대용량 멀티 테라바이트 MySQL 또는 MariaDB 데이터베이스를 로 마이그레이션 AWS

AWS 권장 가이드

AWS 권장 가이드: 대용량 멀티 테라바이트 MySQL 또는 MariaDB 데이터베이스를 로 마이그레이션 AWS

Table of Contents

대용량 멀티 테라바이트 MySQL 또는 MariaDB 데이터베이스를 로 마이그레이션 AWS

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

2024년 11월(문서 기록)

온프레미스 MySQL 및 MariaDB 데이터베이스 서버가 있는 많은 조직은 데이터베이스 워크로드를 AWS 클라우드로 마이그레이션하는 데 관심이 있습니다. 많은 사용자가 Amazon Relational Database Service(Amazon RDS) for MariaDB, Amazon RDS for MySQL 또는 Amazon Aurora MySQL 호환 에디션을 선택합니다. Amazon RDS는 클라우드에서 관계형 데이터베이스의 설치, 운영 및 규모 조정을 용이하게 수행하도록 설계되었습니다. Amazon Aurora는 Amazon RDS의 일부이며 기본 제공 보안, 지속적 백업, 서버프리 컴퓨팅, 최대 15개의 읽기 복제본, 자동화된 다중 리전 복제 및 다른 AWS 서비스와의 통합을 제공합니다.

이러한 작업 중 하나로 마이그레이션하면 많은 이점을 얻을 AWS 서비스 수 있지만 데이터베이스 마이그레이션은 데이터베이스 관리자가 수행해야 하는 가장 시간이 많이 걸리고 중요한 작업 중 하나입니다. 대규모 데이터베이스를 마이그레이션하고 마이그레이션된 워크로드의 성능이 동등하거나 개선되도록 하려면 정확한 계획 및 구현이 필요합니다. 이 가이드에서 대형 데이터베이스는 단일 테라바이트 규모의 데이터베이스를 참조하거나 최대 멀티테라바이트의 데이터를 추가하는 많은 대형 데이터베이스를 참조할 수 있습니다. 올바른 마이그레이션 서비스 및 도구를 선택하는 것이 마이그레이션 성공의 핵심입니다. 데이터베이스 마이그레이션에는 논리적 접근 방식 및 물리적 접근 방식이라는 두 가지 일반적인 접근 방식이 있습니다. 이러한 접근 방식에 대한 자세한 내용은 MySQL 및 MariaDB 설명서를 참조하세요.

이 가이드에서는 대형 온프레미스 멀티테라바이트 규모의 MySQL 및 MariaDB 데이터베이스를 Amazon RDS for MariaDB, Amazon RDS for MySQL 또는 Amazon Aurora MySQL 호환 에디션으로 마이그레이션하는 데 사용할 수 있는 다양한 오픈 소스 MySQL 또는 서드 파티 도구에 대해 설명합니다. 이 가이드에서 설명하는 옵션은 논리적 또는 물리적 마이그레이션 접근 방식을 사용하며, 각 옵션에는 대규모 데이터베이스 백업 파일을 온프레미스 데이터 센터에서 클라우드로 전송하기 위한 여러 접근 방식이 포함되어 있습니다. 이 방법에서는 백업 파일에서 데이터베이스를 복원할 수 있습니다.

대상 독자

이 가이드는 MySQL 또는 MariaDB 데이터베이스를 AWS 클라우드로 마이그레이션하려는 프로그램 데이터베이스 관리자, 데이터베이스 엔지니어, 마이그레이션 엔지니어, 프로젝트 관리자, 운영 또는 인프라 관리자를 대상으로 합니다.

목표 비즈니스 성과

이 가이드의 목표는 다음을 지원합니다.

- 사용 사례 및 환경에 가장 적합한 대규모 데이터베이스의 마이그레이션 접근 방식을 선택합니다.
- 마이그레이션 전략에 결함이 있을 경우 발생할 수 있는 지연 및 재정적 손실을 방지합니다.
- 각 마이그레이션 옵션의 장점과 제한 사항에 대해 알아봅니다.
- 온프레미스 데이터 센터에서 AWS 클라우드로 대용량 데이터베이스 백업 파일을 전송하는 데 사용할 수 있는 다양한 접근 방식에 대해 알아봅니다.
- 대규모 데이터베이스 마이그레이션을 위한 전체 모범 사례를 검토하고 데이터베이스를 보다 효율적으로 마이그레이션하는 데 도움이 되는 각 도구의 모범 사례를 검토합니다.

대규모 MySQL 및 MariaDB 데이터베이스에 대한 마이그레이션 옵션

광범위한 옵션 중에서 선택하여 온프레미스 MySQL 또는 MariaDB 데이터베이스에서 Amazon Relational Database Service(Amazon RDS) 또는 Amazon Aurora MySQL 호환 버전 데이터베이스 인스턴스로 마이그레이션할 수 있습니다. 성공적인 마이그레이션을 위해서는 올바른 마이그레이션 접근 방식과 도구를 선택하는 것이 필수이며, 이 가이드에서는 사용성, 데이터 크기 및 가동 중지 시간 요구 사항에 따라 옵션을 평가합니다.

다음은 멀티테라바이트 규모의 자체 관리형 MySQL 데이터베이스를 Amazon RDS, Aurora 또는 Amazon Elastic Compute Cloud(Amazon EC2) 데이터베이스 인스턴스로 효율적으로 마이그레이션하는 데 사용할 수 있는 일반적인 마이그레이션 도구 및 접근 방식입니다.

- Percona XtraBackup(물리적)
- MyDumper(논리적)
- mysqldump 및 mysqlpump(논리적)
- 백업 분할(물리적, 논리적 또는 둘 다)

다음은 멀티테라바이트 규모의 MySQL 호환 데이터베이스(예: MariaDB)를 Amazon RDS, Aurora 또는 Amazon EC2 인스턴스로 효율적으로 마이그레이션하는 데 사용할 수 있는 일반적인 마이그레이션 도구 및 접근 방식입니다.

- MyDumper(논리적)
- mysqldump 및 mysqlpump(논리적)
- 백업 분할(물리적, 논리적 또는 둘 다)

각 마이그레이션 도구에는 대용량 데이터베이스 백업 파일을 AWS 클라우드로 전송하는 데 사용할 수 있는 몇 가지 접근 방식이 있습니다. 각 도구에 대한 옵션이 제공되며, Amazon S3 File Gateway를 사용할 수도 있습니다. 자세한 내용은 이 안내서의 Amazon S3 File Gateway를 사용하여 백업 파일 전송 섹션을 참조하세요.

