将大型、多 TB 的 MySQL 或 MariaDB 数据库迁移到 AWS

AWS 规范性指导

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

AWS 规范性指导: 将大型、多 TB 的 MySQL 或 MariaDB 数据库迁移到 AWS

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

Amazon 的商标和商业外观不得用于任何非 Amazon 的商品或服务，也不得以任何可能引起客户混淆、贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其他商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

Table of Contents

简介 1
目标受众 1
目标业务成果 1
迁移选项 3
Percona XtraBackup 3
优点 5
限制 6
最佳实践 6
MyDumper 6
优点 8
限制 9
最佳实践 9
mysqldump 和 mysqlpump 9
优点 11
限制 12
最佳实践 12
拆分备份 12
Amazon S3 文件网关 14
优点 15
限制 15
最佳实践 15
最佳实践 16
资源 17
文档历史记录 19
..... xx

将大型、多 TB 的 MySQL 或 MariaDB 数据库迁移到 AWS

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

2024 年 11 月（文档历史记录）

许多拥有本地 MySQL 和 MariaDB 数据库服务器的组织都希望将其数据库工作负载迁移到 AWS Cloud。许多人选择 Amazon Relational Database Service（Amazon RDS）for MariaDB、Amazon RDS for MySQL 或 Amazon Aurora MySQL 兼容版。Amazon RDS 旨在更轻松地在云中设置、操作和扩展关系数据库。Amazon Aurora 是 Amazon RDS 的一部分，它提供内置安全性、持续备份、无服务器计算、多达 15 个只读副本、自动多区域复制以及与其他 AWS 服务的集成。

尽管迁移到其中一个 AWS 服务 可以带来很多好处，但数据库迁移是数据库管理员必须执行的最耗时和最关键的任务之一。迁移大型数据库需要精确的规划和实施，并确保迁移的工作负载性能达到或超过原有水平。在本指南中，大型数据库可以指单个数 TB 的数据库，也可以指多个大型数据库，这些数据库加起来的数据量达到数 TB。选择正确的迁移服务和工具是迁移成功的关键。迁移数据库有两种常见方法：逻辑迁移和物理迁移。有关这些方法的更多信息，请参阅 MySQL 和 MariaDB 文档。

本指南讨论了各种开源或第三方工具，您可以使用这些工具将大型本地、多 TB 的 MySQL 和 MariaDB 数据库迁移到 Amazon RDS for MariaDB、Amazon RDS for MySQL 或 Amazon Aurora MySQL 兼容版。本指南中讨论的选项使用逻辑或物理迁移方法，每个选项都包括将大型数据库备份文件从本地数据中心传输到云的多种方法，您可以在云中从备份文件恢复数据库。

目标受众

本指南面向正计划将其 MySQL 或 MariaDB 数据库迁移到 AWS Cloud的计划数据库管理员、数据库工程师、迁移工程师、项目经理及运营或基础设施经理。

目标业务成果

本指南的目标是帮助您：

- 请选择最适合您的使用案例和环境的大型数据库的迁移方法。
- 避免迁移策略存在缺陷时可能发生的延迟和经济损失。
- 了解每种迁移选项的优势和限制。
- 了解可用于将大型数据库备份文件从本地数据中心传输到 AWS Cloud的不同方法。

- 查看迁移大型数据库的总体最佳实践，并查看每种工具的最佳实践，这可以帮助您更有效地迁移数据库。

大型 MySQL 和 MariaDB 数据库的迁移选项

您可以从多种选项中进行选择，从本地 MySQL 或 MariaDB 数据库迁移到亚马逊关系数据库服务 (Amazon RDS) 或亚马逊 Aurora MySQL-Compatible Edition 数据库实例。选择正确的迁移方法和工具对于成功迁移至关重要，在本指南中，您将根据可用性、数据大小和停机时间要求评估选项。

以下是可用于将多 TB 的自行管理的 MySQL 数据库高效迁移到 Amazon RDS、Aurora 或 Amazon Elastic Compute Cloud（Amazon EC2）数据库实例的常用迁移工具和方法：

- Percona XtraBackup（物理）
- MyDumper（逻辑）
- mysqldump 和 mysqlpump（逻辑）
- 拆分备份（物理、逻辑或两者）

以下是可用于将数 TB MySQL-compatible （例如 MariaDB）数据库高效迁移到亚马逊 RDS、Aurora 或 Amazon EC2 数据库实例的常用迁移工具和方法：

- MyDumper（逻辑）
- mysqldump 和 mysqlpump（逻辑）
- 拆分备份（物理、逻辑或两者）

对于每种迁移工具，您可以使用多种方法将大型数据库备份文件传输到 AWS Cloud。每个工具都提供了选项，您也可以使用 Amazon S3 文件网关。有关更多信息，请参阅本指南中的使用 Amazon S3 文件网关传输备份文件。

