEA). Untuk informasi selengkapnya, lihat Wilayah dan Zona Ketersediaan.	30 Agustus 2022
Amazon RDS for SQL Server mendukung langganan Email SSRS	Anda sekarang dapat menggunakan ekstensi Email SQL Server Reporting Services (SSRS) untuk mengirim laporan ke pengguna dan berlangganan laporan di server laporan. Untuk informasi selengkapnya, lihat Dukungan untuk SQL Server Reporting Services di RDS for SQL Server.	26 Agustus 2022
RDS for Oracle mendukung cadangan replika baca	Anda dapat mengaktifkan cadangan otomatis dan membuat snapshot manual replika RDS for Oracle. Untuk informasi selengkapnya, lihat Menggunakan RDS for Oracle untuk cadangan replika Oracle.	23 Agustus 2022

RDS for Oracle mendukung switchover Oracle Data Guard	Switchover adalah pertukaran peran antara basis data primer dan replika Oracle yang dipasang atau terbuka. Selama switchover, basis data primer asli bertransisi ke peran siaga, sedangkan basis data siaga asli transisi ke peran primer. Untuk informasi selengkapnya, lihat Melakukan switchover Oracle Data Guard.	23 Agustus 2022
Amazon RDS mendukung pengaturan konektivitas secara otomatis dengan instans EC2	Saat membuat instans DB atau cluster DB multi-AZ, Anda dapat menggunakannya AWS Management Console untuk mengatur konektivitas antara instans Amazon Elastic Compute Cloud dan instans DB atau cluster DB baru. Untuk informasi selengkapnya, lihat Mengonfigurasi konektivitas jaringan otomatis dengan instans EC2 untuk instans DB baru dan Mengonfigurasi konektivitas jaringan otomatis dengan instans EC2 untuk klaster DB baru.	22 Agustus 2022

RDS Custom for Oracle mendukung promosi replika Oracle	Jika Anda menggunakan RDS Custom for Oracle, Anda dapat mempromosikan replika Oracle terkelola Anda dengan menggunakan perintah CLI `promote-read-replica` . Selain itu, Anda dapat menghapus instans DB primer Anda, yang menyebabkan RDS Custom for Oracle mempromosikan replika Oracle terkelola Anda ke instans mandiri. Untuk informasi selengkapnya, lihat Menggunakan replika Oracle untuk RDS Custom for Oracle.	5 Agustus 2022
RDS for MySQL mendukung pemberlakuan koneksi SSL/TLS	RDS for MySQL sekarang mendukung pemberlakuan koneksi SSL/TLS dengan mengatur parameter `require_secure_transport` ke `ON`. Untuk informasi selengkapnya, lihat Mewajibkan koneksi SSL/TLS ke instans DB MySQL.	1 Agustus 2022

Amazon RDS telah menghentikan dukungan untuk Oracle Database 12c Rilis 1 (12.1.0.2)	Dukungan untuk versi 12.1.0.2 dihentikan untuk model lisensi BYOL dan LI. Pada tanggal 1 Agustus 2022, RDS for Oracle memulai peningkatan otomatis instans DB 12c Rilis 1 (12.1.0.2) dan snapshot 12.1.0.2 yang dipulihkan ke Oracle Database 19c. Untuk informasi selengkapnya, lihat Oracle Database 12c dengan Amazon RDS dan jadwal akhir dukungan di AWS re:Post.	1 Agustus 2022
Proksi RDS mendukung RDS for MariaDB	Anda sekarang dapat membuat Proksi RDS untuk instans DB RDS yang menjalankan MariaDB versi 10.2, 10.3, 10.4, atau 10.5. Dukungan MariaDB disertakan dalam kelompok mesin MySQL. Untuk informasi selengkapnya tentang Proksi RDS, lihat Menggunakan Proksi Amazon RDS.	26 Juli 2022
RDS for MariaDB mendukung kelas instans db.r5b DB	Anda sekarang dapat membuat RDS untuk instans DB MariaDB yang menggunakan kelas instans DB db.r5b. Untuk informasi selengkapnya, lihat Mesin DB yang didukung untuk kelas instans DB.	25 Juli 2022

RDS for Oracle mendukung tindakan memodifikasi aliran aktivitas basis data	Jika Anda menggunakan RDS for Oracle, Anda dapat mengubah status kebijakan audit dari aliran aktivitas basis data menjadi terkunci (default) atau tidak terkunci. Alih-alih menghentikan aliran aktivitas , Anda dapat membuka kunci status kebijakannya, menyesuaikan kebijakan audit, lalu mengunci kembali status kebijakan. Untuk informasi selengkapnya, lihat Memodifikasi aliran aktivitas basis data.	22 Juli 2022
Wawasan Performa mendukung Wilayah Asia Pasifik (Jakarta)	Sebelumnya, Anda tidak dapat menggunakan Wawasan Performa di Wilayah Asia Pasifik (Jakarta). Pembatasan ini telah dihapus. Untuk informasi selengkapnya, lihat Dukungan Wilayah AWS untuk Wawasan Performa.	21 Juli 2022

Microsoft SQL Server 2012 telah mencapai akhir dukungannya di Amazon RDS	Microsoft SQL Server 2012 telah mencapai akhir dukungannya, bertepatan dengan rencana Microsoft untuk mengakhiri dukungan yang diperluas untuk versi ini pada tanggal 12 Juli 2022. Instans Microsoft SQL Server 2012 yang ada akan ditingkatkan secara otomatis ke versi minor terbaru Microsoft SQL Server 2014 mulai tanggal 1 Juni 2022. Untuk informasi selengkapnya, lihat Dukungan Microsoft SQL Server 2012 di Amazon RDS.	12 Juli 2022
RDS mendukung MariaDB 10.6.8, 10.5.16, 10.4.25, 10.3.35, dan 10.2.44	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MariaDB versi 10.6.8, 10.5.16, 10.4.25, 10.3.35, dan 10.2.44. Untuk informasi selengkapnya, lihat Versi MariaDB yang didukung di Amazon RDS.	8 Juli 2022

Wawasan Performa RDS mendukung periode retensi tambahan	Sebelumnya, Wawasan Performa hanya menawarkan dua periode retensi: 7 hari (default) atau 2 tahun (731 hari). Sekarang, jika Anda perlu mempertahankan data performa Anda selama lebih dari 7 hari, Anda dapat menentukan 1–24 bulan. Untuk informasi selengkapnya, lihat Harga dan retensi data untuk Wawasan Performa.	1 Juli 2022
RDS Custom mendukung Wilayah Asia Pasifik (Mumbai) dan Eropa (London)	Anda dapat membuat RDS Custom untuk Oracle dan RDS Custom untuk SQL Server DB instans dalam dua instans baru Wilayah AWS: Asia Pasifik (Mumbai) dan Eropa (London). Untuk informasi selengkapnya, lihat Dukungan Wilayah AWS untuk RDS Custom for Oracle dan Dukungan Wilayah AWS untuk RDS Custom for SQL Server.	21 Juni 2022

RDS Custom for Oracle mendukung Oracle Database 18c dan 12c Rilis 2 (12.2)	Anda sekarang dapat membuat CEV untuk RDS Custom for Oracle menggunakan file penginstalan untuk Oracle Database 18c dan 12c Rilis 2 (12.2). Anda dapat menggunakan CEV ini untuk membuat instans DB RDS Custom for Oracle. Untuk informasi selengkapnya, lihat Menggunakan versi mesin kustom untuk Amazon RDS Custom for Oracle.	21 Juni 2022
Klaster DB Multi-AZ mendukung kelas instans DB db.m5d dan db.r5d	Anda sekarang dapat membuat klaster DB Multi-AZ yang menggunakan kelas instans DB db.m5d dan db.r5d. Untuk informasi selengkapnya, lihat Deployment klaster DB Multi-AZ dan Jenis kelas instans DB.	21 Juni 2022
Cluster DB multi-AZ tersedia dalam tambahan Wilayah AWS	Anda sekarang dapat membuat klaster DB Multi-AZ di Wilayah berikut: Eropa (Frankfurt) dan Eropa (Stockholm). Untuk informasi selengkapnya, lihat Deployment klaster DB Multi-AZ.	21 Juni 2022

RDS for Microsoft SQL Server mendukung migrasi basis data yang menggunakan Enkripsi Data Transparan (TDE)	RDS for SQL Server sekarang mendukung migrasi basis data Microsoft SQL Server dengan TDE diaktifkan, menggunakan pencadangan dan pemulihan native. Untuk informasi selengkapnya, lihat Dukungan untuk Enkripsi Data Transparan di SQL Server.	14 Juni 2022
Amazon RDS mendukung penerbitan peristiwa ke topik Amazon SNS terenkripsi	Amazon RDS sekarang dapat menerbitkan peristiwa ke topik Amazon Simple Notification Service (Amazon SNS) yang memiliki enkripsi sisi server (SSE) aktif, untuk perlindungan tambahan terhadap peristiwa yang membawa data sensitif. Untuk informasi selengkapnya, lihat Berlangganan notifikasi peristiwa Amazon RDS.	1 Juni 2022
RDS mendukung MySQL 5.7.38	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 5.7.38. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	31 Mei 2022

RDS for PostgreSQL mendukung replika baca kaskade	Anda sekarang dapat menggunakan replika baca kaskade dengan RDS for PostgreSQL versi 14.1 dan rilis yang lebih tinggi. Untuk informasi selengkapnya, lihat Menggunakan replika baca PostgreSQL di Amazon RDS.	4 Mei, 2022
Amazon RDS on AWS Outposts mendukung penyimpanan skala dan operasi penskalaan otomatis	Anda sekarang dapat mengubah ukuran penyimpan an instans DB di Outpost Anda dan menggunakan penskalaan otomatis penyimpanan. Untuk informasi selengkapnya, lihat Dukungan Amazon RDS on AWS Outposts untuk fitur Amazon RDS.	2 Mei 2022
Cluster DB multi-AZ tersedia dalam tambahan Wilayah AWS	Anda sekarang dapat membuat klaster DB Multi-AZ di Wilayah berikut: Asia Pasifik (Singapura) dan Asia Pasifik (Sydney). Untuk informasi selengkapnya, lihat Deployment klaster DB Multi-AZ.	29 April 2022
Amazon RDS mendukung mode tumpukan ganda	Instans DB sekarang dapat berjalan dalam mode tumpukan ganda. Dalam mode tumpukan ganda, sumber daya dapat berkomunikasi dengan instans DB melalui IPv4, IPv6, atau keduanya. Untuk informasi selengkapnya, lihat Alamat IP Amazon RDS.	29 April 2022

Amazon RDS menerbitkan metrik penggunaan ke Amazon CloudWatch	`AWS/Usage` Namespace di Amazon CloudWatch menyertakan metrik penggunaan tingkat akun untuk kuota layanan Amazon RDS Anda. Untuk informasi selengkapnya, lihat Metrik CloudWatch penggunaan Amazon untuk Amazon RDS.	28 April 2022
Amazon RDS for MySQL mendukung kelas instans DB db.m6i dan db.r6i	Anda sekarang dapat menggunakan kelas instans DB db.m6i dan db.r6i untuk instans DB Amazon RDS yang menjalankan MySQL. Untuk informasi selengkapnya, lihat Mesin DB yang didukung untuk kelas instans DB.	28 April 2022
Amazon RDS for PostgreSQL mendukung kelas instans DB db.m6i dan db.r6i	Anda sekarang dapat menggunakan kelas instans DB db.m6i dan db.r6i untuk instans DB Amazon RDS yang menjalankan PostgreSQL. Untuk informasi selengkapnya, lihat Mesin DB yang didukung untuk kelas instans DB.	27 April 2022
Amazon RDS for MariaDB mendukung kelas instans DB db.m6i dan db.r6i	Anda sekarang dapat menggunakan kelas instans DB db.m6i dan db.r6i untuk instans DB Amazon RDS yang menjalankan MariaDB. Untuk informasi selengkapnya, lihat Mesin DB yang didukung untuk kelas instans DB.	26 April 2022

Amazon RDS AWS Outposts mendukung penerapan Multi-AZ	Anda sekarang dapat membuat instans DB siaga di Outpost yang berbeda. Untuk informasi selengkapnya, lihat Amazon RDS tentang AWS Outposts dukungan untuk fitur Amazon RDS.	19 April 2022
Amazon RDS for Oracle mendukung kelas instans db.m6i dan db.r6i	Jika Anda menjalankan Oracle Database 19c, Anda dapat menggunakan kelas instans db.m6i dan db.r6i. Kelas db.m6i adalah kelas instans tujuan umum yang cocok untuk berbagai beban kerja. Untuk informasi selengkapnya, lihat Kelas instans RDS for Oracle.	8 April 2022
Amazon RDS for SQL Server mendukung replikasi pekerjaan SQL Server Agent	Saat Anda mengaktifkan fitur ini, pekerjaan SQL Server Agent yang dibuat, dimodifik asi, atau dihapus pada host primer secara otomatis disinkronkan ke host sekunder dalam konfigurasi Multi-AZ. Untuk informasi selengkap nya, lihat Menggunakan SQL Server Agent.	7 April 2022
Amazon RDS mendukung Proksi RDS dengan RDS for PostgreSQL versi 13	Anda sekarang dapat membuat Proksi RDS dengan basis data RDS for PostgreSQ L versi 13. Untuk informasi selengkapnya tentang Proksi RDS, lihat Menggunakan Proksi Amazon RDS.	4 April 2022

Amazon RDS berencana untuk menghentikan Oracle Database 12c	Oracle Database 12c berada di jalur penghentian. Oracle Corporation tidak akan lagi menyediakan patch untuk rilis Oracle Database 12c setelah tanggal. end-of-support Amazon RDS berencana untuk mulai secara otomatis meningkatkan instans DB Oracle Database 12c ke Oracle Database 19c. Untuk informasi selengkapnya, lihat Oracle Database 12c dengan Amazon RDS dan Mempersiapkan peningkatan otomatis Oracle Database 12c.	22 Maret 2022
Catatan Rilis Amazon RDS for PostgreSQL	Sekarang ada panduan terpisah untuk catatan rilis Amazon RDS for PostgreSQL. Untuk informasi selengkapnya, lihat Catatan Rilis Amazon RDS for PostgreSQL.	22 Maret 2022
Catatan Rilis Amazon RDS for Oracle	Sekarang ada panduan terpisah untuk catatan rilis Amazon RDS for Oracle. Untuk informasi selengkapnya, lihat Catatan Rilis Amazon RDS for Oracle.	22 Maret 2022

Cluster DB multi-AZ tersedia dalam tambahan Wilayah AWS	Anda sekarang dapat membuat klaster DB Multi-AZ di Wilayah berikut: AS Timur (Ohio) dan Asia Pasifik (Tokyo). Untuk informasi selengkapnya, lihat Deployment klaster DB Multi-AZ.	15 Maret 2022
Amazon RDS for PostgreSQL versi 14.2, 13.6, 12.10, 11.15, dan 10.20	RDS for PostgreSQL sekarang mendukung versi 14.2, 13.6, 12.10, 11.15, dan 10.20. Versi 14.2 dan 13.6 menambahkan dukungan untuk dua wrapper data asing baru. Ekstensi `mysql_fdw` memungkinkan PostgreSQL beroperasi dengan data yang disimpan dalam basis data MySQL, MariaDB, dan Aurora MySQL. Ekstensi `tds_fdw` memungkinkan PostgreSQL beroperasi dengan data yang disimpan dalam basis data SQL Server. Untuk informasi selengkapnya, lihat Versi basis data PostgreSQL yang didukung.	12 Maret 2022
RDS mendukung MySQL 5.7.37	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 5.7.37. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	11 Maret 2022

Amazon RDS for SQL Server mendukung kelas instans DB yang baru	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan Microsoft SQL Server yang menggunakan kelas instans DB db.m6i dan db.r6i. Untuk informasi selengkapnya, lihat Dukungan kelas instans DB untuk Microsoft SQL Server.	9 Maret 2022
Amazon RDS for Oracle mendukung Oracle Database 21c	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan Oracle Database 21c (21.0.0.0). Ini adalah rilis Oracle Database pertama yang hanya mendukung arsitektur multitenant (CDB). Untuk informasi selengkapnya, lihat Oracle Database 21c dengan Amazon RDS.	7 Maret 2022
RDS mendukung MariaDB 10.6.7, 10.5.15, 10.4.24, 10.3.34, dan 10.2.43	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MariaDB versi 10.6.7, 10.5.15, 10.4.24, 10.3.34, dan 10.2.43. Untuk informasi selengkapnya, lihat Versi MariaDB on Amazon RDS.	3 Maret 2022

AWS Driver JDBC untuk MySQL umumnya tersedia	Driver AWS JDBC untuk MySQL adalah driver klien yang dirancang untuk RDS untuk MySQL. Driver AWS JDBC untuk MySQL sekarang tersedia secara umum. Untuk informasi selengkapnya, lihat Menghubungkan dengan Amazon Web Services JDBC Driver for MySQL.	2 Maret 2022
Klaster DB Multi-AZ tersedia secara umum	Deployment klaster DB Multi-AZ adalah mode deploymen t ketersediaan tinggi Amazon RDS dengan dua instans DB siaga yang dapat dibaca. Klaster DB Multi-AZ sekarang tersedia secara umum. Untuk informasi selengkapnya, lihat Deployment klaster DB Multi-AZ.	1 Maret 2022
RDS mendukung MySQL 8.0.28	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 8.0.28. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	28 Februari 2022

Amazon RDS for Oracle mendukung pengaturan baru untuk enkripsi jaringan native (NNE)	Untuk mengontrol apakah klien dapat terhubung dengan metode enkripsi dan checksumming yang tidak aman, atur `SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS` dan `SQLNET.ALLOW_WEAK_CRYPTO` dalam opsi NNE. Contoh metode tidak aman termasuk DES, 3DES, RC4, dan MD5. Untuk informasi selengkapnya, lihat Pengaturan opsi NNE.	25 Februari 2022
Amazon RDS for SQL Server mendukung Grup Ketersediaan Selalu Aktif untuk Microsoft SQL Server 2017 Standard Edition	Saat Anda membuat instans DB menggunakan konfigurasi Multi-AZ pada SQL Server 2017 Standard Edition 14.00.3401.7 dan versi yang lebih tinggi, RDS secara otomatis menggunakan Grup Ketersediaan. Untuk informasi selengkapnya, lihat Deployment Multi-AZ untuk Microsoft SQL Server.	18 Februari 2022
RDS for Oracle mendukung Aliran Aktivitas Basis Data di Wilayah Asia Pasifik (Jakarta)	Untuk informasi selengkapnya, lihat Support Wilayah AWS untuk aliran aktivitas database.	16 Februari 2022

Dukungan Amazon RDS Custom for Oracle untuk Oracle Database 12.1	Anda sekarang dapat membuat versi mesin kustom untuk RDS Custom for Oracle yang menggunakan Oracle Database 12.1 Enterprise Edition. Untuk informasi selengkapnya, lihat Menggunakan versi mesin kustom untuk Amazon RDS Custom for Oracle.	4 Februari 2022
Amazon RDS for MariaDB mendukung versi mayor yang baru	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MariaDB versi 10.6. Untuk informasi selengkapnya, lihat Dukungan MariaDB 10.6 di Amazon RDS.	3 Februari, 2022

Wawasan Performa mendukung pengambilan rencana untuk kueri Oracle	Konsol Wawasan Performa mendukung dimensi rencana baru untuk kueri SQL teratas. Saat Anda mengiris berdasarkan rencana, Anda dapat melihat rencana mana yang digunakan kueri Oracle teratas Anda. Jika kueri menggunakan beberapa rencana, Anda dapat membandingkan rencana secara berdampingan di konsol dan menentukan rencana mana yang paling efisien. Anda juga dapat menelusuri untuk melihat langkah-langkah mana dalam rencana yang memiliki biaya tertinggi. Untuk informasi selengkapnya, lihat Menganalisis rencana eksekusi Oracle menggunakan dasbor Wawasan Performa.	27 Januari 2022
Wawasan Performa mendukung API baru	Wawasan Performa mendukung API berikut: `GetResourceMetadata`, `ListAvailableResourceDimensions`, dan `ListAvailableResourceMetrics`. Untuk informasi selengkapnya, lihat Mengambil metrik dengan API Wawasan Performa dalam panduan ini dan Referensi API Wawasan Performa Amazon RDS.	12 Januari 2022

