



Migrasi database MySQL atau MariaDB multi-terabyte yang besar ke AWS

AWS Panduan Preskriptif



AWS Panduan Preskriptif: Migrasi database MySQL atau MariaDB multi-terabyte yang besar ke AWS

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

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

Table of Contents

Pengantar	1
Audiens yang dituju	2
Hasil bisnis yang ditargetkan	2
Opsi migrasi	3
Percona XtraBackup	4
Keuntungan	6
Batasan	7
Praktik terbaik	7
MyDumper	8
Keuntungan	11
Batasan	11
Praktik terbaik	11
mysqldump dan mysqlpump	12
Keuntungan	15
Batasan	15
Praktik terbaik	16
Pisahkan cadangan	16
Gateway File Amazon S3	18
Keuntungan	19
Batasan	19
Praktik terbaik	20
Praktik terbaik	21
Sumber daya	23
Riwayat dokumen	25
.....	xxvi

Migrasi database MySQL atau MariaDB multi-terabyte yang besar ke AWS

Babaiah Valluru dan Ankur Bhanawat, Amazon Web Services (AWS)

November 2024 ([riwayat dokumen](#))

Banyak organisasi yang memiliki server database MySQL dan MariaDB lokal tertarik untuk memigrasikan beban kerja database mereka ke file. AWS Cloud Banyak yang memilih Amazon Relational Database Service (Amazon RDS) untuk MariaDB, Amazon RDS for MySQL, atau Amazon Aurora MySQL Compatible Edition. [Amazon RDS](#) dirancang untuk memudahkan pengaturan, pengoperasian, dan skala basis data relasional di cloud. [Amazon Aurora](#) adalah bagian dari Amazon RDS, dan menawarkan keamanan bawaan, pencadangan berkelanjutan, komputasi bebas server, hingga 15 replika baca, replikasi Multi-wilayah otomatis, dan integrasi dengan yang lain. Layanan AWS

Meskipun migrasi ke salah satu dari ini Layanan AWS dapat memberikan banyak manfaat, migrasi database adalah salah satu tugas yang paling memakan waktu dan penting yang harus dilakukan oleh administrator database. Diperlukan perencanaan dan implementasi yang tepat untuk memigrasikan database besar dan memastikan bahwa kinerja beban kerja yang dimigrasi setara atau ditingkatkan. Dalam panduan ini, database besar dapat merujuk ke database tunggal multi-terabyte atau merujuk ke banyak database besar yang menambahkan hingga beberapa terabyte data. Memilih layanan dan alat migrasi yang tepat adalah kunci keberhasilan migrasi. Ada dua pendekatan umum untuk memigrasi database: logis dan fisik. Untuk informasi selengkapnya tentang pendekatan ini, lihat dokumentasi [MySQL dan MariaDB](#).

Panduan ini membahas berbagai alat sumber terbuka atau pihak ketiga yang dapat Anda gunakan untuk memigrasikan database MySQL dan MariaDB multi-terabyte yang besar, lokal, multi-terabyte ke Amazon RDS untuk Amazon RDS for MariaDB, Amazon RDS for MySQL, atau Amazon Aurora Edisi yang kompatibel dengan MySQL. Opsi yang dibahas dalam panduan ini menggunakan pendekatan migrasi logis atau fisik, dan setiap opsi menyertakan beberapa pendekatan untuk mentransfer file cadangan database besar dari pusat data lokal ke cloud, tempat Anda dapat memulihkan database dari file cadangan.

Audiens yang dituju

Panduan ini untuk administrator database program, insinyur basis data, insinyur migrasi, manajer proyek, dan manajer operasi atau infrastruktur yang berencana untuk memigrasikan database MySQL atau MariaDB mereka ke database. AWS Cloud

Hasil bisnis yang ditargetkan

Tujuan dari panduan ini adalah untuk membantu Anda:

- Pilih pendekatan migrasi untuk database besar yang paling sesuai dengan kasus penggunaan dan lingkungan Anda.
- Hindari keterlambatan dan kerugian finansial yang dapat terjadi ketika strategi migrasi cacat.
- Pelajari tentang kelebihan dan batasan dari setiap opsi migrasi.
- Pelajari tentang berbagai pendekatan yang dapat Anda gunakan untuk mentransfer file cadangan database besar dari pusat data lokal ke pusat data lokal. AWS Cloud
- Tinjau praktik terbaik secara keseluruhan untuk memigrasi database besar dan juga meninjau praktik terbaik untuk setiap alat, yang dapat membantu Anda memigrasi database dengan lebih efisien.

Opsi migrasi untuk database MySQL dan MariaDB besar

Anda dapat memilih dari berbagai pilihan untuk bermigrasi dari database MySQL atau MariaDB lokal ke instans database Amazon Relational Database Service (Amazon RDS) atau Amazon Aurora Edition. MySQL-Compatible Memilih pendekatan dan alat migrasi yang tepat sangat penting untuk migrasi yang sukses, dan dalam panduan ini, Anda mengevaluasi opsi berdasarkan kegunaan, ukuran data, dan persyaratan waktu henti Anda.

Berikut ini adalah alat dan pendekatan migrasi umum yang tersedia untuk memigrasikan database MySQL multi-terabyte yang dikelola sendiri secara efisien ke instans database Amazon RDS, Aurora, atau Amazon Elastic Compute Cloud (Amazon EC2):

- [Percona XtraBackup](#)(Fisik)
- [MyDumper](#)(Logis)
- [mysqldump dan mysqlpump](#)(Logis)
- [Pisahkan cadangan](#)(Fisik, logis, atau keduanya)

Berikut ini adalah alat dan pendekatan migrasi umum yang tersedia untuk memigrasikan database multi-terabyte (MySQL-compatible seperti MariaDB) secara efisien ke instans database Amazon RDS, Aurora, atau Amazon EC2:

- [MyDumper](#)(Logis)
- [mysqldump dan mysqlpump](#)(Logis)
- [Pisahkan cadangan](#)(Fisik, logis, atau keduanya)

Untuk setiap alat migrasi, ada beberapa pendekatan yang dapat Anda gunakan untuk mentransfer file cadangan database besar ke file AWS Cloud. Opsi disediakan untuk setiap alat, dan Anda juga dapat menggunakan Amazon S3 File Gateway. Untuk informasi selengkapnya, lihat [Menggunakan Amazon S3 File Gateway untuk mentransfer file cadangan](#) dalam panduan ini.