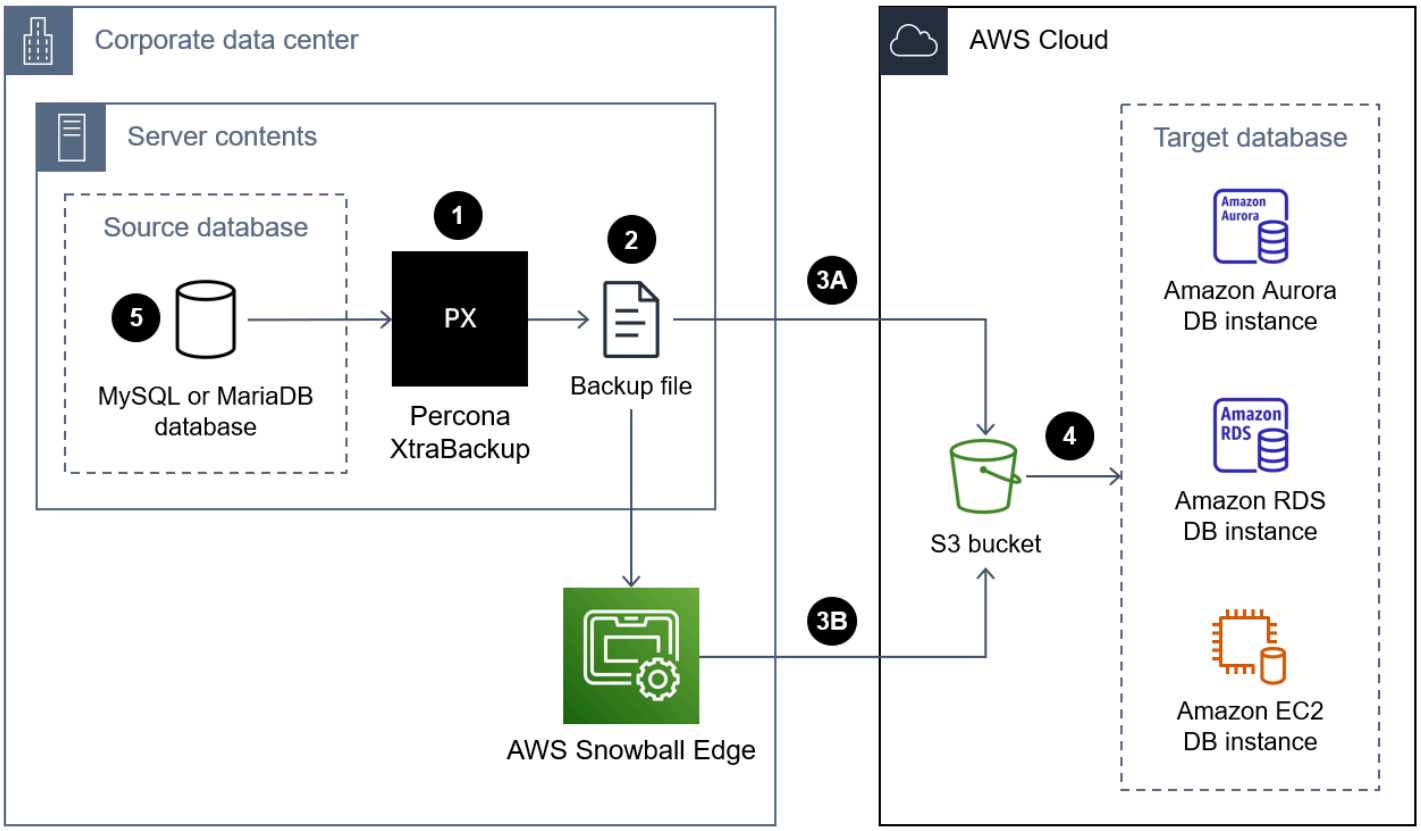
Percona XtraBackup

> **Important**
>
> Percona XtraBackup은 MariaDB 버전 10.3 이상에서는 지원되지 않으며 버전 10.1 및 10.2에서는 부분적으로만 지원됩니다.

Percona XtraBackup은 MySQL 및 MariaDB에 대한 일반적인 오픈 소스 웜 백업 소프트웨어로, InnoDB 및 XtraDB 스토리지 엔진에 대해 비차단 백업을 수행합니다. MySQL 또는 MariaDB 서버에서 작동합니다. 도구 및 몇 가지 기능과 이점에 대한 자세한 내용은 Percona XtraBackup 설명서의 About Percona XtraBackup을 참조하세요.

이 도구는 물리적 마이그레이션 접근 방식을 사용합니다. MySQL 또는 MariaDB 데이터 디렉터리와 안에 있는 파일을 직접 복사합니다. 100GB보다 큰 데이터베이스와 같은 대규모 데이터베이스의 경우 다른 도구보다 훨씬 더 나은 복원 시간을 제공할 수 있습니다. 온프레미스 소스 데이터베이스의 백업을 생성하고 백업 파일을 클라우드로 마이그레이션한 다음 새 대상 데이터베이스 인스턴스에서 백업을 복원합니다.

다음 다이어그램에서는 Percona XtraBackup 백업 파일을 사용하여 데이터베이스를 마이그레이션하는 데 수반되는 상위 수준 단계를 보여줍니다. 백업 파일의 크기에 따라 AWS 클라우드의 Amazon Simple Storage Service(Amazon S3) 버킷으로 백업을 전송하는 데 사용할 수 있는 두 가지 옵션이 있습니다.



다음은 Percona XtraBackup을 사용하여 데이터베이스를 AWS 클라우드로 마이그레이션하는 단계입니다.

1. 온프레미스 서버에 Percona XtraBackup을 설치하세요. Amazon Aurora MySQL 버전 2 또는 Amazon RDS를 사용하는 경우 Percona XtraBackup2.4 설치를 참조하세요. Amazon Aurora MySQL 버전 3을 사용하는 경우 Percona XtraBackup 설명서의 Percona XtraBackup8.0 설치를 참조하세요.
2. 소스 MySQL 또는 MariaDB 데이터베이스의 전체 백업을 생성하세요. Percona XtraBackup 2.4에 대한 지침은 Full backup을 참조하세요. Percona XtraBackup 8.0에 대한 지침은 Create a full backup을 참조하세요.
3. 다음과 같이 조직에서 승인된 서비스 또는 도구를 사용하여 인터넷을 통해 백업 파일을 전송합니다.
   - AWS Site-to-Site VPN
   - AWS Client VPN
   - AWS Direct Connect
   - Amazon S3 File Gateway(자세한 내용은 이 가이드의 Amazon S3 File Gateway를 사용하여 백업 파일 전송 섹션을 참조하세요.)
   - AWS Command Line Interface (AWS CLI)

4. Amazon S3 버킷에서 백업 파일을 대상 데이터베이스 인스턴스로 복원합니다. 지침은 다음을 참조하세요.
   - Aurora MySQL 호환 버전의 경우 Amazon RDS 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.
   - Amazon RDS for MySQL 또는 Amazon EC2의 경우 MySQL DB 인스턴스로 데이터 가져오기를 참조하세요.
   - Amazon RDS for MariaDB 또는 Amazon EC2의 경우 MariaDB DB 인스턴스로 데이터 가져오기를 참조하세요.
5. (선택 사항) 소스 데이터베이스와 대상 데이터베이스 인스턴스 간 복제를 설정할 수 있습니다. 바이너리 로그(binlog) 복제를 사용하여 가동 중지 시간을 줄일 수 있습니다. 자세한 내용은 다음을 참조하세요.
   - Setting the replication source configuration(MySQL 설명서)
   - Amazon Aurora의 경우 다음을 참조하세요.
     - 복제를 사용하여 Amazon Aurora MySQL DB 클러스터를 MySQL 데이터베이스와 동기화(Aurora 설명서)
     - Amazon Aurora에서 binlog 복제 사용(Aurora 설명서)
   - Amazon RDS의 경우 다음을 참조하세요.
     - MySQL 복제 작업(Amazon RDS 설명서)
     - MariaDB 복제 작업(Amazon RDS 설명서)
   - Amazon EC2의 경우 다음을 참조하세요.
     - Setting Up Binary Log File Position Based Replication(MySQL 설명서)
     - Setting Up Replicas(MySQL 설명서)
     - Setting Up Replication(MariaDB 설명서)

