Migrazione di database MySQL o MariaDB di grandi dimensioni da più terabyte a AWS

AWS Linee guida prescrittive

AWS Linee guida prescrittive: Migrazione di database MySQL o MariaDB di grandi dimensioni da più terabyte a AWS

Table of Contents

Migrazione di database MySQL o MariaDB di grandi dimensioni da più terabyte a AWS

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

Novembre 2024 (storia del documento)

Molte organizzazioni che dispongono di server di database MySQL e MariaDB locali sono interessate a migrare i propri carichi di lavoro di database su. Cloud AWS Molti scelgono Amazon Relational Database Service (Amazon RDS) per MariaDB, Amazon RDS for MySQL o Amazon Aurora versione compatibile con MySQL. Amazon RDS è progettato per semplificare la configurazione, il funzionamento e la scalabilità dei database relazionali nel cloud. Amazon Aurora fa parte di Amazon RDS e offre sicurezza integrata, backup continui, elaborazione senza server, fino a 15 repliche di lettura, replica automatizzata in più regioni e integrazione con altri. Servizi AWS

Sebbene la migrazione a uno di questi sistemi Servizi AWS possa offrire molti vantaggi, la migrazione del database è una delle attività più impegnative e dispendiose in termini di tempo che gli amministratori di database devono svolgere. Richiede una pianificazione e un'implementazione precise per migrare database di grandi dimensioni e garantire che le prestazioni del carico di lavoro migrato siano equivalenti o migliorate. In questa guida, i database di grandi dimensioni possono fare riferimento a un singolo database da più terabyte o fare riferimento a molti database di grandi dimensioni che sommano diversi terabyte di dati. La selezione dei servizi e degli strumenti di migrazione giusti è fondamentale per il successo della migrazione. Esistono due approcci comuni per la migrazione di un database: logico e fisico. Per ulteriori informazioni su questi approcci, consulta la documentazione di MySQL e MariaDB.

Questa guida illustra vari strumenti open source o di terze parti che puoi utilizzare per migrare database MySQL e MariaDB di grandi dimensioni, locali e da più terabyte verso Amazon RDS for MariaDB, Amazon RDS for MySQL o Amazon Aurora MySQL Compatible Edition. Le opzioni discusse in questa guida utilizzano approcci di migrazione logica o fisica e ciascuna opzione include diversi approcci per il trasferimento dei file di backup del database di grandi dimensioni dal data center locale al cloud, dove è possibile ripristinare il database dal file di backup.

Destinatari principali

Questa guida è destinata agli amministratori di database del programma, agli ingegneri di database, agli ingegneri della migrazione, ai project manager e ai gestori delle operazioni o dell'infrastruttura che intendono migrare i propri database MySQL o MariaDB su. Cloud AWS

Obiettivi aziendali specifici

L'obiettivo di questa guida è aiutarti a:

- Scegliete un approccio di migrazione per un database di grandi dimensioni che meglio si adatti al vostro caso d'uso e al vostro ambiente.
- Evita i ritardi e le perdite finanziarie che possono verificarsi quando la strategia di migrazione è difettosa.
- Scopri i vantaggi e i limiti di ciascuna opzione di migrazione.
- Scopri i diversi approcci che puoi utilizzare per trasferire file di backup di database di grandi dimensioni dal data center locale al Cloud AWS.
- Esamina le best practice generali per la migrazione di database di grandi dimensioni e consulta anche le best practice per ogni strumento, che possono aiutarti a migrare il database in modo più efficiente.

Opzioni di migrazione per database MySQL e MariaDB di grandi dimensioni

Puoi scegliere tra un'ampia gamma di opzioni per migrare da database MySQL o MariaDB locali a istanze di database Amazon Relational Database Service (Amazon RDS) o Amazon Aurora Edition. MySQL-Compatible La scelta dell'approccio e dello strumento di migrazione giusti è essenziale per una migrazione di successo e in questa guida valuterai le opzioni in base ai tuoi requisiti di usabilità, dimensione dei dati e tempi di inattività.

Di seguito sono riportati gli strumenti e gli approcci di migrazione più comuni disponibili per migrare in modo efficiente database MySQL autogestiti da più terabyte verso istanze di database Amazon RDS, Aurora o Amazon Elastic Compute Cloud (Amazon EC2):

- Percona XtraBackup(Fisico)
- MyDumper(Logico)
- mysqldump e mysqlpump(Logico)
- Backup diviso(Fisico, logico o entrambi)

Di seguito sono riportati gli strumenti e gli approcci di migrazione più comuni disponibili per migrare in modo efficiente database di più terabyte (MySQL-compatible come MariaDB) verso istanze di database Amazon RDS, Aurora o Amazon EC2:

- MyDumper(Logico)
- mysqldump e mysqlpump(Logico)
- Backup diviso(Fisico, logico o entrambi)

Per ogni strumento di migrazione, esistono diversi approcci che è possibile utilizzare per trasferire il file di backup del database di grandi dimensioni su Cloud AWS. Sono disponibili opzioni per ogni strumento e puoi anche utilizzare Amazon S3 File Gateway. Per ulteriori informazioni sul tagging, consulta Utilizzo di Amazon S3 File Gateway per trasferire file di backupin questa guida.