Proksi RDS mendukung peristiwa	Proxy RDS sekarang menghasilkan acara yang dapat Anda berlangganan dan lihat di CloudWatch Acara atau konfigurasikan untuk dikirim ke Amazon EventBridge. Untuk informasi selengkapnya, lihat Menggunakan peristiwa Proksi RDS.	11 Januari 2022
Amazon RDS for SQL Server mendukung mode Multidimensi SSAS	RDS for SQL Server mendukung pengoperasian SQL Server Analysis Services (SSAS) dalam mode Tabular atau Multidimensional. Untuk informasi selengkapnya, lihat Dukungan untuk SQL Server Analysis Services di RDS for SQL Server.	7 Januari 2022
Proxy RDS tersedia dalam tambahan Wilayah AWS	Proksi RDS sekarang tersedia di Wilayah berikut: Afrika (Cape Town), Asia Pasifik (Hong Kong), Asia Pasifik (Osaka), Eropa (Milan), Eropa (Paris), Eropa (Stockholm), Timur Tengah (Bahrain), dan Amerika Selatan (Sao Paulo). Untuk informasi selengkapnya tentang Proksi RDS, lihat Menggunakan Proksi Amazon RDS.	5 Januari 2022

RDS mendukung MySQL 8.0.27	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 8.0.27. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	21 Desember 2021
Amazon RDS tersedia di Wilayah Asia Pasifik (Jakarta)	Amazon RDS kini tersedia di Wilayah Asia Pasifik (Jakarta) . Untuk informasi selengkap nya, lihat Wilayah dan Zona Ketersediaan.	13 Desember 2021
Amazon RDS mendukung MariaDB 10.5.13, 10.4.22, 10.3.32, dan 10.2.41	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MariaDB versi 10.5.13, 10.4.22, 10.3.32, dan 10.2.41. Untuk informasi selengkap nya, lihat Versi MariaDB on Amazon RDS.	8 Desember 2021
Amazon RDS Custom for SQL Server	Amazon RDS Custom adalah layanan basis data terkelola untuk aplikasi lama, kustom, dan yang dipaketkan yang memerlukan akses ke sistem operasi dan lingkungan basis data yang mendasari nya. Dengan Amazon RDS Custom, Anda mendapatk an otomatisasi Amazon RDS dan fleksibilitas Amazon EC2. Untuk informasi selengkapnya, lihat Menggunakan Amazon RDS Custom.	1 Desember 2021

Klaster DB Multi-AZ (pratinjau)	Anda sekarang dapat membuat klaster DB Multi-AZ untuk RDS for MySQL dan RDS for PostgreSQL. Deployment klaster DB Multi-AZ adalah mode deploymen t ketersediaan tinggi Amazon RDS dengan dua instans DB siaga yang dapat dibaca. Klaster DB Multi-AZ berada dalam pratinjau. Untuk informasi selengkapnya, lihat Deployment klaster DB Multi-AZ (pratinjau).	23 November 2021
Amazon RDS mendukung Proksi RDS dengan RDS for PostgreSQL versi 12	Anda sekarang dapat membuat Proksi RDS dengan basis data RDS for PostgreSQ L versi 12. Untuk informasi selengkapnya tentang Proksi RDS, lihat Menggunakan Proksi Amazon RDS.	22 November 2021
Amazon RDS AWS Outposts mendukung cadangan lokal	Anda dapat menyimpan cadangan otomatis dan snapshot manual di pos Anda Wilayah AWS atau lokal di Outpost Anda. Untuk informasi selengkapnya, lihat Amazon RDS tentang AWS Outposts dukungan untuk fitur Amazon RDS.	22 November 2021

Dukungan Amazon RDS untuk lintas akun AWS KMS keys	Anda dapat menggunakan kunci KMS dari AWS akun yang berbeda untuk enkripsi saat mengekspor snapshot DB ke Amazon S3. Untuk informasi selengkapnya, lihat Mengekspor data snapshot DB ke Amazon S3.	3 November 2021
Amazon RDS di AWS Outposts mendukung penerbitan log mesin database ke Log CloudWatch	RDS di Outposts sekarang mendukung penerbitan log mesin database ke Log CloudWatch . Untuk informasi selengkapnya, lihat dukungan Amazon RDS di AWS Outposts untuk fitur Amazon RDS.	2 November 2021
Amazon RDS Custom for Oracle	Amazon RDS Custom adalah layanan basis data terkelola untuk aplikasi lama, kustom, dan yang dipaketkan yang memerlukan akses ke sistem operasi dan lingkungan basis data yang mendasari nya. Dengan Amazon RDS Custom, Anda mendapatk an otomatisasi Amazon RDS dan fleksibilitas Amazon EC2. Untuk informasi selengkapnya, lihat Menggunakan Amazon RDS Custom.	26 Oktober 2021

Dukungan untuk replikasi tertunda untuk RDS for MySQL versi 8.0	Anda dapat mengonfigurasi replikasi tertunda untuk RDS for MySQL versi 8.0.26, Anda dapat mengonfigurasi replikasi tertunda untuk instans DB MySQL versi 8.0. Untuk informasi selengkapnya, lihat Mengonfigurasi replikasi tertunda dengan MySQL.	25 Oktober 2021
Dukungan untuk MySQL 8.0.26	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 8.0.26. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	25 Oktober 2021
Dukungan untuk replikasi berbasis GTID untuk RDS for MySQL versi 8.0	Dimulai dengan RDS for MySQL versi 8.0.26, Anda dapat mengonfigurasi replikasi berbasis GTID untuk instans DB RDS for MySQL versi 8.0. Untuk informasi selengkapnya, lihat Menggunakan replikasi berbasis GTID untuk RDS for MySQL.	25 Oktober 2021
Amazon RDS mendukung Proksi RDS dengan RDS for MySQL 8.0	Anda sekarang dapat membuat Proksi RDS untuk instans basis data MySQL 8.0. Untuk informasi selengkap nya, lihat Menggunakan Proksi Amazon RDS.	21 Oktober 2021

Amazon RDS di AWS Outposts mendukung RDS tambahan untuk versi MySQL	RDS on Outposts sekarang mendukung RDS for MySQL versi 8.0.23 dan 8.0.25. Untuk informasi selengkapnya, lihat dukungan Amazon RDS di AWS Outposts untuk fitur Amazon RDS.	20 Oktober 2021
Amazon RDS for PostgreSQL kini mendukung PostgreSQL versi 14 RC 1 di lingkungan pratinjau basis data	PostgreSQL versi 14 RC 1 sekarang tersedia di lingkungan pratinjau database di AS Timur (Ohio). Wilayah AWS Untuk informasi selengkapnya, lihat Menggunakan lingkungan pratinjau basis data.	19 Oktober 2021
Amazon RDS mendukung Performance Insights sebagai tambahan Wilayah AWS	Wawasan Performa tersedia di Wilayah Timur Tengah (Bahrain), Afrika (Cape Town), Eropa (Milan), dan Asia Pasifik (Osaka). Untuk informasi selengkapnya, lihat Dukungan Wilayah AWS untuk Wawasan Performa.	5 Oktober 2021
Wawasan Performa mendukung statistik tingkat penyerapan untuk Oracle	Jika Anda menggunakan Wawasan Performa, Anda dapat melihat statistik SQL baik pada tingkat pernyataan maupun penyerapan untuk Amazon RDS for Oracle. Untuk informasi selengkapnya, lihat Menganalisis kueri yang berjalan di Oracle.	4 Oktober 2021

Amazon RDS di AWS Outposts mendukung RDS tambahan untuk versi PostgreSQL	RDS on Outposts sekarang mendukung RDS for PostgreSQL versi 12.8 dan 13.4. Untuk informasi selengkapnya, lihat dukungan Amazon RDS di AWS Outposts untuk fitur Amazon RDS.	1 Oktober 2021
Amazon RDS mendukung Oracle APEX versi 21.1.v1	Anda dapat menggunakan APEX 21.1.v1 dengan semua versi Oracle Database yang didukung. Untuk informasi selengkapnya, lihat Oracle Application Express.	24 September 2021
Amazon RDS for Oracle mendukung enkripsi di sisi klien untuk NNE	Saat Anda mengonfigurasi NNE, Anda mungkin ingin menghindari pemaksaan enkripsi di sisi server. Misalnya, Anda mungkin tidak ingin memaksa semua komunikasi klien untuk menggunakan enkripsi karena server memerlukannya. Dalam hal ini, Anda dapat memaksa enkripsi di sisi klien menggunakan opsi `SQLNET.*CLIENT`. Untuk informasi selengkapnya, lihat Enkripsi jaringan native Oracle.	24 September 2021

Amazon RDS for MySQL dan RDS for PostgreSQL mendukung kelas instans DB yang baru	Anda sekarang dapat menggunakan instans db.r5b, db.t4g, dan db.x2g untuk membuat instans DB Amazon RDS yang menjalankan MySQL atau PostgreSQL. Untuk informasi selengkapnya, lihat Mesin DB yang didukung untuk kelas instans DB.	15 September 2021
Amazon RDS for Microsoft SQL Server mendukung Java Database Connectivity (JDBC) dengan Microsoft Distributed Transaction Coordinator (MSDTC)	Transaksi JDBC XA sekarang didukung dengan MSDTC untuk SQL Server 2017 versi 14.00.3223.3 dan lebih tinggi, dan SQL Server 2019. Untuk informasi selengkapnya, lihat Dukungan untuk Microsoft Distributed Transaction Coordinator di RDS for SQL Server.	7 September 2021
Amazon RDS mendukung MariaDB 10.5.12, 10.4.21, 10.3.31, dan 10.2.40	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MariaDB versi 10.5.12, 10.4.21, 10.3.31, dan 10.2.40. Untuk informasi selengkapnya, lihat Versi MariaDB on Amazon RDS.	2 September 2021

Amazon RDS telah mengakhiri dukungan untuk Oracle Database 18c	Anda dapat membuat instans DB hanya untuk Oracle Database 12c dan Oracle Database 19c. Jika Anda memiliki snapshot Oracle Database 18c, tingkatkan ke rilis yang lebih baru. Untuk informasi selengkapnya, lihat Meningkatkan snapshot DB Oracle.	17 Agustus 2021
Amazon RDS for SQL Server mendukung peningkatan versi minor otomatis	Anda sekarang dapat menjadikan instans DB RDS for SQL Server secara otomatis ditingkatkan ke versi minor terbaru. Untuk informasi selengkapnya, lihat Meningkatkan mesin DB Microsoft SQL Server.	13 Agustus 2021
Amazon RDS for PostgreSQL kini mendukung PostgreSQL versi 14 beta 2 di lingkungan pratinjau basis data	Untuk informasi selengkapnya tentang PostgreSQL versi 14 beta 1, lihat Catatan rilis PostgreSQL 14 beta 1. Untuk informasi selengkapnya tentang PostgreSQL versi 14 beta 2, lihat Catatan rilis PostgreSQL 14 beta 2. Untuk informasi tentang Lingkungan Pratinjau Basis Data, lihat Menggunakan lingkungan pratinjau basis data.	9 Agustus 2021

Amazon RDS mendukung Proksi RDS di VPC bersama	Anda sekarang dapat membuat Proksi RDS di VPC bersama. Untuk informasi selengkapnya tentang Proksi RDS, lihat “Mengelola Koneksi dengan Proksi Amazon RDS” dalam Panduan Pengguna Amazon RDS atau Panduan Pengguna Aurora.	6 Agustus 2021
Amazon RDS mendukung MariaDB 10.2.39	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MariaDB versi 10.2.39. Untuk informasi selengkapnya, lihat Versi MariaDB on Amazon RDS.	4 Agustus 2021
Amazon RDS for Oracle menambahkan opsi `TIMEZONE_FILE_AUTOUPGRADE`	Dengan opsi ini, Anda dapat meningkatkan file zona waktu saat ini ke versi terbaru instans Oracle DB Anda. Untuk informasi selengkapnya, lihat Peningkatan otomatis file zona waktu Oracle.	30 Juli 2021
Amazon RDS memperluas dukungan untuk pencadangan otomatis lintas Wilayah	Anda sekarang dapat mereplikasi snapshot DB dan log transaksi di antara lebih banyak Wilayah AWS. Untuk informasi selengkapnya, lihat Mereplikasi pencadangan otomatis ke Wilayah lain. AWS	19 Juli 2021

Dukungan untuk MySQL 5.7.34	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 5.7.34. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	8 Juli 2021
Amazon RDS di AWS Outposts mendukung RDS tambahan untuk versi PostgreSQL	RDS on Outposts sekarang mendukung RDS for PostgreSQL versi 12.7 dan 13.3. Untuk informasi selengkapnya, lihat dukungan Amazon RDS di AWS Outposts untuk fitur Amazon RDS.	8 Juli 2021
Amazon RDS for PostgreSQL mendukung oracle_fdw	Anda sekarang dapat menggunakan ekstensi oracle_fdw untuk menyediakan wrapper data asing untuk akses ke basis data Oracle. Untuk informasi selengkapnya, lihat Mengakses data eksternal dengan ekstensi oracle_fdw.	8 Juli 2021

Amazon RDS mendukung Oracle Management Agent (OMA) versi 13.5	Anda dapat membuat Oracle Management Agent (OMA) versi 13.5 dengan Oracle Enterprise Manager (OEM) Cloud Control 13c Rilis 5 dan yang lebih tinggi. Amazon RDS for Oracle menginstal OMA, yang kemudian berkomunikasi dengan Oracle Management Service (OMS) Anda untuk memberikan informasi pemantauan. Jika Anda menjalankan OMS 13.5, Anda dapat mengelola basis data dengan menginstal OMA 13.5. Untuk informasi selengkapnya, lihat Oracle Management Agent untuk Enterprise Manager Cloud Control.	7 Juli 2021

Amazon RDS for Oracle mendukung pengunduhan log dari Amazon S3	Jika log pengulangan yang diarsipkan tidak ada di instans Anda, tetapi dilindungi oleh periode retensi cadangan, Anda dapat menggunakan `rdsadmin.rdsadmin_archive_log_download` untuk mengunduhnya dari Amazon S3. RDS for Oracle menyimpan log ke direktori `/rdsdbdata/log/arch` pada instans DB Anda. Untuk informasi selengkapnya, lihat Mengunduh log pengulangan yang diarsipkan dari Amazon S3.	2 Juli 2021
Amazon RDS mendukung MariaDB 10.4.18 dan 10.5.9	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MariaDB versi 10.4.18 dan 10.5.9. Untuk informasi selengkapnya, lihat Versi MariaDB on Amazon RDS.	30 Juni 2021

Amazon RDS for Oracle mendukung Aliran Aktivitas Basis Data	Anda sekarang dapat memantau instans DB Oracle menggunakan Aliran Aktivitas Basis Data. Basis data Oracle menulis catatan audit untuk jejak audit terpadu. Jika Anda memulai aliran aktivitas basis data pada instans DB Oracle, Amazon Kinesis akan mengalirkan semua aktivitas yang sesuai dengan kebijakan audit Oracle Database. Untuk informasi selengkapnya, lihat Memantau Amazon RDS dengan Aliran Aktivitas Basis Data.	23 Juni 2021
Amazon RDS for Oracle memperkenalkan kelas instans memori yang dioptimalkan	Kelas instans DB Oracle baru dioptimalkan untuk beban kerja yang memerlukan memori, penyimpanan, dan I/O tambahan per vCPU. Untuk informasi selengkapnya, lihat Kelas instans RDS for Oracle.	23 Juni 2021
Dukungan untuk MySQL 8.0.25	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 8.0.25. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	18 Juni 2021

Amazon RDS di AWS Outposts mendukung RDS tambahan untuk versi PostgreSQL	RDS on Outposts sekarang mendukung RDS for PostgreSQL versi 12.5, 12.6, 13.1 dan 13.2. Untuk informasi selengkapnya, lihat dukungan Amazon RDS di AWS Outposts untuk fitur Amazon RDS.	28 Mei 2021
Amazon RDS mendukung MariaDB 10.2.37 dan 10.3.28	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MariaDB versi 10.2.37 dan 10.3.28. Untuk informasi selengkapnya, lihat Versi MariaDB on Amazon RDS.	27 Mei 2021
Amazon RDS for Oracle mendukung basis data kontainer multi-penghuni (CDB)	Arsitektur multi-penghuni memungkinkan basis data Oracle menjadi CDB. Dalam Oracle Database 19c, CDB Anda dapat mencakup PDB tunggal. Pengalaman pengguna dengan PDB sebagian besar identik dengan pengalaman pengguna dengan non-CDB. Untuk informasi selengkapnya, lihat Arsitektur RDS for Oracle.	25 Mei 2021
Amazon RDS di AWS Outposts mendukung Amazon RDS for SQL Server	RDS on Outposts sekarang mendukung Amazon RDS for SQL Server. Untuk informasi selengkapnya, lihat dukungan Amazon RDS di AWS Outposts untuk fitur Amazon RDS.	11 Mei 2021

Amazon RDS memperluas dukungan untuk pencadangan otomatis lintas Wilayah	Sekarang Anda dapat mengonfigurasi instans database Amazon RDS yang menjalankan Microsoft SQL Server untuk mereplikasi snapshot DB dan log transaksi ke Wilayah yang berbeda. AWS Untuk informasi selengkapnya, lihat Mereplikasi pencadangan otomatis ke Wilayah lain. AWS	7 Mei 2021
Amazon RDS mendukung cadangan otomatis lintas Wilayah untuk instans DB terenkripsi	Anda sekarang dapat mereplikasi snapshot DB dan log transaksi ke Wilayah AWS berbeda untuk instans basis data Amazon RDS terenkripsi yang menjalankan Oracle atau PostgreSQL. Untuk informasi selengkapnya, lihat Mereplikasi pencadangan otomatis ke Wilayah lain. AWS	3 Mei 2021
Amazon RDS di AWS Outposts mendukung pemantauan Amazon CloudWatch	RDS di Outposts sekarang mendukung pemantauan Amazon CloudWatch . Untuk informasi selengkapnya, lihat dukungan Amazon RDS di AWS Outposts untuk fitur Amazon RDS.	21 April 2021

RDS untuk PostgreSQL mendukung fungsi Lambda AWS	Anda sekarang dapat memanggil fungsi AWS Lambda untuk RDS Anda untuk instans PostgreSQL DB. Untuk informasi selengkapnya, lihat Menginvokasi fungsi AWS Lambda dari instans DB RDS for PostgreSQL.	13 April, 2021
RDS for SQL Server mendukung peristiwa yang diperluas	Anda dapat menggunakan peristiwa yang diperluas SQL Server untuk mengambil informasi debugging dan pemecahan masalah. Untuk informasi selengkapnya, lihat Menggunakan peristiwa yang diperluas dengan Amazon RDS for Microsoft SQL Server.	8 April 2021
Dukungan untuk MySQL 8.0.23, 5.7.33 dan 5.6.51	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 8.0.23, 5.7.33, dan 5.6.51. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	31 Maret 2021

Rollback otomatis pada peningkatan Amazon RDS for MySQL yang gagal	Jika peningkatan instans DB dari MySQL versi 5.7 ke MySQL versi 8.0 gagal, Amazon RDS melakukan rollback perubahan yang dilakukan untuk peningkatan secara otomatis. Setelah rollback, instans DB MySQL menjalankan MySQL versi 5.7. Untuk informasi selengkapnya, lihat Rollback setelah kegagalan untuk meningkatkan dari MySQL 5.7 ke 8.0.	18 Maret 2021
Amazon RDS mendukung replika baca lintas Wilayah di Wilayah pilihan	Anda sekarang dapat mereplikasi instans DB ke Wilayah pilihan. Untuk informasi selengkapnya, lihat Membuat replika baca di AWS Wilayah yang berbeda.	18 Maret 2021

Amazon RDS berencana untuk menghentikan Oracle Database 18c	Oracle Database 18c (18.0.0.0) berada di jalur penghentian. Oracle Corporation tidak akan lagi menyediakan patch untuk Oracle Database 18c setelah tanggal tersebut. end-of-support Pada tanggal 1 Juli 2021, Amazon RDS berencana untuk mulai secara otomatis meningkatkan instans Oracle Database 18c ke Oracle Database 19c. Sebelum peningkatan otomatis dimulai, kami sangat menyarankan Anda untuk meningkatkan instans Oracle Database 18c Anda ke Oracle Database 19c. Untuk informasi selengkapnya, lihat Menyiapkan peningkatan otomatis Oracle Database 18c.	11 Maret 2021
Amazon RDS telah mengakhiri dukungan untuk Oracle Database 11g	Anda hanya dapat membuat instans DB untuk Oracle Database 12c Rilis 1 (12.1.0.2) dan yang lebih baru. Jika Anda memiliki snapshot Oracle Database 11g, tingkatkan ke rilis yang lebih baru. Untuk informasi selengkapnya, lihat Meningkatkan snapshot DB Oracle.	11 Maret 2021

Amazon RDS mendukung pencadangan berkelanjutan instans DB di AWS Backup	Anda sekarang dapat membuat backup otomatis AWS Backup dan mengembalikan instans DB dari backup ini ke waktu yang ditentukan. Untuk informasi selengkapnya, lihat Menggunakan AWS Backup untuk mengelola pencadangan otomatis.	10 Maret 2021
Amazon RDS mendukung Oracle Management Agent (OMA) versi 13.4	Anda dapat menggunakan Oracle Management Agent (OMA) versi 13.4 dengan Oracle Enterprise Manager (OEM) Cloud Control 13c Rilis 4 Pembaruan 9. Amazon RDS for Oracle menginstal OMA, yang kemudian berkomunikasi dengan Oracle Management Service (OMS) Anda untuk memberikan informasi pemantauan. Jika Anda menjalankan OMS 13.4, Anda dapat mengelola basis data dengan menginstal OMA 13.4. Untuk informasi selengkapnya, lihat Oracle Management Agent untuk Enterprise Manager Cloud Control.	10 Maret 2021