Percona XtraBackup

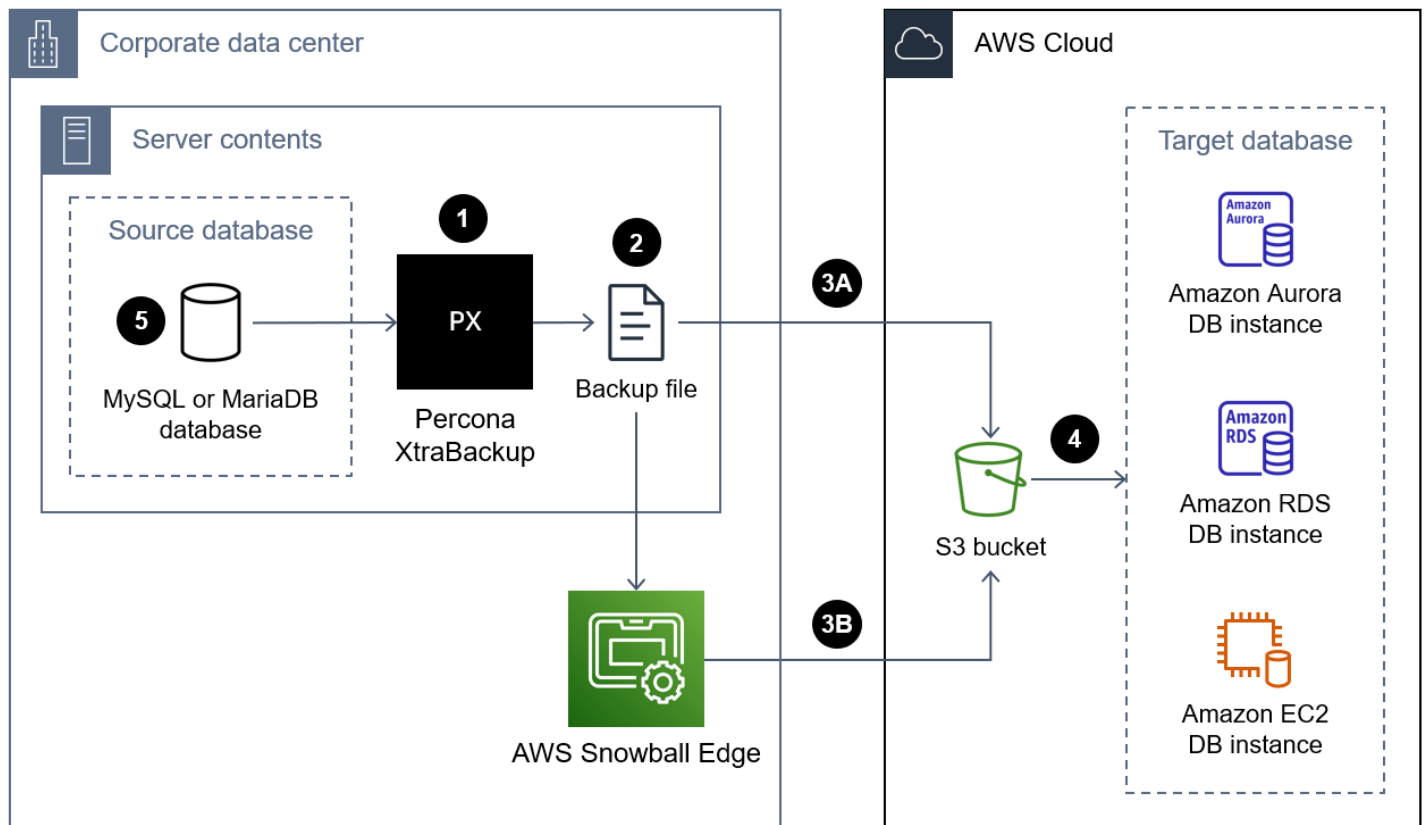
Important

Percona tidak XtraBackup didukung untuk MariaDB versi 10.3 atau yang lebih baru dan hanya didukung sebagian untuk versi 10.1 dan 10.2.

[Percona XtraBackup](#) adalah perangkat lunak cadangan hangat sumber terbuka umum untuk MySQL dan MariaDB yang membuat cadangan non-pemblokiran untuk mesin penyimpanan InnoDB dan XtraDB. Ia bekerja dengan server MySQL atau MariaDB. Untuk informasi lebih lanjut tentang alat dan beberapa fitur dan manfaatnya, lihat [Tentang Percona XtraBackup di dokumentasi Percona](#). XtraBackup

Alat ini menggunakan pendekatan migrasi fisik. Ini langsung menyalin direktori data MySQL atau MariaDB dan file di dalamnya. Untuk database besar, seperti yang lebih besar dari 100 GB, ini dapat memberikan waktu pemulihan yang jauh lebih baik daripada beberapa alat lainnya. Anda membuat cadangan database sumber lokal, memigrasikan file cadangan ke cloud, lalu memulihkan cadangan pada instans database target yang baru.

Diagram berikut menunjukkan langkah-langkah tingkat tinggi yang terlibat dalam migrasi database dengan menggunakan file cadangan XtraBackup Percona. Bergantung pada ukuran file cadangan, ada dua opsi yang tersedia untuk mentransfer cadangan ke bucket Amazon Simple Storage Service (Amazon S3) di bucket. AWS Cloud



Berikut ini adalah langkah-langkah untuk menggunakan Percona XtraBackup untuk memigrasikan database ke: AWS Cloud

1. Instal Percona XtraBackup di server lokal. [Jika Anda menggunakan Amazon Aurora MySQL versi 2 atau Amazon RDS, lihat Menginstal Percona 2.4. XtraBackup](#) Jika Anda menggunakan Amazon Aurora MySQL versi 3, lihat Menginstal Percona 8.0 di dokumentasi [XtraBackupPercona](#). XtraBackup
2. Buat cadangan lengkap dari sumber MySQL atau database MariaDB. Untuk petunjuk untuk Percona XtraBackup 2.4, lihat Cadangan [lengkap](#). Untuk petunjuk untuk Percona XtraBackup 8.0, lihat [Membuat cadangan lengkap](#).
3. Transfer file cadangan melalui internet dengan menggunakan layanan atau alat yang disetujui di organisasi Anda, seperti berikut ini:
 - [AWS Site-to-Site VPN](#)
 - [AWS Client VPN](#)
 - [AWS Direct Connect](#)

- [Amazon S3 File Gateway](#) (Untuk informasi lebih lanjut, lihat [Menggunakan Amazon S3 File Gateway untuk mentransfer file cadangan](#) di panduan ini.)
 - [AWS Command Line Interface \(AWS CLI\)](#)
4. Dari bucket Amazon S3, pulihkan file cadangan ke instance database target. Untuk petunjuk, lihat yang berikut ini:
- Untuk MySQL-Compatible Edisi Aurora, lihat [Memigrasi data dari MySQL menggunakan bucket Amazon S3 dalam dokumentasi Amazon RDS](#).
 - Untuk Amazon RDS for MySQL atau Amazon EC2, [lihat Mengimpor data ke instans MySQL DB](#).
 - Untuk Amazon RDS for MariaDB atau Amazon EC2, [lihat Mengimpor data ke instans MariaDB](#).
5. (Opsional) Anda dapat mengatur replikasi antara database sumber dan instance database target. Anda dapat menggunakan replikasi log biner (binlog) untuk mengurangi waktu henti. Untuk informasi selengkapnya, lihat berikut ini:
- [Mengatur konfigurasi sumber replikasi](#) dalam dokumentasi MySQL
 - Untuk Amazon Aurora, lihat yang berikut ini:
 - [Menyinkronkan cluster DB MySQL Amazon Aurora dengan database MySQL menggunakan replikasi dalam](#) dokumentasi Aurora
 - [Menggunakan replikasi binlog di Amazon Aurora dalam](#) dokumentasi Aurora
 - Untuk Amazon RDS, lihat yang berikut ini:
 - [Bekerja dengan replikasi MySQL](#) dalam dokumentasi Amazon RDS
 - [Bekerja dengan replikasi MariaDB](#) dalam dokumentasi Amazon RDS
 - Untuk Amazon EC2, lihat yang berikut ini:
 - [Menyiapkan Replikasi Berbasis Posisi File Log Biner](#) dalam dokumentasi MySQL
 - [Menyiapkan Replika](#) dalam dokumentasi MySQL
 - [Menyiapkan Replikasi](#) dalam dokumentasi MariaDB

