大規模なマルチテラバイト MySQL または MariaDB データベースの への移行 AWS

AWS 規範ガイダンス

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

AWS 規範ガイダンス: 大規模なマルチテラバイト MySQL または MariaDB データベースの への移行 AWS

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

Amazon の商標およびトレードドレスは Amazon 以外の製品およびサービスに使用することはできません。また、お客様に誤解を与える可能性がある形式で、または Amazon の信用を損なう形式で使用することもできません。Amazon が所有していないその他のすべての商標は Amazon との提携、関連、支援関係の有無にかかわらず、それら該当する所有者の資産です。

Table of Contents

序章 1
対象者 2
ターゲットを絞ったビジネス成果 2
移行オプション 3
Percona XtraBackup 4
利点 6
制限事項 7
ベストプラクティス 7
MyDumper 8
利点 11
制限事項 11
ベストプラクティス 11
mysqldump と mysqlpump 12
利点 15
制限事項 15
ベストプラクティス 16
分割バックアップ 16
Amazon S3 ファイルゲートウェイ 18
利点 19
制限事項 19
ベストプラクティス 20
ベストプラクティス 21
リソース 23
ドキュメント履歴 25
..... xxvi

大規模なマルチテラバイト MySQL または MariaDB データベースの への移行 AWS

Babaiah Valluru および Ankur Bhanawat、Amazon Web Services (AWS)

2024 年 11 月 (ドキュメント履歴)

オンプレミスの MySQL および MariaDB データベースサーバーを使用している組織の多くが、データベースワークロードを AWS クラウドに移行することに関心を持っています。多くは Amazon Relational Database Service (Amazon RDS) for MariaDB、Amazon RDS for MySQL、または Amazon Aurora MySQL 互換エディションを選択します。Amazon RDS は、クラウドでリレーショナルデータベースを簡単にセットアップ、運用、スケーリングできるように設計されています。Amazon Aurora は Amazon RDS の一部であり、組み込みのセキュリティ、継続的バックアップ、サーバーフリーコンピューティング、最大 15 個のリードレプリカ、自動マルチリージョンレプリケーション、および他の AWS のサービスとの統合を提供します。

これらのいずれかへの移行 AWS のサービス には多くの利点がありますが、データベースの移行は、データベース管理者が実行する必要がある最も時間のかかる重要なタスクの 1 つです。大規模なデータベースを移行し、移行後のワークロードが同等またはそれ以上に改善されたパフォーマンスを示すことを確実にするには、正確な計画と実装が要求されます。このガイドで扱う大規模データベースとは、単一のマルチテラバイトのデータベースを指す場合も、数テラバイトのデータを含む多数の大規模データベースを指す場合もあります。適切な移行サービスとツールを選択することが、移行を成功させる鍵となります。データベースを移行するための一般的なアプローチとして、論理と物理の 2 種類があります。これらのアプローチの詳細については「MySQL」と「MariaDB」のドキュメントを参照してください。

このガイドでは、大規模なオンプレミスのマルチテラバイト MySQL および MariaDB データベースを、Amazon RDS for MariaDB、Amazon RDS for MySQL、または Amazon Aurora MySQL 互換エディションに移行するために使用できるさまざまなオープンソースまたはサードパーティーのツールについて説明します。このガイドで説明するオプションでは、論理的または物理的な移行アプローチを使用します。各オプションには、大容量のデータベースバックアップファイルをオンプレミスのデータセンターからクラウドに転送し、クラウド上でデータベースをバックアップファイルから復元するための複数のアプローチが含まれています。

対象者

このガイドは、MySQL または MariaDB データベースを AWS クラウドに移行しようと計画しているプログラムデータベース管理者、データベースエンジニア、移行エンジニア、プロジェクトマネージャー、運用管理者またはインフラ管理者を対象としています。

ターゲットを絞ったビジネス成果

このガイドの目標は、以下を支援することです。

- ユースケースと環境に最も適した大規模データベースの移行アプローチを選択します。
- 移行戦略に欠陥がある場合に発生する可能性のある遅延や財務上の損失を回避します。
- 各移行オプションの利点と制限事項について学びます。
- 大容量のデータベースバックアップファイルをオンプレミスデータセンターから AWS クラウドに転送するために使用できるさまざまなアプローチについて学びます。
- 大規模なデータベースを移行するための全体的なベストプラクティスを確認し、さらに、データベースをより効率的に移行することに役立つ各ツールのベストプラクティスも確認します。

大規模 MySQL データベースと MariaDB データベースの移行オプション

オンプレミスの MySQL または MariaDB データベースから Amazon Relational Database Service (Amazon RDS) または Amazon Aurora MySQL 互換エディションのデータベースインスタンスに移行するには、幅広い選択肢があります。移行を成功させるには、適切な移行アプローチとツールを選択することが不可欠です。このガイドでは、使いやすさ、データサイズ、ダウンタイムの要件に基づいて選択肢を評価できます。

以下に、マルチテラバイトのセルフマネージド MySQL データベースを Amazon RDS、Aurora、または Amazon Elastic Compute Cloud (Amazon EC2) データベースインスタンスに効率的に移行するために使用できる一般的な移行ツールとアプローチを示します。

- Percona XtraBackup (物理)
- MyDumper (論理)
- mysqldump と mysqlpump (論理)
- 分割バックアップ (物理、論理、またはその両方)

以下に、マルチテラバイトの MySQL 互換 (MariaDB など) データベースを Amazon RDS、Aurora、または Amazon EC2 データベースインスタンスに効率的に移行するために使用できる一般的な移行ツールとアプローチを示します。

- MyDumper (論理)
- mysqldump と mysqlpump (論理)
- 分割バックアップ (物理、論理、またはその両方)

移行ツールごとに、大容量のデータベースバックアップファイルを AWS クラウドに転送するために使用できるアプローチがいくつかあります。ツールごとにオプションが用意されているほか、Amazon S3 File Gateway を使用することもできます。詳細については、このガイドの「Amazon S3 File Gateway を使用したバックアップファイルの転送」を参照してください。

