



Migration de grandes bases de données MySQL ou MariaDB de plusieurs téraoctets vers AWS

AWS Conseils prescriptifs



AWS Conseils prescriptifs: Migration de grandes bases de données MySQL ou MariaDB de plusieurs téraoctets vers AWS

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

Les marques et la présentation commerciale d'Amazon ne peuvent être utilisées en relation avec un produit ou un service qui n'est pas d'Amazon, d'une manière susceptible de créer une confusion parmi les clients, ou d'une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d'Amazon appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

Table of Contents

Introduction	1
Public visé	2
Résultats commerciaux ciblés	2
Options de migration	3
Percona XtraBackup	4
Avantages	6
Limitations	7
Bonnes pratiques	8
MyDumper	8
Avantages	11
Limitations	11
Bonnes pratiques	12
mysqldump et mysqlpump	12
Avantages	15
Limitations	15
Bonnes pratiques	16
Sauvegarde fractionnée	16
Passerelle de fichiers Amazon S3	19
Avantages	20
Limitations	20
Bonnes pratiques	21
Bonnes pratiques	22
Ressources	24
Historique du document	26
.....	xxvii

Migration de grandes bases de données MySQL ou MariaDB de plusieurs téraoctets vers AWS

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

Novembre 2024 ([historique du document](#))

De nombreuses organisations disposant de serveurs de base de données MySQL et MariaDB sur site souhaitent migrer leurs charges de travail de base de données vers le cloud. AWS Cloud Beaucoup choisissent Amazon Relational Database Service (Amazon RDS) pour MariaDB, Amazon RDS for MySQL ou Amazon Aurora MySQL compatible Edition. [Amazon RDS](#) est conçu pour faciliter la configuration, l'exploitation et le dimensionnement des bases de données relationnelles dans le cloud. [Amazon Aurora](#) fait partie d'Amazon RDS et propose une sécurité intégrée, des sauvegardes continues, un traitement sans serveur, jusqu'à 15 répliques en lecture, une réplication multirégionale automatisée et une intégration avec d'autres solutions. Services AWS

Bien que la migration vers l'une de ces solutions Services AWS puisse présenter de nombreux avantages, la migration de base de données est l'une des tâches les plus chronophages et les plus critiques que doivent effectuer les administrateurs de bases de données. Cela nécessite une planification et une mise en œuvre précises pour migrer de grandes bases de données et garantir que les performances de la charge de travail migrée sont équivalentes ou améliorées. Dans ce guide, les bases de données volumineuses peuvent faire référence à une seule base de données de plusieurs téraoctets ou à de nombreuses bases de données volumineuses totalisant plusieurs téraoctets de données. La sélection des services et outils de migration appropriés est essentielle au succès de la migration. Il existe deux approches communes pour la migration d'une base de données : logique et physique. Pour plus d'informations sur ces approches, consultez la documentation [MySQL](#) et [MariaDB](#).

Ce guide décrit les différents outils open source ou tiers que vous pouvez utiliser pour migrer de grandes bases de données MySQL et MariaDB locales de plusieurs téraoctets vers Amazon RDS pour MariaDB, Amazon RDS pour MySQL ou Amazon Aurora MySQL Compatible Edition. Les options décrites dans ce guide utilisent des approches de migration logiques ou physiques, et chaque option inclut plusieurs approches pour transférer les fichiers de sauvegarde de base de données volumineux du centre de données sur site vers le cloud, où vous pouvez restaurer la base de données à partir du fichier de sauvegarde.

Public visé

Ce guide s'adresse aux administrateurs de bases de données de programmes, aux ingénieurs de base de données, aux ingénieurs de migration, aux chefs de projet et aux responsables des opérations ou de l'infrastructure qui envisagent de migrer leurs bases de données MySQL ou MariaDB vers le. AWS Cloud

Résultats commerciaux ciblés

L'objectif de ce guide est de vous aider à :

- Choisissez l'approche de migration pour une base de données volumineuse qui correspond le mieux à votre cas d'utilisation et à votre environnement.
- Évitez les retards et les pertes financières qui peuvent survenir lorsque la stratégie de migration est défectueuse.
- Découvrez les avantages et les limites de chaque option de migration.
- Découvrez les différentes approches que vous pouvez utiliser pour transférer des fichiers de sauvegarde de base de données volumineux de votre centre de données sur site vers le AWS Cloud.
- Passez en revue les meilleures pratiques générales pour la migration de bases de données volumineuses ainsi que les meilleures pratiques pour chaque outil, qui peuvent vous aider à migrer la base de données plus efficacement.

Options de migration pour les grandes bases de données MySQL et MariaDB

Vous pouvez choisir parmi une vaste gamme d'options pour migrer des bases de données MySQL ou MariaDB sur site vers des instances de bases de données Amazon Relational Database Service (Amazon RDS) ou Amazon Aurora Edition. MySQL-Compatible Le choix de l'approche et du bon outil de migration est essentiel pour une migration réussie. Dans ce guide, vous évaluez les options en fonction de votre facilité d'utilisation, de la taille des données et de vos exigences en matière de temps d'arrêt.

Voici les outils et approches de migration courants disponibles pour migrer efficacement des bases de données MySQL autogérées de plusieurs téraoctets vers des instances de base de données Amazon RDS, Aurora ou Amazon Elastic Compute Cloud (Amazon EC2) :

- [Percona XtraBackup](#)(Physique)
- [MyDumper](#)(Logique)
- [mysqldump et mysqlpump](#)(Logique)
- [Sauvegarde fractionnée](#)(Physique, logique ou les deux)

Voici les outils et approches de migration courants disponibles pour migrer efficacement des bases de données de plusieurs téraoctets MySQL-compatible (telles que MariaDB) vers des instances de base de données Amazon RDS, Aurora ou Amazon EC2 :

- [MyDumper](#)(Logique)
- [mysqldump et mysqlpump](#)(Logique)
- [Sauvegarde fractionnée](#)(Physique, logique ou les deux)

Pour chaque outil de migration, vous pouvez utiliser plusieurs approches pour transférer le fichier de sauvegarde de base de données volumineux vers le AWS Cloud. Des options sont fournies pour chaque outil, et vous pouvez également utiliser Amazon S3 File Gateway. Pour plus d'informations, consultez [Utilisation d'Amazon S3 File Gateway pour transférer des fichiers de sauvegarde](#) dans ce guide.