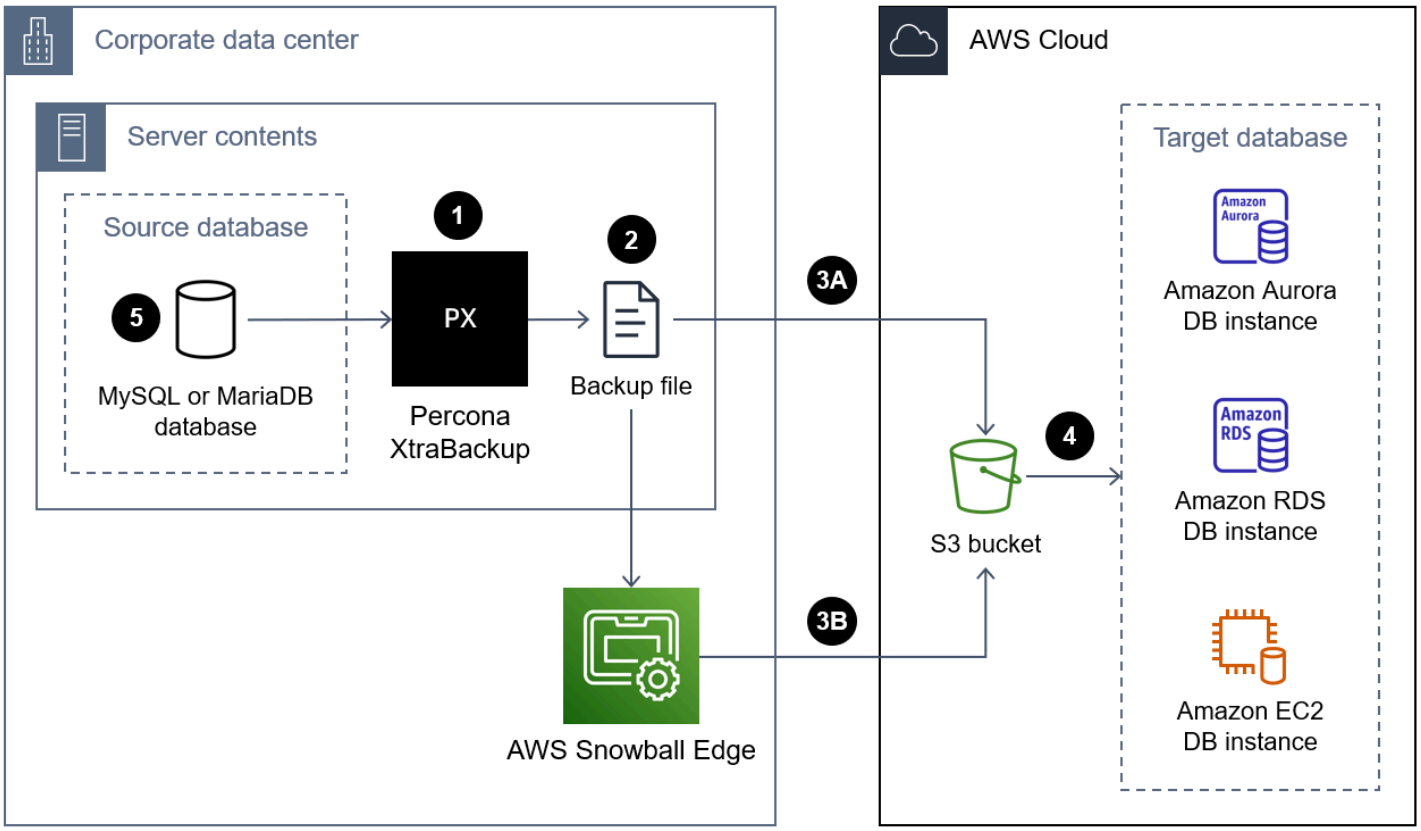
Percona XtraBackup

> **Important**
>
> Percona non XtraBackup è supportato per le versioni 10.3 o successive di MariadB ed è supportato solo parzialmente per le versioni 10.1 e 10.2.

Percona XtraBackup è un comune software di backup caldo open source per MySQL e MariaDB che esegue backup non bloccanti per i motori di archiviazione InnoDB e XtraDB. Funziona con server MySQL o MariadB. Per ulteriori informazioni sullo strumento e su alcune delle sue caratteristiche e vantaggi, consulta Informazioni su Percona nella documentazione di Percona XtraBackup. XtraBackup

Questo strumento utilizza l'approccio della migrazione fisica. Copia direttamente la directory dei dati MySQL o MariaDB e i file al suo interno. Per database di grandi dimensioni, come quelli più grandi di 100 GB, questo può fornire tempi di ripristino significativamente migliori rispetto ad altri strumenti. È necessario creare un backup del database di origine locale, migrare i file di backup nel cloud e quindi ripristinare il backup sulla nuova istanza del database di destinazione.

Il diagramma seguente mostra i passaggi di alto livello coinvolti nella migrazione di un database utilizzando un file di backup Percona. XtraBackup A seconda della dimensione del file di backup, sono disponibili due opzioni per trasferire il backup in un bucket Amazon Simple Storage Service (Amazon S3) nel. Cloud AWS



Di seguito sono riportati i passaggi per utilizzare Percona per XtraBackup migrare un database verso: Cloud AWS

1. Installa Percona XtraBackup sul server locale. Se utilizzi Amazon Aurora MySQL versione 2 o Amazon RDS, consulta Installazione di Percona 2.4. XtraBackup Se utilizzi Amazon Aurora MySQL versione 3, consulta Installazione di Percona 8.0 nella documentazione di XtraBackupPercona. XtraBackup
2. Crea un backup completo del database MySQL o MariaDB di origine. Per istruzioni su Percona XtraBackup 2.4, vedi Backup completo. Per istruzioni per Percona XtraBackup 8.0, vedi Creare un backup completo.
3. Trasferisci i file di backup su Internet utilizzando un servizio o uno strumento approvato dalla tua organizzazione, come il seguente:
   - AWS Site-to-Site VPN
   - AWS Client VPN
   - AWS Direct Connect

- Amazon S3 File Gateway (per ulteriori informazioni, consulta Utilizzo di Amazon S3 File Gateway per trasferire file di backup questa guida).
- AWS Command Line Interface (AWS CLI)

4. Dal bucket Amazon S3, ripristina i file di backup nell'istanza del database di destinazione. Per le istruzioni, consulta quanto segue:
   - Per Aurora MySQL-Compatible Edition, consulta Migrazione dei dati da MySQL utilizzando un bucket Amazon S3 nella documentazione di Amazon RDS.
   - Per Amazon RDS for MySQL o per Amazon EC2, consulta Importazione di dati in un'istanza DB MySQL.
   - Per Amazon RDS for MariaDB o per Amazon EC2, consulta Importazione di dati in un'istanza DB MariaDB.
5. (Facoltativo) Puoi configurare la replica tra il database di origine e l'istanza del database di destinazione. È possibile utilizzare la replica con log binario (binlog) per ridurre i tempi di inattività. Per ulteriori informazioni, consulta gli argomenti seguenti:
   - Impostazione della configurazione della sorgente di replica nella documentazione MySQL
   - Per Amazon Aurora, consulta quanto segue:
     - Sincronizzazione del cluster Amazon Aurora MySQL DB con il database MySQL utilizzando la replica nella documentazione di Aurora
     - Utilizzo della replica binlog in Amazon Aurora nella documentazione di Aurora
   - Per Amazon RDS, consulta quanto segue:
     - Utilizzo della replica MySQL nella documentazione di Amazon RDS
     - Utilizzo della replica MariadB nella documentazione di Amazon RDS
   - Per Amazon EC2, consulta quanto segue:
     - Configurazione della replica basata sulla posizione dei file di log binari nella documentazione MySQL
     - Configurazione delle repliche nella documentazione MySQL
     - Configurazione della replica nella documentazione di MariadB

Vantaggi

- Poiché Percona XtraBackup utilizza un approccio di migrazione fisica, il processo di ripristino è in genere più veloce degli strumenti che utilizzano un approccio di migrazione logico. Questo perché

le prestazioni sono limitate dalla velocità effettiva del disco o della rete piuttosto che dalle risorse di calcolo necessarie per l'elaborazione dei dati.