Peningkatan titik akhir Proksi RDS	Anda dapat membuat titik akhir tambahan yang terkait dengan setiap Proksi RDS. Membuat titik akhir dalam VPC yang berbeda memungkin kan akses lintas VPC untuk proksi tersebut. Proksi untuk klaster Aurora MySQL juga dapat memiliki titik akhir hanya baca. Titik akhir pembaca ini terhubung ke instans DB pembaca di dalam klaster dan dapat meningkatkan skalabili tas dan ketersediaan baca untuk aplikasi intensif kueri. Untuk informasi selengkap nya tentang Proksi RDS, lihat “Mengelola Koneksi dengan Proksi Amazon RDS” dalam Panduan Pengguna Amazon RDS atau Panduan Pengguna Aurora.	8 Maret 2021
Amazon RDS memperluas dukungan untuk cadangan otomatis lintas Wilayah	Sekarang Anda dapat mengonfigurasi instans database Amazon RDS yang menjalankan PostgreSQL untuk mereplikasi snapshot DB dan log transaksi ke Wilayah yang berbeda. AWS Untuk informasi selengkapnya, lihat Mereplikasi pencadangan otomatis ke Wilayah lain. AWS	8 Maret 2021

Filter replikasi untuk Amazon RDS for MariaDB dan Amazon RDS for MySQL didukung di Wilayah Tiongkok (Beijing) dan Tiongkok (Ningxia)	Filter replikasi kini didukung di Wilayah Tiongkok (Beijing) dan Tiongkok (Ningxia). Untuk informasi selengkapnya, lihat Mengonfigurasi filter replikasi dengan MariaDB dan Mengonfigurasi filter replikasi dengan MySQL.	5 Maret 2021
Amazon RDS mendukung salinan snapshot DB lintas Wilayah di Wilayah pilihan	Anda sekarang dapat menyalin snapshot DB ke dan dari Wilayah keikutser taan AWS . Untuk informasi selengkapnya, lihat Menyalin snapshot di seluruh AWS Wilayah.	4 Maret 2021
Amazon RDS for SQL Server mendukung Grup Ketersedi aan Selalu Aktif untuk Standard Edition	Saat Anda membuat instans DB menggunakan konfigura si Multi-AZ pada SQL Server 2019 untuk mesin basis data Standard Edition, RDS secara otomatis menggunak an Grup Ketersediaan. Untuk informasi selengkapnya, lihat Deployment Multi-AZ untuk Microsoft SQL Server.	23 Februari 2021

Amazon RDS for Oracle memperkenalkan prosedur terkait advisor	Paket `rdsadmin_util` mencakup prosedur `advisor_task_set_parameter` , `advisor_task_drop` , dan `dbms_stats_init` . Anda dapat menggunakan prosedur ini untuk mengubah, menghentikan, dan mengaktifkan kembali tugas advisor seperti `AUTO_STATS_ADVISOR_TASK` . Untuk informasi selengkapnya, lihat Menetapkan parameter untuk tugas penasihat.	23 Februari 2021
Amazon RDS memberikan alasan failover untuk instans DB Multi-AZ	Anda sekarang dapat melihat penjelasan lebih mendetail ketika instans DB Multi-AZ gagal melakukan replika siaga. Untuk informasi selengkapnya, lihat Proses failover untuk Amazon RDS.	18 Februari 2021
Amazon RDS memperluas dukungan untuk mengekspor snapshot ke Amazon S3	Anda sekarang dapat mengekspor data snapshot DB ke Amazon S3 di Tiongkok. Untuk informasi selengkapnya, lihat Mengekspor data snapshot DB ke Amazon S3.	17 Februari 2021

Filter replikasi untuk Amazon RDS for MariaDB dan Amazon RDS for MySQL	Anda dapat mengonfig urasi filter replikasi untuk instans MySQL dan MariaDB. Replikasi filter menentukan basis data dan tabel mana yang direplikasi dalam replika baca. Anda dapat membuat daftar basis data dan tabel yang akan disertakan atau dikecualikan untuk setiap replika baca. Untuk informasi selengkapnya, lihat Mengonfig urasi filter replikasi dengan MariaDB dan Mengonfigurasi filter replikasi dengan MySQL.	12 Februari 2021
RDS for Oracle mendukung APEX 20.2v1	Anda dapat menggunakan APEX 20.2.v1 dengan semua versi Oracle Database yang didukung. Untuk informasi selengkapnya, lihat Oracle Application Express.	2 Februari 2021

Amazon RDS for SQL Server mendukung penyimpanan instans lokal untuk basis data tempdb	Anda sekarang dapat meluncurkan Amazon RDS for SQL Server di jenis instans db.r5d dan db.m5d Amazon EC2 dengan basis data tempdb yang dikonfigurasi untuk menggunakan penyimpanan instans. Dengan menempatkan file data dan file log tempdb secara lokal, Anda dapat mencapai latensi baca dan tulis yang lebih rendah jika dibandingkan dengan penyimpanan standar berdasarkan Amazon EBS. Untuk informasi selengkapnya, lihat Dukungan penyimpanan instans untuk basis data tempdb di Amazon RDS for SQL Server.	27 Januari 2021
Amazon RDS for PostgreSQL mendukung pg_partman dan pg_cron	Amazon RDS for PostgreSQL sekarang mendukung ekstensi pg_partman dan pg_cron. Untuk informasi selengkapnya tentang ekstensi pg_partman, lihat Mengelola partisi PostgreSQL dengan ekstensi pg_partman. Untuk informasi selengkapnya tentang ekstensi pg_cron, lihat Menjadwalkan pemeliharaan dengan ekstensi pg_cron PostgreSQL.	12 Januari 2021

Amazon RDS mendukung penerbitan log Oracle Management Agent ke Amazon Logs CloudWatch	Log Oracle Management Agent terdiri dari emctl.log , emdctlj.log, gcagent.log, gcagent_errors.log, emagent.nohup, dan secure.log. Amazon RDS menerbitkan masing-masing log ini sebagai aliran CloudWatch log terpisah. Untuk informasi selengkapnya, lihat Menerbitkan log Oracle ke Amazon CloudWatch Logs.	28 Desember 2020
Amazon RDS di AWS Outposts mendukung versi database tambahan	RDS on Outposts sekarang mendukung versi MySQL dan PostgreSQL tambahan. Untuk informasi selengkapnya, lihat dukungan Amazon RDS di AWS Outposts untuk fitur Amazon RDS.	23 Desember 2020
Amazon RDS di AWS Outposts mendukung COIP	RDS on Outposts sekarang mendukung alamat IP milik pelanggan (CoIP). CoIP menyediakan konektivitas titik waktu atau eksternal untuk sumber daya di subnet Outpost Anda melalui jaringan on-premise Anda. Untuk informasi selengkapnya, lihat Alamat IP milik pelanggan untuk RDS on Outposts.	22 Desember 2020

Amazon RDS for Oracle berencana untuk meningkatkan instans 11g BYOL ke 19c	Pada tanggal 4 Januari 2021, kami berencana untuk mulai secara otomatis meningkatkan semua edisi instans Oracle Database 11g pada model Bawa Lisensi Sendiri (BYOL) untuk Oracle Database 19c. Semua instans Oracle Database 11g, termasuk instans terpesan, akan dipindahkan ke Pembaruan Rilis (RU) terkini yang tersedia. Untuk informasi selengkapnya, lihat Menyiapkan peningkatan otomatis Oracle Database 11g BYOL.	11 Desember 2020
Amazon RDS mendukung replikasi backup otomatis ke Wilayah lain AWS	Sekarang Anda dapat mengonfigurasi instans database Amazon RDS untuk mereplikasi snapshot dan log transaksi ke AWS Wilayah tujuan pilihan Anda. Untuk informasi selengkapnya, lihat Mereplikasi pencadangan otomatis ke Wilayah lain. AWS	4 Desember 2020
Amazon RDS for Oracle dan Amazon RDS for Microsoft SQL Server mendukung kelas instans DB yang baru	Anda sekarang dapat menggunakan kelas instans db.r5b untuk membuat instans DB Amazon RDS yang menjalankan Oracle atau SQL Server. Untuk informasi selengkapnya, lihat Mesin DB yang didukung untuk kelas instans DB.	4 Desember 2020

Dukungan untuk MariaDB 10.2.32	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MariaDB versi 10.2.32. Untuk informasi selengkapnya, lihat Versi MariaDB on Amazon RDS.	25 November 2020
Amazon RDS for SQL Server mendukung Microsoft Business Intelligence Suite di SQL Server 2019	Anda sekarang dapat menjalankan SQL Server Analysis Services, SQL Server Integration Services, dan SQL Server Reporting Services pada instans DB menggunakan versi mayor terbaru. Untuk informasi selengkapnya, lihat Opsi untuk mesin basis data Microsoft SQL Server.	24 November 2020
Amazon RDS for PostgreSQL versi 13 di lingkungan pratinjau basis data	Amazon RDS for PostgreSQL kini mendukung PostgreSQL versi 13 di lingkungan pratinjau basis data. Untuk informasi selengkapnya, lihat Versi PostgreSQL 13.	24 November 2020

Wawasan Performa Amazon RDS memperkenalkan dimensi baru	Anda dapat mengelompokkan beban basis data sesuai dengan grup dimensi untuk jenis basis data (PostgreSQL, MySQL, dan MariaDB), aplikasi (PostgreSQL), dan sesi (PostgreSQL). Amazon RDS juga mendukung dimensi db.name (PostgreSQL, MySQL, dan MariaDB), db.application.name (PostgreSQL), dan db.session_type.name (PostgreSQL). Untuk informasi selengkapnya, lihat Tabel beban teratas.	24 November 2020
Amazon RDS for MariaDB mendukung versi mayor yang baru	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MariaDB versi 10.5. Untuk informasi selengkapnya, lihat Versi MariaDB on Amazon RDS.	23 November 2020
Dukungan untuk MySQL 5.6.49	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 5.6.49. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	20 November 2020

Dukungan untuk MySQL 5.5.62	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 5.5.62. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	20 November 2020
Wawasan Performa mendukung analisis statistik untuk menjalankan kueri PostgreSQL	Anda sekarang dapat menganalisis statistik untuk menjalankan kueri dengan Wawasan Performa untuk instans DB PostgreSQL. Untuk informasi selengkapnya, lihat Statistik untuk PostgreSQL.	18 November 2020
Amazon RDS memperluas dukungan untuk penskalaan otomatis penyimpanan	Anda sekarang dapat mengaktifkan penskalaan otomatis penyimpanan saat membuat replika baca, memulihkan instans DB ke waktu yang ditentukan, atau memulihkan instans DB MySQL dari cadangan Amazon S3. Untuk informasi selengkapnya, lihat Mengelola kapasitas secara otomatis dengan penskalaan otomatis penyimpanan Amazon RDS.	18 November 2020

Amazon RDS for SQL Server mendukung Database Mail	Dengan Database Mail, Anda dapat mengirim pesan email dari instans basis data Amazon RDS for SQL Server. Setelah menentukan penerima email, Anda dapat menambahkan file atau hasil kueri ke pesan yang Anda kirim. Untuk informasi selengkapnya, lihat Menggunakan Database Mail di Amazon RDS for SQL Server.	4 November 2020
Dukungan untuk MySQL 8.0.21	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 8.0.21. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	22 Oktober 2020
Amazon RDS memperluas dukungan untuk mengekspor snapshot ke Amazon S3	Anda sekarang dapat mengekspor data snapshot DB ke Amazon S3 di semua AWS Wilayah komersial. Untuk informasi selengkapnya, lihat Mengekspor data snapshot DB ke Amazon S3.	22 Oktober 2020

Amazon RDS for PostgreSQL mendukung peningkatan replika baca	Dengan Amazon RDS for PostgreSQL, saat Anda melakukan peningkatan versi mayor instans DB primer, replika baca juga ditingkatkan secara otomatis. Untuk informasi selengkapnya, lihat Meningkatkan mesin DB PostgreSQL.	15 Oktober 2020
Amazon RDS for MariaDB, Amazon RDS for MySQL, dan Amazon RDS for PostgreSQL mendukung kelas instans DB Graviton2	Anda sekarang dapat menggunakan instans DB Graviton2 kelas db.m6g.x dan db.r6g.x untuk membuat instans DB Amazon RDS yang menjalankan MariaDB, MySQL, atau PostgreSQL. Untuk informasi selengkapnya, lihat Mesin DB yang Didukung untuk Semua Kelas Instans DB yang Tersedia.	15 Oktober 2020
Amazon RDS for SQL Server mendukung peningkatan ke SQL Server 2019	Anda dapat meningkatkan instans DB SQL Server ke SQL Server 2019. Untuk informasi selengkapnya, lihat Meningkatkan Mesin DB Microsoft SQL Server.	6 Oktober 2020

Amazon RDS for Oracle mendukung penentuan kumpulan karakter nasional	Kumpulan karakter nasional, yang disebut juga sebagai kumpulan karakter NCHAR, digunakan dalam jenis data `NCHAR`, `NVARCHAR2` , dan `NCLOB`. Saat membuat basis data, Anda dapat menentukan AL16UTF16 (default) atau UTF8 sebagai kumpulan karakter NCHAR. Untuk informasi selengkapnya, lihat Kumpulan karakter Oracle yang didukung di Amazon RDS.	2 Oktober 2020
Dukungan untuk MySQL 5.7.31	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 5.7.31. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	1 Oktober 2020
Amazon RDS for PostgreSQL mendukung pengeksporan data ke Amazon S3	Anda dapat mengueri data dari Instans DB PostgreSQL dan mengekspornya secara langsung ke file yang disimpan di bucket Amazon S3. Untuk informasi selengkapnya, lihat Mengekspor data dari instans DB RDS for PostgreSQL ke Amazon S3.	24 September 2020

Amazon RDS for MySQL 8.0 mendukung Percona XtraBackup	Anda sekarang dapat menggunakan Percona XtraBackup untuk memulihkan cadangan ke Amazon RDS for MySQL 8.0 instans DB. Untuk informasi selengkapnya, lihat Memulihkan cadangan ke instans DB MySQL.	17 September 2020
Amazon RDS for SQL Server mendukung pencadangan dan pemulihan native pada instans DB dengan replika baca	Anda dapat memulihkan pencadangan native SQL Server ke instans DB yang memiliki replika baca yang dikonfigurasi. Untuk informasi selengkapnya, lihat Mengimpor dan mengekspor basis data SQL Server.	16 September 2020
Amazon RDS for SQL Server mendukung zona waktu tambahan	Anda dapat mencocokkan zona waktu instans DB Anda dengan zona waktu yang Anda pilih. Untuk informasi selengkapnya, lihat Zona waktu lokal untuk instans DB Microsoft SQL Server.	11 September 2020
Amazon RDS for PostgreSQL versi 13 beta 3 di lingkungan pratinjau basis data	Amazon RDS for PostgreSQL kini mendukung PostgreSQL Versi 13 Beta 3 di Lingkungan Pratinjau Basis Data. Untuk informasi selengkapnya, lihat Versi PostgreSQL 13.	9 September 2020

Amazon RDS for SQL Server mendukung tanda pelacakan 692	Anda sekarang dapat menggunakan tanda pelacakan 692 sebagai parameter pengaktifan menggunakan grup parameter DB. Mengaktifkan tanda pelacakan ini akan menonaktifkan penyisipan cepat saat memuat data secara massal ke heap atau indeks berklaster. Untuk informasi selengkapnya, lihat Menonaktifkan penyisipan cepat selama pemuatan massal.	27 Agustus 2020
Amazon RDS for SQL Server mendukung Microsoft SQL Server 2019	Anda sekarang dapat membuat instans DB RDS yang menggunakan SQL Server 2019. Untuk informasi selengkapnya, lihat Versi Microsoft SQL Server di Amazon RDS.	26 Agustus 2020

RDS for Oracle mendukung basis data replika yang terpasang	Saat membuat atau memodifikasi replika Oracle, Anda dapat menempatkannya dalam mode terpasang. Karena basis data replika tidak menerima koneksi pengguna, basis data replika tidak dapat melayani beban kerja hanya baca. Replika yang terpasang menghapus file log pengulangan yang diarsipkan setelah menerapkannya. Penggunaan primer untuk replika yang terpasang adalah pemulihan bencana lintas Wilayah. Untuk informasi selengkapnya, lihat Gambaran umum replika Oracle.	13 Agustus 2020
RDS for Oracle merencanakan peningkatan instans 11g SE1 LI	Pada tanggal 1 November 2020, kami berencana untuk memulai peningkatan secara otomatis instans Oracle Database 11g SE1 License Included (LI) ke Oracle Database 19c untuk Amazon RDS for Oracle. Semua instans 11g, termasuk instans terpesan, akan dipindahkan ke Pembaruan Rilis (RU) Oracle terkini yang tersedia. Untuk informasi selengkapnya, lihat Mempersiapkan peningkatan otomatis Oracle Database 11g SE1.	31 Juli 2020

Amazon RDS mendukung kelas instans DB Graviton2 baru dalam rilis pratinjau untuk PostgreSQL dan MySQL	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan PostgreSQL atau MySQL yang menggunakan kelas instans DB db.m6g.x dan db.r6g.x. Untuk informasi selengkapnya, lihat Mesin DB yang didukung untuk semua kelas instans DB yang tersedia.	30 Juli 2020
RDS for Oracle mendukung APEX 20.1v1	Anda dapat menggunakan APEX 20.1v1 dengan semua versi Oracle Database yang didukung. Untuk informasi selengkapnya, lihat Oracle Application Express.	28 Juli 2020
Dukungan untuk MySQL 8.0.20	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 8.0.20. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	23 Juli 2020

Amazon RDS for MariaDB dan Amazon RDS for MySQL mendukung kelas instans DB yang baru	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MariaDB dan MySQL yang menggunakan kelas instans DB db.m5.16xlarge, db.m5.8xlarge, db.r5.16xlarge, dan DB.r5.8xlarge. Untuk informasi selengkapnya, lihat Mesin DB yang didukung untuk semua kelas instans DB yang tersedia.	23 Juli 2020
RDS for SQL Server mendukung penonaktifan versi lama TLS dan cipher	Anda dapat mengaktifkan dan menonaktifkan protokol keamanan dan cipher tertentu. Untuk informasi selengkapnya, lihat Mengonfigurasi protokol dan cipher keamanan.	21 Juli 2020
RDS mendukung Oracle Spatial di SE2	Anda dapat menggunakan Oracle Spatial di Standard Edition 2 (SE2) untuk semua versi 12.2, 18c, dan 19c. Untuk informasi selengkapnya, lihat Oracle Spatial.	9 Juli 2020

Amazon RDS mendukung AWS PrivateLink	Amazon RDS sekarang mendukung pembuatan titik akhir Amazon VPC untuk panggilan Amazon RDS API untuk menjaga lalu lintas antara aplikasi dan Amazon RDS di jaringan. AWS Untuk informasi selengkapnya, lihat Amazon RDS dan titik akhir VPC antarmuka (AWS PrivateLink).	9 Juli 2020
Amazon RDS for PostgreSQL versi 9.4.x telah mencapai akhir dukungannya.	Amazon RDS for PostgreSQL tidak lagi mendukung versi 9.4.x. Untuk versi yang didukung, lihat Versi basis data PostgreSQL yang didukung.	8 Juli 2020
Dukungan untuk MariaDB 10.3.23 dan 10.4.13	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MariaDB versi 10.3.23 dan 10.4.13. Untuk informasi selengkapnya, lihat Versi MariaDB on Amazon RDS.	6 Juli 2020
Amazon RDS aktif AWS Outposts	Anda dapat membuat instans DB Amazon RDS di AWS Outposts. Untuk informasi selengkapnya, lihat Menggunakan Amazon RDS on AWS Outposts.	6 Juli 2020

Amazon RDS for Oracle membuat file inventaris secara otomatis	Untuk membuka permintaan layanan untuk pelanggan BYOL, Oracle Support meminta file inventaris yang dibuat oleh Opatch. Amazon RDS for Oracle membuat file inventaris setiap jam secara otomatis di direktori BDUMP. Untuk informasi selengkapnya, lihat Mengakses file Opatch.	6 Juli 2020
Dukungan untuk MySQL 5.7.30 dan 5.6.48	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 5.7.30 dan 5.6.48. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	25 Juni 2020
Amazon RDS for Oracle mendukung ADRCI	Utilitas Automatic Diagnostic Repository Command Interpreter (ADRCI) merupakan alat baris perintah Oracle yang Anda gunakan untuk mengelola data diagnostik. Dengan menggunakan fungsi dalam paket `rdsadmin_adrci_util` Amazon RDS, Anda dapat membuat daftar serta memaketkan masalah dan insiden, dan menampilkan file pelacakan. Untuk informasi selengkapnya, lihat Tugas diagnostik DBA umum untuk instans DB Oracle.	17 Juni 2020