Percona XtraBackup

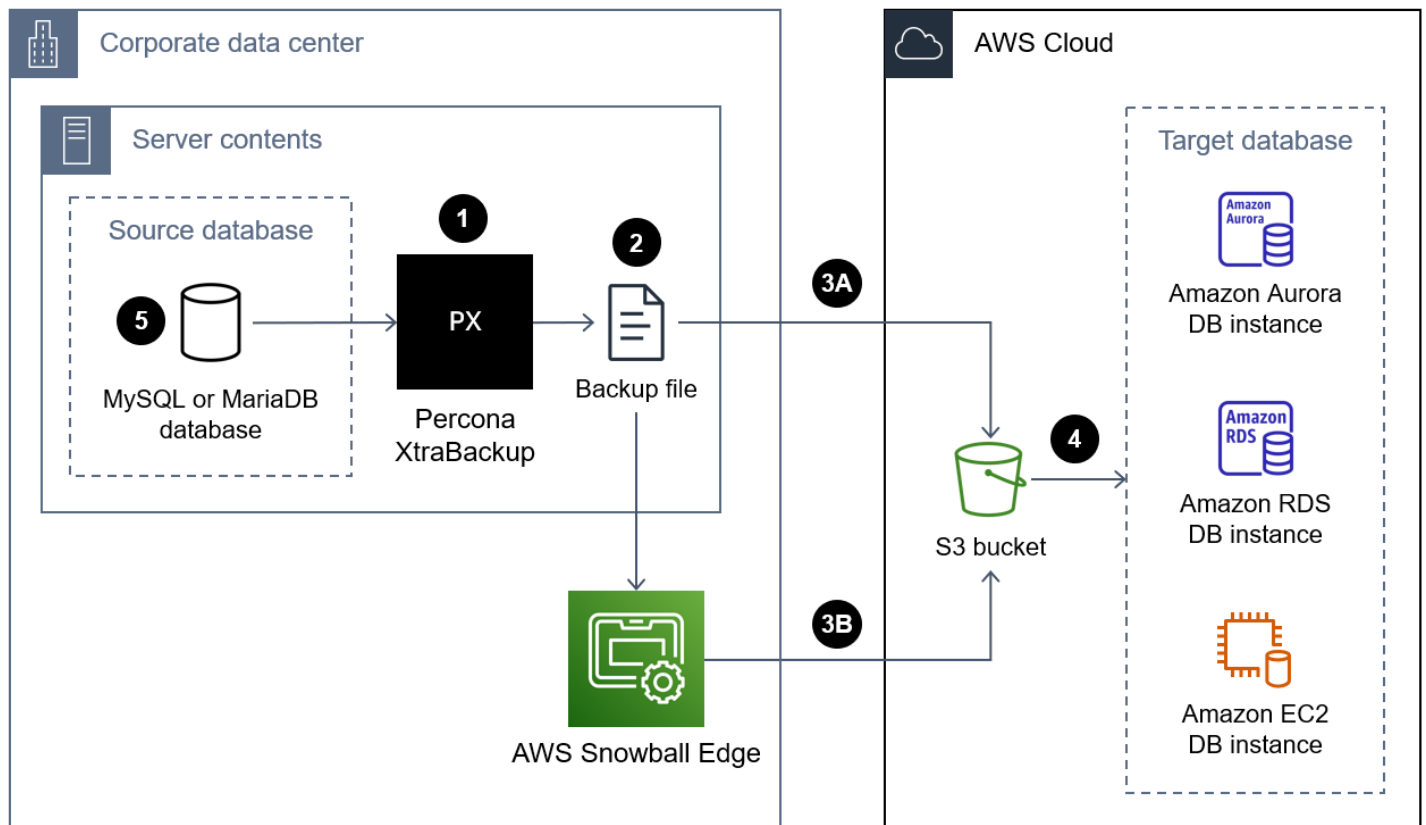
Important

Percona n' XtraBackup est pas pris en charge pour les versions 10.3 ou ultérieures de MariaDB et n'est que partiellement pris en charge pour les versions 10.1 et 10.2.

[Percona XtraBackup](#) est un logiciel de sauvegarde à chaud open source courant pour MySQL et MariaDB qui effectue des sauvegardes non bloquantes pour les moteurs de stockage InnoDB et XtraDB. Il fonctionne avec les serveurs MySQL ou MariaDB. Pour plus d'informations sur l'outil et certaines de ses fonctionnalités et avantages, voir [À propos de Percona XtraBackup](#) dans la documentation Percona XtraBackup .

Cet outil utilise l'approche de migration physique. Il copie directement le répertoire de données MySQL ou MariaDB et les fichiers qu'il contient. Pour les bases de données volumineuses, telles que celles de plus de 100 Go, cela peut fournir un temps de restauration nettement supérieur à celui de certains autres outils. Vous créez une sauvegarde de la base de données source sur site, vous migrez les fichiers de sauvegarde vers le cloud, puis vous restaurez la sauvegarde sur la nouvelle instance de base de données cible.

Le schéma suivant montre les étapes de haut niveau impliquées dans la migration d'une base de données à l'aide d'un fichier de XtraBackup sauvegarde Percona. En fonction de la taille du fichier de sauvegarde, deux options sont disponibles pour transférer la sauvegarde vers un bucket Amazon Simple Storage Service (Amazon S3) situé dans le AWS Cloud



Voici les étapes à suivre pour utiliser Percona pour XtraBackup migrer une base de données vers : AWS Cloud

1. Installez Percona XtraBackup sur le serveur local. Si vous utilisez Amazon Aurora MySQL version 2 ou Amazon RDS, consultez [Installation de Percona 2.4 XtraBackup](#). Si vous utilisez Amazon Aurora MySQL version 3, consultez la section Installation de [Percona XtraBackup 8.0](#) dans la documentation Percona XtraBackup.
2. Créez une sauvegarde complète de la base de données source MySQL ou MariaDB. Pour les instructions relatives à Percona XtraBackup 2.4, voir [Sauvegarde complète](#). Pour obtenir des instructions relatives à Percona XtraBackup 8.0, voir [Création d'une sauvegarde complète](#).
3. Transférez les fichiers de sauvegarde sur Internet à l'aide d'un service ou d'un outil approuvé dans votre organisation, tel que le suivant :
 - [AWS Site-to-Site VPN](#)
 - [AWS Client VPN](#)
 - [AWS Direct Connect](#)

- [Amazon S3 File Gateway](#) (pour plus d'informations, consultez [Utilisation d'Amazon S3 File Gateway pour transférer des fichiers de sauvegarde](#) ce guide.)
 - [AWS Command Line Interface \(AWS CLI\)](#)
4. À partir du compartiment Amazon S3, restaurez les fichiers de sauvegarde sur l'instance de base de données cible. Pour obtenir des instructions, veuillez consulter les sections suivantes :
- Pour l' MySQL-Compatible édition Aurora, consultez la section [Migration de données depuis MySQL à l'aide d'un compartiment Amazon S3](#) dans la documentation Amazon RDS.
 - Pour Amazon RDS for MySQL ou Amazon EC2, [consultez Importation de données dans une instance de base de données MySQL](#).
 - Pour Amazon RDS pour MariaDB ou pour Amazon EC2, [consultez Importation de données dans une instance de base de données MariaDB](#).
5. (Facultatif) Vous pouvez configurer la réplication entre la base de données source et l'instance de base de données cible. Vous pouvez utiliser la réplication des journaux binaires (binlog) pour réduire les temps d'arrêt. Pour plus d'informations, consultez les ressources suivantes :
- [Configuration de la source de réplication](#) dans la documentation MySQL
 - Pour Amazon Aurora, consultez les informations suivantes :
 - [Synchronisation du cluster de base de données Amazon Aurora MySQL avec la base de données MySQL à l'aide de la réplication](#) dans la documentation Aurora
 - [Utilisation de la réplication binlog dans Amazon Aurora](#) dans la documentation d'Aurora
 - Pour Amazon RDS, consultez les rubriques suivantes :
 - [Utilisation de la réplication MySQL](#) dans la documentation Amazon RDS
 - [Utilisation de la réplication MariaDB](#) dans la documentation Amazon RDS
 - Pour Amazon EC2, consultez les rubriques suivantes :
 - [Configuration de la réplication basée sur la position du fichier journal binaire](#) dans la documentation MySQL
 - [Configuration des répliques](#) dans la documentation MySQL
 - [Configuration de la réplication](#) dans la documentation MariaDB

Avantages

- Percona XtraBackup utilisant une approche de migration physique, le processus de restauration ~~est généralement plus rapide que les outils utilisant une approche de migration logique. Cela est~~

dû au fait que les performances sont limitées par le débit du disque ou du réseau plutôt que par les ressources informatiques nécessaires au traitement des données.