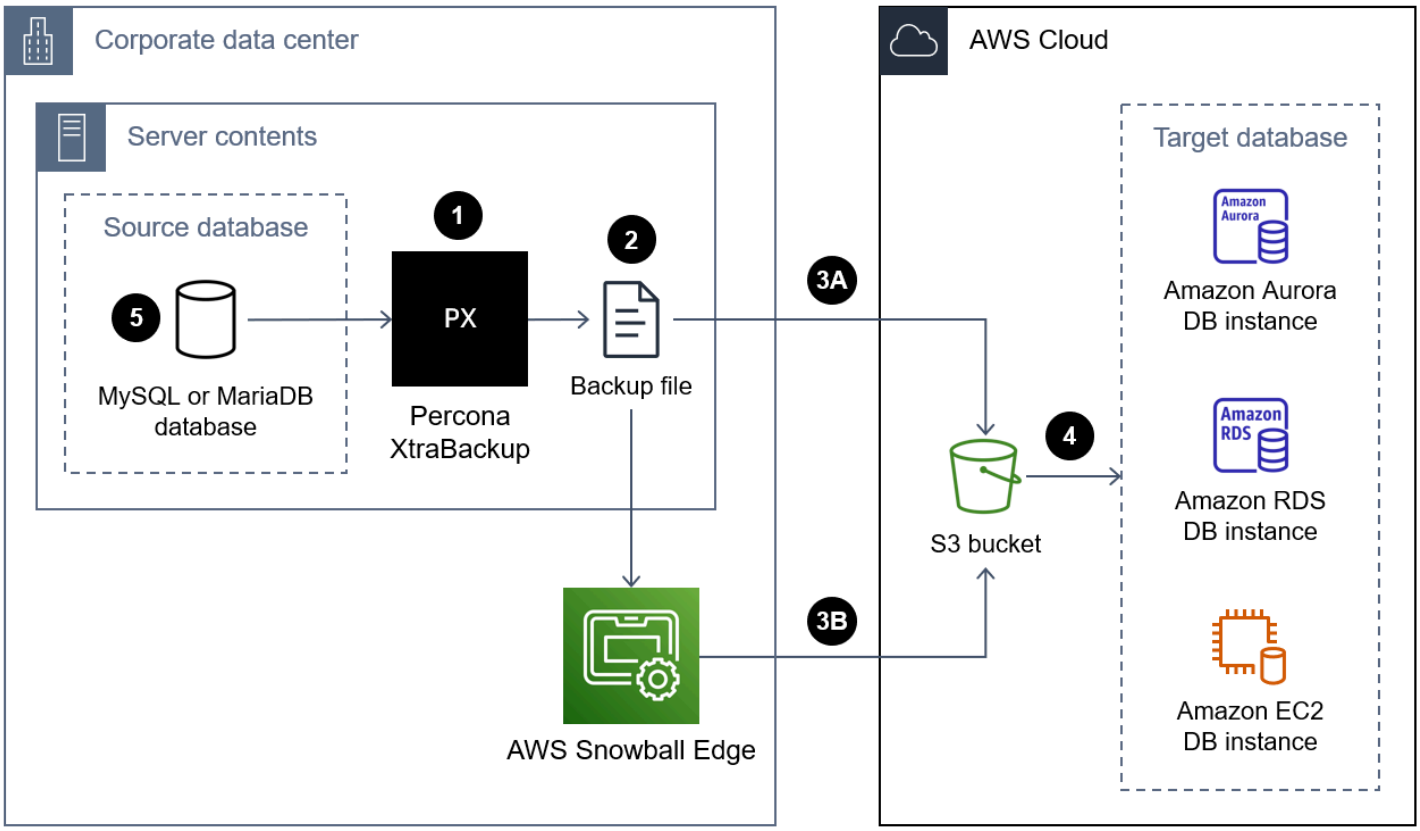
Percona XtraBackup

> **Important**
>
> Percona XtraBackup は MariaDB バージョン 10.3 以降ではサポートされておらず、バージョン 10.1 および 10.2 では部分的にのみサポートされています。

Percona XtraBackup は、MySQL および MariaDB 用の一般的なオープンソースのウォームバックアップソフトウェアで、InnoDB および XtraDB ストレージエンジンのノンブロッキングバックアップを作成します。MySQL または MariaDB サーバーで動作します。このツールとその機能や利点の詳細については、Percona XtraBackup ドキュメントの「Percona XtraBackup の概要」を参照してください。

このツールは、物理的な移行アプローチを使用します。MySQL または MariaDB のデータディレクトリとその中のファイルを直接コピーします。そのため、100 GB を超えるような大規模なデータベースでは、他のツールよりも復元時間を大幅に短縮することができます。オンプレミスのソースデータベースのバックアップを作成し、バックアップファイルをクラウドに移行してから、新しいターゲットデータベースインスタンスにバックアップを復元します。

次の図は、Percona XtraBackup バックアップファイルを使用したデータベースの移行に関する大まかな手順を示しています。バックアップファイルのサイズに応じて、 AWS クラウドの Amazon Simple Storage Service (Amazon S3) バケットにバックアップを転送するオプションが 2 つあります。



Percona XtraBackup を使用してデータベースを AWS クラウドに移行する手順は次のとおりです。

1. Percona XtraBackup をオンプレミスサーバーにインストールします。Amazon Aurora MySQL バージョン 2 または Amazon RDS を使用している場合は、「Installing Percona XtraBackup 2.4」を参照してください。Amazon Aurora MySQL バージョン 3 を使用している場合は、Percona XtraBackup ドキュメントの「Installing Percona XtraBackup 8.0」を参照してください。
2. ソース MySQL または MariaDB データベースのフルバックアップを作成します。Percona XtraBackup 2.4 の手順については、「完全バックアップ」を参照してください。Percona XtraBackup 8.0 の手順については、「完全バックアップの作成」を参照してください。
3. 次のような組織内の承認されたサービスまたはツールを使用して、バックアップファイルをインターネット経由で転送します。
   - AWS Site-to-Site VPN
   - AWS Client VPN
   - AWS Direct Connect
   - Amazon S3 File Gateway (詳細については、このガイドの「Amazon S3 File Gateway を使用したバックアップファイルの転送」を参照してください。)
   - AWS Command Line Interface (AWS CLI)

4. Amazon S3 バケットから、バックアップファイルをターゲットデータベースインスタンスに復元します。手順については、以下を参照してください。
   - Aurora MySQL 互換エディションについては、Amazon RDS ドキュメントの「Migrating data from MySQL by using an Amazon S3 bucket」を参照してください。
   - Amazon RDS for MySQL または Amazon EC2 については、「Importing data into a MySQL DB instance」を参照してください。
   - Amazon RDS for MariaDB または Amazon EC2 については、「Importing data into a MariaDB DB instance」を参照してください。
5. (オプション) ソースデータベースとターゲットデータベースインスタンス間のレプリケーションを設定できます。バイナリログ (binlog) レプリケーションを使用すると、ダウンタイムを短縮できます。詳細については次を参照してください:
   - MySQL ドキュメントの「Setting the Replication Source Configuration」
   - Amazon Aurora については、以下を参照してください。
     - Aurora ドキュメントの「Synchronizing the Amazon Aurora MySQL DB cluster with the MySQL database using replication」
     - Aurora ドキュメントの「Using binlog replication in Amazon Aurora」
   - Amazon RDS については、以下を参照してください。
     - Amazon RDS ドキュメントの「MySQL のレプリケーションの使用」
     - Amazon RDS ドキュメントの「MariaDB のレプリケーションの使用」
   - Amazon EC2 については、以下を参照してください。
     - MySQL ドキュメントの「Setting Up Binary Log File Position Based Replication」
     - MySQL ドキュメントの「Setting Up Replicas」
     - MariaDB ドキュメントの「Setting Up Replication」