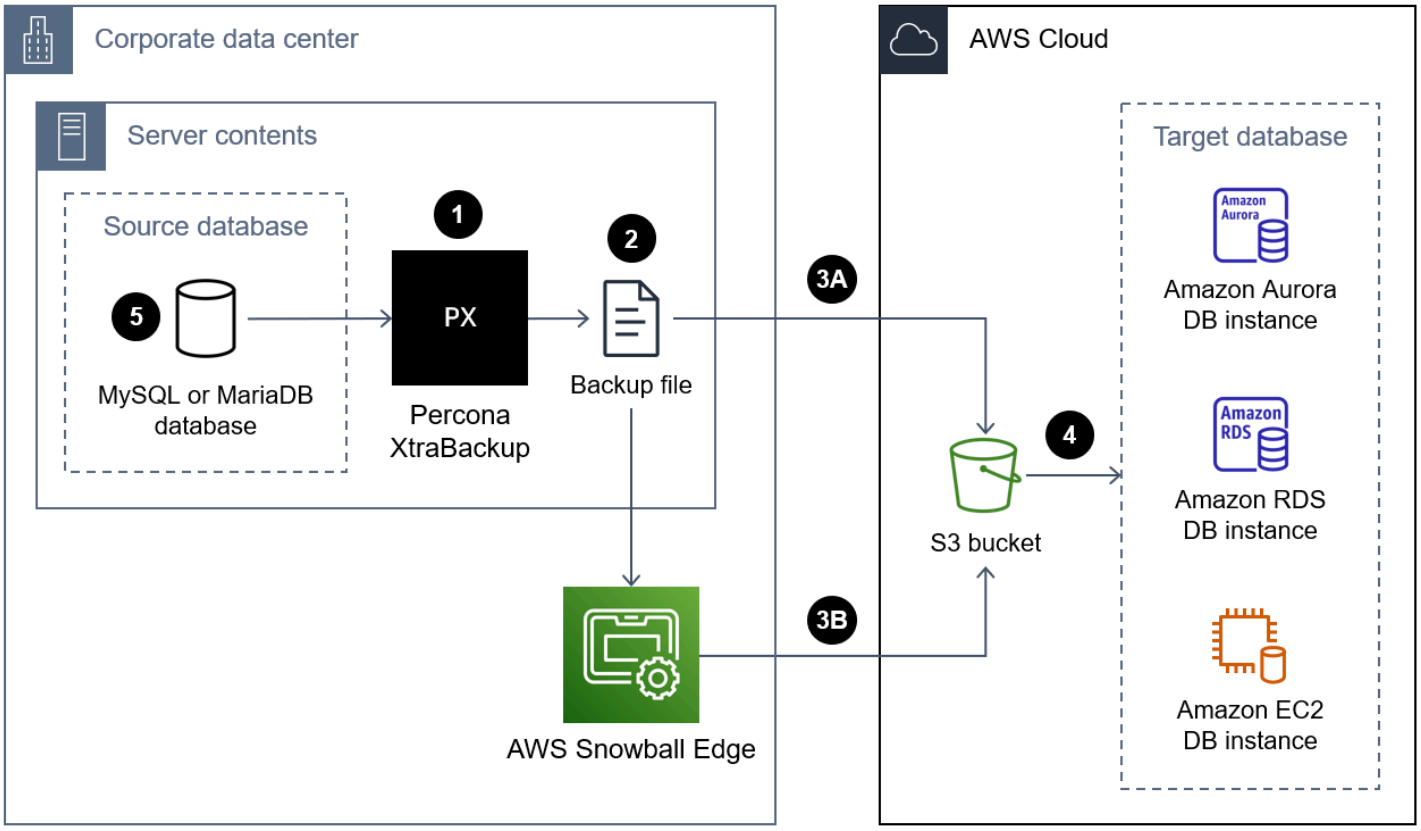
Percona XtraBackup

> **Important**
>
> MariaDB 10.3 或更高版本不支持 Percona XtraBackup ，10.1 和 10.2 版本仅部分支持 Percona。

Percona XtraBackup 是适用于 MySQL和MariaDB的常见开源热备份软件，可为InnoDB和XtraDB存储引擎进行非阻塞备份。它可以与 MySQL 或 MariaDB 服务器协同工作。有关该工具及其部分功能和优势的更多信息，请参阅Percona文档 XtraBackup中的关于Percona。 XtraBackup

此工具使用物理迁移方法。它直接复制 MySQL 或 MariaDB 数据目录及其中的文件。对于大型数据库（例如大于 100 GB 的数据库），与其他一些工具相比，这可以提供明显更好的恢复时间。您可以创建本地源数据库的备份，将备份文件迁移到云，然后在新的目标数据库实例上恢复备份。

下图显示了使用 Percona XtraBackup 备份文件迁移数据库所涉及的高级步骤。根据备份文件的大小，可通过两种选项将备份传输到 AWS Cloud中的 Amazon Simple Storage Service（Amazon S3）存储桶。



以下是使用 Percona 将数据库迁移 XtraBackup 到的步骤： AWS Cloud

1. 在本地服务器 XtraBackup 上安装 Percona。如果你使用的是 Amazon Aurora MySQL 版本 2 或亚马逊 RDS，请参阅安装 Percona XtraBackup 2.4。如果你使用的是 Amazon Aurora MySQL 版本 3，请参阅 Percona 文档中的安装 Percona XtraBackup 8.0。 XtraBackup
2. 创建 MySQL 或 MariaDB 源数据库的完整备份。有关 Percona XtraBackup 2.4 的说明，请参阅完整备份。有关 Percona XtraBackup 8.0 的说明，请参阅创建完整备份。
3. 使用组织中经批准的服务或工具，通过 Internet 传输备份文件，例如：
   - AWS Site-to-Site VPN
   - AWS Client VPN

- AWS Direct Connect
- Amazon S3 文件网关（有关更多信息，请参阅本指南中的 使用 Amazon S3 文件网关传输备份文件。）
- AWS Command Line Interface (AWS CLI)

4. 从 Amazon S3 存储桶中，将备份文件还原到目标数据库实例。有关说明，请参阅：
   - 对于 Aurora MySQL-Compatible 版，请参阅亚马逊 RDS 文档中的使用亚马逊 S3 存储桶从 MySQL 迁移数据。
   - 对于 Amazon RDS for MySQL 或 Amazon EC2，请参阅将数据导入 MySQL 数据库实例。
   - 对于 Amazon RDS for MariaDB 或适用于 Amazon EC2 的 Amazon RDS，请参阅将数据导入 MariaDB 数据库实例。
5. （可选）您可以在源数据库和目标数据库实例之间设置复制。您可以使用二进制日志（binlog）复制来减少停机时间。有关更多信息，请参阅下列内容：
   - MySQL 文档中的 Setting the replication source configuration
   - 对于 Amazon Aurora，请参阅以下内容：
     - Aurora 文档中的通过复制同步 Amazon Aurora MySQL 数据库集群和 MySQL 数据库
     - Aurora 文档中的在 Amazon Aurora 中使用二进制日志复制
   - 对于 Amazon RDS，请参阅以下内容：
     - Amazon RDS 文档中的使用 MySQL 复制
     - Amazon RDS 文档中的使用 MariaDB 复制
   - 对于 Amazon EC2，请参阅以下内容：
     - MySQL 文档中的设置二进制日志文件基于位置的复制
     - MySQL 文档中的设置副本
     - MariaDB 文档中的设置复制