- Poiché il processo di ripristino è una copia diretta dei file dal bucket S3 all'istanza del database di destinazione, i file Percona in genere si ripristinano più rapidamente XtraBackup dei file di backup creati con altri strumenti.
- Percona è adattabile XtraBackup . Ad esempio, supporta più thread per aiutarti a copiare i file più velocemente e supporta la compressione per ridurre le dimensioni del backup.

Limitazioni

- Il backup offline non è possibile perché Percona XtraBackup deve avere accesso al server del database di origine.
- Percona XtraBackup può essere utilizzato solo su sistemi con architetture di sistema identiche. Ad esempio, non è possibile ripristinare un backup di un database di origine in esecuzione su Intel per Windows Server su un server di destinazione ARM per Linux.
- Percona XtraBackup non è supportato per MariaDB versione 10.3 o successiva ed è supportato solo parzialmente per MariaDB versione 10.2 e versione 10.1. Per ulteriori informazioni, vedere XtraBackup Panoramica di Percona: compatibilità con MariaDB nella knowledge base di MariaDB.
- Non è possibile utilizzare Percona XtraBackup per ripristinare un database MariaDB di origine su un'istanza di database MySQL di destinazione, come Amazon RDS for MySQL o Aurora. MySQL-Compatible
- Il volume totale di dati e il numero di oggetti che è possibile archiviare in un bucket S3 sono illimitati, tuttavia la dimensione massima del file è di 5 TB. Se il file di backup supera i 5 TB, puoi suddividerlo in più file più piccoli.
- Quando l'`innodb_file_per_table`impostazione è disattivata, Percona XtraBackup non supporta backup parziali che utilizzano`--tables`,,, `--tables-exclude` `--tables-file--databases`, `--databases-exclude` o. `--databases-file` Per ulteriori informazioni sulla XtraBackup versione 2.4 di Percona, vedere Backup parziali. Per ulteriori informazioni sulla XtraBackup versione 8.0 di Percona, consulta Creare un backup parziale.

Best practice

- Per migliorare le prestazioni del processo di backup, procedi come segue:
  - Copia più file in parallelo usando --parallel= <threads>

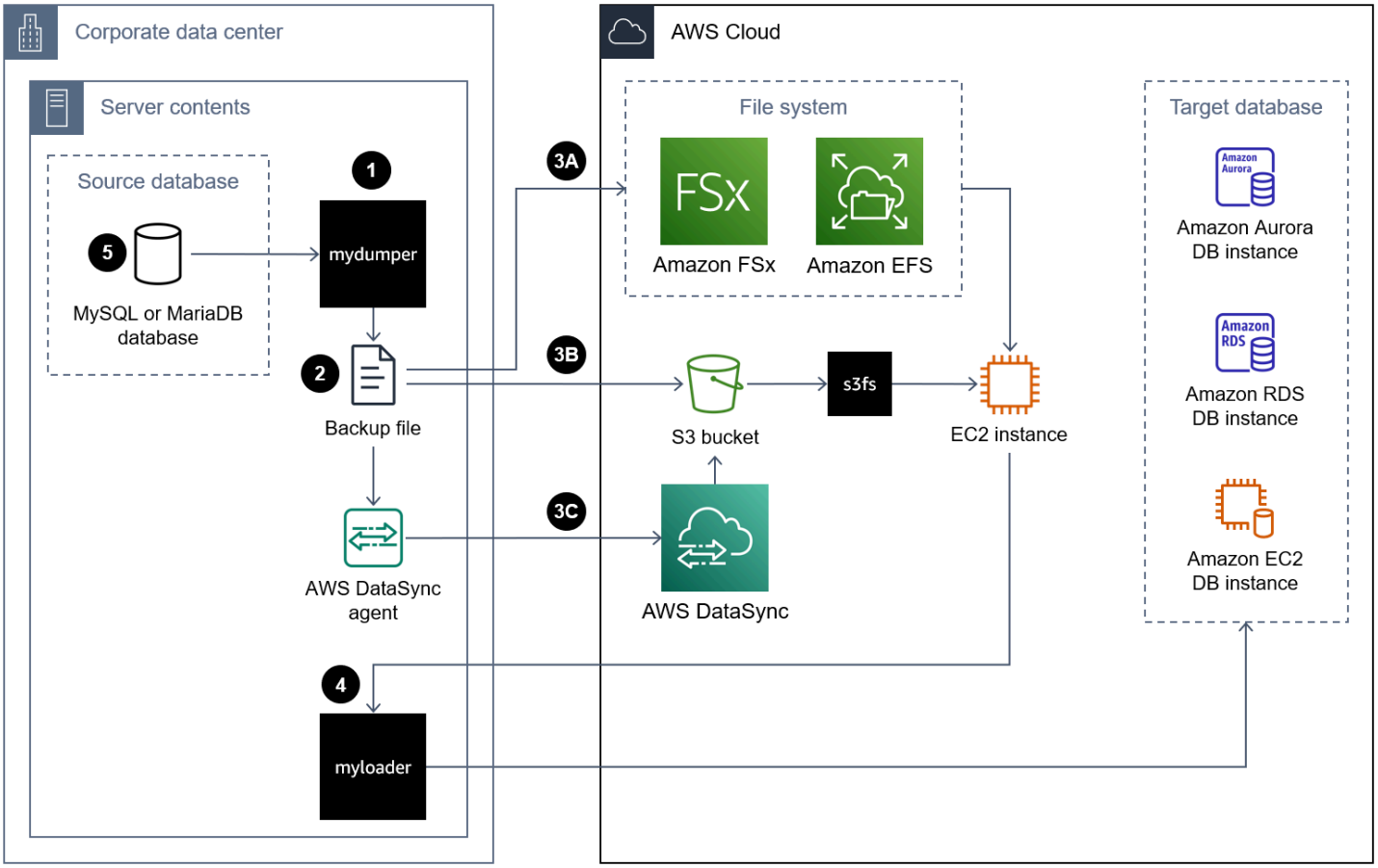
  - Comprimi più file in parallelo usando --compress-threads= <threads>
  - Aumenta la memoria usando --use-memory= <size>
  - Crittografa più file in parallelo usando --encrypt-threads= <threads>
- Assicurati che ci sia spazio sufficiente sul server di origine per archiviare i file di backup del database.
- Genera il backup del database con il file in formato Percona xbstream (.xbstream). Per ulteriori informazioni, vedere La panoramica binaria di xbstream nella documentazione di Percona. XtraBackup

MyDumper

MyDumper(GitHub) è uno strumento di migrazione logica open source composto da due utilità:

- MyDumper esporta un backup coerente dei database MySQL. Supporta il backup del database utilizzando più thread paralleli, fino a un thread per core della CPU disponibile.
- myloader legge i file di backup creati da MyDumper, si connette all'istanza del database di destinazione e quindi ripristina il database.