- Le processus de restauration étant une copie directe des fichiers du compartiment S3 vers l'instance de base de données cible, les XtraBackup fichiers Percona sont généralement restaurés plus rapidement que les fichiers de sauvegarde créés avec d'autres outils.
- Percona XtraBackup est adaptable. Par exemple, il prend en charge plusieurs threads pour vous aider à copier des fichiers plus rapidement et prend en charge la compression pour réduire la taille de la sauvegarde.

Limitations

- La sauvegarde hors ligne n'est pas possible car Percona XtraBackup doit avoir accès au serveur de base de données source.
- Percona ne XtraBackup peut être utilisé que sur des systèmes dotés d'architectures système identiques. Par exemple, il n'est pas possible de restaurer une sauvegarde d'une base de données source exécutée sur Intel pour Windows Server sur un serveur cible ARM pour Linux.
- Percona XtraBackup n'est pas pris en charge pour MariaDB version 10.3 ou ultérieure, et il n'est que partiellement pris en charge pour MariaDB versions 10.2 et 10.1. Pour plus d'informations, voir [XtraBackup Présentation de Percona : compatibilité avec MariaDB dans la base](#) de connaissances MariaDB.
- Vous ne pouvez pas utiliser Percona XtraBackup pour restaurer une base de données MariaDB source sur une instance de base de données MySQL cible, telle qu'Amazon RDS for MySQL ou Aurora. MySQL-Compatible
- Le volume total de données et le nombre d'objets que vous pouvez stocker dans un compartiment S3 sont illimités, mais la taille maximale du fichier est de 5 To. Si votre fichier de sauvegarde dépasse 5 To, vous pouvez le diviser en plusieurs fichiers plus petits.
- Lorsque le `innodb_file_per_table` paramètre est désactivé, Percona XtraBackup ne prend pas en charge les sauvegardes partielles qui utilisent `--tables--tables-exclude`, `--tables-file`, `--databases--databases-exclude`, ou `--databases-file`. Pour plus d'informations sur Percona XtraBackup version 2.4, voir [Sauvegardes partielles](#). Pour plus d'informations sur Percona XtraBackup version 8.0, voir [Création d'une sauvegarde partielle](#).

Bonnes pratiques

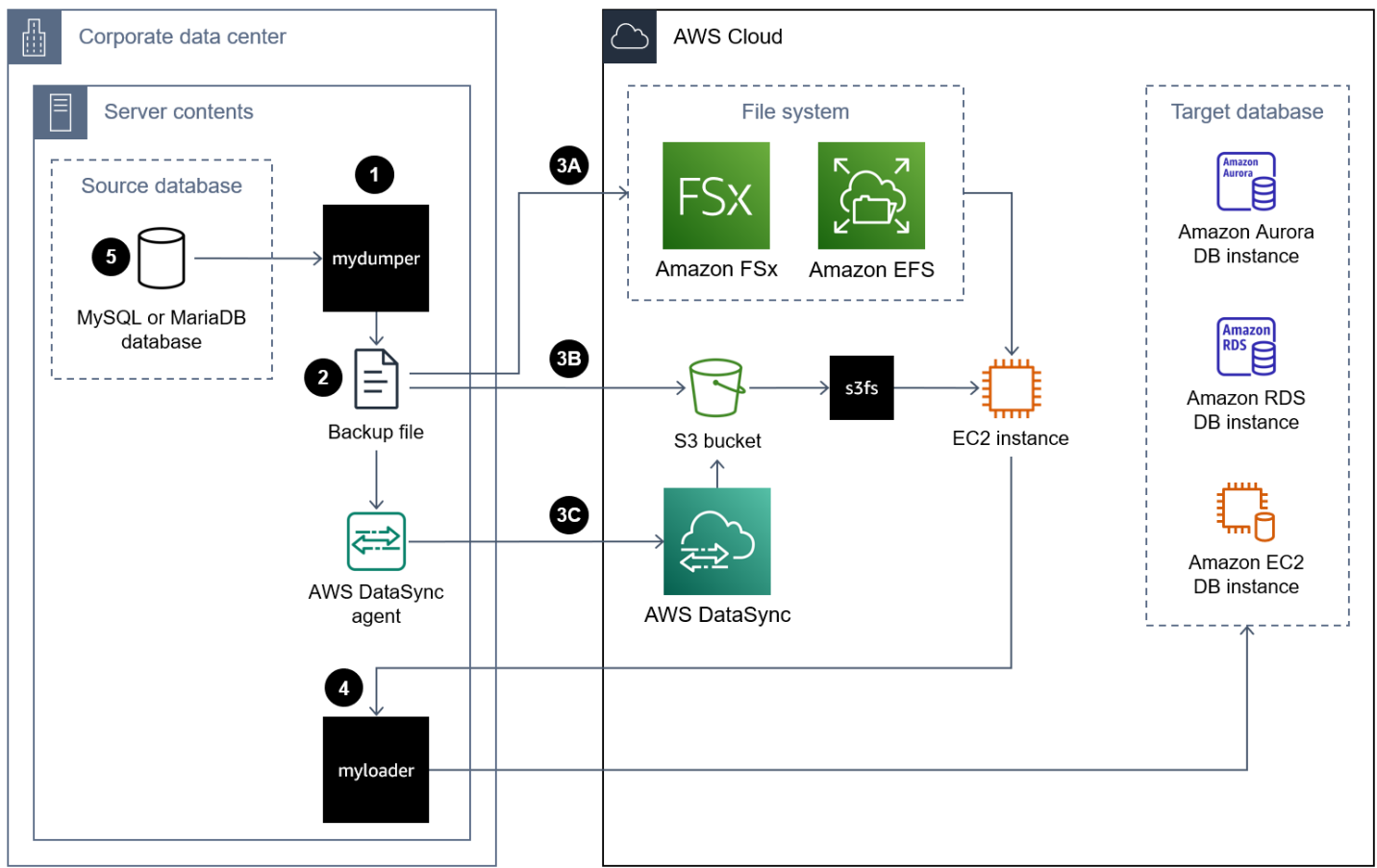
- Pour améliorer les performances du processus de sauvegarde, procédez comme suit :
 - Copiez plusieurs fichiers en parallèle en utilisant `--parallel= <threads>`
 - Comprimez plusieurs fichiers en parallèle en utilisant `--compress-threads= <threads>`
 - Augmentez la mémoire en utilisant `--use-memory= <size>`
 - [Chiffrez plusieurs fichiers en parallèle en utilisant `--encrypt-threads= <threads>`](#)
- Assurez-vous qu'il y a suffisamment d'espace sur le serveur source pour enregistrer les fichiers de sauvegarde de la base de données.
- Générez la sauvegarde de base de données avec le fichier au format Percona xstream (.xstream). Pour plus d'informations, consultez [la présentation du binaire xstream](#) dans la documentation de XtraBackup Percona.