Keuntungan

- Karena Percona XtraBackup menggunakan pendekatan migrasi fisik, proses pemulihan biasanya lebih cepat daripada alat yang menggunakan pendekatan migrasi logis. Ini karena kinerjanya dibatasi oleh disk atau throughput jaringan daripada sumber daya komputasi yang diperlukan untuk pemrosesan data.

- Karena proses pemulihan adalah salinan langsung file dari bucket S3 ke instance database target, file Percona biasanya memulihkan lebih cepat daripada XtraBackup file cadangan yang dibuat dengan alat lain.
- Percona mudah beradaptasi XtraBackup . Misalnya, mendukung beberapa utas untuk membantu Anda menyalin file lebih cepat dan mendukung kompresi untuk mengurangi ukuran cadangan.

Batasan

- Pencadangan offline tidak dimungkinkan karena Percona XtraBackup harus memiliki akses ke server database sumber.
- Percona hanya XtraBackup dapat digunakan pada sistem dengan arsitektur sistem yang identik. Misalnya, tidak mungkin mengembalikan cadangan database sumber yang berjalan di Intel untuk Windows Server ke server target ARM untuk Linux.
- Percona XtraBackup tidak didukung untuk MariaDB versi 10.3 atau yang lebih baru, dan hanya didukung sebagian untuk MariaDB versi 10.2 dan versi 10.1. Untuk informasi lebih lanjut, lihat [XtraBackup Ikhtisar Percona: Kompatibilitas dengan MariaDB di basis pengetahuan MariaDB](#).
- Anda tidak dapat menggunakan Percona XtraBackup untuk memulihkan database MariaDB sumber ke instance database MySQL target, seperti Amazon RDS for MySQL atau Aurora. MySQL-Compatible
- Total volume data dan jumlah objek yang dapat Anda simpan dalam bucket S3 tidak terbatas, namun ukuran file maksimum adalah 5 TB. Jika file cadangan Anda melebihi 5 TB, Anda dapat membaginya menjadi beberapa file yang lebih kecil.
- Saat `innodb_file_per_table` pengaturan dimatikan, Percona XtraBackup tidak mendukung cadangan sebagian yang menggunakan `--tables,,, --tables-exclude--tables-file, --databases` atau `--databases-exclude --databases-file` Untuk informasi selengkapnya tentang Percona XtraBackup versi 2.4, lihat Pencadangan [sebagian](#). Untuk informasi selengkapnya tentang Percona XtraBackup versi 8.0, lihat [Membuat cadangan sebagian](#).

Praktik terbaik

- Untuk meningkatkan kinerja proses pencadangan, lakukan hal berikut:
 - Salin beberapa file secara paralel dengan menggunakan `--parallel= <threads>`
 - Kompres beberapa file secara paralel dengan menggunakan `--compress-threads= <threads>`
 - Meningkatkan memori dengan menggunakan `--use-memory= <size>`

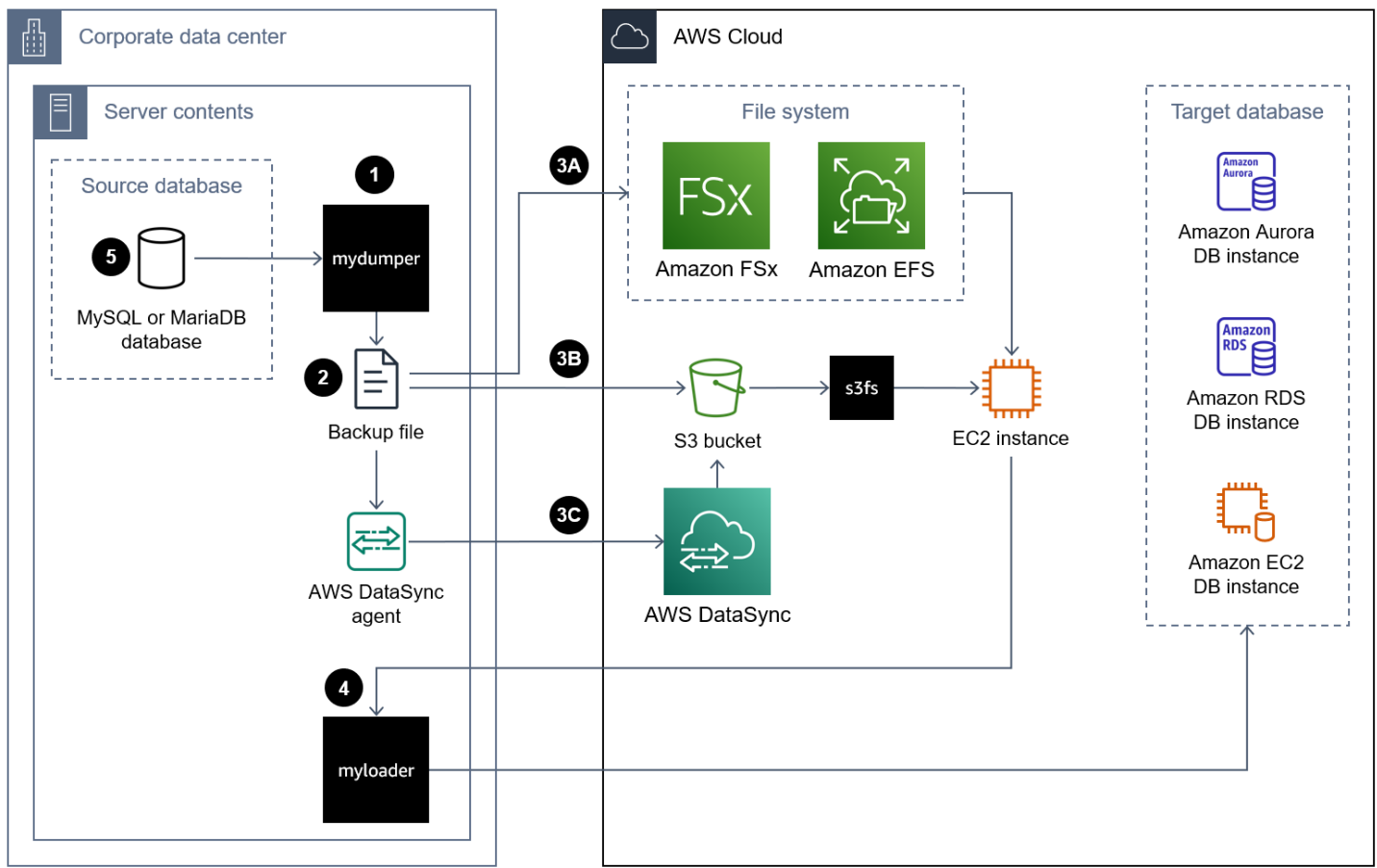
- [Enkripsi beberapa file secara paralel dengan menggunakan --encrypt-threads= <threads>](#)
- Pastikan bahwa ada cukup ruang pada server sumber untuk mengambil file backup database.
- Hasilkan cadangan database dengan file format Percona xstream (.xstream). Untuk informasi lebih lanjut, lihat [Ikhtisar biner xstream di dokumentasi](#) XtraBackup Percona.