Il diagramma seguente mostra i passaggi di alto livello necessari per la migrazione di un database utilizzando un file di backup. MyDumper Questo diagramma di architettura include tre opzioni per la migrazione del file di backup dal data center locale a un'istanza EC2 in. Cloud AWS



Di seguito sono riportati i passaggi da utilizzare per MyDumper migrare un database verso: Cloud AWS

1. Install MyDumper e myloader. Per istruzioni, vedi Come installare mydumper/myloader ()GitHub.
2. MyDumper Da utilizzare per creare un backup del database MySQL o MariaDB di origine. Per istruzioni, vedi Come usare. MyDumper
3. Sposta il file di backup in un'istanza EC2 Cloud AWS utilizzando uno dei seguenti approcci:

   Approccio 3A: monta un file system Amazon FSx o Amazon Elastic File System (Amazon EFS) sul server locale che esegue l'istanza di database. È possibile utilizzare AWS Direct Connect o Site-to-Site VPN per stabilire la connessione. È possibile eseguire il backup direttamente del database nella condivisione di file montata oppure eseguire il backup in due passaggi eseguendo il backup del database su un file system locale e quindi caricandolo sul volume FSx o EFS montato. Successivamente, monta il file system Amazon FSx o Amazon EFS, anch'esso montato sul server locale, su un'istanza EC2.

Approccio 3B: utilizza l' AWS CLI AWS SDK o l'API REST di Amazon S3 per spostare direttamente il file di backup dal server locale a un bucket S3. Se il bucket S3 di destinazione si trova in un Regione AWS ambiente lontano dal data center, puoi utilizzare Amazon S3 Transfer Acceleration per trasferire il file più rapidamente. Usa il file system s3fs-fuse per montare il bucket S3 sull'istanza EC2.

Approccio 3C: installa l' AWS DataSync agente nel data center locale, quindi utilizzalo AWS DataSyncper spostare il file di backup in un bucket Amazon S3. Usa il file system s3fs-fuse per montare il bucket S3 sull'istanza EC2.

> **Note**
>
> Puoi anche utilizzare Amazon S3 File Gateway per trasferire i file di backup del database di grandi dimensioni in un bucket S3 nel. Cloud AWS Per ulteriori informazioni sul tagging, consulta Utilizzo di Amazon S3 File Gateway per trasferire file di backupin questa guida.

4. Usa myloader per ripristinare il backup sull'istanza del database di destinazione. Per istruzioni, consulta myloader usage (). GitHub
5. (Facoltativo) È possibile impostare la replica tra il database di origine e l'istanza del database di destinazione. È possibile utilizzare la replica con log binario (binlog) per ridurre i tempi di inattività. Per ulteriori informazioni, consulta gli argomenti seguenti:
   - Impostazione della configurazione della sorgente di replica nella documentazione MySQL
   - Per Amazon Aurora, consulta quanto segue:
     - Sincronizzazione del cluster Amazon Aurora MySQL DB con il database MySQL utilizzando la replica nella documentazione di Aurora
     - Utilizzo della replica binlog in Amazon Aurora nella documentazione di Aurora
   - Per Amazon RDS, consulta quanto segue:
     - Utilizzo della replica MySQL nella documentazione di Amazon RDS
     - Utilizzo della replica MariadB nella documentazione di Amazon RDS
   - Per Amazon EC2, consulta quanto segue:
     - Configurazione della replica basata sulla posizione dei file di log binari nella documentazione MySQL
     - Configurazione delle repliche nella documentazione MySQL
     - Configurazione della replica nella documentazione di MariadB

Vantaggi

- MyDumper supporta il parallelismo utilizzando il multithreading, che migliora la velocità delle operazioni di backup e ripristino.
- MyDumper evita costose routine di conversione dei set di caratteri, il che contribuisce a garantire l'elevata efficienza del codice.
- MyDumper semplifica la visualizzazione e l'analisi dei dati utilizzando il dumping di file separati per tabelle e metadati.
- MyDumper mantiene le istantanee su tutti i thread e fornisce posizioni accurate dei log primari e secondari.
- È possibile utilizzare Perl Compatible Regular Expressions (PCRE) per specificare se includere o escludere tabelle o database.

Limitazioni

- È possibile scegliere uno strumento diverso se i processi di trasformazione dei dati richiedono file di dump intermedi in formato flat anziché in formato SQL.
- myloader non importa automaticamente gli account utente del database. Se stai ripristinando il backup su Amazon RDS o Aurora, ricrea gli utenti con le autorizzazioni richieste. Per ulteriori informazioni, consulta Privilegi dell'account utente principale nella documentazione di Amazon RDS. Se stai ripristinando il backup su un'istanza di database Amazon EC2, puoi esportare manualmente gli account utente del database di origine e importarli nell'istanza EC2.