MyDumper

[MyDumper](#)(GitHub) est un outil de migration logique open source composé de deux utilitaires :

- MyDumper exporte une sauvegarde cohérente des bases de données MySQL. Il prend en charge la sauvegarde de la base de données en utilisant plusieurs threads parallèles, jusqu'à un thread par cœur de processeur disponible.
- myloader lit les fichiers de sauvegarde créés par MyDumper, se connecte à l'instance de base de données cible, puis restaure la base de données.

Le schéma suivant montre les étapes de haut niveau de la migration d'une base de données à l'aide d'un fichier MyDumper de sauvegarde. Ce schéma d'architecture inclut trois options pour migrer le fichier de sauvegarde du centre de données sur site vers une instance EC2 dans le AWS Cloud



Voici les étapes à suivre pour MyDumper migrer une base de données vers AWS Cloud :

1. Installer MyDumper et myloader. Pour obtenir des instructions, voir [Comment installer mydumper/myloader](#) (GitHub).
2. MyDumper À utiliser pour créer une sauvegarde de la base de données source MySQL ou MariaDB. Pour obtenir des instructions, reportez-vous [à la section Comment utiliser MyDumper](#).
3. Déplacez le fichier de sauvegarde vers une instance EC2 dans le en AWS Cloud utilisant l'une des approches suivantes :

Approche 3A — Montez un système de fichiers [Amazon FSx](#) ou [Amazon Elastic File System \(Amazon EFS\)](#) sur le serveur local qui exécute votre instance de base de données. Vous pouvez utiliser AWS Direct Connect ou Site-to-Site VPN pour établir la connexion. Vous pouvez sauvegarder directement la base de données sur le partage de fichiers monté, ou vous pouvez effectuer la sauvegarde en deux étapes en sauvegardant la base de données sur un système de fichiers local, puis en la téléchargeant sur le volume FSx ou EFS monté. Ensuite, montez le

système de fichiers Amazon FSx ou Amazon EFS, qui est également monté sur le serveur local, sur une instance EC2.

Approche 3B — Utilisez le AWS CLI AWS SDK ou l'API REST Amazon S3 pour déplacer directement le fichier de sauvegarde du serveur sur site vers un compartiment S3. Si le compartiment S3 cible se trouve dans un Région AWS endroit éloigné du centre de données, vous pouvez utiliser [Amazon S3 Transfer Acceleration](#) pour transférer le fichier plus rapidement. Utilisez le système de fichiers [s3fs-fuse](#) pour monter le compartiment S3 sur l'instance EC2.

Approche 3C — Installez l' AWS DataSync agent dans le centre de données local, puis utilisez-le [AWS DataSync](#) pour déplacer le fichier de sauvegarde vers un compartiment Amazon S3. Utilisez le système de fichiers [s3fs-fuse](#) pour monter le compartiment S3 sur l'instance EC2.

 Note

Vous pouvez également utiliser Amazon S3 File Gateway pour transférer les fichiers de sauvegarde de base de données volumineux vers un compartiment S3 du AWS Cloud. Pour plus d'informations, consultez [Utilisation d'Amazon S3 File Gateway pour transférer des fichiers de sauvegarde](#) dans ce guide.

4. Utilisez myloader pour restaurer la sauvegarde sur l'instance de base de données cible. Pour obtenir des instructions, consultez [myloader usage](#) (GitHub).
5. (Facultatif) Vous pouvez configurer la réplication entre la base de données source et l'instance de base de données cible. Vous pouvez utiliser la réplication des journaux binaires (binlog) pour réduire les temps d'arrêt. Pour plus d'informations, consultez les ressources suivantes :
 - [Configuration de la source de réplication](#) dans la documentation MySQL
 - Pour Amazon Aurora, consultez les informations suivantes :
 - [Synchronisation du cluster de base de données Amazon Aurora MySQL avec la base de données MySQL à l'aide de la réplication](#) dans la documentation Aurora
 - [Utilisation de la réplication binlog dans Amazon Aurora](#) dans la documentation d'Aurora
 - Pour Amazon RDS, consultez les rubriques suivantes :
 - [Utilisation de la réplication MySQL](#) dans la documentation Amazon RDS
 - [Utilisation de la réplication MariaDB](#) dans la documentation Amazon RDS
 - Pour Amazon EC2, consultez les rubriques suivantes :

- [Configuration de la réplication basée sur la position du fichier journal binaire](#) dans la documentation MySQL
- [Configuration des répliques](#) dans la documentation MySQL
- [Configuration de la réplication](#) dans la documentation MariaDB

Avantages

- MyDumper prend en charge le parallélisme en utilisant le multithreading, qui améliore la vitesse des opérations de sauvegarde et de restauration.
- MyDumper évite les routines coûteuses de conversion de jeux de caractères, ce qui contribue à garantir l'efficacité du code.
- MyDumper simplifie l'affichage et l'analyse des données en utilisant le vidage de fichiers séparés pour les tables et les métadonnées.
- MyDumper gère les instantanés sur tous les threads et fournit des positions précises des journaux principaux et secondaires.
- Vous pouvez utiliser les expressions régulières compatibles Perl (PCRE) pour spécifier s'il faut inclure ou exclure des tables ou des bases de données.

Limitations

- Vous pouvez choisir un autre outil si vos processus de transformation de données nécessitent des fichiers de vidage intermédiaires au format plat au lieu du format SQL.
- myloader n'importe pas automatiquement les comptes utilisateur de la base de données. Si vous restaurez la sauvegarde sur Amazon RDS ou Aurora, créez les utilisateurs avec les autorisations requises. Pour plus d'informations, consultez la section [Privilèges du compte utilisateur principal](#) dans la documentation Amazon RDS. Si vous restaurez la sauvegarde sur une instance de base de données Amazon EC2, vous pouvez exporter manuellement les comptes utilisateur de la base de données source et les importer dans l'instance EC2.