장점

- Percona XtraBackup은 물리적 마이그레이션 접근 방식을 사용하기 때문에 복원 프로세스는 일반적으로 논리적 마이그레이션 접근 방식을 사용하는 도구보다 빠릅니다. 데이터 처리에 필요한 컴퓨팅 리소스가 아닌 디스크 또는 네트워크 처리량에 의해 성능이 제한되기 때문입니다.
- 복원 프로세스는 S3 버킷에서 대상 데이터베이스 인스턴스로 파일을 직접 복사하기 때문에 Percona XtraBackup 파일은 일반적으로 다른 도구로 생성된 백업 파일보다 빠르게 복원됩니다.
- Percona XtraBackup은 적응 가능합니다. 예를 들어 파일을 더 빠르게 복사할 수 있도록 여러 스레드를 지원하고 백업 크기를 줄이기 위해 압축을 지원합니다.

제한 사항

- Percona XtraBackup은 소스 데이터베이스 서버에 액세스해야 하므로 오프라인 백업이 불가능합니다.
- Percona XtraBackup은 시스템 아키텍처가 동일한 시스템에서만 사용할 수 있습니다. 예를 들어 Windows Server용 인텔에서 실행되는 소스 데이터베이스의 백업을 Linux용 ARM 대상 서버로 복원할 수 없습니다.
- Percona XtraBackup은 MariaDB 버전 10.3 이상에서는 지원되지 않으며 MariaDB 버전 10.2 및 버전 10.1에서는 부분적으로만 지원됩니다. 자세한 내용은 MariaDB 지식 기반의 Percona XtraBackup Overview: Compatibility with MariaDB를 참조하세요.
- Percona XtraBackup을 사용하여 소스 MariaDB 데이터베이스를 Amazon RDS for MySQL 또는 Aurora MySQL 호환과 같은 대상 MySQL 데이터베이스 인스턴스로 복원할 수 없습니다.
- S3 버킷에 저장할 수 있는 총 데이터 볼륨과 객체 수는 무제한이지만 최대 파일 크기는 5TB입니다. 백업 파일이 5TB를 초과하면 파일을 여러 개의 더 작은 파일로 분할할 수 있습니다.
- `innodb_file_per_table` 설정이 꺼져 있으면 Percona XtraBackup은 `--tables`, `--tables-exclude`, `--tables-file`, `--databases` `--databases-exclude` 또는 `--databases-file`을 사용하는 부분 백업을 지원하지 않습니다. Percona XtraBackup 버전 2.4에 대한 자세한 내용은 Partial backups를 참조하세요. Percona XtraBackup 버전 8.0에 대한 자세한 내용은 Create a partial backup을 참조하세요.

모범 사례

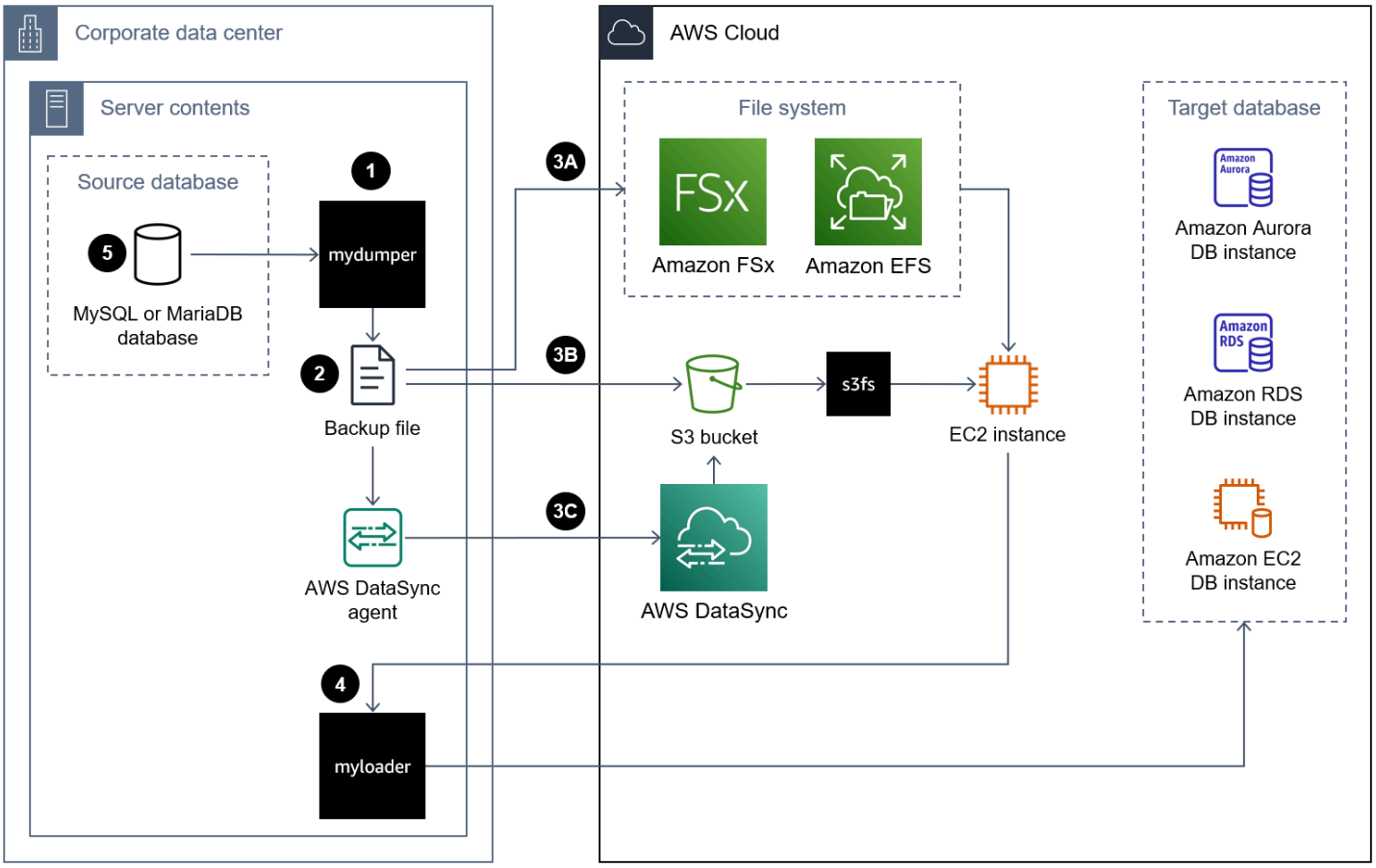
- 백업 프로세스의 성능을 개선하려면 다음을 수행합니다.
  - --parallel=<threads>를 사용하여 병렬로 여러 파일 복사
  - --compress-threads=<threads>를 사용하여 병렬로 여러 파일 압축
  - --use-memory=<size>를 사용하여 메모리 증가
  - --encrypt-threads=<threads>를 사용하여 병렬로 여러 파일 암호화
- 소스 서버에 데이터베이스 백업 파일을 가져올 충분한 공간이 있는지 확인합니다.
- Percona xbstream(.xbstream) 형식 파일을 사용하여 데이터베이스 백업을 생성합니다. 자세한 내용은 Percona XtraBackup 설명서의 The xbstream binary overview를 참조하세요.