Best practice

- Configura MyDumper per dividere ogni tabella in segmenti, ad esempio 10.000 righe in ogni segmento, e scrivere ogni segmento in un file separato. In questo modo è possibile importare i dati in parallelo in un secondo momento.
- Se si utilizza il motore InnoDB, utilizzare l'`--trx-consistency-only`opzione per ridurre al minimo il blocco.
- L'utilizzo MyDumper per esportare il database può richiedere molte letture e il processo può influire sulle prestazioni complessive del database di produzione. Se disponi di un'istanza di database di replica, esegui il processo di esportazione dalla replica. Prima di eseguire l'esportazione dalla

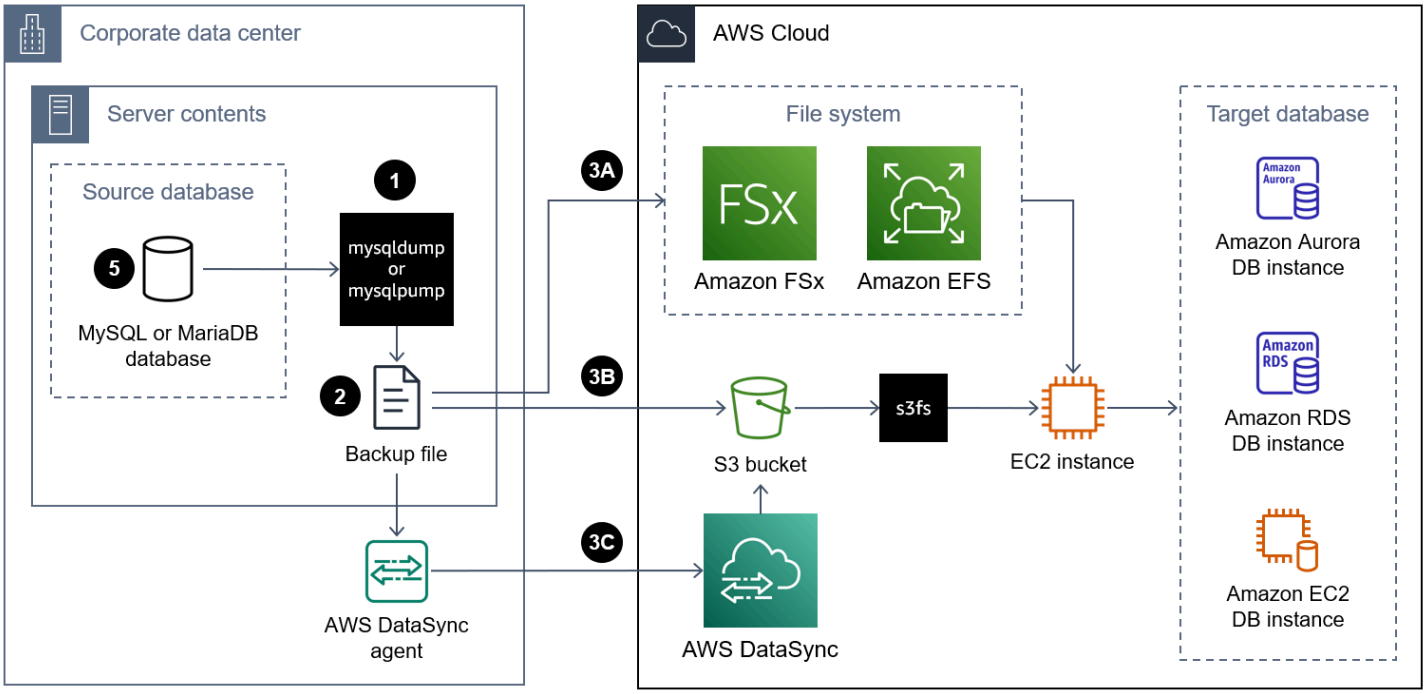
replica, interrompi il thread SQL di replica. Ciò consente di velocizzare l'esecuzione del processo di esportazione.

- Non esportare il database durante le ore lavorative di punta. Evitare le ore di punta può stabilizzare le prestazioni del database di produzione principale durante l'esportazione del database.
- Amazon RDS for MySQL non supporta il plug-in. `keyring_aws` Per ulteriori informazioni, consulta Problemi noti e limitazioni. Per migrare le tabelle crittografate locali all'istanza Amazon RDS, negli script di backup, è necessario rimuovere `ENCRYPTION` o `DEFAULT ENCRYPTION` eliminare la sintassi. `CREATE TABLE` Per la crittografia a riposo, puoi usare una chiave (). AWS Key Management Service AWS KMS Per ulteriori informazioni, consulta Crittografia delle risorse Amazon RDS.

mysqldump e mysqlpump

mysqldump e mysqlpump sono strumenti nativi di backup del database per MySQL. MariaDB supporta mysqldump ma non supporta mysqlpump. Entrambi questi strumenti creano backup logici e fanno parte dei programmi client MySQL. mysqldump supporta l'elaborazione a thread singolo. mysqlpump supporta l'elaborazione parallela di database e oggetti all'interno dei database, per accelerare il processo di dump. È stato introdotto nella versione 5.7.8 di MySQL. mysqlpump è stato rimosso nella versione 8.4 di MySQL.

Il diagramma seguente mostra i passaggi di alto livello coinvolti nella migrazione di un database utilizzando un file di backup mysqldump o mysqlpump.



Di seguito sono riportati i passaggi per utilizzare mysqldump o mysqlpump per migrare un database verso: Cloud AWS

1. Installa MySQL Shell sul server locale. Per istruzioni, consulta Installazione di MySQL Shell nella documentazione di MySQL. Questo installa sia mysqldump che mysqlpump.
2. Usando mysqldump o mysqlpump, crea un backup del database di origine locale. Per istruzioni, consulta mysqldump e mysqlpump nella documentazione di MySQL, oppure vedi Fare backup con mysqldump nella documentazione di MariaDB. Per ulteriori informazioni su come richiamare i programmi MySQL e specificare le opzioni, vedere Uso dei programmi MySQL.
3. Sposta il file di backup in un'istanza EC2 utilizzando uno dei Cloud AWS seguenti approcci:

   Approccio 3A: monta un file system Amazon FSx o Amazon Elastic File System (Amazon EFS) sul server locale che esegue l'istanza di database. È possibile utilizzare AWS Direct Connect o Site-to-Site VPN per stabilire la connessione. È possibile eseguire il backup direttamente del database nella condivisione di file montata oppure eseguire il backup in due passaggi eseguendo il backup del database su un file system locale e quindi caricandolo sul volume FSx o EFS montato. Successivamente, monta il file system Amazon FSx o Amazon EFS, anch'esso montato sul server locale, su un'istanza EC2.

   Approccio 3B: utilizza l' AWS CLI AWS SDK o l'API REST di Amazon S3 per spostare direttamente il file di backup dal server locale a un bucket S3. Se il bucket S3 di destinazione si trova in un

Regione AWS ambiente lontano dal data center, puoi utilizzare Amazon S3 Transfer Acceleration per trasferire il file più rapidamente. Usa il file system s3fs-fuse per montare il bucket S3 sull'istanza EC2.

Approccio 3C: installa l' AWS DataSync agente nel data center locale, quindi utilizzalo AWS DataSyncper spostare il file di backup in un bucket Amazon S3. Usa il file system s3fs-fuse per montare il bucket S3 sull'istanza EC2.