优点

- 由于 Percona XtraBackup 使用物理迁移方法，因此恢复过程通常比使用逻辑迁移方法的工具更快。这是因为性能受磁盘或网络吞吐量的限制，而不是受数据处理所需的计算资源的限制。
- 由于还原过程是将文件从 S3 存储桶直接复制到目标数据库实例，因此 Percona XtraBackup 文件的恢复速度通常比使用其他工具创建的备份文件更快。
- Percona 适应 XtraBackup 性强。例如，它支持多线程以帮助您更快地复制文件，并支持压缩以减小备份的大小。

限制

- 无法进行离线备份，因为 Percona XtraBackup 必须有权访问源数据库服务器。
- Percona XtraBackup 只能用于具有相同系统架构的系统。例如，无法将在 Intel for Windows Server 上运行的源数据库的备份恢复到 ARM for Linux 目标服务器。
- MariaDB 版本 10.3 或更高版本 XtraBackup 不支持 Percona，MariaDB 版本 10.2 和 10.1 版本仅部分支持 Percona。有关更多信息，请参阅 MariaDB 知识库中的 Percona XtraBackup 概述：与 MariaDB 的兼容性。
- 你不能使用 Percona XtraBackup 将源 MariaDB 数据库恢复到目标 MySQL 数据库实例，例如适用于 MySQL 的 Amazon RDS 或 Aurora。 MySQL-Compatible
- 您可以在 S3 存储桶中存储的数据总量和对象数量不受限制，但最大文件大小为 5 TB。如果备份文件超过 5 TB，则可以将其分为多个较小的文件。
- 关闭该`innodb_file_per_table`设置后，Percona XtraBackup 不支持使用`--tables`、、、`--tables-exclude``--tables-file``--databases``--databases-exclude`、或的部分备份。`--databases-file`有关 Percona XtraBackup 版本 2.4 的更多信息，请参阅部分备份。有关 Percona 8.0 XtraBackup 版本的更多信息，请参阅创建部分备份。

最佳实践

- 要提高备份过程的性能，请执行以下操作：
  - 使用 --parallel=<threads> 并行复制多个文件
  - 使用 --compress-threads=<threads> 并行压缩多个文件
  - 使用 --use-memory=<size> 增加内存
  - 使用 --encrypt-threads=<threads> 并行加密多个文件
- 确保源服务器上有足够的空间来生成数据库备份文件。
- 使用 Percona xbstream（.xbstream）格式文件生成数据库备份。有关更多信息，请参阅 Per XtraBackup cona 文档中的 xbstream 二进制文件概述。