MyDumper

MyDumper(GitHub)는 다음 두 가지 유틸리티로 구성된 오픈 소스 논리적 마이그레이션 도구입니다.

- MyDumper는 MySQL 데이터베이스의 일관된 백업을 내보냅니다. 사용 가능한 CPU 코어당 최대 하나의 스레드까지 여러 병렬 스레드를 사용하여 데이터베이스 백업을 지원합니다.
- myloader는 MyDumper에서 생성한 백업 파일을 읽고 대상 데이터베이스 인스턴스에 연결한 다음 데이터베이스를 복원합니다.

다음 다이어그램에서는 MyDumper 백업 파일을 사용하여 데이터베이스를 마이그레이션하는 데 수반되는 상위 수준 단계를 보여줍니다. 이 아키텍처 다이어그램에는 백업 파일을 온프레미스 데이터 센터에서 AWS 클라우드의 EC2 인스턴스로 마이그레이션하는 세 가지 옵션이 포함되어 있습니다.



다음은 MyDumper를 사용하여 데이터베이스를 AWS 클라우드로 마이그레이션하는 단계입니다.

1. MyDumper 및 myloader를 설치하세요. 관련 지침은 How to install mydumper/myloader(GitHub)를 참조하세요.
2. MyDumper를 사용하여 소스 MySQL 또는 MariaDB 데이터베이스의 백업을 생성하세요. 관련 지침은 How to use MyDumper를 참조하세요.
3. 다음 방법 중 하나를 사용하여 백업 파일을의 EC2 인스턴스 AWS 클라우드 로 이동합니다.

접근 방식 3A - Amazon FSx 또는 Amazon Elastic File System(Amazon EFS) 파일 시스템을 데이터베이스 인스턴스를 실행하는 온프레미스 서버에 탑재합니다. AWS Direct Connect 또는 Site-to-Site VPN 를 사용하여 연결을 설정할 수 있습니다. 데이터베이스를 탑재된 파일 공유에 직접 백업하거나 데이터베이스를 로컬 파일 시스템에 백업한 다음 탑재된 FSx 또는 EFS 볼륨에 업로드하여 두 단계로 백업을 수행할 수 있습니다. 그런 다음 온프레미스 서버에도 탑재된 Amazon FSx 또는 Amazon EFS 파일 시스템을 EC2 인스턴스에 탑재하세요.

접근 방식 3B - AWS CLI, AWS SDK 또는 Amazon S3 REST API를 사용하여 백업 파일을 온프레미스 서버에서 S3 버킷으로 직접 이동합니다. 대상 S3 버킷이 데이터 센터와 멀리 떨어진에 AWS 리전 있는 경우 Amazon S3 Transfer Acceleration을 사용하여 파일을 더 빠르게 전송할 수 있습니다. s3fs-fuse 파일 시스템을 사용하여 EC2 인스턴스에 S3 버킷을 탑재하세요.

접근 방식 3C - 온프레미스 데이터 센터에 AWS DataSync 에이전트를 설치한 다음 AWS DataSync를 사용하여 백업 파일을 Amazon S3 버킷으로 이동합니다. s3fs-fuse 파일 시스템을 사용하여 EC2 인스턴스에 S3 버킷을 탑재하세요.

> **Note**
>
> Amazon S3 File Gateway를 사용하여 대용량 데이터베이스 백업 파일을 AWS 클라우드의 S3 버킷으로 전송할 수도 있습니다. 자세한 내용은 이 안내서의 Amazon S3 File Gateway를 사용하여 백업 파일 전송 섹션을 참조하세요.

4. myloader를 사용하여 대상 데이터베이스 인스턴스에서 백업을 복원하세요. 관련 지침은 myloader usage(GitHub)를 참조하세요.
5. (선택 사항) 소스 데이터베이스와 대상 데이터베이스 인스턴스 간 복제를 설정할 수 있습니다. 바이너리 로그(binlog) 복제를 사용하여 가동 중지 시간을 줄일 수 있습니다. 자세한 내용은 다음을 참조하세요.
   - Setting the replication source configuration(MySQL 설명서)
   - Amazon Aurora의 경우 다음을 참조하세요.
     - 복제를 사용하여 Amazon Aurora MySQL DB 클러스터를 MySQL 데이터베이스와 동기화(Aurora 설명서)
     - Amazon Aurora에서 binlog 복제 사용(Aurora 설명서)
   - Amazon RDS의 경우 다음을 참조하세요.
     - MySQL 복제 작업(Amazon RDS 설명서)
     - MariaDB 복제 작업(Amazon RDS 설명서)

- Amazon EC2의 경우 다음을 참조하세요.
  - Setting Up Binary Log File Position Based Replication(MySQL 설명서)
  - Setting Up Replicas(MySQL 설명서)
  - Setting Up Replication(MariaDB 설명서)

장점

- MyDumper는 다중 스레드를 사용하여 병렬화를 지원하므로 백업 및 복원 작업 속도가 향상됩니다.
- MyDumper는 비용이 많이 드는 문자 세트 변환 루틴을 방지하므로 코드가 매우 효율적입니다.
- MyDumper는 테이블 및 메타데이터에 대해 별도의 파일을 덤핑하여 데이터 보기 및 구문 분석을 단순화합니다.
- MyDumper는 모든 스레드에서 스냅샷을 유지 관리하고 기본 및 보조 로그의 정확한 위치를 제공합니다.
- Perl 호환 정규 표현식(PCRE)을 사용하여 테이블 또는 데이터베이스를 포함할지, 제외할지를 지정할 수 있습니다.

제한 사항

- 데이터 변환 프로세스에 SQL 형식 대신 플랫 형식의 중간 덤프 파일이 필요한 경우 다른 도구를 선택할 수 있습니다.
- myloader는 데이터베이스 사용자 계정을 자동으로 가져오지 않습니다. Amazon RDS 또는 Aurora로 백업을 복원하는 경우 필요한 권한을 가진 사용자를 다시 생성합니다. 권한에 대한 자세한 내용은 MariaDB 설명서에서 마스터 사용자 계정 권한을 참조하세요. 백업을 Amazon EC2 데이터베이스 인스턴스로 복원하는 경우 소스 데이터베이스 사용자 계정을 수동으로 내보내 EC2 인스턴스로 가져올 수 있습니다.