> **Note**
>
> Puoi anche utilizzare Amazon S3 File Gateway per trasferire i file di backup del database di grandi dimensioni in un bucket S3 nel. Cloud AWS Per ulteriori informazioni sul tagging, consulta Utilizzo di Amazon S3 File Gateway per trasferire file di backupin questa guida.

4. Utilizza il metodo di ripristino nativo per ripristinare il backup sul database di destinazione. Per istruzioni, consulta Reloading SQL-Format Backups nella documentazione MySQL o consulta Restoring Data from Dump Files nella documentazione di MariadB.
5. (Facoltativo) È possibile impostare la replica tra il database di origine e l'istanza del database di destinazione. È possibile utilizzare la replica con log binario (binlog) per ridurre i tempi di inattività. Per ulteriori informazioni, consulta gli argomenti seguenti:
   - Impostazione della configurazione della sorgente di replica nella documentazione MySQL
   - Per Amazon Aurora, consulta quanto segue:
     - Sincronizzazione del cluster Amazon Aurora MySQL DB con il database MySQL utilizzando la replica nella documentazione di Aurora
     - Utilizzo della replica binlog in Amazon Aurora nella documentazione di Aurora
   - Per Amazon RDS, consulta quanto segue:
     - Utilizzo della replica MySQL nella documentazione di Amazon RDS
     - Utilizzo della replica MariadB nella documentazione di Amazon RDS
   - Per Amazon EC2, consulta quanto segue:
     - Configurazione della replica basata sulla posizione dei file di log binari nella documentazione MySQL
     - Configurazione delle repliche nella documentazione MySQL
     - Configurazione della replica nella documentazione di MariadB

Vantaggi

- mysqldump e mysqlpump sono inclusi nell'installazione di MySQL Server
- I file di backup generati da questi strumenti sono in un formato più leggibile.
- Prima di ripristinare il file di backup, è possibile modificare il file.sql risultante utilizzando un editor di testo standard.
- È possibile eseguire il backup di una tabella, di un database o anche di una particolare selezione di dati.
- mysqldump e mysqlpump sono indipendenti dall'architettura della macchina.

Limitazioni

- mysqldump è un processo di backup a thread singolo. Le prestazioni per l'esecuzione di un backup sono buone per i database di piccole dimensioni, ma possono diventare inefficienti quando le dimensioni del backup superano i 10 GB.
- I file di backup in formato logico sono voluminosi, soprattutto se salvati come testo, e spesso sono lenti da creare e ripristinare.
- Il ripristino dei dati può essere lento perché la riapplicazione delle istruzioni SQL nell'istanza DB di destinazione comporta un'elaborazione intensiva del disco I/O e della CPU per l'inserimento, la creazione di indici e l'applicazione dei vincoli di integrità referenziale.
- L'utilità mysqlpump non è supportata per le versioni di MySQL precedenti alla 5.7.8 o per le versioni 8.4 e successive.
- Per impostazione predefinita, mysqlpump non esegue un backup dei database di sistema, ad esempio o. `performance_schema` `sys` Per eseguire il backup di parte del database di sistema, denominalo esplicitamente nella riga di comando.
- mysqldump non esegue il backup delle istruzioni InnoDB. `CREATE TABLESPACE`

> **Note**
>
> I backup delle istruzioni CREATE TABLESPACE e dei database di sistema sono utili solo quando si ripristinano i backup del database MySQL o MariadB su un'istanza EC2. Questi backup non vengono utilizzati per Amazon RDS o Aurora.

Best practice

- Quando ripristini il backup del database, disabilita i controlli chiave, ad esempio a livello di sessione nel database di destinazione. `FOREIGN_KEY_CHECKS` Ciò aumenta la velocità di ripristino.
- Assicurati che l'utente del database disponga di privilegi sufficienti per creare e ripristinare il backup.

Backup diviso

Una strategia di backup suddiviso consiste nella migrazione di un server di database di grandi dimensioni dividendo il backup in più parti. È possibile utilizzare approcci diversi per migrare ogni parte del backup. Questa può essere l'opzione migliore per i seguenti casi d'uso:

- Server di database di grandi dimensioni ma database singoli di piccole dimensioni: questo è un buon approccio quando la dimensione del server di database totale è di più TB ma la dimensione di ogni singolo database utente indipendente è inferiore a 1 TB. Per ridurre il periodo di migrazione complessivo, è possibile migrare i singoli database separatamente e in parallelo.

  Usiamo un esempio di un server di database locale da 2 TB. Questo server è composto da quattro database da 0,5 TB ciascuno. È possibile eseguire i backup di ogni singolo database separatamente. Quando si ripristina il backup, è possibile ripristinare tutti i database su un'istanza in parallelo oppure, se i database sono indipendenti, è possibile ripristinare ogni backup su un'istanza separata. È consigliabile ripristinare database indipendenti su istanze separate, anziché ripristinarli sulla stessa istanza. Per ulteriori informazioni, consulta le procedure consigliate in questa guida.
- Server di database di grandi dimensioni ma piccole tabelle di database individuali: questo è un buon approccio quando la dimensione del server di database totale è di più TB ma la dimensione di ogni tabella di database indipendente è inferiore a 1 TB. Per ridurre il periodo di migrazione complessivo, è possibile migrare le tabelle indipendenti singolarmente.

  Usiamo un esempio di database per utente singolo di 1 TB, l'unico database in un server di database locale. Il database contiene 10 tabelle, ognuna delle quali è di 100 GB. È possibile eseguire i backup di ogni singola tabella separatamente. Quando si ripristina il backup, è possibile ripristinare tutte le tabelle su un'istanza in parallelo.

- Un database contiene tabelle di carichi di lavoro transazionali e non transazionali: analogamente al caso d'uso precedente, è possibile utilizzare un approccio di backup suddiviso quando nello stesso database sono presenti tabelle di carichi di lavoro transazionali e non transazionali.