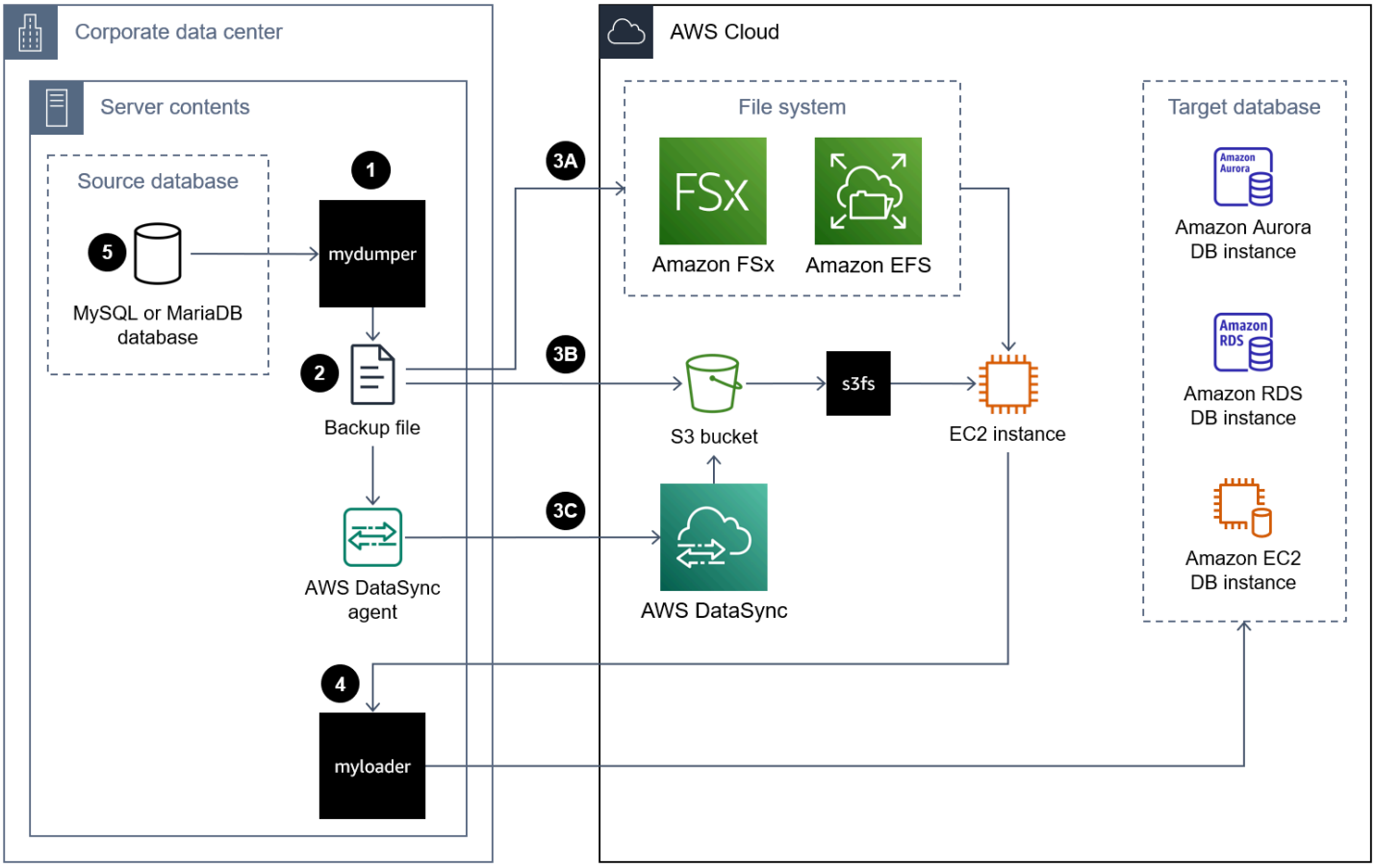
MyDumper

MyDumper(GitHub) 是一个开源的逻辑迁移工具，由两个实用程序组成：

- MyDumper 导出 MySQL 数据库的一致备份。它支持使用多个并行线程备份数据库，每个可用 CPU 核心最多一个线程。

- myloader 读取由创建的备份文件 MyDumper，连接到目标数据库实例，然后恢复数据库。

下图显示了使用 MyDumper 备份文件迁移数据库所涉及的高级步骤。此架构图包括三个选项，用于将备份文件从本地数据中心迁移到 AWS Cloud中的 EC2 实例。



以下是使用将数据库迁移 MyDumper 到的步骤 AWS Cloud：

1. 安装 MyDumper 和我的加载器。有关说明，请参阅如何安装 mydumper/myloader (GitHub)。
2. MyDumper 用于创建源 MySQL 或 MariaDB 数据库的备份。有关说明，请参阅如何使用 MyDumper。
3. 使用以下方法之一将备份文件移动到中的 EC2 实例： AWS Cloud

   方法 3A：将 Amazon FSx 或 Amazon Elastic File System（Amazon EFS）文件系统挂载到运行数据库实例的本地服务器。您可以使用 AWS Direct Connect 或 Site-to-Site VPN 来建立连接。您可以直接将数据库备份到已挂载的文件共享，也可以分两步执行备份，方法是将数据库备份到本地文件系统，然后将其上传到已挂载的 FSx 或 EFS 卷。接下来，将 Amazon FSx 或 Amazon EFS 文件系统（也挂载在本地服务器上）挂载到 EC2 实例上。

方法 3B — 使用 AWS CLI、 AWS 软件开发工具包或 Amazon S3 REST API 将备份文件从本地服务器直接移动到 S3 存储桶。如果目标 S3 存储桶位于距离数据中心较远的地方 AWS 区域 ，则可以使用 Amazon S3 Transfer Accel eration 更快地传输文件。使用 s3fs-fuse 文件系统在 EC2 实例上挂载 S3 存储桶。

方法 3C：在本地数据中心安装 AWS DataSync 代理，然后使用 AWS DataSync 将备份文件移到 Amazon S3 存储桶。使用 s3fs-fuse 文件系统在 EC2 实例上挂载 S3 存储桶。

> **Note**
>
> 您也可以使用 Amazon S3 文件网关将大型数据库备份文件传输到 AWS Cloud中的 S3 存储桶。有关更多信息，请参阅本指南中的使用 Amazon S3 文件网关传输备份文件。

4. 使用 myloader 恢复目标数据库实例上的备份。有关说明，请参阅 myloader 用法 (GitHub)。
5. （可选）您可以在源数据库和目标数据库实例之间设置复制。您可以使用二进制日志（binlog）复制来减少停机时间。有关更多信息，请参阅下列内容：
   - MySQL 文档中的 Setting the replication source configuration
   - 对于 Amazon Aurora，请参阅以下内容：
     - Aurora 文档中的通过复制同步 Amazon Aurora MySQL 数据库集群和 MySQL 数据库
     - Aurora 文档中的在 Amazon Aurora 中使用二进制日志复制
   - 对于 Amazon RDS，请参阅以下内容：
     - Amazon RDS 文档中的使用 MySQL 复制
     - Amazon RDS 文档中的使用 MariaDB 复制
   - 对于 Amazon EC2，请参阅以下内容：
     - MySQL 文档中的设置二进制日志文件基于位置的复制
     - MySQL 文档中的设置副本
     - MariaDB 文档中的设置复制

优点

- MyDumper 通过使用多线程支持并行性，这可以提高备份和恢复操作的速度。
- MyDumper 避免了昂贵的字符集转换例程，这有助于确保代码的高效。
- MyDumper 通过为表和元数据转储单独的文件来简化数据查看和解析。

- MyDumper 维护所有线程的快照，并提供主日志和辅助日志的准确位置。
- 您可以使用 Perl 兼容正则表达式（PCRE），来指定是包含还是排除表或数据库。

限制

- 如果您的数据转换过程需要采用平面格式而不是 SQL 格式的中间转储文件，则可以选择其他工具。
- myloader 不会自动导入数据库用户账户。如果您要将备份恢复到 Amazon RDS 或 Aurora，请重新创建具有所需权限的用户。有关更多信息，请参阅 Amazon RDS 文档中的主用户账户权限。如果您要将备份恢复到 Amazon EC2 数据库实例，则可以手动导出源数据库用户账户并将其导入 EC2 实例。