모범 사례

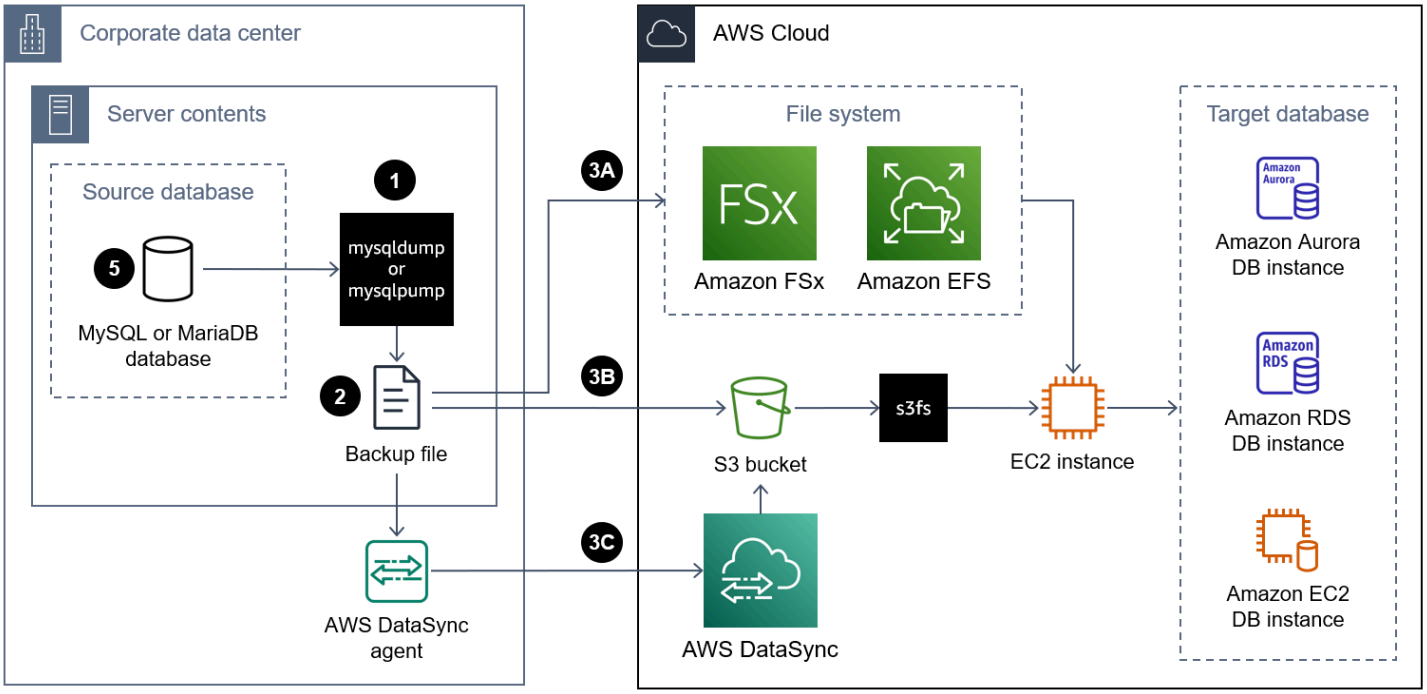
- 각 테이블을 여러 세그먼트(예: 각 세그먼트의 10,000개 행)로 나누고 각 세그먼트를 별도의 파일에 기록하도록 MyDumper를 구성합니다. 이렇게 하면 나중에 데이터를 병렬로 가져올 수 있습니다.
- InnoDB 엔진을 사용하는 경우 `--trx-consistency-only` 옵션을 사용하여 잠금을 최소화합니다.

- MyDumper를 사용하여 데이터베이스를 내보내면 읽기 집약적 작업이 될 수 있으며 프로세스가 프로덕션 데이터베이스의 전반적인 성능에 영향을 미칠 수 있습니다. 복제본 데이터베이스 인스턴스가 있는 경우 복제본에서 내보내기 프로세스를 실행합니다. 복제본에서 내보내기를 실행하기 전에 복제 SQL 스레드를 중지합니다. 이렇게 하면 내보내기 프로세스를 더 빠르게 실행할 수 있습니다.
- 피크 업무 시간에는 데이터베이스를 내보내지 마세요. 피크 시간을 피하면 데이터베이스 내보내기 중에 기본 프로덕션 데이터베이스의 성능이 안정화될 수 있습니다.
- Amazon RDS for MySQL은 `keyring_aws` 플러그인을 지원하지 않습니다. 자세한 내용은 Known issues and limitations를 참조하세요. 온프레미스 암호화된 테이블을 Amazon RDS 인스턴스로 마이그레이션하려면 백업 스크립트의 `CREATE TABLE` 구문에서 `ENCRYPTION` 또는 `DEFAULT ENCRYPTION`을 제거해야 합니다. 저장 시 암호화를 위해 AWS Key Management Service (AWS KMS) 키를 사용할 수 있습니다. 자세한 내용은 Amazon RDS 리소스 암호화 섹션을 참조하십시오.

mysqldump 및 mysqlpump

mysqldump 및 mysqlpump는 MySQL용 네이티브 데이터베이스 백업 도구입니다. MariaDB는 mysqldump를 지원하지만 mysqlpump는 지원하지 않습니다. 이러한 두 도구 모두 논리적 백업을 생성하며 MySQL 클라이언트 프로그램의 일부입니다. mysqldump는 단일 스레드 처리를 지원합니다. mysqlpump는 데이터베이스 및 데이터베이스 내 객체의 병렬화를 지원하여 덤프 프로세스의 속도를 높입니다. MySQL 버전 5.7.8에 도입되었습니다. MySQL 버전 8.4에서는 mysqlpump가 제거되었습니다.

다음 다이어그램에서는 mysqldump 또는 mysqlpump 백업 파일을 사용하여 데이터베이스를 마이그레이션하는 데 수반되는 상위 수준 단계를 보여줍니다.



다음은 mysqldump 또는 mysqlpump를 사용하여 데이터베이스를 AWS 클라우드로 마이그레이션하는 단계입니다.

1. 온프레미스 서버에 MySQL Shell을 설치하세요. 관련 지침은 MySQL 설명서의 Installing MySQL Shell을 참조하세요. 그러면 mysqldump와 mysqlpump가 모두 설치됩니다.
2. mysqldump 또는 mysqlpump를 사용하여 소스 온프레미스 데이터베이스의 백업을 생성하세요. 관련 지침은 MySQL 설명서의 mysqldump 및 mysqlpump를 참조하거나 MariaDB 설명서의 Making Backups with mysqldump를 참조하세요. MySQL 프로그램 간접 호출 및 옵션 지정에 대한 자세한 내용은Using MySQL programs를 참조하세요.
3. 다음 방법 중 하나를 사용하여 백업 파일을의 EC2 인스턴스 AWS 클라우드 로 이동합니다.

   접근 방식 3A - Amazon FSx 또는 Amazon Elastic File System(Amazon EFS) 파일 시스템을 데이터베이스 인스턴스를 실행하는 온프레미스 서버에 탑재합니다. AWS Direct Connect 또는 Site-to-Site VPN 를 사용하여 연결을 설정할 수 있습니다. 데이터베이스를 탑재된 파일 공유에 직접 백업하거나 데이터베이스를 로컬 파일 시스템에 백업한 다음 탑재된 FSx 또는 EFS 볼륨에 업로드하여 두 단계로 백업을 수행할 수 있습니다. 그런 다음 온프레미스 서버에도 탑재된 Amazon FSx 또는 Amazon EFS 파일 시스템을 EC2 인스턴스에 탑재하세요.

   접근 방식 3B - AWS CLI, AWS SDK 또는 Amazon S3 REST API를 사용하여 백업 파일을 온프레미스 서버에서 S3 버킷으로 직접 이동합니다. 대상 S3 버킷이 데이터 센터와 멀리 떨어진에 AWS 리

전 있는 경우 Amazon S3 Transfer Acceleration을 사용하여 파일을 더 빠르게 전송할 수 있습니다. s3fs-fuse 파일 시스템을 사용하여 EC2 인스턴스에 S3 버킷을 탑재하세요.