  Prendiamo un esempio di database da 2 TB composto da 0,5 TB di tabelle di carichi di lavoro critiche utilizzate per l'elaborazione delle transazioni online (OLTP) e una singola tabella da 1,5 TB utilizzata per l'archiviazione di vecchi dati. È possibile eseguire il backup di tutti gli oggetti del database ad eccezione della tabella di archiviazione come backup coerente e a transazione singola. Quindi, si esegue solo un altro backup separato della tabella di archiviazione. Per il backup della tabella di archiviazione, puoi anche prendere in considerazione l'idea di eseguire più backup paralleli utilizzando condizioni per dividere il numero di righe nel file di backup. Di seguito è riportato un esempio:

  ```
  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
  ```

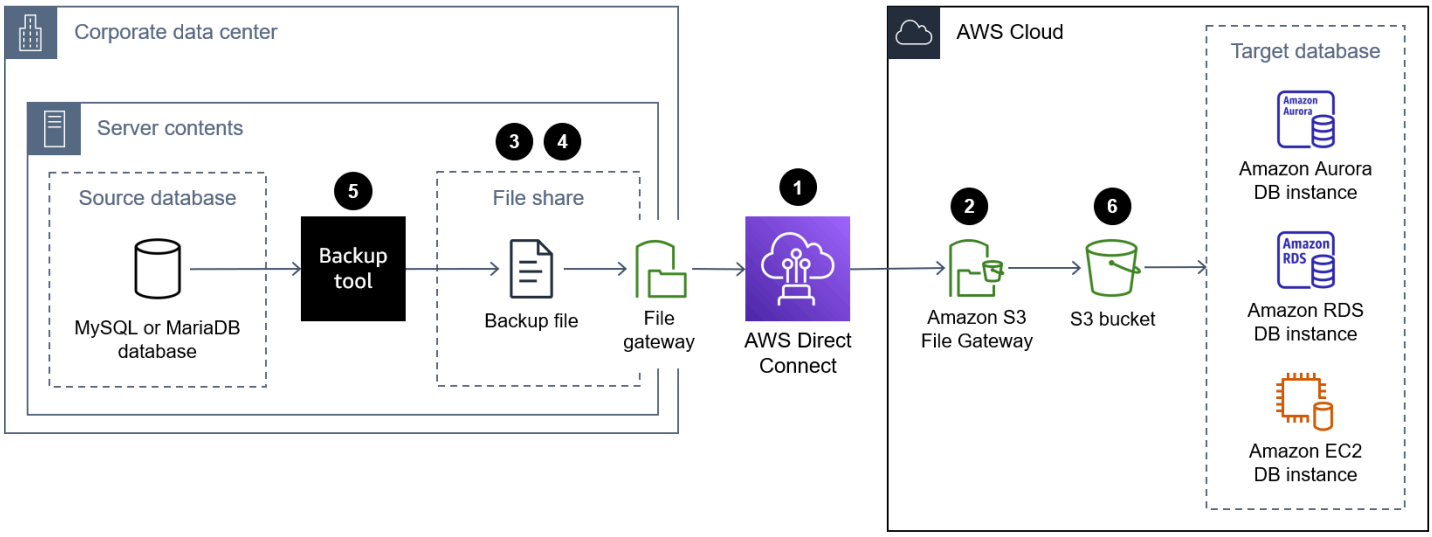
  Quando si ripristinano i file di backup, è possibile ripristinare il backup del carico di lavoro transazionale e il backup della tabella di archiviazione in parallelo.
- Limitazioni delle risorse di elaborazione: se nel server locale sono disponibili risorse di elaborazione limitate, ad esempio CPU, memoria o disco I/O, ciò può influire sulla stabilità e sulle prestazioni durante l'esecuzione del backup. Invece di eseguire un backup completo, puoi dividerlo in parti.

  Ad esempio, un server di produzione locale potrebbe essere sovraccarico di carichi di lavoro e disporre di risorse CPU limitate. Se si esegue un backup in un'unica esecuzione di un database di più terabyte su questo server, può consumare risorse CPU aggiuntive e influire negativamente sul server di produzione. Invece di eseguire il backup completo del database, suddividetelo in più parti, ad esempio 2-3 tabelle ciascuna.

Utilizzo di Amazon S3 File Gateway per trasferire file di backup

Amazon S3 File Gateway collega l'ambiente locale ad Amazon Simple Storage Service (Amazon S3) tramite un'interfaccia di file in modo da poter archiviare e recuperare oggetti Amazon S3 utilizzando protocolli di file standard del settore, come Network File System (NFS) e Server Message Block (SMB). È progettato per essere una soluzione economica e scalabile per l'archiviazione dei dati nel cloud. Poiché è possibile utilizzarlo per archiviare i file di backup del database, questo servizio può aiutarti a migrare database locali di grandi dimensioni verso. Cloud AWS Ad esempio, puoi utilizzare Amazon S3 File Gateway e il tuo strumento di backup del database preferito per eseguire il backup del grande database MySQL o MariaDB direttamente su un bucket Amazon S3. Puoi quindi montare il bucket S3 sull'istanza di destinazione e ripristinare il backup.

Il diagramma seguente mostra i passaggi di alto livello necessari quando si utilizza Amazon S3 File Gateway per trasferire il file di backup per un database locale in un bucket S3 nel. Cloud AWS



Di seguito sono riportati i passaggi per utilizzare Amazon S3 File Gateway per trasferire un file di backup del database da un data center locale a un bucket S3 in: Cloud AWS

1. Connect il data center locale a Cloud AWS tramite un servizio come AWS Direct Connect AWS Site-to-Site VPN oppure utilizzando una connessione Internet pubblica.
2. Crea un S3 File Gateway. Per istruzioni, consulta Creazione del gateway.

3. Crea una condivisione di file NFS o SMB ospitata da S3 File Gateway. Per istruzioni, consulta Creare una condivisione di file.
4. Monta la condivisione di file NFS o SMB sul server locale che ospita il tuo database MySQL o MariaDB. Per istruzioni, consulta Montare e utilizzare la condivisione di file.
5. Esegui il backup del database MySQL o MariaDB locale nella directory in cui è montata la condivisione di file NFS. È possibile utilizzare uno qualsiasi degli strumenti di backup descritti in questa guida.
6. Ripristina il backup del database sull'istanza del database di destinazione utilizzando uno degli approcci descritti in questa guida.

Vantaggi

- Producendo i backup del database direttamente nel bucket S3 e ripristinando il backup sull'istanza DB di destinazione direttamente dallo stesso bucket S3, è possibile accelerare in modo significativo il processo di migrazione end-to-end.
- I file di backup del database vengono archiviati in modo duraturo in Amazon S3 e puoi scegliere la politica di gestione del ciclo di vita e la classe di storage S3.