Dukungan untuk MySQL 8.0.19	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 8.0.19. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	2 Juni 2020
MySQL 8.0 mendukung nama tabel huruf kecil	Anda sekarang dapat menetapkan parameter `lower_case_table_names` ke 1 untuk instans DB Amazon RDS yang menjalankan MySQL versi 8.0.19 dan versi 8.0 yang lebih tinggi. Untuk informasi selengkapnya, lihat Pengecualian parameter MySQL untuk instans DB Amazon RDS.	2 Juni 2020
Amazon RDS for Microsoft SQL Server mendukung SQL Server Integration Services (SSIS)	SSIS merupakan platform untuk integrasi data dan aplikasi alur kerja. Anda dapat mengaktifkan SSIS pada instans DB yang sudah ada atau yang baru. Layanan ini diinstal pada instans DB yang sama dengan mesin basis data Anda. Untuk informasi selengkapnya, lihat Dukungan untuk SQL Server Integration Services di SQL Server.	19 Mei 2020

Amazon RDS for Microsoft SQL Server mendukung SQL Server Reporting Services (SSRS)	SSRS merupakan aplikasi berbasis server yang digunakan untuk pembuatan dan distribusi laporan. Anda dapat mengaktifkan SSRS pada instans DB yang sudah ada atau yang baru. Layanan ini diinstal pada instans DB yang sama dengan mesin basis data Anda. Untuk informasi selengkapnya, lihat Dukungan untuk SQL Server Reporting Services di SQL Server.	15 Mei 2020
Amazon RDS for Microsoft SQL Server mendukung integrasi S3 pada instans Multi-AZ	Anda sekarang dapat menggunakan Amazon S3 dengan fitur SQL Server seperti penyisipan massal pada instans DB Multi-AZ. Untuk informasi selengkap nya, lihat Mengintegrasikan instans DB Amazon RDS for SQL Server dengan Amazon S3.	15 Mei 2020
Amazon RDS for Oracle mendukung pembersihan keranjang sampah	Prosedur `rdsadmin.rdsadmin_util.purge_dba_recyclebin` membersihkan keranjang sampah. Untuk informasi selengkapnya, lihat Membersihkan keranjang sampah.	13 Mei, 2020

Amazon RDS for Oracle meningkatkan kemudahan pengelolaan Automatic Workload Repository (AWR)	Prosedur `rdsadmin.rdsadmin_diagnostic_util` membuat laporan AWR dan mengekstrak data AWR ke dalam file dump. Untuk informasi selengkapnya, lihat Membuat laporan performa dengan Automatic Workload Repository (AWR).	13 Mei, 2020
Amazon RDS for Microsoft SQL Server mendukung Microsoft Distributed Transaction Coordinator (MSDTC)	Amazon RDS for SQL Server mendukung transaksi terdistribusi antar-host. Untuk informasi selengkapnya, lihat Dukungan untuk Microsoft Distributed Transaction Coordinator di SQL Server.	4 Mei 2020
Amazon RDS for Microsoft SQL Server mendukung versi yang baru	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan SQL Server versi 2017 CU19 14.00.3281.6, 2016 SP2 CU11 13.00.5598.27, 2014 SP3 CU4 12.00.6329.1, dan 2012 SP4 GDR 11.0.7493.4 untuk semua edisi. Untuk informasi selengkapnya, lihat Versi Microsoft SQL Server di Amazon RDS.	28 April 2020
Amazon RDS tersedia di Wilayah Eropa (Milan)	Amazon RDS kini tersedia di Wilayah Eropa (Milan). Untuk informasi selengkapnya, lihat Wilayah dan Zona Ketersediaan.	28 April 2020

Dukungan Amazon RDS untuk Zona Lokal	Anda sekarang dapat meluncurkan instans DB ke subnet Zona Lokal. Untuk informasi selengkapnya, lihat Wilayah, Zona Ketersediaan, dan Zona Lokal.	23 April 2020
Amazon RDS tersedia di Wilayah Afrika (Cape Town)	Amazon RDS kini tersedia di Wilayah Afrika (Cape Town). Untuk informasi selengkap nya, lihat Wilayah dan Zona Ketersediaan.	22 April 2020
Amazon RDS for Microsoft SQL Server mendukung SQL Server Analysis Services (SSAS)	SSAS merupakan pemrosesa n analitik online (OLAP) dan alat penambangan data yang diinstal dalam SQL Server. Anda dapat mengaktifkan SSAS pada instans DB yang sudah ada atau yang baru. Layanan ini diinstal pada instans DB yang sama dengan mesin basis data Anda. Untuk informasi selengkapnya, lihat Dukungan untuk SQL Server Analysis Services di SQL Server.	17 April 2020

Proksi Amazon RDS untuk PostgreSQL	Proksi Amazon RDS kini tersedia untuk PostgreSQL. Anda dapat menggunakan Proksi RDS untuk mengurangi overhead manajemen koneksi pada instans DB Anda dan juga kemungkinan kesalahan “terlalu banyak koneksi”. Proksi RDS saat ini dalam pratinjau publik untuk PostgreSQL. Untuk informasi selengkapnya, lihat Mengelola koneksi dengan proksi Amazon RDS (pratinjau).	8 April 2020
Amazon RDS for Oracle mendukung Oracle APEX versi 19.2.v1	Amazon RDS for Oracle kini mendukung Oracle Application Express (APEX) versi 19.2.v1. Untuk informasi selengkapnya, lihat Oracle Application Express.	8 April 2020
Amazon RDS for MariaDB mendukung versi mayor yang baru	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MariaDB versi 10.4. Untuk informasi selengkapnya, lihat Versi MariaDB on Amazon RDS.	6 April 2020

Wawasan Performa Amazon RDS tersedia untuk Amazon RDS for MariaDB 10.4	Wawasan Performa Amazon RDS kini tersedia untuk Amazon RDS for MariaDB versi 10.4. Untuk informasi selengkapnya, lihat Menggunakan Wawasan Performa Amazon RDS.	6 April 2020
Amazon RDS for PostgreSQL versi 9.3.x telah mencapai akhir dukungannya	Amazon RDS for PostgreSQL tidak lagi mendukung versi 9.3.x. Untuk versi yang didukung, lihat Versi basis data PostgreSQL yang didukung.	3 April 2020
Amazon RDS for Microsoft SQL Server mendukung replika baca	Anda sekarang dapat membuat replika baca untuk instans DB SQL Server. Untuk informasi selengkapnya, lihat Menggunakan replika baca.	3 April 2020
Amazon RDS for Microsoft SQL Server mendukung pencadangan multi-file	Anda sekarang dapat mencadangkan basis data ke beberapa file menggunakan pencadangan dan pemulihan native SQL Server. Untuk informasi selengkapnya, lihat Mencadangkan basis data.	2 April 2020

Amazon RDS for Oracle integrasi dengan AWS License Manager	Amazon RDS for Oracle sekarang terintegrasi dengan. AWS License Manager Jika Anda menggunakan model Bring Your Own License, AWS License Manager integrasi membuatnya lebih mudah untuk memantau penggunaan lisensi Oracle Anda dalam organisasi Anda. Untuk informasi selengkapnya, lihat Mengintegrasikan dengan AWS License Manager.	23 Maret 2020
Dukungan untuk 64 TiB pada instans db.r5 di Amazon RDS for MariaDB dan Amazon RDS for MySQL	Anda sekarang dapat membuat instans DB Amazon RDS for MariaDB dan Amazon RDS for MySQL yang menggunakan kelas instans DB db.r5 dengan maksimal 64 TiB penyimpanan. Untuk informasi selengkapnya, lihat Faktor yang memengaruhi performa penyimpanan.	18 Maret 2020
Dukungan untuk MySQL 8.0.17	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 8.0.17. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	10 Maret 2020

Wawasan Performa Amazon RDS tersedia untuk Amazon RDS for MySQL 8.0	Wawasan Performa Amazon RDS kini tersedia untuk Amazon RDS for MySQL versi 8.0.17 dan versi 8.0 yang lebih tinggi. Untuk informasi selengkapnya, lihat Menggunakan Wawasan Performa Amazon RDS.	10 Maret 2020
Dukungan untuk MySQL 5.6.46	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 5.6.46. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	28 Februari 2020
Wawasan Performa Amazon RDS tersedia untuk Amazon RDS for MariaDB 10.3	Wawasan Performa Amazon RDS kini tersedia untuk Amazon RDS for MariaDB versi 10.3.13 dan versi 10.3 yang lebih tinggi. Untuk informasi selengkapnya, lihat Menggunakan Wawasan Performa Amazon RDS.	26 Februari 2020
Dukungan untuk MySQL 5.7.28	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 5.7.28. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	20 Februari 2020

Dukungan untuk MariaDB 10.3.20	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MariaDB versi 10.3.20. Untuk informasi selengkapnya, lihat Versi MariaDB on Amazon RDS.	20 Februari 2020
Amazon RDS for Microsoft SQL Server mendukung kelas instans DB yang baru	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan SQL Server yang menggunakan kelas instans DB db.z1d. Untuk informasi selengkap nya, lihat Dukungan kelas instans DB untuk Microsoft SQL Server.	19 Februari 2020
Dukungan untuk domain Direktori Aktif lintas akun dan lintas VPC di Amazon RDS for SQL Server	Amazon RDS for Microsoft SQL Server kini mendukung pengaitan instans DB dengan domain Direktori Aktif yang dimiliki oleh akun dan VPC yang berbeda-beda. Untuk informasi selengkapnya, lihat Menggunakan autentikasi Windows dengan instans DB Microsoft SQL Server.	13 Februari 2020

Opsi Oracle OLAP	Amazon RDS for Oracle kini mendukung opsi Pemrosesan Analitik Online (OLAP) untuk instans DB Oracle. Anda dapat menggunakan Oracle OLAP untuk menganalisis data dalam jumlah besar dengan membuat objek dan kubus dimensi sesuai dengan standar OLAP. Untuk informasi selengkapnya, lihat Oracle OLAP.	13 Februari 2020
Dukungan FIPS 140-2 untuk Oracle	Amazon RDS for Oracle mendukung Federal Information Processing Standard Publication 140-2 (FIPS 140-2) untuk koneksi SSL/TLS. Untuk informasi selengkapnya, lihat Dukungan FIPS.	11 Februari 2020
Amazon RDS for PostgreSQL mendukung kelas instans DB yang baru	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan PostgreSQL yang menggunakan kelas instans DB db.m5.16xlarge, db.m5.8xlarge, db.r5.16xlarge, dan db.r5.8xlarge. Untuk informasi selengkapnya, lihat Mesin DB yang didukung untuk semua kelas instans DB yang tersedia.	11 Februari 2020

Wawasan Performa mendukung analisis statistik untuk menjalankan kueri MariaDB dan MySQL	Anda sekarang dapat menganalisis statistik untuk menjalankan kueri dengan Wawasan Performa untuk instans DB MariaDB dan MySQL. Untuk informasi selengkapnya, lihat Menganalisis statistik kueri yang berjalan.	4 Februari 2020
Dukungan untuk mengekspor data snapshot DB yang ada ke Amazon S3 for MariaDB, Amazon S3 for MySQL, dan Amazon S3 for PostgreSQL	Amazon RDS mendukung pengeksporan data snapshot DB ke Amazon S3 untuk MariaDB, MySQL, dan PostgreSQL. Untuk informasi selengkapnya, lihat Mengekspor data snapshot DB ke Amazon S3.	23 Januari 2020
Amazon RDS for MySQL mendukung autentikasi Kerberos	Anda sekarang dapat menggunakan autentikasi Kerberos untuk mengautentikasi pengguna saat terhubung ke instans DB Amazon RDS for MySQL Anda. Untuk informasi selengkapnya, lihat Menggunakan autentikasi Kerberos untuk MySQL.	21 Januari 2020

Wawasan Performa Amazon RDS mendukung tampilan lebih banyak teks SQL untuk Amazon RDS for Microsoft SQL Server	Wawasan Performa Amazon RDS kini mendukung tampilan lebih banyak teks SQL di dasbor Wawasan Performa untuk instans DB Amazon RDS for Microsoft SQL Server. Untuk informasi selengkap nya, lihat Menampilkan lebih banyak teks SQL di dasbor Wawasan Performa.	17 Desember 2019
Proksi Amazon RDS	Anda dapat mengurang i overhead manajemen koneksi pada klaster, dan mengurangi kemungkinan kesalahan "terlalu banyak koneksi", dengan menggunak an Proksi Amazon RDS. Anda mengaitkan setiap proksi dengan instans DB RDS atau klaster DB Aurora. Kemudian, Anda menggunakan titik akhir proksi dalam string koneksi untuk aplikasi Anda. Proksi Amazon RDS kini dalam status pratinjau publik. Layanan ini mendukung mesin basis data RDS for MySQL. Untuk informasi selengkap nya, lihat Mengelola koneksi dengan proksi Amazon RDS (pratinjau).	3 Desember 2019

Amazon RDS aktif AWS Outposts (pratinjau)	Dengan Amazon RDS aktif AWS Outposts, Anda dapat membuat database relasional yang AWS dikelola di pusat data lokal Anda. RDS on Outposts memungkinkan Anda menjalankan basis data RDS di AWS Outposts. Untuk informasi selengkapnya, lihat Amazon RDS on AWS Outposts (pratinjau).	3 Desember 2019
Amazon RDS for Oracle mendukung replika baca lintas wilayah	Amazon RDS for Oracle kini mendukung replika baca lintas wilayah dengan Active Data Guard. Untuk informasi selengkapnya, lihat Menggunakan replika baca dan Menggunakan replika baca Oracle.	26 November 2019
Wawasan Performa mendukung analisis statistik untuk menjalankan kueri Oracle	Anda sekarang dapat menganalisis statistik kueri yang berjalan dengan Wawasan Performa untuk instans DB Oracle. Untuk informasi selengkapnya, lihat Menganalisis statistik kueri yang berjalan.	25 November 2019

Amazon RDS untuk Microsoft SQL Server mendukung penerbitan log ke Log CloudWatch	Anda dapat mengonfigurasi instans Amazon RDS for SQL Server DB untuk mempublikasikan peristiwa log langsung ke Amazon Logs. CloudWatch Untuk informasi selengkapnya, lihat Menerbitkan log SQL Server ke Amazon CloudWatch Logs.	25 November 2019
Amazon RDS for Microsoft SQL Server mendukung kelas instans DB yang baru	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan SQL Server yang menggunakan kelas instans DB db.x1e dan db.x1. Untuk informasi selengkapnya, lihat Dukungan kelas instans DB untuk Microsoft SQL Server.	25 November 2019
Amazon RDS for Microsoft SQL Server mendukung pemulihan diferensial dan log	Anda dapat memulihkan cadangan dan log diferensial menggunakan pencadangan dan pemulihan native SQL Server. Untuk informasi selengkapnya, lihat Menggunakan pencadangan dan pemulihan native.	25 November 2019
Multi-AZ didukung di Amazon RDS for Microsoft SQL Server di wilayah baru	Multi-AZ di SQL Server kini tersedia di Tiongkok, Timur Tengah (Bahrain), dan Eropa (Stockholm). Untuk informasi selengkapnya, lihat Deployment Multi-AZ untuk Microsoft SQL Server.	22 November 2019

Amazon RDS for Microsoft SQL Server kini mendukung penyisipan massal dan integrasi S3	Anda dapat mentransfer file antara instans DB SQL Server dan bucket Amazon S3. Kemudian, Anda dapat menggunakan Amazon S3 dengan fitur SQL Server seperti penyisipan massal. Untuk informasi selengkapnya, lihat Mengintegrasikan instans DB Amazon RDS for SQL Server dengan Amazon S3.	21 November 2019
Penghitung Wawasan Performa untuk Amazon RDS for Microsoft SQL Server	Anda sekarang dapat menambahkan penghitung performa ke grafik Wawasan Performa Anda untuk instans DB Microsoft SQL Server. Untuk informasi selengkapnya, lihat Penghitung Wawasan Performa untuk Amazon RDS for Microsoft SQL Server.	12 November 2019
Amazon RDS for Microsoft SQL Server mendukung ukuran kelas instans DB yang baru	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan SQL Server yang menggunakan ukuran instans 8xlarge dan 16xlarge untuk kelas instans DB db.m5 dan db.r5. Ukuran instans mulai dari small hingga 2xlarge kini tersedia untuk kelas instans db.t3. Untuk informasi selengkapnya, lihat Dukungan kelas instans DB untuk Microsoft SQL Server.	11 November 2019

Dukungan untuk peningkatan snapshot PostgreSQL	Jika Anda sudah memiliki snapshot DB manual instans DB Amazon RDS for PostgreSQL, Anda sekarang dapat meningkatkan-nya ke mesin basis data PostgreSQL versi lebih baru. Untuk informasi selengkapnya, lihat Meningkatkan snapshot DB PostgreSQL.	7 November 2019
Amazon RDS for Oracle mendukung versi mayor yang baru	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan Oracle Database 19c (19.0). Untuk informasi selengkapnya, lihat Oracle Database 19c dengan Amazon RDS.	7 November 2019
Amazon RDS for PostgreSQL versi 12.0 di lingkungan pratinjau basis data	Amazon RDS for PostgreSQL kini mendukung PostgreSQL Versi 12.0 di Lingkungan Pratinjau Basis Data. Untuk informasi selengkapnya, lihat PostgreSQL versi 12.0 di lingkungan pratinjau basis data.	1 November 2019

Amazon RDS for PostgreSQL mendukung autentikasi Kerberos	Anda sekarang dapat menggunakan autentikasi Kerberos untuk mengautentikasi pengguna saat terhubung ke instans DB Amazon RDS Anda yang menjalankan PostgreSQL Anda. Untuk informasi selengkapnya, lihat Menggunakan autentikasi Kerberos dengan Amazon RDS for PostgreSQL.	28 Oktober 2019
Tugas basis data OEM Management Agent untuk instans DB Oracle	Instans DB Amazon RDS for Oracle kini mendukung prosedur untuk menginvokasi perintah EMCTL tertentu pada Management Agent. Untuk informasi selengkapnya, lihat Tugas basis data agen OEM.	24 Oktober 2019
Amazon RDS for PostgreSQL mendukung basis data PostgreSQL yang dapat dipindahkan	Basis Data yang Dapat Dipindahkan PostgreSQL menyediakan metode yang sangat cepat untuk melakukan migrasi basis data RDS PostgreSQL di antara dua instans DB. Untuk informasi selengkapnya, lihat Memindahkan basis data PostgreSQL antar-instans DB.	8 Oktober 2019

Amazon RDS for Oracle mendukung autentikasi Kerberos	Anda sekarang dapat menggunakan autentikasi Kerberos untuk mengauten tikasi pengguna saat terhubung ke instans DB Amazon RDS Anda yang menjalankan Oracle. Untuk informasi selengkapnya, lihat Menggunakan autentikasi Kerberos dengan Amazon RDS for Oracle.	30 September 2019
Amazon RDS for PostgreSQL versi 12 beta 3 di lingkungan pratinjau basis data	Amazon RDS for PostgreSQ L kini mendukung PostgreSQ L Versi 12 Beta 3 di Lingkunga n Pratinjau Basis Data. Untuk informasi selengkapnya, lihat PostgreSQL versi 12 beta 3 di Amazon RDS di lingkungan pratinjau basis data.	28 Agustus 2019
Dukungan untuk MySQL 8.0.16	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 8.0.16. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	19 Agustus 2019
Amazon RDS for Oracle mendukung versi mayor yang baru	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan Oracle Database 18c (18.0). Untuk informasi selengkapnya, lihat Oracle Database 18c dengan Amazon RDS.	15 Agustus 2019

Management Agent untuk OEM 13c rilis 3	Instans DB Amazon RDS for Oracle kini mendukung Management Agent untuk Oracle Enterprise Manager (OEM) Cloud Control 13c Release 3. Untuk informasi selengkapnya, lihat Oracle Management Agent untuk Enterprise Manager Cloud Control.	7 Agustus 2019
Amazon RDS for PostgreSQL versi 12 beta 2 di lingkungan pratinjau basis data	Amazon RDS for PostgreSQL kini mendukung PostgreSQL Versi 12 Beta 2 di Lingkungan Pratinjau Basis Data. Untuk informasi selengkapnya, lihat PostgreSQL versi 12 beta 2 di Amazon RDS di lingkungan pratinjau basis data.	6 Agustus 2019
Amazon RDS mendukung kolasi server untuk SQL Server	Amazon RDS for SQL Server mendukung pemilihan kolasi untuk instans DB baru. Untuk informasi selengkapnya, lihat Kolasi dan kumpulan karakter untuk Microsoft SQL Server.	29 Juli 2019
Amazon RDS for Oracle mendukung Oracle APEX versi 19.1.v1	Amazon RDS for Oracle kini mendukung Oracle Application Express (APEX) versi 19.1.v1. Untuk informasi selengkapnya, lihat Oracle Application Express.	28 Juni 2019