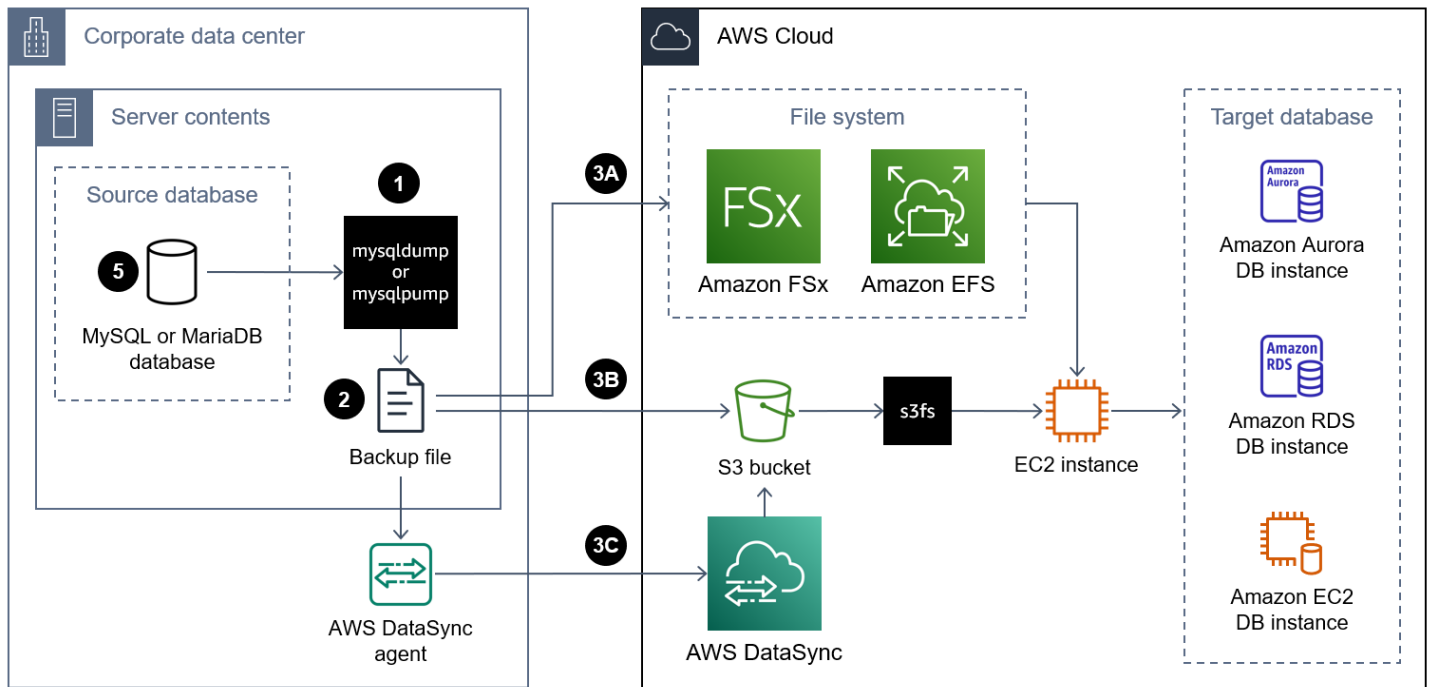
Bonnes pratiques

- Configurez MyDumper pour diviser chaque table en segments, par exemple 10 000 lignes dans chaque segment, et écrivez chaque segment dans un fichier distinct. Cela permet d'importer les données en parallèle ultérieurement.
- Si vous utilisez le moteur InnoDB, utilisez l'option `--trx-consistency-only` pour minimiser le verrouillage.
- L'exportation MyDumper de la base de données peut nécessiter beaucoup de lecture et le processus peut avoir un impact sur les performances globales de la base de données de production. Si vous disposez d'une instance de base de données répliquée, exécutez le processus d'exportation à partir de la réplique. Avant d'exécuter l'exportation depuis le réplica, arrêtez le thread SQL de réplication. Cela permet au processus d'exportation de s'exécuter plus rapidement.
- N'exportez pas la base de données pendant les heures de pointe. En évitant les heures de pointe, vous pouvez stabiliser les performances de votre base de données de production principale lors de l'exportation de la base de données.
- Amazon RDS for MySQL ne prend pas en charge `keyring_aws` le plug-in. Pour plus d'informations, consultez la section [Problèmes et limites connus](#). Pour migrer les tables chiffrées sur site vers l'instance Amazon RDS, vous devez supprimer `ENCRYPTION` ou supprimer la syntaxe dans les scripts `DEFAULT ENCRYPTION` de sauvegarde. `CREATE TABLE` Pour le chiffrement au repos, vous pouvez utiliser une clé AWS Key Management Service (AWS KMS). Pour plus d'informations, consultez [Chiffrer des ressources Amazon RDS](#).

mysqldump et mysqlpump

[mysqldump et mysqlpump](#) sont des outils de sauvegarde de base de données natifs pour MySQL. MariaDB supporte `mysqldump` mais ne supporte pas `mysqlpump`. Ces deux outils créent des sauvegardes logiques et font partie des programmes clients MySQL. `mysqldump` prend en charge le traitement monothread. `mysqlpump` prend en charge le traitement parallèle des bases de données et des objets au sein des bases de données, afin d'accélérer le processus de vidage. Il a été introduit dans la version 5.7.8 de MySQL. `mysqlpump` a été supprimé dans la version 8.4 de MySQL.

Le schéma suivant montre les étapes de haut niveau impliquées dans la migration d'une base de données à l'aide d'un fichier de sauvegarde `mysqldump` ou `mysqlpump`.



Voici les étapes à suivre pour utiliser mysqldump ou mysqlpump pour migrer une base de données vers : AWS Cloud

1. Installez MySQL Shell sur le serveur local. Pour obtenir des instructions, consultez la section [Installation de MySQL Shell](#) dans la documentation MySQL. Ceci installe à la fois mysqldump et mysqlpump.
2. À l'aide de mysqldump ou mysqlpump, créez une sauvegarde de la base de données source sur site. Pour obtenir des instructions, consultez [mysqldump et mysqlpump dans la documentation MySQL](#), ou [consultez Making Backups with mysqldump](#) dans la documentation MariaDB. Pour plus d'informations sur l'appel de programmes MySQL et la spécification d'options, consultez [Utilisation de programmes MySQL](#).
3. Déplacez le fichier de sauvegarde vers une instance EC2 dans le en AWS Cloud utilisant l'une des approches suivantes :

Approche 3A — Montez un système de fichiers [Amazon FSx](#) ou [Amazon Elastic File System \(Amazon EFS\)](#) sur le serveur local qui exécute votre instance de base de données. Vous pouvez utiliser AWS Direct Connect ou Site-to-Site VPN pour établir la connexion. Vous pouvez sauvegarder directement la base de données sur le partage de fichiers monté, ou vous pouvez effectuer la sauvegarde en deux étapes en sauvegardant la base de données sur un système de fichiers local, puis en la téléchargeant sur le volume FSx ou EFS monté. Ensuite, montez le

système de fichiers Amazon FSx ou Amazon EFS, qui est également monté sur le serveur local, sur une instance EC2.

Approche 3B — Utilisez le AWS CLI AWS SDK ou l'API REST Amazon S3 pour déplacer directement le fichier de sauvegarde du serveur sur site vers un compartiment S3. Si le compartiment S3 cible se trouve dans un Région AWS endroit éloigné du centre de données, vous pouvez utiliser [Amazon S3 Transfer Acceleration](#) pour transférer le fichier plus rapidement. Utilisez le système de fichiers [s3fs-fuse](#) pour monter le compartiment S3 sur l'instance EC2.