접근 방식 3C - 온프레미스 데이터 센터에 AWS DataSync 에이전트를 설치한 다음 AWS DataSync를 사용하여 백업 파일을 Amazon S3 버킷으로 이동합니다. s3fs-fuse 파일 시스템을 사용하여 EC2 인스턴스에 S3 버킷을 탑재하세요.

> **Note**
>
> Amazon S3 File Gateway를 사용하여 대용량 데이터베이스 백업 파일을 AWS 클라우드의 S3 버킷으로 전송할 수도 있습니다. 자세한 내용은 이 안내서의 Amazon S3 File Gateway를 사용하여 백업 파일 전송 섹션을 참조하세요.

4. 네이티브 복원 방법을 사용하여 대상 데이터베이스에서 백업을 복원하세요. 관련 지침은 MySQL 설명서의 Reloading SQL-Format Backups를 참조하거나 MariaDB 설명서의 Restoring Data from Dump Files를 참조하세요.
5. (선택 사항) 소스 데이터베이스와 대상 데이터베이스 인스턴스 간 복제를 설정할 수 있습니다. 바이너리 로그(binlog) 복제를 사용하여 가동 중지 시간을 줄일 수 있습니다. 자세한 내용은 다음을 참조하세요.
   - Setting the replication source configuration(MySQL 설명서)
   - Amazon Aurora의 경우 다음을 참조하세요.
     - 복제를 사용하여 Amazon Aurora MySQL DB 클러스터를 MySQL 데이터베이스와 동기화(Aurora 설명서)
     - Amazon Aurora에서 binlog 복제 사용(Aurora 설명서)
   - Amazon RDS의 경우 다음을 참조하세요.
     - MySQL 복제 작업(Amazon RDS 설명서)
     - MariaDB 복제 작업(Amazon RDS 설명서)
   - Amazon EC2의 경우 다음을 참조하세요.
     - Setting Up Binary Log File Position Based Replication(MySQL 설명서)
     - Setting Up Replicas(MySQL 설명서)
     - Setting Up Replication(MariaDB 설명서)

장점

- mysqldump 및 mysqlpump는 MySQL Server 설치에 포함됨

- 이러한 도구에서 생성된 백업 파일은 더 읽기 쉬운 형식입니다.
- 백업 파일을 복원하기 전에 표준 텍스트 편집기를 사용하여 결과 .sql 파일을 수정할 수 있습니다.
- 특정 테이블, 데이터베이스 또는 특정 데이터 선택 항목을 백업할 수 있습니다.
- mysqldump 및 mysqlpump는 시스템 아키텍처에 독립적입니다.

제한 사항

- mysqldump는 단일 스레드 백업 프로세스입니다. 백업을 수행하는 성능은 작은 데이터베이스에서 적당하지만 백업 크기가 10GB보다 크면 비효율적일 수 있습니다.
- 논리적 형식의 백업 파일은 특히 텍스트로 저장될 때 볼륨이 크며, 종종 생성 및 복원 속도가 느립니다.
- 대상 DB 인스턴스에서 SQL 문을 다시 적용하려면 삽입, 인덱스 생성 및 참조 무결성 제약 조건 적용을 위해 집약적 디스크 I/O 및 CPU 처리가 필요하기 때문에 데이터 복원 속도가 느려질 수 있습니다.
- mysqlpump 유틸리티는 MySQL 버전 5.7.8 이하 또는 버전 8.4 이상에서 지원되지 않습니다.
- 기본적으로 mysqlpump는 `performance_schema` 또는 `sys`와 같은 시스템 데이터베이스를 백업하지 않습니다. 시스템 데이터베이스의 일부를 백업하려면 명령줄에 명시적으로 이름을 지정합니다.
- mysqldump는 InnoDB `CREATE TABLESPACE` 문을 백업하지 않습니다.

> **Note**
>
> CREATE TABLESPACE 문 및 시스템 데이터베이스 백업은 MySQL 또는 MariaDB 데이터베이스 백업을 EC2 인스턴스로 복원하는 경우에만 유용합니다. 이러한 백업은 Amazon RDS 또는 Aurora에서 사용되지 않습니다.

모범 사례

- 데이터베이스 백업을 복원하는 경우 대상 데이터베이스의 세션 수준에서 `FOREIGN_KEY_CHECKS`와 같은 키 검사를 비활성화합니다. 그러면 복원 속도가 빨라집니다.
- 데이터베이스 사용자에게 백업을 생성하고 복원할 수 있는 충분한 권한이 있는지 확인합니다.

백업 분할

분할 백업 전략은 백업을 여러 부분으로 나누어 대규모 데이터베이스 서버를 마이그레이션하는 방법입니다. 다양한 접근 방식을 사용하여 백업의 각 부분을 마이그레이션할 수 있습니다. 다음 사용 사례에 가장 적합한 옵션일 수 있습니다.

- 대규모 데이터베이스 서버이지만 소규모 개별 데이터베이스 - 총 데이터베이스 서버의 크기가 TB 단위이지만 개별 독립 사용자 데이터베이스의 크기가 1TB 미만인 경우 바람직한 접근 방식입니다. 전체 마이그레이션 기간을 줄이기 위해 개별 데이터베이스를 개별적으로 병렬로 마이그레이션할 수 있습니다.

  온프레미스 2TB 데이터베이스 서버에 대한 예제를 살펴봅니다. 이 서버는 각각 0.5TB인 4개의 데이터베이스로 구성됩니다. 각 개별 데이터베이스의 백업을 별도로 수행할 수 있습니다. 백업을 복원할 때 인스턴스의 모든 데이터베이스를 병렬로 복원하거나 독립된 데이터베이스인 경우 각 백업을 별도의 인스턴스에서 복원할 수 있습니다. 독립된 데이터베이스를 동일한 인스턴스에서 복원하는 대신 별도의 인스턴스에서 복원하는 것이 모범 사례입니다. 자세한 내용은 이 가이드의 모범 사례를 참조하세요.
- 대규모 데이터베이스 서버이지만 작은 개별 데이터베이스 테이블 - 총 데이터베이스 서버의 크기가 TB 단위이지만 각 독립 데이터베이스 테이블의 크기가 1TB 미만인 경우 바람직한 접근 방식입니다. 전체 마이그레이션 기간을 줄이기 위해 독립된 테이블을 개별적으로 마이그레이션할 수 있습니다.

  1TB인 단일 사용자 데이터베이스 예제를 사용해 보겠습니다. 이 데이터베이스는 온프레미스 데이터베이스 서버의 유일한 데이터베이스입니다. 데이터베이스에는 10개의 테이블이 있으며 각 테이블은 100GB입니다. 각 개별 테이블의 백업을 별도로 수행할 수 있습니다. 백업을 복원하는 경우 인스턴스의 모든 테이블을 병렬로 복원할 수 있습니다.
- 데이터베이스에 트랜잭션 워크로드 테이블과 비트랜잭션 워크로드 테이블 모두 포함됨 - 이전 사용 사례와 마찬가지로 동일한 데이터베이스에 트랜잭션 워크로드 테이블과 비트랜잭션 워크로드 테이블이 모두 있는 경우 분할 백업 접근 방식을 사용할 수 있습니다.