Amazon RDS for PostgreSQL versi 13 beta 1 di lingkungan pratinjau basis data	Amazon RDS for PostgreSQL kini mendukung PostgreSQL Versi 13 Beta 1 di Lingkungan Pratinjau Basis Data. Untuk informasi selengkapnya, lihat Versi PostgreSQL 13.	22 Juni 2019
Penskalaan otomatis penyimpanan Amazon RDS	Penskalaan otomatis penyimpanan untuk instans Amazon RDS DB memungkinkan Amazon RDS untuk secara otomatis memperluas penyimpanan yang terkait dengan instans DB untuk mengurangi kemungkinan kondisi. out-of-space Untuk informasi tentang penskalaan otomatis penyimpanan, lihat Menggunakan penyimpanan untuk instans DB Amazon RDS.	20 Juni 2019
Amazon RDS for Oracle mendukung kelas instans DB db.z1d	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan Oracle yang menggunakan kelas instans DB db.z1d. Untuk informasi selengkapnya, lihat Kelas instans DB.	13 Juni 2019

Wawasan Performa Amazon RDS mendukung tampilan lebih banyak teks SQL untuk Amazon RDS for Oracle	Wawasan Performa Amazon RDS kini mendukung tampilan lebih banyak teks SQL di dasbor Wawasan Performa untuk instans DB Amazon RDS for Oracle. Untuk informasi selengkapnya, lihat Menampilkan lebih banyak teks SQL di dasbor Wawasan Performa.	10 Juni 2019
Amazon RDS menambahkan dukungan pemulihan native basis data SQL Server hingga 16 TB	Anda sekarang dapat melakukan pemulihan native untuk maksimal 16 TB dari SQL Server ke Amazon RDS. Untuk informasi selengkapnya, lihat Amazon RDS for SQL Server: Batasan dan rekomendasi.	4 Juni 2019
Amazon RDS menambahkan dukungan untuk audit Microsoft SQL Server	Dengan menggunakan Amazon RDS for Microsoft SQL Server, Anda dapat mengaudit peristiwa tingkat server dan basis data menggunakan SQL Server Audit, dan melihat hasilnya pada instans DB Anda atau mengirim file log audit secara langsung ke Amazon S3. Untuk informasi selengkapnya, lihat SQL Server Audit.	23 Mei 2019

Peningkatan pada rekomendasi Amazon RDS	Amazon RDS telah meningkatkan rekomendasi otomatisnya untuk sumber daya basis data. Misalnya, Amazon RDS kini memberikan rekomendasi untuk parameter basis data. Untuk informasi selengkapnya, lihat Menggunakan rekomendasi Amazon RDS.	22 Mei 2019
Dukungan untuk lebih banyak basis data per instans DB untuk Amazon RDS for SQL Server	Anda dapat membuat hingga 30 basis data di setiap instans DB yang menjalankan Microsoft SQL Server. Untuk informasi selengkapnya, lihat Batas untuk instans DB Microsoft SQL Server.	21 Mei 2019
Dukungan untuk 64 TiB dan 80 ribu IOPS penyimpanan untuk Amazon RDS for MariaDB, Amazon RDS for MySQL, dan Amazon RDS for PostgreSQL	Anda sekarang dapat membuat instans DB Amazon RDS for MariaDB, MySQL dan PostgreSQL dengan maksimal 64 TiB penyimpanan dan maksimal 80.000 IOPS yang tersedia. Untuk informasi selengkapnya, lihat Penyimpanan instans DB.	20 Mei 2019
Amazon RDS for MySQL mendukung pemeriksaan awal peningkatan	Saat Anda meningkatkan instans DB dari MySQL 5.7 ke MySQL 8.0, Amazon RDS melakukan pemeriksaan awal terhadap inkompatibilitas. Untuk informasi selengkapnya, lihat Pemeriksaan awal untuk peningkatan dari MySQL 5.7 ke 8.0.	17 Mei 2019

Dukungan untuk plugin validasi kata sandi MySQL	Anda sekarang dapat menggunakan plugin `validate_password` MySQL untuk meningkatkan keamanan instans DB Amazon RDS for MySQL. Untuk informasi selengkapnya, lihat Menggunakan Plugin Validasi Kata Sandi.	16 Mei 2019
Penghitung Wawasan Performa untuk Amazon RDS for Oracle	Anda sekarang dapat menambahkan penghitung performa ke grafik Wawasan Performa Anda untuk instans DB Oracle. Untuk informasi selengkapnya, lihat Penghitung Wawasan Performa untuk Amazon RDS for Oracle.	8 Mei 2019
Dukungan untuk penagihan per detik	Amazon RDS sekarang ditagih dalam kenaikan 1 detik di semua AWS Wilayah kecuali AWS GovCloud (AS) untuk instans sesuai permintaan. Untuk informasi selengkapnya, lihat Penagihan instans DB untuk Amazon RDS.	25 April 2019
Dukungan untuk mengimpor data dari Amazon S3 untuk Amazon RDS for PostgreSQL	Anda sekarang dapat mengimpor data dari file Amazon S3 ke dalam tabel di instans DB RDS for PostgreSQL. Untuk informasi selengkapnya, lihat Mengimpor data Amazon S3 ke dalam instans DB RDS for PostgreSQL.	24 April 2019

Dukungan untuk memulihkan cadangan 5.7 dari Amazon S3	Anda sekarang dapat membuat cadangan basis data MySQL versi 5.7, menyimpannya di Amazon S3 lalu memulihkan file cadangan ke instans DB Amazon RDS baru yang menjalankan MySQL. Untuk informasi selengkapnya, lihat Memulihkan cadangan ke instans DB MySQL.	17 April 2019
Dukungan untuk beberapa peningkatan versi mayor untuk Amazon RDS for PostgreSQL	Dengan Amazon RDS for PostgreSQL, Anda sekarang dapat memilih dari beberapa versi mayor saat Anda meningkatkan mesin DB. Fitur ini memungkinkan Anda langsung beralih ke versi mayor yang lebih baru saat meningkatkan versi mesin PostgreSQL yang dipilih. Untuk informasi selengkapnya, lihat Meningkatkan mesin DB PostgreSQL.	16 April 2019
Dukungan untuk 64 TiB penyimpanan untuk Amazon RDS for Oracle	Anda sekarang dapat membuat instans DB Amazon RDS for Oracle dengan maksimal 64 TiB penyimpanan dan maksimal 80.000 IOPS yang tersedia. Untuk informasi selengkapnya, lihat Penyimpanan instans DB.	4 April 2019

Dukungan untuk MySQL 8.0.15	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 8.0.15. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	3 April 2019
Dukungan untuk MariaDB 10.3.13	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MariaDB versi 10.3.13. Untuk informasi selengkapnya, lihat Versi MariaDB on Amazon RDS.	3 April 2019
Microsoft SQL Server 2008 R2 telah mencapai akhir dukungannya di Amazon RDS	Microsoft SQL Server 2008 R2 telah mencapai akhir dukungannya, bertepatan dengan rencana Microsoft untuk mengakhiri dukungan yang diperluas untuk versi ini pada tanggal 9 Juli 2019. Snapshot Microsoft SQL Server 2008 R2 yang ada akan ditingkatkan secara otomatis ke versi minor terbaru Microsoft SQL Server 2012 mulai tanggal 1 Juni 2019. Untuk informasi selengkapnya, lihat Dukungan Microsoft SQL Server 2008 R2 di Amazon RDS.	2 April 2019

Grup ketersediaan Selalu Aktif didukung di Microsoft SQL Server 2017	Anda sekarang dapat menggunakan Grup Ketersediaan Selalu Aktif di SQL Server 2017 Enterprise Edition 14.00.3049.1 atau versi yang lebih baru. Untuk informasi selengkapnya, lihat Deployment Multi-AZ untuk Microsoft SQL Server.	29 Maret 2019
Lihat metrik volume	Anda sekarang dapat melihat metrik untuk volume Amazon Elastic Block Store (Amazon EBS), yang merupakan perangkat fisik yang digunakan untuk penyimpanan basis data dan log. Untuk informasi selengkapnya, lihat Melihat Pemantauan yang Ditingkatkan.	20 Maret 2019
Dukungan untuk MySQL 5.7.25	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 5.7.25. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	19 Maret 2019

Amazon RDS for Oracle mendukung tugas DBA RMAN	Amazon RDS for Oracle kini mendukung tugas DBA Oracle Recovery Manager (RMAN), termasuk pencadangan RMAN. Untuk informasi selengkapnya, lihat Tugas Common DBA Recovery Manager (RMAN) untuk instans DB Oracle.	14 Maret 2019
Amazon RDS for PostgreSQL mendukung versi 11.1	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan PostgreSQL versi 11.1. Untuk informasi selengkapnya, lihat PostgreSQL versi 11.1 di Amazon RDS.	12 Maret 2019
Pemulihan beberapa file tersedia di Amazon RDS for SQL Server	Anda sekarang dapat melakukan pemulihan dari beberapa file dengan Amazon RDS for SQL Server. Untuk informasi selengkapnya, lihat Memulihkan basis data.	11 Maret 2019
MariaDB 10.2.21	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MariaDB versi 10.2.21. Untuk informasi selengkapnya, lihat Versi MariaDB on Amazon RDS.	11 Maret 2019

Amazon RDS for Oracle mendukung replika baca	Amazon RDS for Oracle kini mendukung replika baca dengan Active Data Guard. Untuk informasi selengkapnya, lihat Menggunakan replika baca dan Menggunakan replika baca Oracle.	11 Maret 2019
Wawasan Performa Amazon RDS tersedia untuk Amazon RDS for MariaDB	Wawasan Performa Amazon RDS kini tersedia untuk Amazon RDS for MariaDB. Untuk informasi selengkapnya, lihat Menggunakan Wawasan Performa Amazon RDS.	11 Maret 2019
MySQL 8.0.13 dan 5.7.24	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 8.0.13 dan 5.7.24. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	8 Maret 2019
Wawasan Performa Amazon RDS tersedia untuk Amazon RDS for SQL Server	Wawasan Performa Amazon RDS kini tersedia untuk Amazon RDS for SQL Server. Untuk informasi selengkapnya, lihat Menggunakan Wawasan Performa Amazon RDS.	4 Maret 2019

Amazon RDS for Oracle mendukung integrasi Amazon S3	Anda sekarang dapat mentransfer file antara instans DB Amazon RDS for Oracle dan bucket Amazon S3. Untuk informasi selengkapnya, lihat Mengintegrasikan Amazon RDS for Oracle dan Amazon S3.	26 Februari 2019
Amazon RDS for MySQL dan Amazon RDS for MariaDB mendukung kelas instans DB db.t3	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL atau MariaDB yang menggunakan kelas instans DB db.t3. Untuk informasi selengkapnya, lihat Kelas instans DB.	20 Februari 2019
Amazon RDS for MySQL dan Amazon RDS for MariaDB mendukung kelas instans DB db.r5	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL atau MariaDB yang menggunakan kelas instans DB db.r5. Untuk informasi selengkapnya, lihat Kelas instans DB.	20 Februari 2019
Penghitung Wawasan Performa untuk RDS for MySQL dan RDS for PostgreSQL	Anda sekarang dapat menambahkan penghitung performa ke grafik Wawasan Performa untuk instans DB MySQL dan PostgreSQL. Untuk informasi selengkap nya, lihat Komponen dasbor Wawasan Performa.	19 Februari 2019

Amazon RDS for PostgreSQL kini mendukung pengaturan parameter autovacuum adaptif	Penyesuaian parameter autovacuum adaptif dengan Amazon RDS for PostgreSQL membantu mencegah wraparound ID transaksi dengan menyesuaikan nilai parameter autovacuum secara otomatis. Untuk informasi selengkapnya, lihat Mengurangi kemungkinan wraparound ID transaksi.	12 Februari 2019
Amazon RDS for Oracle mendukung Oracle APEX versi 18.1.v1 dan 18.2.v1	Amazon RDS for Oracle kini mendukung Oracle Application Express (APEX) versi 18.1.v1 dan 18.2.v1. Untuk informasi selengkapnya, lihat Oracle Application Express.	11 Februari 2019
Wawasan Performa Amazon RDS mendukung tampilan lebih banyak teks SQL untuk RDS for MySQL	Sekarang Wawasan Performa Amazon RDS mendukung tampilan lebih banyak teks SQL dalam dasbor Wawasan Performa untuk instans DB MySQL. Untuk informasi selengkapnya, lihat Menampilkan lebih banyak teks SQL di dasbor Wawasan Performa.	6 Februari 2019
Amazon RDS for PostgreSQL mendukung kelas instans DB db.t3	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan PostgreSQL yang menggunakan kelas instans DB db.t3. Untuk informasi selengkapnya, lihat Kelas instans DB.	25 Januari 2019

Amazon RDS for Oracle mendukung kelas instans DB db.t3	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan Oracle yang menggunakan kelas instans DB db.t3. Untuk informasi selengkapnya, lihat Kelas instans DB.	25 Januari 2019
Wawasan Performa Amazon RDS mendukung tampilan lebih banyak teks SQL untuk Amazon RDS for PostgreSQL	Wawasan Performa Amazon RDS kini mendukung tampilan lebih banyak teks SQL di dasbor Wawasan Performa untuk instans DB Amazon RDS for PostgreSQL. Untuk informasi selengkapnya, lihat Menampilkan lebih banyak teks SQL di dasbor Wawasan Performa.	24 Januari 2019
Amazon RDS for Oracle mendukung versi SQLT yang baru	Amazon RDS for Oracle kini mendukung SQLT versi 12.2.180725. Untuk informasi selengkapnya, lihat Oracle SQLT.	22 Januari 2019
Amazon RDS for PostgreSQL mendukung kelas instans DB db.r5	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan PostgreSQL yang menggunakan kelas instans DB db.r5. Untuk informasi selengkapnya, lihat Kelas instans DB.	19 Desember 2018

Amazon RDS for PostgreSQL kini mendukung manajemen kata sandi terbatas	Amazon RDS for PostgreSQL memungkinkan Anda membatasi siapa yang dapat mengelola kata sandi pengguna dan perubahan kedaluwarsa kata sandi dengan menggunakan parameter `rds.restrict_password_commands` dan peran `rds_password`. Untuk informasi selengkapnya, lihat Membatasi pengelolaan kata sandi.	19 Desember 2018
Amazon RDS for PostgreSQL mendukung pengunggahan log database ke Amazon Logs CloudWatch	Amazon RDS for PostgreSQL mendukung pengunggahan log database ke Log. CloudWatch Untuk informasi selengkapnya, lihat Menerbitkan log PostgreSQL ke Log. CloudWatch	10 Desember 2018
Amazon RDS for Oracle mendukung kelas instans DB db.r5	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan Oracle yang menggunakan kelas instans DB db.r5. Untuk informasi selengkapnya, lihat Kelas instans DB.	20 November 2018
Pertahankan cadangan saat menghapus instans DB	Amazon RDS mendukung penyimpanan pencadangan otomatis saat Anda menghapus instans DB. Untuk informasi selengkapnya, lihat Menggunakan cadangan.	15 November 2018

Amazon RDS for PostgreSQL mendukung kelas instans DB db.m5	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan PostgreSQL yang menggunakan kelas instans DB db.m5. Untuk informasi selengkapnya, lihat Kelas instans DB.	15 November 2018
Amazon RDS for Oracle mendukung versi mayor yang baru	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan Oracle versi 12.2. Untuk informasi selengkapnya, lihat Oracle Database 12c Rilis 2 (12.2.0.1) dengan Amazon RDS.	13 November 2018
Amazon RDS for SQL Server mendukung Selalu Aktif	Amazon RDS for SQL Server mendukung Grup Ketersediaan Selalu Aktif. Untuk informasi selengkapnya, lihat Deployment Multi-AZ untuk Microsoft SQL Server.	8 November 2018
Amazon RDS for PostgreSQL mendukung akses jaringan keluar menggunakan server DNS kustom	Amazon RDS for PostgreSQL mendukung akses jaringan keluar menggunakan server DNS kustom. Untuk informasi selengkapnya, lihat Menggunakan server DNS kustom untuk akses jaringan keluar.	8 November 2018

Amazon RDS for MariaDB, Amazon RDS for MySQL, dan Amazon RDS for PostgreSQL mendukung 32 TiB penyimpanan	Anda sekarang dapat membuat instans DB Amazon RDS dengan maksimal 32 TiB penyimpanan untuk MySQL, MariaDB, dan PostgreSQL. Untuk informasi selengkapnya, lihat Penyimpanan instans DB.	7 November 2018
Amazon RDS for Oracle mendukung jenis data yang diperluas	Anda sekarang dapat mengaktifkan jenis data yang diperluas pada instans DB Amazon RDS yang menjalankan Oracle. Dengan jenis data yang diperluas , ukuran maksimumnya adalah 32.767 byte untuk jenis data VARCHAR2, NVARCHAR2, dan RAW. Untuk informasi selengkapnya, lihat Menggunakan jenis data yang diperluas.	6 November 2018
Amazon RDS for Oracle mendukung kelas instans DB db.m5	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan Oracle yang menggunakan kelas instans DB db.m5. Untuk informasi selengkapnya, lihat Kelas instans DB.	2 November 2018

Migrasi Amazon RDS for Oracle dari SE, SE1, atau SE2 ke EE	Anda sekarang dapat bermigrasi dari Oracle Database Standard Edition (SE, SE1, atau SE2) ke Oracle Database Enterprise Edition (EE). Untuk informasi selengkapnya, lihat Bermigrasi antar-edisi Oracle.	31 Oktober 2018
Amazon RDS kini dapat menghentikan instans Multi-AZ	Amazon RDS kini dapat menghentikan instans DB yang merupakan bagian dari deployment Multi-AZ. Sebelumnya, fitur penghenti an instans memiliki batasan untuk instans Multi-AZ. Untuk informasi selengkapnya, lihat Menghentikan sementara instans DB Amazon RDS.	29 Oktober 2018
Wawasan Performa Amazon RDS tersedia untuk Amazon RDS for Oracle	Wawasan Performa Amazon RDS kini tersedia untuk Amazon RDS for Oracle. Untuk informasi selengkapnya, lihat Menggunakan Wawasan Performa Amazon RDS.	29 Oktober 2018
Amazon RDS for PostgreSQL mendukung PostgreSQL versi 11 di lingkungan pratinjau basis data	Amazon RDS for PostgreSQ L kini mendukung PostgreSQ L versi 11 di Lingkungan Pratinjau Basis Data. Untuk informasi selengkapnya, lihat PostgreSQL versi 11 di Amazon RDS di lingkungan pratinjau basis data.	25 Oktober 2018

MySQL mendukung versi mayor yang baru	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 8.0. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	23 Oktober 2018
MariaDB mendukung versi mayor yang baru	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MariaDB versi 10.3. Untuk informasi selengkapnya, lihat Versi MariaDB on Amazon RDS.	23 Oktober 2018
Amazon RDS for Oracle mendukung Oracle JVM	Amazon RDS for Oracle kini mendukung opsi Oracle Java Virtual Machine (JVM). Untuk informasi selengkapnya, lihat Mesin virtual Oracle Java.	16 Oktober 2018
Grup parameter kustom untuk pengembalian dan pemulihan titik waktu	Anda sekarang dapat menentukan grup parameter kustom saat memulihkan snapshot atau melakukan operasi pemulihan titik waktu tertentu. Untuk informasi selengkapnya, lihat Memulihkan dari snapshot DB dan Memulihkan instans DB ke waktu yang ditentukan.	15 Oktober 2018

Amazon RDS for Oracle mendukung penyimpanan 32 TiB	Anda sekarang dapat membuat instans DB Oracle RDS dengan maksimal 32 TiB penyimpanan. Untuk informasi selengkapnya, lihat Penyimpanan instans DB.	15 Oktober 2018
Amazon RDS for MySQL mendukung GTID	Amazon RDS for MySQL kini mendukung pengident ifikasi transaksi global (GTID), yang bersifat unik di semua instans DB dan dalam konfigurasi replikasi. Untuk informasi selengkapnya, lihat Menggunakan replikasi berbasis GTID untuk RDS for MySQL.	10 Oktober 2018
MySQL 5.7.23, 5.6.41, dan 5.5.61	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 5.7.23, 5.6.41, dan 5.5.61. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	8 Oktober 2018
Amazon RDS for Oracle mendukung versi SQLT yang baru	Amazon RDS for Oracle kini mendukung SQLT versi 12.2.180331. Untuk informasi selengkapnya, lihat Oracle SQLT.	4 Oktober 2018