Limitazioni

Di seguito sono riportate le limitazioni relative all'utilizzo delle condivisioni di file Amazon S3 File Gateway:

- Il numero massimo di condivisioni di file per gateway è 50.
- Per evitare conflitti di lettura e scrittura quando più condivisioni di file utilizzano lo stesso bucket S3, è necessario configurare ogni condivisione di file in modo che utilizzi un nome di prefisso univoco.
- La dimensione massima di un singolo file è di 5 TB, che è la dimensione massima di ogni singolo oggetto in Amazon S3.
- La lunghezza massima del percorso è di 1024 caratteri.
- Gli ACL di Windows sono supportati solo sulle condivisioni di file abilitate per Active Directory quando si utilizzano client Windows SMB per accedere alle condivisioni di file.
- Amazon S3 File Gateway supporta un massimo di 10 voci ACL per ogni file e directory.
- Le impostazioni ACL principali delle condivisioni di file SMB si trovano solo sul gateway. Queste impostazioni sono permanenti anche dopo gli aggiornamenti e i riavvii del gateway.

> **Note**
>
> Se configuri gli ACL nella cartella principale anziché nella cartella principale, le autorizzazioni ACL non sono persistenti in Amazon S3.

Best practice

Per ulteriori informazioni sulle best practice per Amazon S3 File Gateway, consulta Best practice nella documentazione di S3 File Gateway.

Le migliori pratiche per la migrazione di database MySQL e MariaDB di grandi dimensioni

Oltre alle best practice specifiche degli strumenti elencate per ciascuna opzione di migrazione, consulta le seguenti best practice generali. Queste best practice si applicano alla migrazione di database MySQL e MariaDB di grandi dimensioni da più terabyte, indipendentemente dallo strumento selezionato:

- Assicurati che ci sia spazio sufficiente sui database di origine e di destinazione per eseguire e ripristinare il backup.
- Non creare indici secondari sull'istanza del database di destinazione fino al completamento della migrazione. Gli indici secondari aggiungono ulteriore sovraccarico di manutenzione durante l'importazione e possono rallentare il processo di importazione.
- Se utilizzi un approccio multithread, scegli il numero corretto di thread. Per l'esportazione, ti consigliamo di utilizzare un thread per ogni core della CPU. Per l'importazione, ti consigliamo di utilizzare un thread ogni due core della CPU.
- I dump dei dati vengono spesso eseguiti da server di database attivi che fanno parte di un ambiente di produzione di importanza critica. Se il dump dei dati influisce gravemente sulle prestazioni e ciò non è accettabile nel tuo ambiente, prendi in considerazione una delle seguenti soluzioni:
  - Il server di origine dispone di repliche, è possibile eseguire il dump dei dati da una delle repliche.
  - Il server di origine è coperto da procedure di backup regolari:
    - Se il formato di backup è adatto per l'importazione diretta nel database di destinazione, utilizzate i dati di backup come input per il processo di importazione.
    - Se il formato di backup non è adatto per l'importazione diretta nel database di destinazione, utilizza il backup per fornire un database temporaneo e scaricare i dati da esso.
  - Se le repliche e i backup non sono disponibili:
    - Esegui i dump durante le ore non di punta, quando il traffico di produzione è al minimo.
    - Riduci la concomitanza delle operazioni di dump in modo che il server disponga di una capacità di riserva sufficiente per gestire il traffico di produzione.
- Crea dump solo di database creati dall'utente.
- Ricrea gli utenti nel database di destinazione e configura le loro autorizzazioni. Per ulteriori informazioni, consulta Gestione delle identità e degli accessi per Amazon RDS, Gestione delle

identità e degli accessi per Amazon Aurora o Gestione delle identità e degli accessi per Amazon EC2.

- Quando esegui la migrazione di un server di database di grandi dimensioni composto da più database indipendenti, crea un'istanza separata per ogni database. Ciò consente di gestire il database in modo più efficiente e può migliorare l'approvvigionamento delle risorse, mentre le risorse di elaborazione separate possono migliorare le prestazioni del database.

Risorse

AWS Linee guida prescrittive

- Guida al portfolio per migrazioni di grandi dimensioni AWS
- Strategia di migrazione per database relazionali
- Esegui la migrazione di un database MySQL locale su Amazon RDS for MySQL
- Configura la replica dei dati tra Amazon RDS for MySQL e MySQL su Amazon utilizzando GTID EC2
- Esegui la migrazione di un database MariaDB locale su Amazon RDS for MariaDB utilizzando strumenti nativi

AWS post di blog

- Best practice di sicurezza per Amazon RDS per istanze MySQL e MariaDB
- Migrazione di MariadB autogestito su Amazon Aurora MySQL

Risorse per il ripristino del backup

- Creazione di un bucket (documentazione Amazon S3)
- Connessione alla tua istanza Linux tramite SSH (EC2 documentazione Amazon)
- Configurazione di AWS CLI(documentazione)AWS CLI
- comando sync (AWS CLI Command Reference)
- Creazione di una policy IAM per accedere alle risorse Amazon S3 (documentazione Aurora)
- Prerequisiti del cluster DB (documentazione Aurora)
- Utilizzo dei gruppi di sottoreti DB (documentazione Aurora)
- Creazione di un gruppo di sicurezza VPC per un cluster DB privato (documentazione Aurora)
- Ripristino di un cluster Aurora MySQL DB da un bucket Amazon S3 (documentazione Aurora)
- Configurazione della replica con MySQL o un altro cluster Aurora DB (documentazione Aurora)
- Procedura rds_set_external_master (documentazione Amazon RDS)
- procedura rds_start_replication (documentazione Amazon RDS)

AWS marketing

- Amazon Aurora
- Amazon RDS per MariaDB
- Amazon RDS per MySQL
- Gateway di file Amazon S3

Altre risorse

- Percona XtraBackup
- MyDumper
- dump mysql
- pompa mysql

Cronologia dei documenti

La tabella seguente descrive le modifiche significative apportate a questa guida. Per ricevere notifiche sugli aggiornamenti futuri, puoi abbonarti a un feed RSS.

Modifica	Descrizione	Data
disponibilità di mysqlpump	mysqlpump è stato rimosso nella versione 8.4 di MySQL. Abbiamo aggiornato la sezione mysqldump e mysqlpump per riflettere questa modifica della disponibilità.	21 novembre 2024
MyDumper migliori pratiche	Abbiamo aggiornato le migliori pratiche per MyDumper aggiungere informazioni sulla migrazione delle tabelle di database crittografate.	24 ottobre 2024
Versioni Percona XtraBackup	Nella XtraBackup sezione Percona, abbiamo aggiornat o le istruzioni per riflettere le versioni di Percona XtraBacku p supportate da Amazon Aurora MySQL e Amazon RDS.	3 agosto 2023
Pubblicazione iniziale	—	6 aprile 2023

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.