  온라인 트랜잭션 처리(OLTP)에 사용되는 0.5TB의 중요 워크로드 테이블과 이전 데이터를 아카이브하는 데 사용되는 단일 1.5TB 테이블로 구성된 2TB 데이터베이스 예제를 살펴보겠습니다. 아카이브 테이블을 제외한 모든 데이터베이스 객체의 백업을 단일 트랜잭션 및 일관된 백업으로 사용할 수 있습니다. 그런 다음 아카이브 테이블에 대한 다른 별도의 백업을 수행합니다. 아카이브 테이블 백업의 경우 조건을 사용하여 백업 파일의 행 수를 분할함으로써 여러 병렬 백업을 수행하는 방법도 고려할 수 있습니다. 다음은 예제입니다.

```
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
```

백업 파일을 복원할 때 트랜잭션 워크로드 백업과 아카이브 테이블 백업을 병렬로 복원할 수 있습니다.

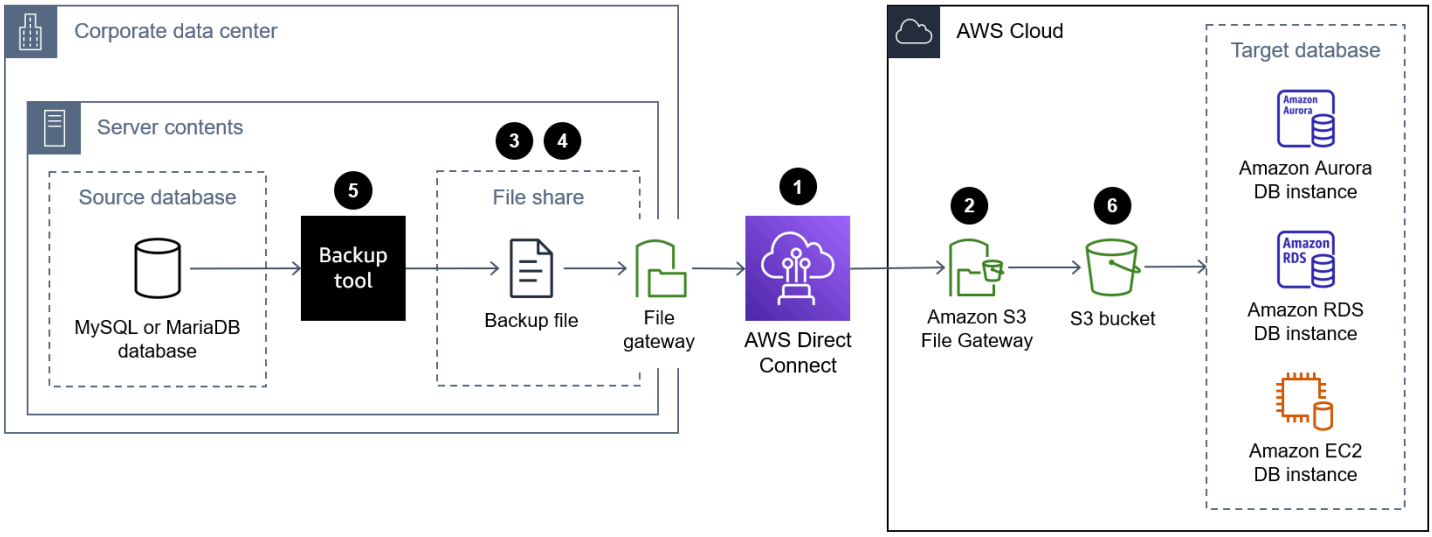
- 컴퓨팅 리소스 제한 사항 - CPU, 메모리 또는 디스크 I/O와 같은 온프레미스 서버의 컴퓨팅 리소스가 제한된 경우 백업을 수행할 때 안정성과 성능에 영향을 미칠 수 있습니다. 전체 백업을 가져오는 대신 부분으로 나눌 수 있습니다.

  예를 들어 온프레미스 프로덕션 서버는 워크로드가 많고 CPU 리소스가 제한적일 수 있습니다. 이 서버에서 멀티테라바이트 데이터베이스의 단일 실행 백업을 수행하는 경우 추가 CPU 리소스를 소비하고 프로덕션 서버에 부정적인 영향을 미칠 수 있습니다. 전체 데이터베이스 백업을 수행하는 대신 백업을 각각 2~3개의 테이블과 같이 여러 부분으로 나눕니다.

Amazon S3 File Gateway를 사용하여 백업 파일 전송

Amazon S3 File Gateway는 파일 인터페이스를 통해 온프레미스 환경을 Amazon Simple Storage Service(Amazon S3)에 연결하므로 Network File System(NFS) 및 Server Message Block(SMB)과 같은 업계 표준 파일 프로토콜을 사용하여 Amazon S3 객체를 저장하고 검색할 수 있습니다. 이는 클라우드에 데이터를 저장하기 위한 비용 효율적이고 확장 가능한 솔루션으로 설계되었습니다. 이를 사용하여 데이터베이스 백업 파일을 저장할 수 있으므로 이 서비스는 대규모 온프레미스 데이터베이스를 AWS 클라우드로 마이그레이션하는 데 도움이 될 수 있습니다. 예를 들어 Amazon S3 File Gateway와 선호하는 데이터베이스 백업 도구를 사용하여 대용량 MySQL 또는 MariaDB 데이터베이스를 Amazon S3 버킷에 직접 백업할 수 있습니다. 그런 다음 S3 버킷을 대상 인스턴스에 탑재하고 백업을 복원할 수 있습니다.

다음 다이어그램에서는 Amazon S3 File Gateway를 사용하여 온프레미스 데이터베이스의 백업 파일을 AWS 클라우드의 S3 버킷으로 전송할 때 포함되는 상위 수준 단계를 보여줍니다.



다음은 Amazon S3 File Gateway를 사용하여 온프레미스 데이터 센터에서 AWS 클라우드의 S3 버킷으로 데이터베이스 백업 파일을 전송하는 단계입니다.

1. AWS Direct Connect AWS Site-to-Site VPN 또는와 같은 서비스를 AWS 클라우드 사용하거나 퍼블릭 인터넷 연결을 사용하여 온프레미스 데이터 센터를에 연결합니다.
2. S3 File Gateway를 생성하세요. 관련 지침은 Creating your gateway를 참조하세요.
3. S3 File Gateway에서 호스팅하는 NFS 또는 SMB 파일 공유를 생성하세요. 관련 지침은 Create a file share를 참조하세요.

4. MySQL 또는 MariaDB 데이터베이스를 호스팅하는 온프레미스 서버에 NFS 또는 SMB 파일 공유를 탑재하세요. 관련 지침은 Mount and use your file share를 참조하세요.
5. 온프레미스 MySQL 또는 MariaDB 데이터베이스를 NFS 파일 공유가 탑재된 디렉터리에 백업하세요. 이 가이드에서 설명하는 모든 백업 도구를 사용할 수 있습니다.
6. 이 가이드에서 설명하는 접근 방식 중 하나를 사용하여 대상 데이터베이스 인스턴스에서 데이터베이스 백업을 복원하세요.

장점

- S3 버킷에서 직접 데이터베이스 백업을 생성하고 동일한 S3 버킷에서 직접 대상 DB 인스턴스의 백업을 복원하면 포괄적인 마이그레이션 프로세스를 크게 가속화할 수 있습니다.
- 데이터베이스 백업 파일은 내구성을 유지하며 Amazon S3에 저장되고, 수명 주기 관리 정책과 S3 스토리지 클래스를 선택합니다.

제한 사항

다음은 Amazon S3 File Gateway 파일 공유 사용 시 제한 사항입니다.