利点

- Percona XtraBackup は物理的な移行アプローチを使用するため、復元プロセスは通常、論理的な移行アプローチを使用するツールよりも高速になります。これは、パフォーマンスの制限がデータ処理に必要なコンピューティングリソースによるものではなく、ディスクまたはネットワークのスループットによるためです。
- 復元プロセスでは S3 バケットからターゲットデータベースインスタンスへファイルが直接コピーされるため、Percona XtraBackup ファイルは通常、他のツールで作成されたバックアップファイルよりも短時間で復元されます。

- Percona XtraBackup は順応性があります。例えば、ファイルをすばやくコピーできるように複数のスレッドをサポートし、バックアップサイズを減らすための圧縮をサポートしています。

制限事項

- Percona XtraBackup はソースデータベースサーバーにアクセスできる必要があるため、オフラインバックアップはできません。
- Percona XtraBackup は、同一のシステムアーキテクチャを持つシステムでのみ使用できます。例えば、Intel for Windows Server で実行されているソースデータベースのバックアップを、ARM for Linux のターゲットサーバーに復元することはできません。
- Percona XtraBackup は MariaDB バージョン 10.3 以降ではサポートされておらず、MariaDB バージョン 10.2 およびバージョン 10.1 では部分的にのみサポートされています。詳細については、MariaDB ナレッジベースの「Percona XtraBackup Overview: Compatibility with MariaDB」を参照してください。
- Percona XtraBackup を使用して、ソース MariaDB データベースを Amazon RDS for MySQL や Aurora MySQL 互換などのターゲット MySQL データベースインスタンスに復元することはできません。
- S3 バケットに保存できるデータの合計量とオブジェクトの数は無制限ですが、最大ファイルサイズは 5 TB です。バックアップファイルが 5 TB を超える場合は、複数の小さいファイルに分割できます。
- `innodb_file_per_table` 設定がオフの場合、Percona XtraBackup は、`--tables`、`--tables-exclude`、`--tables-file`、`--databases`、`--databases-exclude`、または `--databases-file` を使用する部分バックアップをサポートしていません。Percona XtraBackup バージョン 2.4 の詳細については、「部分バックアップ」を参照してください。Percona XtraBackup バージョン 8.0 の詳細については、「部分バックアップの作成」を参照してください。

ベストプラクティス

- バックアッププロセスのパフォーマンスを向上させるには、以下を実行します。
  - --parallel=<threads> を使用して複数のファイルを並行してコピーする
  - --compress-threads=<threads> を使用して複数のファイルを並行して圧縮する
  - --use-memory=<size> を使用してメモリを増やす
  - --encrypt-threads=<threads> を使用して複数のファイルを並行して暗号化する

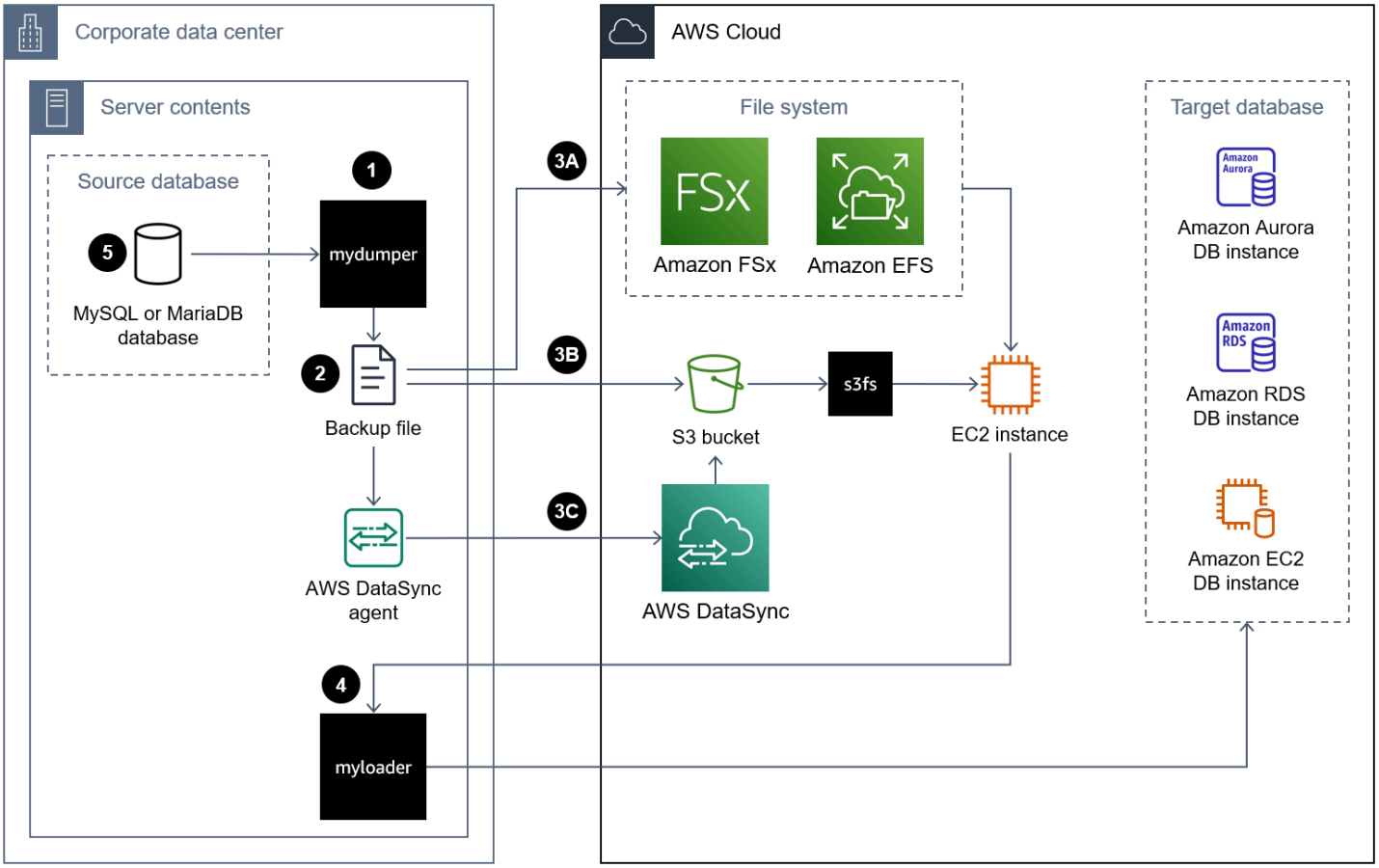
- ソースサーバーに、データベースバックアップファイルを取得するのに十分なスペースがあることを確認します。
- Percona xbstream (.xbstream) 形式のファイルを使用してデータベースバックアップを生成します。詳細については、Percona XtraBackup ドキュメントの「The xbstream binary overview」を参照してください。