MyDumper

[MyDumper](#)(GitHub) adalah alat migrasi logis sumber terbuka yang terdiri dari dua utilitas:

- MyDumper mengekspor cadangan database MySQL yang konsisten. Ini mendukung pencadangan database dengan menggunakan beberapa thread paralel, hingga satu utas per inti CPU yang tersedia.
- myloader membaca file cadangan yang dibuat oleh MyDumper, menghubungkan ke instance database target, dan kemudian mengembalikan database.

Diagram berikut menunjukkan langkah-langkah tingkat tinggi yang terlibat dalam migrasi database dengan menggunakan MyDumper file cadangan. Diagram arsitektur ini mencakup tiga opsi untuk memigrasikan file cadangan dari pusat data lokal ke instans EC2 di AWS Cloud



Berikut ini adalah langkah-langkah yang digunakan MyDumper untuk memigrasikan database ke AWS Cloud:

1. Instal MyDumper dan myloader. Untuk petunjuk, lihat [Cara menginstal mydumper/myloader](#) (GitHub).
2. Gunakan MyDumper untuk membuat cadangan dari sumber MySQL atau database MariaDB. Untuk petunjuk, lihat [Cara menggunakan MyDumper](#).
3. Pindahkan file cadangan ke instans EC2 AWS Cloud dengan menggunakan salah satu pendekatan berikut:

Pendekatan 3A — Pasang sistem file [Amazon FSx](#) atau [Amazon Elastic File System \(Amazon EFS\)](#) ke server lokal yang menjalankan instance database Anda. Anda dapat menggunakan AWS Direct Connect atau Site-to-Site VPN membuat koneksi. Anda dapat langsung mencadangkan database ke berbagi file yang dipasang, atau Anda dapat melakukan pencadangan dalam dua langkah dengan mencadangkan database ke sistem file lokal dan kemudian mengunggahnya ke

volume FSx atau EFS yang dipasang. Selanjutnya, pasang sistem file Amazon FSx atau Amazon EFS, yang juga dipasang di server lokal, pada instans EC2.

Pendekatan 3B — Gunakan REST API AWS CLI, AWS SDK, atau Amazon S3 untuk memindahkan file cadangan secara langsung dari server lokal ke bucket S3. Jika bucket S3 target berada jauh dari pusat data, Anda dapat menggunakan [Amazon S3 Transfer Acceleration untuk mentransfer](#) file lebih cepat. Wilayah AWS Gunakan sistem file [s3fs-fuse](#) untuk memasang bucket S3 pada instans EC2.

Pendekatan 3C — Instal AWS DataSync agen di pusat data lokal, lalu gunakan [AWS DataSync](#) untuk memindahkan file cadangan ke bucket Amazon S3. Gunakan sistem file [s3fs-fuse](#) untuk memasang bucket S3 pada instans EC2.

 Note

Anda juga dapat menggunakan Amazon S3 File Gateway untuk mentransfer file cadangan database besar ke bucket S3 di file. AWS Cloud Untuk informasi selengkapnya, lihat [Menggunakan Amazon S3 File Gateway untuk mentransfer file cadangan](#) dalam panduan ini.

4. Gunakan myloader untuk memulihkan cadangan pada instance database target. Untuk petunjuk, lihat [penggunaan myloader](#) (GitHub).
5. (Opsional) Anda dapat mengatur replikasi antara database sumber dan instance database target. Anda dapat menggunakan replikasi log biner (binlog) untuk mengurangi waktu henti. Untuk informasi selengkapnya, lihat berikut ini:
 - [Mengatur konfigurasi sumber replikasi](#) dalam dokumentasi MySQL
 - Untuk Amazon Aurora, lihat yang berikut ini:
 - [Menyinkronkan cluster DB MySQL Amazon Aurora dengan database MySQL menggunakan replikasi dalam](#) dokumentasi Aurora
 - [Menggunakan replikasi binlog di Amazon Aurora dalam](#) dokumentasi Aurora
 - Untuk Amazon RDS, lihat yang berikut ini:
 - [Bekerja dengan replikasi MySQL](#) dalam dokumentasi Amazon RDS
 - [Bekerja dengan replikasi MariaDB](#) dalam dokumentasi Amazon RDS
 - Untuk Amazon EC2, lihat yang berikut ini:
 - [Menyiapkan Replikasi Berbasis Posisi File Log Biner](#) dalam dokumentasi MySQL

- [Menyiapkan Replika](#) dalam dokumentasi MySQL
- [Menyiapkan Replikasi](#) dalam dokumentasi MariaDB

Keuntungan

- MyDumper mendukung paralelisme dengan menggunakan multi-threading, yang meningkatkan kecepatan operasi pencadangan dan pemulihan.
- MyDumper menghindari rutinitas konversi set karakter yang mahal, yang membantu memastikan kode sangat efisien.
- MyDumper menyederhanakan tampilan data dan parsing dengan menggunakan dumping file terpisah untuk tabel dan metadata.
- MyDumper memelihara snapshot di semua thread dan menyediakan posisi akurat log primer dan sekunder.
- Anda dapat menggunakan Perl Compatible Regular Expressions (PCRE) untuk menentukan apakah akan menyertakan atau mengecualikan tabel atau database.

Batasan

- Anda dapat memilih alat yang berbeda jika proses transformasi data Anda memerlukan file dump menengah dalam format datar, bukan format SQL.
- myloader tidak mengimpor akun pengguna database secara otomatis. Jika Anda memulihkan cadangan ke Amazon RDS atau Aurora, buat ulang pengguna dengan izin yang diperlukan. Untuk informasi selengkapnya, lihat [Menguasai hak istimewa akun pengguna](#) di dokumentasi Amazon RDS. Jika memulihkan cadangan ke instans database Amazon EC2, Anda dapat mengeksport akun pengguna database sumber secara manual dan mengimpornya ke instans EC2.