- 게이트웨이당 최대 파일 공유 개수는 50개입니다.
- 여러 파일 공유가 동일한 S3 버킷을 사용하는 경우 읽기 및 쓰기 충돌을 방지하려면 고유한 접두사 이름을 사용하도록 각 파일 공유를 구성해야 합니다.
- 개별 파일의 최대 크기는 5TB이며, 이는 Amazon S3에서 개별 개체의 최대 크기입니다.
- 최대 경로 길이는 1,024자입니다.
- Windows ACL은 Windows SMB 클라이언트를 사용하여 파일 공유에 액세스할 때 Active Directory용으로 활성화된 파일 공유에만 지원됩니다.
- Amazon S3 File Gateway는 각 파일과 디렉터리에 대해 최대 10개의 ACL 항목을 지원합니다.
- SMB 파일 공유의 루트 ACL 설정은 게이트웨이에만 있습니다. 이러한 설정은 게이트웨이 업데이트 및 재시작 전반에 걸쳐 지속됩니다.

> **Note**
>
> 루트 아래에 있는 상위 폴더 대신 루트에서 ACL을 구성할 경우에는 ACL 권한은 Amazon S3에서 유지되지 않습니다.

모범 사례

Amazon S3 File Gateway의 모범 사례에 대한 자세한 내용은 S3 File Gateway 설명서의 Best practices를 참조하세요.

대규모 MySQL 및 MariaDB 데이터베이스 마이그레이션 모범 사례

각 마이그레이션 옵션에 대해 나열된 도구별 모범 사례 외에도 다음 일반 모범 사례를 검토합니다. 이러한 모범 사례는 선택한 도구에 관계없이 멀티테라바이트 규모의 대규모 MySQL 및 MariaDB 데이터베이스를 마이그레이션할 때 적용됩니다.

- 소스 및 대상 데이터베이스에 백업을 가져와 복원할 수 있는 충분한 공간이 있는지 확인합니다.
- 마이그레이션이 완료될 때까지 대상 데이터베이스 인스턴스에 보조 인덱스를 생성하지 마세요. 보조 인덱스는 가져오는 중에 추가 유지 관리 오버헤드를 추가하며 가져오기 프로세스의 속도를 늦출 수 있습니다.
- 다중 스레드 접근 방식을 사용하는 경우 적절한 수의 스레드를 선택합니다. 내보내기의 경우 각 CPU 코어에 하나의 스레드를 사용하는 것이 좋습니다. 가져오기의 경우 CPU 코어 2개마다 스레드 1개를 사용하는 것이 좋습니다.
- 데이터 덤프는 종종 미션 크리티컬 프로덕션 환경의 일부인 활성 데이터베이스 서버에서 수행됩니다. 데이터 덤프가 성능에 심각한 영향을 미치고 환경에서 허용되지 않는 경우 다음 중 하나를 고려합니다.
  - 소스 서버에는 복제본이 있으므로 복제본 중 하나에서 데이터를 덤프할 수 있습니다.
  - 소스 서버는 정기 백업 절차의 적용을 받습니다.
    - 백업 형식이 대상 데이터베이스로 직접 가져오는 데 적합한 경우 백업 데이터를 가져오기 프로세스의 입력으로 사용합니다.
    - 백업 형식을 대상 데이터베이스로 직접 가져오는 데 적합하지 않은 경우 백업을 사용하여 임시 데이터베이스를 프로비저닝하고 해당 데이터베이스에서 데이터를 덤프합니다.
  - 복제본 및 백업을 사용할 수 없는 경우:
    - 프로덕션 트래픽의 사용량이 가장 낮은 시간에 덤프를 수행합니다.
    - 서버에서 프로덕션 트래픽을 처리하기에 충분한 예비 용량을 보유하도록 덤프 작업의 동시성을 줄입니다.
- 사용자가 생성한 데이터베이스의 덤프만 생성합니다.
- 대상 데이터베이스에서 사용자를 다시 생성하고 해당 권한을 구성합니다. 자세한 내용은 Amazon RDS에 대한 자격 증명 및 액세스 관리, Amazon Aurora에 대한 자격 증명 및 액세스 관리 또는 Amazon EC2에 대한 자격 증명 및 액세스 관리를 참조하세요.
- 여러 개의 독립적인 데이터베이스로 구성된 대규모 데이터베이스 서버를 마이그레이션할 때 각 데이터베이스에 대해 별도의 인스턴스를 생성합니다. 이를 통해 데이터베이스를 보다 효율적으로 관

리하고 리소스 프로비저닝을 개선할 수 있으며, 별도의 컴퓨팅 리소스로 데이터베이스 성능을 개선할 수 있습니다.

리소스

AWS 권장 가이드

- AWS 대규모 마이그레이션을 위한 포트폴리오 플레이북
- Migration strategy for relational databases
- 온프레미스 MySQL 데이터베이스를 Amazon RDS for MySQL로 마이그레이션
- GTID를 사용하여 Amazon RDS for MySQL와 Amazon EC2의 MySQL 간에 데이터 복제를 설정합니다
- 온프레미스 MariaDB 데이터베이스를 기본 도구를 사용하여 Amazon RDS for MariaDB로 마이그레이션

AWS 블로그 게시물

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

백업 복원을 위한 리소스

- 버킷 생성(Amazon S3 설명서)
- SSH를 사용하여 Linux 인스턴스에 연결(Amazon Ec2 설명서)
- AWS CLI 구성(AWS CLI 설명서)
- sync command(AWS CLI 명령 참조)
- Amazon S3 리소스에 액세스할 수 있는 IAM 정책 생성(Aurora 설명서)
- DB 클러스터 사전 요구 사항(Aurora 설명서)
- DB 서브넷 그룹을 사용한 작업(Aurora 설명서)
- 프라이빗 DB 클러스터에 대한 VPC 보안 그룹 생성(Aurora 설명서)
- Amazon S3 버킷에서 Aurora MySQL DB 클러스터 복원(Aurora 설명서)
- MySQL 또는 다른 Aurora DB 클러스터를 사용한 복제 설정(Aurora 설명서)
- rds_set_external_master 프로시저(Amazon RDS 설명서)
- rds_start_replication 프로시저(Amazon RDS 설명서)

AWS 마케팅

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

기타 리소스:

- Percona XtraBackup
- MyDumper
- mysqldump
- mysqlpump

문서 기록

아래 표에 이 가이드의 주요 변경 사항이 설명되어 있습니다. 향후 업데이트에 대한 알림을 받으려면 RSS 피드를 구독하십시오.

변경 사항	설명	날짜
mysqlpump 가용성	MySQL 버전 8.4에서 mysqlpump가 제거되었습니다. 이 가용성 변경을 반영하도록 mysqldump 및 mysqlpump 섹션을 업데이트했습니다.	2024년 11월 21일
MyDumper 모범 사례	암호화된 데이터베이스 테이블 마이그레이션에 대한 정보를 추가하기 위해 MyDumper의 모범 사례를 업데이트했습니다.	2024년 10월 24일
Percona XtraBackup 버전	Percona XtraBackup 섹션에서 Amazon Aurora MySQL 및 Amazon RDS에서 지원하는 Percona XtraBackup의 버전을 반영하도록 지침을 업데이트했습니다.	2023년 8월 3일
최초 게시	—	2023년 4월 6일

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.