MyDumper

MyDumper (GitHub) は、2 つのユーティリティで構成されるオープンソースの論理移行ツールです。

- MyDumper は、MySQL データベースの整合性のあるバックアップをエクスポートします。複数の並列スレッドを使用したデータベースのバックアップが可能で、使用可能な CPU コアごとに最大 1 つのスレッドをサポートします。
- myloader は、MyDumper によって作成されたバックアップファイルを読み取り、ターゲットデータベースインスタンスに接続してから、データベースを復元します。

次の図は、MyDumper バックアップファイルを使用したデータベースの移行に関する大まかな手順を示しています。このアーキテクチャ図には、バックアップファイルをオンプレミスデータセンターから AWS クラウドの EC2 インスタンスに移行するための 3 つのオプションが含まれています。



MyDumper を使用してデータベースを AWS クラウドに移行する手順は次のとおりです。

1. MyDumper と myloader をインストールします。手順については、「How to install mydumper/myloader」(GitHub) を参照してください。
2. MyDumper を使用して、ソース MySQL または MariaDB データベースのバックアップを作成します。手順については、「MyDumper の使用方法」を参照してください。
3. 次のいずれかの方法 AWS クラウド を使用して、バックアップファイルを の EC2 インスタンスに移動します。

   アプローチ 3A – Amazon FSx または Amazon Elastic File System (Amazon EFS) ファイルシステムを、データベースインスタンスを実行するオンプレミスサーバーにマウントします。 AWS Direct Connect または を使用して接続 Site-to-Site VPN を確立できます。データベースをマウントされたファイル共有に直接バックアップすることも、データベースをローカルファイルシステムにバックアップしてからマウントされた FSx または EFS ボリュームにアップロードする 2 段階の手順でバックアップを実行することもできます。次に、オンプレミスサーバーにマウントされている Amazon FSx または Amazon EFS ファイルシステムを EC2 インスタンスにマウントします。

アプローチ 3B – AWS CLI、 AWS SDK、または Amazon S3 REST API を使用して、バックアップファイルをオンプレミスサーバーから S3 バケットに直接移動します。ターゲット S3 バケットがデータセンターから AWS リージョン 離れた にある場合は、Amazon S3 Transfer Acceleration を使用してファイルをより迅速に転送できます。s3fs-fuse ファイルシステムを使用して、EC2 インスタンスに S3 バケットをマウントします。

アプローチ 3C – オンプレミスデータセンターに AWS DataSync エージェントをインストールし、AWS DataSync を使用してバックアップファイルを Amazon S3 バケットに移動します。s3fs-fuse ファイルシステムを使用して、EC2 インスタンスに S3 バケットをマウントします。

> **Note**
>
> Amazon S3 File Gateway を使用して、大容量のデータベースバックアップファイルを AWS クラウドの S3 バケットに転送することもできます。詳細については、このガイドの「Amazon S3 File Gateway を使用したバックアップファイルの転送」を参照してください。

4. myloader を使用して、ターゲットデータベースインスタンスのバックアップを復元します。手順については、「myloader usage」(GitHub) を参照してください。
5. (オプション) ソースデータベースとターゲットデータベースインスタンス間のレプリケーションを設定できます。バイナリログ (binlog) レプリケーションを使用すると、ダウンタイムを短縮できます。詳細については次を参照してください:
   - MySQL ドキュメントの「Setting the Replication Source Configuration」
   - Amazon Aurora については、以下を参照してください。
     - Aurora ドキュメントの「Synchronizing the Amazon Aurora MySQL DB cluster with the MySQL database using replication」
     - Aurora ドキュメントの「Using binlog replication in Amazon Aurora」
   - Amazon RDS については、以下を参照してください。
     - Amazon RDS ドキュメントの「MySQL のレプリケーションの使用」
     - Amazon RDS ドキュメントの「MariaDB のレプリケーションの使用」
   - Amazon EC2 については、以下を参照してください。
     - MySQL ドキュメントの「Setting Up Binary Log File Position Based Replication」
     - MySQL ドキュメントの「Setting Up Replicas」

- MariaDB ドキュメントの「Setting Up Replication」

利点

- MyDumper はマルチスレッドを使用して並列処理に対応しているため、バックアップと復元オペレーションの速度が向上します。
- MyDumper はコストのかかる文字セットの変換ルーチンを回避し、コードの効率を高めるのに役立ちます。
- MyDumper は、テーブルとメタデータで別々のファイルをダンプすることで、データビューと解析を簡素化します。
- MyDumper は、すべてのスレッドにわたってスナップショットを維持し、プライマリログとセカンダリログの正確な位置を提供します。
- Perl 互換正規表現 (PCRE) を使用して、テーブルまたはデータベースを含めるか除外するかを指定できます。

制限事項

- データ変換プロセスで SQL 形式ではなくフラット形式の中間ダンプファイルが必要な場合は、別のツールを選択することが考えられます。
- myloader はデータベースユーザーアカウントを自動的にインポートしません。Amazon RDS または Aurora にバックアップを復元する場合は、必要なアクセス許可を持つユーザーを再作成します。詳細については、Amazon RDS ドキュメントの「Master user account privileges」を参照してください。Amazon EC2 データベースインスタンスにバックアップを復元する場合は、ソースデータベースユーザーアカウントを手動でエクスポートしてから EC2 インスタンスにインポートできます。