Praktik terbaik

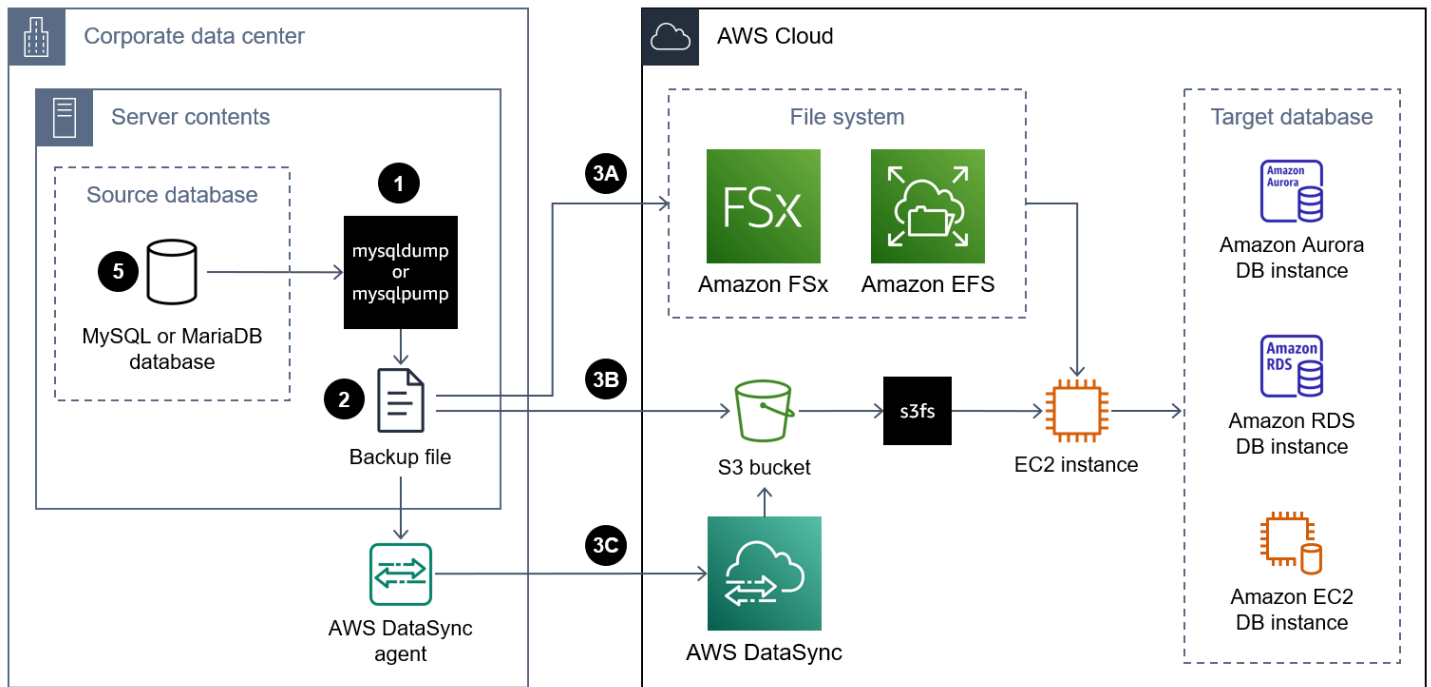
- Konfigurasikan MyDumper untuk membagi setiap tabel menjadi segmen, seperti 10.000 baris di setiap segmen, dan tulis setiap segmen dalam file terpisah. Ini memungkinkan untuk mengimpor data secara paralel nanti.
- Jika Anda menggunakan mesin InnoDB, gunakan `--trx-consistency-only` opsi untuk meminimalkan penguncian.

- Menggunakan MyDumper untuk mengekspor database dapat menjadi read-intensive, dan prosesnya dapat berdampak pada kinerja keseluruhan database produksi. Jika Anda memiliki contoh database replika, jalankan proses ekspor dari replika. Sebelum Anda menjalankan ekspor dari replika, hentikan thread SQL replikasi. Ini membantu proses ekspor berjalan lebih cepat.
- Jangan mengekspor database selama jam kerja puncak. Menghindari jam sibuk dapat menstabilkan kinerja database produksi utama Anda selama ekspor database.
- Amazon RDS for MySQL tidak mendukung plugin. `keyring_aws` Untuk informasi selengkapnya, lihat [Masalah dan batasan yang diketahui](#). Untuk memigrasikan tabel terenkripsi lokal ke instans Amazon RDS, dalam skrip cadangan, Anda harus menghapus atau dari sintaks. `ENCRYPTION DEFAULT ENCRYPTION CREATE TABLE` Untuk enkripsi saat istirahat, Anda dapat menggunakan kunci AWS Key Management Service (AWS KMS). Untuk informasi selengkapnya, lihat [Mengkripsi sumber daya Amazon RDS](#).

mysqldump dan mysqlpump

[mysqldump](#) dan [mysqlpump](#) adalah alat cadangan database asli untuk MySQL. MariaDB mendukung mysqldump tetapi tidak mendukung mysqlpump. Kedua alat ini membuat backup logis dan merupakan bagian dari program klien MySQL. mysqldump mendukung pemrosesan single-threaded. mysqlpump mendukung pemrosesan paralel database dan objek dalam database, untuk mempercepat proses dump. Itu diperkenalkan di MySQL versi 5.7.8. mysqlpump telah dihapus di MySQL versi 8.4.

Diagram berikut menunjukkan langkah-langkah tingkat tinggi yang terlibat dalam migrasi database dengan menggunakan file cadangan mysqldump atau mysqlpump.



Berikut ini adalah langkah-langkah untuk menggunakan mysqldump atau mysqlpump untuk memigrasikan database ke: AWS Cloud

1. Instal MySQL Shell di server lokal. Untuk petunjuk, lihat [Menginstal MySQL Shell dalam dokumentasi MySQL](#). Ini menginstal mysqldump dan mysqlpump.
2. Menggunakan mysqldump atau mysqlpump, buat cadangan sumber, database lokal. [Untuk petunjuk, lihat mysqldump dan mysqlpump di dokumentasi MySQL, atau lihat Membuat Backup dengan mysqldump dalam dokumentasi MariaDB. Untuk informasi selengkapnya tentang menjalankan program MySQL dan menentukan opsi, lihat Menggunakan program MySQL.](#)
3. Pindahkan file cadangan ke instans EC2 AWS Cloud dengan menggunakan salah satu pendekatan berikut:

Pendekatan 3A — Pasang sistem file [Amazon FSx](#) atau [Amazon Elastic File System \(Amazon EFS\)](#) ke server lokal yang menjalankan instance database Anda. Anda dapat menggunakan AWS Direct Connect atau Site-to-Site VPN membuat koneksi. Anda dapat langsung mencadangkan database ke berbagi file yang dipasang, atau Anda dapat melakukan pencadangan dalam dua langkah dengan mencadangkan database ke sistem file lokal dan kemudian mengunggahnya ke volume FSx atau EFS yang dipasang. Selanjutnya, pasang sistem file Amazon FSx atau Amazon EFS, yang juga dipasang di server lokal, pada instans EC2.

Pendekatan 3B — Gunakan REST API AWS CLI, AWS SDK, atau Amazon S3 untuk memindahkan file cadangan secara langsung dari server lokal ke bucket S3. Jika bucket S3 target berada jauh dari pusat data, Anda dapat menggunakan [Amazon S3 Transfer Acceleration untuk mentransfer](#) file lebih cepat. Wilayah AWS Gunakan sistem file [s3fs-fuse](#) untuk memasang bucket S3 pada instans EC2.