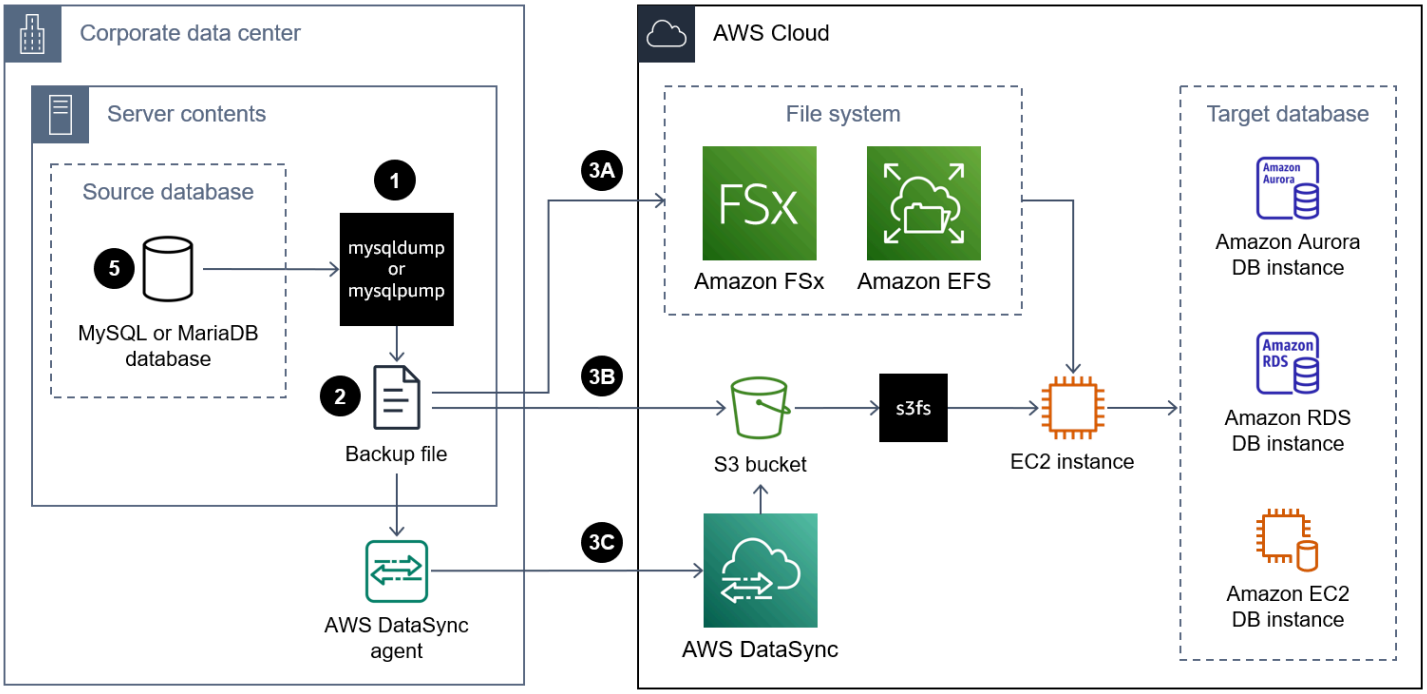
最佳实践

- 配置 MyDumper 为将每个表划分为多个段，例如每个段中的 10,000 行，并将每个段写入单独的文件中。这样，以后就可以并行导入数据。
- 如果您使用的是 InnoDB 引擎，请使用 `--trx-consistency-only` 选项来最大限度地减少锁定。
- 使用导 MyDumper 出数据库可能会成为读取密集型，并且该过程可能会影响生产数据库的整体性能。如果您有副本数据库实例，请从该副本运行导出过程。在从副本运行导出之前，停止复制 SQL 线程。这有助于更快地运行导出过程。
- 不要在工作高峰时段导出数据库。避免高峰时段可以在数据库导出期间稳定主生产数据库的性能。
- Amazon RDS for MySQL 不支持 `keyring_aws` 插件。有关更多信息，请参阅已知问题和限制。要将本地加密表迁移到 Amazon RDS 实例，在备份脚本中，您需要从 `CREATE TABLE` 语法中删除 `ENCRYPTION` 或 `DEFAULT ENCRYPTION`。对于静态加密，您可以使用 AWS Key Management Service（AWS KMS）密钥。有关更多信息，请参阅加密 Amazon RDS 资源。

mysqldump 和 mysqlpump

mysqldump 和 mysqlpump 是 MySQL 的原生数据库备份工具。MariaDB 支持 mysqldump 但不支持 mysqlpump。这两个工具都创建逻辑备份，并且是 MySQL 客户端程序的一部分。mysqldump 支持单线程处理。mysqlpump 支持并行处理数据库和数据库内的对象，以加快转储过程。它是在 MySQL 5.7.8 版中引入的。mysqlpump 在 MySQL 8.4 版中已删除。

下图显示了使用 mysqldump 或 mysqlpump 备份文件迁移数据库所涉及的概要步骤。



以下是使用 mysqldump 或 mysqlpump 将数据库迁移到 AWS Cloud的步骤：

1. 在本地服务器上安装 MySQL Shell。有关说明，请参阅 MySQL 文档中的 Installing MySQL Shell。这将同时安装 mysqldump 和 mysqlpump。
2. 使用 mysqldump 或 mysqlpump 创建源本地数据库的备份。有关说明，请参阅 MySQL 文档中的 mysqldump 和 mysqlpump，或者参阅 MariaDB 文档中的 Making Backups with mysqldump。有关调用 MySQL 程序和指定选项的更多信息，请参阅 Using MySQL programs。
3. 使用以下方法之一将备份文件移动到中的 EC2 实例： AWS Cloud