ベストプラクティス

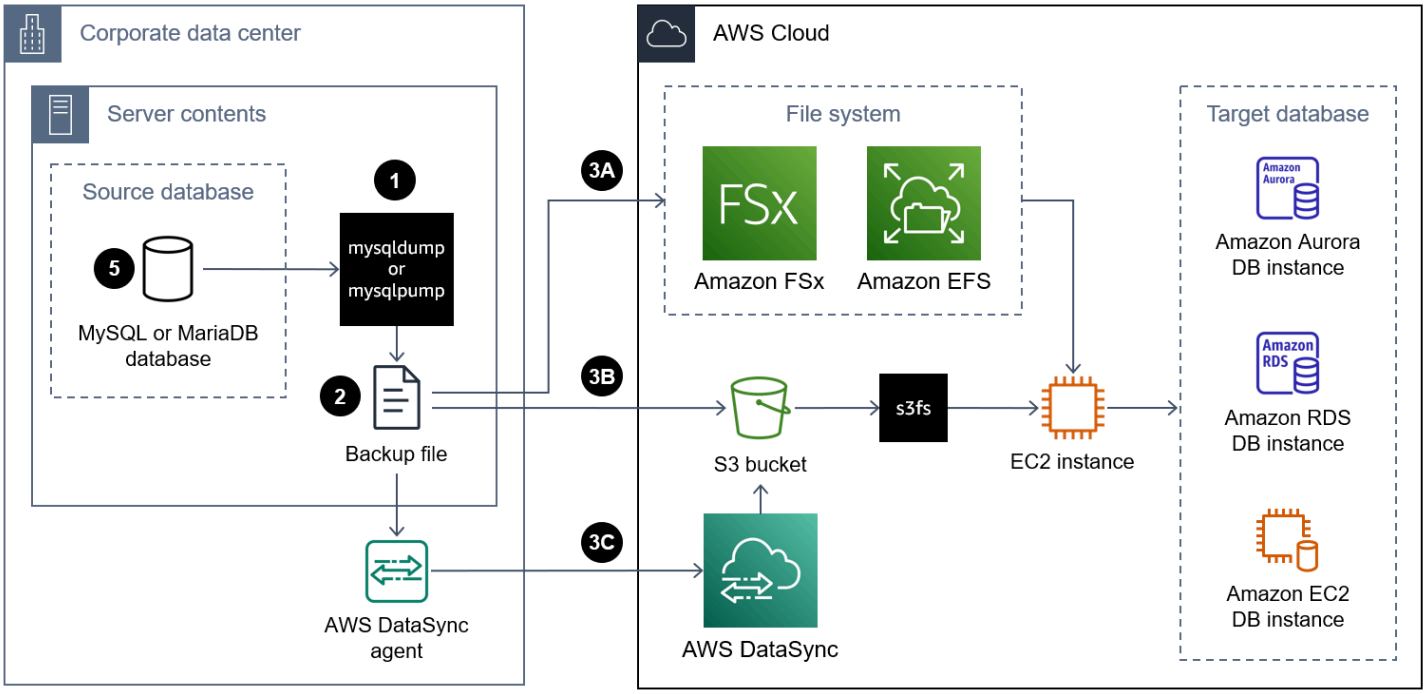
- 各テーブルをセグメントに分割し (セグメントごとに 10,000 行など)、各セグメントを個別のファイルに書き込むように MyDumper を設定します。これにより、後でデータを並行してインポートできます。
- InnoDB エンジンを使用している場合は、`--trx-consistency-only` オプションを使用してロックを最小限に抑えます。

- MyDumper を使用してデータベースをエクスポートすると、読み取り負荷が高まり、処理が本番データベースの全体的なパフォーマンスに影響を与える可能性があります。レプリカデータベースインスタンスがある場合は、レプリカからエクスポートプロセスを実行します。レプリカからエクスポートを実行する前に、レプリケーション SQL スレッドを停止します。これにより、エクスポートプロセスをより迅速に実行できます。
- 営業時間のピーク時にデータベースをエクスポートしないでください。ピーク時間を避けることで、データベースのエクスポート中に、プライマリ本番データベースのパフォーマンスを安定させることができます。
- Amazon RDS for MySQL は `keyring_aws` プラグインをサポートしていません。詳細については、「既知の問題と制限」を参照してください。オンプレミスの暗号化されたテーブルを Amazon RDS インスタンスに移行するには、バックアップスクリプトで、`CREATE TABLE` 構文から `ENCRYPTION` または `DEFAULT ENCRYPTION` を削除する必要があります。保存時の暗号化には、 AWS Key Management Service (AWS KMS) キーを使用してください。詳細については、「Amazon RDS リソースの暗号化」を参照してください。

mysqldump と mysqlpump

mysqldump と mysqlpump は、MySQL のネイティブデータベースバックアップツールです。MariaDB は mysqldump をサポートしていますが、mysqlpump はサポートしていません。これらのツールはいずれも論理バックアップを作成するもので、MySQL クライアントプログラムの一部です。mysqldump はシングルスレッド処理をサポートする一方、mysqlpump はデータベースとデータベース内のオブジェクトの並列処理をサポートし、ダンププロセスを高速化します。mysqlpump は MySQL バージョン 5.7.8 で導入されましたが、MySQL バージョン 8.4 で削除されました。

次の図は、mysqldump または mysqlpump バックアップファイルを使用したデータベースの移行に関する大まかな手順を示しています。



mysqldump または mysqlpump を使用してデータベースを AWS クラウドに移行する手順は次のとおりです。

1. オンプレミスサーバーに MySQL シェルをインストールします。手順については、MySQL ドキュメントの「MySQL Shell のインストール」を参照してください。これにより、mysqldump と mysqlpump の両方がインストールされます。
2. mysqldump または mysqlpump を使用して、ソースのオンプレミスデータベースのバックアップを作成します。手順については、MySQL ドキュメントの「mysqldump」と「mysqlpump」を参照するか、または、MariaDB ドキュメントの「Making Backups with mysqldump」を参照してください。MySQL プログラムを呼び出す方法とオプションを指定する方法の詳細については、「MySQLプログラムの使用」を参照してください。
3. 次のいずれかの方法 AWS クラウド を使用して、バックアップファイルを の EC2 インスタンスに移動します。

   アプローチ 3A – Amazon FSx または Amazon Elastic File System (Amazon EFS) ファイルシステムを、データベースインスタンスを実行するオンプレミスサーバーにマウントします。 AWS Direct Connect または を使用して接続 Site-to-Site VPN を確立できます。データベースをマウントされたファイル共有に直接バックアップすることも、データベースをローカルファイルシステムにバックアップしてからマウントされた FSx または EFS ボリュームにアップロードする 2 段階の手順でバックアップを実行することもできます。次に、オンプレミスサーバーにマウントさ