Pendekatan 3C — Instal AWS DataSync agen di pusat data lokal, lalu gunakan [AWS DataSync](#) untuk memindahkan file cadangan ke bucket Amazon S3. Gunakan sistem file [s3fs-fuse](#) untuk memasang bucket S3 pada instans EC2.

 Note

Anda juga dapat menggunakan Amazon S3 File Gateway untuk mentransfer file cadangan database besar ke bucket S3 di file. AWS Cloud Untuk informasi selengkapnya, lihat [Menggunakan Amazon S3 File Gateway untuk mentransfer file cadangan](#) dalam panduan ini.

4. Gunakan metode pemulihan asli untuk memulihkan cadangan pada database target. Untuk petunjuk, lihat [SQL-Format Memuat Ulang Cadangan](#) dalam dokumentasi MySQL, atau lihat [Memulihkan Data dari](#) File Dump dalam dokumentasi MariaDB.
5. (Opsional) Anda dapat mengatur replikasi antara database sumber dan instance database target. Anda dapat menggunakan replikasi log biner (binlog) untuk mengurangi waktu henti. Untuk informasi selengkapnya, lihat berikut ini:
 - [Mengatur konfigurasi sumber replikasi](#) dalam dokumentasi MySQL
 - Untuk Amazon Aurora, lihat yang berikut ini:
 - [Menyinkronkan cluster DB MySQL Amazon Aurora dengan database MySQL menggunakan replikasi dalam](#) dokumentasi Aurora
 - [Menggunakan replikasi binlog di Amazon Aurora dalam](#) dokumentasi Aurora
 - Untuk Amazon RDS, lihat yang berikut ini:
 - [Bekerja dengan replikasi MySQL](#) dalam dokumentasi Amazon RDS
 - [Bekerja dengan replikasi MariaDB](#) dalam dokumentasi Amazon RDS
 - Untuk Amazon EC2, lihat yang berikut ini:
 - [Menyiapkan Replikasi Berbasis Posisi File Log Biner](#) dalam dokumentasi MySQL
 - [Menyiapkan Replika](#) dalam dokumentasi MySQL

- [Menyiapkan Replikasi](#) dalam dokumentasi MariaDB

Keuntungan

- mysqldump dan mysqlpump termasuk dalam instalasi MySQL Server
- File cadangan yang dihasilkan oleh alat-alat ini dalam format yang lebih mudah dibaca.
- Sebelum memulihkan file cadangan, Anda dapat memodifikasi file.sql yang dihasilkan dengan menggunakan editor teks standar.
- Anda dapat membuat cadangan tabel, database, atau bahkan pemilihan data tertentu.
- mysqldump dan mysqlpump adalah arsitektur mesin independen.

Batasan

- mysqldump adalah proses pencadangan single-threaded. Kinerja untuk mengambil cadangan baik untuk database kecil, tetapi bisa menjadi tidak efisien ketika ukuran cadangan lebih besar dari 10 GB.
- Backup file dalam format logis sangat banyak, terutama ketika disimpan sebagai teks, dan sering lambat untuk membuat dan memulihkan.
- Pemulihan data bisa lambat karena menerapkan kembali pernyataan SQL dalam instans DB target melibatkan pemrosesan disk I/O dan CPU intensif untuk penyisipan, pembuatan indeks, dan penegakan batasan integritas referensial.
- Utilitas mysqlpump tidak didukung untuk versi MySQL lebih awal dari 5.7.8 atau untuk versi 8.4 dan yang lebih baru.
- Secara default, mysqlpump tidak mengambil cadangan dari database sistem, seperti atau. `performance_schema` `sys` Untuk membuat cadangan bagian dari database sistem, beri nama secara eksplisit di baris perintah.
- mysqldump tidak mencadangkan pernyataan InnoDB. `CREATE TABLESPACE`

 Note

Cadangan pernyataan CREATE TABLESPACE dan database sistem hanya berguna ketika Anda memulihkan cadangan database MySQL atau MariaDB ke instance EC2. Cadangan ini tidak digunakan untuk Amazon RDS atau Aurora.

Praktik terbaik

- Ketika Anda memulihkan cadangan database, nonaktifkan pemeriksaan kunci, seperti FOREIGN_KEY_CHECKS, pada tingkat sesi dalam database target. Ini meningkatkan kecepatan restorasi.
- Pastikan pengguna database memiliki [hak istimewa](#) yang cukup untuk membuat dan memulihkan cadangan.

Pisahkan cadangan

Strategi pencadangan terpisah adalah ketika Anda memigrasikan server database besar dengan membagi cadangan menjadi beberapa bagian. Anda dapat menggunakan pendekatan yang berbeda untuk memigrasikan setiap bagian cadangan. Ini bisa menjadi pilihan terbaik untuk kasus penggunaan berikut:

- Server database besar tetapi database individu kecil — Ini adalah pendekatan yang baik ketika ukuran total server database adalah beberapa TB tetapi ukuran masing-masing individu, database pengguna independen kurang dari 1 TB. Untuk mengurangi periode migrasi secara keseluruhan, Anda dapat memigrasikan database individual secara terpisah dan paralel.

Mari kita gunakan contoh server database 2 TB lokal. Server ini terdiri dari empat database yang masing-masing 0,5 TB. Anda dapat mengambil cadangan dari setiap database individu secara terpisah. Saat memulihkan cadangan, Anda dapat memulihkan semua database pada instance secara paralel, atau jika database independen, Anda dapat memulihkan setiap cadangan pada instance terpisah. Ini adalah praktik terbaik untuk memulihkan database independen pada instance terpisah, alih-alih memulihkannya pada instance yang sama. Untuk informasi selengkapnya, lihat Praktik terbaik dalam panduan ini.

- Server database besar tetapi tabel database individu kecil — Ini adalah pendekatan yang baik ketika ukuran total server database adalah beberapa TB tetapi ukuran setiap tabel database

independen kurang dari 1 TB. Untuk mengurangi periode migrasi secara keseluruhan, Anda dapat memigrasikan tabel independen satu per satu.

Mari kita gunakan contoh database pengguna tunggal yaitu 1 TB, dan itu adalah satu-satunya database di server database lokal. Ada 10 tabel dalam database, dan masing-masing 100 GB. Anda dapat mengambil cadangan dari masing-masing tabel secara terpisah. Saat memulihkan cadangan, Anda dapat mengembalikan semua tabel pada sebuah instance secara paralel.

- Database berisi tabel beban kerja transaksional dan non-transaksional — Mirip dengan kasus penggunaan sebelumnya, Anda dapat menggunakan pendekatan cadangan terpisah ketika Anda memiliki tabel beban kerja transaksional dan non-transaksional dalam database yang sama.