   方法 3A：将 Amazon FSx 或 Amazon Elastic File System（Amazon EFS）文件系统挂载到运行数据库实例的本地服务器。您可以使用 AWS Direct Connect 或 Site-to-Site VPN 来建立连接。您可以直接将数据库备份到已挂载的文件共享，也可以分两步执行备份，方法是将数据库备份到本地文件系统，然后将其上传到已挂载的 FSx 或 EFS 卷。接下来，将 Amazon FSx 或 Amazon EFS 文件系统（也挂载在本地服务器上）挂载到 EC2 实例上。

   方法 3B — 使用 AWS CLI、 AWS 软件开发工具包或 Amazon S3 REST API 将备份文件从本地服务器直接移动到 S3 存储桶。如果目标 S3 存储桶位于距离数据中心较远的地方 AWS 区域 ，则可以使用 Amazon S3 Transfer Accel eration 更快地传输文件。使用 s3fs-fuse 文件系统在 EC2 实例上挂载 S3 存储桶。

   方法 3C：在本地数据中心安装 AWS DataSync 代理，然后使用 AWS DataSync 将备份文件移到 Amazon S3 存储桶。使用 s3fs-fuse 文件系统在 EC2 实例上挂载 S3 存储桶。

> **Note**
>
> 您也可以使用 Amazon S3 文件网关将大型数据库备份文件传输到 AWS Cloud中的 S3 存储桶。有关更多信息，请参阅本指南中的使用 Amazon S3 文件网关传输备份文件。

4. 使用原生恢复方法恢复目标数据库上的备份。有关说明，请参阅 MySQL 文档中的重新加载 SQL-Format 备份，或参见 MariaDB 文档中的从转储文件恢复数据。
5. （可选）您可以在源数据库和目标数据库实例之间设置复制。您可以使用二进制日志（binlog）复制来减少停机时间。有关更多信息，请参阅下列内容：
   - MySQL 文档中的 Setting the replication source configuration
   - 对于 Amazon Aurora，请参阅以下内容：
     - Aurora 文档中的通过复制同步 Amazon Aurora MySQL 数据库集群和 MySQL 数据库
     - Aurora 文档中的在 Amazon Aurora 中使用二进制日志复制
   - 对于 Amazon RDS，请参阅以下内容：
     - Amazon RDS 文档中的使用 MySQL 复制
     - Amazon RDS 文档中的使用 MariaDB 复制
   - 对于 Amazon EC2，请参阅以下内容：
     - MySQL 文档中的设置二进制日志文件基于位置的复制
     - MySQL 文档中的设置副本
     - MariaDB 文档中的设置复制

优点

- mysqldump 和 mysqlpump 包含在 MySQL Server 安装中
- 这些工具生成的备份文件采用更易读的格式。
- 在恢复备份文件之前，您可以使用标准文本编辑器修改生成的.sql 文件。
- 您可以备份特定的表、数据库，甚至是特定的数据选择。
- mysqldump 和 mysqlpump 是独立于机器架构的。

限制

- mysqldump 是一个单线程备份过程。对于小型数据库，备份性能良好，但当备份大小超过 10 GB 时，效率可能会降低。
- 逻辑格式的备份文件很大，尤其是在另存为文本时，而且创建和恢复速度通常很慢。
- 数据恢复速度可能很慢，因为在目标数据库实例中重新应用 SQL 语句需要大量的磁盘 I/O 和 CPU 处理插入、创建索引和强制执行引用完整性约束。
- 5.7.8 之前的 MySQL 版本或 8.4 版及更高版本不支持 mysqlpump 实用程序。
- 默认情况下，mysqlpump 不备份系统数据库，例如 `performance_schema` 或 `sys`。要备份部分系统数据库，请在命令行中对其进行显式命名。
- mysqldump 不备份 InnoDB `CREATE TABLESPACE` 语句。

> **Note**
>
> 只有在将 MySQL 或 MariaDB 数据库备份恢复到 EC2 实例时，CREATE TABLESPACE 语句和系统数据库的备份才有用。这些备份不用于 Amazon RDS 或 Aurora。

最佳实践

- 恢复数据库备份时，在目标数据库的会话级别禁用密钥检查，例如 `FOREIGN_KEY_CHECKS`。这会提高恢复速度。
- 确保数据库用户具有足够的权限来创建和恢复备份。

拆分备份

拆分备份策略是指通过将备份分成多个部分来迁移大型数据库服务器。您可以使用不同的方法来迁移备份的每个部分。对于以下使用案例，这可能是最佳选择：

- 数据库服务器规模庞大，但各个数据库都很小：当数据库服务器的总大小为多 TB，但每个独立的用户数据库的大小小于 1 TB 时，这是一种不错的方法。为了缩短总体迁移期，您可以单独并行迁移单个数据库。

  让我们举一个本地 2 TB 数据库服务器的示例。此服务器由四个数据库组成，每个数据库大小为 0.5 TB。您可以分别对每个单独的数据库进行备份。恢复备份时，您可以并行恢复实例上的所有数据

库，或者如果这些数据库是独立的，则可以在单独的实例上恢复每个备份。最佳做法是在单独的实例上恢复独立的数据库，而不是在同一个实例上进行恢复。有关更多信息，请参阅本指南中的“最佳实践”。

- 数据库服务器规模庞大，但各个数据库表都很小：当数据库服务器的总大小为多 TB，但每个独立的数据库表的大小小于 1 TB 时，这是一种不错的方法。为了缩短整体迁移期，您可以单独迁移独立的表。

  让我们举一个大小为 1 TB 的单个用户数据库的示例，它是本地数据库服务器中唯一的数据库。数据库中有 10 个表，每个表的容量为 100 GB。您可以分别对每个单独的表进行备份。恢复备份时，您可以并行恢复实例上的所有表。