Approche 3C — Installez l' AWS DataSync agent dans le centre de données local, puis utilisez-le [AWS DataSync](#) pour déplacer le fichier de sauvegarde vers un compartiment Amazon S3. Utilisez le système de fichiers [s3fs-fuse](#) pour monter le compartiment S3 sur l'instance EC2.

 Note

Vous pouvez également utiliser Amazon S3 File Gateway pour transférer les fichiers de sauvegarde de base de données volumineux vers un compartiment S3 du AWS Cloud. Pour plus d'informations, consultez [Utilisation d'Amazon S3 File Gateway pour transférer des fichiers de sauvegarde](#) dans ce guide.

4. Utilisez la méthode de restauration native pour restaurer la sauvegarde sur la base de données cible. Pour obtenir des instructions, consultez [Reloading SQL-Format Backups](#) dans la documentation MySQL ou consultez [Restaurer des données à partir de fichiers Dump](#) dans la documentation MariaDB.
5. (Facultatif) Vous pouvez configurer la réplication entre la base de données source et l'instance de base de données cible. Vous pouvez utiliser la réplication des journaux binaires (binlog) pour réduire les temps d'arrêt. Pour plus d'informations, consultez les ressources suivantes :
 - [Configuration de la source de réplication](#) dans la documentation MySQL
 - Pour Amazon Aurora, consultez les informations suivantes :
 - [Synchronisation du cluster de base de données Amazon Aurora MySQL avec la base de données MySQL à l'aide de la réplication](#) dans la documentation Aurora
 - [Utilisation de la réplication binlog dans Amazon Aurora](#) dans la documentation d'Aurora
 - Pour Amazon RDS, consultez les rubriques suivantes :
 - [Utilisation de la réplication MySQL](#) dans la documentation Amazon RDS
 - [Utilisation de la réplication MariaDB](#) dans la documentation Amazon RDS

- Pour Amazon EC2, consultez les rubriques suivantes :
 - [Configuration de la réplication basée sur la position du fichier journal binaire](#) dans la documentation MySQL
 - [Configuration des répliques](#) dans la documentation MySQL
 - [Configuration de la réplication](#) dans la documentation MariaDB

Avantages

- mysqldump et mysqlpump sont inclus dans l'installation de MySQL Server
- Les fichiers de sauvegarde générés par ces outils sont dans un format plus lisible.
- Avant de restaurer le fichier de sauvegarde, vous pouvez modifier le fichier .sql obtenu à l'aide d'un éditeur de texte standard.
- Vous pouvez sauvegarder une table, une base de données ou même une sélection de données spécifique.
- mysqldump et mysqlpump sont indépendants de l'architecture de la machine.

Limitations

- mysqldump est un processus de sauvegarde monothread. Les performances de sauvegarde sont bonnes pour les petites bases de données, mais elles peuvent devenir inefficaces lorsque la taille de sauvegarde est supérieure à 10 Go.
- Les fichiers de sauvegarde au format logique sont volumineux, en particulier lorsqu'ils sont enregistrés sous forme de texte, et leur création et leur restauration sont souvent lentes.
- La restauration des données peut être lente car la réapplication des instructions SQL dans l'instance de base de données cible implique un traitement intensif du disque I/O et du processeur pour l'insertion, la création d'index et l'application des contraintes d'intégrité référentielle.
- L'utilitaire mysqlpump n'est pas pris en charge pour les versions de MySQL antérieures à 5.7.8 ou pour les versions 8.4 et ultérieures.
- Par défaut, mysqlpump n'effectue pas de sauvegarde des bases de données du système, telles que `performance_schema` ou `sys`. Pour sauvegarder une partie de la base de données système, nommez-la explicitement dans la ligne de commande.
- mysqldump ne sauvegarde pas les instructions InnoDB. `CREATE TABLESPACE`

 Note

Les sauvegardes des instructions CREATE TABLESPACE et des bases de données système ne sont utiles que lorsque vous restaurez des sauvegardes de bases de données MySQL ou MariaDB sur une instance EC2. Ces sauvegardes ne sont pas utilisées pour Amazon RDS ou Aurora.

Bonnes pratiques

- Lorsque vous restaurez la sauvegarde de la base de données, désactivez les vérifications des clés, par exemple au niveau de la session dans la base de données cible. FOREIGN_KEY_CHECKS Cela augmente la vitesse de restauration.
- Assurez-vous que l'utilisateur de la base de données dispose de [privilèges](#) suffisants pour créer et restaurer la sauvegarde.

Sauvegarde fractionnée

Une stratégie de sauvegarde fractionnée consiste à migrer un serveur de base de données volumineux en divisant la sauvegarde en plusieurs parties. Vous pouvez utiliser différentes approches pour migrer chaque partie de la sauvegarde. Cela peut être la meilleure option pour les cas d'utilisation suivants :

- Grand serveur de base de données mais petites bases de données individuelles : cette approche est recommandée lorsque la taille totale du serveur de base de données est de plusieurs To, mais que la taille de chaque base de données utilisateur indépendante est inférieure à 1 To. Pour réduire la période de migration globale, vous pouvez migrer chaque base de données séparément et en parallèle.

Prenons l'exemple d'un serveur de base de données de 2 To sur site. Ce serveur est composé de quatre bases de données de 0,5 To chacune. Vous pouvez effectuer des sauvegardes de chaque base de données séparément. Lors de la restauration de la sauvegarde, vous pouvez soit restaurer toutes les bases de données d'une instance en parallèle, soit, si les bases de données sont indépendantes, vous pouvez restaurer chaque sauvegarde sur une instance distincte. Il est recommandé de restaurer des bases de données indépendantes sur des instances distinctes, au

lieu de les restaurer sur la même instance. Pour plus d'informations, consultez la section Meilleures pratiques de ce guide.

- Grand serveur de base de données mais petites tables de base de données individuelles : il s'agit d'une bonne approche lorsque la taille totale du serveur de base de données est de plusieurs To, mais que la taille de chaque table de base de données indépendante est inférieure à 1 To. Pour réduire la période de migration globale, vous pouvez migrer des tables indépendantes individuellement.

Prenons l'exemple d'une base de données mono-utilisateur de 1 To, qui est la seule base de données d'un serveur de base de données sur site. La base de données contient 10 tables, chacune d'une capacité de 100 Go. Vous pouvez effectuer des sauvegardes de chaque table séparément. Lors de la restauration de la sauvegarde, vous pouvez restaurer toutes les tables d'une instance en parallèle.

- Une base de données contient des tables de charge de travail transactionnelles et non transactionnelles. Comme dans le cas d'utilisation précédent, vous pouvez utiliser une approche de sauvegarde fractionnée lorsque vous avez des tables de charge de travail transactionnelles et non transactionnelles dans la même base de données.