Mari kita gunakan contoh database 2 TB yang terdiri dari 0,5 TB tabel beban kerja kritis yang digunakan untuk pemrosesan transaksi online (OLTP) dan tabel 1,5 TB tunggal yang digunakan untuk pengarsipan data lama. Anda dapat mengambil cadangan semua objek database kecuali tabel arsip sebagai transaksi tunggal dan cadangan yang konsisten. Kemudian, Anda hanya mengambil cadangan terpisah dari tabel arsip. Untuk cadangan tabel arsip, Anda juga dapat mempertimbangkan untuk mengambil beberapa backup paralel dengan menggunakan kondisi untuk membagi jumlah baris dalam file cadangan. Berikut ini adalah contohnya:

```
mysqldump -p your_db1 --tables your_table1 --where="column1 between 1 and 1000000 " >
  your_table1_part1.sql
mysqldump -p your_db1 --tables your_table1 --where="column1 between 1000001 and
  2000000 " > your_table1_part2.sql
mysqldump -p your_db1 --tables your_table1 --where="column1 > 2000000 " >
  your_table1_part3.sql
```

Saat memulihkan file cadangan, Anda dapat mengembalikan cadangan beban kerja transaksional dan cadangan tabel arsip secara paralel.

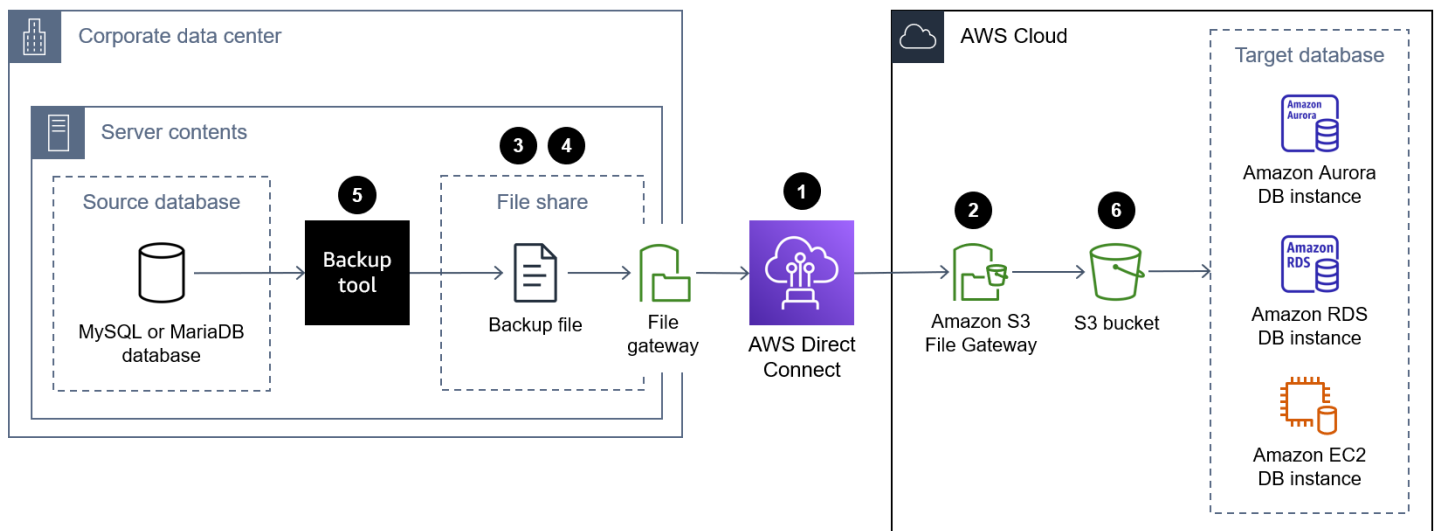
- Batasan sumber daya komputasi — Jika Anda memiliki sumber daya komputasi terbatas di server lokal, seperti CPU, memori, atau disk I/O, hal ini dapat memengaruhi stabilitas dan kinerja saat mengambil cadangan. Alih-alih mengambil cadangan lengkap, Anda dapat membaginya menjadi beberapa bagian.

Misalnya, server produksi lokal mungkin penuh dengan beban kerja dan memiliki sumber daya CPU yang terbatas. Jika Anda mengambil cadangan tunggal dari database multi-terabyte di server ini, ia dapat mengkonsumsi sumber daya CPU tambahan dan berdampak buruk pada server produksi. Alih-alih mengambil cadangan database lengkap, bagilah cadangan menjadi beberapa bagian, seperti masing-masing 2-3 tabel.

Menggunakan Amazon S3 File Gateway untuk mentransfer file cadangan

[Amazon S3 File Gateway](#) menghubungkan lingkungan lokal Anda ke Amazon Simple Storage Service (Amazon S3) melalui antarmuka file sehingga Anda dapat menyimpan dan mengambil objek Amazon S3 dengan menggunakan protokol file standar industri, seperti Network File System (NFS) dan Server Message Block (SMB). Ini dirancang untuk menjadi solusi yang hemat biaya dan terukur untuk menyimpan data di cloud. Karena Anda dapat menggunakannya untuk menyimpan file cadangan database, layanan ini dapat membantu Anda memigrasikan database lokal yang besar ke file. AWS Cloud Misalnya, Anda dapat menggunakan Amazon S3 File Gateway dan alat pencadangan basis data pilihan Anda untuk mencadangkan database MySQL atau MariaDB besar langsung ke bucket Amazon S3. Anda kemudian dapat memasang bucket S3 ke instance target dan memulihkan cadangan.

Diagram berikut menunjukkan langkah-langkah tingkat tinggi yang terlibat saat menggunakan Amazon S3 File Gateway untuk mentransfer file cadangan untuk database lokal ke bucket S3 di AWS Cloud



Berikut ini adalah langkah-langkah untuk menggunakan Amazon S3 File Gateway untuk mentransfer file cadangan database dari pusat data lokal ke bucket S3 di AWS Cloud

1. Hubungkan pusat data lokal ke AWS Cloud dengan menggunakan layanan seperti AWS Direct Connect atau AWS Site-to-Site VPN atau dengan menggunakan koneksi internet publik.

2. Buat Gateway File S3. Untuk petunjuk, lihat [Membuat gateway Anda](#).
3. Buat berbagi file NFS atau SMB yang di-host oleh S3 File Gateway. Untuk petunjuk, lihat [Membuat berbagi file](#).
4. Pasang berbagi file NFS atau SMB di server lokal yang menghosting database MySQL atau MariaDB Anda. Untuk petunjuk, lihat [Memasang dan menggunakan berbagi file Anda](#).
5. Cadangkan database MySQL atau MariaDB lokal ke direktori tempat berbagi file NFS dipasang. Anda dapat menggunakan salah satu alat cadangan yang dibahas dalam panduan ini.
6. Kembalikan cadangan database pada instance database target dengan menggunakan salah satu pendekatan yang dibahas dalam panduan ini.

Keuntungan

- Dengan membuat cadangan database secara langsung di bucket S3 dan memulihkan cadangan pada instans DB target langsung dari bucket S3 yang sama, Anda dapat mempercepat proses migrasi ujung ke ujung secara signifikan.
- File cadangan database disimpan secara tahan lama di Amazon S3, dan Anda memilih kebijakan manajemen siklus hidup dan kelas penyimpanan S3.