- 数据库同时包含事务性工作负载表和非事务性工作负载表：与上一个使用案例类似，当事务性工作负载表和非事务性工作负载表都在同一个数据库中时，您可以使用拆分备份方法。

  让我们举一个 2 TB 数据库的示例，该数据库由用于在线事务处理（OLTP）的 0.5 TB 关键工作负载表和一个用于存档旧数据的 1.5 TB 表组成。您可以将除归档表之外的所有数据库对象的备份作为单一事务且一致的备份。然后，您只对归档表进行另一个单独的备份。对于归档表备份，您还可以考虑使用条件来拆分备份文件中的行数，从而进行多个并行备份。以下是示例：

  ```
  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
  ```

  恢复备份文件时，您可以并行恢复事务性工作负载备份和归档表备份。

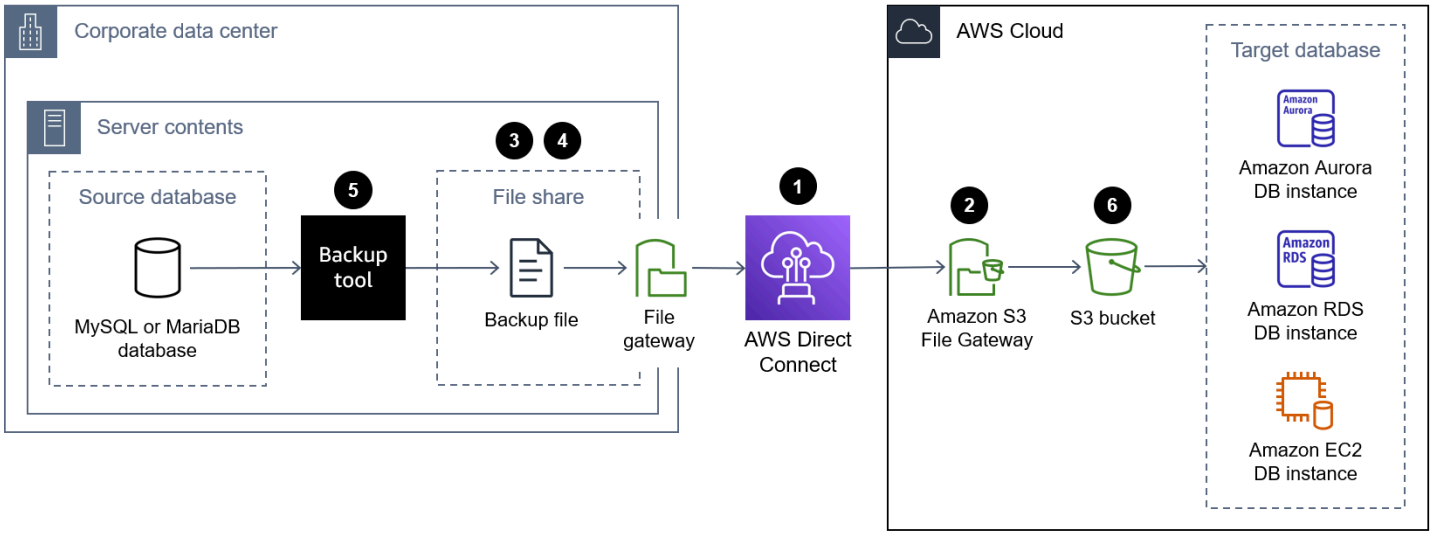
- 计算资源限制-如果本地服务器中的计算资源（例如 CPU、内存或磁盘）有限 I/O，这可能会影响备份时的稳定性和性能。您可以将其分成几个部分，而不是进行完整备份。

  例如，本地生产服务器可能负载繁重，而 CPU 资源有限。如果您在此服务器上对多 TB 的数据库进行一次性备份，则可能会消耗额外的 CPU 资源，并对生产服务器产生不利影响。不要进行完整的数据库备份，而是将备份分成多个部分，例如每个部分备份 2-3 个表。

使用 Amazon S3 文件网关传输备份文件

Amazon S3 文件网关通过文件接口将您的本地环境连接到 Amazon Simple Storage Service（Amazon S3），以便您可以使用行业标准文件协议 [例如网络文件系统（NFS）和服务器消息块（SMB）] 存储和检索 Amazon S3 对象。它旨在成为一种经济高效、可扩展的解决方案，用于在云中存储数据。由于您可以使用它来存储数据库备份文件，因此该服务可以帮助您将大型本地数据库迁移到 AWS Cloud。例如，您可以使用 Amazon S3 文件网关和首选的数据库备份工具，将大型 MySQL 或 MariaDB 数据库直接备份到 Amazon S3 存储桶。然后，您可以将 S3 存储桶挂载到目标实例并恢复备份。

下图显示了使用 Amazon S3 文件网关将本地数据库的备份文件传输到 AWS Cloud中的 S3 存储桶时所涉及的概要步骤。



以下是使用 Amazon S3 文件网关将数据库备份文件从本地数据中心传输到 AWS Cloud中的 S3 存储桶的步骤：

1. 使用或之类的 AWS Cloud 服务将本地数据中心连接到， AWS Direct Connect AWS Site-to-Site VPN 或者使用公共 Internet 连接。
2. 创建 S3 文件网关 有关说明，请参阅创建网关。
3. 创建由 S3 文件网关托管的 NFS 或 SMB 文件共享。有关说明，请参阅创建文件共享。
4. 将 NFS 或 SMB 文件共享挂载在托管 MySQL 或 MariaDB 数据库的本地服务器上。有关说明，请参阅挂载并使用您的文件共享。
5. 将本地 MySQL 或 MariaDB 数据库备份到挂载 NFS 文件共享的目录中。您可以使用本指南中讨论的任何备份工具。
6. 使用本指南中讨论的任何方法，恢复目标数据库实例上的数据库备份。