Prenons l'exemple d'une base de données de 2 To composée de 0,5 To de tables de charge de travail critiques utilisées pour le traitement des transactions en ligne (OLTP) et d'une seule table de 1,5 To utilisée pour archiver les anciennes données. Vous pouvez effectuer la sauvegarde de tous les objets de base de données, à l'exception de la table d'archive, dans le cadre d'une sauvegarde cohérente et unique. Ensuite, vous effectuez une autre sauvegarde séparée de la table d'archive uniquement. Pour la sauvegarde de la table d'archive, vous pouvez également envisager d'effectuer plusieurs sauvegardes parallèles en utilisant des conditions pour diviser le nombre de lignes du fichier de sauvegarde. Voici un exemple :

```
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
```

Lors de la restauration des fichiers de sauvegarde, vous pouvez restaurer la sauvegarde de la charge de travail transactionnelle et la sauvegarde de la table d'archivage en parallèle.

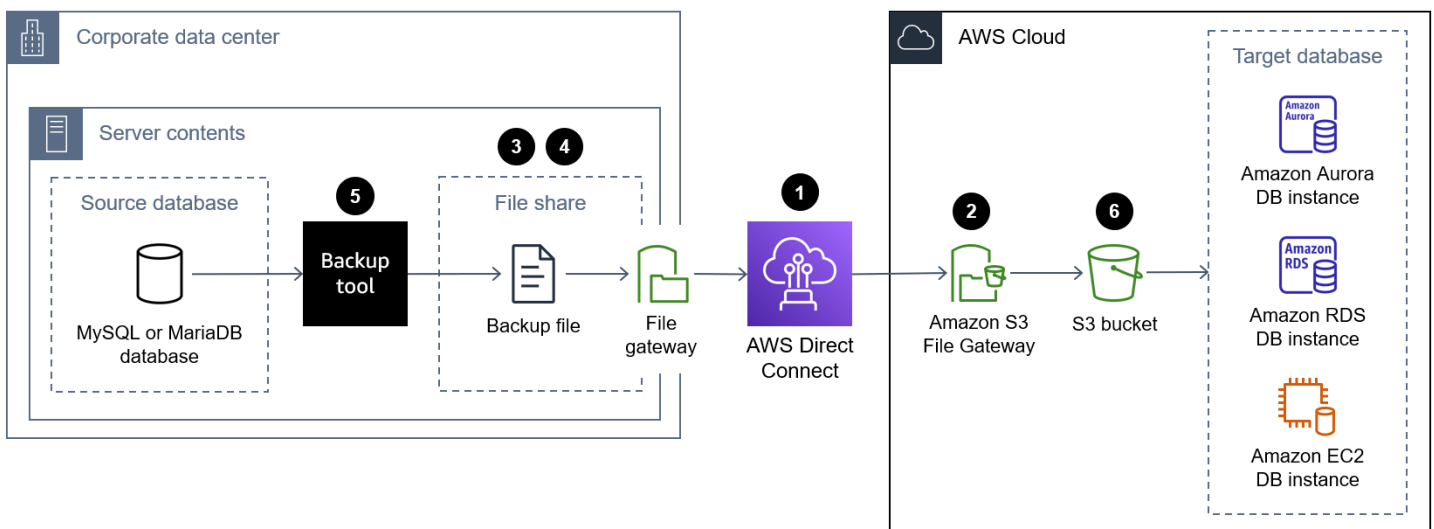
- Limitations des ressources de calcul : si les ressources de calcul du serveur local sont limitées, telles que le processeur, la mémoire ou le disque I/O, cela peut affecter la stabilité et les performances lors de la sauvegarde. Au lieu de faire une sauvegarde complète, vous pouvez la diviser en plusieurs parties.

Par exemple, un serveur de production sur site peut être chargé de charges de travail importantes et disposer de ressources CPU limitées. Si vous effectuez une sauvegarde en une seule exécution d'une base de données de plusieurs téraoctets sur ce serveur, cela peut consommer des ressources CPU supplémentaires et affecter négativement le serveur de production. Au lieu d'effectuer la sauvegarde complète de la base de données, divisez-la en plusieurs parties, telles que 2 à 3 tables chacune.

Utilisation d'Amazon S3 File Gateway pour transférer des fichiers de sauvegarde

[Amazon S3 File Gateway](#) connecte votre environnement sur site à Amazon Simple Storage Service (Amazon S3) via une interface de fichiers afin que vous puissiez stocker et récupérer des objets Amazon S3 en utilisant des protocoles de fichiers standard, tels que le Network File System (NFS) et le Server Message Block (SMB). Il est conçu pour être une solution rentable et évolutive pour le stockage de données dans le cloud. Comme vous pouvez l'utiliser pour stocker des fichiers de sauvegarde de base de données, ce service peut vous aider à migrer de grandes bases de données locales vers le AWS Cloud. Par exemple, vous pouvez utiliser Amazon S3 File Gateway et votre outil de sauvegarde de base de données préféré pour sauvegarder la grande base de données MySQL ou MariaDB directement dans un compartiment Amazon S3. Vous pouvez ensuite monter le compartiment S3 sur l'instance cible et restaurer la sauvegarde.

Le schéma suivant montre les étapes de haut niveau nécessaires lors de l'utilisation d'Amazon S3 File Gateway pour transférer le fichier de sauvegarde d'une base de données sur site vers un compartiment S3 du AWS Cloud.



Voici les étapes à suivre pour utiliser Amazon S3 File Gateway afin de transférer un fichier de sauvegarde de base de données d'un centre de données sur site vers un compartiment S3 dans le AWS Cloud :

1. Connectez le centre de données sur site au AWS Cloud en utilisant un service tel que AWS Direct Connect AWS Site-to-Site VPN ou en utilisant une connexion Internet publique.

2. Créez une passerelle de fichiers S3. Pour obtenir des instructions, consultez [la section Création de votre passerelle](#).
3. Créez un partage de fichiers NFS ou SMB hébergé par la passerelle de fichiers S3. Pour obtenir des instructions, voir [Création d'un partage de fichiers](#).
4. Montez le partage de fichiers NFS ou SMB sur le serveur local qui héberge votre base de données MySQL ou MariaDB. Pour obtenir des instructions, voir [Monter et utiliser votre partage de fichiers](#).
5. Sauvegardez la base de données MySQL ou MariaDB locale dans le répertoire où le partage de fichiers NFS est monté. Vous pouvez utiliser tous les outils de sauvegarde décrits dans ce guide.
6. Restaurez la sauvegarde de base de données sur l'instance de base de données cible en utilisant l'une des approches décrites dans ce guide.

Avantages

- En produisant des sauvegardes de base de données directement dans le compartiment S3 et en restaurant la sauvegarde sur l'instance de base de données cible directement à partir du même compartiment S3, vous pouvez accélérer considérablement le processus de migration de bout en bout.
- Les fichiers de sauvegarde de base de données sont stockés de manière durable dans Amazon S3, et vous choisissez la politique de gestion du cycle de vie et la classe de stockage S3.