Amazon RDS for PostgreSQL kini mendukung autentikasi IAM	Amazon RDS for PostgreSQL kini mendukung autentikasi IAM. Untuk informasi selengkapnya, lihat Autentikasi basis data IAM untuk MySQL dan PostgreSQL.	27 September 2018
Anda dapat mengaktifkan perlindungan penghapusan untuk instans DB Amazon RDS Anda	Saat Anda mengaktifkan perlindungan penghapusan untuk instans DB, basis data tidak dapat dihapus oleh pengguna mana pun. Untuk informasi selengkapnya, lihat Menghapus instans DB.	26 September 2018
Amazon RDS for MySQL dan Amazon RDS for MariaDB mendukung kelas instans DB db.m5	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL atau MariaDB yang menggunakan kelas instans DB db.m5. Untuk informasi selengkapnya, lihat Kelas instans DB.	18 September 2018
Amazon RDS kini mendukung peningkatan ke SQL Server 2017	Anda dapat meningkatkan instans DB yang sudah ada ke SQL Server 2017 dari versi apa pun, kecuali SQL Server 2008. Untuk meningkatkan dari SQL Server 2008, pertama-tama, tingkatkan ke salah satu versi lain terlebih dahulu. Untuk informasi, lihat Meningkatkan mesin DB Microsoft SQL Server.	11 September 2018

Amazon RDS for PostgreSQL kini mendukung PostgreSQL versi 11 beta 3 di lingkungan pratinjau basis data	Dalam rilis ini, ukuran segmen Write-Ahead Log (WAL) (wal_segment_size) kini ditetapkan ke 64 MB. Untuk informasi selengkapnya tentang PostgreSQL versi 11 Beta 3, lihat PostgreSQL 11 beta 3 dirilis. Untuk informasi tentang Lingkungan Pratinjau Basis Data, lihat Menggunakan lingkungan pratinjau basis data.	7 September 2018
Panduan Pengguna Amazon Aurora	Panduan Pengguna Amazon Aurora menjelaskan semua konsep Amazon Aurora dan memberikan petunjuk tentang penggunaan berbagai fitur dengan konsol dan antarmuka baris perintah. Panduan Pengguna Amazon RDS kini mencakup mesin basis data non-Aurora.	31 Agustus 2018
Wawasan Performa Amazon RDS tersedia untuk RDS for MySQL	Wawasan Performa Amazon RDS kini tersedia untuk RDS for MySQL. Untuk informasi selengkapnya, lihat Menggunakan Wawasan Performa Amazon RDS.	28 Agustus 2018

Aurora Edisi Kompatibel PostgreSQL kini mendukung Penskalaan Otomatis Aurora	Penskalaan Otomatis replika Aurora kini tersedia untuk Aurora Edisi Kompatibel PostgreSQL. Untuk informasi selengkapnya, lihat Menggunakan penskalaan otomatis Amazon Aurora dengan replika Aurora.	16 Agustus 2018
Aurora Serverless for Aurora MySQL	Aurora Serverless merupakan konfigurasi penskalaan otomatis sesuai permintaan untuk Amazon Aurora. Untuk informasi selengkapnya, lihat Menggunakan Amazon Aurora Serverless.	9 Agustus 2018
MySQL 5.7.22 dan 5.6.40	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 5.7.22 dan 5.6.40. Untuk informasi selengkapnya, lihat Versi MySQL on Amazon RDS.	6 Agustus 2018
Aurora kini tersedia di wilayah Tiongkok (Ningxia)	Aurora MySQL dan Aurora PostgreSQL kini tersedia di wilayah Tiongkok (Ningxia). Untuk informasi selengkapnya, lihat Ketersediaan untuk Amazon Aurora MySQL dan Ketersediaan untuk Amazon Aurora PostgreSQL.	6 Agustus 2018

Amazon RDS for MySQL mendukung replikasi tertunda	Amazon RDS for MySQL kini mendukung replikasi tertunda sebagai strategi pemulihan bencana. Untuk informasi selengkapnya, lihat Mengonfigurasi replikasi tertunda dengan MySQL.	6 Agustus 2018
Wawasan Performa Amazon RDS tersedia untuk Aurora MySQL	Wawasan Performa Amazon RDS kini tersedia untuk Aurora MySQL. Untuk informasi selengkapnya, lihat Menggunakan Wawasan Performa Amazon RDS.	6 Agustus 2018
Integrasi Performance Insights Amazon RDS dengan Amazon CloudWatch	Amazon RDS Performance Insights secara otomatis menerbitkan metrik ke Amazon. CloudWatch Untuk informasi selengkapnya, lihat Metrik Performance Insights yang dipublikasikan ke. CloudWatch	6 Agustus 2018
Rekomendasi Amazon RDS	Amazon RDS kini memberikan rekomendasi otomatis untuk sumber daya basis data. Untuk informasi selengkapnya, lihat Menggunakan rekomendasi Amazon RDS.	25 Juli 2018

Salinan snapshot tambahan di seluruh Wilayah AWS	Amazon RDS mendukung salinan snapshot tambahan di seluruh AWS Wilayah untuk instans yang tidak terenkripsi dan terenkripsi. Untuk informasi selengkapnya, lihat Menyalin snapshot di seluruh AWS Wilayah.	24 Juli 2018
Wawasan Performa Amazon RDS tersedia untuk Amazon RDS for PostgreSQL	Wawasan Performa Amazon RDS kini tersedia untuk Amazon RDS for PostgreSQL. Untuk informasi selengkapnya, lihat Menggunakan Wawasan Performa Amazon RDS.	18 Juli 2018
Amazon RDS for Oracle mendukung Oracle APEX versi 5.1.4.v1	Amazon RDS for Oracle kini mendukung Oracle Application Express (APEX) versi 5.1.4.v1. Untuk informasi selengkapnya, lihat Oracle Application Express.	10 Juli 2018
Amazon RDS for Oracle mendukung penerbitan log ke Amazon Logs CloudWatch	Amazon RDS for Oracle sekarang mendukung penerbitan data peringatan, audit, penelusuran, dan log pendengar ke grup log di Log. CloudWatch Untuk informasi selengkapnya, lihat Menerbitkan log Oracle ke Amazon CloudWatch Logs.	9 Juli 2018

MariaDB 10.2.15, 10.1.34, dan 10.0.35	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MariaDB versi 10.2.15, 10.1.34, dan 10.0.35. Untuk informasi selengkapnya, lihat Versi MariaDB on Amazon RDS.	5 Juli 2018
Aurora PostgreSQL 1.2 tersedia dan kompatibel dengan PostgreSQL 9.6.8	Aurora PostgreSQL 1.2 kini tersedia dan kompatibel dengan PostgreSQL 9.6.8. Untuk informasi selengkapnya, lihat Versi 1.2.	27 Juni 2018
Replika baca untuk Amazon RDS PostgreSQL mendukung deployment Multi-AZ	Replika baca RDS di Amazon RDS for PostgreSQL kini mendukung beberapa Zona Ketersediaan. Untuk informasi selengkapnya, lihat Menggunakan replika baca PostgreSQL.	25 Juni 2018
Wawasan Performa tersedia untuk Aurora PostgreSQL	Wawasan Performa tersedia secara umum untuk Aurora PostgreSQL, dengan dukungan untuk retensi data performa yang diperluas. Untuk informasi selengkapnya, lihat Menggunakan Wawasan Performa Amazon RDS.	21 Juni 2018

Aurora PostgreSQL tersedia di wilayah AS bagian barat (California utara)	Aurora PostgreSQL kini tersedia di wilayah Amerika Serikat bagian barat (California Utara). Untuk informasi selengkapnya, lihat Ketersediaan untuk Amazon Aurora PostgreSQL.	11 Juni 2018
Amazon RDS for Oracle kini mendukung konfigurasi CPU	Amazon RDS for Oracle mendukung konfigurasi jumlah inti CPU dan jumlah thread di setiap inti untuk prosesor kelas instans DB. Untuk informasi selengkapnya, lihat Mengonfigurasi prosesor kelas instans DB.	5 Juni 2018

Pembaruan sebelumnya

Tabel berikut menjelaskan perubahan penting dalam setiap rilis Panduan Pengguna Amazon RDS sebelum Juni 2018.

Perubahan	Deskripsi	Tanggal diubah
Amazon RDS for PostgreSQL kini mendukung PostgreSQL Versi 11 Beta 1 di Lingkungan Pratinjau Basis Data	PostgreSQL versi 11 Beta 1 mencakup beberapa peningkatan yang dijelaskan dalam PostgreSQL 11 beta 1 dirilis! Untuk informasi tentang Lingkungan Pratinjau Basis Data, lihat Menggunakan lingkungan Pratinjau Basis Data.	31 Mei 2018
Amazon RDS for Oracle sekarang	Amazon RDS for Oracle mendukung Keamanan Lapisan Pengangkutan (TLS) versi 1.0 dan 1.2. Untuk	30 Mei 2018

Perubahan	Deskripsi	Tanggal diubah
mendukung TLS versi 1.0 dan 1.2	informasi selengkapnya, lihat Versi TLS untuk opsi SSL Oracle.	
Aurora MySQL mendukung penerbitan log ke Amazon Logs CloudWatch	Aurora MySQL sekarang mendukung penerbitan data log umum, lambat, audit, dan kesalahan ke grup log di Log. CloudWatch Untuk informasi selengkapnya, lihat Menerbitkan Aurora MySQL ke Log. CloudWatch	23 Mei 2018
Lingkungan Pratinjau Basis Data untuk Amazon RDS PostgreSQL	Anda sekarang dapat meluncurkan instans baru Amazon RDS PostgreSQL dalam mode pratinjau. Untuk informasi selengkapnya tentang Lingkungan Pratinjau Basis Data, lihat Menggunakan lingkungan Pratinjau Basis Data.	22 Mei 2018
Instans DB Amazon RDS for Oracle mendukung kelas instans DB yang baru	Instans DB Oracle sekarang mendukung kelas instans DB db.x1e dan db.x1. Untuk informasi lebih lanjut, lihat Kelas instans DB dan Kelas instans RDS for Oracle.	22 Mei 2018
Amazon RDS PostgreSQL sekarang mendukung postgres_fdw pada replika baca.	Anda sekarang dapat menggunakan postgres_fdw untuk terhubung ke server jarak jauh dari replika baca. Untuk informasi selengkapnya, lihat Menggunakan ekstensi postgres_fdw untuk mengakses data eksternal.	17 Mei 2018
Amazon RDS for Oracle sekarang mendukung pengaturan parameter sqlnet.ora	Anda sekarang dapat mengatur parameter sqlnet.ora dengan Amazon RDS for Oracle. Untuk informasi selengkapnya, lihat Memodifikasi properti koneksi menggunakan parameter sqlnet.ora.	10 Mei 2018

Perubahan	Deskripsi	Tanggal diubah
Aurora PostgreSQL tersedia di wilayah Asia Pasifik (Seoul).	Aurora PostgreSQL sekarang tersedia di wilayah Asia Pasifik (Seoul). Untuk informasi selengkapnya, lihat Ketersediaan untuk Amazon Aurora PostgreSQL.	9 Mei 2018
Aurora MySQL mendukung pelacakan mundur	Aurora MySQL sekarang mendukung "mengembalikan" klaster DB ke waktu tertentu, tanpa memulihkan data dari cadangan. Untuk informasi selengkapnya, lihat Menelusuri balik klaster Aurora DB.	9 Mei 2018
Aurora MySQL migrasi dan replikasi terenkripsi dari MySQL eksternal	Aurora MySQL sekarang mendukung migrasi dan replikasi terenkripsi dari basis data MySQL eksternal. Untuk informasi selengkapnya, lihat Memigrasikan data dari basis data MySQL eksternal ke klaster DB Amazon Aurora MySQL dan Replikasi antara Aurora dan MySQL atau antara Aurora dan klaster DB Aurora lainnya.	25 April 2018
Dukungan Aurora Edisi Kompatibel PostgreSQL untuk protokol Copy-on-Write.	Anda sekarang dapat mengkloning basis data di klaster basis data Aurora PostgreSQL. Untuk informasi selengkapnya, lihat Mengkloning basis data di klaster DB Aurora.	10 April 2018
MariaDB 10.2.12, 10.1.31, dan 10.0.34	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MariaDB versi 10.2.12, 10.1.31, dan 10.0.34. Untuk informasi selengkapnya, lihat Versi-versi MariaDB pada Amazon RDS.	21 Maret 2018
Dukungan Aurora PostgreSQL untuk wilayah baru	Aurora PostgreSQL sekarang tersedia di wilayah UE (London) dan Asia Pasifik (Singapura). Untuk informasi selengkapnya, lihat Ketersediaan untuk Amazon Aurora PostgreSQL.	13 Maret 2018

Perubahan	Deskripsi	Tanggal diubah
MySQL 5.7.21, 5.6.39, dan 5.5.59	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 5.7.21, 5.6.39, dan 5.5.59. Untuk informasi selengkapnya, lihat Versi MySQL di Amazon RDS.	9 Maret 2018
Amazon RDS for Oracle sekarang mendukung Oracle REST Data Services	Amazon RDS for Oracle mendukung Oracle REST Data Services sebagai bagian dari opsi APEX. Untuk informasi selengkapnya, lihat Oracle Application Express (APEX).	9 Maret 2018
Amazon Aurora Edisi yang kompatibel dengan MySQL tersedia di Wilayah baru AWS	Aurora MySQL kini tersedia di wilayah Asia Pasifik (Singapura). Untuk daftar lengkap AWS Wilayah untuk Aurora MySQL, lihat Ketersediaan untuk Amazon Aurora MySQL.	6 Maret 2018
Instans DB Amazon RDS yang menjalankan Microsoft SQL Server mendukung pengambilan data perubahan (CDC)	Instans DB yang menjalankan Amazon RDS for Microsoft SQL Server kini mendukung pengambilan data perubahan (CDC). Untuk informasi selengkapnya, lihat Dukungan Change Data Capture (CDC) untuk instans DB Microsoft SQL Server.	6 Februari 2018
Aurora MySQL mendukung versi mayor yang baru	Anda sekarang dapat membuat klaster DB Aurora MySQL yang menjalankan MySQL versi 5.7. Untuk informasi selengkapnya, lihat Pembaruan mesin basis data Amazon Aurora MySQL 2018-02-06.	6 Februari 2018

Perubahan	Deskripsi	Tanggal diubah
Publikasikan log MySQL dan MariaDB ke Amazon Logs CloudWatch	Anda sekarang dapat mempublikasikan data log MySQL dan MariaDB ke Log. CloudWatch Untuk informasi selengkapnya, lihat Menerbitkan log MySQL ke Amazon Logs CloudWatch dan Menerbitkan log MariaDB ke Log Amazon CloudWatch .	17 Januari 2018
Dukungan Multi-AZ untuk replika baca	Anda sekarang dapat membuat replika baca sebagai instans DB Multi-AZ. Amazon RDS membuat instans siaga replika Anda di Zona Ketersediaan lain untuk dukungan failover untuk replika tersebut. Membuat replika baca Anda sebagai instans DB Multi-AZ tidak tergantung pada apakah basis data sumber adalah instans DB Multi-AZ. Untuk informasi selengkapnya, lihat Menggunakan replika baca instans DB.	11 Januari 2018
Amazon RDS for MariaDB mendukung versi mayor yang baru	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MariaDB versi 10.2. Untuk informasi selengkapnya, lihat dukungan MariaDB 10.2 di Amazon RDS.	3 Januari 2018
Amazon Aurora Edisi Kompatibel PostgreSQL tersedia di Wilayah AWS baru	Aurora PostgreSQL kini tersedia di wilayah UE (Paris). Untuk daftar lengkap AWS Wilayah Aurora PostgreSQL, lihat Ketersediaan untuk Amazon Aurora PostgreSQL.	22 Desember 2017
Aurora PostgreSQL mendukung jenis instans baru.	Aurora PostgreSQL sekarang mendukung jenis instans baru. Untuk daftar lengkap jenis instans, lihat Memilih kelas instans DB.	20 Desember 2017

Perubahan	Deskripsi	Tanggal diubah
Amazon Aurora Edisi yang kompatibel dengan MySQL tersedia di Wilayah baru AWS	Aurora MySQL kini tersedia di wilayah UE (Paris). Untuk daftar lengkap AWS Wilayah untuk Aurora MySQL, lihat Ketersediaan untuk Amazon Aurora MySQL.	18 Desember 2017
Aurora MySQL mendukung hash join	Fitur ini dapat meningkatkan performa kueri saat Anda perlu menggabungkan data dalam jumlah besar dengan menggunakan equijoin. Untuk informasi selengkapnya, lihat Menggunakan hash join di Aurora MySQL.	11 Desember 2017
Aurora MySQL mendukung fungsi native untuk menginvokasi fungsi AWS Lambda	Anda dapat memanggil fungsi native `lambda_sync` dan `lambda_async` saat Anda menggunakan Aurora MySQL. Untuk informasi selengkapnya, lihat Menginvokasi fungsi Lambda dari klaster DB Amazon Aurora MySQL.	11 Desember 2017
Menambahkan kelayakan HIPAA Aurora PostgreSQL	Aurora PostgreSQL sekarang mendukung pembuatan aplikasi yang mematuhi HIPAA. Untuk informasi selengkapnya, lihat Menggunakan Amazon Aurora PostgreSQL.	6 Desember 2017
AWS Wilayah Tambahan tersedia untuk Amazon Aurora dengan kompatibilitas PostgreSQL	Amazon Aurora dengan kompatibilitas PostgreSQL sekarang tersedia di empat Wilayah baru. AWS Untuk informasi selengkapnya, lihat Ketersediaan untuk Amazon Aurora PostgreSQL.	22 November 2017

Perubahan	Deskripsi	Tanggal diubah
Ubah penyimpanan untuk instans DB Amazon RDS yang menjalankan Microsoft SQL Server	Anda sekarang dapat mengubah penyimpanan instans DB Amazon RDS Anda yang menjalankan SQL Server. Untuk informasi selengkapnya, lihat Memodifikasi instans DB Amazon RDS.	21 November 2017
Amazon RDS mendukung penyimpanan 16 TiB untuk mesin berbasis Linux	Anda sekarang dapat membuat instans DB MySQL, MariaDB, PostgreSQL, dan Oracle RDS dengan maksimal 16 TiB penyimpanan. Untuk informasi selengkapnya, lihat Penyimpanan instans DB Amazon RDS.	21 November 2017
Amazon RDS mendukung peningkatan skala penyimpanan dengan cepat	Anda sekarang dapat menambahkan penyimpanan ke instans DB MySQL, MariaDB, PostgreSQL, dan Oracle RDS dalam beberapa menit. Untuk informasi selengkapnya, lihat Penyimpanan instans DB Amazon RDS.	21 November 2017
Amazon RDS mendukung MariaDB versi 10.1.26 dan 10.0.32	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MariaDB versi 10.1.26 dan 10.0.32. Untuk informasi selengkapnya, lihat Versi-versi MariaDB pada Amazon RDS.	20 November 2017
Amazon RDS for Microsoft SQL Server kini mendukung kelas instans DB yang baru	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan SQL Server yang menggunakan kelas instans DB db.r4 dan db.m4.16xlarge. Untuk informasi selengkapnya, lihat Dukungan kelas instans DB untuk Microsoft SQL Server.	20 November 2017

Perubahan	Deskripsi	Tanggal diubah
Amazon RDS for MySQL dan Amazon RDS for MariaDB kini mendukung kelas instans DB yang baru	Anda sekarang dapat membuat instans DB Amazon RDS menjalankan MySQL dan MariaDB yang menggunakan kelas instans DB db.r4, db.m4.16x large, db.t2.xlarge, dan db.t2.2xlarge. Untuk informasi selengkapnya, lihat Kelas instans DB .	20 November 2017
SQL Server 2017	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan Microsoft SQL Server 2017. Anda juga dapat membuat instans DB yang menjalank an SQL Server 2016 SP1 CU5. Untuk informasi selengkapnya, lihat Amazon RDS for Microsoft SQL Server.	17 November 2017
Memulihkan cadangan MySQL dari Amazon S3	Anda sekarang dapat membuat cadangan basis data on-premise Anda, menyimpannya di Amazon S3, lalu memulihkan file cadangan ke instans DB Amazon RDS baru yang menjalankan MySQL. Untuk informasi selengkapnya, lihat Memulihkan cadangan ke instans DB MySQL.	17 November 2017
Penskalaan Otomatis dengan Replika Aurora	Amazon Aurora MySQL sekarang mendukung Penskalaan Otomatis Aurora. Penskalaan Otomatis Aurora secara dinamis menyesuaikan jumlah Replika Aurora berdasarkan peningkatan atau penurunan konektivitas atau beban kerja. Untuk informasi selengkapnya, lihat Menggunakan Penskalaan Otomatis Amazon Aurora dengan replika Aurora.	17 November 2017
Dukungan edisi default Oracle	Instans DB Amazon RDS for Oracle kini mendukung pengaturan edisi default untuk instans DB. Untuk informasi selengkapnya, lihat Mengatur edisi default untuk instans DB.	3 November 2017