れている Amazon FSx または Amazon EFS ファイルシステムを EC2 インスタンスにマウントします。

アプローチ 3B – AWS CLI、 AWS SDK、または Amazon S3 REST API を使用して、バックアップファイルをオンプレミスサーバーから S3 バケットに直接移動します。ターゲット S3 バケットがデータセンターから AWS リージョン 離れた にある場合は、Amazon S3 Transfer Acceleration を使用してファイルをより迅速に転送できます。s3fs-fuse ファイルシステムを使用して、EC2 インスタンスに S3 バケットをマウントします。

アプローチ 3C – オンプレミスデータセンターに AWS DataSync エージェントをインストールし、AWS DataSync を使用してバックアップファイルを Amazon S3 バケットに移動します。s3fs-fuse ファイルシステムを使用して、EC2 インスタンスに S3 バケットをマウントします。

> **Note**
>
> Amazon S3 File Gateway を使用して、大容量のデータベースバックアップファイルを AWS クラウドの S3 バケットに転送することもできます。詳細については、このガイドの「Amazon S3 File Gateway を使用したバックアップファイルの転送」を参照してください。

4. ネイティブ復元メソッドを使用して、ターゲットデータベースのバックアップを復元します。手順については、MySQL ドキュメントの「Reloading SQL-Format Backups」または MariaDB ドキュメントの「Restoring Data from Dump Files」を参照してください。
5. (オプション) ソースデータベースとターゲットデータベースインスタンス間のレプリケーションを設定できます。バイナリログ (binlog) レプリケーションを使用すると、ダウンタイムを短縮できます。詳細については次を参照してください:
   - MySQL ドキュメントの「Setting the Replication Source Configuration」
   - Amazon Aurora については、以下を参照してください。
     - Aurora ドキュメントの「Synchronizing the Amazon Aurora MySQL DB cluster with the MySQL database using replication」
     - Aurora ドキュメントの「Using binlog replication in Amazon Aurora」
   - Amazon RDS については、以下を参照してください。
     - Amazon RDS ドキュメントの「MySQL のレプリケーションの使用」
     - Amazon RDS ドキュメントの「MariaDB のレプリケーションの使用」
   - Amazon EC2 については、以下を参照してください。

- MySQL ドキュメントの「Setting Up Binary Log File Position Based Replication」
- MySQL ドキュメントの「Setting Up Replicas」
- MariaDB ドキュメントの「Setting Up Replication」

利点

- mysqldump と mysqlpump は MySQL Server のインストールに含まれています。
- これらのツールによって生成されたバックアップファイルは、より読みやすい形式になります。
- 結果として得られる .sql ファイルは、バックアップファイルを復元する前に、標準のテキストエディタを使用して変更できます。
- 特定のテーブル、データベース、または特定のデータの選択をバックアップできます。
- mysqldump と mysqlpump はマシンアーキテクチャに依存しません。

制限事項

- mysqldump はシングルスレッドのバックアッププロセスです。バックアップ取得時のパフォーマンスは、小規模データベースには適していますが、バックアップサイズが 10 GB を超えると非効率になる可能性があります。
- 論理形式のバックアップファイルは、特にテキストとして保存された場合は大量であり、作成と復元に時間がかかることがよくあります。
- ターゲット DB インスタンスに SQL ステートメントを再適用するには、挿入、インデックス作成、参照整合性制約の適用のために大量のディスク I/O と CPU 処理が必要になるため、データの復元が遅くなる可能性があります。
- mysqlpump ユーティリティは、MySQL バージョン 5.7.8 以前またはバージョン 8.4 以降ではサポートされていません。
- デフォルトでは、mysqlpump は `performance_schema` や `sys` などのシステムデータベースのバックアップを取りません。システムデータベースの一部をバックアップするには、コマンドラインで明示的に名前を指定します。
- mysqldump は InnoDB `CREATE TABLESPACE` ステートメントをバックアップしません。

> **Note**
>
> CREATE TABLESPACE ステートメントとシステムデータベースのバックアップは、MySQL または MariaDB データベースのバックアップを EC2 インスタンスに復元する場合にのみ有用です。これらのバックアップは Amazon RDS または Aurora には使用されません。

ベストプラクティス

- データベースバックアップを復元する場合は、ターゲットデータベースのセッションレベルで、`FOREIGN_KEY_CHECKS` などのキーチェックを無効にします。これにより、復元速度が向上します。
- データベースユーザーにバックアップを作成および復元するための十分な権限があることを確認します。

分割バックアップ

分割バックアップ戦略とは、バックアップを複数のパーツに分割して大規模データベースサーバーを移行することをいいます。バックアップの各パーツを移行するには、さまざまな方法を使用できます。これは、次のユースケースに最適なオプションです。

- データベースサーバーは大きいが個々のデータベースは小さい – データベースサーバーの合計サイズは数 TB であるが、個々の独立したユーザーデータベースのサイズは 1 TB 未満の場合に適しています。移行期間全体を短縮するために、個々のデータベースを個別に移行することも並行して移行することもできます。

  オンプレミスの 2 TB データベースサーバーの例を見てみましょう。このサーバーは、それぞれ 0.5 TB の 4 つのデータベースで構成されています。個々のデータベースのバックアップは個別に作成できます。バックアップを復元するときは、インスタンス上のすべてのデータベースを並行して復元するか、データベースが独立している場合は、各バックアップを個別のインスタンスで復元できます。ベストプラクティスは、独立したデータベースを同じインスタンスで復元するのではなく、別のインスタンスで復元することです。詳細については、このガイドの「ベストプラクティス」を参照してください。
- データベースサーバーは大きいが個々のデータベーステーブルは小さい – データベースサーバーの合計サイズが複数 TB であるが、個々の独立したデータベーステーブルのサイズが 1 TB 未満の場合に適しています。移行期間全体を短縮するために、独立したテーブルを個別に移行できます。

1 TB の単一のユーザーデータベースが、オンプレミスデータベースサーバー内の唯一のデータベースである例を見てみましょう。データベースには 10 個のテーブルがあり、それぞれが 100 GB です。個々のテーブルのバックアップはそれぞれ個別に作成できます。バックアップを復元するときは、インスタンス上のすべてのテーブルを並行して復元できます。

- データベースにトランザクションワークロードテーブルと非トランザクションワークロードテーブルの両方が含まれている – 前のユースケースと同様に、同じデータベース内にトランザクションワークロードテーブルと非トランザクションワークロードテーブルの両方がある場合、分割バックアップアプローチを使用できます。