Batasan

Berikut ini adalah batasan saat menggunakan berbagi file Amazon S3 File Gateway:

- Jumlah maksimum pembagian file per gateway adalah 50.
- Untuk mencegah konflik baca dan tulis ketika beberapa berbagi file menggunakan bucket S3 yang sama, Anda harus mengonfigurasi setiap berbagi file untuk menggunakan nama awalan yang unik.
- Ukuran maksimum file individual adalah 5 TB, yang merupakan ukuran maksimum dari setiap objek individu di Amazon S3.
- Panjang jalur maksimum adalah 1024 karakter.
- Windows ACL didukung hanya pada berbagi file yang diaktifkan untuk Active Directory ketika Anda menggunakan klien Windows SMB untuk mengakses berbagi file.
- Amazon S3 File Gateway mendukung maksimal 10 entri ACL untuk setiap file dan direktori.
- Pengaturan ACL root dari berbagi file SMB hanya ada di gateway. Pengaturan ini persisten di seluruh pembaruan gateway dan restart.

 Note

Jika Anda mengonfigurasi ACL di root alih-alih folder induk di bawah root, izin ACL tidak tetap di Amazon S3.

Praktik terbaik

Untuk informasi selengkapnya tentang praktik terbaik untuk Amazon S3 File Gateway, lihat [Praktik terbaik dalam dokumentasi](#) Gateway File S3.

Praktik terbaik untuk memigrasi database MySQL dan MariaDB yang besar

Selain praktik terbaik khusus alat yang tercantum untuk setiap opsi migrasi, tinjau praktik terbaik umum berikut. Praktik terbaik ini berlaku saat memigrasikan database MySQL dan MariaDB multi-terabyte yang besar, terlepas dari alat yang Anda pilih:

- Pastikan ada cukup ruang pada basis data sumber dan tujuan untuk mengambil dan memulihkan cadangan.
- Jangan membuat indeks sekunder pada instance database target sampai migrasi selesai. Indeks sekunder menambahkan overhead pemeliharaan tambahan selama impor dan dapat memperlambat proses impor.
- Jika Anda menggunakan pendekatan multi-utas, pilih jumlah utas yang tepat. Untuk ekspor, kami sarankan Anda menggunakan satu utas untuk setiap inti CPU. Untuk impor, kami sarankan Anda menggunakan satu thread untuk setiap dua core CPU.
- Dump data sering dilakukan dari server database aktif yang merupakan bagian dari lingkungan produksi mission-critical. Jika dump data sangat memengaruhi kinerja dan ini tidak dapat diterima di lingkungan Anda, pertimbangkan salah satu hal berikut:
 - Server sumber memiliki replika, Anda dapat membuang data dari salah satu replika.
 - Server sumber dicakup oleh prosedur pencadangan reguler:
 - Jika format cadangan cocok untuk impor langsung ke database target, gunakan data cadangan sebagai input untuk proses impor.
 - Jika format cadangan tidak cocok untuk impor langsung ke database target, gunakan cadangan untuk menyediakan database sementara dan membuang data darinya.
 - Jika replika dan cadangan tidak tersedia:
 - Lakukan dump selama jam-jam off-peak, ketika lalu lintas produksi berada pada titik terendah.
 - Kurangi konkurensi operasi dump sehingga server memiliki kapasitas cadangan yang cukup untuk menangani lalu lintas produksi.
- Buat dump database yang dibuat pengguna saja.
- Buat ulang pengguna pada database target dan konfigurasi izin mereka. Untuk informasi selengkapnya, lihat [Manajemen identitas dan akses untuk Amazon RDS](#), [Identitas, dan manajemen akses untuk Amazon Aurora](#), [atau Manajemen identitas dan akses untuk Amazon EC2](#).

- Saat memigrasikan server database besar yang terdiri dari beberapa database independen, buat instance terpisah untuk setiap database. Ini membantu Anda mengelola database secara lebih efisien dan dapat meningkatkan penyediaan sumber daya, dan sumber daya komputasi terpisah dapat meningkatkan kinerja database.

Sumber daya

AWS Bimbingan Preskriptif

- [Buku pedoman portofolio untuk migrasi AWS besar](#)
- [Strategi migrasi untuk database relasional](#)
- [Memigrasikan database MySQL lokal ke Amazon RDS for MySQL](#)
- [Mengatur replikasi data antara Amazon RDS for MySQL dan MySQL di Amazon menggunakan GTID EC2](#)
- [Memigrasi database MariaDB lokal ke Amazon RDS for MariaDB menggunakan alat bawaan](#)

AWS posting blog

- [Praktik terbaik keamanan untuk Amazon RDS untuk instans MySQL dan MariaDB](#)
- [Migrasi MariaDB yang dikelola sendiri ke Amazon Aurora MySQL](#)

Sumber daya untuk memulihkan cadangan

- [Membuat ember](#) (dokumentasi Amazon S3)
- [Menyambung ke instans Linux Anda menggunakan SSH](#) (EC2 dokumentasi Amazon)
- [Mengkonfigurasi AWS CLI](#) (AWS CLI dokumentasi)
- [perintah sinkronisasi](#) (Referensi AWS CLI Perintah)
- [Membuat kebijakan IAM untuk mengakses sumber daya Amazon S3](#) (dokumentasi Aurora)
- [Prasyarat cluster DB](#) (dokumentasi Aurora)
- [Bekerja dengan grup subnet DB](#) (dokumentasi Aurora)
- [Membuat grup keamanan VPC untuk cluster DB pribadi](#) (dokumentasi Aurora)
- [Memulihkan cluster DB MySQL Aurora dari bucket Amazon S3](#) (dokumentasi Aurora)
- [Menyiapkan replikasi dengan MySQL atau cluster Aurora DB lainnya](#) (dokumentasi Aurora)
- [prosedur rds_set_external_master](#) (dokumentasi Amazon RDS)
- [prosedur rds_start_replication](#) (dokumentasi Amazon RDS)

AWS pemasaran

- [Amazon Aurora](#)
- [Amazon RDS for MariaDB](#)
- [Amazon RDS for MySQL](#)
- [Amazon S3 File Gateway](#)

Sumber daya lainnya

- [Percona XtraBackup](#)
- [MyDumper](#)
- [mysqldump](#)
- [mysqlpump](#)

Riwayat dokumen

Tabel berikut menjelaskan perubahan signifikan pada panduan ini. Jika Anda ingin diberi tahu tentang pembaruan masa depan, Anda dapat berlangganan umpan [RSS](#).

Perubahan	Deskripsi	Tanggal
ketersediaan mysqlpump	mysqlpump telah dihapus di MySQL versi 8.4. Kami memperbarui bagian mysqldump dan mysqlpump untuk mencerminkan perubahan ketersediaan ini.	November 21, 2024
MyDumper praktik terbaik	Kami memperbarui praktik terbaik MyDumper untuk menambahkan informasi tentang migrasi tabel database terenkripsi.	Oktober 24, 2024
Versi XtraBackup Percona	Di XtraBackup bagian Percona , kami memperbarui instruksi untuk mencerminkan versi Percona yang didukung oleh XtraBackup Amazon Aurora MySQL dan Amazon RDS.	3 Agustus 2023
Publikasi awal	—	6 April 2023

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