优点

- 通过直接在 S3 存储桶中生成数据库备份，并直接从同一 S3 存储桶恢复目标数据库实例上的备份，您可以显著加快端到端迁移过程。
- 数据库备份文件持久存储在 Amazon S3 中，您可以选择生命周期管理策略和 S3 存储类别。

限制

使用 Amazon S3 文件网关文件共享时存在以下限制：

- 每个网关的最大文件共享数为 50。
- 为了防止在多个文件共享使用同一 S3 存储桶时发生读写冲突，您必须将每个文件共享配置为使用唯一的前缀名称。
- 单个文件的最大大小为 5TB，即 Amazon S3 中任何单个对象的最大大小。
- 路径最大长度为 1024 个字符。
- 仅当使用 Windows SMB 客户端访问文件共享时，为 Active Directory 启用的文件共享才支持 Windows ACL。
- Amazon S3 文件网关针对每个文件和目录最多支持 10 个 ACL 条目。
- SMB 文件共享的根 ACL 设置仅在网关上。这些设置在网关更新和重启后均保持不变。

> **Note**
>
> 如果在根目录而不是根目录下的父文件夹上配置 ACL，则 ACL 权限不会保留在 Amazon S3 中。

最佳实践

有关 Amazon S3 文件网关最佳实践的更多信息，请参阅 S3 文件网关文档中的最佳实践。

迁移大型 MySQL 和 MariaDB 数据库的最佳实践

除了为每个迁移选项列出的特定于工具的最佳实践外，请查看以下一般最佳实践。无论您选择哪种工具，这些最佳实践都适用于迁移大型、多 TB 的 MySQL 和 MariaDB 数据库：

- 确保源数据库和目标数据库上有足够的空间，来创建和恢复备份。
- 迁移完成之前，不要对目标数据库实例创建二级索引。二级索引在导入期间会增加额外的维护开销，并可能减慢导入过程。
- 如果您使用多线程方法，请选择正确的线程数。对于导出，我们建议您为每个 CPU 核心使用一个线程。对于导入，我们建议您每两个 CPU 核心使用一个线程。
- 数据转储通常由活跃数据库服务器执行，这些服务器是任务关键型生产环境的一部分。如果数据转储严重影响性能，而这在您的环境中是不可接受的，请考虑以下操作之一：
  - 源服务器有副本，您可以从其中一个副本转储数据。
  - 源服务器受定期备份程序保护：
    - 如果备份格式适合直接导入目标数据库，则使用备份数据作为导入过程的输入。
    - 如果备份格式不适合直接导入目标数据库，请使用备份来预调配临时数据库并从中转储数据。
  - 如果副本和备份不可用：
    - 在生产流量最低的非高峰时段执行转储。
    - 减少转储操作的并发性，以便服务器有足够的备用容量来处理生产流量。
- 仅创建用户创建的数据库的转储。
- 在目标数据库上重新创建用户并配置其权限。有关更多信息，请参阅 Amazon RDS 的身份和访问管理、Amazon Aurora 的身份和访问管理或 Amazon EC2 的身份和访问管理。
- 迁移由多个独立数据库组成的大型数据库服务器时，为每个数据库创建一个单独的实例。这可以帮助您更有效地管理数据库并改善资源预调配，而单独的计算资源可以提高数据库性能。

资源

AWS 规范指引

- AWS 大型迁移的组合手册
- 关系数据库的迁移策略
- 将本地 MySQL 数据库迁移至 Amazon RDS for MySQL
- 使用 GTID 在 Amazon RDS for MySQL 和 Amazon EC2 上的 MySQL 之间设置数据复制
- 使用原生工具将本地 MariaDB 数据库迁移至 Amazon RDS for MariaDB

AWS 博客文章

- Amazon RDS for MySQL 和 MariaDB 实例的安全最佳实践
- 将自行管理的 MariaDB 迁移到 Amazon Aurora MySQL

用于恢复备份的资源

- 创建存储桶（Amazon S3 文档）
- 使用 SSH 连接到 Linux 实例（Amazon EC2 文档）
- 配置 AWS CLI（AWS CLI 文档）
- 同步命令（AWS CLI 命令参考）
- 创建 IAM policy 以访问 Amazon S3 资源（Aurora 文档）
- 数据库集群先决条件（Aurora 文档）
- 使用数据库子网组（Aurora 文档）
- 为私有数据库集群创建 VPC 安全组（Aurora 文档）
- 从 Amazon S3 存储桶恢复 Aurora MySQL 数据库集群（Aurora 文档）
- 使用 MySQL 或其他 Aurora 数据库集群设置复制（Aurora 文档）
- rds_set_external_master 程序（Amazon RDS 文档）
- rds_start_replication 程序（Amazon RDS 文档）

AWS 营销

- ：Amazon Aurora

- Amazon RDS for MariaDB
- Amazon RDS for MySQL。
- Amazon S3 文件网关

其他资源：

- Percona XtraBackup
- MyDumper
- mysqldump
- mysqlpump

文档历史记录

下表介绍了本指南的一些重要更改。如果您希望收到有关未来更新的通知，可以订阅 RSS 源。

变更	说明	日期
mysqlpump 可用性	在 MySQL 8.4 版中删除了 mysqlpump。我们更新了 mysqldump 和 mysqlpump 部分，以反映此可用性更改。	2024 年 11 月 21 日
MyDumper 最佳实践	我们更新了 MyDumper 的最佳实践，添加了有关迁移加密数据库表的信息。	2024 年 10 月 24 日
Percona XtraBackup 版本	在 Percona XtraBackup 部分中，我们更新了说明，以反映 Amazon Aurora MySQL 和 Amazon RDS 支持的 Percona XtraBackup 版本。	2023 年 8 月 3 日
初次发布	—	2023 年 4 月 6 日

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。