Perubahan	Deskripsi	Tanggal diubah
Validasi file instans DB Oracle	Instans DB Amazon RDS for Oracle kini mendukung validasi file instans DB dengan utilitas validasi logis Oracle Recovery Manager (RMAN). Untuk informasi selengkapnya, lihat Memvalidasi file database dalam RDS untuk Oracle .	3 November 2017
Management Agent untuk OEM 13c	Instans DB Amazon RDS for Oracle kini mendukung Management Agent untuk Oracle Enterprise Manager (OEM) Cloud Control 13c. Untuk informasi selengkapnya, lihat Oracle Management Agent untuk Kontrol Cloud Enterprise Manager.	1 November 2017
Konfigurasi ulang penyimpanan untuk snapshot Microsoft SQL Server	Anda sekarang dapat mengonfigurasi ulang penyimpanan saat Anda memulihkan snapshot ke instans DB Amazon RDS yang menjalankan Microsoft SQL Server. Untuk informasi selengkapnya, lihat Memulihkan dari snapshot DB.	26 Oktober 2017
Prapengambilan kunci asinkron untuk Aurora Edisi Kompatibel MySQL	Prapengambilan kunci asinkron (AKP) meningkatkan performa join indeks yang tidak di-cache, dengan mengambil lebih dulu kunci dalam memori sebelum dibutuhkan. Untuk informasi selengkapnya, lihat Menggunakan prapengambilan kunci asinkron di Amazon Aurora.	26 Oktober 2017
MySQL 5.7.19, 5.6.37, dan 5.5.57	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MySQL versi 5.7.19, 5.6.37, dan 5.5.57. Untuk informasi selengkapnya, lihat Versi MySQL di Amazon RDS.	25 Oktober 2017

Perubahan	Deskripsi	Tanggal diubah
Ketersediaan umum Amazon Aurora dengan kompatibilitas PostgreSQL	Amazon Aurora dengan kompatibilitas PostgreSQL memudahkan dan menghemat biaya untuk menyiapkan, mengoperasikan, dan menskalakan deployment PostgreSQL Anda yang baru dan yang sudah ada, sehingga membebaskan Anda untuk fokus pada bisnis dan aplikasi Anda. Untuk informasi selengkapnya, lihat Menggunakan Amazon Aurora PostgreSQL.	24 Oktober 2017
Instans DB Amazon RDS for Oracle mendukung kelas instans DB yang baru	Instans DB Amazon RDS for Oracle sekarang mendukung kelas instans memori yang dioptimalkan generasi berikutnya (db.r4). Instans DB Amazon RDS for Oracle sekarang juga mendukung kelas instans generasi baru berikut ini: db.m4.16xlarge, db.t2.xlarge, dan db.t2.2xlarge. Untuk informasi selengkapnya, lihat Kelas instans DB dan Kelas instans RDS for Oracle.	23 Oktober 2017
Fitur baru	Instans Terpesan Anda yang baru dan yang sudah ada sekarang dapat mencakup beberapa ukuran dalam kelas instans DB yang sama. Instans cadangan fleksibel ukuran tersedia untuk instans DB dengan AWS Region, mesin database, dan keluarga instans yang sama, dan di seluruh konfigurasi AZ. Instans terpesan ukuran fleksibel tersedia untuk mesin basis data berikut: Amazon Aurora, MariaDB, MySQL, Oracle (Bawa Lisensi Sendiri), PostgreSQL. Untuk informasi selengkapnya, lihat Instans DB terpesan berukuran fleksibel.	11 Oktober 2017
Fitur baru	Anda sekarang dapat menggunakan opsi Oracle SQLT untuk menyetel pernyataan SQL untuk performa yang optimal. Untuk informasi selengkapnya, lihat Oracle SQLT.	22 September 2017

Perubahan	Deskripsi	Tanggal diubah
Fitur baru	Jika Anda memiliki snapshot DB manual yang ada untuk instans DB Amazon RDS for Oracle, Anda sekarang dapat meningkatkan-nya ke mesin basis data Oracle versi lebih baru. Untuk informasi selengkapnya, lihat Meng-upgrade snapshot DB Oracle.	20 September 2017
Fitur baru	Anda sekarang dapat menggunakan Oracle Spatial untuk menyimpan, mengambil, memperbarui, dan membuat kueri data spasial di instans DB Amazon RDS Anda yang menjalankan Oracle. Untuk informasi selengkapnya, lihat Oracle Spatial.	15 September 2017
Fitur baru	Anda sekarang dapat menggunakan Oracle Locator untuk mendukung aplikasi berbasis internet dan layanan nirkabel serta solusi GIS berbasis mitra dengan instans DB Amazon RDS yang menjalankan Oracle. Untuk informasi selengkapnya, lihat Oracle Locator.	15 September 2017
Fitur baru	Anda sekarang dapat menggunakan Oracle Multimedi a untuk menyimpan, mengelola, dan mengambil gambar, audio, video, dan data media heterogen lainnya di instans DB Amazon RDS yang menjalank an Oracle. Untuk informasi selengkapnya, lihat Oracle Multimedia.	15 September 2017
Fitur baru	Anda sekarang dapat mengekspor log audit dari cluster Amazon Aurora MySQL DB Anda ke Amazon Logs. CloudWatch Untuk informasi selengkapnya, lihat Menerbitkan log Aurora MySQL ke Amazon Logs. CloudWatch	14 September 2017

Perubahan	Deskripsi	Tanggal diubah
Fitur baru	Amazon RDS sekarang mendukung beberapa versi Oracle Application Express (APEX) untuk instans DB Anda yang menjalankan Oracle. Untuk informasi selengkapnya, lihat Oracle Application Express (APEX).	13 September 2017
Fitur baru	Anda sekarang dapat menggunakan Amazon Aurora untuk memigrasikan snapshot DB atau instans DB MySQL yang terenkripsi atau tidak terenkripsi ke klaster DB Aurora MySQL yang terenkripsi. Untuk informasi selengkapnya, lihat Memigrasikan snapshot RDS for MySQL dan Memigrasikan data dari instans DB MySQL ke klaster DB Amazon Aurora MySQL dengan menggunakan replika baca Aurora.	5 September 2017
Fitur baru	Anda dapat menggunakan basis data Amazon RDS for Microsoft SQL Server untuk membangun aplikasi yang mematuhi HIPAA. Untuk informasi selengkapnya, lihat Dukungan program kepatuhan untuk instans DB Microsoft SQL Server.	31 Agustus 2017
Fitur baru	Anda sekarang dapat menggunakan basis data Amazon RDS for MariaDB untuk membangun aplikasi yang mematuhi HIPAA. Untuk informasi selengkapnya, lihat Amazon RDS for MariaDB.	31 Agustus 2017
Fitur baru	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan Microsoft SQL Server dengan penyimpanan yang dialokasikan hingga 16 TiB, dan IOPS yang tersedia untuk rentang penyimpanan 1:1–50:1. Untuk informasi selengkapnya, lihat Penyimpanan instans DB Amazon RDS.	22 Agustus 2017

Perubahan	Deskripsi	Tanggal diubah
Fitur baru	Anda sekarang dapat menggunakan deployment Multi-AZ untuk instans DB yang menjalankan Microsoft SQL Server di wilayah UE (Frankfurt). Untuk informasi selengkapnya, lihat Deployment Multi-AZ untuk Amazon RDS for Microsoft SQL Server.	3 Agustus 2017
Fitur baru	Anda sekarang dapat membuat instans DB Amazon RDS yang menjalankan MariaDB versi 10.1.23 dan 10.0.31. Untuk informasi selengkapnya, lihat Versi-versi MariaDB pada Amazon RDS.	17 Juli 2017
Fitur baru	Amazon RDS sekarang mendukung Microsoft SQL Server Enterprise Edition dengan model Lisensi Termasuk di semua AWS Wilayah. Untuk informasi selengkapnya, lihat Melisensikan Microsoft SQL Server di Amazon RDS.	13 Juli 2017
Fitur baru	Amazon RDS for Oracle sekarang mendukung halaman besar kernel Linux untuk meningkatkan skalabilitas basis data. Penggunaan halaman yang besar menghasilkan tabel halaman yang lebih kecil dan lebih sedikit waktu CPU yang dihabiskan untuk manajemen memori, sehingga meningkatkan performa instans basis data yang besar. Anda dapat menggunakan halaman besar dengan instans DB Amazon RDS yang menjalankan semua edisi Oracle versi 12.1.0.2 dan 11.2.0.4. Untuk informasi selengkapnya, lihat Mengaktifkan HugePages untuk instans RDS for Oracle.	7 Juli 2017
Fitur baru	Diperbarui untuk mendukung enkripsi saat diam (EAR) untuk kelas instans DB db.t2.small dan db.t2.medium untuk semua mesin DB non-Aurora. Untuk informasi selengkapnya, lihat Ketersediaan enkripsi Amazon RDS.	27 Juni 2017

Perubahan	Deskripsi	Tanggal diubah
Fitur baru	Diperbarui untuk mendukung Amazon Aurora di wilayah Eropa (Frankfurt). Untuk informasi selengkap nya, lihat Ketersediaan untuk Amazon Aurora MySQL.	16 Juni 2017
Fitur baru	Sekarang Anda dapat menentukan grup opsi saat Anda menyalin snapshot DB di seluruh AWS wilayah. Untuk informasi selengkapnya, lihat Pertimbangan grup opsi.	12 Juni 2017
Fitur baru	Anda sekarang dapat menyalin snapshot DB yang dibuat dari instans DB khusus di seluruh wilayah AWS . Anda dapat menyalin snapshot dari instans DB yang menggunakan TDE Oracle, TDE Microsoft SQL Server, dan Multi-AZ Microsoft SQL Server dengan Pencerminan. Untuk informasi selengkapnya, lihat Menyalin snapshot DB.	12 Juni 2017
Fitur baru	Amazon Aurora sekarang memungkinkan Anda dengan cepat dan hemat biaya menyalin semua basis data Anda di klaster DB Amazon Aurora. Untuk informasi selengkapnya, lihat Mengkloning basis data di klaster DB Aurora.	12 Juni 2017
Fitur baru	Amazon RDS sekarang mendukung Microsoft SQL Server 2016 SP1 CU2. Untuk informasi selengkapnya, lihat Amazon RDS for Microsoft SQL Server.	7 Juni 2017
Pratinjau	Pratinjau publik Amazon Aurora dengan kompatibi litas PostgreSQL. Untuk informasi selengkapnya, lihat Menggunakan Amazon Aurora PostgreSQL.	19 April 2017

Perubahan	Deskripsi	Tanggal diubah
Fitur baru	Amazon Aurora sekarang memungkinkan Anda menjalankan operasi ALTER TABLE tbl_name ADD COLUMN col_name column_definition hampir secara seketika. Operasi tersebut akan diselesaikan tanpa perlu menyalin tabel dan tanpa memengaruhi pernyataan DML lainnya secara signifikan. Untuk informasi selengkapnya, lihat Mengubah tabel di Amazon Aurora menggunakan DDL cepat.	5 April 2017
Fitur baru	Kami telah menambahkan perintah pemantauan baru, SHOW VOLUME STATUS, untuk menampilkan jumlah simpul dan disk dalam satu volume. Untuk informasi selengkapnya, lihat Menampilkan status volume untuk klaster DB Aurora.	5 April 2017
Fitur baru	Anda sekarang dapat menggunakan logika kustom Anda sendiri dalam fungsi verifikasi kata sandi kustom untuk Oracle on Amazon RDS. Untuk informasi selengkapnya, lihat Membuat fungsi kustom untuk memverifikasi kata sandi.	21 Maret 2017
Fitur baru	Anda sekarang dapat mengakses file log pengulangan online dan yang diarsipkan di instans DB Oracle Anda di Amazon RDS. Untuk informasi selengkapnya, lihat Mengakses log pengulangan online dan yang diarsipkan.	21 Maret 2017
Fitur baru	Anda sekarang dapat menyalin snapshot klaster DB terenkripsi dan tidak terenkripsi antar-akun di wilayah yang sama. Untuk informasi selengkapnya, lihat Menyalin snapshot klaster DB antar-akun.	7 Maret 2017
Fitur baru	Anda sekarang dapat berbagi snapshot klaster DB terenkripsi antar-akun di wilayah yang sama. Untuk informasi selengkapnya, lihat Berbagi snapshot klaster DB.	7 Maret 2017

Perubahan	Deskripsi	Tanggal diubah
Fitur baru	Anda sekarang dapat mereplikasi klaster DB Amazon Aurora MySQL yang dienkripsi untuk membuat Replika Aurora lintas wilayah. Untuk informasi selengkapnya, lihat Mereplikasi kluster DB MySQL Aurora di seluruh Wilayah. AWS	7 Maret 2017
Fitur baru	Anda sekarang dapat meminta agar semua koneksi ke instans DB Anda yang menjalankan Microsoft SQL Server menggunakan Lapisan Soket Aman (SSL). Untuk informasi selengkapnya, lihat Menggunakan SSL dengan instans DB Microsoft SQL Server.	27 Februari 2017
Fitur baru	Anda sekarang dapat menetapkan zona waktu lokal Anda ke salah satu dari 15 zona waktu tambahan. Untuk informasi selengkapnya, lihat Zona waktu yang didukung.	27 Februari 2017
Fitur baru	Anda sekarang dapat menggunakan prosedur Amazon RDS `msdb.dbo.rds_shrink_tempdbfile` untuk memperkecil basis data tempdb pada instans DB Anda yang menjalankan Microsoft SQL Server. Untuk informasi selengkapnya, lihat Mengurangi basis data tempdb.	17 Februari 2017
Fitur baru	Anda sekarang dapat mengompresi file cadangan Anda saat mengekspor basis data Microsoft SQL Server Enterprise Edition dan Standard Edition dari instans DB Amazon RDS ke Amazon S3. Untuk informasi selengkapnya, lihat Mengompresi file backup.	17 Februari 2017
Fitur baru	Amazon RDS sekarang mendukung server DNS kustom untuk me-resolve nama DNS yang digunakan dalam akses jaringan keluar pada instans DB Anda yang menjalankan Oracle. Untuk informasi selengkap nya, lihat Menyiapkan server DNS kustom.	26 Januari 2017

Perubahan	Deskripsi	Tanggal diubah
Fitur baru	Amazon RDS sekarang mendukung pembuatan replika baca terenkripsi di wilayah lain. Untuk informasi lebih lanjut, lihat Membuat replika baca di tempat yang berbeda Wilayah AWS dan createdB InstanceReadReplica.	23 Januari 2017
Fitur baru	Amazon RDS sekarang mendukung peningkatan snapshot DB MySQL dari MySQL 5.1 ke MySQL 5.5.	20 Januari 2017
Fitur baru	Amazon RDS sekarang mendukung penyalinan snapshot DB terenkripsi ke wilayah lain untuk mesin basis data MariaDB, MySQL, Oracle, PostgreSQL, dan Microsoft SQL Server. Untuk informasi selengkapnya, lihat Menyalin snapshot DB dan CopyDBSnapshot.	20 Desember 2016
Fitur baru	Amazon Aurora MySQL sekarang mendukung pengindeksan spasial. Pengindeksan spasial meningkatkan performa kueri pada set data besar untuk kueri yang menggunakan data spasial. Untuk informasi selengkapnya, lihat Amazon Aurora MySQL dan data spasial.	14 Desember 2016
Fitur baru	Amazon RDS sekarang mendukung akses jaringan keluar pada instans DB Anda yang menjalankan Oracle. Anda dapat menggunakan utl_http, utl_tcp, dan utl_smtp untuk menghubungkan dari instans DB Anda ke jaringan. Untuk informasi selengkapnya, lihat Mengonfigurasi akses UTL_HTTP menggunakan sertifikat dan dompet Oracle.	5 Desember 2016
Fitur baru	Amazon RDS telah menghentikan dukungan untuk MySQL versi 5.1. Namun, Anda dapat memulihkan snapshot MySQL 5.1 yang ada ke instans MySQL 5.5. Untuk informasi selengkapnya, lihat Mesin penyimpanan yang didukung untuk RDS for MySQL.	15 November 2016

Perubahan	Deskripsi	Tanggal diubah
Fitur baru	Amazon RDS sekarang mendukung Microsoft SQL Server 2016 RTM CU2. Untuk informasi selengkapnya, lihat Amazon RDS for Microsoft SQL Server.	4 November 2016
Fitur baru	Amazon RDS sekarang mendukung peningkatan versi mayor untuk instans DB yang menjalankan Oracle. Anda sekarang dapat meningkatkan instans DB Oracle Anda dari 11g ke 12c. Untuk informasi selengkapnya, lihat Meng-upgrade mesin DB Oracle.	2 November 2016
Fitur baru	Anda sekarang dapat membuat instans DB yang menjalankan Microsoft SQL Server 2014 Enterpris e Edition. Amazon RDS sekarang mendukung SQL Server 2014 SP2 untuk semua edisi dan semua wilayah. Untuk informasi selengkapnya, lihat Amazon RDS for Microsoft SQL Server.	25 Oktober 2016
Fitur baru	Amazon Aurora MySQL sekarang terintegrasi dengan AWS layanan lain: Anda dapat memuat teks atau data XHTML ke dalam tabel dari bucket Amazon S3, atau menjalankan fungsi dari kode database. AWS Lambda Untuk informasi selengkapnya, lihat Mengintegrasikan Aurora MySQL dengan layanan lain. AWS	18 Oktober 2016
Fitur baru	Anda sekarang dapat mengakses basis data tempdb di instans DB Amazon RDS Anda yang menjalankan Microsoft SQL Server. Anda dapat mengakses basis data tempdb dengan menggunakan Transact-SQL melalui Microsoft SQL Server Management Studio (SSMS), atau aplikasi klien SQL standar lainnya. Untuk informasi selengkapnya, lihat Mengakses basis data tempdb pada instans DB Microsoft SQL Server di Amazon RDS.	29 September 2016

Perubahan	Deskripsi	Tanggal diubah
Fitur baru	Anda sekarang dapat menggunakan paket UTL_MAIL dengan instans DB Amazon RDS Anda yang menjalankan Oracle. Untuk informasi selengkapnya, lihat Oracle UTL_MAIL.	20 September 2016
Fitur baru	Anda sekarang dapat mengatur zona waktu instans DB Microsoft SQL Server baru Anda ke zona waktu lokal, agar sesuai dengan zona waktu aplikasi Anda. Untuk informasi selengkapnya, lihat Zona waktu lokal untuk instans DB Microsoft SQL Server.	19 September 2016
Fitur baru	Anda sekarang dapat menggunakan opsi Oracle Label Security untuk mengontrol akses ke baris tabel individual di instans DB Amazon RDS Anda yang menjalankan Oracle Database 12c. Dengan Oracle Label Security, Anda dapat memberlakukan kepatuhan peraturan dengan model administrasi berbasis kebijakan, dan memastikan bahwa akses ke data sensitif dibatasi hanya untuk pengguna dengan tingkat izin yang sesuai. Untuk informasi selengkap nya, lihat Keamanan Label Oracle.	8 September 2016
Fitur baru	Anda sekarang dapat terhubung ke klaster DB Amazon Aurora menggunakan titik akhir pembaca, yang menyeimbangkan beban koneksi di seluruh Replika Aurora yang tersedia di klaster DB. Saat klien meminta koneksi baru ke titik akhir pembaca, Aurora mendistribusikan permintaan koneksi di antara Replika Aurora di klaster DB. Fungsionalitas ini dapat membantu menyeimbangkan beban kerja baca Anda ke beberapa Replika Aurora di klaster DB Anda. Untuk informasi selengkapnya, lihat Titik akhir Amazon Aurora.	8 September 2016

Perubahan	Deskripsi	Tanggal diubah
Fitur baru	Anda sekarang dapat mendukung Oracle Enterprise Manager Cloud Control pada instans DB Amazon RDS Anda yang menjalankan Oracle. Anda dapat mengaktifkan Agen Manajemen pada instans DB Anda, dan berbagi data dengan Oracle Management Service (OMS) Anda. Untuk informasi selengkapnya, lihat Oracle Management Agent untuk Kontrol Cloud Enterprise Manager.	1 September 2016
Fitur baru	Rilis ini menambahkan dukungan untuk mendapatkan ARN untuk sumber daya. Untuk informasi selengkapnya, lihat Mendapatkan ARN yang sudah ada.	23 Agustus 2016
Fitur baru	Anda sekarang dapat menetapkan hingga 50 tag untuk setiap sumber daya Amazon RDS, untuk mengelola sumber daya dan melacak biaya Anda. Untuk informasi selengkapnya, lihat Memberi tag pada sumber daya Amazon RDS.	19 Agustus 2016
Fitur baru	Amazon RDS sekarang mendukung model Termasuk Lisensi untuk Oracle Standard Edition Two. Untuk informasi selengkapnya, lihat Membuat instans DB Amazon RDS. Anda sekarang dapat mengubah model lisensi instans DB Amazon RDS Anda yang menjalankan Microsoft SQL Server dan Oracle. Untuk informasi lebih lanjut, lihat Melisensikan Microsoft SQL Server di Amazon RDS dan Opsi lisensi RDS for Oracle.	5 Agustus 2016