Limitations

Les limites suivantes s'appliquent à l'utilisation des partages de fichiers Amazon S3 File Gateway :

- Le nombre maximum de partages de fichiers par passerelle est de 50.
- Pour éviter les conflits de lecture et d'écriture lorsque plusieurs partages de fichiers utilisent le même compartiment S3, vous devez configurer chaque partage de fichiers pour utiliser un nom de préfixe unique.
- La taille maximale d'un fichier individuel est de 5 To, soit la taille maximale de tout objet individuel dans Amazon S3.
- La longueur maximale du chemin est de 1024 caractères.
- Les ACL Windows ne sont prises en charge que sur les partages de fichiers activés pour Active Directory lorsque vous utilisez des clients Windows SMB pour accéder aux partages de fichiers.

- Amazon S3 File Gateway prend en charge un maximum de 10 entrées ACL pour chaque fichier et répertoire.
- Les paramètres ACL racine des partages de fichiers SMB se trouvent uniquement sur la passerelle. Ces paramètres sont persistants lors des mises à jour et des redémarrages de la passerelle.

 Note

Si vous configurez les ACL sur la racine plutôt que sur le dossier parent situé sous la racine, les autorisations ACL ne sont pas persistantes dans Amazon S3.

Bonnes pratiques

[Pour plus d'informations sur les meilleures pratiques relatives à Amazon S3 File Gateway, consultez la documentation relative à la passerelle de fichiers S3.](#)

Meilleures pratiques pour la migration de grandes bases de données MySQL et MariaDB

Outre les meilleures pratiques spécifiques aux outils répertoriées pour chaque option de migration, consultez les meilleures pratiques générales suivantes. Ces bonnes pratiques s'appliquent lors de la migration de grandes bases de données MySQL et MariaDB de plusieurs téraoctets, quel que soit l'outil sélectionné :

- Assurez-vous qu'il y a suffisamment d'espace dans les bases de données source et de destination pour effectuer et restaurer la sauvegarde.
- Ne créez pas d'index secondaires sur l'instance de base de données cible tant que la migration n'est pas terminée. Les index secondaires ajoutent des frais de maintenance supplémentaires lors de l'importation et peuvent ralentir le processus d'importation.
- Si vous utilisez une approche multithread, choisissez le bon nombre de threads. Pour l'exportation, nous vous recommandons d'utiliser un thread pour chaque cœur de processeur. Pour l'importation, nous vous recommandons d'utiliser un thread pour deux cœurs de processeur.
- Les vidages de données sont souvent effectués à partir de serveurs de base de données actifs faisant partie d'un environnement de production critique. Si le vidage des données affecte sérieusement les performances et que cela n'est pas acceptable dans votre environnement, envisagez l'une des solutions suivantes :
 - Le serveur source possède des répliques, vous pouvez transférer les données de l'une des répliques.
 - Le serveur source est couvert par des procédures de sauvegarde régulières :
 - Si le format de sauvegarde convient à l'importation directe dans la base de données cible, utilisez les données de sauvegarde comme entrée pour le processus d'importation.
 - Si le format de sauvegarde ne convient pas à l'importation directe dans la base de données cible, utilisez la sauvegarde pour approvisionner une base de données temporaire et en extraire les données.
 - Si les répliques et les sauvegardes ne sont pas disponibles :
 - Effectuez des vidages en dehors des heures de pointe, lorsque le trafic de production est au plus bas.
 - Réduisez la simultanéité des opérations de vidage afin que le serveur dispose d'une capacité de réserve suffisante pour gérer le trafic de production.

- Créez des dumps de bases de données créées par les utilisateurs uniquement.
- Recréez les utilisateurs dans la base de données cible et configurez leurs autorisations. Pour plus d'informations, consultez [Gestion des identités et des accès pour Amazon RDS](#), [Gestion des identités et des accès pour Amazon Aurora](#) ou Gestion des [identités et des accès pour Amazon EC2](#).
- Lors de la migration d'un serveur de base de données volumineux composé de plusieurs bases de données indépendantes, créez une instance distincte pour chaque base de données. Cela vous permet de gérer la base de données plus efficacement et d'améliorer le provisionnement des ressources, tandis que les ressources de calcul distinctes peuvent améliorer les performances de la base de données.

Ressources

AWS Directives prescriptives

- [Guide de portefeuille pour les AWS grandes migrations](#)
- [Migration strategy for relational databases](#)
- [Migrer une base de données MySQL sur site vers Amazon RDS for MySQL](#)
- [Configurer la réplication des données entre Amazon RDS for MySQL et MySQL sur EC2 Amazon à l'aide de GTID](#)
- [Migrer une base de données MariaDB sur site vers Amazon RDS for MariaDB à l'aide d'outils natifs](#)

AWS articles de blog

- [Bonnes pratiques de sécurité pour les instances Amazon RDS for MySQL et MariaDB](#)
- [Migrer MariaDB autogéré vers Amazon Aurora MySQL](#)

Ressources pour restaurer la sauvegarde

- [Création d'un compartiment](#) (documentation Amazon S3)
- [Connexion à votre instance Linux via SSH](#) (EC2 documentation Amazon)
- [Configuration du AWS CLI](#) (AWS CLI documentation)
- [commande de synchronisation](#) (référence de AWS CLI commande)
- [Création d'une politique IAM pour accéder aux ressources Amazon S3](#) (documentation Aurora)
- [Conditions requises pour le cluster](#) de bases de données (documentation Aurora)
- [Utilisation de groupes de sous-réseaux de base](#) de données (documentation Aurora)
- [Création d'un groupe de sécurité VPC pour un cluster de base de données privé](#) (documentation Aurora)
- [Restauration d'un cluster de base de données Aurora MySQL à partir d'un compartiment Amazon S3](#) (documentation Aurora)
- [Configuration de la réplication avec MySQL ou un autre cluster](#) de base de données Aurora (documentation Aurora)
- [Procédure rds_set_external_master](#) (documentation Amazon RDS)

- [procédure rds_start_replication](#) (documentation Amazon RDS)

AWS commercialisation

- [Amazon Aurora](#)
- [Amazon RDS for MariaDB](#)
- [Amazon RDS for MySQL](#)
- [Passerelle de fichiers Amazon S3](#)

Autres ressources

- [Percona XtraBackup](#)
- [MyDumper](#)
- [mysqldump](#)
- [pompe MySQL](#)

Historique du document

Le tableau suivant décrit les modifications importantes apportées à ce guide. Pour être averti des mises à jour à venir, abonnez-vous à un [fil RSS](#).

Modification	Description	Date
Disponibilité de mysqlpump	mysqlpump a été supprimé dans la version 8.4 de MySQL. Nous avons mis à jour les sections mysqldump et mysqlpump pour refléter ce changement de disponibilité.	21 novembre 2024
MyDumper meilleures pratiques	Nous avons mis à jour les meilleures pratiques MyDumper pour ajouter des informations sur la migration des tables de base de données chiffrées.	24 octobre 2024
Versions Percona XtraBackup	Dans la XtraBackup section Percona , nous avons mis à jour les instructions afin de refléter les versions de Percona XtraBackup prises en charge par Amazon Aurora MySQL et Amazon RDS.	3 août 2023
Publication initiale	—	6 avril 2023

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.