  オンライントランザクション処理 (OLTP) に使用される 0.5 TB の重要なワークロードテーブルと、古いデータのアーカイブに使用される単一の 1.5 TB テーブルで構成される、2 TB のデータベースの例を見てみましょう。アーカイブテーブルを除くすべてのデータベースオブジェクトのバックアップを、単一トランザクションの一貫性のあるバックアップとして取得できます。次に、もう 1 つアーカイブテーブルのみの個別のバックアップを取ります。アーカイブテーブルのバックアップでは、条件を使用してバックアップファイルの行数を分割することで、複数の並列バックアップを取ることも検討できます。以下に例を示します。

  ```
  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
  ```

  バックアップファイルを復元するときは、トランザクションワークロードのバックアップとアーカイブテーブルのバックアップを並行して復元できます。

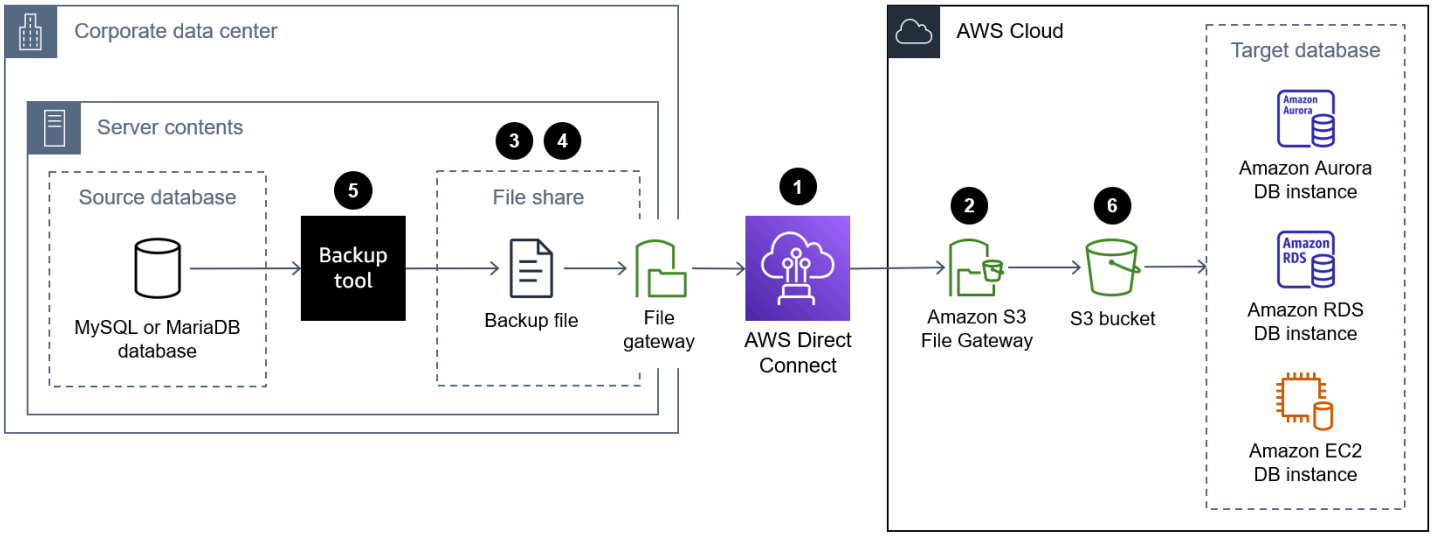
- コンピューティングリソースの制限事項 – CPU、メモリ、ディスク I/O など、オンプレミスサーバー内のコンピューティングリソースが限られている場合、バックアップを取る際の安定性とパフォーマンスが影響を受ける可能性があります。完全なバックアップを取る代わりに、パーツに分割できます。

  例えば、オンプレミスの本番稼働用サーバーで、ワークロードの負荷が大きく、CPU リソースが限られているとします。このサーバーでマルチテラバイトデータベースのバックアップを一度に実行すると、CPU リソースの消費量が増え、本番稼働用サーバーに悪影響を及ぼす可能性があります。データベース全体のバックアップを取る代わりに、テーブル 2 ~ 3 個ずつなど、複数のパーツに分割してバックアップを取ります。

Amazon S3 File Gateway を使用したバックアップファイルの転送

Amazon S3 File Gateway は、オンプレミス環境をファイルインターフェイスを介して Amazon Simple Storage Service (Amazon S3) に接続し、ネットワークファイルシステム (NFS) やサーバーメッセージブロック (SMB) などの業界標準のファイルプロトコルを使用して Amazon S3 オブジェクトを保存および取得できるようにします。クラウドにデータを保存するための費用対効果に優れたスケーラブルなソリューションとして設計されています。データベースバックアップファイルの保存に使用できるため、大規模なオンプレミスデータベースを AWS クラウドに移行するのに役立ちます。例えば、Amazon S3 File Gateway と任意のデータベースバックアップツールを使用して、大規模な MySQL または MariaDB データベースを Amazon S3 バケットに直接バックアップできます。その後、S3 バケットをターゲットインスタンスにマウントし、バックアップを復元できます。

次の図は、Amazon S3 File Gateway を使用してオンプレミスデータベースのバックアップファイルを AWS クラウドの S3 バケットに転送する際の大まかな手順を示しています。



次に、Amazon S3 File Gateway を使用して、オンプレミスデータセンターから AWS クラウドの S3 バケットにデータベースバックアップファイルを転送する手順を示します。

1. オンプレミスデータセンターを に接続するには、 AWS Direct Connect や などのサービス AWS クラウド を使用する AWS Site-to-Site VPN か、パブリックインターネット接続を使用します。
2. S3 ファイルゲートウェイの作成 手順については、「ゲートウェイを作成する」を参照してください。

3. S3 File Gateway によってホストされる NFS または SMB ファイル共有を作成します。手順については、「ファイル共有を作成する」を参照してください。
4. MySQL または MariaDB データベースをホストするオンプレミスサーバーに NFS または SMB ファイル共有をマウントします。手順については、「ファイル共有をマウントして使用する」を参照してください。
5. オンプレミスの MySQL または MariaDB データベースを、NFS ファイル共有がマウントされているディレクトリにバックアップします。このガイドで説明している任意のバックアップツールを使用できます。
6. このガイドで説明しているいずれかの方法を使用して、ターゲットデータベースインスタンスでデータベースバックアップを復元します。

利点

- データベースバックアップを S3 バケットで直接生成し、ターゲット DB インスタンスのバックアップを同じ S3 バケットから直接復元することで、エンドツーエンドの移行プロセスを大幅に高速化できます。
- データベースバックアップファイルは Amazon S3 に永続的に保存され、ユーザーはライフサイクル管理ポリシーと S3 ストレージクラスを選択します。