Perubahan	Deskripsi	Tanggal diubah
Fitur baru	Amazon RDS sekarang mendukung pencadangan dan pemulihan native untuk basis data Microsoft SQL Server yang menggunakan file cadangan penuh (file .bak). Anda sekarang dapat dengan mudah memigrasikan database SQL Server ke Amazon RDS, dan mengimpor dan mengekspor database dalam satu file yang mudah diportabel, menggunakan Amazon S3 untuk penyimpanan, dan untuk enkripsi. AWS KMS Untuk informasi selengkapnya, lihat Mengimpor dan mengekspor basis data SQL Server menggunakan pencadangan dan pemulihan native.	27 Juli 2016
Fitur baru	Anda sekarang dapat menyalin file sumber dari basis data MySQL ke bucket Amazon Simple Storage Service (Amazon S3), lalu memulihkan klaster DB Amazon Aurora dari file tersebut. Opsi ini dapat jauh lebih cepat dibandingkan memigrasikan data menggunakan `mysqldump` . Untuk informasi selengkapnya, lihat Memigrasikan data dari basis data MySQL eksternal ke klaster DB Aurora MySQL.	20 Juli 2016
Fitur baru	Anda sekarang dapat memulihkan snapshot kluster Amazon Aurora DB yang tidak terenkripsi untuk membuat kluster Amazon Aurora DB terenkripsi dengan menyertakan kunci enkripsi () selama operasi pemulihan. AWS Key Management Service AWS KMS Untuk informasi selengkapnya, lihat Mengenkripsi sumber daya Amazon RDS.	30 Juni 2016
Fitur baru	Anda dapat menggunakan Oracle Repository Creation Utility (RCU) untuk membuat repositori di Amazon RDS for Oracle. Untuk informasi selengkapnya, lihat Menggunakan Oracle Repository Creation Utility pada RDS for Oracle .	17 Juni 2016

Perubahan	Deskripsi	Tanggal diubah
Fitur baru	Menambahkan dukungan untuk replika baca lintas wilayah PostgreSQL. Untuk informasi selengkapnya, lihat Membuat replika baca di tempat yang berbeda Wilayah AWS.	16 Juni 2016
Fitur baru	Anda sekarang dapat menggunakan AWS Management Console untuk dengan mudah menambahkan Multi-AZ dengan Mirroring ke instans Microsoft SQL Server DB. Untuk informasi selengkap nya, lihat Menambahkan Multi-AZ ke instans DB Microsoft SQL Server.	9 Juni 2016
Fitur baru	Anda sekarang dapat menggunakan Deployment Multi-AZ menggunakan Pencerminan SQL Server di wilayah tambahan berikut: Asia Pasifik (Sydney), Asia Pasifik (Tokyo), dan Amerika Selatan (Sao Paulo). Untuk informasi selengkapnya, lihat Deployment Multi-AZ untuk Amazon RDS for Microsoft SQL Server.	9 Juni 2016
Fitur baru	Diperbarui untuk mendukung MariaDB versi 10.1. Untuk informasi selengkapnya, lihat Amazon RDS for MariaDB.	1 Juni 2016
Fitur baru	Diperbarui untuk mendukung klaster DB lintas wilayah Amazon Aurora yang merupakan replika baca. Untuk informasi selengkapnya, lihat Mereplikasi klaster DB Aurora MySQL antar-Wilayah AWS.	1 Juni 2016
Fitur baru	Pemantauan yang Ditingkatkan sekarang tersedia untuk instans DB Oracle. Untuk informasi lebih lanjut, lihat Memantau metrik OS dengan Pemantauan yang Disempurnakan dan Memodifikasi instans DB Amazon RDS.	27 Mei 2016

Perubahan	Deskripsi	Tanggal diubah
Fitur baru	Diperbarui untuk mendukung kemampuan berbagi snapshot manual untuk snapshot klaster DB Amazon Aurora. Untuk informasi selengkapnya, lihat Berbagi snapshot klaster DB.	18 Mei 2016
Fitur baru	Anda sekarang dapat menggunakan MariaDB Audit Plugin untuk mencatat log aktivitas basis data di instans basis data MariaDB dan MySQL. Untuk informasi lebih lanjut, lihat Opsi untuk mesin basis data MariaDB dan Opsi untuk instans DB MySQL.	27 April 2016
Fitur baru	Peningkatan versi mayor di tempat sekarang tersedia untuk peningkatan dari MySQL versi 5.6 ke versi 5.7. Untuk informasi selengkapnya, lihat Meng-upgrade mesin DB MySQL.	26 April 2016
Fitur baru	Pemantauan yang Ditingkatkan sekarang tersedia untuk instans DB Microsoft SQL Server. Untuk informasi selengkapnya, lihat Memantau metrik OS dengan Pemantauan yang Disempurnakan.	22 April 2016
Fitur baru	Diperbarui untuk memberikan tampilan Klaster Amazon Aurora di konsol Amazon RDS. Untuk informasi selengkapnya, lihat Melihat klaster DB Aurora.	1 April 2016
Fitur baru	Diperbarui untuk mendukung Multi-AZ SQL Server dengan pencerminan di wilayah Asia Pasifik (Seoul). Untuk informasi selengkapnya, lihat Deployment Multi-AZ untuk Amazon RDS for Microsoft SQL Server.	31 Maret 2016
Fitur baru	Diperbarui untuk mendukung Multi-AZ Amazon Aurora dengan pencerminan di wilayah Asia Pasifik (Seoul). Untuk informasi selengkapnya, lihat Ketersediaan untuk Amazon Aurora MySQL.	31 Maret 2016

Perubahan	Deskripsi	Tanggal diubah
Fitur baru	Instans DB PostgreSQL memiliki kemampuan untuk mewajibkan koneksi agar menggunakan SSL. Untuk informasi selengkapnya, lihat Menggunakan SSL dengan instans DB PostgreSQL.	25 Maret 2016
Fitur baru	Pemantauan yang Ditingkatkan sekarang tersedia untuk instans DB PostgreSQL. Untuk informasi selengkapnya, lihat Memantau metrik OS dengan Pemantauan yang Disempurnakan.	25 Maret 2016
Fitur baru	Instans DB Microsoft SQL Server sekarang dapat menggunakan Autentikasi Windows untuk autentika si pengguna. Untuk informasi selengkapnya, lihat Menggunakan AWS Managed Active Directory dengan RDS for SQL Server.	23 Maret 2016
Fitur baru	Pemantauan yang Ditingkatkan sekarang tersedia di wilayah Asia Pasifik (Seoul). Untuk informasi selengkapnya, lihat Memantau metrik OS dengan Pemantauan yang Disempurnakan.	16 Maret 2016
Fitur baru	Anda sekarang dapat menyesuaikan urutan promosi Replika Aurora ke instans primer selama failover. Untuk informasi selengkapnya, lihat Toleransi kesalahan untuk klaster DB Aurora.	14 Maret 2016
Fitur baru	Diperbarui untuk mendukung enkripsi saat bermigras i ke klaster DB Aurora. Untuk informasi selengkapnya, lihat Memigrasikan data ke klaster DB Aurora.	2 Maret 2016
Fitur baru	Diperbarui untuk mendukung zona waktu lokal untuk klaster Aurora DB. Untuk informasi selengkapnya, lihat Zona waktu lokal untuk klaster Aurora DB.	1 Maret 2016

Perubahan	Deskripsi	Tanggal diubah
Fitur baru	Diperbarui untuk menambahkan dukungan untuk MySQL versi 5.7 untuk kelas instans DB Amazon RDS generasi saat ini.	22 Februari 2016
Fitur baru	Diperbarui untuk mendukung kelas instans db.r3 dan db.t2 DB di wilayah (AS-Barat). AWS GovCloud	11 Februari 2016
Fitur baru	Diperbarui untuk mendukung salinan enkripsi snapshot DB dan berbagi snapshot DB terenkripsi. Untuk informasi lebih lanjut, lihat Menyalin snapshot DB dan Berbagi snapshot DB.	11 Februari 2016
Fitur baru	Diperbarui untuk mendukung Amazon Aurora di wilayah Asia Pasifik (Sydney). Untuk informasi selengkapnya, lihat Ketersediaan untuk Amazon Aurora MySQL.	11 Februari 2016
Fitur baru	Diperbarui untuk mendukung SSL untuk Instans DB Oracle. Untuk informasi selengkapnya, lihat Menggunakan SSL dengan instans DB RDS for Oracle.	9 Februari 2016
Fitur baru	Diperbarui untuk mendukung zona waktu lokal untuk instans DB MySQL dan MariaDB. Untuk informasi lebih lanjut, lihat Zona waktu lokal untuk instans DB MySQL dan Zona waktu lokal untuk instans basis data MariaDB.	21 Desember 2015
Fitur baru	Diperbarui untuk mendukung Pemantauan yang Ditingkatkan terhadap metrik OS untuk instans MySQL dan MariaDB serta klaster DB Aurora. Untuk informasi selengkapnya, lihat Melihat metrik di konsol Amazon RDS.	18 Desember 2015

Perubahan	Deskripsi	Tanggal diubah
Fitur baru	Diperbarui untuk mendukung kelas instans DB db.t2, db.r3, dan db.m4 untuk MySQL versi 5.5. Untuk informasi selengkapnya, lihat Kelas instans DB .	4 Desember 2015
Fitur baru	Diperbarui untuk mendukung modifikasi port basis data untuk instans DB yang ada.	3 Desember 2015
Fitur baru	Diperbarui untuk mendukung peningkatan versi mayor mesin basis data untuk instans PostgreSQL. Untuk informasi selengkapnya, lihat Meningkatkan mesin DB PostgreSQL untuk Amazon RDS.	19 November 2015
Fitur baru	Diperbarui untuk mendukung modifikasi aksesibil itas publik instans DB yang ada. Diperbarui untuk mendukung kelas instans DB standar db.m4.	11 November 2015
Fitur baru	Diperbarui untuk mendukung berbagi snapshot DB manual. Untuk informasi selengkapnya, lihat Berbagi snapshot DB.	28 Oktober 2015
Fitur baru	Diperbarui untuk mendukung Microsoft SQL Server 2014 untuk edisi Web, Express, dan Standard.	26 Oktober 2015
Fitur baru	Diperbarui untuk mendukung mesin basis data MariaDB berbasis MySQL. Untuk informasi selengkap nya, lihat Amazon RDS for MariaDB.	7 Oktober 2015
Fitur baru	Diperbarui untuk mendukung Amazon Aurora di wilayah Asia Pasifik (Tokyo). Untuk informasi selengkapnya, lihat Ketersediaan untuk Amazon Aurora MySQL.	7 Oktober 2015
Fitur baru	Diperbarui untuk mendukung kelas instans DB berkemampuan burst db.t2 untuk semua mesin DB dan penambahan kelas instans DB db.t2.large. Untuk informasi selengkapnya, lihat Kelas instans DB .	25 September 2015

Perubahan	Deskripsi	Tanggal diubah
Fitur baru	Diperbarui untuk mendukung instans Oracle DB pada kelas instans DB R3 dan T2. Untuk informasi selengkapnya, lihat Kelas instans DB .	5 Agustus 2015
Fitur baru	Microsoft SQL Server Enterprise Edition sekarang tersedia dengan model layanan Termasuk Lisensi. Untuk informasi selengkapnya, lihat Melisensikan Microsoft SQL Server di Amazon RDS.	29 Juli 2015
Fitur baru	Amazon Aurora telah resmi dirilis. Mesin DB Amazon Aurora mendukung beberapa instans DB dalam klaster DB. Untuk informasi terperinci, lihat Apa itu Amazon Aurora?.	27 Juli 2015
Fitur baru	Diperbarui untuk mendukung penyalinan tag ke snapshot DB.	20 Juli 2015
Fitur baru	Diperbarui untuk mendukung peningkatan ukuran penyimpanan untuk semua mesin DB dan peningkatan IOPS yang Tersedia untuk SQL Server.	18 Juni 2015
Fitur baru	Opsi yang diperbarui untuk instans DB terpesan.	15 Juni 2015
Fitur baru	Diperbarui untuk mendukung penggunaan Amazon CloudHSM dengan instans DB Oracle yang menggunakan TDE.	8 Januari 2015
Fitur baru	Diperbarui untuk mendukung enkripsi data diam dan API versi baru 2014-10-31.	6 Januari 2015
Fitur baru	Diperbarui untuk menyertakan mesin DB Amazon baru: Aurora. Mesin DB Amazon Aurora mendukung beberapa instans DB dalam klaster DB. Amazon Aurora saat ini dalam rilis pratinjau dan dapat berubah. Untuk informasi terperinci, lihat Apa itu Amazon Aurora?.	12 November 2014

Perubahan	Deskripsi	Tanggal diubah
Fitur baru	Diperbarui untuk mendukung replika baca PostgreSQL.	10 November 2014
API dan fitur baru	Diperbarui untuk mendukung jenis penyimpanan GP2 dan versi API baru 2014-09-01. Diperbarui untuk mendukung kemampuan menyalin opsi atau grup parameter yang ada untuk membuat opsi atau grup parameter baru.	7 Oktober 2014
Fitur baru	Diperbarui untuk mendukung Pemanasan Cache InnoDB untuk instans DB yang menjalankan MySQL versi 5.6.19 dan yang lebih baru.	3 September 2014
Fitur baru	Diperbarui untuk mendukung verifikasi sertifikat SSL saat menghubungkan ke mesin basis data MySQL versi 5.6, SQL Server, dan PostgreSQL.	5 Agustus 2014
Fitur baru	Diperbarui untuk mendukung kelas instans DB db.t2 yang dapat melonjak.	4 Agustus 2014
Fitur baru	Diperbarui untuk mendukung kelas instans DB db.r3 dengan memori yang dioptimalkan untuk digunakan dengan mesin basis data MySQL (versi 5.6), SQL Server, dan PostgreSQL.	28 Mei 2014
Fitur baru	Diperbarui untuk mendukung deployment Multi-AZ SQL Server menggunakan Pencerminan SQL Server.	19 Mei 2014
Fitur baru	Diperbarui untuk mendukung peningkatan dari MySQL versi 5.5 ke versi 5.6.	23 April 2014
Fitur baru	Diperbarui untuk mendukung Oracle GoldenGate.	3 April 2014
Fitur baru	Diperbarui untuk mendukung kelas instans DB M3.	20 Februari 2014
Fitur baru	Diperbarui untuk mendukung opsi Oracle Timezone.	13 Januari 2014

Perubahan	Deskripsi	Tanggal diubah
Fitur baru	Diperbarui untuk mendukung replikasi antara instans DB MySQL di berbagai wilayah.	26 November 2013
Fitur baru	Diperbarui untuk mendukung mesin DB PostgreSQL.	14 November 2013
Fitur baru	Diperbarui untuk mendukung enkripsi data transparan (TDE) SQL Server.	7 November 2013
API baru dan fitur baru	Diperbarui untuk mendukung salinan snapshot DB lintas wilayah; versi API baru, 2013-09-09.	31 Oktober 2013
Fitur baru	Diperbarui untuk mendukung Oracle Statspack.	26 September 2013
Fitur baru	Diperbarui untuk mendukung penggunaan replikasi untuk mengimpor atau mengekspor data antar-instans MySQL yang berjalan di Amazon RDS dan instans MySQL yang berjalan on-premise atau di Amazon EC2.	5 September 2013
Fitur baru	Diperbarui untuk mendukung kelas instans DB db.cr1.8xlarge untuk MySQL 5.6.	4 September 2013
Fitur baru	Diperbarui untuk mendukung replikasi replika baca.	28 Agustus 2013
Fitur baru	Diperbarui untuk mendukung pembuatan replika baca paralel.	22 Juli 2013
Fitur baru	Diperbarui untuk mendukung izin dan pemberian tag yang sangat ketat untuk semua sumber daya Amazon RDS.	8 Juli 2013
Fitur baru	Diperbarui untuk mendukung MySQL 5.6 untuk instans baru, termasuk dukungan untuk antarmuka memcached dan akses log biner MySQL 5.6.	1 Juli 2013

Perubahan	Deskripsi	Tanggal diubah
Fitur baru	Diperbarui untuk mendukung peningkatan versi mayor dari MySQL 5.1 ke MySQL 5.5.	20 Juni 2013
Fitur baru	Memperbarui grup parameter DB agar memungkinkan ekspresi untuk nilai parameter.	20 Juni 2013
API baru dan fitur baru	Diperbarui untuk mendukung status replika baca; versi API baru, 2013-05-15.	23 Mei 2013
Fitur baru	Diperbarui untuk mendukung fitur Oracle Advanced Security untuk enkripsi jaringan native dan Oracle Transparent Data Encryption.	18 April 2013
Fitur baru	Diperbarui untuk mendukung peningkatan versi mayor untuk SQL Server dan fungsionalitas tambahan untuk IOPS yang Tersedia.	13 Maret 2013
Fitur baru	Diperbarui untuk mendukung VPC Secara Default untuk RDS.	11 Maret 2013
API dan fitur baru	Diperbarui untuk mendukung akses log; versi API baru 2013-02-12	4 Maret 2013
Fitur baru	Diperbarui untuk mendukung langganan notifikasi peristiwa RDS.	4 Februari 2013
API dan fitur baru	Diperbarui untuk mendukung penggantian nama instans DB dan migrasi anggota grup keamanan DB di VPC ke grup keamanan VPC.	14 Januari 2013
Fitur baru	Diperbarui untuk dukungan AWS GovCloud (AS-Barat).	17 Desember 2012
Fitur baru	Diperbarui untuk mendukung kelas instans DB m1.medium dan m1.xlarge.	6 November 2012
Fitur baru	Diperbarui untuk mendukung promosi replika baca.	11 Oktober 2012

Perubahan	Deskripsi	Tanggal diubah
Fitur baru	Diperbarui untuk mendukung SSL di Instans DB Microsoft SQL Server.	10 Oktober 2012
Fitur baru	Diperbarui untuk mendukung Instans DB mikro Oracle.	27 September 2012
Fitur baru	Diperbarui untuk mendukung SQL Server 2012.	26 September 2012
API dan fitur baru	Diperbarui untuk mendukung IOPS yang tersedia. Versi API 2012-09-17.	25 September 2012
Fitur baru	Diperbarui untuk dukungan SQL Server untuk Instans DB di VPC dan dukungan Oracle untuk Data Pump.	13 September 2012
Fitur baru	Diperbarui untuk dukungan untuk SQL Server Agent.	22 Agustus 2012
Fitur baru	Diperbarui untuk dukungan pemberian tag Instans DB.	21 Agustus 2012
Fitur baru	Diperbarui untuk dukungan untuk Oracle APEX dan XML DB, zona waktu Oracle, dan Instans DB Oracle di VPC.	16 Agustus 2012
Fitur baru	Diperbarui untuk dukungan untuk SQL Server Database Engine Tuning Advisor dan instans DB Oracle di VPC.	18 Juli 2012
Fitur baru	Diperbarui untuk dukungan untuk grup opsi dan opsi pertama, Oracle Enterprise Manager Database Control.	29 Mei 2012
Fitur baru	Diperbarui untuk dukungan replika baca di Amazon Virtual Private Cloud.	17 Mei 2012
Fitur baru	Diperbarui untuk dukungan Microsoft SQL Server.	8 Mei 2012

Perubahan	Deskripsi	Tanggal diubah
Fitur baru	Diperbarui untuk dukungan untuk failover paksa, deployment Multi-AZ instans DB Oracle, dan kumpulan karakter non-default untuk Instans DB Oracle.	2 Mei 2012
Fitur baru	Diperbarui untuk Dukungan Amazon Virtual Private Cloud (VPC).	13 Februari 2012
Konten diperbarui	Diperbarui untuk jenis Instans Terpesan baru.	19 Desember 2011
Fitur baru	Diperbarui untuk dukungan mesin Oracle.	23 Mei 2011
Konten diperbarui	Pembaruan konsol.	13 Mei 2011
Konten diperbarui	Konten yang diedit untuk mempersingkat periode pencadangan dan pemeliharaan.	28 Februari 2011
Fitur baru	Menambahkan dukungan untuk MySQL 5.5.	31 Januari 2011
Fitur baru	Menambahkan dukungan untuk replika baca.	4 Oktober 2010
Fitur baru	Ditambahkan dukungan untuk AWS Identity and Access Management (IAM).	2 September 2010
Fitur baru	Menambahkan Manajemen Versi Mesin DB.	16 Agustus 2010
Fitur baru	Menambahkan instans DB Terpesan.	16 Agustus 2010
Fitur Baru	Amazon RDS sekarang mendukung koneksi SSL ke instans DB Anda.	28 Juni 2010
Panduan Baru	Ini adalah rilis pertama dari Panduan Pengguna Amazon RDS.	7 Juni 2010

AWS Glosarium

Untuk AWS terminologi terbaru, lihat AWS glosarium di Referensi.Glosarium AWS

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