制限事項

Amazon S3 File Gateway ファイル共有を使用する場合の制限事項は次のとおりです。

- ゲートウェイごとのファイル共有の最大数は 50 です。
- 複数のファイル共有が同じ S3 バケットを使用する場合の読み取りと書き込みの競合を防ぐには、一意のプレフィックス名を使用するように各ファイル共有を設定する必要があります。
- 個々のファイルの最大サイズ (Amazon S3 の個々のオブジェクトの最大サイズ) は 5 TB です。
- 最大パス長は 1024 文字です。
- Windows ACL は、Windows SMB クライアントを使用してファイル共有にアクセスするときに、Active Directory が有効化されているファイル共有のみでサポートされています。
- Amazon S3 File Gateway では、ファイルとディレクトリごとに最大で 10 個の ACL エントリがサポートされています。
- SMB ファイル共有のルート ACL 設定は、ゲートウェイでのみ行われます。これらの設定は、ゲートウェイの更新や再起動を行っても保持されます。

> **Note**
>
> ルートの親フォルダではなくルートで ACL を設定した場合、ACL のアクセス許可は Amazon S3 では保持されません。

ベストプラクティス

Amazon S3 File Gateway のベストプラクティスの詳細については、S3 File Gateway ドキュメントの「ベストプラクティス」を参照してください。

大規模な MySQL データベースと MariaDB データベースを移行するためのベストプラクティス

移行オプションごとにリストアップされているツール固有のベストプラクティスに加えて、以下の一般的なベストプラクティスを確認してください。これらのベストプラクティスは、選択したツールに関係なく、大規模なマルチテラバイト MySQL および MariaDB データベースを移行する場合に適用されます。

- ソースデータベースと宛先データベースにバックアップを取得および復元するための十分なスペースがあることを確認します。
- 移行が完了するまで、ターゲットデータベースインスタンスにセカンダリインデックスを作成しないでください。セカンダリインデックスは、インポート中のメンテナンスオーバーヘッドを増加させ、インポートプロセスを遅らせる可能性があります。
- マルチスレッドアプローチを使用する場合は、適切な数のスレッドを選択します。エクスポートでは、CPU コアごとに 1 つのスレッドを使用することをお勧めします。インポートでは、2 つの CPU コアごとに 1 つのスレッドを使用することをお勧めします。
- データダンプは、ミッションクリティカルな本番環境の一部であるアクティブなデータベースサーバーから実行されることがよくあります。データダンプがパフォーマンスに大きく影響し、お使いの環境で許容できない場合は、次のいずれかを検討してください。
  - ソースサーバーにレプリカがある場合、レプリカの 1 つからデータをダンプできます。
  - ソースサーバーが通常のバックアップ手順の対象である場合:
    - バックアップ形式がターゲットデータベースへの直接インポートに適している場合は、バックアップデータをインポートプロセスの入力として使用します。
    - バックアップ形式がターゲットデータベースへの直接インポートに適していない場合は、バックアップを使用して一時データベースをプロビジョニングし、そこからデータをダンプします。
  - レプリカとバックアップが利用できない場合:
    - 本番トラフィックが最も低いオフピークの時間帯にダンプを実行します。
    - ダンプオペレーションの同時実行数を減らして、サーバーが本番トラフィックを処理するのに十分な予備容量を確保します。
- ユーザーが作成したデータベースのダンプのみを作成します。
- ターゲットデータベースでユーザーを再作成し、アクセス許可を設定します。詳細については、「Amazon RDS での Identity And Access Management」、「Amazon Aurora での Identity And

Access Management」、または「Amazon EC2 の Identity And Access Management」を参照してください。

- 複数の独立したデータベースで構成される大規模なデータベースサーバーを移行する場合は、データベースごとに個別のインスタンスを作成します。これにより、データベースをより効率的に管理し、リソースのプロビジョニングを改善し、個別のコンピューティングリソースによってデータベースのパフォーマンスを向上させることができます。

リソース

AWS 規範ガイダンス

- AWS の大規模な移行のためのポートフォリオプレイブック
- リレーショナルデータベースの移行戦略
- オンプレミス MySQL データベースを Amazon RDS for MySQL に移行する
- GTID を使用して Amazon RDS for MySQL と Amazon EC2 上の MySQL との間のデータレプリケーションを設定する
- ネイティブツールを使用して オンプレミスの MariaDB Amazon RDS for MariaDB に移行する

AWS ブログ記事

- Security best practices for Amazon RDS for MySQL and MariaDB instances
- Migrate self-managed MariaDB to Amazon Aurora MySQL

バックアップの復元に関するリソース

- 「バケットの作成」(Amazon S3 ドキュメント)
- Connecting to your Linux instance using SSH (Amazon EC2 ドキュメント)
- Configuring the AWS CLI (AWS CLI ドキュメント)
- 同期コマンド (AWS CLI コマンドリファレンス)
- 「Amazon S3 リソースにアクセスするための IAM ポリシーの作成」(Aurora ドキュメント)
- 「DB クラスターの前提条件」 (Aurora ドキュメント)
- 「DB サブネットグループの使用」(Aurora ドキュメント)
- Creating a VPC security group for a private DB cluster (Aurora ドキュメント)
- Restoring an Aurora MySQL DB cluster from an Amazon S3 bucket (Aurora ドキュメント)
- Setting up replication with MySQL or another Aurora DB cluster (Aurora ドキュメント)
- rds_set_external_master procedure (Amazon RDS ドキュメント)
- rds_start_replication procedure (Amazon RDS ドキュメント)

AWS マーケティング

- – Amazon Aurora
- Amazon RDS for MariaDB
- Amazon RDS for MySQL
- Amazon S3 File Gateway

その他のリソース

- Percona XtraBackup
- MyDumper
- mysqldump
- mysqlpump

ドキュメント履歴

以下の表は、本ガイドの重要な変更点について説明したものです。今後の更新に関する通知を受け取る場合は、RSS フィード をサブスクライブできます。

変更	説明	日付
mysqlpump の可用性	mysqlpump は MySQL バージョン 8.4 で削除されました。この可用性の変更を反映するために、mysqldump と mysqlpump セクションを更新しました。	2024 年 11 月 21 日
MyDumper のベストプラクティス	MyDumper のベストプラクティスを更新し、暗号化されたデータベーステーブルの移行に関する情報を追加しました。	2024 年 10 月 24 日
Percona XtraBackup のバージョン	Percona XtraBackup セクションで、Amazon Aurora MySQL と Amazon RDS でサポートされている Percona XtraBackup のバージョンを反映するために手順を更新しました。	2023 年 8 月 3 日
初版発行	—	2023 年 4 月 6 日

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。